



Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C

Dissertationen

Heft Nr. 826

Alexander Schlichting

**Fahrzeuglokalisierung durch Automotive Laserscanner unter
Verwendung statischer Merkmale**

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325

ISBN 978-3-7696-5238-3

Diese Arbeit ist gleichzeitig veröffentlicht in:

Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover

ISSN 0174-1454, Nr. 346, Hannover 2018



Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C

Dissertationen

Heft Nr. 826

Fahrzeuglokalisierung durch Automotive Laserscanner unter Verwendung statischer Merkmale

Von der Fakultät für Bauingenieurwesen und Geodäsie

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

Vorgelegt von

M. Sc. Alexander Schlichting

Geboren am 20.05.1986 in Hamburg

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325

ISBN 978-3-7696-5238-3

Diese Arbeit ist gleichzeitig veröffentlicht in:

Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Universität Hannover

ISSN 0174-1454, Nr. 346, Hannover 2018

Adresse der DGK:



Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 • D – 80 539 München
Telefon +49 – 331 – 288 1685 • Telefax +49 – 331 – 288 1759
E-Mail post@dgk.badw.de • <http://www.dgk.badw.de>

Prüfungskommission:

Vorsitzender: apl. Prof. Dr.-Ing. Franz Rottensteiner

Referent: apl. Prof. Dr.-Ing. Claus Brenner

Korreferenten: Prof. Dr.-Ing. Steffen Schön
Prof. Dr.-Ing. Uwe Stilla

Tag der mündlichen Prüfung: 02.10.2018

© 2018 Bayerische Akademie der Wissenschaften, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,
die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen

Abstract

Vehicle localization is an important aspect of autonomous driving. It is common to combine the measurements of different sensors in a filter approach. To reach a high accuracy and integrity, redundant methods to estimate a position are applied. In this work automotive laser scanners are used for localization. In contrast to cameras, laser scanners work mostly independent of illumination conditions and directly provide 3D information. Three different methods for the purpose of vehicle localization are developed: localization by detected pole-like objects and planes, a scan image correlation and a sequence analysis for absolute positioning. These approaches are evaluated by their resulting localization accuracy and reliability. In addition, several automotive laser scanners in different mounting positions on the vehicle are compared to each other.

Pole-like objects and planes are segmented from the measurements of automotive laser scanners and assigned to a landmark reference. These references are measured by a mobile mapping system with a higher level of accuracy. The resulting completeness of the localization is 97 % with an accuracy of 0.08 m if the data of the last 50 m is processed. Landmark patterns are used to reduce the number of false positive detections. As a drawback the completeness hereby decreases to 7 %.

The scan image correlation compares the intensity and distance measurements of the automotive data to scan images generated by processing several data sets of a mobile mapping system. An important aspect is the detection of dynamic objects. A change detection algorithm and a classification approach are used to detect dynamics in the reference data. In case of the automotive data, dynamic objects are detected by machine learning methods. By only considering static objects, the completeness can be increased from 88 % to 93 % with a localization accuracy of 0.05 m.

Global localization is done by a sequence analysis, without using GNSS measurements. Every scan of a laser scanner, which is mounted vertical on the vehicle, along a reference trajectory is associated to a position and a specific label using a clustering method. In the localization step a number of successive scans is associated to labels by the same clustering. The position results from a comparison of the current sequence to the reference sequence. After a travelled distance of 120 m the position can be estimated with a completeness of 100 % and an accuracy below 2 m.

The resulting accuracies of the presented methods are relatively high compared to state of the art techniques. The reliability of the sequence analysis is even higher than current global localization approaches.

Keywords: Localization, Laser Scanning, LiDAR, Change Detection, Segmentation, Sequence Analysis

Kurzfassung

Ein elementarer Aspekt des autonomen Fahrens ist die Eigenlokalisierung des Fahrzeuges. In der Regel werden hierzu die Messungen verschiedener Sensoren in einem Filteransatz kombiniert. Um eine hohe Genauigkeit und Integrität zu ermöglichen, können auch redundante Positionslösungen in den Filter einfließen. In der vorliegenden Arbeit wird die Position aus den Messungen von Automotive-Laserscannern bestimmt. Diese haben im Vergleich zu Kameradaten den Vorteil, dass sie weitestgehend beleuchtungsunabhängig arbeiten und direkt 3D-Informationen liefern. Zur Lokalisierung werden drei verschiedene Methoden entwickelt: die Verwendung stangenförmiger Objekte und Ebenen, eine Scanbildkorrelation sowie eine Sequenzanalyse zur globalen Positionsbestimmung. Zur Bewertung der Methoden wird die Genauigkeit und Zuverlässigkeit der erzielten Lokalisierungsergebnisse untersucht. Hierbei kommen weiterhin unterschiedliche Automotive-Laserscanner in unterschiedlichen Anbringungen am Fahrzeug zum Einsatz, welche ebenfalls miteinander verglichen werden.

Zunächst werden stangenförmige Objekte und Ebenen aus den Messungen von Automotive-Laserscannern segmentiert und entsprechenden Referenz-Landmarken zugeordnet. Diese Referenzen stammen aus den Daten eines Mobile-Mapping-Systems mit übergeordneter Genauigkeit. Durch die Verwendung von Stangen und Ebenen kann die Position in 97 % der Fälle bestimmt werden, wenn die entlang der zuletzt zurückgelegten 50 m segmentierten Landmarken betrachtet werden. Die erzielbare Genauigkeit beträgt hierbei 0,08 m. Durch Landmarken-Muster kann weiterhin eine globale Position bestimmt sowie die Anzahl an False-Positive-Detektionen reduziert werden. Allerdings sinkt hierbei die Vollständigkeit auf lediglich 7 %.

Mit der Sequenzanalyse wird ein Verfahren vorgestellt, mit welchem unabhängig von GNSS-Messungen eine globale Position bestimmt werden kann. Hierzu werden jedem Laserscan eines am Fahrzeug vertikal montierten Laserscanners entlang einer Referenztrajektorie ein durch ein Clustering-Verfahren bestimmtes Label und eine Position zugeordnet. Im Lokalisierungsschritt können einer Abfolge von Laserscans ebenfalls mit demselben Verfahren Labels zugeordnet werden. Die Position ergibt sich anschließend durch den Abgleich der aktuellen Sequenz mit der Referenzsequenz. Es zeigt sich, dass bereits nach 120 m eine eindeutige Positionslösung mit einer Genauigkeit von unter 2 m korrekt bestimmt werden kann.

Die Scanbildkorrelation vergleicht die Messungen verschiedener Automotive-Laserscanner anhand der Distanz- und Intensitätswerte mit Referenz-Scanbildern. Die Referenzbilder werden aus mehreren Datensätzen eines Mobile-Mapping-Systems erstellt. Ein wichtiger Aspekt hierbei ist die Detektion dynamischer Objekte. Für die Referenzbilder geschieht dies durch eine Änderungsdetektion sowie durch eine Objekt-Klassifizierung. Für die Automotive-Daten werden hierfür Methoden des maschinellen Lernens angewandt. Werden lediglich statische Objekte verwendet, kann die Vollständigkeit des Verfahrens von 88 % auf 93 % erhöht werden, bei einer Genauigkeit von 0,05 m.

Ein Vergleich der erzielten Ergebnisse mit dem aktuellen Stand der Forschung zeigt, dass insbesondere die erzielten Genauigkeiten hoch sind. Die Zuverlässigkeit der Sequenzanalyse ist zudem höher als aktuelle Ansätze zur Bestimmung einer absoluten Position ohne die Verwendung von GNSS-Messungen.

Schlagworte: Lokalisierung, Laserscanning, LiDAR, Änderungsdetektion, Segmentierung, Sequenzanalyse

Inhaltsverzeichnis

1. Einleitung	9
1.1. Motivation	9
1.2. Ziel der Arbeit	11
1.3. Struktur der Arbeit	12
2. Grundlagen	15
2.1. Laserscanning	15
2.1.1. Distanzmessung	15
2.1.2. Scanmechanismus	17
2.1.3. Mobile Mapping	19
2.1.4. Registrierung von Punktwolken	20
2.1.5. Räumliche Datenstrukturen	24
2.1.6. Segmentierung in Punktwolken	25
2.2. Maschinelles Lernen	29
2.2.1. K-Means-Algorithmus	29
2.2.2. Bewertung einer Klassifizierung	30
2.2.3. Random Forests	30
2.2.4. Neuronale Netze	32
2.3. Filterverfahren zur Lokalisierung	39
2.3.1. Bayessche Filter	39
2.3.2. Kalman-Filter	40
2.3.3. Extended Kalman-Filter	41
3. Stand der Forschung	43
3.1. Detektion dynamischer Objekte	43
3.1.1. Änderungsdetektion in Punktwolken	43
3.1.2. Klassifizierung dynamischer Objekte	44
3.2. Fahrzeuglokalisierung	45
3.2.1. Feinlokalisierung	46
3.2.2. Globale Positionsbestimmung	47
4. Verwendete Sensoren und Daten	49
4.1. Mobile-Mapping-System Riegl VMX-250	49
4.2. Valeo Scala	50
4.3. Oxford Robotcar Dataset	51
4.4. Velodyne VLP-16	52
4.4.1. Kalibrierung des Systems	53

4.5. Vergleich der verwendeten Laserscanner	56
5. Detektion dynamischer Objekte	57
5.1. Änderungsdetektion	57
5.2. Klassifizierung dynamischer Objekte	60
5.2.1. Klassifizierung von Vegetation	60
5.2.2. Klassifizierung dynamischer Objekte in einzelnen Zeilenscans	63
5.2.3. Klassifizierung dynamischer Objekte in Scanbildern	64
6. Fahrzeug-Lokalisierung durch Automotive-Laserscanner	67
6.1. Verwendung stangenförmiger Objekte und Ebenen	67
6.1.1. Erstellung der Referenzkarte	68
6.1.2. Extraktion der Landmarken aus den Messungen eines horizontal montierten Automotive-Zeilenscanners	69
6.1.3. Extraktion der Landmarken aus den Messungen eines vertikal montierten Automotive-Laserscanners	71
6.1.4. Landmarken-Muster	72
6.2. Sequenzanalyse zur globalen Lokalisierung	75
6.2.1. Reduzierung der Dimension durch einen Autoencoder	76
6.2.2. Clustering der Daten	76
6.2.3. Bestimmung der Position	77
6.3. Scanbildkorrelation zur hochgenauen Lokalisierung	79
6.3.1. Automotive Daten eines Einzeilenscanners	79
6.3.2. Verwendung der Daten des Velodyne VLP-16	81
7. Ergebnisse der Fahrzeug-Lokalisierung	85
7.1. Verwendung stangenförmiger Objekte und Ebenen zur Fahrzeug-Lokalisierung . . .	85
7.1.1. Valeo Scala	86
7.1.2. Velodyne VLP-16	88
7.1.3. Auswertung der Zuordnung über Landmarken-Muster	91
7.1.4. Laufzeitbetrachtung	93
7.2. Sequenzanalyse	95
7.2.1. Automotive-Daten eines Einzeilen-Laserscanners	95
7.2.2. Oxford Robotcar Dataset	99
7.2.3. Velodyne VLP-16	100
7.2.4. Verwendung eines Filteransatzes	105
7.2.5. Laufzeitbetrachtung	107
7.3. Scanbildkorrelation	108
7.3.1. Automotive-Daten eines Einzeilen-Laserscanners	108
7.3.2. Velodyne VLP-16	111
7.3.3. Laufzeitbetrachtung	113
7.4. Diskussion der Ergebnisse	114

8. Zusammenfassung und Ausblick	117
8.1. Zusammenfassung	117
8.2. Ausblick	119
Anhang	120
A. Ergebnisse der Lokalisierung über stangenförmige Objekte und Ebenen	123
B. Ergebnisse der Sequenzanalyse durch einen Velodyne VLP-16	131
Abbildungsverzeichnis	135
Tabellenverzeichnis	139
Literaturverzeichnis	141
Lebenslauf	151
Danksagung	153

1. Einleitung

1.1. Motivation

Allein in Deutschland starben im Jahr 2017 mehr als 3000 Menschen im Straßenverkehr. Dabei sind über 90 % aller Unfälle auf menschliches Fehlverhalten zurückzuführen. Alexander Dobrindt, bis zum 24. Oktober 2017 deutscher Bundesminister für Verkehr und digitale Infrastruktur, sieht das automatisierte Fahren als größte Mobilitätsrevolution seit der Erfindung des Automobils, das die Fahrer deutlich entlastet und zu einer massiven Reduzierung kritischer Verkehrssituationen führen kann (DEKRA, 2017). Hierfür sollen verschiedene Fahrerassistenzsysteme (FAS) zu einer Erhöhung des Komforts und der Sicherheit führen. Zu nennen sind hier insbesondere die Detektion von Hindernissen, die Bestimmung der relativen Position anderer Verkehrsteilnehmer oder auch die automatisierte Lenkung des Fahrzeuges. Die notwendigen Messungen werden dafür, wie in Abbildung 1.1 gezeigt, von verschiedenen Sensoren ausgeführt: Mono- oder Stereo-Kameras, Infrarot-Kameras, Ultraschallsensoren, Radar oder Laserscanner. Laserscanner, welche auch als LiDAR-Sensoren (Light Detection and Ranging) bezeichnet werden, werden in aktuellen FAS insbesondere zur Detektion statischer und dynamischer Objekte eingesetzt (Benedikt Still, 2016). Durch die vergleichsweise hohen Genauigkeiten der LiDAR-Sensoren von wenigen Zentimetern bei einer Reichweite von etwa 100 Metern und einer Winkelauflösung von $0,25^\circ$ (Valeo, 2017, Velodyne LiDAR, 2017), kann die Umgebung des Fahrzeuges präzise erfasst werden. Ein weiterer Vorteil der Laserscanner gegenüber Kameras ist, dass ihre Messungen unabhängig von Belichtungsverhältnissen sind und so zuverlässig 3D-Informationen der Fahrzeugumgebung liefern. Wie LiDAR-Sensoren die Sicherheit in Fahrzeugen erhöhen können, verdeutlicht das Beispiel des sich am 7. Mai 2016 ereigneten Unfalls eines Tesla Model S in Florida, USA. Ein als Autopilot bezeichnetes FAS hatte einen LKW versehentlich als Verkehrsschild klassifiziert, wodurch sich ein für den Fahrer des Model S tödlicher Zusammenstoß ereignete. Aufgrund des hellen Himmels konnte das Kamerasystem den weißen Anhänger des LKW nicht detektieren. Zusätzlich war die Fahrhöhe des Anhängers

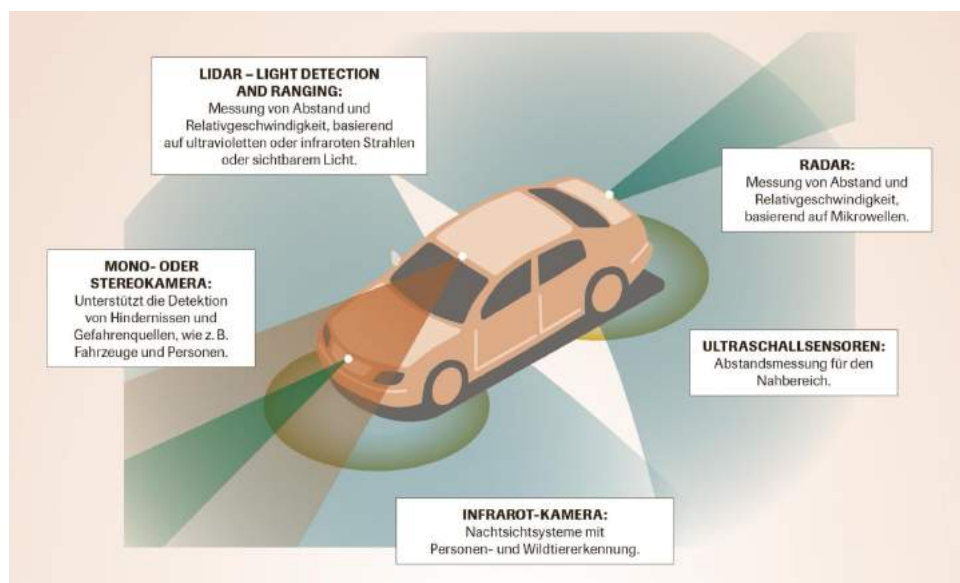


Abbildung 1.1.: Sensoren für Fahrerassistenzsysteme und deren Anwendung (Quelle: Verband der Automobilindustrie e.V. (2018)).

vergleichsweise hoch, wodurch das Radar-System das Objekt als Verkehrsschild klassifiziert hat (Tesla Inc., 2016). Durch die Verwendung eines Laserscanners kann die Wahrscheinlichkeit solcher Fehlklassifizierungen deutlich verringert werden.

Laserscanner können auch für eine weitere essentielle Aufgabe im Rahmen des autonomen Fahrens verwendet werden: die Lokalisierung des Fahrzeuges. Zur Wahl des korrekten Fahrstreifens, zur Pfadplanung oder auch zur Anfahrt auf Kreuzungen muss die Position des Fahrzeuges jederzeit mit einer hohen Genauigkeit im Zenti- bis geringem Dezimeterbereich bekannt sein (Levinson u. a., 2007). Fährt ein Fahrzeug auf eine Kreuzung zu, so muss es seine Geschwindigkeit bereits im Vorfeld anpassen und die eigene Route entsprechend berechnen. Hierfür muss bekannt sein, auf welcher Fahrspur sich das Fahrzeug befindet und welche Befehle an das Lenk- und Antriebssystem nötig sind, um gegebenenfalls die Fahrbahn zu wechseln. Im Vergleich zu klassischen Navigationssystemen ist hierbei eine Genauigkeit von wenigen Metern nicht ausreichend.

Bisherige Ansätze, wie beispielsweise in Levinson u. a. (2007) beschrieben, kombinieren LiDAR-Messungen mit Messungen einer Inertialen Messeinheit (IMU) sowie GNSS-Messungen (Globales Navigationssatellitensystem) zur Bestimmung einer globalen Position. Als globale Position wird die Position in einem globalen Koordinatensystem verstanden. Insbesondere in urbanen Gebieten ist eine zuverlässige GNSS-Positionsbestimmung jedoch nicht immer gegeben. Verdeckungen durch Bäume oder hohe Gebäude sowie Mehrwegeeffekte in Häuserschluchten können zu einem starken Genauigkeitsverlust bis hin zum Ausfall der Positionsbestimmung führen. Weiterhin sind alle GNSS-Systeme anfällig gegenüber Jamming oder Spoofing. Jamming bezeichnet hierbei eine Störung des Signals, um eine Positionsbestimmung zu verhindern, Spoofing beschreibt das Aussenden künstlicher Signale, um eine geänderte Position vorzutäuschen (Kerns u. a., 2014, Hu und Wei, 2009).

Um die Zuverlässigkeit und Genauigkeit der Fahrzeug-Lokalisierung autonom fahrender Fahrzeuge zu erhöhen, sollen in dieser Arbeit Methoden zur globalen Positionsbestimmung in urbanen Gebieten durch LiDAR-Messungen entwickelt werden. Diese Methoden müssen zuverlässig unabhängig von GNSS-Messungen eine Positionslösung bereitstellen. Ein weiterer relevanter Aspekt hierbei ist die Erzeugung einer hochgenauen Referenzkarte, mit der die Messungen der Automotive-Laserscanner abgeglichen werden. Wie in Seif und Hu (2016) beschrieben, können diese durch eine Fahrzeug-Flotte erstellt werden, welche mit einem Referenz-LiDAR-System und einer hochgenauen Lokalisierungseinheit, in diesem Fall bestehend aus einer Inertialeinheit und einem GNSS-Sensor, ausgestattet sind. Die Anzahl der Referenz-Fahrzeuge des Geodatendienstes Nokia HERE beläuft sich beispielsweise auf 400 (Stand: 2016) (Seif und Hu, 2016). Die somit stets aktuell gehaltenen, detaillierten Karten enthalten Landmarken, welche zur Lokalisierung verwendet werden können. Über eine Mobilfunkverbindung werden die Kartendaten an die entsprechenden Fahrzeuge gesendet, welche diese als Referenz zur eigenen Positionsbestimmung verwenden. Die Bedeutung einer hochgenauen Referenzlösung für die Zukunft des autonomen Fahrens zeigt sich weiterhin auch darin, dass ein aus der BMW Group, AUDI AG sowie Daimler AG bestehendes Konsortium bereits 2015 Nokia HERE als Basis für künftige FAS sowie das vollautomatisierte Fahren gekauft hat, um hochpräzise digitale Karten mit Echtzeit-Fahrzeugdaten zu verbinden (BMW Group, 2015).

Zur Lokalisierung wird zunächst die gesamte Umgebung, welche im Sichtbereich des Laserscanners liegt, erfasst. Neben statischen Objekten, wie Häuserfassaden oder Straßenlaternen schließt das auch dynamische Objekte, wie Fahrzeuge oder Fußgänger, mit ein. Wird die aktuell erfasste Umgebung mit einer Referenzkarte, welche zu einem anderen Zeitpunkt aufgenommen wurde, verglichen, so können dynamische Objekte zu einer fehlerhaften Positionsbestimmung führen. Aus diesem Grund müssen die entsprechenden Punktmessungen sowohl für die Generierung einer Referenzkarte als auch während des Lokalisierungsprozesses detektiert werden. Da für die Erstellung der Referenzdaten

die tatsächliche Position bekannt ist, können in diesem Fall Messungen zu unterschiedlichen Zeiten verglichen werden. Für den in dieser Arbeit Online-Lokalisierung genannten Vorgang der Positionsbestimmung des autonom oder hochautomatisiert fahrenden Fahrzeuges besteht diese Möglichkeit nicht. Stattdessen können hier zur Detektion dynamischer Objekte Methoden des maschinellen Lernens verwendet werden.

1.2. Ziel der Arbeit

Das Ziel der vorliegenden Arbeit ist die Entwicklung von Methoden, welche Laserscanner zur Eigenbestimmung der Position eines Fahrzeuges verwenden. In Abgrenzung zu vorherigen Arbeiten werden hierzu lediglich Automotive-Sensoren untersucht, welche auch bereits für weitere Fahrerassistenzsysteme genutzt werden. Weiter werden Ansätze vorgestellt, mit denen eine eigenständige Positionsbestimmung durch LiDAR-Messungen möglich ist, ohne die Zuhilfenahme der Messungen weiterer Sensoren. Es werden zwei wesentliche Fragestellungen zur Bewertung der Lokalisierung durch die entwickelten Methoden herangezogen:

- Wie hoch ist die **Zuverlässigkeit** der jeweiligen Methode?
- Mit welcher **Genauigkeit** ist eine Lokalisierung möglich?

Als Zuverlässigkeit wird der prozentuale Anteil an korrekten Positionsbestimmungen entlang einer Test-Trajektorie verstanden. Eine Position ist als korrekt definiert, sobald ihre Abweichung zu einer Referenz geringer als ein festgelegter Grenzwert ist. Als Genauigkeit wird die Wurzel der mittleren quadratischen Abweichung (Root Mean Square Error, *RMSE*) bezeichnet, wobei zur Bestimmung des Wertes nur korrekte Positionen berücksichtigt werden. In Levinson u. a. (2007) wird 0,10 m als benötigte Lokalisierungsgenauigkeit für das autonome Fahren angegeben. Diese Genauigkeit soll in dieser Arbeit erreicht werden. Neben Methoden, in denen vorausgesetzt wird, dass eine initiale Position bereits bekannt ist, werden hierfür auch Ansätze vorgestellt, welche unabhängig von einer beispielsweise aus GNSS-Messungen erzielten Initial-Position die Lokalisierung durch den Einsatz von LiDAR-Messungen durchführen können.

Für alle Methoden soll eine Referenzkarte durch den Einsatz eines Mobile-Mapping-Systems erstellt werden. Im Vergleich zu den verwendeten Automotive-Laserscannern verfügen die Sensoren des Referenzsystems über eine übergeordnete Genauigkeit. Die Referenzkarte enthält bestimmte Merkmale, welche zur Lokalisierung verwendet werden können. Hierzu werden aus den Messungen des Automotive-Laserscanners ebenfalls Merkmale extrahiert, welche in den folgenden Kapiteln als Online-Merkmale bezeichnet werden. Die Lokalisierung erfolgt anschließend durch einen Abgleich der Online-Merkmale mit der Referenzkarte.

Im Wesentlichen werden drei unterschiedliche Verfahren vorgestellt, um aus den LiDAR-Messungen Merkmale zu extrahieren und hiermit die Fahrzeug-Lokalisierung durchzuführen:

1. Die Verwendung **stangenförmiger Objekte und Ebenen** als Merkmale. Diese Merkmale werden in diesem Fall auch als Landmarken bezeichnet und sowohl aus den Daten des Mobile-Mapping-Systems als auch aus den Daten des Automotive-Laserscanners extrahiert. Über einen Abgleich der Lage der Stangen sowie der Lage und der Ausrichtung der Ebenen soll eine Zuordnung der Merkmale aus den Referenz- und den Online-Daten stattfinden und hieraus die Position bestimmt werden. Zur eindeutigen Zuordnung der Merkmale können sogenannte Landmarken-Muster verwendet werden. Hierbei wird die relative Position der Online-Landmarken zueinander analysiert und mit den Referenzdaten verglichen. Dies soll zu einer Vermeidung von fehlerhaften Zuordnungen führen sowie dazu, dass aus den extrahierten Landmarken eine globalen Positionsbestimmung erfolgen kann.

2. Die Verwendung eines **neuronalen Netzes** zur Extraktion von Merkmalen aus den Scans eines Automotive-Laserscanners. Die Merkmale sollen durch eine spezielle Form eines neuronalen Netzes, einen Autoencoder, bestimmt und durch ein Clustering-Verfahren einem Label zugeordnet werden. Die Lokalisierung kann anschließend durch eine **Sequenzanalyse** einer Referenzsequenz mit den Online-Daten erfolgen. Dadurch, dass die Merkmale lediglich aus den Distanz- und Winkelmessungen der einzelnen Scans bestimmt werden, werden keine ergänzenden Sensoren benötigt, um die globale Position des Fahrzeuges zu ermitteln.
3. Die Generierung von **Scanbildern** aus den Höhen- und Intensitätswerten der LiDAR-Daten, wobei sich die Merkmale aus den Werten an den jeweiligen Pixeln ergeben. Aus den Messungen des Mobile-Mapping-Systems soll hieraus eine Referenzkarte, welche aus den jeweiligen Scanbildern entlang einer Test-Trajektorie besteht, erstellt werden. Als Datenquelle sollen mehrere Messfahrten zu unterschiedlichen Tages- und Jahreszeiten verwendet werden. Ausgehend von einer Initial-Position kann die Lokalisierung über eine Korrelation der Höhen- und Intensitätswerte des Automotive-Laserscanners mit den Werten der entsprechenden Pixel im Referenzbild erfolgen. Das Ergebnis der Lokalisierung ergibt sich aus der Position und Orientierung in einem begrenzten Bereich um die initiale Position mit dem höchsten Korrelationswert.
Ein besonderer Fokus soll auf der **Detektion dynamischer Objekte** liegen. Um eine genaue und zuverlässige Lokalisierung zu ermöglichen, sollen nur Objekte zur Positionsbestimmung verwendet werden, welche über einen langen Zeitraum weder ihre Position noch ihre Form verändern. Dynamische Objekte können in den Referenzdaten durch eine Änderungsdetektion in den verschiedenen Datensätzen extrahiert werden. In den Online-Daten kann die Detektion durch Verfahren des maschinellen Lernens erfolgen.

In den unterschiedlichen Methoden sollen jeweils verschiedene Automotive-Laserscanner untersucht werden. Diese unterscheiden sich neben deren Spezifikationen, wie der Genauigkeit der Messungen, dem Sichtbereich oder der Messfrequenz auch in der Einbauposition und -Ausrichtung am Fahrzeug.

1.3. Struktur der Arbeit

Die Arbeit gliedert sich in die folgenden Kapitel: nach der Einleitung in Kapitel 1 folgt Kapitel 2, welches einen theoretischen Überblick über Laserscanning und damit verbundene Verfahren sowie über maschinelles Lernen und die Filterung von Zuständen gibt. Neben verschiedener Messprinzipien und Formen des Laserscannings wird auch auf den effizienten Zugriff auf die Daten durch räumliche Datenstrukturen und die Segmentierung bestimmter Objekte aus Punktwolken eingegangen. Abschnitt 2.2 beinhaltet verschiedene Aspekte des maschinellen Lernens: das Clustering in Form eines K-Means-Algorithmus, die Klassifizierung durch Random Forests sowie neuronale Netze. Weiterhin werden in Abschnitt 2.3 Filterverfahren eingeführt, wobei insbesondere auf das Kalman-Filter sowie das Extended Kalman-Filter eingegangen wird.

In Kapitel 3 wird der aktuelle Stand der Forschung erarbeitet. Hierbei werden verschiedene Methoden zur Detektion dynamischer Objekte durch eine Änderungsdetektion oder eine Klassifizierung der LiDAR-Daten sowie Ansätze zur Fahrzeuglokalisierung präsentiert.

Kapitel 4 stellt die in der Arbeit verwendeten Sensoren und Daten vor. Dabei wird auf das zur Messung von Referenzdaten dienende Mobile-Mapping-System sowie auf die unterschiedlichen Automotive-Laserscanner eingegangen.

Die in dieser Arbeit zur Detektion dynamischer Objekte entwickelten Methoden werden in Kapitel 5 behandelt. Auf eine Präsentation der Änderungsdetektion unter Verwendung mehrerer Datensätze des Mobile-Mapping-Systems folgen Methoden des maschinellen Lernens, welche dynamische Objekte in Punktwolken, in einzelnen Zeilenscans oder in Scanbildern detektieren.

Die verschiedenen Methoden zur Fahrzeug-Lokalisierung durch Automotive-Laserscanner werden in Kapitel 6 erläutert. Abschnitt 6.1 geht auf die Segmentierung von Stangen und Ebenen aus den Referenzdaten sowie den Automotive-Daten ein. Weiterhin wird die Zuordnung der segmentierten Merkmale über Landmarken-Muster vorgestellt. Abschnitt 6.2 hat die Lokalisierung über eine Sequenzanalyse, welche aus einer Bestimmung von Merkmalen durch einen Autoencoder, dem Clustering der Merkmale sowie der Bestimmung der aktuellen globalen Position besteht, zum Inhalt. In Abschnitt 6.3 werden schließlich Scanbilder zur Lokalisierung verwendet. Hierbei wird ebenfalls auf Besonderheiten eingegangen, die sich aus der Verwendung eines Velodyne VLP-16 ergeben.

Die Ergebnisse der verschiedenen Methoden werden anschließend in Kapitel 7 untersucht und diskutiert. Kapitel 8 enthält eine Zusammenfassung der Arbeit sowie einen Ausblick. Ergänzende Abbildungen und Tabellen sind im darauffolgenden Anhang enthalten.

2. Grundlagen

In diesem Kapitel werden die wesentlichen Methoden und Prinzipien vorgestellt, die in der vorliegenden Arbeit angewandt werden. Zur Datengewinnung wird in dieser Arbeit das Laserscanning verwendet. Eine Einführung in die Sensorik und Methodik des Laserscannings ist in Abschnitt 2.1 zu finden. Abschnitt 2.2 beinhaltet verschiedene Methoden des Machine Learnings, die in dieser Arbeit genutzt werden. Anschließend werden in Abschnitt 2.3 Filter-Verfahren präsentiert, die zur Lokalisierung verwendet werden.

2.1. Laserscanning

Ein wesentliches Ziel des Laserscannings ist die Erfassung der dreidimensionalen Umgebung des jeweiligen Messsystems. Hierzu werden in einer hohen zeitlichen Auflösung mehrere Distanz- und Winkelmessungen durchgeführt, wobei zur Distanzmessung Laserlicht mit einer Wellenlänge von $0,4 - 1,6 \mu\text{m}$ (Beraldin u. a., 2010a) verwendet wird. Zusätzlich besteht bei abbildenden Laserscannern die Möglichkeit, die Intensität des vom jeweiligen Objekt reflektierten Laserstrahls zu erfassen und zu speichern.

Laserscanning lässt sich in drei Bereiche aufteilen: terrestrisches Laserscanning, Airborne-Laserscanning sowie Mobile Mapping. Beim terrestrischen Laserscanning findet die Messung von einem oder mehreren festen Standpunkten aus statt. Um die Umgebung in drei Dimensionen zu erfassen, ist folglich eine Ablenkung des Laserstrahls in zwei Dimensionen nötig. Mehrere Messungen können im Anschluss, wie in Abschnitt 2.1.4 beschrieben, zu einem Datensatz zusammengefasst werden. Das Airborne-Laserscanning bezeichnet die Aufnahme eines Messgebietes aus der Luft, wobei der Sensor in einem Flugzeug oder Helikopter verbaut ist (Beraldin u. a., 2010a). In den vergangenen Jahren haben sich weiterhin UAV (Unmanned Aerial Vehicle) als Sensorträger etabliert (Lin u. a., 2011, Wallace u. a., 2014). In Gardner (1992) wurde außerdem eine Messung der Erdoberfläche durch an Satelliten gebundene Laser vorgestellt.

In dieser Arbeit liegt der Fokus auf der Verwendung von Mobile-Mapping-Systemen. Hierbei ist ein LiDAR-System an einem mobilen Träger auf der Erdoberfläche montiert, in diesem Fall an Fahrzeugen im regulären Straßenverkehr.

Im Folgenden sollen zunächst die grundlegenden Messprinzipien des Laserscannings vorgestellt werden. Anschließend wird auf den Bereich Mobile Mapping eingegangen, bevor im letzten Abschnitt Verfahren zur Punktwolkenregistrierung eingeführt werden.

2.1.1. Distanzmessung

Um die Distanz eines Punktes mithilfe eines Laserstrahls zu bestimmen, können drei verschiedene Messprinzipien verwendet werden. Zwei dieser Prinzipien bestimmen die Distanz, indem die Laufzeit des Lasers direkt gemessen oder aus den Messungen abgeleitet wird: das Impulslaufzeitverfahren sowie das Phasenvergleichsverfahren. Dem entgegen steht das Triangulationsverfahren, welches die Distanz geometrisch bestimmt. Da dieses Verfahren typischerweise nur eine begrenzte Reichweite von weniger als 5 m aufweist und somit für den Automotive-Bereich nicht geeignet ist, soll es in dieser Arbeit nicht weiter erläutert werden (Kahmen, 2006).

Impulslaufzeitverfahren

Wird ein Laserimpuls vom Sensor ausgesendet, so lässt sich die zurückgelegte Strecke d über die Laufzeit τ bestimmen, die der Impuls zum Objekt und wieder zurück zum Empfänger benötigt. Dies ist möglich, da die Geschwindigkeit des Laserlichts in einem gleichbleibenden Medium konstant ist. Im Vakuum beträgt die Lichtgeschwindigkeit $c = 299\,792\,458\text{ m/s}$. In der Luft muss zusätzlich ein Refraktionskoeffizient n angebracht werden, welcher von der Temperatur, dem Luftdruck und der Luftfeuchtigkeit abhängt (Beraldin u. a., 2010a). Die Distanz zum Objekt ergibt sich somit zu

$$d = \frac{c \tau}{n 2} \quad (2.1)$$

Je größer die Distanz zum Sensor, desto größer ist auch die vom Laserimpuls erfasste Fläche, der sogenannte Footprint. Der Footprint des im Riegl VMX-250 verbauten Laserscanners Riegl VQ-250 beträgt beispielsweise auf einer Distanz von 50 m 18 mm (RIEGL Laser Measurement Systems GmbH, 2012b). Dies kann zur Folge haben, dass ein Teil des Laserpulses auf ein Objekt trifft und reflektiert wird, ein anderer Teil jedoch nicht dasselbe, sondern ein anderes Objekt trifft. Hierdurch kehren für einen ausgesendeten Puls mehrere Impulse zum Empfänger zurück. Im Airborne-Laserscanning wird dieses Verhalten genutzt, um mithilfe einer Full-Waveform-Analyse Vegetation vom Boden innerhalb eines ausgesendeten Laserimpulses zu unterscheiden und somit die Höhe der Baumkronen zu bestimmen (Hofton u. a., 2000, Höfle u. a., 2012). Um zwei Pulse ohne eine ausführliche Analyse der reflektierten Wellenform unterscheiden zu können, müssen diese eine bestimmte zeitliche Differenz aufweisen, die sich wie in Beraldin u. a. (2010a) gezeigt bestimmen lässt: zunächst wird hierfür die Pulsbreite t_b benötigt. Ein typischer Wert für die Pulsbreite ist $t_b = 5\text{ ns}$, was einer in dieser Zeit vom Laserlicht zurückgelegten Distanz von 1,5 m entspricht. Zwei reflektierte Impulse können nur unterschieden werden, wenn deren Differenz größer als die halbe Pulsbreite ist. Für $t_b = 5\text{ ns}$ entspricht dies einer Distanz von 0,75 m.

Wie in Jutzi und Stilla (2006) und Stilla und Jutzi (2008) gezeigt, lässt sich durch eine Full-Waveform-Analyse die minimale Distanzdifferenz zweier Objekte auf ein Zehntel der Pulsbreite verringern. Dies wird durch eine Modellierung der reflektierten Wellenform in Abhängigkeit der Wellenform und Energieverteilung des emittierten Laserpulses sowie der Reflektionseigenschaften der Objektoberfläche ermöglicht. Die Unterscheidung zweier Pulse gelingt auch unter schwierigen Rückstreuverhältnissen, welche beispielsweise durch Vegetation bedingt sein können (Reitberger u. a., 2008).

Da die kalkulierte Strecke unmittelbar von der Laufzeit des Signals abhängt, ist es essentiell, diese möglichst genau zu messen. Neben der Genauigkeit der Zeitmessung ist hier auch eine genaue Detektion des reflektierten Laserimpulses am Empfänger von großer Bedeutung. Ein Vergleich verschiedener Detektionsmöglichkeiten findet sich in Wagner u. a. (2004) sowie Jutzi und Stilla (2008). Die Laufzeit kann entweder analog oder digital bestimmt werden. Bei der digitalen Zeitmessung findet die Zeitmessung durch einen Oszillator mit einer Frequenz von etwa 300 MHz statt. Bei der analogen Messung werden die ansteigenden Flanken einer Taktfrequenz von 15 MHz gezählt. Anschließend bestimmt ein Kondensator die Restintervalle des Signals, was zu einer höheren Genauigkeit der Messung führt (Schlemmer, 1996).

Vorteile des Impulslaufzeitverfahrens sind die Möglichkeit, mehrere Pulse auszuwerten sowie die große Reichweite der Scanner. Kommerzielle Airborne-Laserscanner weisen beispielsweise eine Reichweite von bis zu 5000 m auf (Jutzi u. a., 2016). Die Genauigkeit typischer kommerzieller Systeme beträgt meist 5 - 10 mm (Beraldin u. a., 2010a).

Phasenvergleichsverfahren

Im Gegensatz zum Impulslaufzeitverfahren bestimmt das Phasenvergleichsverfahren nicht die Zeit, sondern die Phasendifferenz $\Delta\varphi$ zwischen den Messsignalen der Sende- und Empfangseinheit. Der Laserstrahl wird hierbei zu einem sinusförmigen Signal mit der Wellenlänge λ_M aufmoduliert (Schlemmer, 1996). Die Wellenlänge des Signals lässt sich aus der jeweiligen Modulationsfrequenz f_M bestimmen:

$$\lambda_M = \frac{c}{f_M} \quad (2.2)$$

Aus

$$\Delta\lambda_M = \frac{\Delta\varphi}{2\pi} \lambda_M \quad (2.3)$$

ergibt sich das aus der Phasendifferenz entstehende Reststück der Welle. Wird nur eine einzelne Modulationsfrequenz verwendet, so ist die Distanz nur bis zur halben Wellenlänge des Signals eindeutig bestimmbar (Beraldin u. a., 2010a). Bei einer größeren Distanz müssen die Mehrdeutigkeiten N bestimmt werden:

$$d = N \frac{\lambda_M}{2} + \frac{\Delta\lambda_M}{2} \quad (2.4)$$

Hierzu werden mehrere Signale mit unterschiedlicher Frequenz ausgesendet. In Beraldin u. a. (2010a) wird angegeben, dass typische kommerzielle Systeme einen Messbereich von 10 kHz bis etwa 625 kHz aufweisen, woraus sich eine Reichweite von etwa 100 m ergibt.

Das Phasenvergleichssystem ermöglicht sehr hohe Datenraten von 1 000 000 Punktmessungen pro Sekunde, die Reichweite solcher Systeme ist im Vergleich zu Impulslaufzeit-Sensoren jedoch beschränkt. Die Genauigkeit des Verfahrens ist abhängig von den gewählten Messfrequenzen und kann bis in den Mikrometer-Bereich gehen (Beraldin u. a., 2010a). Die Entfernungsauflösung eines aktuellen Phasenvergleichs-Laserscanners (Z+F Imager 5010X) beträgt 0,1 mm mit einem Entfernungsrauschen von 0,3 - 0,6 mm bei einer Entfernung von 25 m, in Abhängigkeit von den Reflektionseigenschaften der Oberfläche (Zoller + Fröhlich GmbH, 2018).

2.1.2. Scanmechanismus

Um die Umgebung eines Laserscanners von einer Position aus in mehreren Dimensionen zu erfassen, ist es erforderlich, den Laserstrahl in unterschiedliche Richtungen auszustrahlen. Dies kann erreicht werden, indem der Laser durch zwei orthogonal angeordnete, sich drehende Spiegel abgelenkt wird. Terrestrische Laserscanner erfassen den dreidimensionalen Raum üblicherweise, indem sich das Gehäuse des Scanners selbst dreht. Parallel dazu rotiert ebenfalls ein Spiegel im Inneren des Gehäuses (Pfeifer u. a., 2016). Im Falle mobiler Plattformen wird die dritte Dimension durch eine Bewegung der Plattform und damit auch des Laserscanners erzielt (Beraldin u. a., 2010a). Ein weiteres Verfahren, welches von der Continental AG als „Schlüsseltechnologie für automatisiertes Verfahren“ (Continental AG, 2018) beschrieben wird, ist das Flash LiDAR.

In diesem Abschnitt soll die 3D-Raumerfassung durch verschiedene rotierende Spiegel, die Bewegung einer mobilen Plattform sowie durch Flash-LiDAR-Sensoren näher erläutert werden.

Strahlablenkung durch eine Spiegelkonstruktion

Wird ein Spiegel zur Ablenkung des Laserstrahls verwendet, muss der Spiegel um mindestens eine Achse rotieren. Die aus Beraldin u. a. (2010a) entnommene Abbildung 2.1 zeigt verschiedene

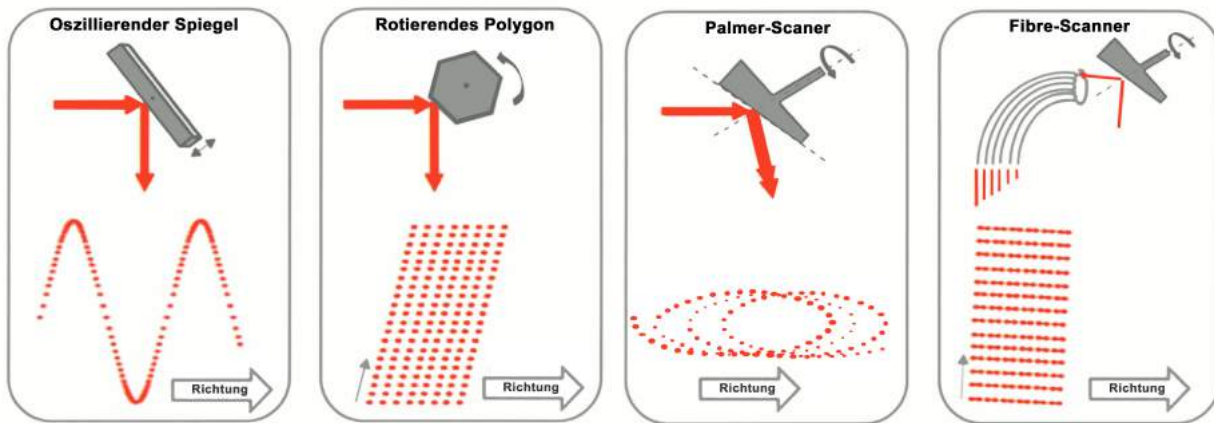


Abbildung 2.1.: Ablenkmechanismen des Laserstrahls durch einen sich drehenden Spiegel (Quelle: Beraldin u. a. (2010b), verändert). Die Abbildung zeigt das Prinzip eines oszillierenden Spiegels, eines rotierenden Polygons, eines Palmer- sowie eines Fibre-Scanners (von links nach rechts).

Ablenkmechanismen des Laserstrahls.

Im Falle eines **oszillierenden Spiegels** nimmt die Dichte der Punkte von außen nach innen entlang einer Scanlinie ab. Dies resultiert daraus, dass die Rotationsgeschwindigkeit des Spiegels nicht konstant ist. Da sich die Richtung der Rotation ändert, ist die Rotationsgeschwindigkeit zum Ende der Scanlinie niedriger als im mittleren Bereich. Durch die Oszillation des Spiegels ergibt sich ein Z-Muster an der Oberfläche (siehe Abbildung 2.1). Wird ein **rotierendes Polygon** als Spiegel verwendet, sind die Scanlinien parallel angeordnet, mit einer konstanten Winkeldistanz entlang einer Scanlinie. Die Punktdichte wird für mobile Plattformen entlang der Bewegungsrichtung nur von der Geschwindigkeit der Plattform bestimmt. Sogenannte **Palmer-Scanner** verwenden eine Spiegelanordnung, in welcher der Winkel zwischen Spiegeloberfläche und Rotationsachse des Spiegels ungleich 90° ist. Hieraus resultiert ein elliptisches Muster entlang der Oberfläche. **Fibre-Scanner** verwenden fest angeordnete Glasfasern, durch welche sequentiell von einem rotierenden Spiegel reflektierte Laserstrahlen ausgesendet werden. Hierdurch entsteht ebenfalls eine parallele Anordnung der Scanlinien. Die feste Anbringung der Fasern hat zur Folge, dass die Aussendewinkel des Laserstrahls sehr stabil sind (Beraldin u. a., 2010a).

Flash LiDAR

Im Gegensatz zu herkömmlichen LiDAR-Systemen im Automotive-Bereich, benötigt das Flash LiDAR keine mechanischen Teile. Stattdessen wird von einer Diffusor-Linse modulierte Licht auf die Oberfläche projiziert. Das reflektierte Signal wird anschließend wieder durch eine elektrooptische Vorrichtung demoduliert, sodass jedem Signal die korrekten Aussendewinkel zugeordnet werden können (Beraldin u. a., 2010a). Hierdurch entsteht aus einem einzelnen Laserstrahl ein gesamtes Scanbild, welches Entfernungswerte beinhaltet und aus dem sich somit direkt eine 3D-Punktwolke des gemessenen Bereichs berechnen lässt. Das von der Continental AG entwickelte System liefert beispielsweise pro Messung eine Auflösung von 64 Pixeln in der Höhe, 192 Pixeln in der Breite und somit 12 000 3D-Punkte bei einer Messfrequenz von 30 Hz (Continental AG, 2018). Als Nachteil kann angesehen werden, dass ein Flash-LiDAR-Sensor aufgrund des geringen Öffnungswinkels nur einen sehr begrenzten Bereich erfassen kann.

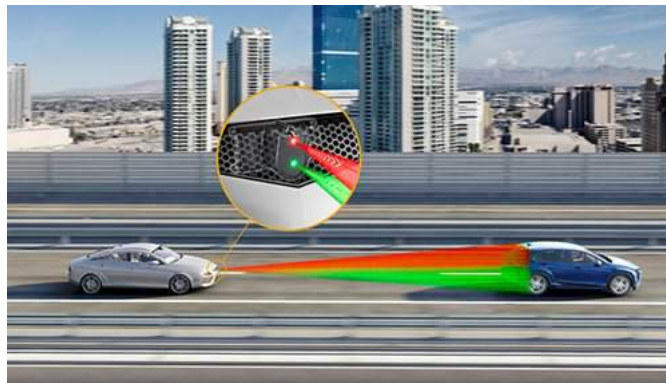


Abbildung 2.2.: Bei einem Flash-LiDAR-System wird ein modulierter Laserstrahl auf die Oberfläche des zu messenden Objektes projiziert (rot). Das reflektierte Laserlicht (grün) wird vom Empfänger erfasst und demoduliert (Quelle: Continental AG (2018)).

3D-Raumerfassung durch die Bewegung der Plattform

Mobile Systeme benötigen zur 3D-Erfassung des Raumes grundsätzlich nur Messungen des Laserscanners in zwei Dimensionen. Die dritte Dimension kann durch die Eigenbewegung des Systems erfasst werden. Die Position und Orientierung (Pose) des Laserscanners wird hierbei durch eine zum Scanner kalibrierte Lokalisierungseinheit bestimmt. In der Regel wird hierfür eine Kombination aus GNSS- und Inertialsensoren verwendet, deren Daten in einem Filteransatz kombiniert werden (Toth und Jutzi, 2016). Da durch die Messung des GNSS-Sensors Koordinaten in einem globalen Koordinatensystem bekannt sind, kann gleichzeitig auch eine Georeferenzierung der Punkte stattfinden. Die Genauigkeit der GNSS-Messungen kann verbessert werden, indem Daten einer GNSS-Referenzstation verwendet werden. Zusätzlich zu GNSS-Sensoren können noch Odometer sowie Beschleunigungs- und Drehratensensoren zur fortlaufenden Positionsbestimmung genutzt werden.

Um jeder Messung des Laserscanners eine genaue Position und Orientierung zuzuweisen, muss eine zeitliche Synchronisierung der Messungen erfolgen. Hierfür wird in der Regel der vom GNSS-Sensor bereitgestellte PPS-Impuls (pulse per second) verwendet (Kutterer, 2010). Eine Übersicht verschiedener Methoden zur Zeitsynchronisierung liefern Ding u. a. (2005).

2.1.3. Mobile Mapping

Kutterer (2010) definiert Mobile Mapping als Aufnahme und Bereitstellung geometrischer 2D- oder 3D-Umgebungsinformationen mithilfe eines bildgebenden Sensors, welcher auf einer mobilen Plattform angebracht ist. Als bildgebende Sensoren können hierbei beispielsweise Kameras oder Laserscanner verwendet werden. Weitere Bestandteile eines Mobile-Mapping-Systems sind Sensoren zur Georeferenzierung, also zur Bestimmung der fortlaufenden Position und Orientierung des Systems, eine Recheneinheit zur Zusammenführung und Speicherung der Messungen sowie zur Bedienung des Systems und die Plattform, auf der die einzelnen Komponenten angebracht sind (El-Sheimy, 2005). In dieser Arbeit werden Laserscanner als bildgebende Sensoren verwendet.

Das erste kommerzielle LiDAR-Mobile-Mapping-System StreetMapper wurde ursprünglich zur Vermessung von Autobahnen entwickelt und war 2005 erstmals im Einsatz (Haala u. a., 2008, Kremer, 2007). Die heutigen Einsatzgebiete fahrzeuggebundener LiDAR-Mobile-Mapping-Systeme reichen von der Straßenerfassung, über die Erfassung von Baumbeständen, bis hin zur 3D-Erfassung von Minen (Zlot und Bosse, 2014, Neumann u. a., 2014).

In Kaartinen u. a. (2012) werden verschiedene Mobile-Mapping-Systeme vorgestellt und in einem Testgebiet hinsichtlich der absoluten Positionsgenauigkeit der resultierenden Punktwolken untersucht. Es zeigt sich, dass unter guten GNSS-Bedingungen Genauigkeiten von 3,5 mm in der Lage der Punkte bei einer Messdistanz von 35 m erreicht werden können. Diese Ergebnisse unterstreichen, dass sich Mobile Mapping auch für einen weiteren aktuellen Forschungsschwerpunkt eignet: die Erstellung und Bereitstellung hochgenauer Karten für das autonome Fahren. Die Bedeutung hochgenauer Karten für das autonome Fahren wird in Seif und Hu (2016), das Potential von LiDAR Mobile Mapping für die Erstellung hochgenauer Karten in Hofmann (2017) beschrieben.

Automotive-Laserscanner

Nach der Definition von Kutterer (2010) lassen sich im weiteren Sinne auch im Automotive-Bereich verwendete Laserscanner in den Bereich des Mobile Mappings einordnen. Im Vergleich zu herkömmlichen Scannern unterscheiden sich Automotive-Laserscanner in den Anforderungen, die an sie gestellt werden. Da sie in Zukunft auch in Serienfahrzeugen verbaut werden sollen, müssen sie eine möglichst kompakte Form aufweisen, kostengünstig sein und zuverlässig arbeiten. Zwar treffen diese Anforderungen auch auf herkömmliche Systeme zu, allerdings ist deren Bedeutung im Automotive-Bereich höher. Der verfügbare Raum für Sensoren in einem Fahrzeug ist begrenzt. Zusätzlich sollen verbaute Sensoren das Design des Fahrzeuges nicht beeinflussen. Hohe Sensorkosten würden sich unmittelbar auf den Kaufpreis des Fahrzeuges auswirken, sodass die Kosten für einen gewöhnlichen Verbraucher den Nutzen weit überschreiten würden. Im Fahrzeug verbaute Laserscanner werden in der Regel für sicherheitsrelevante Aufgaben verwendet, ein Ausfall des Sensors ist somit kritischer, als im Falle einer reinen Vermessung eines bestimmten Gebietes. Gleichzeitig ist die Laufleistung des Laserscanners sehr hoch. Im Jahr 2016 hat ein Fahrzeug in Deutschland im Durchschnitt 13 341 km pro Jahr zurückgelegt (Kraftfahrt-Bundesamt, 2017). Um Ausfälle der Sensoren zu vermeiden, muss in der Konstruktion besonderen Wert auf deren Robustheit gelegt werden.

2.1.4. Registrierung von Punktwolken

Nach der Messung des Laserscanners befinden sich die Punkte zunächst im Sensorkoordinatensystem der Laserscanner. Da die Scanner eine Strecke d und im Falle von 3D-Scannern zwei Winkel α und φ messen, liegen die Werte in der Regel zunächst in einem sphärischen Koordinatensystem vor. Diese lassen sich jedoch für einen Messpunkt $p(x,y,z)$ aus der folgenden Beziehung in ein kartesisches Koordinatensystem überführen (Pfeifer u. a., 2016):

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = d \begin{pmatrix} \cos \alpha \cos \varphi \\ \sin \alpha \cos \varphi \\ \sin \varphi \end{pmatrix} \quad (2.5)$$

In vielen Fällen ist es notwendig, die Transformation zwischen mehreren Scans, beispielsweise aus unterschiedlichen Messepochen, zu bestimmen. Hierfür wird in der Regel eine Starrkörpertransformation T bestimmt, welche aus einer Rotationsmatrix R sowie einer Translation t besteht und die Punktwolke eines Scans in das Koordinatensystem eines anderen, dessen Werte in diesem Fall nicht verändert werden, überführt (Lichti und Skaloud, 2010). Eine Skalierung muss nicht angebracht werden, da für alle Messungen metrische Koordinaten vorliegen. Im Folgenden soll davon ausgegangen werden, dass eine Referenzpunktwolke und eine zu dieser Referenz transformierende Punktwolke vorhanden sind. Zur Bestimmung der Transformation kann zwischen zwei Methoden unterschieden werden: die Verwendung gemeinsamer Punkte in beiden Datensätzen und das Heranziehen aller in den Daten enthaltenen Punkte (Pfeifer u. a., 2016).

Insbesondere für terrestrische Laserscanner werden häufig Zielmarken verwendet. Diese Zielmarken sind so angebracht, dass sie im Sichtbereich der Laserscannerstandpunkte liegen. Die

Mittelpunkte der Zielmarken gehen als Beobachtungen in die Bestimmung der Transformation ein. Sie können entweder manuell oder automatisch anhand der auf der Zielmarke liegenden Punkte bestimmt werden. Weiterhin können die Intensitätswerte der Punktmessungen zur Bestimmung der Mittelpunkte verwendet werden (Lichti und Skaloud, 2010).

Als Zielmarken werden geometrische Objekte wie Kugeln oder Ebenen verwendet. Im Falle von Ebenen sind diese häufig durch ein Hell-Dunkel-Muster geprägt. Diese Zielmarken haben weiterhin den Vorteil, dass sie auch zur Kalibrierung des Systems zu einer Kamera genutzt werden können. Weiterhin ist die Verwendung retroreflektierender Materialien möglich. Da diese Zielmarken einen Großteil der einfallenden Strahlung wieder zum Sensor reflektieren, heben sie sich deutlich von der Umgebung ab und sind somit auch automatisch auffindbar (Lichti und Skaloud, 2010).

Ein weitere Möglichkeit, gemeinsame Punkte in beiden Datensätzen zu definieren, ist, statt künstlicher Punkte natürliche Punkte zu wählen, die markante geometrische Merkmale aufweisen oder im Intensitätsbild über Bildoperatoren gewählt werden. In Feng u. a. (2016) wird ein neuronales Netz trainiert, um solche Punkte im Distanzbild zu detektieren. Barnea und Filin (2007) verwenden Distanzbilder und einen bildbasierten Ansatz, um detektierte Punkte einander zuzuordnen. Untersuchungen zu künstlichen Merkmalspunkten in Intensitätsbildern finden sich in Kang u. a. (2009) und Böhm und Becker (2007). Bei der Verwendung automatisch detektierter Merkmalspunkte kann es zu vielen Fehlzusordnungen kommen. Dieses Problem kann jedoch durch die Verwendung eines RANSAC-Algorithmus behoben werden (Lichti und Skaloud, 2010).

In Tuttas u. a. (2017) findet eine kontinuierliche Registrierung photogrammetrischer Punktwolken mithilfe von Zielmarken und natürlicher Merkmalspunkten in einer dynamischen Umgebung statt. Die künstlichen Zielmarken dienen hierbei zur Festlegung des Koordinatensystems. Mithilfe von durch einen SIFT-Operator (Lowe, 1999) detektierten Verknüpfungspunkten wird zur Registrierung neuer Aufnahmen ein Structure-from-Motion-Prozess mit einer anschließenden Bündelblockausgleichung durchgeführt.

In Weinmann u. a. (2011) und Weinmann (2016) wird ein SIFT-Operator verwendet, um im Intensitäts- und Distanzbild gemeinsame Punkte zweier Laserscans zu detektieren. Anschließend wird eine virtuelle Kamera eingeführt, indem die entsprechenden korrespondierenden 3D-Punkte auf eine virtuelle Ebene projiziert werden. Ursprung und Orientierung der Kamera entsprechen hierbei den Werten des Laserscanners. Die resultierenden 2D-Punkte werden verwendet, um eine Vorregistrierung der Datensätze durchzuführen. Neben der Registrierung von Punktwolken kann die Einführung virtueller Ebenen auch zur Zuordnung von 3D- zu Bildpunkten genutzt werden.

Sind für zwei Punktwolken n korrespondierende Punkte bekannt, so kann eine Transformation bestimmt werden, die einen Punkt l_i aus einer hier als linke Punktwolke bezeichneten Datensatz $\mathbf{l} = \{l_i\}$ in eine als rechte Punktwolke $\mathbf{r} = \{r_i\}$ bezeichnete Referenz überführt. Hierfür gilt folgende Beziehung:

$$r_i = Rl_i + t \quad (2.6)$$

R bezeichnet hierbei eine 3×3 -Rotationsmatrix, t einen 3×1 -Translationsvektor. Das Ziel ist es, die jeweiligen Punktabstände zwischen den Referenz- und den transformierten Punkten zu minimieren:

$$\sum_i ||Rl_i + t - r_i||^2 \rightarrow \min \quad (2.7)$$

Die notwendige Transformation kann wie in Horn (1987) beschrieben bestimmt werden. Zunächst müssen beide Punktwolken um ihren Schwerpunkt $\bar{l}$ und $\bar{r}$ reduziert werden:

$$\begin{aligned} l'_i &= l_i - \bar{l} \\ r'_i &= r_i - \bar{r} \end{aligned} \quad (2.8)$$

Die Schwerpunkte ergeben sich aus einer Mittelwertbildung der jeweiligen Punkte:

$$\begin{aligned} \bar{l} &= \frac{1}{n} \sum_{i=1}^n l_i \\ \bar{r} &= \frac{1}{n} \sum_{i=1}^n r_i \end{aligned} \quad (2.9)$$

Wie in Horn (1987) gezeigt, ergibt sich R aus dem Eigenvektor zum größten Eigenwert der symmetrischen Matrix

$$N = \begin{pmatrix} s_{xx} + s_{yy} + s_{zz} & s_{yz} - s_{zy} & s_{zx} - s_{xz} & s_{xy} - s_{yx} \\ s_{yz} - s_{zy} & s_{xx} - s_{yy} - s_{zz} & s_{xy} + s_{yx} & s_{zx} + s_{xz} \\ s_{zx} - s_{xz} & s_{xy} + s_{yx} & -s_{xx} + s_{yy} - s_{zz} & s_{yz} + s_{zy} \\ s_{xy} - s_{yx} & s_{zx} + s_{xz} & s_{yz} + s_{zy} & -s_{xx} - s_{yy} + s_{zz} \end{pmatrix}, \quad (2.10)$$

wobei die jeweiligen Einträge aus den Summen der Elemente der Kreuzkovarianzmatrix S der Punkte stammen:

$$S = \sum_{i=1} r_i l_i^T = \begin{pmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{pmatrix} \quad (2.11)$$

Der größte Eigenwert zur Matrix N ergibt nun das Einheitsquaternion $q = q_0 + q_1 i + q_2 j + q_3 k$. Dieses Quaternion beschreibt die sich ergebende Rotation und kann verwendet werden, um die Rotationsmatrix R aufzustellen:

$$R = \frac{1}{q_0^2 + q_1^2 + q_2^2 + q_3^2} \cdot \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 \cdot q_2 - q_0 \cdot q_3) & 2(q_1 \cdot q_3 + q_0 \cdot q_2) \\ 2(q_2 \cdot q_1 + q_0 \cdot q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 \cdot q_3 - q_0 \cdot q_1) \\ 2(q_3 \cdot q_1 - q_0 \cdot q_2) & 2(q_3 \cdot q_2 + q_0 \cdot q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (2.12)$$

Ist die Rotation bekannt, so kann die Translation aus der folgenden einfachen Beziehung bestimmt werden:

$$t = \bar{r} - R\bar{l} \quad (2.13)$$

Sind keine korrespondierenden Punkte bekannt, so kann der in Besl und McKay (1992) präsentierte **ICP-Algorithmus** (Iterative Closest Point) angewandt werden. Dieser Algorithmus verwendet alle Regionen der Scans und nicht nur solche, in denen auch Verknüpfungspunkte liegen, was zu geringeren Diskrepanzen in der Berechnung führt (Pfeifer u. a., 2016). Im ersten Schritt des ICP-Algorithmus erfolgt die Bestimmung von Näherungswerten für die Transformation, welche auf die linke Punktwolke angebracht wird. Anschließend werden die folgenden vier Schritte iterativ wiederholt, bis ein Grenzwert τ für die Wurzel der mittleren quadratische Abweichung $RMSE$ unterschritten wird.

1. Bestimmung der Korrespondenzen aus den nächstliegenden Nachbarn in $\mathbf{r}$ für l_i anhand der minimalen Euklidischen Distanz.
2. Bestimmung der sich aus den Korrespondenzen ergebenden Transformation.
3. Anwendung der Transformation auf $\mathbf{l}$.

4. Berechnung des *RMSE*. Wenn $RMSE \geq \tau$, so wird der Prozess wiederholt.

Seit der Veröffentlichung des ICP wurden mehrere Methoden zur Steigerung der Zuverlässigkeit und Effizienz des Algorithmus vorgeschlagen. Eine Übersicht verschiedener Methoden befindet sich in Rusinkiewicz und Levoy (2001).

In einigen Varianten wird die Anzahl der betrachteten Punkte reduziert. So wird in Turk und Levoy (1994) zunächst eine Dreiecksvermaschung für beide Scans durchgeführt. Statt alle Punkte aus $\mathbf{l}$ zu verwenden, werden diejenigen Punktpaare entfernt, deren Distanzen zu groß sind oder in denen mindestens ein Punkt am Rand der Oberfläche liegt. Hierbei wird vorausgesetzt, dass sich die jeweiligen Punktwolken bereits hinreichend genau überlappen. Außerdem werden zur Beschleunigung des Algorithmus nicht alle Punkte verwendet. Stattdessen werden die Punkte vor den Iterationsschritten zufällig aus $\mathbf{l}$ gezogen. In Masuda u. a. (1996) werden in jeder Iteration neue Punkte zufällig ausgewählt. Weik (1997) schlägt vor, die zu betrachtenden Punkte aus dem Tiefenbild auszuwählen. Dies geschieht, indem nur Punkte mit einem hohen Gradientenwert gewählt werden. Zusätzlich werden Punkte, die nicht in beiden Datensätzen sichtbar sind, verworfen. In Kang u. a. (2009) sowie Barnea und Filin (2007) werden die Punkte ebenfalls anhand gradientenbasierter Merkmale im Distanzbild ausgewählt. Statt im Bild nach Merkmalspunkten zu suchen, verwenden Bae und Lichti (2008) die Krümmungswerte der Punkte, welche sich aus der Kovarianzmatrix einer lokalen Ebene ergeben (siehe Formel 2.18). In Rusinkiewicz und Levoy (2001) wird weiterhin vorgeschlagen, die Auswahl der Punkte anhand deren Normalenvektoren zu treffen. Dabei wird darauf geachtet, dass die Werte der Normalenvektoren möglichst gleichmäßig verteilt sind.

Eine weitere Möglichkeit, die Genauigkeit und Zuverlässigkeit des Algorithmus zu verbessern, ist die Methode zur Auswahl der Korrespondenzen in Schritt 1 zu ändern. Statt der minimalen Euklidischen Distanz wird in Chen und Medioni (1992) der Schnittpunkt des Normalenvektors von l_i mit der Oberfläche von $\mathbf{r}$ bestimmt. In Blais und Levine (1995) sowie Neugebauer (1997) wird l_i aus Richtung des Sensorstandpunktes auf die Oberfläche von $\mathbf{r}$ projiziert und der sich ergebende Punkt als Korrespondenz verwendet. Diese Methode wird in Benjemaa und Schmitt (1999), Dorai u. a. (1998), Weik (1997) und Pulli und Shapiro (2000) erweitert, indem zusätzlich anhand verschiedener Distanzmetriken ein korrespondierender Punkt aus $\mathbf{r}$ gewählt wird.

Sind korrespondierende Punkte gefunden, so ist ein weiterer Ansatz zur Verbesserung der Registrierungsergebnisse die Gewichtung der Korrespondenzen. Diese Gewichtung kann beispielsweise anhand der Punktdistanzen (Godin u. a., 1994) erfolgen. In Rusinkiewicz und Levoy (2001) wird außerdem vorgeschlagen, die Ähnlichkeit der lokalen Normalenvektoren und den Effekt des Sensorrauschens mit in die Gewichtung einzubeziehen.

Statt der Verwendung eines ICP-Algorithmus wird in Brenner u. a. (2008) die in Biber und Straßer (2003) vorgestellte Methode der Normal Distributions Transform (NDT) auf 3D-Daten erweitert. Die NDT ist eine alternative Darstellung von Punktwolken, in der die Punkte anhand lokaler Normalverteilungen dargestellt werden. Außerdem werden in Brenner u. a. (2008) Ebenen aus den jeweiligen Punktwolken segmentiert und zur Punktwolkenregistrierung verwendet. In Förstner und Khoshelham (2017) werden verschiedene Methoden zur direkten Bestimmung einer Transformation aus Ebenenkorrespondenzen vorgestellt und mit einem klassischen ICP-Algorithmus verglichen. Hierbei zeigte sich, dass die ebenenbasierten Ansätze genauer und laufzeiteffizienter als ein klassischer ICP sind.

Um eventuelle Messfehler des Gesamtsystems, welche zum größten Teil aus Fehlern der Lokalisierungseinheit bestehen, zu minimieren, wird insbesondere im Airborne-Laserscanning häufig eine

Streifenausgleichung durchgeführt, in der entsprechend auch die georeferenzierte Trajektorie des Messsystems einbezogen und korrigiert wird (Lichti und Skaloud, 2010). Ein Verfahren für die in dieser Arbeit relevanten Daten eines Mobile-Mapping-Systems findet sich in Hofmann (2017). Hierbei werden zusätzlich Passpunkte eines hochgenauen Referenznetzes in einer Streifenausgleichung nach der Methode der kleinsten Quadrate einbezogen. In Hebel und Stilla (2012) werden Restfehler in der Georeferenzierung durch einen Punkt-zu-Ebene-Abgleich mehrerer Airborne-Punktwolken behoben. Hierzu werden künstliche planare Objekte, wie Häuserdächer oder -Fassaden, in den Daten detektiert und zur Registrierung sowie zur Bore-sight-Kalibrierung verwendet.

2.1.5. Räumliche Datenstrukturen

Bei der Bearbeitung von Punktwolken treten häufig sehr großen Datenmengen auf. Eine Nachbarschaftssuche, wie sie beispielsweise in dem im vorherigen Abschnitt beschriebenen ICP-Algorithmus benötigt wird, kann somit zu sehr hohen Laufzeiten führen. Ohne eine passende Datenstruktur würde die Laufzeit hier $O(nm)$ betragen, da für jeden Punkt aus $\mathbf{r}$ die Distanz zu jedem Punkt aus $\mathbf{l}$ berechnet werden muss. Bei einer Punktzahl von $n, m = 100\,000$ müsste die Punktdistanzberechnung somit 10^{10} mal durchgeführt werden. Um die Anzahl der notwendigen Zugriffe zu verringern, werden räumliche Datenstrukturen verwendet.

In Elseberg u. a. (2012) werden verschiedene Datenstrukturen und deren Implementierungen hinsichtlich ihrer Laufzeit bei der Durchführung einer Nachbarschaftssuche untersucht: der K-d-Baum (Bentley, 1975), der Octree (Meagher, 1982) und der R-Baum (Guttman, 1984). Auf eine Untersuchung einer Gridfile-Implementierung (Nievergelt u. a., 1984) wird aufgrund der hohen Speicheranforderungen verzichtet. Es zeigt sich, dass für die verwendete Aufgabe der K-d-Baum und Octree ein besseres Laufzeitverhalten als der R-Baum aufweisen, die Geschwindigkeit jedoch stark von der jeweiligen Implementierung abhängt. Da in der vorliegenden Arbeit ein K-d-Baum verwendet wird, soll dieser hier auch näher vorgestellt werden.

Der in Bentley (1975) vorgestellte **K-d-Baum** ist ein Binärbaum, der sich unter anderem sehr gut für Aufgaben wie die Nachbarschaftssuche eignet (Vosselmann und Klein, 2010). Jeder Knoten des Baumes beinhaltet einen Punkt, welcher die jeweiligen Kindsknoten alternierend nach ihren Raumdimensionen aufteilt. Wird der Baum durchlaufen, so liegen alle Punkte, deren Werte der auf der jeweils betrachteten Ebene verwendeten Dimension x, y oder z kleiner als der Wert des Knotens sind, links von diesem, alle Punkte, deren Werte größer gleich sind, rechts von diesem. Die Blätter des Knotens enthalten anschließend Zeiger auf eine Liste mit den jeweils gespeicherten Punkten. Ist die Eingabereihenfolge der Punkte entsprechend vorsortiert, so ergibt sich ein ausbalancierter Baum. Die Zugriffszeit auf einen Punkt beträgt hier $O(\log n)$ (Bentley, 1975).

Eine weitere in dieser Arbeit verwendete Datenstruktur ist das **räumliche Hashing**. Hierbei werden die Punkte in ein regelmäßiges Raster überführt. Jede Rasterzelle beinhaltet eine Menge von Punkten, auf die über einen eindeutigen Hashwert zugegriffen werden kann. In Pabst u. a. (2010) ergibt sich der Hashwert $H(P)$ eines Punktes $P(x, y, z)$ aus

$$H(P) = z \cdot y_{dim} \cdot x_{dim} + y \cdot x_{dim} + x, \quad (2.14)$$

wobei x_{dim} und y_{dim} die maximale Ausdehnung der Punkte in der jeweiligen Dimension beschreiben. Durch das Hashing kann auf die jeweiligen Zellen in $O(1)$ zugegriffen werden. Innerhalb der Rasterzellen können die Punkte in einer Liste oder auch in einer weiteren räumlichen Datenstruktur gespeichert werden. Ein Problem bei der Verwendung des Hashings ist, dass die Zellgrößen statisch sind und sich nicht den vorliegenden Daten anpassen können. Dies kann durch die Verwendung

eines Grid-Files behoben werden. Da in der vorliegenden Arbeit die Struktur der Eingangsdaten jedoch bekannt ist, wurde auf die Verwendung eines Grid-Files verzichtet.

2.1.6. Segmentierung in Punktwolken

Der Begriff Segmentierung stammt aus der Bildverarbeitung und beschreibt die Aufteilung eines Bildes in konsistente Regionen oder Objekte (Gonzalez u. a., 2002). Übertragen auf LiDAR-Daten ist es die Aufteilung der Punktwolke in einzelne Segmente mit übereinstimmenden Kriterien, wobei die Segmente räumlich miteinander verbunden sind. Häufig wird eine Segmentierung dazu genutzt, bestimmte geometrische Objekte in den Daten zu detektieren (Vosselmann und Klein, 2010). Zur Segmentierung in Punktwolken existiert eine Vielzahl an Algorithmen. Dieser Abschnitt orientiert sich an den später auch verwendeten Algorithmen. Eine Übersicht über weitere Verfahren befindet sich in Vosselmann und Klein (2010) sowie Nguyen und Le (2013).

Ebenenbestimmung und Normalenvektoren

In vielen Situation ist es hilfreich, die Struktur in der Nachbarschaft eines Punktes zu untersuchen. So gibt beispielsweise der Normalenvektor n Auskunft über die Struktur und Ausrichtung der Punktumgebung. Ein Punkt auf einer horizontalen Ebene zeichnet sich beispielsweise durch einen zur Ebene orthogonalen Normalenvektor mit einem hohen Wert in z - und niedrigen Werten in x - und y -Richtung aus. Die Ebenenparameter können in der Hesseschen Normalform aus dem normierten Normalenvektor n_0 und der Distanz der Ebene zum Ursprung des Koordinatensystems d beschrieben werden:

$$d = x \cdot n_0 \quad (2.15)$$

Der Normalenvektor kann wie in Kirschvink (1980) gezeigt aus einer Hauptkomponentenanalyse (Principal Component Analysis, PCA) bestimmt werden. Er ergibt sich aus einer Eigenwertzerlegung der Kovarianzmatrix der schwerpunktreduzierten Punkte (x'_i, y'_i, z'_i) (Formeln 2.8 und 2.9) der Ebene.

$$H = \begin{pmatrix} \sum x_i'^2 & \sum x_i' y_i' & \sum x_i' z_i' \\ \sum x_i' y_i' & \sum y_i'^2 & \sum y_i' z_i' \\ \sum x_i' z_i' & \sum y_i' z_i' & \sum z_i'^2 \end{pmatrix} \quad (2.16)$$

Der bereits normierte Normalenvektor ist nun der Eigenvektor zum kleinsten Eigenwert von H . Hieraus lässt sich d mit Hilfe des Schwerpunkts $\bar{p}$ aus der folgenden Beziehung bestimmen:

$$d = -n^T \bar{p} \quad (2.17)$$

Region Growing

Das Region Growing hat seine Ursprünge in der Bildverarbeitung. In Ballard und Brown (1982) werden Regionen R_k als eine Ansammlung von Punkten definiert, die bestimmte Homogenitätskriterien erfüllen. Hierbei gelten folgende Bedingungen für diese Regionen:

1. Die Punkte der einzelnen Regionen müssen miteinander verbunden sein.
2. Die Gesamtmenge aller Regionen ergibt das Gesamtbild I .

$$\bigcup_{k=1}^m R_k = I$$

3. Es treten keine Überschneidungen zwischen Regionen auf.

$$R_i \cap R_j = \emptyset, i \neq j$$

4. In allen Regionen gilt ein Homogenitätskriterium

$$H(R_K) = \text{wahr für alle } k$$

5. Die Vereinigung zweier benachbarter Regionen darf nicht das Homogenitätskriterium erfüllen.

$$H(R_i \cup R_j) = \text{falsch für } i \neq j \text{ und } R_i, R_j \text{ benachbart}$$

Diese Homogenitätskriterien lassen sich leicht auf 3D-Punktwolken übertragen. Für Kriterium 1 kann die direkte Nachbarschaft im Bild durch den nächstliegenden Nachbarn in der Punktwolke ersetzt werden, für den eine maximale euklidische Distanz gilt.

In Adams und Bischof (1994) wird das Seeded Region Growing vorgestellt. Hierbei werden zu Beginn des Algorithmus Saatpunkte (Seeds) detektiert. Die Auswahl dieser Seeds ist von den jeweiligen Homogenitätskriterien abhängig. Sollen Ebenen detektiert werden, können beispielsweise die Abstände der Punkte zur Ebene sowie, wie in Hoffman und Jain (1987) vorgeschlagen, zusätzlich die Normalenvektoren der jeweiligen Punkte als Kriterium verwendet werden. Ein guter Saatpunkt zeichnet sich hierbei durch eine geringe Krümmung σ_P aus. Die Krümmung an einem Punkt P kann, wie in Rusu (2009) gezeigt, aus den Eigenwerten λ_j der Kovarianzmatrix der lokalen Ebene um P approximiert werden:

$$\sigma_P = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \text{ wobei } \lambda_0 \leq \lambda_1 \leq \lambda_2 \quad (2.18)$$

Ausgehend von den Saatpunkten werden anschließend deren jeweilige Nachbarn überprüft. Erfüllt ein Nachbarspunkt die jeweiligen Kriterien der Region, so wird er der sich zunächst aus dem einzelnen Saatpunkt ergebenden Region hinzugefügt. Im weiteren Verlauf wird ausgehend von den zur Region hinzugefügten Punkten die Region rekursiv erweitert, bis keine neuen Punkte mehr hinzugefügt werden können. Zur Region hinzugefügte Punkte werden als behandelt markiert und können somit keinen weiteren Regionen hinzugefügt werden. Dieser Prozess wird wiederholt, bis alle Saatpunkte bearbeitet wurden. Es ist anzumerken, dass die von Ballard und Brown (1982) aufgestellte Bedingung 2 hier nicht zwingend gilt, da gegebenenfalls nur eine Teilpunktmenge der gesamten Punktwolke zu Regionen vereinigt wird.

Sollen planare Segmente in den Daten detektiert werden, so ergibt sich der Normalenvektor der Region im ersten Schritt aus dem Normalenvektor des Saatpunktes. Wurden neue Punkte hinzugefügt, so kann der Normalenvektor mithilfe einer Hauptkomponentenanalyse (Abschnitt 2.1.6) bestimmt und anschließend mit den Normalen der aktuellen Nachbarschaftspunkten verglichen werden. Aus Effizienzgründen wird in Vosselmann und Klein (2010) vorgeschlagen, die Berechnung der Ebenenparameter nicht nach jedem neu hinzugefügten Punkt durchzuführen, sondern nur, wenn die Ebenen um einen bestimmten prozentualen Anteil (20–50 %) gewachsen sind.

Aus dem sequentiellen Wachsen der Ebenen ergibt sich, dass der Algorithmus gierig (greedy) ist. Die Lösung hängt von der Reihenfolge der betrachteten Ebenen ab und führt nicht zwangsläufig zu einem optimalen Ergebnis. Vosselmann und Klein (2010) schlagen daher vor, dass auch bereits behandelte Punkte beim Wachsen der Ebene betrachtet werden. Kommt ein Punkt für mehrere

Ebenen in Frage, so soll er der Ebene hinzugefügt werden, zu der er die geringsten Abweichungen hat.

RANSAC

Das in Bolles und Fischler (1981) präsentierte Verfahren RANSAC (Random Sample Consensus) ist ursprünglich ein Verfahren zur robusten Parameterschätzung, welches insbesondere bei einer hohen Anzahl an Ausreißern vergleichsweise effektiv ein vorgegebenes Modell findet. Es eignet sich jedoch auch zur Segmentierung in Punktwolken, was im Folgenden anhand des Beispiels einer Ebenensegmentierung dargestellt werden soll.

Grundlage des Verfahrens ist, dass der folgende Ablauf so oft wiederholt wird, bis eine maximale Anzahl an Iterationen k erreicht oder das gesuchte Modell gefunden wird:

- Zufällige Wahl an n Beobachtungen aus den gegebenen Daten, wobei sich n aus der Mindestanzahl an Beobachtungen ergibt, die zum Aufstellen eines Modells benötigt werden. Im Falle einer Ebenenschätzung werden mindestens $n = 3$ Punkte benötigt.
- Bestimmung der Ebenenparameter aus den zufällig gewählten Punkten.
- Ermittlung des sogenannten Consensus Sets, welches sich aus den zum Modell kompatiblen Beobachtungen ergibt.

Die minimale Anzahl an Iterationen k , die erforderlich ist, um eine vorgegebene Zielwahrscheinlichkeit z zu erreichen, kann, wie in Bolles und Fischler (1981) vorgestellt, aus n , der Wahrscheinlichkeit w , dass eine Beobachtung zum Modell kompatibel ist, und z bestimmt werden:

$$k \geq \frac{\log(1 - z)}{\log(1 - w^n)}, \quad (k \in \mathbb{N}) \quad (2.19)$$

Liegt beispielsweise eine Punktwolke vor, in der $w = 50\%$ der Punkte Bestandteil einer Ebene sind, so ergibt sich die Anzahl an benötigten Iterationen, um mit einer Wahrscheinlichkeit von 99% diese Ebene zu detektieren, zu 35 Durchläufen.

Wird die Anzahl an maximalen Iterationen oder während einer Iteration die Mindestanzahl an zum Modell gehörigen Punkten erreicht, so können die endgültigen Ebenenparameter aus den Punkten des maximalen Consensus Sets bestimmt werden.

Segmentierung stangenförmiger Objekte

In Abschnitt 6.1 sollen stangenförmige Objekte zur Fahrzeuglokalisierung verwendet werden. Die Segmentierung der Stangen erfolgt in dieser Arbeit vornehmlich durch den in Brenner (2009a) vorgestellten Algorithmus. Hierbei werden drei Kerneigenschaften stangenförmiger Objekte ausgenutzt: die vertikale Ausrichtung der Stangen, eine Anhäufung von Punkten in einem schmalen Radius r_1 um den Stangenmittelpunkt und einen Bereich im Umkreis $(r_2 - r_1)$ der Stange, in dem nur wenige Punkte auftreten (siehe Abbildung 2.3). Da an den Stangen im Straßenraum oftmals Elemente wie Schilder oder Ampeln angebracht sind oder von den Stangen weitere Stützen abgehen, wird die Struktur der Objekte wie in Abbildung 2.3 (rechts) skizziert, in zylindrischen Schichten untersucht. Ein Objekt wird als Stange identifiziert, sobald für eine Position in x/y eine Mindestanzahl von Schichten auftritt, die die oben genannten Eigenschaften erfüllen. Im Anschluss können die in dem inneren Bereich der Schichten befindlichen Punkte zur Bestimmung der exakten Position genutzt werden.

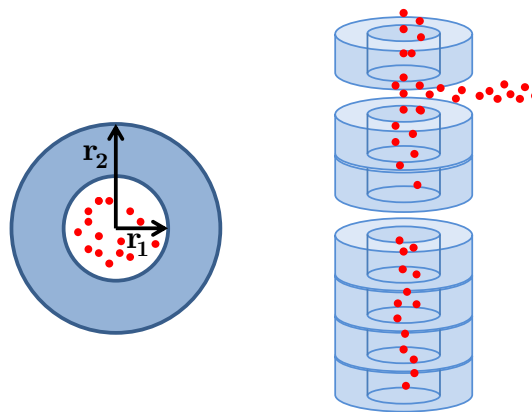


Abbildung 2.3.: Segmentierung stangenförmiger Objekte in Punktwolken. Die untersuchten Objekte werden in Brenner (2009a) in vertikale Schichten aufgeteilt (rechts). Innerhalb dieser Schichten wird der Bereich um den Stangenmittelpunkt mit dem Radius r_1 sowie ein äußerer Bereich $r_2 - r_1$ (links) untersucht.

Zur Segmentierung stangenförmiger Objekte existieren eine Menge weiterer Algorithmen. Wu u. a. (2017) wenden beispielsweise Verfahren des maschinellen Lernens an, um freistehende Objekte im Straßenraum zu klassifizieren. Lehtomäki u. a. (2010) detektieren typische Stangenmuster in den einzelnen Scanlinien, welche anschließend aggregiert werden. Anhand bestimmter Kriterien wird überprüft, ob es sich bei den resultierenden Objekten um stangenförmige Objekte handelt. In Xu u. a. (2018) werden bestimmte Merkmale potentieller Stangen in Histogrammen akkumuliert. Die Klassifizierung der Histogramme zur Detektion der stangenförmigen Objekte wird mithilfe eines Random Forests durchgeführt. Weiterhin nennenswert ist die in Polewski u. a. (2016) vorgestellte Methode zur Detektion toter Baumstämme in Airborne-LiDAR-Daten. Hierbei werden aktives und teilüberwachtes Lernen kombiniert und das durch die Shannon-Entropie (Lin, 1991) regularisierte Regressionsmodell zur Familie der Renyi-Entropien (Xu und Erdogmuns, 2010) verallgemeinert. Dies führt zu einer Steigerung der Performance im Vergleich zu nicht-regularisierten Verfahren, welche sich insbesondere bei kleinen Trainings-Datensätzen bemerkbar macht.

2.2. Maschinelles Lernen

Als Haupteigenschaft der Algorithmen des maschinellen Lernens wird in Michalski u. a. (2013) beschrieben, dass diese in der Lage sind, sich zu verbessern, indem sie Erfahrungen sammeln. Typischerweise wird hierbei zwischen zwei Konzepten unterschieden: dem überwachten und dem unüberwachten Lernen (Hastie u. a., 2001).

Beim überwachten Lernen lernt der Algorithmus, wie aus gewissen Eingangsdaten gewünschte Ergebnisse erzielt werden können. Der Lernprozess wird als Training bezeichnet, die entsprechenden Trainingsdaten T bestehen aus verschiedenen Eingangsdaten x^T , zu denen die jeweiligen Ergebnisse y^T bekannt sind. Durch das Training soll der Algorithmus eine Funktion $\hat{f}(x)$ lernen, die die tatsächliche Beziehung $f(x)$ der Eingangsdaten zu den Ergebnissen approximiert. Somit können die Ergebnisse aus den Eingangsdaten prädiziert werden. Aus der Differenz $y_i^T - \hat{f}(x_i^T)$ der tatsächlichen und prädizierten Werte kann hierbei eine Fehlerfunktion bestimmt und im Training verwendet werden. Diese Fehlerfunktion wird auch als Loss bezeichnet. Beim unüberwachten Lernen werden keine bekannten Ergebnisse verwendet. Stattdessen erfolgt eine Bestimmung bestimmter Eigenschaften der Eingangswerte alleine aus den Beobachtungen (Hastie u. a., 2001).

In dieser Arbeit werden verschiedene Methoden des maschinellen Lernens angewandt, die in den folgenden Abschnitten beschrieben werden. In Abschnitt 6.2 wird ein K-Means-Algorithmus zum Clustering, also zur Gruppierung von Laserscans, verwendet. Ein Random Forest wird in Abschnitt 5.2.1 zur Klassifizierung von Vegetation in 3D-Punktwolken angewendet. Ebenfalls im Kapitel 6.2 finden auch neuronale Netze ihre Anwendung. Außerdem sollen verschiedene Maße zur Bewertung einer Klassifizierung vorgestellt werden.

2.2.1. K-Means-Algorithmus

Der in MacQueen u. a. (1967) vorgestellte K-Means-Algorithmus ist eine Methode des unüberwachten Lernens und wird zum Clustering von Daten verwendet. Hierbei werden k Mittelpunkte (Means) $\{\mu_k\}$ der auftretenden Gruppierungen (Cluster) im multidimensionalen euklidischen Raum bestimmt, wobei k als Eingangsparameter bekannt sein muss. Das Ziel des Algorithmus ist die Minimierung der quadrierten Summe der euklidischen Distanzen eines jeden Datenpunktes x_n zum jeweils nächstliegenden Mittelpunkt (Bishop, 2007). Zur Beschreibung des Fehlermaßes J wird ein Binärindikator $r_{nk} \in \{0,1\}$ eingeführt, der beschreibt, zu welchem Cluster ein Punkt x_n zugeordnet wurde, sodass in diesem Fall $r_{nk} = 1$ gilt. Für alle anderen Cluster ist $r_{nk} = 0$. J kann nun durch

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2 \quad (2.20)$$

bestimmt werden. Ziel ist es, r_{nk} und $\{\mu_k\}$ so zu wählen, dass J minimal wird. Hierzu wird iterativ vorgegangen. In jeder Iteration wird zunächst jeder Punkt seinem nächstliegenden Mittelpunkt zugewiesen, wodurch sich neue Werte für r_{nk} ergeben. Im zweiten Schritt werden dann die Koordinaten der Mittelpunkte $\{\mu_k\}$ aus den ihnen zugewiesenen Punkten neu bestimmt. Die Initialwerte für $\{\mu_k\}$ können zufällig gewählt werden (MacQueen u. a., 1967). Die Iteration wird so oft wiederholt, bis der Algorithmus konvergiert oder ein Maximalwert für die Anzahl der Iterationen erreicht wurde. Dadurch, dass der Wert von J in jedem Iterationsschritt verringert wird, konvergiert der Algorithmus in jedem Fall, allerdings nicht zwangsweise zu einem globalen Minimum (Bishop, 2007).

2.2.2. Bewertung einer Klassifizierung

Bevor auf Methoden zur Klassifizierung eingegangen wird, sollen an dieser Stelle verschiedene Metriken zur Bewertung von Klassifizierungsergebnissen vorgestellt werden. Zur Darstellung der Ergebnisse einer Klassifizierung bietet sich eine Konfusionsmatrix wie in Tabelle 2.1 dargestellt an. Hierbei wird eine binäre Klassifikation angenommen, in der die Elemente hin-

Tabelle 2.1.: Aufbau einer Konfusionsmatrix zur Beurteilung von Klassifizierungsergebnissen.

Zustand	Prädiktion	
	<i>wahr</i>	<i>falsch</i>
<i>wahr</i>	TP	FN
<i>falsch</i>	FP	TN

sichtlich eines bestimmten Zustandes klassifiziert werden sollen. Trifft dieser Zustand zu, werden sie als *wahr* markiert, andernfalls als *falsch*. Es ergeben sich vier mögliche Bewertungen:

- True Positive (TP)** Ein Zustand wurde korrekterweise als *wahr* klassifiziert
- False Positive (FP)** Ein Zustand wurde fälschlicherweise als *wahr* klassifiziert
- False Negative (FN)** Ein Zustand wurde fälschlicherweise als *falsch* klassifiziert
- True Negative (TN)** Ein Zustand wurde korrekterweise als *falsch* klassifiziert

Aus den jeweiligen Bewertungen lassen sich verschiedene Bewertungsmetriken ableiten. In Michalski u. a. (2013) werden die Precision- und Recall-Metrik sowie der F₁-Score genannt. Die Precision gibt an, wie viele der als *wahr* prädizierten Elemente korrekt prädiziert wurden:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.21)$$

Der Recall-Wert gibt an, wie viele Elemente der Klasse *wahr* detektiert, also als *wahr* prädiziert wurden:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.22)$$

Der F₁-Score gibt Auskunft über die Genauigkeit der Klassifizierung und ist das harmonische Mittel zwischen Precision und Recall. Er ergibt sich aus

$$F_1 = \frac{2 \text{ Precision Recall}}{\text{Precision} + \text{Recall}} \quad (2.23)$$

2.2.3. Random Forests

Random Forests sollen in dieser Arbeit zur Klassifizierung von Vegetation in 3D-Punktwolken (siehe Abschnitt 5.2.1) sowie zur Detektion dynamischer Punkte in einzelnen Zeilenscans (siehe Abschnitt 5.2.2) verwendet werden. Das in Breiman (2001) vorgestellte Verfahren basiert auf der Kombination mehrerer Entscheidungsbäume. Zunächst soll daher auf die Funktionsweise von Entscheidungsbäumen eingegangen werden.

Ein **Entscheidungsbaum** ist eine Methode des überwachten Lernens, die den Parameterraum ausgehend von einem Wurzelknoten rekursiv aufteilt. Die Knoten des Baumes werden dabei so gewählt, dass sie die Daten möglichst optimal aufteilen, sodass diese möglichst korrekt klassifiziert werden. Um zu entscheiden, anhand welcher Attribute an den Knoten geteilt wird, kann der

Informationsgewinn I betrachtet werden (Michalski u. a., 2013). Um diesen zu bestimmen, wird häufig die Shannon-Entropie verwendet. Diese gibt Auskunft über die beseitigte Unsicherheit durch eine bestimmte Sammlung an Daten S , die aus Attributen und Zuständen besteht. Kann S c verschiedene Zustände einnehmen und ist p_i die Wahrscheinlichkeit für das Eintreten eines bestimmten Zustands, so ergibt sich die Entropie aus

$$\text{Entropie}(S) = \sum_{i=1}^c -p_i \log_2 p_i \quad (2.24)$$

Je höher der Wert für die Entropie, desto geringer ist die beseitigte Unsicherheit. Der Informationsgewinn I ergibt sich anschließend aus der Differenz zwischen der Entropie aller Zustände und der gewichteten Entropie der einzelnen Untermengen der Zustände S_z , die sich aus der Aufteilung des Baumes anhand der Attribute A ergibt:

$$I(S) = \text{Entropie}(S) - \sum_{z \in \text{Zustände}(A)} \frac{|S_z|}{|S|} \text{Entropie}(S_z) \quad (2.25)$$

Dies soll anhand des folgenden Beispiels verdeutlicht werden. Hierbei soll für gemessene Objekte anhand deren zugehörigen Punktmengen eine binäre Klassifizierung durchgeführt werden, welche aussagt, ob diese als Häuserfassade gelten oder nicht. Als Attribute dienen die Höhe h (Ausdehnung in z -Richtung) der Punktmenge sowie die mittlere Krümmung der Punkte $\overline{\sigma_P}$. Die Eingangsdaten bestehen aus fünf Objekten. Die jeweiligen Attribute und Zustände sind in Tabelle 2.2 aufgeführt.

Tabelle 2.2.: Beispiel-Trainingsdaten für einen Entscheidungsbaum.

Index	h [m]	$\overline{\sigma_P}$ [m]	Zustand
1	6	0,05	<i>wahr</i>
2	6	0,8	<i>falsch</i>
3	8	0,15	<i>wahr</i>
4	2	0,1	<i>falsch</i>
5	5	0,3	<i>falsch</i>

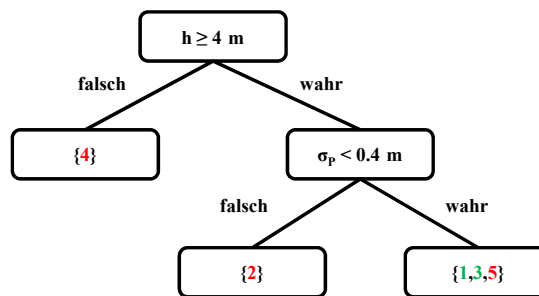


Abbildung 2.4.: Beispiel eines Entscheidungsbaums. Elemente, deren Zustand wahr ist, sind grün, Elemente deren Zustand falsch ist, rot eingefärbt.

Aus den Daten kann beispielsweise ein Entscheidungsbaum aufgestellt werden, der im Wurzelknoten anhand des Attributes h den Baum aufteilt und im rechten Teilbaum anschließend anhand des Attributes σ_P , siehe Abbildung 2.4. Die Wahrscheinlichkeit für das Auftreten einer Fassade

beträgt 0,4, die Wahrscheinlichkeit, dass es sich um keine Fassade handelt 0,6. Die Gesamtentropie ergibt sich somit zu

$$\text{Entropie}(S) = -0,4 \log_2 0,4 - 0,6 \log_2 0,6 = 0,97$$

Anhand der Gesamtentropie und der Entropie der Blätter des Baumes kann nun der Informationsgewinn berechnet werden. In diesem einfachen Fall ergibt sich für den linken Teilbaum sowie für den von der zweiten Ebene abgehenden linken Teilbaum eine Entropie von 0. Dessen Nachbarblatt enthält ein Objekt mit dem Zustand *falsch* sowie zwei Objekte mit dem Zustand *wahr*. Somit ergibt sich dessen Entropie zu

$$\text{Entropie}(S_2) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0,92$$

Der Informationsgewinn beträgt somit

$$I(S) = 0,97 - (0 + \frac{3}{5} 0,92 + 0) = 0,42$$

Als weiteres Kriterium zur Beurteilung der gewählten Knoten kann auch der Gini-Index verwendet werden. Zusätzliche Informationen zur Verwendung des Index in Entscheidungsbäumen finden sich in Rokach und Maimon (2009) und Breiman u. a. (1984).

Im oberen Beispiel werden die Knoten an der jeweiligen Attribut-Achse des Koordinatensystems geteilt. Dies entspricht einer Teilung durch eine achsparallele Hyperebene. Wie in Criminisi u. a. (2012) gezeigt, können jedoch auch andere, teils komplexere geometrische Primitive verwendet werden. Als Beispiele werden eine nicht-achsparallele Hyperebene (linear) sowie eine quadratische Funktion aufgeführt.

Random Forests verwenden nicht nur einen, sondern mehrere Entscheidungsbäume. Diese Bäume weisen Unterschiede auf, was durch verschiedene Wege herbeigeführt werden kann. Die zwei populärsten Methoden sind eine zufällige Wahl der Trainingsmenge aus der Gesamtmenge an Trainingsdaten sowie eine zufällige Wahl der Attribute, durch die der Baum an den Knoten geteilt wird (Criminisi u. a., 2012). Auch eine Kombination beider Methoden ist möglich. Zur Wahl der Attribute wird zunächst für jeden Baum eine zufällige Untermenge an Attributen A_j aus A gewählt. Anschließend wird, wie in Criminisi u. a. (2012) aufgezeigt, jeder Knoten θ_j durch die folgende Funktion optimiert:

$$\theta_j^* = \arg \max_{\theta_j \in A_j} I_j \quad (2.26)$$

Zur zufälligen Wahl der Trainingsdaten eines jeden Baumes kann das Bagging (bootstrap aggregating) verwendet werden. Hierbei werden die Daten durch Ziehen mit Zurücklegen aus der Gesamtmenge gezogen. Bagging eignet sich vor allem zur Erhöhung der Genauigkeit der Klassifizierung, wenn zufällige Attribute gewählt werden (Breiman, 2001).

2.2.4. Neuronale Netze

Das Ziel eines neuronalen Netzes ist es, eine Funktion f^* zu approximieren. Ein Netz besteht dabei aus verschiedenen Parametern θ . Aus den Eingangsdaten (Input) $\mathbf{x}$ sollen die Ergebnisse

$$\mathbf{y} = \mathbf{f}(\mathbf{x}; \theta) \quad (2.27)$$

bestimmt werden. Die Approximation von f^* geschieht durch die Verwendung von Trainingsdaten mit bekanntem Output, wodurch die entsprechenden Parameter bestimmt werden sollen (Goodfellow u. a., 2016a).

Typischerweise bestehen neuronale Netze aus mehreren Schichten (Layer), welche jeweils verschiedene Funktionen ausführen können. Die Tiefe des Netzes ergibt sich aus der Anzahl der Layer n . In der vorliegenden Arbeit werden die Notationen aus Goodfellow u. a. (2016a) verwendet. Hierbei werden die Layer mit $f^{(k)}$ bezeichnet, wobei $k \in \{0, \dots, n\}$ für die Position des Layers im Netz steht. Der Input-Layer ist somit $f^{(0)}$, der Output-Layer $f^{(n)}$. Layer $f^{(1)}$ bis $f^{(n-1)}$ werden als hidden Layer, also versteckte Layer, bezeichnet, da ihre Werte in der Regel nicht ausgegeben werden. Ein Beispiel eines einfachen neuronalen Netzes mit einem hidden Layer ist in Abbildung 2.5 skizziert (Goodfellow u. a., 2016a).

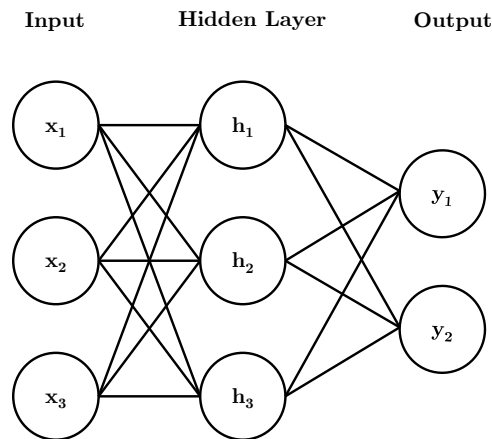


Abbildung 2.5.: Beispiel eines neuronalen Netzes mit einem Input-Layer, einem hidden Layer und einem Output-Layer.

Jeder hidden Layer besteht aus einem Vektor an Neuronen, die dem Netz ihren Namen geben. Die Anzahl der Elemente innerhalb des Vektors gibt die Breite des Layers an. Diese Neuronen erhalten eine bestimmte Menge an Eingangswerten $\mathbf{x}$ und rechnen diese in ein skalares Ausgangssignal um. Hierfür wird in der Regel zunächst ein lineares Modell, bestehend aus einer Menge an Gewichten $\mathbf{w}$ und einer Verschiebungskonstante (Bias) b , verwendet, sodass gilt:

$$\mathbf{f}(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^T \mathbf{w} + b \quad (2.28)$$

Die Nicht-Linearität eines Layers wird anschließend über eine Aktivierungsfunktion

$$\mathbf{g}(\mathbf{f}(\mathbf{x}^{(k)}, \mathbf{w}^{(k)}, b^{(k)})) \quad (2.29)$$

erzielt.

In den folgenden Abschnitten soll das gradientenbasierte Lernen, durch welches die Parameter des Netzes θ bestimmt werden, und der Aufbau der hidden Layer näher erläutert werden. Weiterhin soll auf die in der vorliegenden Arbeit verwendete Faltung (Convolution) und einer speziellen Form des Netzes, den Autoencoder, eingegangen werden.

Gradientenbasiertes Lernen

Durch ein iteratives, gradientenbasiertes Lernen ist ein neuronales Netz in der Lage, auch eine Funktion $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$ zur Lösung eines nicht-linearen Problems zu finden. Dies geschieht durch die Minimierung einer Kostenfunktion $J(\boldsymbol{\theta})$. In den meisten Netzen liegt eine Verteilung $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ vor. Zur Minimierung der Kosten wird in der Regel das Verfahren der Maximum-Likelihood-Schätzung angewendet. Hierbei werden die Kosten aus der Kreuzentropie zwischen Trainingsdaten und den entsprechenden Prädiktionswerten des Modells $\mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \hat{p}}$ bestimmt:

$$J(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \hat{p}} \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \quad (2.30)$$

Wie in Goodfellow u. a. (2016a) gezeigt, kann aus Gleichung 2.30 auch der mit $\frac{1}{2}$ skalierte mittlere quadratische Fehler plus einer nicht von $\boldsymbol{\theta}$ abhängigen Konstante (c) verwendet werden:

$$J(\boldsymbol{\theta}) = \frac{1}{2} \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \hat{p}} \|\mathbf{y} - \mathbf{f}(\mathbf{x}; \boldsymbol{\theta})\|^2 + c \quad (2.31)$$

Neben der Bestimmung einer Kostenfunktion ist auch die Wahl einer Output-Funktion wichtig. Diese wandelt das Ergebnis der Neuronen der letzten Layer in Wahrscheinlichkeiten für das Auftreten bestimmter Outputs y um. Liegt eine binäre Klassifizierung vor, so muss eine Bernoulli-Verteilung über y in Abhängigkeit von $\mathbf{x}$ gefunden werden. Um einen starken Gradienten zu erzielen, der sich positiv auf das Gradientenverfahren auswirkt, kann eine **Sigmoid-Funktion**, wie die logistische Funktion (siehe Abbildung 2.6), verwendet werden. Diese lässt sich für einen Output z durch

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (2.32)$$

bestimmen (Goodfellow u. a., 2016a).

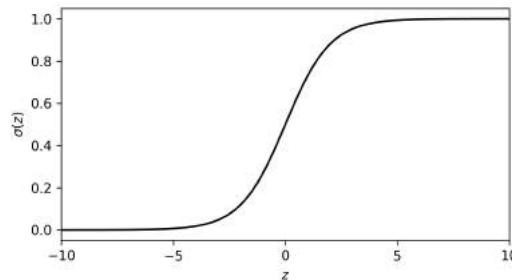


Abbildung 2.6.: Verlauf der logistischen Sigmoid-Funktion.

Bei n möglichen Zuständen muss für jeden Zustand eine Wahrscheinlichkeit ausgegeben werden, woraus sich der Vektor $\hat{y}$ ergibt, der sich entsprechend zu 1 summieren muss. Hierfür wird in der Regel eine sogenannte **Softmax-Funktion** verwendet, welche eine Generalisierung der Sigmoid-Funktion ist und sich aus der einfachen Beziehung

$$\text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \quad (2.33)$$

ergibt (Goodfellow u. a., 2016a).

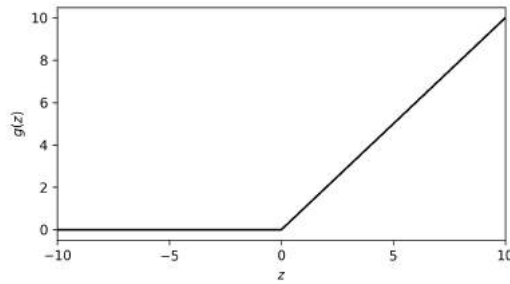


Abbildung 2.7.: Verlauf der ReLU.

Hidden Layer

Die Neuronen eines hidden Layers bestehen üblicherweise aus einer Affintransformation (Gleichung 2.28) sowie einer nicht-linearen Aktivierungsfunktion $g(z)$. Welche Form der Aktivierungsfunktion für welchen Layer am besten geeignet ist, lässt sich nicht pauschal beantworten. In Goodfellow u. a. (2016a) wird empfohlen, mehrere Funktionen hinsichtlich ihrer Performance zu testen. Häufig werden **Rectified Linear Units** (ReLU) verwendet, welche sich aus

$$g(z) = \max\{0, z\} \quad (2.34)$$

ergeben (siehe Abbildung 2.7). Dies hat den Vorteil, dass ihre Gradienten selbst für beliebig kleine oder große z entweder 0 für $z \leq 0$ oder 1 für $z > 0$ sind und sich die Funktion somit sehr gut zum gradientenbasierten Lernen eignet. Um mögliche Probleme zu beheben, die sich aus der festen Nichtaktivierung der Funktion für $z \leq 0$ ergeben, wird in Maas u. a. (2013) eine sogenannte leaky ReLU vorgeschlagen, die zusätzlich einen Term $0.01 \cdot \min\{0, z\}$ addiert (Goodfellow u. a., 2016a).

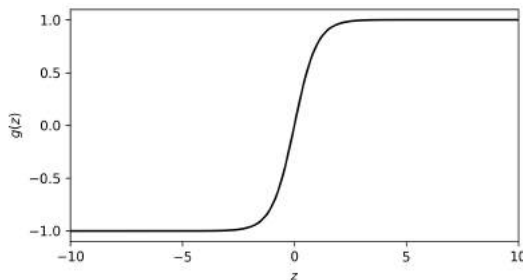


Abbildung 2.8.: Verlauf der Tangens-hyperbolicus-Funktion.

Weitere bekannte Aktivierungsfunktionen, die vor allem vor der Einführung der ReLU verwendet wurden, sind die logistische Sigmoid-Funktion (Gleichung 2.32) sowie der Tangens hyperbolicus $g(z) = \tanh(z)$ (Abbildung 2.8). Beide Funktionen haben jedoch den Nachteil, dass sie nur sensitiv für Werte nahe 0 sind und für große z -Werte sättigen, was sich negativ auf das gradientenbasierte Lernen auswirken kann (Goodfellow u. a., 2016a).

Backpropagation und Stochastic Gradient Descent

Zur Bestimmung des Gradienten der Kostenfunktion $\nabla_{\theta} J(\theta)$ wird das Verfahren der **Backpropagation**, wie in Rumelhart u. a. (1986) gezeigt, verwendet. Hierbei werden angefangen beim letzten hidden Layer $f^{(n-1)}$ des Netzes die Ableitungen $\frac{\delta J}{\delta \theta^k}$ der Kosten nach den Parametern

des Layers bestimmt. Diese Ableitungen geben an, wie sich die Parameter und damit die Aktivierungen der Neuronen des Layers verändern müssen, um die Gesamtkosten zu minimieren. Die resultierenden Gradienten werden anschließend weiter verwendet, um wiederum die Gradienten des vorherigen Layers $f^{(n-2)}$ zu berechnen. Dieses Verfahren wird bis zum ersten hidden Layer des Netzes $f^{(1)}$ weitergeführt. Eine ausführliche Beschreibung des Verfahrens in Hinblick auf die Verwendung in neuronalen Netzen ist in Rumelhart u. a. (1986) und Nielsen (2015) zu finden.

Die Minimierung der Kostenfunktion durch Backpropagation wird parallel für eine Vielzahl an Trainingsbeispielen durchgeführt und anschließend gemittelt. Da die gleichzeitige Einbeziehung der gesamten Trainingsdaten die Berechnung stark verlangsamen würde, wird häufig das stochastische gradientenbasierte Lernen (**Stochastic Gradient Descent**, SGD) verwendet (Nielsen, 2015). Anstatt die Kosten für die gesamte Trainingsmenge zu bestimmen, wird iterativ eine zufällig gewählte Untermenge (Batch) verwendet. Die Parameter des Netzes werden nach jeder Iteration in Richtung des aus dem aktuellen Batch gemittelten Gradienten verändert, wobei die Schrittweite zusätzlich durch die Lernrate bestimmt wird. In Goodfellow u. a. (2016b) wird empfohlen, die Lernrate kontinuierlich mit jeder Iteration zu verringern. Mit einer anfänglich größeren Schrittweite kann das Netz schnell ein lokales Minimum finden. Durch die Verkleinerung der Schrittweite können die Parameter anschließend präziser bestimmt werden, als bei einer höheren Lernrate.

Der **Adam-Optimierer** (Kingma und Ba, 2014) definiert die Lernrate anhand des quadratischen Mittels der vergangenen Gradienten sowie der Varianz des quadratischen Mittelwertes, was zur einer im Vergleich zum klassischen SGD effizienteren Reduzierung des Gradienten führt.

In Hinton u. a. (2012) wird das **Rprop-Verfahren** vorgestellt. Hierbei wird die Schrittweite nicht aus der Stärke des Gradienten, sondern lediglich aus dessen Vorzeichen ermittelt. Stimmt das Vorzeichen zweier aufeinanderfolgenden Gradienten in zwei Iterationen überein, wird die Schrittweite erhöht. Zusätzlich wird die Schrittweite für jedes Gewicht einzeln angepasst. Es wird angemerkt, dass das Rprop-Verfahren nicht für kleine Batch-Größen geeignet ist.

Eine Übersicht über Verfahren zum gradientenbasierten Lernen ist in Goodfellow u. a. (2016b) zu finden.

Convolutional Layer

Der Einsatz von Faltungen (Convolutions) in neuronalen Netzen wurde erstmals in LeCun u. a. (1989) vorgeschlagen. Solche Netze werden auch als CNN (Convolutional Neural Networks) bezeichnet. Convolutions setzen drei Methoden zur Verbesserung der Performance eines Netzes ein: das Entfernen „unnützer“ Verbindungen (LeCun u. a., 1989), eine Teilung der Parameter sowie eine äquivariante Abbildung der Parameter (Goodfellow u. a., 2016c).

Ursprünglich sind die Neuronen zweier benachbarter Layer vollständig miteinander verbunden (fully connected Layer). Für einen convolutional Layer trifft dies nicht zu. Stattdessen werden mehrere, unterschiedliche Matrizen (Kernel) über die Eingangsdaten $\mathbf{x}^k$ des Layers geschoben, deren Größe geringer als die der Eingangsdaten ist. Die Elemente der Kernel ergeben die Parameter oder Gewichte des Layers. Ein Kernel K der Größe $l \times m$ führt eine Faltung über den von ihm jeweils abgedeckten Bereich der Eingangsdaten durch, indem er die jeweiligen Elemente gewichtet summiert. Die Faltung ergibt sich im 2D-Fall aus der folgenden Beziehung und führt zu einer Merkmalskarte (Feature Map) S :

$$S(i,j) = \sum_l \sum_m \mathbf{x}^k(i,j) K(i-l,j-m) \quad (2.35)$$

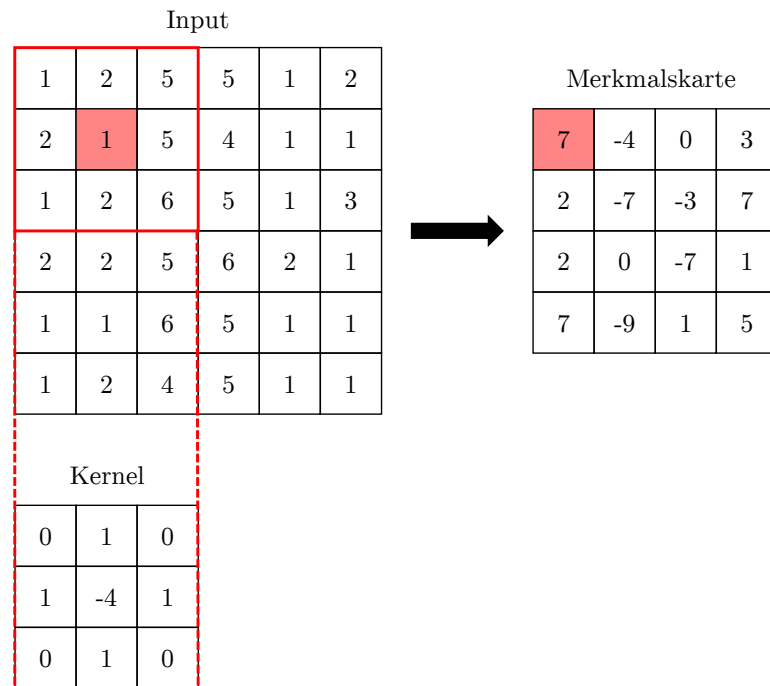


Abbildung 2.9.: Beispiel einer valid Convolution. Auf den Input mit einer Dimension von 6×6 Werten wird ein 3×3 -Kernel angewendet. Hieraus resultiert eine Merkmalskarte der Größe 4×4 .

Wird K auf die äußeren Werte der Eingangsdaten angewendet, so würde sich ohne das Hinzufügen künstlicher Werte an den Außenbereichen (Padding) das Bild in Abhängigkeit der Kernelgröße verkleinern. Eine solche Convolution wird auch als valid bezeichnet. Beträgt in 1D die Breite des Kernels l und die Länge der Eingangsdaten n , so verringert sich die Länge der Daten zu $n - l + 1$. Ein Beispiel einer valid Convolution durch einen einzigen Kernel ist in Abbildung 2.9 skizziert. Wird ein Padding so angebracht, dass die Dimension der Eingangsdaten erhalten bleibt, wird von einer same Convolution gesprochen. Hierbei werden die entsprechenden Padding-Werte in der Regel mit 0 aufgefüllt. Um zu vermeiden, dass die Werte in den Außenbereichen weniger Einfluss auf S haben, kann das Padding so weit erhöht werden, dass diese Werte ebenfalls l mal besucht werden (full Convolution) (Goodfellow u. a., 2016c).

Werden mehrere convolutional Layer hintereinander angewendet, so wird indirekt der Interaktionsbereich der einzelnen Neuronen erhöht. In 2D kann ein einzelner convolutional Layer beispielsweise Kanten in einem Bild detektieren. Ein folgender convolutional Layer kann wiederum genutzt werden, um die gefundenen Kanten miteinander zu verbinden.

Für jeden Bereich im Bild werden dieselben Kernel mit denselben Gewichten verwendet, woraus sich die zweite Eigenschaft, die Teilung der Parameter, ergibt. Die genannten Eigenschaften verringern den Speicherbedarf des Netzes und erhöhen zugleich die Effizienz (Goodfellow u. a., 2016c).

Die Teilung der Parameter bewirkt weiterhin, dass die einzelnen Layer äquivariant gegenüber Translationen sind. Dies bedeutet, dass bestimmte Formen unabhängig von ihrer Position in den Eingangsdaten erkannt werden können. Soll beispielsweise ein Fahrzeug in einem 2D-Bild detektiert werden, so wirken dieselben Kernel-Parameter auf jeden Bereich im Bild. Ist ein Fahrzeug im Bild enthalten, so kann es auf diese Art grundsätzlich positionsunabhängig detektiert werden (Goodfellow u. a., 2016c).

Im Anschluss an eine Convolution wird häufig ein sogenanntes **Pooling** durchgeführt. Hierbei wird die Größe der jeweiligen Merkmalskarten verringert, indem die Werte bestimmter, sich nicht überlappender Bereiche in den Merkmalskarten aggregiert werden. Die Aggregation kann beispielsweise durch die Verwendung des Maximalwertes des entsprechenden Bereiches geschehen (Max Pooling) (Goodfellow u. a., 2016c). Das Pooling soll helfen, das neuronale Netz weniger empfindlich gegenüber Translationen zu machen, sodass sich das Ergebnis des Layers bei geringen Translationen nicht verändert, was wiederum die Äquivarianz des Netzes erhöht. Weiterhin reduziert sich der Speicherbedarf durch die Verringerung der Dimensionen des Outputs (Goodfellow u. a., 2016c).

Autoencoder

Autoencoder zeichnen sich dadurch aus, dass ihr Output möglichst den Input abbilden soll. Sie bestehen aus einem Encoder und einem Decoder. Der Decoder versucht, aus dem Output des Encoders die Eingangsdaten des gesamten Netzes zu rekonstruieren. Hierdurch soll erzielt werden, dass das Netz Merkmale lernt, die den Input möglichst gut repräsentieren. Um dies zu gewährleisten und zu verhindern, dass der Input nur kopiert wird, werden verschiedene Restriktionen in das Netz implementiert. Ein häufig verwendetes Mittel ist die Reduktion der Layer-Dimensionen im Encoder. In diesem Fall wird von einem **Undercomplete Autoencoder** gesprochen. Hierbei ist darauf zu achten, dass die Kapazität des Autoencoders, welche sich unter anderem aus der Anzahl der Parameter und Layer ergibt, nicht zu groß ist. Dies hätte zur Folge, dass das Netz nicht in der Lage ist, relevante Merkmale zu bestimmen, welche den Input beschreiben (Goodfellow u. a., 2016d).

Regularized Autoencoder wählen die Komplexität und Dimension des Netzes entsprechend der Komplexität der zu modellierenden Verteilung. Dies geschieht durch eine Veränderung der Kostenfunktion, die verschiedene Kriterien erfüllt. Beispiele hierfür sind die Resistenz gegenüber Rauschen oder fehlenden Eingangsdaten, möglichst geringen Gradienten und möglichst dünn (sparse) besetzte Layer (Goodfellow u. a., 2016d).

Sparsity ist gegeben, wenn eine Vielzahl der Parameter nahe Null ist und somit einen sehr geringen Einfluss auf die Ergebnisse hat. Beinhaltet ein Autoencoder einen entsprechenden Term in der Fehlerfunktion, so wird auch von einem **Sparse Autoencoder** gesprochen. Sparse Autoencoder werden typischerweise verwendet, um Merkmale für weiterreichende Aufgaben, wie die Klassifizierung, zu bestimmen (Goodfellow u. a., 2016d).

Wird auf den Input $\mathbf{x}$ ein Rauschen angebracht, woraus sich $\tilde{\mathbf{x}}$ ergibt, und wird anschließend der Output $\tilde{\mathbf{x}}$ verglichen, so wird dieses Netz als **Denoising Autoencoder** bezeichnet. Ebenfalls ist es möglich, $\tilde{\mathbf{x}}$ als Input und in der Kostenfunktion $\mathbf{x}$ zu verwenden. Da hierbei die Kostenfunktion nicht reguliert wird, handelt es sich in diesem Fall folglich um keinen Regularized Autoencoder (Goodfellow u. a., 2016d).

Hilfsmittel und Werkzeuge

Zur Verwendung neuronaler Netze ist eine Vielzahl an Programmen und Programmbibliotheken verfügbar. Zu nennen sind hier beispielsweise das Microsoft Cognitive Toolkit (Microsoft, 2018) für die Programmiersprachen Python, C++ und BrainScript, das von Berkeley AI Research entwickelte Caffe (Jia u. a., 2014) für C++, die MATLAB Neural Network Toolbox (MathWorks, 2018) sowie die in der vorliegenden Arbeit verwendete Open-Source-Programmbibliothek TensorFlow (Abadi u. a., 2015).

TensorFlow ist eine Weiterentwicklung des von Google entwickelten Systems DistBelief. Die verschiedenen Rechenoperationen wurden in C++ implementiert. Verwenden lässt sich TensorFlow neben C++ jedoch auch mit Python und MATLAB. Im Vergleich zu DistBelief stehen dem Nutzer in TensorFlow eine Vielzahl mathematischer Operatoren zur Verfügung, wodurch die Programm-bibliothek flexibler anwendbar ist. In TensorFlow wird ein neuronales Netz durch einen einzigen Datenfluss-Graphen umgesetzt. Dieser Graph besteht aus den Eingangsdaten, den mathematischen Operatoren, den Parametern und entsprechenden Regeln, wie diese zu aktualisieren sind sowie gegebenenfalls einer Vorverarbeitung des Inputs. Die Daten, die zwischen den jeweiligen Elementen des Graphs fließen, werden als Tensors bezeichnet. Ein Tensor ist ein n -dimensionales Array eines bestimmten Datentyps. Neben der hohen Variabilität wird als weiterer großer Vorteil der Bibliothek beschrieben, dass sich TensorFlow leicht parallelisieren und somit auch für sehr große Datenmengen effizient einsetzen lässt (Abadi u. a., 2016).

2.3. Filterverfahren zur Lokalisierung

Stehen verschiedene Messungen eines Fahrzeuges zur Eigenlokalisierung zur Verfügung, so bedarf es geeigneter Verfahren, die diese kombinieren und hieraus eine genaue und verlässliche Position bestimmen können. In der vorliegenden Arbeit sollen globale Positionen, welche aus LiDAR-Messungen bestimmt werden, mit Messungen einer Inertialeinheit kombiniert werden. Üblicherweise werden für vergleichbare Probleme Bayessche Filter angewendet. Eine verbreitete Umsetzung eines Bayesschen Filters ist das Kalman-Filter, welches den aktuellen Zustand eines Systems durch eine Normalverteilung beschreibt. In dieser Arbeit wurde ein Kalman-Filter zur Beschreibung nicht-linearer Probleme, das Extended Kalman-Filter (EKF) verwendet. Weitere zu nennende Verfahren sind das Histogramm-Filter sowie das Partikelfilter, welche den Zustand mithilfe einer endlichen Anzahl von Werten im Zustandsraum beschreiben (Thrun u. a., 2005a).

2.3.1. Bayessche Filter

Eine für das Bayessche Filter grundlegende Formel ist das Bayes-Theorem:

$$p(x|z) = \frac{p(z|x) p(x)}{p(z)} \quad (2.36)$$

Es gibt die bedingte Wahrscheinlichkeit $p(x|z)$ eines Zustands x unter der Bedingung z an. Diese berechnet sich aus dem Produkt der A-Priori-Wahrscheinlichkeitsverteilung (Prior) $p(x)$ und der bedingten Wahrscheinlichkeit $p(z|x)$, geteilt durch den Normalisierungsfaktor $p(z)$. Nach dem Gesetz der totalen Wahrscheinlichkeit lässt sich $p(z)$ aus

$$p(z) = \int p(z|x) p(x) dx \quad (2.37)$$

bestimmen. Ist x von mehreren Variablen, z und u , abhängig, so lässt sich das Bayes-Theorem zu

$$p(x|z,u) = \frac{p(z|x,u) p(x|u)}{p(z|u)} \quad (2.38)$$

umformen (Thrun u. a., 2005b).

Das Ziel eines Bayesschen Filters ist es nun, den Zustand x_t zum Zeitpunkt t zu bestimmen. Dieser wird über die bedingte Wahrscheinlichkeit $p(x_t|z_{1:t}, u_{1:t})$ angegeben. Hierbei bezeichnet $z_{1:t}$ alle vergangenen Messungen und $u_{1:t}$ alle vergangenen Bewegungen (Controls) des Systems. Die

bedingte Wahrscheinlichkeit wird auch als Belief bezeichnet und soll in dieser Arbeit als $bel(x_t)$ gekennzeichnet werden. Somit gilt

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t}) \quad (2.39)$$

Die Bestimmung von $bel(x_t)$ nach einer Messung z_t wird auch als Korrektur oder Update bezeichnet. Wird der Belief $\overline{bel}(x_t)$ vor der Messung gesucht, so kann dieser aus

$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t}) \quad (2.40)$$

bestimmt werden. Dieser Schritt wird als Prädiktion bezeichnet. In einem Bayesschen Filter werden nun Prädiktion und Korrektur fortlaufend durchgeführt. Durch eine Prädiktion anhand von Control-Daten u_t , wie beispielsweise die Stärke der Betätigung des Gaspedals im Fahrzeug, wird der Belief $\overline{bel}(x_t)$ zum Zeitpunkt t prädiziert. Wird eine Messung z_t durchgeführt, so lässt sich anschließend $bel(x_t)$ bestimmen und der jeweilige Zustand des Fahrzeuges somit korrigieren (Thrun u. a., 2005b).

2.3.2. Kalman-Filter

Das Kalman-Filter ist eine Implementierung eines Bayesschen Filters, welches verwendet werden kann, um den Zustand eines linearen, normalverteilten Systems zu filtern und zu prädizieren (Thrun u. a., 2005c). Der Belief zu einem Zeitpunkt t wird durch die Parameter einer mehrdimensionalen Normalverteilung, dem Vektor der Erwartungswerte μ_t und der Kovarianzmatrix Σ_t bestimmt. Die Bewegungsgleichung, welche den Zustandsvektor x_t zum Zeitpunkt t prädiziert, ist durch eine lineare Funktion der Form

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \quad (2.41)$$

bestimmt. Beträgt die Dimensionen des Zustandsvektors x n und die des Control-Vektors u m , so ist A durch eine Matrix der Dimension $n \times n$ und B durch eine Matrix der Dimension $n \times m$ definiert. A wird auch als Übergangsmatrix vom Zeitpunkt $t - 1$ zum Zeitpunkt t bezeichnet und beschreibt die Dynamik des Systems. B beschreibt die Dynamik des Control-Modells, welches auf u wirkt. Durch den Vektor ϵ_t wird zusätzlich ein normalverteiltes Rauschen angebracht. Die Dimension von ϵ_t beträgt n , der Mittelwert ist Null. Die Kovarianzmatrix von ϵ_t wird durch R_t bestimmt (Kalman, 1960).

Der prädizierte Zustandsvektor $\overline{\mu}_t$ und die entsprechende Kovarianzmatrix $\overline{\Sigma}_t$ können aus den Gleichungen 2.42 und 2.43 bestimmt werden:

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \quad (2.42)$$

$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \quad (2.43)$$

Wird eine Messung z_t durchgeführt, so kann diese durch

$$z_t = C_t x_t + \delta_t \quad (2.44)$$

beschrieben werden, wobei C_t das Messmodell beschreibt. Der Vektor δ_t gibt Auskunft über die Unsicherheit der Messung, welche durch eine Normalverteilung mit dem Mittelwert Null und einer Kovarianz Q_t beschrieben wird. Zur Korrektur des Zustandes wird das sogenannte Kalman-Gain K eingeführt. Es beschreibt, wie groß der Einfluss einer Messung auf die Bestimmung des Zustandes ist und lässt sich nach Gleichung 2.45 bestimmen. Aus den Gleichungen 2.46 und 2.47 ergibt sich letztendlich $bel(x_t)$, welcher durch μ_t und Σ_t beschrieben wird. Die Differenz zwischen der

tatsächlichen Messung z_t und der erwarteten Messung $C_t \cdot \bar{\mu}_t$ wird auch Innovation genannt (Bishop und Welch, 2001).

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \quad (2.45)$$

$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \quad (2.46)$$

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \quad (2.47)$$

Eine ausführliche Herleitung der Gleichungen 2.42-2.47 findet sich in Kalman (1960) und Thrun u. a. (2005c).

2.3.3. Extended Kalman-Filter

Das in Abschnitt 2.3.2 vorgestellte Kalman-Filter ist nur für Systeme anwendbar, deren Änderungen der Zustände sich durch einen linearen Zusammenhang beschreiben lassen. Liegt ein nicht-lineares Problem vor, mit einer nicht-linearen Zustandsübergangsfunktion $g(u_t, x_{t-1})$ und einer nicht-linearen Messfunktion $h(x_t)$, so muss das Filter erweitert werden. Eine solche Erweiterung ist das Extended Kalman-Filter (EKF). Hierbei werden A_t und B_t durch die Jacobi-Matrix G_t ersetzt, C_t durch die Jacobi-Matrix H_t (Thrun u. a., 2005c).

Die Linearisierung nicht-linearer Funktionen geschieht im EKF durch eine Taylorentwicklung. Hierbei wird die Funktion g an x_{t-1} und u_t approximiert, indem die momentane Steigung der Kurve aus der Ableitung g' bestimmt wird:

$$g'(u_t, x_{t-1}) := \frac{\delta g(u_t, x_{t-1})}{\delta(x_{t-1}, u_t)} \quad (2.48)$$

Aus $g(u_t, x_{t-1})$ und $\mu_t - 1$ lässt sich nun $g(u_t, x_{t-1})$ durch

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}) \quad (2.49)$$

bestimmen, mit der entsprechenden Jacobi-Matrix G_t , welche die partiellen Ableitungen von $g(u_t, \mu_{t-1})$ mit dem aktuellen geschätzten Mittelwert μ_{t-1} nach dem vorherigen Zustandsvektor x_{t-1} enthält. Die Messfunktion $h(x_t)$ wird ebenfalls approximiert, indem die Steigung an $\bar{\mu}_t$ bestimmt wird:

$$h(u_t, x_t) := h(\bar{\mu}_t) + H_t(x_{t-1} - \mu_{t-1}) \quad (2.50)$$

H_t beinhaltet hierbei die partiellen Ableitungen der Messfunktion nach x_{t-1} .

Nach der Linearisierung des Systems ändert sich der im linearen Fall durch die Gleichungen 2.42 und 2.43 sowie 2.45 bis 2.47 beschriebene Ablauf des Kalman-Filters zu:

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) \quad (2.51)$$

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \quad (2.52)$$

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \quad (2.53)$$

$$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \quad (2.54)$$

$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \quad (2.55)$$

Zur Anwendung des Filters ist es weiterhin notwendig, ein Bewegungsmodell für das Fahrzeug zu definieren. Beinhaltet x die 2D-Position und Ausrichtung des Fahrzeuges (p_x, p_y, θ) , so kann die Bewegung beispielsweise durch die Geschwindigkeit v und Drehrate ω beschrieben werden. Das Bewegungsmodell zur Bestimmung der Pose $x_t = (p'_x, p'_y, \theta')^T$ zum Zeitpunkt t lässt sich wie in

Thrun u. a. (2005a) gezeigt aus der vergangenen Pose $x_{t-1} = (p_x, p_y, \theta)^T$ und dem Control-Vektor $u_t = (v, \omega)^T$ herleiten. Zusätzlich wird mit $\mathcal{N}(0, R_t)$ ein um 0 normalverteiltes Rauschen addiert, wodurch sich in diesem Fall das folgende Bewegungsmodell ergibt:

$$\begin{pmatrix} p'_x \\ p'_y \\ \theta' \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix} + \mathcal{N}(0, R_t). \quad (2.56)$$

3. Stand der Forschung

Dieses Kapitel soll einen Überblick über den aktuellen Stand der Forschung geben. Abschnitt 3.1 widmet sich der Detektion dynamischer Objekte in LiDAR-Daten, wobei hier nach den Methoden der Änderungsdetektion und der Klassifizierung unterschieden wird. Abschnitt 3.2 beinhaltet eine Übersicht über Methoden zur Fahrzeuglokalisierung im Kontext des autonomen Fahrens.

3.1. Detektion dynamischer Objekte

Werden Punktwolken zur Lokalisierung verwendet, so ergibt sich die Position in der Regel aus den Punktkoordinaten der im Straßenraum vorhandenen Objekte. Hierfür ist es wichtig, dass die Objekte statisch sind. In dieser Arbeit soll für statische Objekte für den Zeitraum zwischen der Erstellung einer Karte und dem Zeitpunkt der Positionsbestimmung gelten, dass sich weder die Form, noch die Position und Orientierung ändert.

Demgegenüber stehen dynamische Objekte, deren Form, Position und Orientierung veränderbar sind. Beispiele für dynamische Objekte sind Fahrzeuge und Personen. Auch Vegetation kann somit als dynamisch angesehen werden. Ebenfalls kann sich das Erscheinungsbild von Häuserfassaden ändern, indem beispielsweise eine Markise ausgefahren wird. Bei Vegetation und großen Objekten wie Häuserfassaden sollte jedoch die Anzahl der sich ändernden Punkte in Relation zu der Gesamtpunktzahl der zum Objekt gehörenden Punkte berücksichtigt werden.

In den folgenden Abschnitten werden verschiedene Methoden zur Detektion dynamischer Objekte in LiDAR-Daten vorgestellt.

3.1.1. Änderungsdetektion in Punktwolken

Liegen für ein bestimmtes Messgebiet mehrere Datensätze mit unterschiedlichen Messzeitpunkten vor, so kann eine Änderungsdetektion in den jeweiligen Punktwolken durchgeführt werden. Die Änderungsdetektion wird häufig für Airborne-Laserscanning verwendet, um großflächige Veränderungen, beispielsweise von Gebäuden, zu detektieren (Vu u. a., 2004, Champion u. a., 2009). In diesem Fall soll sich aber auf die Änderungsdetektion in LiDAR-Daten bodengebundener Systeme konzentriert werden, um so dynamische Objekte zu detektieren.

Eine der ersten Arbeiten zu Änderungsdetektion in terrestrischen Punktwolken wird in Girardeau-Montaut u. a. (2005) präsentiert. Zunächst werden hier mehrere Punktwolken mithilfe eines ICP-Algorithmus registriert. Anschließend werden die Daten in jeweils einen Octree pro Punktwolke überführt. Eine Baumstruktur sichert hierbei geringe Zugriffszeiten von $O(\log n)$ auf die jeweiligen Voxel. Die Wurzel der jeweiligen Baumstruktur ist stets dieselbe, sodass die Voxelkoordinaten in beiden Punktwolken ebenfalls übereinstimmen. Im letzten Schritt werden verschiedene Distanzmetriken verwendet, um die Punkte in den jeweiligen Voxeln mit den nächstliegenden Nachbarspunkten in den korrespondierenden Voxel in den weiteren Datensätzen zu vergleichen: eine mittlere Distanz für alle Punkte innerhalb eines Voxels, der Schnittwinkel der Ebenen, die in die jeweilige Voxel geschätzt wurden sowie die Hausdorff-Distanz. Es zeigt sich, dass die Hausdorff-Distanz die schlechteste und das Schätzen von lokalen Ebenen die besten Werte für die Laufzeit aufweisen. Außerdem werden Distanzbilder ausgehend von den jeweiligen Laserscanner-Standpunkten verwendet, um Bereiche zu detektieren und aus der Berechnung auszuschließen, welche während der Messung verdeckt waren

und somit nicht vom Laserscanner erfasst werden konnten.

In Aijazi u. a. (2013) werden zunächst dynamische Objekte in Mobile-Mapping-Daten verschiedener Messepochen durch eine Änderungsdetektion detektiert und aus den Daten entfernt. Anschließend werden alle übrigen statischen Objekte in eine gemeinsame 3D-Belegungskarte gespeichert, auf dessen Basis für jedes Voxel Ähnlichkeitswerte in jeder Epoche bestimmt werden, was ein sukzessives Update einer 3D-Karte ermöglicht.

In Schachtschneider u. a. (2017) und Gehring u. a. (2017) wird eine 3D-Belegungskarte verwendet, in der jedes Voxel einen Konfidenzwert für den Zustand *frei* oder *belegt* beinhaltet. Für den Fall, dass ein Pixel nicht besucht wurde, gilt dieses als *unbekannt*. Dynamische Objekte treten auf, wenn sich der Status eines Voxels von *frei* zu *belegt* oder andersherum ändert.

Xiao u. a. (2015) erstellen statt eines voxelbasierten Ansatzes eine Belegungskarte unter Betrachtung der einzelnen Scanstrahlen eines LiDAR-Mobile-Mapping-Systems. Jeder Scanstrahl von der jeweiligen Sensorposition zum gemessenen Punkt wird durch einen Zylinder repräsentiert. Der Bereich zwischen einem Punkt und der Sensorposition wird als *frei* markiert, der Bereich hinter dem Punkt als *unbekannt*. Als *belegt* wird der Bereich um den Punkt entlang einer durch den Normalenvektor des Punktes definierten lokalen Ebene markiert.

Zeibak und Filin (2008) wählen eine Präsentation der Daten in Distanzbilder, um Änderungen zu detektieren. Lücken in den Daten, wie beispielsweise an Fenstern, werden geschlossen, indem für diese Regionen deren Nachbapixel betrachtet werden und ein Dilatationsprozess ausgeführt wird.

Eine Echtzeit-Detektion von Fahrzeugen wird in Petrovskaya und Thrun (2009) vorgeschlagen. Hierbei werden die Fahrzeuggeometrie und -Dynamik in einem Bayes-Filter modelliert. In einem Bayesschen Filter wird anschließend die Position und Bewegung der detektierten Fahrzeuge bestimmt.

3.1.2. Klassifizierung dynamischer Objekte

Eine weitere Möglichkeit, dynamische Objekte zu erkennen, ist eine Klassifizierung durch Methoden des maschinellen Lernens. In Schlichting und Brenner (2016a) wird ein Random Forest verwendet, um Fußgänger und Fahrradfahrer in 3D-Laserscanning-Daten zu detektieren. Nach einer Segmentierung einzelner Objekte werden deren Bounding Box, also die Höhe, Breite und Tiefe, die Kovarianzmatrix und ein 2D-Histogramm der aus einer Hauptkomponentenanalyse bestimmten, zueinander orthogonalen Hauptebenen als Merkmale in der Klassifizierung verwendet. Außerdem werden sogenannte Slice-Merkmale, welche in Kidono u. a. (2011) zur Fußgängerdetektion vorgestellt werden, in den Merkmalsvektor aufgenommen. Als Klassifikator wird in Kidono u. a. (2011) eine Support Vector Machine (SVM) verwendet. Ähnliche Arbeiten zur Fußgängerdetektion finden sich in Navarro-Serment u. a. (2010) und Spinello u. a. (2010). In Spinello u. a. (2010) werden die jeweiligen Objekte zusätzlich in vertikale Zellen aufgeteilt, für welche anschließend einzelne Merkmale bestimmt werden.

Arras u. a. (2007) und Premevida u. a. (2009) verwenden Automotive-Laserscanner mit nur einer horizontalen Scanzeile, um Fußgänger zu detektieren, indem in den einzelnen Scans nach Strukturen gesucht wird, die den Beinen eines menschlichen Körpers entsprechen.

Eine Fahrzeugdetektion ist in Bock u. a. (2015) zu finden. Hierbei wird ein Random Forest genutzt, welcher als Eingangsdaten verschiedene Merkmale, wie die Bounding Box, die Anzahl der Punkte sowie die Ergebnisse einer Hauptkomponentenanalyse, nutzt. Asvadi u. a. (2017) nutzen

das Tiefenbild eines 3D-Laserscanners, um mithilfe eines CNN Fahrzeuge zu detektieren.

Wie eingangs in diesem Kapitel beschrieben, kann auch Vegetation als dynamisch angesehen werden. Pu und Vosselmann (2006) detektieren Bäume, indem getrennt nach stangenförmigen Objekten, wie den Baumstämmen, und den Baumkronen gesucht wird. Wu u. a. (2013) verwenden eine voxelbasierte Region-Growing-Methode, um Bäume zu identifizieren.

Statt in der Punktwolke nach einzelnen, spezifischen Kategorien zu suchen, findet in Golovinskiy u. a. (2009) und Douillard u. a. (2014) eine Gesamtklassifizierung der Szene statt. Im Anschluss an eine Segmentierung werden in Golovinskiy u. a. (2009) für aus der Punktwolke segmentierte Objekte verschiedene Merkmale, welche das Aussehen und die Position des Objektes in dessen Nachbarschaft beschreiben, bestimmt. Zur Prädiktion der Klassen werden verschiedene Klassifikatoren verwendet. In Douillard u. a. (2014) geschieht die Klassifizierung anhand eines Vergleiches der jeweiligen Objekte mit Templates. Hierzu werden die Punktdistanzen der entsprechenden Punktwolken der untersuchten Objekten zu den Templates durch einen ICP-Algorithmus minimiert. Die Ähnlichkeit wird anschließend durch eine Fehlermetrik bestimmt, welche sich aus der Summe der euklidischen Distanzen korrespondierender Punkte ergibt.

In den bisher genannten Arbeiten werden zumeist geometrische Eigenschaften verwendet, um dynamische Objekte zu detektieren. In Steinhauser u. a. (2008) und Dewan u. a. (2016) wird stattdessen eine mögliche Bewegung der Objekte während der Messungen untersucht. Steinhauser u. a. (2008) ordnen in mehreren 3D-Scans segmentierte Objekte anhand von Merkmalspunkten einander zu. Ist die Veränderung der Position der zum Objekt zugehörigen Punkte nun größer als ein gewisser Grenzwert, so wird von einem dynamischen Objekt ausgegangen. Dewan u. a. (2016) verwenden ebenfalls einen 3D-Laserscanner und Merkmalspunkte. Zusätzlich wird die Position der Objekte über ein Bewegungsmodell beschrieben. Mithilfe eines Bayesschen Filters wird untersucht, ob die Punkte der jeweiligen Objekte dem Bewegungsmodell folgen.

Statt in der Punktwolke, können dynamische Objekte auch im Distanzbild des Laserscanners detektiert werden. In Premebida u. a. (2014) und Schlosser u. a. (2016) werden hierfür LiDAR- und Kameradaten zur Fußgängerdetektion kombiniert. Premebida u. a. (2014) verwenden einen Sliding-Window-Ansatz sowie einen HOG-Detektor (Histogram of Oriented Gradients) (Dalal und Triggs, 2005) um Merkmale aus Tiefen- und RGB-Bildern zu extrahieren. Diese Merkmale gehen in eine SVM ein, welche den entsprechenden Bereich im Bild klassifiziert. Schlosser u. a. (2016) verwenden verschiedene CNN, deren Eingangsdaten aus RGB- und HHA-Kanälen bestehen. HHA steht für die drei Eigenschaften horizontale Disparität, Höhe über Grund und Winkel (Angle), siehe Gupta u. a. (2014). In Li u. a. (2016a) wird ebenfalls ein CNN zur Detektion von Fahrzeugen in Distanzbildern verwendet. Da die Suche nach dynamischen Objekten in Distanzbildern eng verwandt mit der Suche in Kamerabildern ist, sei an dieser Stelle auch auf entsprechende Methoden der semantischen Klassifizierung in RGB- und Grauwertbildern verwiesen. Einen Überblick über aktuelle Arbeiten bietet der Benchmark des Cityscape-Datensatzes (Cordts u. a., 2016), welcher verschiedene Arbeiten zur Bildklassifizierung geordnet nach der entsprechenden Performance für gegebene Testdaten auflistet.

3.2. Fahrzeuglokalisierung

Die Fahrzeuglokalisierung soll an dieser Stelle auf Verfahren zur Feinlokalisierung mit einer Genauigkeit von wenigen Dezimetern sowie der globalen Positionsbestimmung mit einer geforderten Genauigkeit von wenigen Metern unterteilt werden.

3.2.1. Feinlokalisierung

Eine der ersten Arbeiten zur Fahrzeuglokalisierung findet sich in Pomerleau und Jochem (1996). Aus Grauwert-Bildern einer Kamera werden die Krümmung der Fahrbahn berechnet sowie Fahrbahnmarkierungen detektiert. Anschließend kann die Position des Fahrzeuges auf der Fahrbahn bestimmt werden. Spätestens mit der DARPA Grand Challenge 2004 und der DARPA Urban Challenge 2007 (DARPA, 2018) ist die Positionsbestimmung selbstfahrender Fahrzeuge in den breiten Fokus der Forschung gerückt. Hierbei werden weiterhin häufig Fahrbahnmarkierungen zur Bestimmung der Position des Fahrzeuges verwendet. Suhr u. a. (2017) gleichen zur Lokalisierung detektierte Fahrbahnmarkierungen mit einer Referenzkarte ab und kombinieren diese Daten mit Messungen einer Inertialen Messeinheit (IMU) und eines GPS-Sensors in einem Partikelfilter. Der hierdurch resultierende RMSE liegt noch bei 0,94 m. Ziegler u. a. (2014) verwenden zusätzlich in Bildern detektierte Merkmalspunkte mit bekannten 3D-Koordinaten, wodurch die Genauigkeit auf 0,10 m verbessert werden kann. Die Merkmalspunkte werden durch einen beleuchtungsunabhängigen Deskriptor (DIRD) beschrieben, welcher in Lategahn u. a. (2014) vorgestellt wird. Hierdurch können durch unterschiedliche Beleuchtungsstärken hervorgerufene Fehler zwar behoben werden, gänzlich unabhängig von der Beleuchtung ist dieser Ansatz dennoch nicht. Weiterhin bestimmen auch die Fahrzeugposition und -orientierung die Anzahl der mit einer Referenz übereinstimmenden Merkmalspunkte. Statt in Kamerabildern werden in Hata und Wolf (2016) und Nothdurft u. a. (2011) Fahrbahnmarkierungen aus LiDAR-Daten extrahiert, indem die Intensitätswerte der Messungen analysiert werden. Zur fortlaufenden Positionsbestimmung wird ebenfalls ein Filteransatz verwendet, in welchen auch Daten eines GPS-Sensors und einer IMU eingehen, wodurch die Position auf weniger als 0,3 m genau bestimmt werden kann. In Hata und Wolf (2016) werden neben den Fahrbahnmarkierungen auch Bordsteine aus einem 3D-Laserscanner extrahiert. Dies geschieht durch eine Analyse der einzelnen ringförmigen Layer des Laserscanners. Die in Nothdurft u. a. (2011) präsentierte Methode wird in Choi und Maurer (2015) um eine 3D-Belegungskarte erweitert.

Wei u. a. (2013) präsentieren einen Ansatz, in dem ein Abgleich der Messungen eines Laserscanners mit den Daten eines Geoinformationssystems (GIS) erfolgt. Durch eine Hough-Transformation werden horizontale und vertikale Linien in einem 2D-Laserscan gefunden und akkumuliert. Die Bestimmung der Position erfolgt anschließend über einen Vergleich der Linien mit den in einem GIS gespeicherten Gebäudegrundrissen in Kombination mit einem GPS-System. Hierdurch kann eine Genauigkeit von 1,3 m erzielt werden.

In Qu u. a. (2015) werden georeferenzierte Straßenschilder in Bildsequenzen mithilfe von SIFT-Merkmalen detektiert. Über eine lokale Bündelblockausgleichung findet eine Ausgleichung der Kamerastandpunkte statt, in welche die detektierten und zu einer Referenz zugeordneten Schilder als Kontrollpunkte eingehen. Eine Initialposition wird von einem Low-Cost-GPS-System bereitgestellt. Das Verfahren wird in Qu u. a. (2018) um georeferenzierte Landmarken erweitert, welche im gesamten Straßenraum auftreten. Hierdurch wird eine auf 0,12 m genaue Lokalisierung ermöglicht.

Spangenberg u. a. (2016) detektieren stangenförmige Objekte in Stereo-Bildern. Zur Lokalisierung werden diese Stangen mit einer Referenzkarte abgeglichen und zusammen mit GPS- und Fahrzeugodometrie-Daten in einem Partikelfilter verarbeitet, welcher zusätzlich durch ein Kalman-Filter gestützt wird. Für den RMSE ergibt sich hierbei ein Wert von 0,14 m. In Brenner (2009a) wird ein Ansatz zur Segmentierung von stangenförmigen Objekten aus 3D-Punktwolken vorgestellt. Dieser Ansatz wird auch in dieser Arbeit angewendet. Eine ausführliche Beschreibung des Algorithmus findet sich in Abschnitt 2.1.6.

In Rohde u. a. (2016) werden die 3D-Daten eines Laserscanners zunächst auf eine durch die x - und y -Achse aufgespannte 2D-Ebene projiziert. Über eine Fourier-Mellin-Transformation wird die Transformation der aktuellen Messung zu einer Karte und hieraus die globale Position auf 0,12 m genau bestimmt. Weiterhin findet eine Bestimmung der Bewegung des Fahrzeuges statt, welche sich aus dem Abgleich aufeinanderfolgender Scans ergibt. Aus den Residuen der Pose wird zusätzlich Auskunft über die Integrität der Ergebnisse gegeben. In Yoneda u. a. (2014) wird die gesamte Punktwolke eines 3D-Scanners verwendet, um über einen Abgleich mit einer Referenzpunktwolke die aktuelle Position zu bestimmen. Dieser Ansatz hat allerdings zum Nachteil, dass die Referenzkarte hierbei große Datenmengen beinhaltet. Die Genauigkeit des Verfahrens liegt bei 0,20 m.

Levinson u. a. (2007) verwenden aus LiDAR-Daten erstellte 2D-Orthobilder, welche die Intensitätswerte des Bodens darstellen. Anschließend findet eine Korrelation der Bilder zu einer Referenzkarte statt. Die Position des Fahrzeuges wird hierauf folgend durch einen Partikelfilter mit einer Genauigkeit von unter 0,10 m bestimmt. Wolcott und Eustice (2014) verwenden ebenfalls die Intensitätswerte eines Laserscanners zur Erstellung einer 2D-Referenzkarten. Die Lokalisierung des Fahrzeuges erfolgt allerdings unter Verwendung herkömmlicher Kameras, mit einer Genauigkeit von 0,24 m. In Kim u. a. (2017) werden neben den Intensitätswerten auch die lokalen Höhen der jeweiligen 2D-Zellen verwendet, wodurch der RMSE auf 0,10 m sinkt. Maddern u. a. (2015) nutzen ebenfalls 2D-Höhen- und Intensitätsdarstellungen, gleichen die aktuelle Messung aber mit mehreren Referenzmessungen aus der Vergangenheit ab, um so aus dynamischen Objekten resultierende Fehler zu minimieren. Insgesamt führt dieses Verfahren dennoch zu einer niedrigeren Genauigkeit von 0,3–0,5 m.

3.2.2. Globale Positionsbestimmung

In Li u. a. (2016b) wird durch die Projektion von 3D-Punkten auf zur Straßenebene senkrechte Flächen eine rasterförmige Referenzkarte erstellt. Die Straßenebene wird hierbei durch einen RANSAC-Algorithmus bestimmt. Zur Lokalisierung werden die an der jeweiligen Position zuletzt zurückgelegten 50 m anhand der Normalized Information Distance (Vitányi u. a., 2009) mit einer Referenzdatenbank verglichen, ausgehend von einer initialen Position, welche durch ein Low-Cost-GPS-System bestimmt wird. Die Genauigkeit des Verfahrens liegt unter 1 m.

Eine globale Positionsbestimmung ohne Zuhilfenahme von GNSS-Sensoren ist eng verwandt mit dem Lost-Robot-Problem in der Robotik. Ohne Vorinformationen muss die Position in einem globalen Referenzsystem bestimmt werden. In Kendall u. a. (2015) und Kendall und Cipolla (2017) wird eine Relokalisierung auf zuvor erstellte Referenzbilder mit bekannter Position und Orientierung durch das Anlernen eines CNNs (*PoseNet*) durchgeführt. Die resultierenden Genauigkeiten im Außenbereich liegen hierbei im Meterbereich. Zeisl u. a. (2015) reduzieren die Laufzeit zur Bestimmung der Pose aus Bildmerkmalen durch die Verwendung einer Hough-Transformation. Durch zusätzliche Filteroperationen, welche die lokale Bildgeometrie untersuchen, liefert das Verfahren auch bei einer hohen Anzahl an falschen Korrespondenzen korrekte Ergebnisse.

In Brubaker u. a. (2016) ergibt sich die Position aus einem Abgleich der gefahrenen Trajektorie mit Daten des Kartendienstes OpenStreetMap (OSM). Die Bestimmung der Trajektorie geschieht in diesem Fall durch visuelle Odometrie, unter Verwendung von Kameradaten des Fahrzeuges. Die auf wenige Meter genaue Position in einem Bereich von 1 km^2 konnte durch diesen Ansatz in etwa einer Minute gefunden werden.

Bosse und Roberts (2007) führen eine Korrelation von Laserscans mit einer Referenz durch, um die Position eines Fahrzeuges in unstrukturierten Außenumgebungen zu bestimmen. In Tipaldi u. a. (2013) werden sogenannte geometrische FLIRT-Phrasen eingeführt, mit deren Hilfe 2D-Laserscans mit Referenzdaten verglichen werden. Der Vergleich der Daten und damit die Bestimmung der Position geschieht durch einen Bag-of-Words-Ansatz. Mit einer Genauigkeit von 85 % kann somit die Position auf 0,5 m genau bestimmt werden. Himstedt u. a. (2014) verwenden geometrische Landmarkenbeziehungen (GLARE), um 2D-Scans in eine von der Pose unabhängige Histogrammdarstellung zu transformieren. Über eine Nearest-Neighbor-Suche werden die Daten anschließend mit einer Referenz abgeglichen. Der Anteil an korrekten Positionsbestimmungen steigt durch die Verwendung der GLARE-Merkmale auf 93 % entlang einer 6,5 km langen Trajektorie.

4. Verwendete Sensoren und Daten

Dieses Kapitel gibt einen Überblick über die verschiedenen Sensorsysteme, die im Rahmen dieser Arbeit zum Einsatz kommen. Zur Erstellung der Referenzdaten der in den Kapiteln 6.1 und 6.3 vorgestellten Methoden wird ein Riegl VMX-250 Mobile-Mapping-System (Abschnitt 4.1) verwendet. Zur Online-Lokalisierung kommen in allen Methoden verschiedene Automotive-Sensoren (Abschnitt 4.2-4.4) zum Einsatz. An dieser Stelle ist anzumerken, dass sich die in den folgenden Abschnitten angegebenen Genauigkeiten stets auf die einfache Standardabweichung (1σ) beziehen. Ein Vergleich der hier vorgestellten Laserscanner ist abschließend in Abschnitt 4.5 zu finden.

4.1. Mobile-Mapping-System Riegl VMX-250

Das in dieser Arbeit verwendete Mobile-Mapping-System Riegl VMX-250 besteht aus zwei Laserscannern (Riegl VQ-250), einem Kamerasystem, einer GNSS- sowie Inertialeinheit, einem als Distance Measurement Indicator (DMI) bezeichneten Odometer und einer Rechen- und Speichereinheit (Control Unit). Abgesehen von dem am Rad befestigten DMI sind alle Sensoren über einen Gepäckträger auf dem Dach eines Volkswagen-T5-Kleinbusses angebracht, siehe Abbildung 4.1. Über einen mit der Control Unit verbundenen Bildschirm kann das System bedient werden.



Abbildung 4.1.: Auf dem Dach eines Volkswagen-Kleinbusses montiertes Mobile-Mapping-System Riegl VMX-250.

Die Riegl VQ-250-Laserscanner wenden das Prinzip des Impulslaufzeitverfahrens mit einer Full-Waveform-Analyse an. In dieser Arbeit wird jedoch stets der erste Impuls, also das erste vom Empfänger detektierte Signal, verwendet. Der Messbereich, der durch eine vollständige Umdrehung eines Scanners erfasst wird, beträgt 360° . Jeder Scanner misst 100 dieser Scanlinien pro Sekunde, die jeweils bis zu 3000 Punkte enthalten, sodass insgesamt bis zu 300 000 Punkte pro Sekunde aufgenommen werden können. Hieraus ergibt sich eine Winkelauflösung von $0,12^\circ$. Dies entspricht bei einer Entfernung von 10 m etwa 0,02 m. Die Distanz der Scanlinien in Fahrtrichtung

ist abhängig von der Geschwindigkeit des Messfahrzeuges. Bei einer Geschwindigkeit von 15 m/s beträgt die Distanz 0,15 m. Die minimale Messdistanz der Laserscanner beträgt 1,5 m, die maximale Messdistanz in Abhängigkeit der Reflektionseigenschaften der reflektierenden Oberfläche 75 - 200 m. Die absolute Genauigkeit der Distanzmessungen wird mit 10 mm, die Präzision, also die Wiederholgenauigkeit, mit 5 mm angegeben (RIEGL Laser Measurement Systems GmbH, 2012a).

Die Lokalisierung des Systems erfolgt durch das GNSS/Inertialsystem in Kombination mit dem DMI, wobei letzterer eingesetzt wird, um Drift zu verhindern, während das Fahrzeug steht. Die Genauigkeit des Roll- und Nick-Winkels wird mit $0,005^\circ$, die des Gierwinkels mit $0,015^\circ$ angegeben. In der Nachbearbeitung können zusätzlich Daten einer GNSS-Referenzstation verwendet werden, um die Genauigkeit der GNSS-Positionsbestimmung zu erhöhen. Der Hersteller gibt die resultierende absolute Genauigkeit mit 2 - 5 cm an. Erfahrungsgemäß sind diese Werte jedoch vor allem in urbanen Gebieten mit hohen Häuserschluchten aufgrund auftretender Effekte wie Abschattungen und Multi-Path nicht zu erreichen. Realistischer ist hier eine absolute Genauigkeit von 10 - 20 cm. In Hofmann (2017), in der auch die Genauigkeit des vorliegenden Mobile-Mapping-Systems untersucht wird, wird diese Annahme bekräftigt. Hier wird eine mittlere Abweichung der Mittelpunkte von 128 Masten der Straßenbeleuchtung von 12 cm zu den entsprechenden Referenzen festgestellt. Die Referenzmessungen wurden tachymetrisch mit einer Genauigkeit von < 2 cm durchgeführt. Die Untersuchung von Kaartinen u. a. (2012) zeigt jedoch, dass unter guten GNSS-Bedingungen das Riegl VMX-250 Genauigkeiten von bis zu 2,5 cm in der Lage und 3,5 cm in der Höhe erzielen kann (RIEGL Laser Measurement Systems GmbH, 2012a).

Die Daten des Riegl VMX-250 werden zusätzlich verwendet, um Messungen eines Automotive-Laserscanners zu simulieren. Hierbei wird angenommen, dass ein Laserscanner, welcher mit 25 Hz einzelne Scanlinien mit jeweils 1160 Punkte misst, vertikal am Heck des Fahrzeuges angebracht wurde. Es wird außerdem angenommen, dass sich das Fahrzeug mit 15 m/s fortbewegt, sodass die Distanz zwischen zwei Scanlinien 0,3 m beträgt. Der Sichtbereich wird zu 290° mit einer Winkelauflösung von $0,5^\circ$ festgelegt.

4.2. Valeo Scala

Der Automotive-Laserscanner Valeo Scala (Abbildung 4.2) wird von Valeo als erster in der Massenproduktion hergestellte Laserscanner bezeichnet (Valeo, 2017) und geht aus dem Multilayer-Laserscanner Ibeo Lux hervor (Lages u. a., 2013).

Der Laserscanner misst vier vertikale Ebenen im Abstand von jeweils $0,8^\circ$. Die horizontale Auflösung des Sensors beträgt $0,25^\circ$, der Sichtbereich 145° . Die Messfrequenz des Scanners beträgt 25 Hz, der Messbereich reicht von 0,3 – 150 m (Lages u. a., 2013). Die Distanzgenauigkeit des Valeo Scala ist nicht bekannt, weswegen hier die Werte des Ibeo Lux herangezogen werden. Diese betragen 40 mm für die Präzision und maximal 100 mm für die absolute Genauigkeit. Der Scanner



Abbildung 4.2.: Valeo Scala (Quelle: Valeo (2017)).

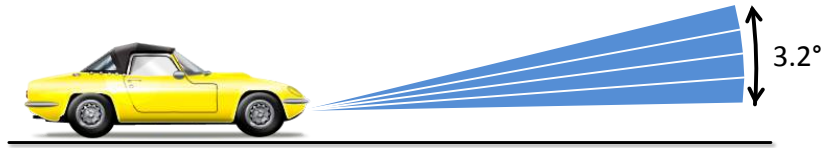


Abbildung 4.3.: Schematische Darstellung der Scannerposition und -Orientierung des Valeo Scala am Messfahrzeug.

arbeitet nach dem Impulslaufzeitverfahren, wobei dem Nutzer die ersten drei reflektierten Pulse zur Auswertung zur Verfügung stehen (GmbH, 2013). In dieser Arbeit wird stets der erste empfangene Puls verwendet.

Für die Messungen der in Abschnitt 7.1 dargestellten Experimente ist der Laserscanner etwa 0,5 m über dem Boden an der Frontstoßstange montiert (siehe Abbildung 4.3). Die Positionierung des Messfahrzeuges geschieht durch eine OXTS RT3002 Inertial/GNSS-Einheit. Um die Genauigkeit der GNSS-Messungen zu verbessern, werden zusätzlich RTK-Korrekturdaten¹ des Satellitenpositionierungsservice AXIO-NET verwendet. Die Positionierungsgenauigkeit der RT3002 in Verbindung mit RTK wird mit 0,01 m angegeben (Oxford Technical Solutions Ltd., 2013). Wie bereits in Abschnitt 4.1 erläutert, dürfte die tatsächliche Genauigkeit, insbesondere in urbanen Gebieten, diesen Wert jedoch überschreiten. Die Genauigkeit des Roll- und Nick-Winkels beträgt $0,03^\circ$, die des Gierwinkels $0,1^\circ$.

4.3. Oxford Robotcar Dataset

Der Datensatz des Oxford Robotcars besteht aus regelmäßigen LiDAR- und Kamera-Aufnahmen über einen Zeitraum von Mai 2014 bis Dezember 2015 in Oxford, Großbritannien. Abbildung 4.4

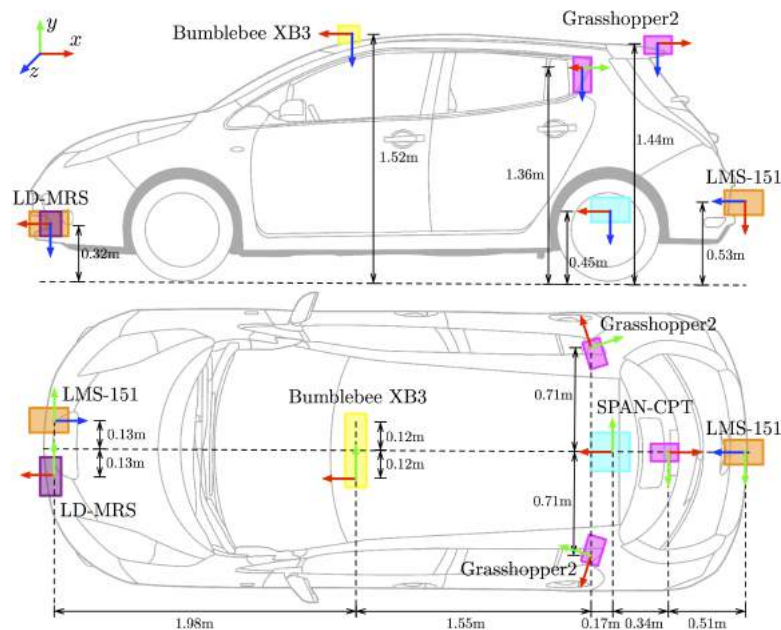


Abbildung 4.4.: Schematische Zeichnung des Oxford Robotcars (Quelle: Maddern u. a. (2017)).

¹Real-time Kinematic



Abbildung 4.5.: SICK LMS151 (Quelle: SICK AG (2017)).

zeigt eine schematische Zeichnung des Fahrzeuges und der angebrachten Sensoren. Eine ausführliche Beschreibung des Datensatzes und des Messfahrzeuges ist in Maddern u. a. (2017) zu finden. Am Robotcar sind vier Kameras, drei Laserscanner sowie ein Inertial/GNSS-System verbaut. In dieser Arbeit wird lediglich der an der Heckstoßstange montierte SICK LMS151 (Abbildung 4.5) zur Lokalisierung verwendet und mit der Positionslösung der GNSS/Inertial-Einheit verglichen.

Der SICK LMS151 ist ein Einzeilen-Laserscanner mit einem Sichtbereich von 270° . Als Messfrequenz sind 25 Hz oder 50 Hz bei einer Winkelauflösung von $0,25^\circ$ bzw. $0,5^\circ$ wählbar. In dieser Arbeit wird eine Frequenz von 50 Hz gewählt, wodurch jede Scanzeile 540 Messungen beinhaltet. Die minimale Messdistanz der Distanzmessungen des Lasers beträgt 0,5 m, die maximale Distanz 18 - 50 m, abhängig von den Reflektionseigenschaften der Oberfläche. Die absolute Genauigkeit wird mit 30 mm angegeben, die Präzision mit 12 mm. Der nach dem Impulslaufzeitverfahren arbeitende Scanner gibt zwei Impulse pro ausgesendetem Signal aus, wobei hier stets der erste ankommende Impuls verwendet wird (SICK AG, 2017).

Als GNSS/Inertial-Sensor kommt ein Novatel-Span-CPT-Sensor zum Einsatz. Dieser liefert eine Positionsgenauigkeit von 1,2 m sowie eine Genauigkeit von $0,02^\circ$ für den Nick- und Roll- und $0,06^\circ$ für den Gier-Winkel (Novatel, 2016). Aufgrund der stark variierenden Genauigkeit der Positionslösung während der Messfahrten empfehlen Maddern u. a. (2017), diese Daten nicht als Referenz zu verwenden. Da die Daten des Robotcars in Abschnitt 7.2.2 lediglich zur Groblokalisierung verwendet werden, kann die Positionslösung der SPAN CPT hier dennoch zum Vergleich herangezogen werden.

4.4. Velodyne VLP-16

Der Velodyne VLP-16 (Puck) ist am selben Fahrzeug angebracht, wie das in Abschnitt 4.1 vorgestellte Riegl VMX-250 Mobile-Mapping-System. Über einen Adapter ist der Laserscanner zusammen mit einer GNSS/Inertial-Einheit Xsens MTi-G vertikal an der Anhängerkupplung montiert (siehe Abbildung 4.6). Zusätzlich befindet sich eine mit dem Laserscanner verbundene GNSS-Antenne zur Zeitsynchronisierung des VLP-16 auf dem Dach des Fahrzeuges.

Der horizontale² Sichtbereich des Velodyne VLP-16 beträgt 360° . In vertikaler Richtung misst der Laserscanner 16 Scanzeilen mit einer Winkelauflösung von 2° , sodass sich 16 Scanstrahlen pro Scan ergeben. Die horizontale Winkelauflösung lässt sich in vier Stufen von $0,1^\circ$ - $0,4^\circ$ in Abständen von $0,1^\circ$ einstellen. Die entsprechende Messfrequenz reicht von 5 - 20 Hz. In dieser Arbeit wird eine Frequenz von 20 Hz gewählt, um so eine möglichst geringe Distanz zwischen zwei Scans zu

²Für die Ausrichtung der Strahlen wurden hier die Konventionen des Herstellers verwendet. Dieser geht davon aus, dass der Sensor horizontal befestigt ist.



Abbildung 4.6.: Anbringung des Velodyne VLP-16 zusammen mit einer Xsens MTi-G an der Anhängerkuppelung des Messfahrzeuges.

gewährleisten. Die Reichweite der Distanzmessungen beträgt bis zu 100 m, die absolute Genauigkeit 30 mm. Zu der minimalen Punktdistanz sowie der Präzision der Messungen wurden keine Angaben gemacht. Eine Überprüfung der minimalen Distanz zeigt, dass sie bei 0,3 m liegt. Die Distanzmessungen finden nach dem Impulslaufzeitverfahren statt, wobei zwei reflektierte Pulse ausgewertet werden können. In dieser Arbeit wird stets der erste reflektierte Puls verwendet (Velodyne LiDAR, 2017).

Um aus den einzelnen Scans des Velodyne VLP-16 eine Punktwolke zu erstellen, ist es nötig, dass für jede Punktmessung die aktuelle Ausrichtung und Position des Scanners bekannt ist. Die Ausrichtung wird zunächst mithilfe der Xsens MTi-G bestimmt. Die absolute Genauigkeit wird für den Nick- und Roll-Winkel mit 1° , für den Gier-Winkel mit 2° angegeben. Die für die Detektion von Änderungen in der Ausrichtung wichtige Winkelauflösung beträgt $0,05^\circ$ in alle drei Raumrichtungen. Zusätzlich verfügt der verwendete Sensor über eine externe GNSS-Antenne, welche ebenfalls auf dem Dach des Fahrzeuges angebracht ist. Die Genauigkeit der Positionslösung wird hier mit 2,5 m angegeben. Die Messfrequenz für alle Werte beträgt 120 Hz. Die Zeitsynchronisierung erfolgt über die durch die GNSS-Antennen des Velodyne VLP-16 und der MTi-G bereitgestellte UTC-Zeit der jeweiligen Messungen (Xsens Technologies B.V., 2012).

4.4.1. Kalibrierung des Systems

In Kapitel 7 soll die aus den Daten des Velodyne VLP-16 bestimmte Position mit der parallel aufgenommenen Referenztrajektorie des Riegl VMX-250 verglichen werden. Hierzu muss die relative Position und Orientierung des Velodyne-Laserscanners zum Referenzpunkt des Riegl Mobile-Mapping-Systems bekannt sein. Die Daten der Sensoren müssen also im selben Koordinatensystem vorliegen. Hierfür wird ein globales Koordinatensystem, hier als Navigationsrahmen N bezeichnet, verwendet, wobei N als eine UTM-Projektion (Universal Transverse Mercator) unter Bezug auf das Europäische Terrestrische Referenzsystem von 1989 ETRS89 definiert ist.

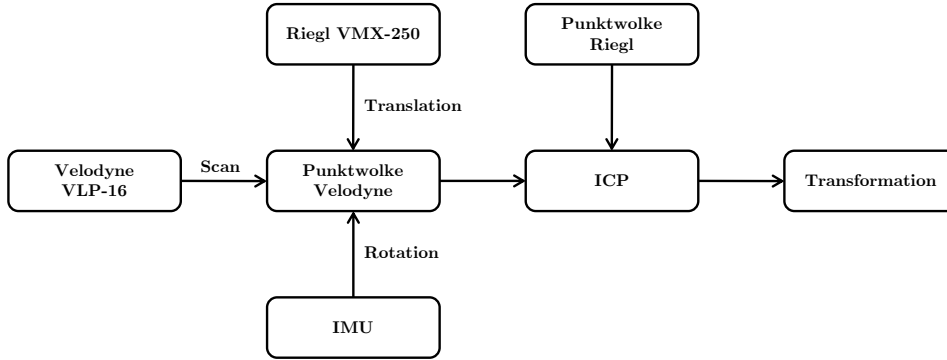


Abbildung 4.7.: Ablauf der Kalibrierung des Velodyne VLP-16 zum Riegl VMX-250.

Zur Bestimmung der Transformation T_{Puck}^N der Messungen des Velodyne VLP-16 in den Navigationsrahmen N sind die folgenden Transformationsparameter bekannt:

- R_{Puck}^{Xsens} : Rotation zur Überführung der Messungen des Velodyne Puck in das Sensorkoordinatensystem der Xsens MTi-G.
- R_{Xsens}^N : Rotation vom Sensorkoordinatensystem der Xsens MTi-G nach N .
- R_{Riegl}^N : Rotation vom Sensorkoordinatensystem des Riegl VMX-250 nach N .
- t_{Riegl}^N : Translation vom Sensorkoordinatensystem des Riegl VMX-250 nach N .

Die Zuordnung der Messungen der verschiedenen Sensoren zueinander geschieht über den GNSS-Zeitstempel. Da die entsprechenden Zeitstempel in der Regel nicht exakt übereinstimmen, werden die entsprechenden Rotationen und Translationen zum Zeitpunkt der Punktmessungen aus den jeweiligen Daten interpoliert.

Der Ablauf der Kalibrierung wird in Abbildung 4.7 skizziert. Zunächst werden die Messungen des Velodyne-Sensors in ein globales Koordinatensystem, den Navigationsrahmen N , überführt. Die entsprechende Rotation R_{Puck}^N ergibt sich aus der folgenden Beziehung:

$$R_{Puck}^N = R_{Xsens}^N R_{Puck}^{Xsens} \quad (4.1)$$

Anschließend wird die Translation t_{Riegl}^N an die Punktkoordinaten des Velodyne-Sensors angebracht, welche sich aus dem interpolierten Standpunkt des Riegl-Referenzpunktes zum Zeitpunkt der Messung eines Punktes ergibt.

Aus der Punktwolkenregistrierung, welche durch einen in Abschnitt 2.1.4 vorgestellten ICP-Algorithmus durchgeführt wird, ergibt sich die Translation $t_{Puck,N}^{Riegl}$ im Navigationsrahmen. Hieraus lässt sich aus

$$t_{Puck}^{Riegl} = R_{Puck}^N{}^{-1} t_{Puck,N}^{Riegl} \quad (4.2)$$

die Translation des Velodyne-Sensorkoordinatensystems in das Sensorkoordinatensystem des Riegl VMX-250 bestimmen. Somit kann ein Punkt X_{Puck} im Sensorkoordinatensystem des Velodyne VLP-16 durch die folgende Beziehung in den Navigationsrahmen N transformiert werden:

$$X_N = R_{Xsens}^N (R_{Puck}^{Xsens} X_{Puck} + t_{Puck}^{Riegl}) + t_{Riegl}^N \quad (4.3)$$

Zur Bestimmung der Transformation wird angenommen, dass die Orientierung des Fahrzeuges und somit der Sensoren, konstant ist. Es werden lediglich Daten aus zehn Scans des Velodyne VLP-16 verwendet, bei einer Fahrtgeschwindigkeit von etwa 10 m/s und einer Messfrequenz von



(a) Orientierung aus Messungen der Xsens MTi-G. (b) Orientierung aus Messungen der Inertialeinheit des Riegl VMX-250.

Abbildung 4.8.: Vergleich der Punktwolken des Velodyne VLP-16 mit den Daten des Riegl VMX-250 nach Anbringung der Rotationsmatrix aus den Messungen der Xsens MTi-G (links, rot) und aus den Messungen der Inertialeinheit des Riegl VMX-250 Mobile-Mapping-Systems (rechts, grün). Die mit dem Riegl VMX-250 aufgenommene Referenzpunktwolke ist nach den Intensitätswerten der jeweiligen Punkte eingefärbt.

20 Hz. In dieser Zeit beträgt die Abweichung aller drei Richtungswinkel weniger als $0,1^\circ$. Nach zehn Messungen (5 m Fahrweg) ergibt sich eine mögliche Positionsabweichung von weniger als 1 cm.

Abbildung 4.8a zeigt die resultierende Punktwolke nach Anbringung der Transformation T_{Puck}^N auf die Messungen des Velodyne VLP-16. Es ist zu erkennen, dass die Punktwolke im Vergleich zur Referenzpunktwolke des Riegl VMX-250 verkippt ist. Dies ist auf Ungenauigkeiten im Roll- und Nick-Winkel der Xsens MTi-G zurückzuführen. Aus diesem Grund wird die entsprechende Rotationsmatrix zur Überführung in das übergeordnete Referenzkoordinatensystem statt aus den Messungen der Xsens MTi-G aus den Messungen der Inertialeinheit des Riegl VMX-250 Mobile-Mapping-Systems bestimmt. Zusätzlich wird die Rotation R_{Puck}^{Riegl} zwischen dem Sensorkoordinatensystem des Velodyne VLP-16 und dem System des Riegl VMX-250 durch einen ICP-Algorithmus bestimmt. In diesem Fall ergeben sich die globalen Koordinaten eines Punktes X_{Puck} aus

$$X_N = R_{Riegl}^N (R_{Puck}^{Riegl} X_{Puck} + t_{Puck}^{Riegl}) + t_{Riegl}^N \quad (4.4)$$

Das Ergebnis der Kalibrierung unter Verwendung von R_{Riegl}^N wird in Abbildung 4.8b dargestellt. Es ist ersichtlich, dass zwar weiterhin Abweichungen zwischen den Punktwolken auftreten, diese im Vergleich zu Abbildung 4.8a jedoch deutlich geringer sind.

Da für verschiedene Messfahrten keine statische Orientierung des Velodyne VLP-16 zur Riegl-Referenzsystem angenommen werden kann, muss diese Kalibrierung für jede Messfahrt durchgeführt werden. Außerdem ist anzumerken, dass die Position des Riegl VMX-250 am Fahrzeug durch die Anbringung über einen Dachgepäckträger ebenfalls um wenige Zentimeter variieren kann.

4.5. Vergleich der verwendeten Laserscanner

In diesem Abschnitt sollen die in den vorherigen Abschnitten vorgestellten LiDAR-Systeme kurz tabellarisch verglichen werden (siehe Tabelle 4.1). Die Werte beziehen sich jeweils auf die in Kapitel 7 verwendete Konfiguration. Für die Präzision und absolute Genauigkeit des Valeo Scala werden, wie bereits in Abschnitt 4.2 beschrieben, die Werte des Ibeo Lux übernommen.

Die für die Referenzmessung verwendeten Laserscanner VQ-250 des Riegl VMX-250 verfügen über die höchste Genauigkeit sowie mit 300 000 Punkte pro Sekunde über die höchste Messfrequenz. Zwar beträgt die Messfrequenz des Velodyne VLP-16 ebenfalls 300 000 Punkte pro Sekunde, jedoch enthält das vorliegende System im Gegensatz zum Riegl-System nur einen Laserscanner. Die absolute Genauigkeit des Velodyne VLP-16 ist mit 30 mm für einen Automotive-Laserscanner ebenfalls vergleichsweise hoch. Der SICK LMS151 verfügt zwar über eine ähnliche Genauigkeit, jedoch über eine deutlich geringere Messfrequenz (29 000 Punkte pro Sekunde) und Reichweite (18 - 50 m). Weiterhin ist es nur möglich, eine Scanzeile aufzunehmen. Dem entgegen steht der Valeo Scala, welcher vier Scanzeilen mit einer maximalen Reichweite von 150 m erfassen kann. Jedoch beträgt hier die absolute Genauigkeit der Distanzmessungen lediglich 100 mm.

Tabelle 4.1.: Vergleich der in dieser Arbeit verwendeten LiDAR-Systeme.

Kriterium	Riegl VMX-250	Valeo Scala	SICK LMS151	Velodyne VLP-16
Minimale Messdistanz	1,5 m	0,3 m	0,5 m	0,3 m
Maximale Messdistanz	75 – 200 m	150 m	18 – 50 m	100 m
Messbereich horizontal	360°	145°	270°	360°
Messbereich vertikal	1 Scanzeile	3,2°	1 Scanzeile	30°
Winkelauflösung horizontal	0,12°	0,25°	0,25°	0,4°
Winkelauflösung vertikal	-	0,8°	-	2°
Absolute Genauigkeit	10 mm	< 100 mm	30 mm	30 mm
Präzision	5 mm	40 mm	12 mm	Keine Angabe
Messfrequenz	100 Hz	25 Hz	50 Hz	20 Hz
Punkte pro Sekunde	600 000	58 000	29 000	300 000

5. Detektion dynamischer Objekte

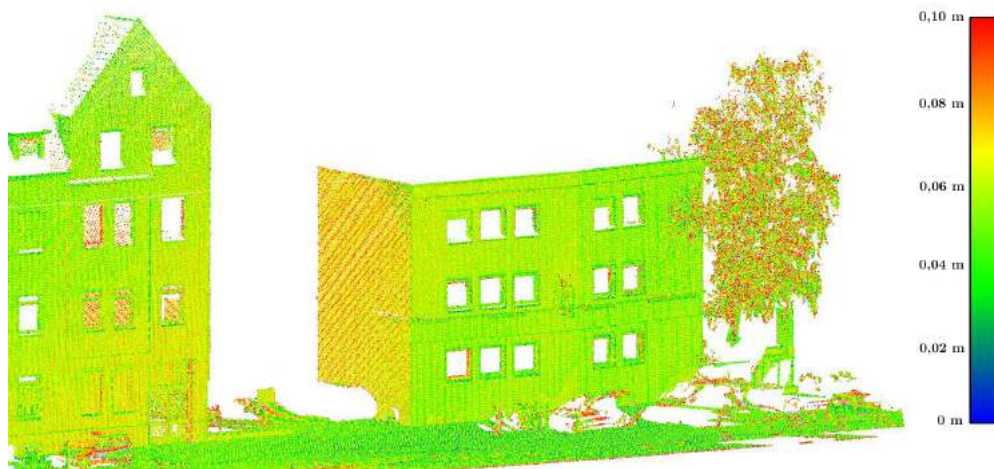
Dynamische Objekte werden in dieser Arbeit durch eine Änderungsdetektion (Abschnitt 5.1) sowie durch eine Klassifizierung mithilfe von Verfahren des maschinellen Lernens (Abschnitt 5.2) detektiert. Eine Änderungsdetektion kann in diesem Fall verwendet werden, wenn mehrere Datensätze vorhanden sind, deren Punktkoordinaten in einem übergeordneten Koordinatensystem näherungsweise bekannt sind. Außerdem ist die Laufzeit einer Änderungsdetektion durch die verschiedenen notwendigen Schritte in der Regel höher als die einer Klassifizierung, die bestenfalls in Echtzeit durchgeführt werden kann. Der Algorithmus zur Änderungsdetektion sowie zur Klassifizierung von Vegetation und dynamischer Objekte in einzelnen Scans wurde bereits in Schlichting und Brenner (2016b) vorgestellt.

5.1. Änderungsdetektion

Dynamische Objekte in den Daten des Mobile-Mapping-Systems Riegl VMX-250 werden, auch aufgrund der großen Datenmengen, durch einen voxelbasierten Vergleich mehrerer Punktwolken detektiert. Hierfür werden die Punktwolken verschiedener Messfahrten zunächst zu einem Referenzdatensatz registriert. Anschließend findet eine Segmentierung freistehender Objekte statt, auf welche eine objektweise Änderungsdetektion angewendet wird.

Abbildung 5.1a zeigt die Abstände der Punkte eines Datensatzes zum nächstliegenden Punkt in einem Referenzdatensatz. Es ist ersichtlich, dass die Punktdistanzen ohne eine Registrierung der Punktwolken zueinander an statischen Objekten wie der Häuserfassade zwischen 6 und 10 cm betragen. In einigen Gebieten der aufgenommenen Daten mit schlechten GNSS-Bedingungen belaufen sich diese Abweichungen auf bis zu 30 cm. Um diese Abstände zu minimieren, wird ein ICP-Algorithmus (siehe Abschnitt 2.1.4) auf Ausschnitte der Daten mit einer Seitenlänge von 40×40 m durchgeführt. Der Anteil an Punkten mit einem zur Fahrtrichtung parallelen Normalenvektor ist vergleichsweise gering. Dies kann dazu führen, dass wenn die gesamte Datenmenge zweier Punktwolken verwendet wird, zur Fahrtrichtung senkrechte Flächen nicht ausreichend gewichtet werden und somit die Translation in Fahrtrichtung nicht ausreichend genau bestimmt werden kann. Weiterhin führen dynamische Objekte inklusive Vegetation zu einer Verschlechterung der Ergebnisse. Ein beispielhaftes Ergebnis des ICP-Algorithmus unter Verwendung aller Punkte ist in Abbildung 5.1b dargestellt. Es ist zu erkennen, dass die Punktabstände der senkrecht zur Fahrtrichtung stehenden Häuserwände weiterhin relativ groß sind. Um diese zu verringern, werden die Daten anhand der lokalen Normalenvektoren der Punkte (mit $r = 0,25$ m) in drei Punktmengen aufgeteilt: Punkte mit einem vertikalen, einem orthogonal zur Fahrtrichtung und einem parallel zur Fahrtrichtung zeigendem Normalenvektor. Zusätzlich werden nur Punkte verwendet, die eine Krümmung σ_P von weniger als 0,05 aufweisen (siehe Formel 2.18). Hierdurch soll sichergestellt werden, dass lediglich planare Flächen in die Registrierung eingehen.

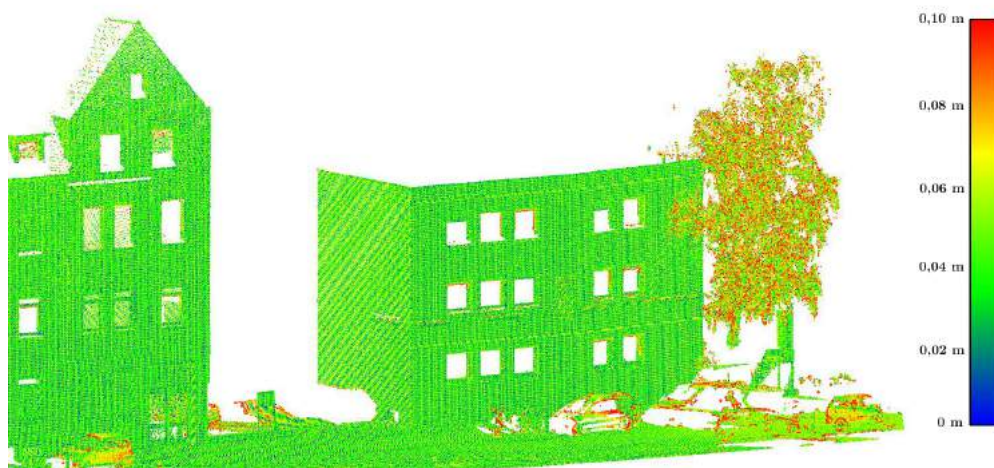
Für jede Punktmenge wird anschließend ein ICP-Algorithmus durchgeführt, wobei lediglich die Translation zur Transformation der Daten verwendet wird. Anschließend werden die translatorischen Elemente der jeweils betrachteten Richtung zu einer Gesamttranslation zusammengefügt und auf die gesamte Datenmenge angewendet. Für den Fall, dass für eine Richtung nicht ausreichend viele Punkte zur Verfügung stehen, wird die ursprüngliche Punktmenge in einem einzelnen ICP-Algorithmus verwendet. Das Ergebnis der hier vorgestellten Methode ist in Abbildung 5.1c dargestellt.



(a) Punktabstände ohne Registrierung.



(b) Punktabstände nach einfachem ICP.



(c) Punktabstände nach getrenntem ICP für alle drei Richtungen und unter Berücksichtigung einer maximalen Krümmung von 0,05

Abbildung 5.1.: Punktabstände einer Punktwolke eines Mobile-Mapping-Systems zu einem Referenzdatensatz.

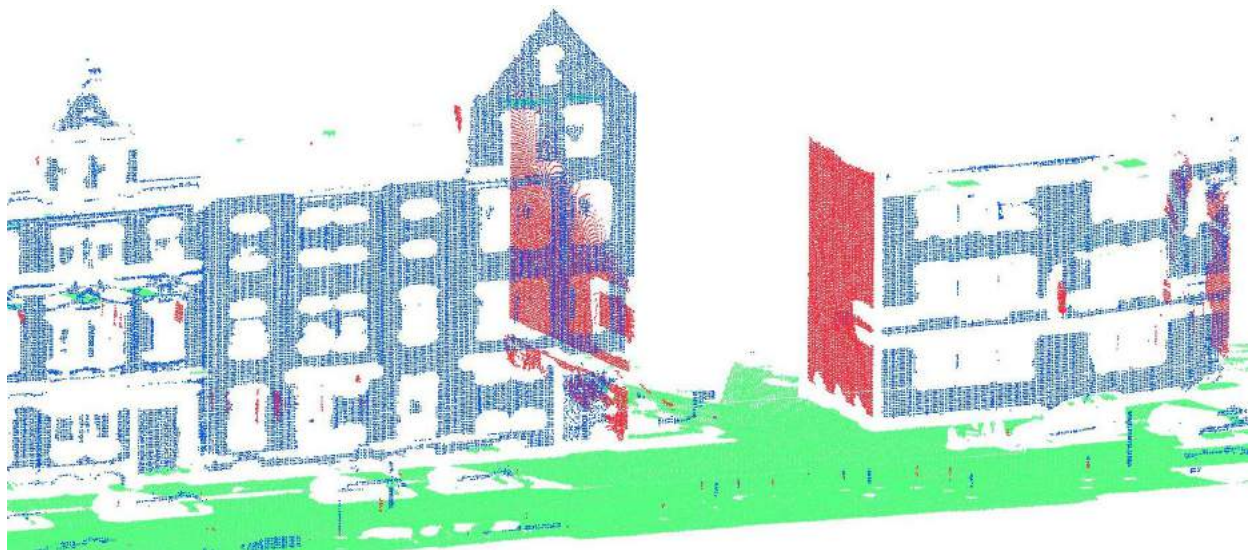


Abbildung 5.2.: Aufteilung der ursprünglichen Punktwolke in drei Punktmengen anhand des lokalen Normalenvektors und einer maximalen Krümmung von 0,05. Punkte mit einem vertikalen Normalenvektor sind grün, Punkte mit einem in Fahrtrichtung zeigenden Normalenvektor rot und Punkte mit einem senkrecht zur Fahrtrichtung zeigenden Normalenvektor blau eingefärbt.

Um freistehende Objekte in den Daten zu segmentieren, wird ein zweischrittiger Region-Growing-Ansatz verfolgt. Zur Segmentierung der einzelnen Objekte wird lediglich die euklidischen Punktdistanz zur Nachbarschaftssuche verwendet. Da die Objekte zunächst durch den Boden miteinander verbunden sind, wird dieser im ersten Schritt aus den Daten entfernt, wobei zusätzlich der Normalenvektor als Homogenitätskriterium verwendet wird. Der Normalenvektor wird unter der Verwendung aller Punkte in einem Radius von 0,5 m um den entsprechenden Punkt berechnet. Punkte mit einem Winkel von weniger als 65° werden der Ebene nicht hinzugefügt. Nachdem 0,5 % der Punkte als Saatpunkte gewählt wurden, wird das Region Growing für die Bodenpunkte mit einer maximalen Punktdistanz von 0,2 m durchgeführt. Nach dem Entfernen der Bodenpunkte wird für die folgende Segmentierung der einzelnen Objekte die maximale Punktdistanz zu 0,25 m gesetzt. Die Wahl der jeweiligen Parameter der Segmentierung geschah zuvor durch eine händische Evaluierung der Ergebnisse.

Liegen unterschiedliche Bodenebenen mit unterschiedlichen Höhen vor, so liegen die gewählten Saatpunkte nur auf der niedrigeren Ebene. Dies hat zur Folge, dass die auf der höheren Bodenebene befindlichen Objekte nicht getrennt werden können. Aus diesem Grund wird der gesamte Algorithmus ein zweites Mal auf alle segmentierten Objekte durchgeführt.

Für jedes segmentierte Objekt des Referenzdatensatzes wird im letzten Schritt eine Nachbarschaftssuche mit Hilfe eines K-d-Baumes durchgeführt, der die Punkte eines weiteren Datensatzes enthält, siehe Abbildung 5.3. Können bei einem Objekt für mehr als 15 % der Punkte keine Nachbarschaftspunkte in einem Radius von 0,25 m gefunden werden, so wird das Objekt als dynamisch markiert. Diese Analyse wird für die einzelnen Objekte auf jeden weiteren Datensatz angewendet. Ein Objekt wird als dynamisch angenommen, sofern es in einem der Datensätze als dynamisch markiert wurde. Objekte mit einer Höhe von mindestens 5 m oder eine Breite von mindestens 10 m werden ohne weitere Betrachtung als statisch angenommen. Abbildung 5.4 zeigt das Ergebnis einer Änderungsdetektion. Die in grün eingefärbten Bäume werden durch einen Klassifizierungsansatz detektiert, welcher im folgenden Abschnitt vorgestellt wird.

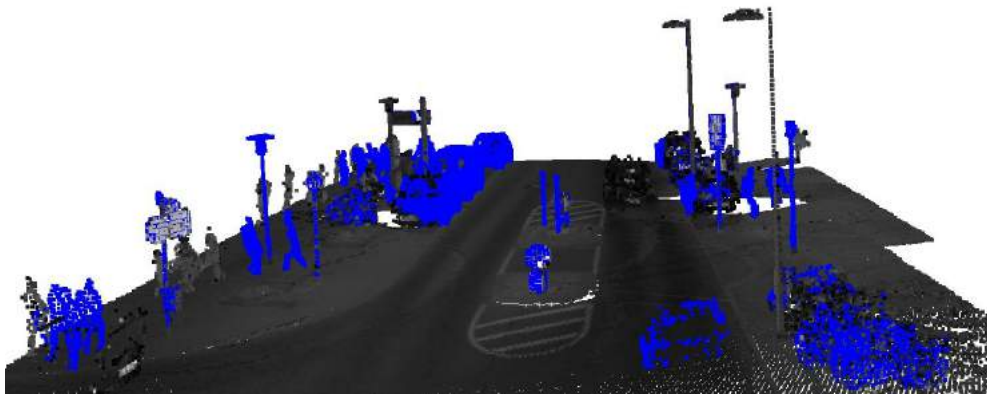


Abbildung 5.3.: Die Segmente einer Referenzpunktwolke (blau) werden mit Segmenten einer weiteren Punkt-
wolke verglichen (eingefärbt nach den Intensitätswerten).

5.2. Klassifizierung dynamischer Objekte

Zur Detektion dynamischer Objekte durch eine Klassifizierung wird ein objektbasierter sowie ein punktbasierter Ansatz verwendet. Das objektbasierte Verfahren wendet einen Random Forest zur Detektion von Vegetation in Daten eines Mobile-Mapping-Systems an. Das punktbasierte Verfahren führt eine pixelbasierte Klassifizierung auf Scanbildern eines mehrzeiligen Automotive-Laserscanners durch, wobei jedes Pixel einem Messpunkt entspricht.

5.2.1. Klassifizierung von Vegetation

Um eine objektbasierte Klassifizierung durch einen Random Forest durchführen zu können, müssen zunächst Merkmale gefunden werden, die diese Objekte beschreiben. In dieser Arbeit werden hierfür 81 Merkmale verwendet. Das erste Merkmal ist die Höhe des Objektes. Zur Bestimmung der weiteren Merkmale wird das Objekt wie in Abbildung 5.5 illustriert in 20 Blöcke mit einer festen Höhe von je 0,5 m aufgeteilt. Alle Punkte, die sich höher als zehn Meter über dem Boden befinden, werden lediglich zur Bestimmung der Höhe verwendet. In jedem Block wird eine lokale PCA für jeden enthaltenen Punkt mit einem Radius von 1 m durchgeführt. Aus den resultierenden

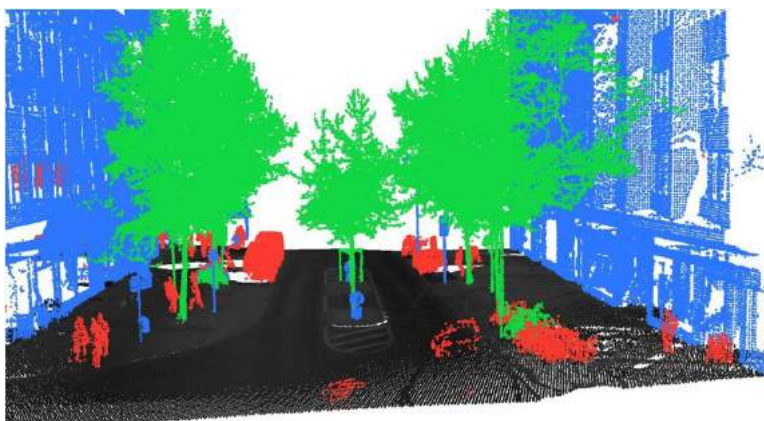


Abbildung 5.4.: Zur Detektion dynamischer Objekte (rot) in den Punkt-Daten eines Mobile-Mapping-Systems wird eine Änderungsdetektion verwendet. Vegetation (grün) wird durch eine Random-Forest-Klassifizierung detektiert. Statische Objekte sind blau, der Boden ist anhand der Intensitätswerte der jeweiligen Punkte eingefärbt.

Eigenwerten λ_1 , λ_2 und λ_3 werden die Linear-, Planar- und Volumenelemente a_1 , a_2 und a_3 wie in Demantke u. a. (2011) und West u. a. (2004) gezeigt nach der folgenden Beziehung bestimmt:

$$a_1 = \frac{\lambda_0 - \lambda_1}{\lambda_0}, a_2 = \frac{\lambda_1 - \lambda_2}{\lambda_0}, a_3 = \frac{\lambda_2}{\lambda_0} \quad (5.1)$$

Hierbei gilt, dass $\lambda_0 > \lambda_1 > \lambda_2$ und $a_1 + a_2 + a_3 = 1$. Die resultierenden Elemente werden anschließend für jeden Block gemittelt. Abbildung 5.5 zeigt einen nach den für jeden Punkt jeweils dominierenden Elementen eingefärbten Baum. Es ist zu erkennen, dass der Stamm aus Linearelementen besteht, wobei die Baumkrone alle drei Varianten enthält. Sind in einem Block keine Punkte enthalten, so werden die Elemente zu Null gesetzt. Weiterhin werden für jeden Block die Breite sowie das Volumen bestimmt.

Tabelle 5.1.: Evaluierung der Detektion von Vegetation durch einen Random Forest.

Precision	Recall	F_1
96 %	95 %	96 %

Zur Erstellung der Trainingsdaten wurden 338 segmentierte Bäume und 14 793 sonstige Objekte manuell mit der entsprechenden Klasse versehen. Der Random Forest besteht in dieser Arbeit aus

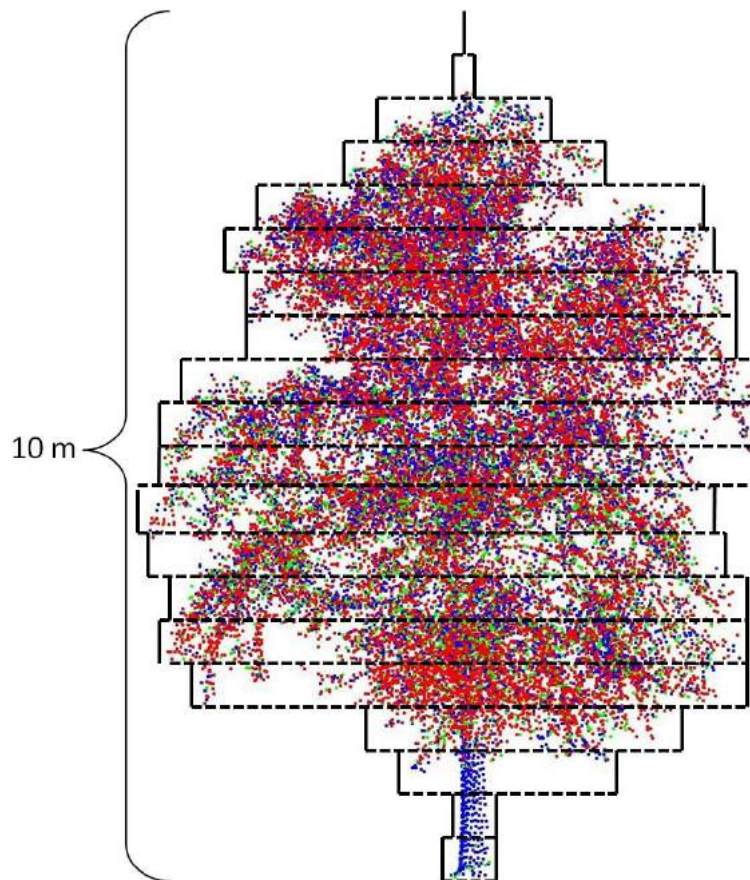


Abbildung 5.5.: Die Punktmenge eines Objektes wird in 20 Blöcke mit einer Höhe von jeweils 0,5 m aufgeteilt. Für jeden Block werden die Breite, das Volumen sowie die mittleren Linear- (blau), Planar- (grün) und Volumenelemente (rot) bestimmt. Die Punkte sind nach dem jeweils dominierenden Element eingefärbt.



(a) Segmente vor der Trennung anhand der Volumenelemente.

(b) Segmente nach der Trennung anhand der Volumenelemente.

Abbildung 5.6.: Da einige der nach unterschiedlichen Farben eingefärbten Objekte durch den ursprünglichen Region-Growing-Algorithmus nicht getrennt werden können, findet im Anschluss an den Algorithmus eine Trennung anhand der Volumenelemente statt.

zehn binären Entscheidungsbäumen mit einer Baumtiefe von zehn Knoten. Eine Kreuzvalidierung ergibt einen Precision-Wert von 96 %, einen Recall-Wert von 95 %, sowie einen F_1 -Score von 96 % (siehe Tabelle 5.1).

Ein Problem, welches die Detektion von Vegetation erschwert, ist, dass entsprechende Objekte oftmals mit weiteren Objekten verbunden sind, siehe Abbildung 5.6a. Insbesondere an Häuserfassaden

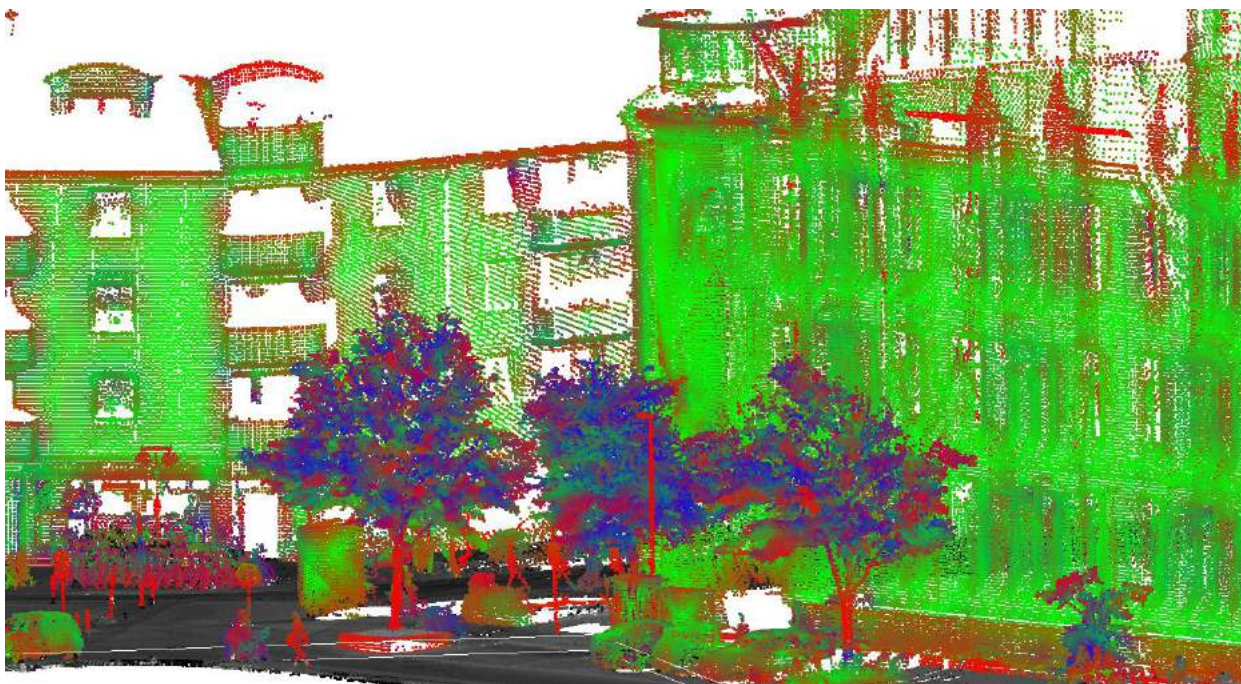


Abbildung 5.7.: Einfärbung der Punktwolke anhand der mit einem Radius von 1 m bestimmten Linear- (blau), Planar- (grün) und Volumenelemente (rot).

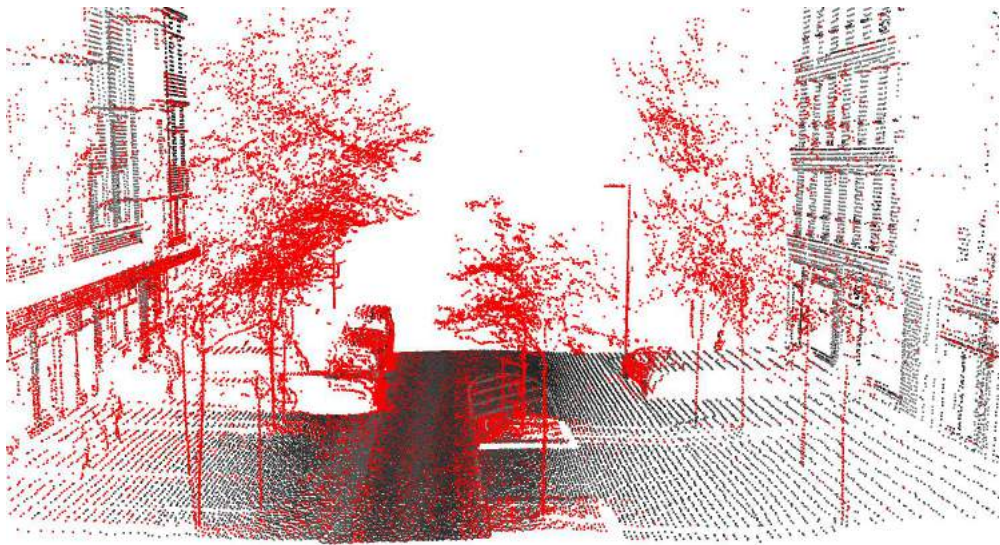


Abbildung 5.8.: Punktwolke eines Zeilen-Laserscanners. Durch einen Random-Forest-Ansatz klassifizierte dynamische Punkte sind rot eingefärbt.

tritt dieser Fall häufig auf. Dadurch, dass im Region Growing lediglich die Punktdistanzen betrachtet werden, ist eine Trennung der Objekte voneinander nicht möglich. Um dieses Problem zu lösen, werden zunächst für jeden Punkt der Segmente die Normalenvektoren n_i und Volumenelemente $a_{3,i}$ durch eine lokale PCA mit einem Radius von 1 m bestimmt. Anschließend werden 3D-Voxel mit einer Zellengröße von 0,5 m erstellt und die jeweils mittleren Normalenvektoren $\bar{n}_j$ und Volumenelemente $\bar{a}_{3,j}$ bestimmt. Abbildung 5.7 zeigt eine nach den Linear-, Planar- und Volumenelementen eingefärbte Punktwolke. Es ist sichtbar, dass sich Vegetation, beispielsweise Baumkronen, stark von der Häuserfassade unterscheiden, an denen die planaren Elemente dominieren. Diese Eigenschaft wird ausgenutzt, um eine Trennung der entsprechenden Objekte durchzuführen. Ausgehend von den Voxeln mit einem mittleren Volumenwert von $\bar{a}_{3,j} > 0.4$ wird erneut ein Region Growing durchgeführt. Als Homogenitätskriterien dienen hierbei $\bar{a}_{3,j}$ und die Länge des Normalenvektors in z -Richtung $\bar{n}_{z,j}$, welche die Bedingungen $\bar{a}_{3,j} > 0.2$ und $\bar{n}_{z,j} < 0.17$ erfüllen müssen. $n_z = 0.17$ entspricht einem Winkel zwischen dem Normalenvektor und der x/y -Ebene von 80° . Entsteht durch das Wachsen der Voxel ein neues Segment mit mindestens 500 Punkten, einer Höhe von mindestens 2 m sowie einer Breite von mindestens 0,5 m, so wird das ursprüngliche Objekt entsprechend aufgeteilt, siehe Abbildung 5.6b.

5.2.2. Klassifizierung dynamischer Objekte in einzelnen Zeilenscans

Zur Klassifizierung der Punktmessungen einzelner Scans wird ebenfalls ein Random-Forest-Ansatz verwendet. In diesem Fall besteht der Merkmalsvektor aus 124 Werten. Die ersten drei Werte ergeben sich aus der relativen Position zum Laserscanner, welche durch die Höhe und horizontale Distanz sowie den Intensitätswert des Punktes beschrieben wird. Zusätzlich werden für jeden Punkt die 20 vorherigen und folgenden Punktmessungen betrachtet. Für diese Messungen werden die relative Höhe zum Referenzpunkt, die horizontale Distanz sowie die Intensitätswerte in den Merkmalsvektor eingetragen. Zuletzt wird die Standardabweichung der eingetragenen Intensitätswerte berechnet und als Merkmal verwendet.

Die Trainingsdaten werden erstellt, indem dynamische Objekte zunächst durch das in Abschnitt 5.1 auf die zur Erstellung der Automotive-Scans verwendeten Referenzdaten angewendet wird. Als Vergleichsdaten dienen fünf Messfahrten, welche zu unterschiedlichen Tageszeiten und an

unterschiedlichen Tagen durchgeführt wurden.

Für den Random-Forest-Klassifizierer werden zehn binäre Entscheidungsbäume mit einer Baumtiefe von 25 Knoten gewählt. Eine Kreuzvalidierung des Verfahrens ergibt einen Recall-Wert von 97 %, einen Precision-Wert von 91 % sowie einen F_1 -Score von 94 % (siehe Tabelle 5.2).

Tabelle 5.2.: Evaluierung der Detektion dynamischer Objekte in Scanzeilen durch einen Random Forest.

Precision	Recall	F_1
97 %	91 %	94 %

Abbildung 5.8 zeigt eine Punktwolke, welche sich aus einzelnen klassifizierten Scans ergibt. Fahrzeuge, Fußgänger und Bäume können korrekt klassifiziert werden. False-Positive-Detektionen treten auf dem Bürgersteig, an Straßenlaternen sowie am unteren Bereich des Gebäudes auf.

5.2.3. Klassifizierung dynamischer Objekte in Scanbildern

Zur Klassifizierung dynamischer Objekte in den Scanbildern eines Automotive-Laserscanners wird ein CNN verwendet, siehe Abschnitt 2.2.4. In diesem Fall soll Vegetation, beispielsweise in Form von Bäumen, nicht als dynamisch betrachtet werden (siehe Abschnitt 6.3.2). Aus diesem Grund findet während der Erstellung der Trainingsdaten mit Hilfe des in 5.1 vorgestellten Verfahrens keine Klassifizierung von Vegetation statt. Die Dimension der Scanbilder ergibt sich aus der verwendeten Messfrequenz von 20 Hz zu 900×16 Pixeln, welche die auf einen Wert zwischen 0 und 1 normierte gemessene Distanz enthalten.

Die Trainingsdaten werden ähnlich wie in Abschnitt 5.2.2 beschrieben erstellt. Zunächst wird eine globale Registrierung der Punktmessungen des Laserscanners durchgeführt (siehe Abschnitt 4.4.1), unter Verwendung der vom Riegl VMX-250-System gemessenen Position und Orientierung. Anschließend findet eine Änderungsdetektion der Daten des Velodyne-Laserscanners zu fünf Datensätzen des Riegl-Referenzsystems statt. Hierbei werden Objekte nach demselben Region-Growing-Verfahren wie in Abschnitt 5.1 aus den Daten des Velodyne VLP-16 segmentiert. Objekte, deren Anteil an dynamischen Punkten größer als 15 % beträgt, werden in den Scanbildern als dynamisch gekennzeichnet. Ein Beispiel eines mithilfe der Änderungsdetektion erstellten Scanbildes ist in Abbildung 5.9a dargestellt. Die in rot eingefärbten Pixel stellen hierbei dynamische Punkte dar. Es ist zu erkennen, dass einige Bereiche des Fahrzeuges nicht als dynamisch markiert sind. Dies resultiert daraus, dass das Fahrzeug aufgrund der Reflektionseigenschaften der Fahrzeugoberfläche, insbesondere im Fensterbereich, vom Laserscanner nicht vollständig erfasst werden konnte. Weiterhin zeigt sich, dass Bereiche des Bodens ebenfalls als dynamisch markiert sind. Grund hierfür ist, dass nur Punkte mit einem senkrechten Normalenvektor während des Segmentierungsprozesses aus den Daten entfernt werden (siehe Abschnitt 5.1). Zur Berechnung der lokalen Normalenvektoren wird ein Umkreis mit einem Radius von 0,5 m verwendet, sodass in diesem Fall zum Fahrzeug gehörige Punkte ebenfalls in die Berechnung des Normalenvektors eingehen. Dies kann wiederum zur Folge haben, dass der Normalenvektor nicht hinreichend senkrecht zur eigentlichen Bodenebene ist und der entsprechende Punkt somit nicht als Bodenpunkt erkannt wird. Da die verbleibenden Bodenpunkte in diesem Beispiel durch die Fahrzeugaufbauten mit dem Boden verbunden sind, werden sie dem Objekt zugewiesen und folglich als dynamische Punkte markiert.

Abbildung 5.10 zeigt den Aufbau des CNN, welches zur Detektion dynamischer Objekte in den Scanbildern verwendet wird. Das eingehende Scanbild der Dimension 900×16 enthält die Distanz-

messungen des Laserscanners. In den ersten drei Schichten findet eine Convolution mit 32 Merkmalskarten und einer Kernelgröße von 3 statt, wobei die Reduktion der Dimension durch ein Max-Pooling durchgeführt wird. Anschließend folgen zwei fully connected Layer. In den Deconvolution-Schichten, welche nach Zeiler u. a. (2010) durchgeführt werden, wird die ursprüngliche Zieldimension von $900 \times 16 \times 2$ erreicht. Für jedes Pixel findet im Output-Layer ein One-Hot-Encoding durch eine Softmax-Funktion statt. In diesem Fall bedeutet dies, dass der jeweilige Kanal die Wahrscheinlichkeit angibt, dass ein Pixel dynamisch oder statisch ist. Als Aktivierungsfunktion in den einzelnen Layern wird eine leaky ReLU, als Optimierer ein Adam-Optimierer verwendet. Das Netzwerk wurde mit 55 000 durch eine Änderungsdetektion klassifizierten Scanbildern und einer Batch-Größe von 100 trainiert.

Tabelle 5.3.: *Evaluierung der Detektion dynamischer Objekte in Scanbilder eines Velodyne VLP-16 durch ein CNN.*

Precision	Recall	F_1
77 %	58 %	66 %

Abbildung 5.9b zeigt ein beispielhaftes Ergebnis der Klassifizierung für ein Scanbild. Es ist zu erkennen, dass zwar ein Großteil des Fahrzeuges detektiert, in diesem Fall jedoch insbesondere der obere Bereich, welcher in Abbildung 5.9a noch rot markiert ist, in Abbildung 5.9b nicht als dynamisch markiert wurde. Tabelle 5.3 gibt Auskunft über die Genauigkeit der Klassifizierung, welche durch eine Evaluierung eines Datensatzes durchgeführt wurde, welcher nicht Bestandteil der Trainingsdaten ist. Es zeigt sich, dass die Genauigkeit der Klassifizierung relativ niedrig ist. Ein Grund könnte in den inkonsistenten Trainingsdaten liegen, in welchen zum Teil der Boden

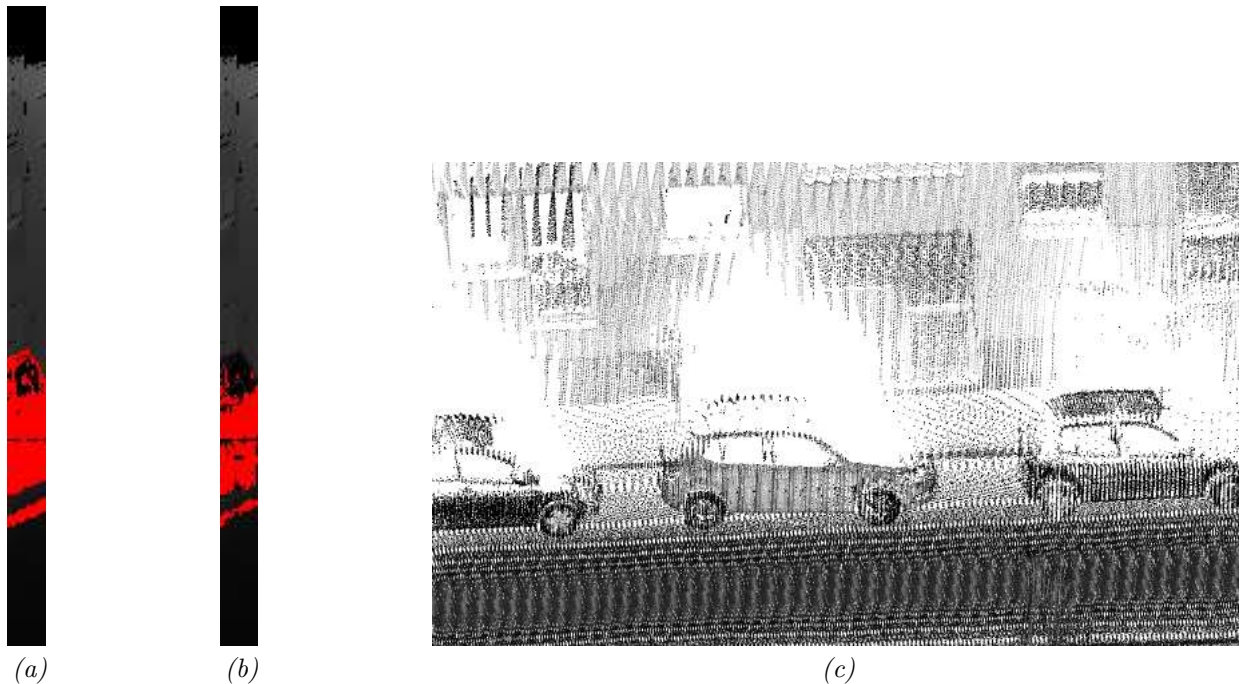


Abbildung 5.9.: *Ergebnis der Klassifizierung dynamischer Objekte durch eine Änderungsdetektion (a) sowie durch ein CNN (b). Dynamische Punkte sind rot, sonstige Punkte nach der gemessenen Distanz eingefärbt. Die aus mehreren Scans des Gebietes zusammengesetzte Punktwolke wird rechts (c) dargestellt, eingefärbt nach den Intensitätswerten der Punkte.*

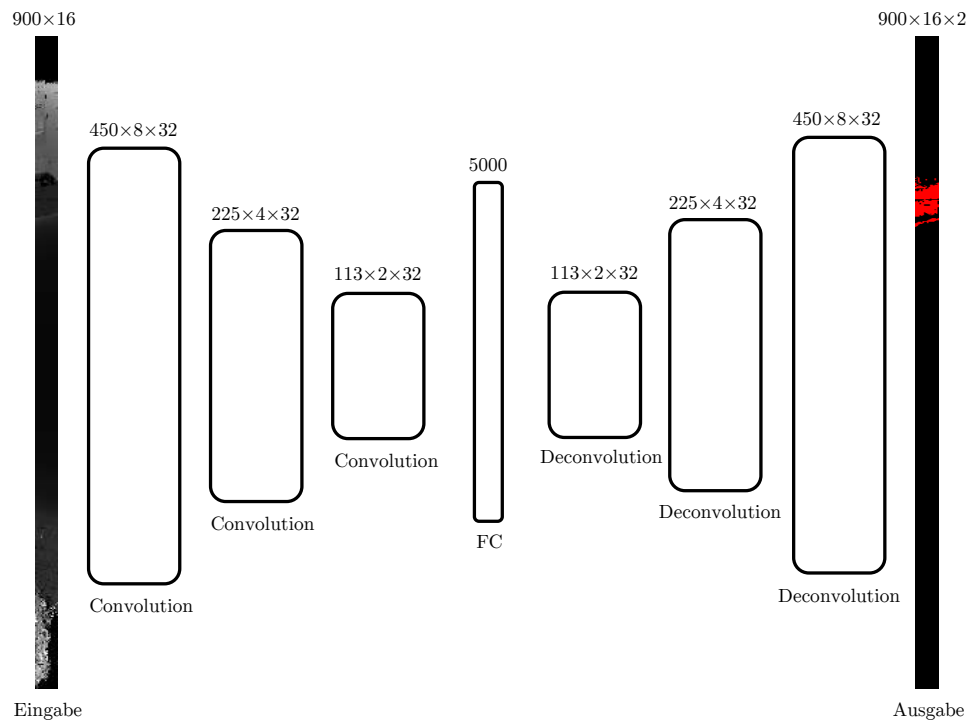


Abbildung 5.10.: Das verwendete CNN zur Detektion dynamischer Objekte in Scanbildern des Velodyne VLP-16. Nach drei Convolutions mit jeweils 32 Merkmalskarten folgen zwei fully connected Layer. Anschließend werden drei Deconvolutions durchgeführt, deren Ergebnis ein Scanbild mit zwei Kanälen ist.

als dynamisch und Teile der dynamischen Objekte, beispielsweise. Fahrzeuge, nicht als dynamisch markiert wurden.

6. Fahrzeug-Lokalisierung durch Automotive-Laserscanner

Es werden verschiedene Methoden zur Bestimmung einer Fahrzeug-Position durch die Messungen von Automotive-Laserscannern vorgestellt: die Lokalisierung über aus den Daten extrahierte stangenförmige Objekte und Ebenen, die Durchführung einer Sequenzanalyse sowie die Verwendung von Scanbildern zur Positionsbestimmung. Hierbei wird jeweils eine Referenzkarte erstellt, welche bestimmte aus den Referenzdaten extrahierte Merkmale enthält. Diese Merkmale werden ebenfalls aus den Online-Daten extrahiert und anschließend mit der Referenzkarte abgeglichen.

6.1. Verwendung stangenförmiger Objekte und Ebenen

In diesem Abschnitt werden aus den LiDAR-Daten eines Automotive-Laserscanners detektierte stangenförmige Objekte und Ebenen zur Fahrzeug-Lokalisierung verwendet. Die hier aufgeführten Ansätze basieren auf den in Schlichting und Brenner (2014a) und Schlichting und Brenner (2014b) vorgestellten Methoden.

Stangenförmige Objekte und Ebenen werden hier durch den Mittelpunkt x_S/y_S und den Radius r_S der Stangen sowie den Normalenvektor n und die Distanz zum Ursprung eines lokalen Koordinatensystems d_0 für Ebenen parametrisiert. Diese Landmarken werden sowohl aus den Daten eines Mobile-Mapping-Systems als auch aus Automotive-Daten segmentiert, wobei die Mobile-Mapping-Landmarken als Referenz dienen. Die aus den Daten des Automotive-Scanners segmentierten Objekte werden im Folgenden als Online-Landmarken bezeichnet.

Die Verwendung von Stangen und Ebenen hat verschiedene Vorteile. So sind diese Objekte in der Regel nicht dynamisch. Sie verändern sich weder in ihrer Lage im Raum noch in ihrer Form maßgeblich. Zwar können stangenförmige Objekte entfernt oder aufgestellt werden, beispielsweise im Fall von mobilen Ampeln, jedoch lässt sich diese Änderungen bereits nach wenigen Messungen verschiedener Fahrzeuge zuverlässig detektieren. Selbiges gilt für entfernte oder neu entstandene Ebenen, wie beispielsweise Fassaden eines neu errichteten Hauses. Eine besondere Herausforderung ergibt sich jedoch für Arbeiten an der Fassade. Dämmungsarbeiten erweitern die Fassade beispielsweise in Richtung der Straße und verändern somit die Ebenenparameter. Die Detektion dieser geringfügigen Änderungen ist jedoch nicht Bestandteil dieser Arbeit. Die Messungen zur Erstellung einer Referenzkarte sowie die Messungen des Automotive-Laserscanners liegen hier zeitlich so nah aneinander, dass von konstanten Werten für die Stangen- und Ebenenparametern ausgegangen wird.

Ein weiterer Vorteil bei der Verwendung von Stangen und Ebenen zur Positionsbestimmung ist, dass sie insbesondere in urbanen Gebieten häufig auftreten. Stangen werden für Straßenlaternen, Verkehrsschilder, Ampeln und weitere Bestandteile der Städte benötigt, klar definierte Ebenen finden sich vor allem an Gebäuden.

Des Weiteren sind 3D-Objekte weniger anfällig gegenüber Witterungs- und Belichtungsbedingungen, da die Oberfläche der Objekte nicht zur Positionsbestimmung benötigt wird. Entsprechend können die Stangen- und Ebenenparameter auch bei Nacht und Schneefall problemlos bestimmt und zur Fahrzeug-Lokalisierung verwendet werden.

6.1.1. Erstellung der Referenzkarte

Die Referenzkarte enthält Stangen und Ebenen, welche aus den Daten eines Messsystems mit übergeordneter Genauigkeit extrahiert werden. In diesem Fall werden Laserscanning-Daten des in Abschnitt 4.1 vorgestellten Riegl VMX-250 Mobile-Mapping-Systems verwendet.

Stangenförmige Objekte werden mithilfe der in Brenner (2009a) vorgestellten Methode detektiert (siehe Abschnitt 2.1.6). Für die Höhe der einzelnen Zellen, aus denen sich die Stangen bilden, wird ein Wert von 0,2 m gewählt, wobei mindestens zwei in der Höhe aufeinanderfolgende Zellen auftreten müssen, damit diese berücksichtigt werden. Die minimale Höhe einer Stange beträgt 1,5 m. Als maximaler Zylinderradius wird 0,25 m verwendet, für den Bereich, in dem keine weiteren Punkte auftreten dürfen, ein Radius 1 m. Die entsprechenden Parameter sind auch in Tabelle 6.1 aufgeführt.

Der Mittelpunkt der Stangen wird aus den Punkten der zur Stange gehörigen Zylinderzellen bestimmt. Hierzu wird die Klasse `pcl::SACSegmentationFromNormals` der C++-Bibliothek Point Cloud Library (PCL) (Point Cloud Library, 2017) verwendet, die einen RANSAC-Algorithmus zur Bestimmung der Zylinderparameter ausführt.

Tabelle 6.1.: Gewählte Parameter zur Segmentierung der Stangen aus den Referenzdaten.

Parameter	Wert
Zellenhöhe	0,20 m
Mindesthöhe aufeinanderfolgender Zellen	0,40 m
Max. Zylinderradius	0,25 m
Radius des äußeren Bereichs	1,00 m

Ebenen werden mit Hilfe eines Region-Growing-Algorithmus (siehe Abschnitt 2.1.6) aus den Daten segmentiert. Hierzu wurden zuvor für jeden Punkt Normalenvektoren berechnet, indem die Oberflächennormale einer lokalen Ebene n_P innerhalb eines festgelegten Radius (hier 1 m) bestimmt wird. Für den Fall, dass der Normalenvektor nicht in Richtung des Nullpunktes zeigt, findet ein Vorzeichenwechsel des Vektors statt. Der Winkel α zwischen der Normalen der lokalen Ebene n_P eines Punktes p und der Ebenennormalen der aktuell im Region-Growing betrachteten Ebene n_E wird zusammen mit der Distanz des Punktes zur Ebene d_{Ebene} als Homogenitätskriterium verwendet. Der Winkel zwischen den Normalen α_{Ebene} ergibt sich aus

$$\alpha = \arccos \frac{n_{PE}}{|n_P| \cdot |n_E|}, \quad (6.1)$$

die Distanz zur Ebene kann aus

$$d = \frac{|(p - a) \cdot n_E|}{|n_E|}, \quad (6.2)$$

bestimmt werden, wobei a hierbei der Stützvektor der Ebene in der Hesseschen Normalform ist. Die Grenzwerte α_{max} und d_{max} werden zu 45° und 0,1 m gesetzt. Um zu vermeiden, dass viele kleine und gegebenenfalls dynamische Objekte segmentiert werden, werden nur Ebenen mit einer Mindestpunkanzahl von 2000 Punkten und einer minimalen horizontalen und vertikalen Ausdehnung von 3 m verwendet. Tabelle 6.2 zeigt alle gewählten Grenzwerte auf.

Tabelle 6.2.: Gewählte Parameter zur Segmentierung der Ebenen aus den Referenzdaten.

Parameter	Wert
α_{max}	45°
d_{max}	0,10 m
Mindestpunktanzahl	2000 Punkte
Max. Ausdehnung vertikal	3,00 m
Max. Ausdehnung horizontal	3,00 m

6.1.2. Extraktion der Landmarken aus den Messungen eines horizontal montierten Automotive-Zeilenscanners

Im Vergleich zu einem Mobile-Mapping-System ist der Sichtbereich horizontal montierter Automotive-Laserscanner stark begrenzt. Abbildung 6.1 zeigt einen typischen Laserscan eines horizontal befestigten Laserscanners, wie er in Abschnitt 4.2 beschrieben wird. Die Detektion der Landmarken erfolgt direkt in den Distanzprofilen der LiDAR-Messungen des Zeilenscanners. Stangenförmige Objekte treten in der Regel freistehend im Raum auf. In den Distanzprofilen macht sich diese Eigenschaft durch Distanzsprünge in den vorherigen und nachfolgenden Messungen innerhalb einer Scanlinie bemerkbar. Da der Durchmesser der Stangen begrenzt ist, muss im Falle einer Stange die Anzahl der aufeinanderfolgenden Messungen ohne Distanzsprung ebenfalls niedrig sein. Die maximalen Anzahl an Messungen k_{max} (aufgerundet) kann aus der Beziehung

$$k_{max} = \frac{r_{S,max}}{2 \cdot (d_S + r_{S,max}) \cdot \tan \frac{\varphi}{2}} \quad (6.3)$$

bestimmt werden. Der maximale Radius $r_{S,max}$ wird auf die gemessene Distanz d_S addiert, da der Laserstrahl nicht den Mittelpunkt des Zylinders, sondern die Oberfläche trifft. Bei einem maximalen Stangenradius von in diesem Fall $r_{S,max} = 0,25$ m, einer Distanz d_S von 10 m und einer Winkelauflösung φ von 0,25° beträgt die Anzahl der maximalen aufeinanderfolgenden Messungen $n_{S,10m} = 6$.

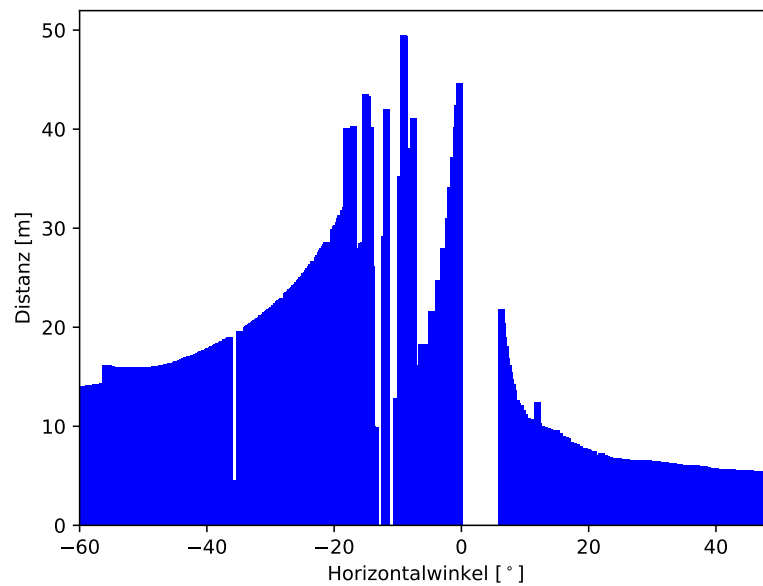


Abbildung 6.1.: Beispielhaftes Profil eines Zeilenscans des Valeo Scala.

Im nächsten Schritt werden die 3D-Koordinaten der Stangen berechnet. Aufgrund der geringen Genauigkeiten der Distanzmessungen ($\sigma_d = 10 \text{ cm}$, siehe Abschnitt 4.2), ergibt eine Zylinderschätzung nicht die gewünschten Ergebnisse. Stattdessen wird das arithmetische Mittel aller Radien der in der Referenzkarte gespeicherten Stangen $\bar{r}_S = 0,08 \text{ m}$ verwendet. Um zu berücksichtigen, dass Stangen in mehreren Scans detektiert werden und somit die Zuverlässigkeit der Detektion zu erhöhen, wird ein Score s_S eingeführt. Für jede neu detektierte Stange wird überprüft, ob in einem Suchradius von $r_{Such} = 0,5 \text{ m}$ bereits eine Stange detektiert und in die Online-Karte eingetragen wurde. Ist dies der Fall, so wird s_S um $\frac{1}{d_S^2}$ erhöht, wobei d_S die euklidische Distanz der Stangen zueinander ist. Der Startwert des Scores beträgt 1. Im Lokalisierungsschritt soll stets die Position anhand der aktuellen Messungen bestimmt werden. Aus diesem Grund wird anstatt den Stangenmittelpunkt über mehrere Stangen zu mitteln stets der aus der aktuellen Messung bestimmte Mittelpunkt verwendet.

Im linken Bereich des Laserscans aus Abbildung 6.1 ist eine typische Abfolge von Messungen an einer Ebene dargestellt, welche sich durch kontinuierliche, geringe Änderungen der gemessenen Distanzen auszeichnet. Ebenenkanten können wiederum durch Distanzsprünge detektiert werden. Beträgt die Ausdehnung einer Ebene mindestens 1 m , so werden alle zur Ebene gehörigen Punkte sowie die Ebenenparameter in der Online-Karte gespeichert. Die Berechnung der Parameter geschieht durch eine PCA. Da bekannt ist, dass die Ebenen vertikal ausgerichtet sind, wird die z-Komponente des Normalenvektors zu $n_{P_i,z} = 0$ gesetzt.

Mit einem einzelnen Scan kann nur ein kleiner Bereich einer Ebene abgedeckt werden. Ähnlich wie auch bei detektierten Stangen werden die Daten der Punkte aus den detektierten Ebenen der einzelnen Scans aggregiert. Für den Fall, dass sich zwei Ebenen überlappen und der Winkel zwischen den Ebenen β ebenfalls unter einem bestimmten Grenzwert (hier $\beta_{max} = 20^\circ$) liegt, werden die Ebenen vereinigt. Die Berechnung des Winkels erfolgt wie in Gleichung 6.2 anhand eines Vergleichs der Normalenvektoren. Auch hier werden die Ebenenparameter nicht aus allen vergangenen Messungen, sondern nur aus der jeweils aktuellen Messung bestimmt.

Insbesondere durch die Erfassung von Fahrzeugen durch den Automotive-Laserscanner können False-Positive-Detektionen von Ebenen auftreten. Die Anzahl an fälschlichen Detektionen wird durch die folgenden zwei Maßnahmen reduziert (siehe Abbildung 6.2):

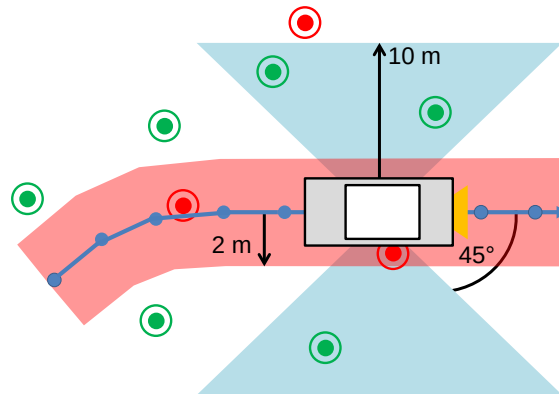


Abbildung 6.2.: Die Anzahl der False-Positive-Detektionen stangenförmiger Objekte und Ebenen wird durch eine maximale Distanz zur Trajektorie von 10 m (blauer Bereich) sowie einer minimalen Distanz von 2 m (roter Bereich) reduziert. Verwendete Landmarken sind grün, verworfene Landmarken sind rot eingefärbt. Die Trajektorie ist in blau dargestellt.

- Üblicherweise befinden sich sowohl Stangen als auch Ebenen am Straßenrand. Auf der Straße sind keine Landmarken zu finden. Aus diesem Grund werden nur Landmarken verwendet, die sich mindestens 2 m von der Fahrzeug-Trajektorie entfernt befinden.
- Es werden nur detektierte Landmarken berücksichtigt, welche sich weniger als 10 m von mindestens einem Trajektorie-Punkt entfernt befinden. Der Abstand aufeinanderfolgender Trajektorie-Punkte zueinander beträgt 0,1 m.

6.1.3. Extraktion der Landmarken aus den Messungen eines vertikal montierten Automotive-Laserscanners

Im Falle eines vertikal montierten Laserscanners entfällt das Problem, dass lediglich ein kleiner Bereich der Stangen und Ebenen im Sichtbereich des Sensors liegt. Durch die zusätzlich hohe Messfrequenz von bis zu 300 000 Punkten pro Sekunde ist die resultierende Punktdichte relativ hoch, sodass das in Abschnitt 6.1.1 vorgestellte Verfahren zur Extraktion stangenförmiger Objekte verwendet werden kann. Statt auf einzelne Scans wird das Verfahren für einzelne Standpunkte entlang der Trajektorie auf die Punktmengen angewendet, welche sich aus den Messungen der zuletzt zurückgelegten 20 m ergeben. Abbildung 6.3 zeigt die aus Messungen des Velodyne VLP-16 extrahierten stangenförmige Objekte. Im Vergleich zu Abschnitt 6.1.2 können hierbei die Stangen vollständig erfasst werden. False-Positive-Detektionen treten an Fußgängern sowie Gebäuden auf. Da zur Fahrtrichtung senkrechte Gebäudefassaden nur von wenige Scanlinien erfasst werden, ergeben sich hier aufgrund der in diesem Bereich geringen horizontalen Auflösung ebenfalls scheinbar stangenförmige Objekte.

Statt eines Region-Growing-Ansatzes wird die Ebenendetektion zur Reduzierung der Rechenzeit mithilfe eines RANSAC-Algorithmus (siehe Abschnitt 2.1.6) durchgeführt. Hierbei wird als Wahrscheinlichkeit, dass ein passendes Modell gefunden wird, $z = 99 \%$, als Wahrscheinlichkeit,

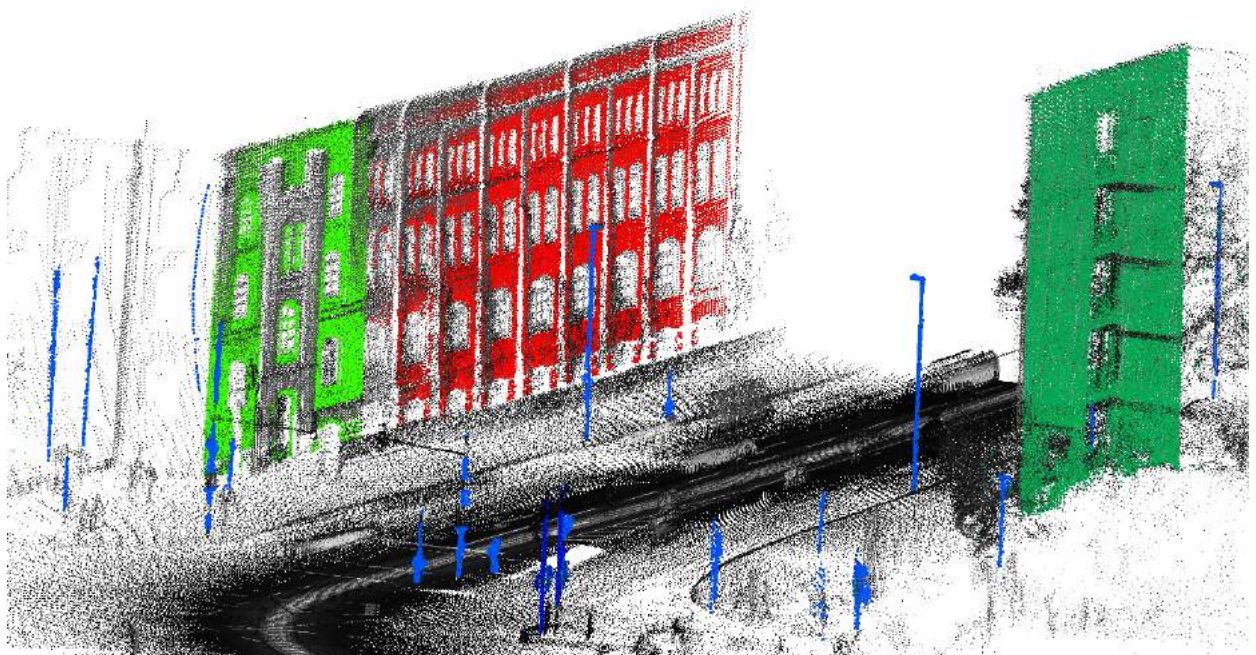


Abbildung 6.3.: Ergebnisse der Segmentierung stangenförmiger Objekte (blau) und Ebenen (hellgrün, rot, dunkelgrün) aus den Messungen eines Velodyne VLP-16, welche nach der Intensität des reflektierten Signals eingefärbt sind.

dass ein Punkt zu einem Modell gehört $w = 10\%$ gewählt, sodass insgesamt 4603 Iterationen benötigt werden. Es werden nur Ebenen mit einem Normalenvektor mit einem Winkel von 75° bis 105° verwendet. Abbildung 6.3 zeigt mit Hilfe des RANSAC-Algorithmus extrahierte Ebenen in hellgrün, rot und dunkelgrün. Da der grün eingefärbte Abschnitt der Häuserfassade näher zur Straße steht als der in rot eingefärbte Abschnitt, ist die Fassade in diesem Fall in zwei Abschnitte aufgeteilt.

6.1.4. Landmarken-Muster

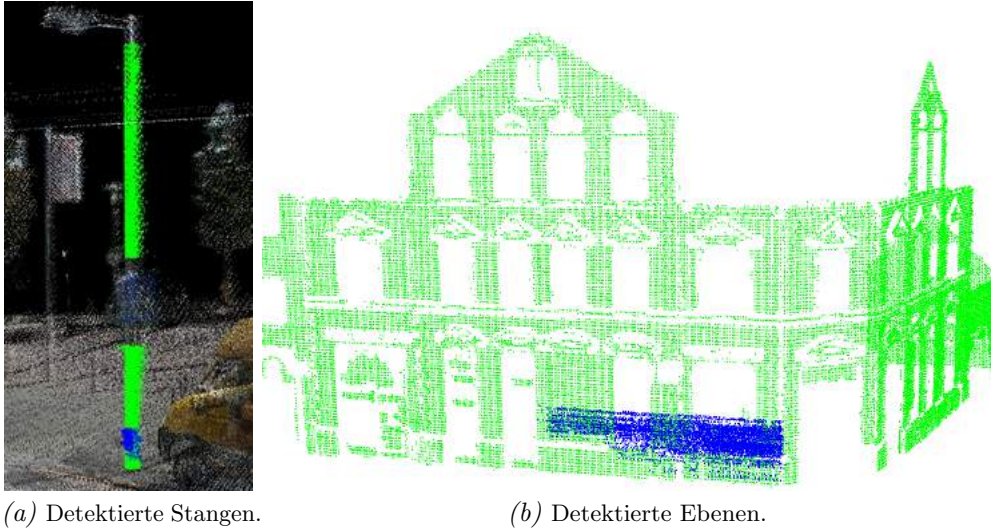


Abbildung 6.4.: Detektierte stangenförmiger Objekte und Ebenen in den verschiedenen LiDAR-Daten. Die Punkte des Riegl VMX-250 Mobile-Mapping-Systems sind grün, die Punkte des Automotive-Zeilenscanners blau eingefärbt.

Abbildung 6.4 vergleicht die zu den Landmarken gehörigen Punkte, welche durch ein Mobile-Mapping-System sowie durch einen Automotive-Zeilenscanner (siehe Abschnitt 6.1.2) erfasst wurden. Es ist ersichtlich, dass in den Automotive-Daten nur ein kleiner Bereich der Stange erfasst wird. Dies erschwert eine direkte Zuordnung der Landmarken aus den Automotive-Daten zu den entsprechenden Referenzen und verursacht Fehldetektionen. Aus diesem Grund wird ein Deskriptor zwischen mindestens drei Landmarken verwendet, der ein Landmarken-Muster beschreibt. Dieser Deskriptor wird sowohl für die Referenz- als auch für die Online-Landmarken bestimmt. Werden neue Landmarken detektiert, so wird hierfür der Deskriptor berechnet und der passende Referenz-Deskriptor gesucht. Im Wesentlichen wird zur Umsetzung das in Brenner (2009b) beschriebene Verfahren verwendet, jedoch um Ebenen ergänzt.

Deskriptor

Der in Brenner (2009b) beschriebene Deskriptor beinhaltet $\dim = 2 \cdot k_{\text{Stangen}} - 3$ Werte. Für eine Menge von $k_{\text{Stangen}} \geq 2$ Landmarken ist der erste Wert D durch die Maximaldistanz zwischen zwei Landmarken der aktuellen Menge bestimmt. Dieser Durchmesser definiert außerdem die x -Achse des lokalen Landmarkenmuster-Koordinatensystems. Die y -Achse steht entsprechend senkrecht hierzu. Die Orientierung des Koordinatensystems wird dadurch festgelegt, dass die Ausdehnung der verbleibenden $k_{\text{Stangen}} - 2$ Landmarken in $+y$ größer als in $-y$ ist. Nachdem der Schwerpunkt des Koordinatensystems von den verbleibenden Landmarken subtrahiert wird, werden diese in lexikographischer Ordnung dem Deskriptor hinzugefügt.

Im Gegensatz zu Brenner (2009b) werden hier neben stangenförmigen Objekten auch aus den LiDAR-Daten extrahierte Ebenen als Landmarken verwendet, um so die Landmarken-Zuordnung zu verbessern. Die Erstellung des Landmarkenmusters wird in Abbildung 6.5 dargestellt. Zunächst werden die Ebenensegmente ebenfalls in lexikographischer Ordnung nach ihrem Mittelpunkt sortiert, welcher sich aus dem Mittelwert der zwei äußeren Punkte der jeweiligen Ebene ergibt. Die Mittelpunkte werden anschließend auf die gleiche Weise wie die Stangenpunkte in Brenner (2009b) zum Deskriptor hinzugefügt. Zusätzlich wird jedoch noch die Orientierung der Ebene gespeichert, welche sich aus dem Winkel zwischen dem Normalenvektor der Ebene im lokalen Landmarkenmuster-Koordinatensystem und dessen x -Achse ergibt:

$$\alpha = \arccos\left(\frac{n_E n_x}{|n_1| |n_x|}\right) \quad (6.4)$$

mit

$$n_x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (6.5)$$

Durch das Hinzufügen der Ebenen ergibt sich die Dimension des Deskriptors somit zu $\dim = 2 \cdot k_{\text{Stangen}} + 3 \cdot k_{\text{Ebenen}} - 3$.

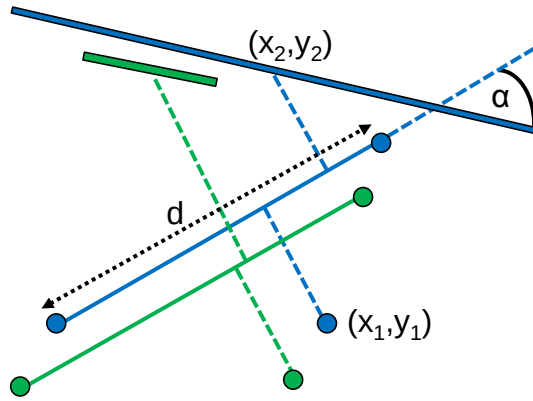


Abbildung 6.5.: Die Erstellung eines Landmarkenmusters für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 1$. Die Referenzstangen und -ebenen sind blau eingefärbt, die gemessenen Landmarken grün. Die Verbindungen zwischen den Landmarken beschreiben die einzelnen Deskriptorwerte. Der hier resultierende Referenzdeskriptor (blau) würde sich zu $D\{d, x_1, y_1, x_2, y_2, \alpha\}$ ergeben.

Alternativ zum Ebenenmittelpunkt könnte auch die minimale Distanz der Ebene zum Ursprung des lokalen Koordinatensystems d_E verwendet werden, welche sich direkt aus den Ebenenparametern in der Hesseschen Normalform ergibt. Hierdurch könnte die Dimension zu $2 \cdot k_{\text{Stangen}} + 2 \cdot k_{\text{Ebenen}} - 3$ reduziert werden. Allerdings würde der Deskriptor somit auch weniger einheitlich sein. Häufig extrahierte Ebenen sind Häuserfassaden, die wiederum häufig im selben Abstand parallel zur Straße stehen und somit sehr ähnliche Ebenenparameter d_E und n_E besitzen. Weiterhin ist die Genauigkeit dieses Wertes stark beeinflusst durch die Genauigkeit der Ebenenorientierung.

Zuordnung

In Brenner (2009b) werden für jede Referenzlandmarke p_i alle möglichen Kombinationen von $k - 1$ Landmarken in der Umgebung L_i von i ausgewählt. Für diese Landmarken wird der Deskriptor D berechnet und in einer Datenbank gespeichert. Die Zuordnung der Landmarken geschieht anschließend wie folgt. Abbildung 6.6 zeigt alle lokalen Referenzdeskriptoren aus P_i im 3D-Raum. Wenn für eine gegebene Szene mindestens k Landmarken l_j detektiert wurden, kann der Deskriptor D_j aus k zufällig gewählten Landmarken bestimmt werden. Eine Menge hierzu passender Lösungen der korrekten Landmarken-Korrespondenzen kann aus der Datenbank abgegriffen werden. Im

nächsten Schritt wird erneut ein Deskriptor aus wiederum zufällig gewählten Landmarken bestimmt und die Überschneidungsmenge aus der aktuellen Ausgabe der Datenbank mit den bisherigen gewählten Landmarken-Korrespondenzen gebildet. Dieses Vorgehen wird so lange wiederholt, bis nur noch eine Lösung möglich ist. Ein Referenzdeskriptor wird bei diesem Verfahren als mögliche Lösung anerkannt, wenn dessen Werte nahezu identisch mit denen des aktuell zufällig bestimmten Deskriptors sind. Dies ist der Fall, wenn die Wertdifferenzen der einzelnen Deskriptorelemente geringer als die zugehörigen Genauigkeiten der Messgenauigkeitsfaktoren e_{Stangen} , $e_{x/y,\text{Ebenen}}$ und $e_{\alpha,\text{Ebenen}}$ der jeweiligen Landmarken sind. Die Messgenauigkeit stangenförmiger Objekte e_{Stangen} kann, wie in Brenner (2009b) vorgeschlagen, auf wenige Zentimeter gesetzt werden. Der Mittelpunkt der Ebenen wird im Gegensatz zum Mittelpunkt der Stangen nicht nur von der Messgenauigkeit beeinflusst. Es ist nicht garantiert, dass eine Ebenen in ihrer gesamten Ausdehnung gemessen wurde. Verdeckungen können dazu führen, dass die Außenpunkte und somit auch der Mittelpunkt der Ebene falsch bestimmt werden (siehe Abbildung 6.5). Aufgrund dessen wird in Abschnitt 7.1.3 für die Messgenauigkeit der Ebenen e_{Ebenen} ein deutlich höherer Wert von 10 m gewählt.

Der in Brenner (2009b) vorgestellte Ansatz hat zum Nachteil, dass fehlerhafte Landmarkendetektionen dazu führen können, dass die korrekte Zuordnung nicht gefunden wird. Tritt in einer Iteration eine fehlerhafte Detektion auf, so ist es wahrscheinlich, dass hierdurch die korrekte Lösung verworfen wird. Aus diesem Grund wird der Ablauf wie folgt geändert. Für jede mögliche sich aus der Datenbank ergebende Referenzlösung des aus k zufällig gewählten Landmarken bestimmten Deskriptors D_j wird eine Translation T_k bestimmt und in einer Liste gespeichert. Diese Translation ergibt sich aus den gemittelten 2D-Koordinatendifferenzen der aktuellen Landmarken (Mittelpunkte der stangenförmigen Objekte) zu den entsprechenden Referenzlandmarken. Dieser Schritt wird iterativ wiederholt. Für jede Translation wird außerdem ein Score s bestimmt. Beinhaltet die Liste bereits eine übereinstimmende Translation T_j , so werden die Werte der Translation gemittelt und nach ihrem jeweiligen Score gewichtet, der anschließend um +1 erhöht wird. In dieser Arbeit wird angenommen, dass zwei Translationen T_k und T_j übereinstimmen, sofern ein gewisser Grenzwert $\Delta T_{\max}$ nicht überschritten wird, hier 0,5 m. Eine finale Translation wird angenommen, sobald eine einheitliche Translation bestimmt wurde. Dies ist in dieser Arbeit der Fall, wenn alle möglichen Kombinationen durchlaufen sind oder der Score der bestbewerteten Translation den Wert der

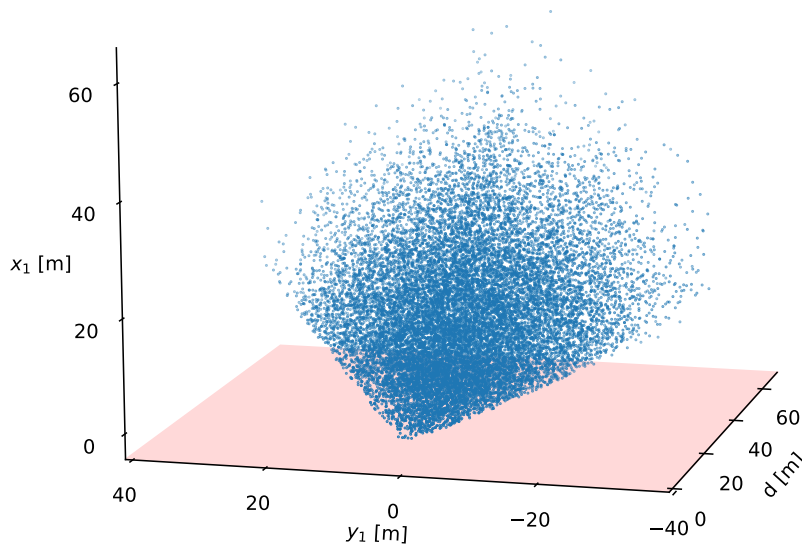


Abbildung 6.6.: 3D-Darstellung aller lokalen Referenzdeskriptoren eines Testgebietes für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 0$.

Translation mit der zweithöchsten Bewertung um $\Delta s_{final} = 5$ übersteigt. Es ist anzumerken, dass dieser Wert für die aktuellen Daten (siehe Abschnitt 7.1.3) passend gewählt wurde. Sind die Landmarken schlecht verteilt, so steigt die Wahrscheinlichkeit einer fehlerhaften Deskriptor-Zuordnung. In diesem Fall sollte Δs_{final} erhöht werden.

Die Erstellung der aus den Referenz-Landmarkenmuster bestehenden Datenbank muss grundsätzlich nicht während des Lokalisierungsschrittes erfolgen und ist in diesem Fall nicht zeitkritisch. Allerdings würde die Datenbank gegebenenfalls viel Speicherplatz benötigen. Um dieses Problem zu umgehen, werden in dieser Arbeit nur Landmarken in einem Umkreis von 50 m zur Erstellung der Referenz-Datenbank in jeden Lokalisierungsschritt verwendet. Die geringe Anzahl an Landmarken führt dazu, dass dieser Schritt in Echtzeit erfolgen kann.

6.2. Sequenzanalyse zur globalen Lokalisierung

Liegen keine Daten eines globalen Verfahrens zur Positionsbestimmung, wie GNSS, vor, so muss die Lokalisierung über ein zusätzliches System mit einer hohen Zuverlässigkeit erfolgen. In diesem Abschnitt wird ein Verfahren vorgestellt, welches LiDAR-Daten eines Automotive-Laserscanners verwendet, um durch den Abgleich mit einer Referenzkarte die Position in longitudinaler Richtung (in Fahrtrichtung) in urbanen Gebieten zu bestimmen. Der Ablauf der Methode wird in Abbildung 6.7 dargestellt. Die grundlegende Idee hierbei ist, dass jeder vollständige Scan eines Laserscanners durch ein Clustering-Verfahren mit einem Label versehen wird. Über eine Sequenzanalyse, in der die zuletzt erfolgten und mit Labels versehenen Scans mit einer Referenztrajektorie verglichen werden, wird anschließend die Position ermittelt. Die aktuelle Sequenz zum Zeitpunkt der Lokalisierung wird fortan als Online-Sequenz bezeichnet. Zusätzlich wird ein neuronales Netz in Form eines Autoencoders verwendet, um die einzelnen Scans der Referenz- und Online-Daten durch repräsentative Merkmale zu beschreiben. Die Methode wird auf den simulierten Daten eines Einzeilen-Laserscanners (Abschnitt 4.1), auf den Daten eines SICK LMS151 des Oxford Robotcars (Abschnitt 4.3) sowie auf den Daten eines Velodyne VLP-16 (Abschnitt 4.4) angewendet.

In allen drei Datensätzen ist der Laserscanner vertikal montiert, sodass mit jedem Scan ein Profil des Straßenraums um das Fahrzeug erstellt wird. Dies hat im Vergleich zu einem horizontal montierten Laserscanner den Vorteil, dass der Sichtbereich nicht auf einzelne Objekte beschränkt ist und durch die Bewegung des Fahrzeuges ein Großteil des 3D-Raums erfasst wird. Ein Beispiel eines solchen Scans für eine Scanzeile ist in Abbildung 6.8a dargestellt.

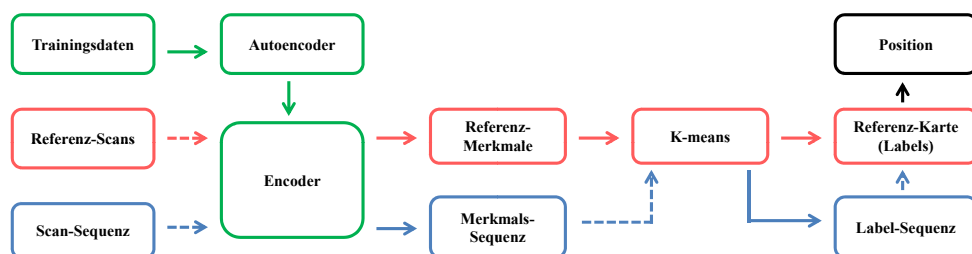


Abbildung 6.7.: Ablauf der Sequenzanalyse zur Bestimmung der globalen Positionen. Mit Hilfe eines Autoencoders werden die Laserscans zunächst durch Merkmale beschrieben. Diese Merkmale werden verwendet, um jedem Scan mit Hilfe eines K-Means-Algorithmus ein Label zuzuweisen. Durch einen Abgleich der Labels einer Referenztrajektorie und der aus den aktuellen Messungen bestehenden Sequenz wird anschließend die Position bestimmt.

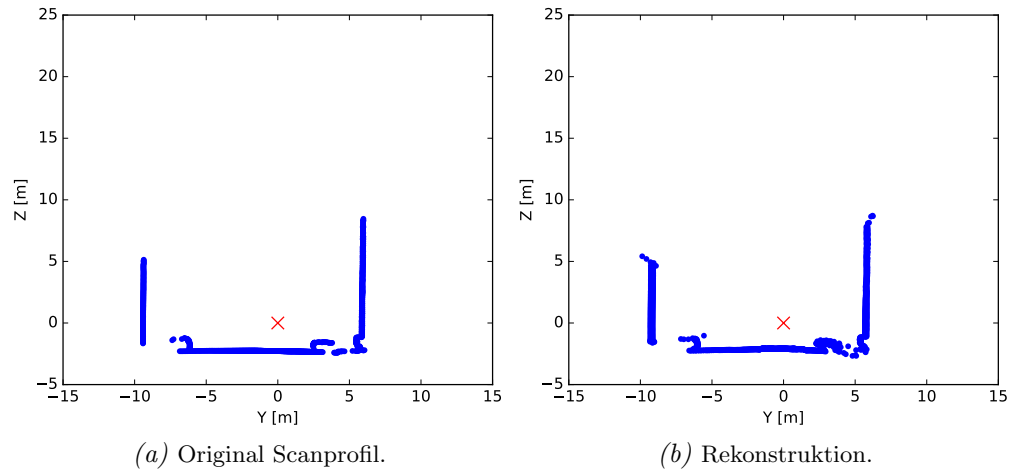


Abbildung 6.8.: Aus Mobile-Mapping-Daten erstelltes Scanprofil und das Ergebnis der Rekonstruktion durch den Decoder eines Autoencoders.

6.2.1. Reduzierung der Dimension durch einen Autoencoder

Die Dimensionsreduktion durch einen Autoencoder bewirkt, dass der Informationsgehalt der jeweiligen Merkmale im Vergleich zu den einzelnen Punktmessungen der Scans steigt und der Speicherbedarf weiter sinkt. Weiterhin sind die bestimmten Merkmale weniger abhängig von der latitudinalen Position des Fahrzeuges, sodass bestenfalls auch bei einer veränderten Fahrspur der aktuelle Scan mit demselben Label versehen wird, wie die Referenzmessung an der entsprechenden Position.

Abbildung 6.8b zeigt ein durch den in Abbildung 6.9 skizzierten Autoencoder rekonstruiertes Scanprofil. Das Netzwerk besteht aus mehreren Convolutional Layern, welche eine bestimmte Anzahl an Merkmalskarten für jeden Eingangswert erstellen. Im Anschluss an jede Convolution findet hierbei ein Max Pooling statt. In den folgenden fully connected Layern erfolgt bis zum letzten Layer des Encoders eine Reduktion der Dimension. Die n_{Merkmale} Werte des entsprechenden Layers gehen anschließend als Eingangswerte in den Decoder ein, in dem die fully connected Layer sowie die folgenden Deconvolutions wiederum die Dimension erhöhen und die Eingangsdaten rekonstruieren. Dieser Aufbau hat zur Folge, dass die Werte des letzten Layers des Encoders eines trainierten Netzwerks den Scan durch möglichst aussagekräftige Werte beschreiben. Aus diesem Grund wird dieser Vektor auch als Merkmalsvektor in den folgenden Schritten der Lokalisierung verwendet.

In Abschnitt 7.2 werden die Parameter des Netzwerks, wie n_{Merkmale} , die Anzahl und Größe der Layer, die Kernelgröße und die Anzahl der Merkmalskarten, für die verschiedenen Datensätze variiert und hinsichtlich ihrer Lokalisierungs-Ergebnisse bewertet.

6.2.2. Clustering der Daten

Jeder Standpunkt, also jedes Element der Sequenz, wird mit einem Label versehen. Als Clustering-Verfahren wird hierbei ein K-Means-Algorithmus verwendet, wobei die Referenzdaten zur Bestimmung der Cluster-Mittelpunkte dienen. Das Clustering hat zur Folge, dass das Verfahren weniger empfindlich gegenüber Veränderungen in den Daten ist. Zusätzlich werden hierdurch der Speicherbedarf und die Laufzeit des Verfahrens reduziert. Bei einem Verzicht auf das Clustering müsste hingegen statt eines einzelnen Wertes pro Scan eine Ähnlichkeit aus allen Elementen des Merkmalsvektors bestimmt werden. Die geforderte Ähnlichkeit der Scans mit selbem Label wird durch den

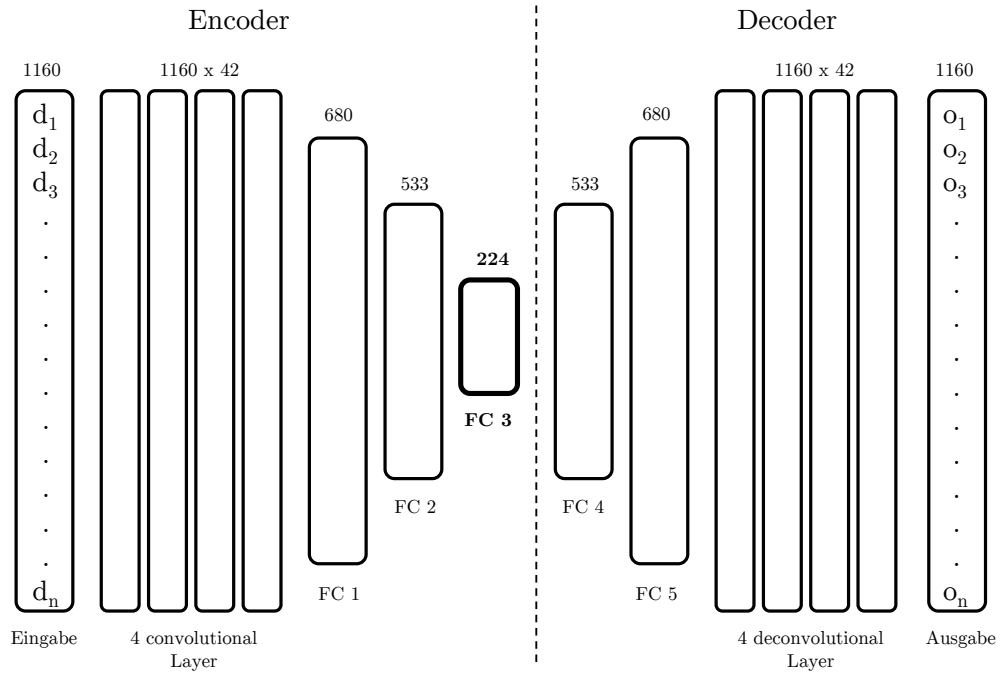


Abbildung 6.9.: Beispielhafter Aufbau des Autoencoders für die aus den Mobile-Mapping-Daten erstellten Automotive-Laserscans mit einer Scanzeile. Auf die Eingangsdaten werden vier Faltungen angewendet. Anschließend wird die Dimension in den folgenden fully connected Layern in diesem Fall auf 224 Merkmale reduziert. In den folgenden Layern des Decoders werden die ursprünglichen Eingangsdaten rekonstruiert.

Parameter k des K-Means-Verfahrens beeinflusst. Eine hohe Anzahl an Clustern bewirkt, dass eine Zuordnung zweier unterschiedlicher Scans zum selben Cluster-Mittelpunkt weniger wahrscheinlich ist. Ein geringerer Wert für k wirkt sich hingegen positiv auf die Robustheit gegenüber kleinen Veränderungen in den Scans an derselben Position aus. In Abschnitt 7.2 werden die Ergebnisse des Verfahrens mit unterschiedlichen Werten für k ausgewertet.

6.2.3. Bestimmung der Position

Die Position des Fahrzeuges wird durch den Abgleich der Online-Sequenz mit einer Referenzsequenz erzeugt. Der Ablauf des Verfahrens wird in Abbildung 6.10 dargestellt. Die Referenztrajektorie enthält eine Abfolge von Positionen, wobei jede Position dem entsprechend zuvor bestimmten Label des zugehörigen Scans zugeordnet ist. Die Endposition der Referenzsequenz mit der geringsten Distanz zur aktuellen Sequenz wird schließlich als Ergebnis der Positionsbestimmung verwendet. Zur Bestimmung der Distanz zwischen zwei Scans wird die Hamming-Distanz (Hamming, 1950) gewählt. Hierbei wird die Distanz zwischen zwei Sequenzen aus der Anzahl an ungleichen Paaren zweier Sequenzen derselben Länge bestimmt. Die Länge n_{Seq} der aktuellen Sequenz hat hierbei Einfluss auf die Ergebnisse und wird in Abschnitt 7.2 variiert.

Da die Distanzen zwischen zwei Messungen für die Daten des Oxford Robotcars (Abschnitt 4.3) sowie des Velodyne VLP-16 (Abschnitt 4.4) nicht konstant sondern abhängig von der jeweiligen Geschwindigkeit sind, müssen diese entsprechend angepasst werden. Hierzu findet nach der Bestimmung der Labels der einzelnen Scans eine Interpolation der Scanpositionen mit konstanten Distanzen zwischen den Messungen statt, wobei den interpolierten Positionen die Labels der jeweils nächstliegenden ursprünglichen Scans zugeordnet werden. Es wird jeweils das Label übernommen, dessen euklidische Distanz zur aktuell interpolierten Position am geringsten ist. In diesem Fall wird

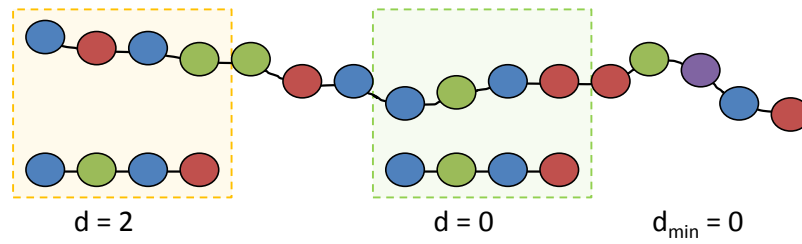


Abbildung 6.10.: Ablauf der Sequenzanalyse. Jeder Merkmalsvektor der Referenzkarte (oben) und der aktuellen Sequenz (unten) wird durch ein Clusterverfahren mit insgesamt k unterschiedlichen Labels versehen, welche hier blau, rot, grün und lila eingefärbt sind. Durch einen Sliding-Window-Ansatz wird anschließend eine punktweise Distanzbestimmung durchgeführt und verwendet, um die geringsten Kosten zwischen Sequenz und Referenz zu bestimmen.

vorausgesetzt, dass die Trajektorie bekannt ist. Für den Fall der Online-Messungen kann jedoch die Entfernung zwischen zwei Scans aus weiteren Sensoren, wie einer Inertialeinheit oder der Odometrie des Fahrzeuges, bestimmt werden.

6.3. Scanbildkorrelation zur hochgenauen Lokalisierung

In Abschnitt 6.1 werden einzelne aus den LiDAR-Daten extrahierte Landmarken zur Lokalisierung verwendet. Folglich wird nur ein geringer Anteil der Messungen der Laserscanner betrachtet, während ein Großteil der Messungen unbeachtet bleibt. In diesem Abschnitt soll eine Referenzkarte erstellt werden, die aus Scanbildern besteht, in denen die Intensitäts- und Höhenwerte aller statischen 3D-Punkte als Raster in der x/y -Ebene eingetragen sind.

6.3.1. Automotive Daten eines Einzeilenscanners

Im Falle der aus den Messungen des Riegl VMX-250 erstellten Automotive-Daten werden zunächst durch eine Änderungsdetektion (Abschnitt 5.1) und eine Klassifizierung (Abschnitt 5.2.1) dynamische Objekte, wie Fahrzeuge oder Bäume, detektiert und aus den Daten entfernt. Somit soll gewährleistet werden, dass die Referenzkarte keine Objekte enthält, die zum Zeitpunkt der Online-Messung nicht, an einer unterschiedlichen Position oder in unterschiedlicher Form auftreten. Zur Erstellung der Intensitäts- und Höhenbilder wird die Punktwolke auf die x/y -Ebene mit einer Zellenbreite von 0,02 m aufgerastert. Die Intensitätswerte eines Pixels ergeben sich aus den gemittelten Intensitätswerten aller Punkte, die in die entsprechende Zelle fallen. Im Höhenbild ergibt sich der Grauwert eines Pixels aus dem normalisierten Höhenwert des Punktes mit der maximalen Höhe in der jeweiligen Zelle. Die Größe der Referenzbilder beträgt in dieser Arbeit 4000×4000 Pixel, was einer Ausdehnung von 80×80 m entspricht. Um die Punktdichte zu erhöhen, werden die Bilder aus den Punktmengen acht unterschiedlicher Messfahrten gewonnen. Abbildung 6.11 und 6.12 zeigen die sich ergebenden Scanbilder unter Verwendung eines einzelnen sowie acht Datensätzen. Es ist zu erkennen, dass deutlich weniger nicht belegte, in dieser Darstellung schwarze, Pixel auftreten und Regionen, welche in einigen Datensätzen von Objekten wie Fahrzeugen verdeckt sind, ebenfalls ausgefüllt sind.

Zu jedem Bild wird eine Referenzkoordinate der entsprechenden Trajektorie gespeichert. Die zugehörige Bildkoordinaten betragen $px = 4000$ und $py = 2000$. Zusätzlich wurden die Punkte zuvor durch den aktuellen Gier-Winkel des Fahrzeuges rotiert. In dieser Arbeit werden entlang einer Referenztrajektorie alle 10 m Referenzbilder erstellt.

Die Lokalisierung des Fahrzeuges geschieht durch eine Korrelation der aktuellen Messungen des Automotive-Scanners mit den Referenzbildern. Hierfür muss die Position hinreichend genau bekannt sein, in Abhängigkeit der Distanzen zweier aufeinanderfolgender Referenzbilder. In dieser Arbeit werden die Intensitäts- und Höhenbilder, deren Distanz zur aktuell angenommenen Position am geringsten ist, gewählt, wobei ebenfalls die Ausrichtung des Fahrzeuges betrachtet wird. Zusätzlich werden die entlang der Referenztrajektorie folgenden Referenzbilder betrachtet.

Zur Bestimmung der Korrelation zwischen den aktuellen Messungen und den Referenzen werden die Messungen verwendet, welche entlang der zuletzt zurückgelegten 20 m aufgenommen wurden. Vor der Berechnung der Korrelation werden in den entsprechenden Laserscans dynamische Objekte nach den in Abschnitt 5.2.2 und 5.2.3 vorgestellten Methoden extrahiert. Die Korrelation der Messungen zu den Bildern wird durch den Korrelationskoeffizienten C_j beschrieben, welcher sich für ein einzelnes Bild aus

$$C_j = \sum_{i=1}^n \frac{(m_{j,i} - \overline{m_j})(px_{j,i} - \overline{px_j})}{\sqrt{\sum_{i=1}^n (m_{j,i} - \overline{m_j})^2 \sum_{i=1}^n (px_{j,i} - \overline{px_j})^2}}, \text{ für } j = 1, 2 \quad (6.6)$$

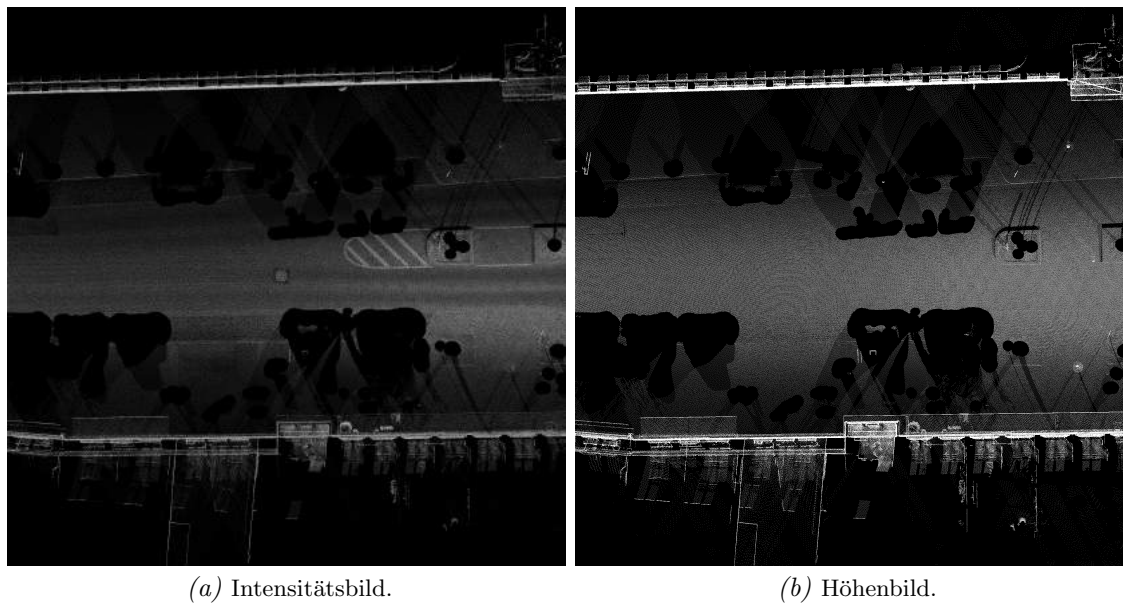


Abbildung 6.11.: Beispiel eines aus einem Datensatz erstellten Intensitäts- und Höhenbildes bei einer Zellenbreite von 0,02 m.

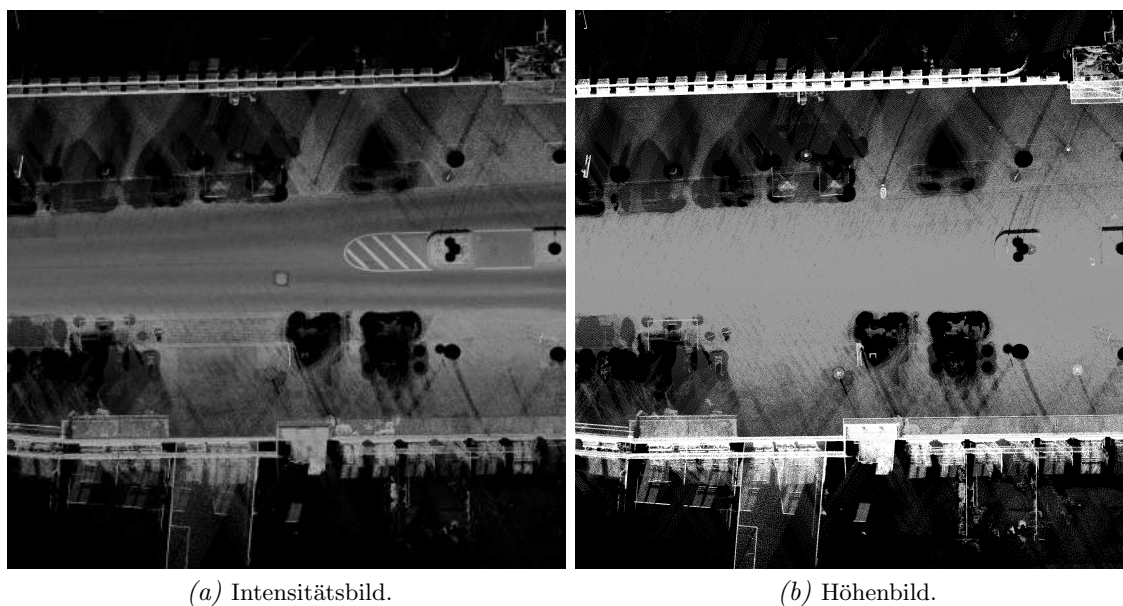


Abbildung 6.12.: Beispiel eines aus acht Datensätzen erstellten Intensitäts- und Höhenbildes bei einer Zellenbreite von 0,02 m.

ergibt. Hierbei bezeichnen $m_{i,j}$ die einzelnen Messungen und $px_{i,j}$ die Bildwerte an den zugehörigen Bildkoordinaten der Referenzbilder, $\overline{m_j}$ und $\overline{px_j}$ die entsprechenden Mittelwerte. Die Gesamtkorrelation C wird durch

$$C = C_1 \cdot C_2 \quad (6.7)$$

bestimmt. Treten in einem Gebiet keine signifikanten Höhenvariationen auf, so wird in diesem Fall C lediglich aus den Intensitätswerten bestimmt.

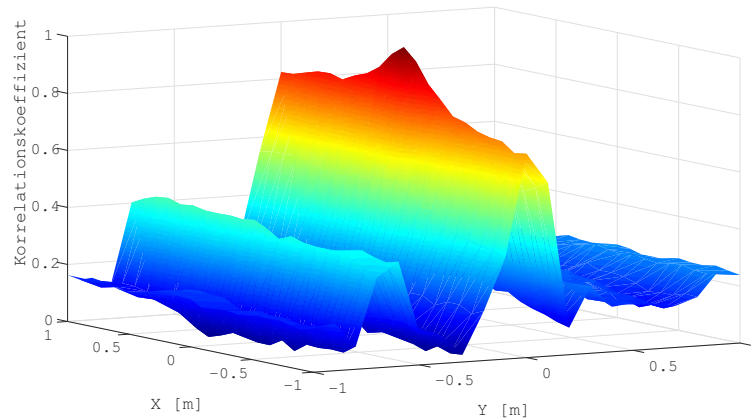


Abbildung 6.13.: Korrelation zwischen den Messungen des Laserscanners und den Referenzbildern in einem Bereich von $2 \times 2\text{m}$ bei einer Auflösung von $0,1\text{ m}$.

Die Lokalisierung erfolgt, indem die angenommene Pose, welche durch x , y und θ definiert ist, um die durch dx , dy und $d\theta$ definierte Transformation T_C verschoben, rotiert und für jede sich ergebene Pose C bestimmt wird. Die Transformation T_{max} , welche den höchsten Korrelationskoeffizienten C_{max} ergibt, wird anschließend auf die aktuelle Pose angewendet. Um die Rechenzeit zu verringern, wird ein Pyramiden-Ersatz verwendet, in dem dx , dy und $d\theta$ stetig reduziert werden. Ein beispielhaftes Ergebnis einer Korrelationsanalyse für die in Abbildung 6.12 gezeigten Referenzbilder ist in Abbildung 6.13 dargestellt. Es ist zu erkennen, dass die Werte in Fahrtrichtung (dx) für $dy = 0$ vergleichsweise hoch sind. Orthogonal zur Fahrtrichtung tritt jedoch nur ein einzelner hoher Peak auf. Ursache für dieses Verhalten sind in diesem Fall die Häuserfassaden, welche parallel zur Fahrtrichtung verlaufen.

6.3.2. Verwendung der Daten des Velodyne VLP-16

Für eine Korrelation der Punktmessungen eines Automotive-Laserscanners mit den Referenz-Scanbildern muss die Orientierung des Sensors mit einer hohen Genauigkeit bekannt sein. Abbildung 6.14 zeigt die auf die x/y -Ebene projizierten Punkte, welche von dem im Abschnitt 4.4 vorgestelltem Velodyne VLP-16 aufgenommen wurden, wobei die Orientierung des Scanners durch die Messungen der Xsens MTi-G durchgeführt wurde.

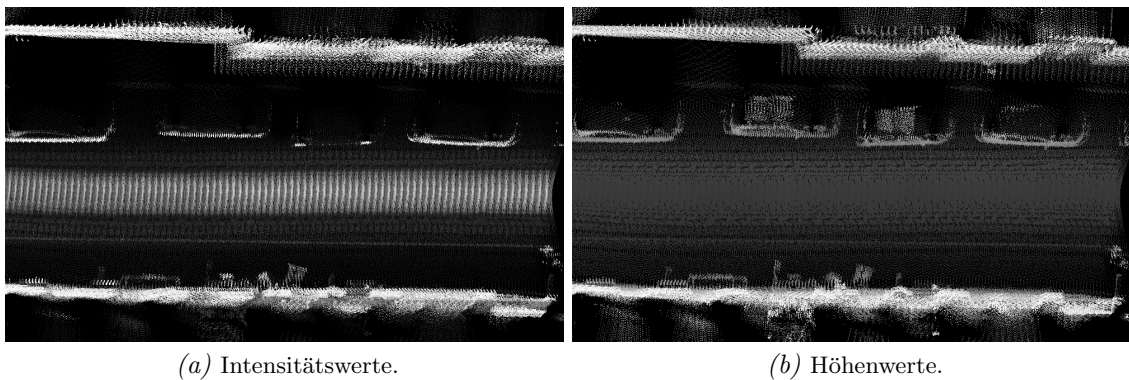


Abbildung 6.14.: Auf die x/y -Ebene projizierte Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Xsens MTi-G bei einer Zellenbreite von $0,02\text{ m}$.

Durch die ungenaue Orientierung des Laserscanners, in diesem Fall im Roll-Winkel, beträgt die Breite der Häuserfassaden hier 48 cm (24 Pixel). Diese Ungenauigkeit kann durch die Verwendung der Orientierung aus den Messungen der Inertialeinheit des Riegl VMX-250 deutlich verringert werden (siehe Abbildung 6.15). Hier beträgt die Breite 20 cm (10 Pixel).

Weiterhin fällt bei Betrachtung der Intensitätswerte auf, dass in dem Bereich der Fahrzeugtrajektorie die Intensitätswerte vergleichsweise hoch sind. Eine Erklärung könnte die hier geringe Messdistanz von etwa 0,5 m sein.

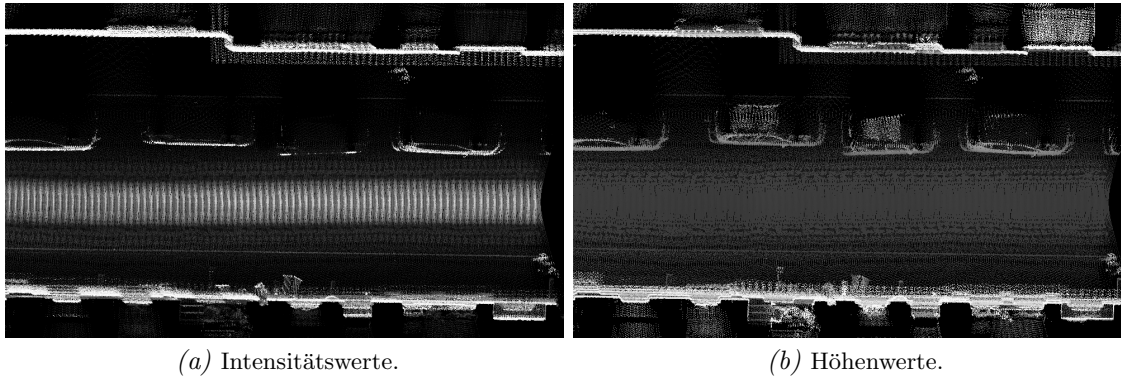


Abbildung 6.15.: Auf die x/y -Ebene projizierte Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Inertialeinheit des Riegl VMX-250 bei einer Zellenbreite von 0,02 m.

Da bei einer falschen Orientierung im Roll- oder Nick-Winkel die Abweichungen mit der Höhe zunehmen, werden zusätzlich nur Punkte mit einer Höhe von maximal 3 m über der Straßenebene verwendet, wodurch die Fassadenbreite lediglich maximal 8 cm (4 Pixel) beträgt (siehe Abbildung 6.16).

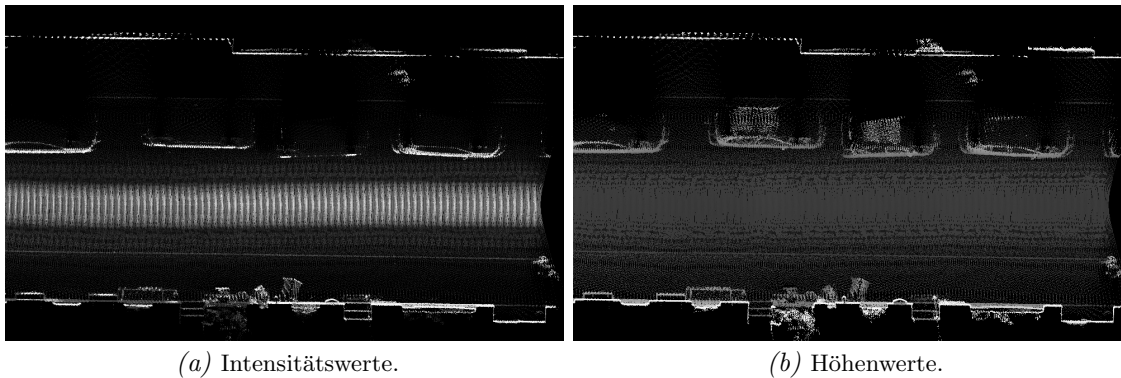


Abbildung 6.16.: Auf die x/y -Ebene projizierte Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Inertialeinheit des Riegl VMX-250 bei einer maximalen Höhe von 3 m über der Straßenebene und einer Zellenbreite von 0,02 m.

In Abschnitt 5.2.3 wird die Klassifizierung dynamischer Punkte in den Scanbildern des Velodyne VLP-16 beschrieben. Durch die Klassifizierung konnte ein Recall-Wert von 58 erzielt werden. Dies bedeutet, dass weiterhin 42 % der dynamischen Punkte nicht detektiert werden können. Aus diesem Grund werden Punkte, welche in der x/y -Ebene in durch den Radius r_{dyn} bestimmten Umkreis

einen dynamischen Punkt aufweisen, ebenfalls als dynamisch markiert. Der vergleichsweise geringen Erfassungsrate dynamischer Punkte wirkt entgegen, dass sich in Abhängigkeit der Fahrtgeschwindigkeit zwei aufeinanderfolgende Scans überlappen und somit bestenfalls mehrere Messungen und somit klassifizierte Daten desselben Bereichs vorliegen. Die hieraus resultierenden Scanbilder sind in Abbildung 6.17 aufgeführt. Es zeigt sich, dass Fahrzeuge weitestgehend entfernt werden können. Lediglich Teile der Dächer der Fahrzeuge sind noch in den Scanbildern enthalten. Weiterhin werden im Straßenbereich sowie zu einem geringen Teil an den Häuserfassaden statische Punkte nicht berücksichtigt.

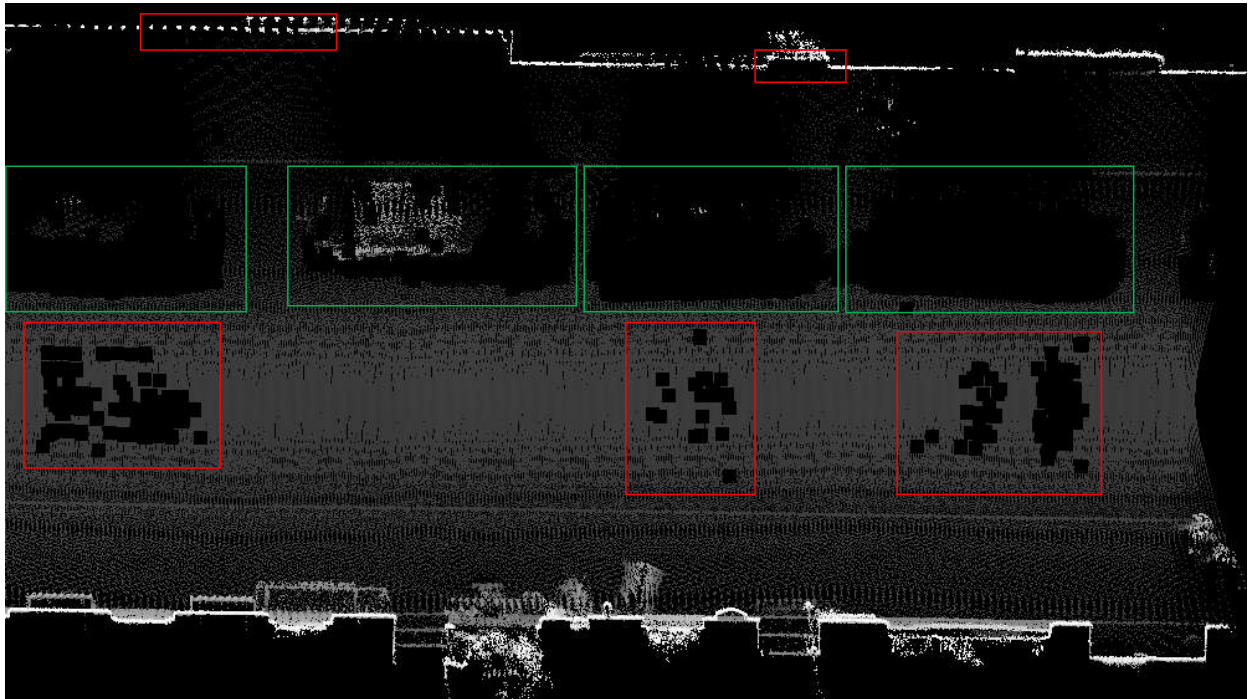


Abbildung 6.17.: Höhenwerte der auf die x/y -Ebene projizierten statischen Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Inertialeinheit des Riegl VMX-250 bei einer maximalen Höhe von 3m über der Straßenebene und einer Zellenbreite von 0,02m. Bereiche mit korrekterweise entfernten dynamischen Punkten sind grün, Bereiche in denen fälschlicherweise Punkte entfernt werden rot markiert.

7. Ergebnisse der Fahrzeug-Lokalisierung

Im Folgenden werden die Ergebnisse der im vorherigen Abschnitt präsentierten Methoden ausgewertet. Entlang einer bestimmten Test-Trajektorie werden hierfür die Vollständigkeit sowie die Wurzel der mittleren quadratischen Abweichung ($RMSE$) zur Sollposition betrachtet. Weiterhin werden die Ergebnisse in einer Kartendarstellung präsentiert, welche einen Überblick über verschiedene Bereiche gibt, in denen die jeweiligen Verfahren unterschiedliche Werte für die Genauigkeit und Vollständigkeit erzielen.

7.1. Verwendung stangenförmiger Objekte und Ebenen zur Fahrzeug-Lokalisierung

Zur Beurteilung der Vollständigkeit der Lokalisierung über detektierte stangenförmige Objekte und Ebenen (Abschnitt 6.1) wird untersucht, an wie vielen Positionen entlang einer Test-Trajektorie eine Positionsbestimmung möglich ist. Auf die zu erwartenden Genauigkeiten wird geschlossen, indem an den entsprechenden Positionen Fehlerellipsen berechnet werden. Zur Untersuchung dieser Methode werden die Daten eines Valeo Scala sowie eines Velodyne VLP-16 ausgewertet.

Erstellung von Fehlerellipsen

Eine Positionsbestimmung nur unter der Verwendung von Stangen und Ebenen wird als möglich angenommen, sobald sich um Umfeld einer Position mindestens drei detektierte Stangen, zwei Ebenen mit jeweils zwei Eckpunkten oder zwei Stangen und mindestens eine Ebene mit entsprechenden Referenzen befinden. Im Falle des Valeo Scala werden alle Landmarken in einem Bereich Radius von 30 m berücksichtigt (siehe Abbildung 7.1). Für den Velodyne VLP-16 alle Landmarken entlang der zuletzt zurückgelegten 30 m oder alternativ 50 m. Die Fehlerellipsen werden für den Fall, dass ausreichend Landmarken innerhalb dieses Bereichs vorliegen, durch eine Varianzfortpflanzung, wie in Brenner (2009a) beschrieben, bestimmt. Die detektierten Ebenen werden hierbei auf ihre

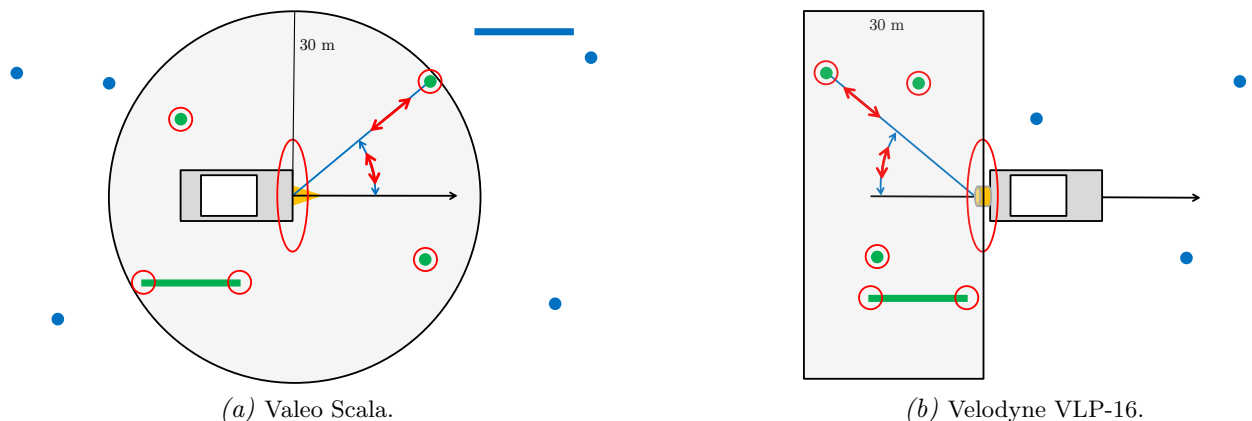


Abbildung 7.1.: Ermittlung der Fehlerellipse (rot) des Standpunktes aus erfolgreich detektierten und zugeordneten stangenförmigen Objekten und Ebenen (grün). In die Berechnung der Ellipsen gehen neben den Landmarkenparametern auch die ebenfalls in rot eingezeichneten Unsicherheiten der Distanz- und Winkelmessungen sowie der Landmarken ein. Für die Messungen des Valeo Scala wird ein Umkreis von 30 m berücksichtigt, für die Daten des Velodyne VLP-16 hier die Landmarken entlang der zuletzt zurückgelegten 30 m.

Ebenenendpunkte reduziert. In die Bestimmung der Fehlerellipsen gehen die Unsicherheiten der Messungen sowie die der Bestimmung der Landmarkenparameter ein (siehe Tabelle 7.1). Für die Genauigkeiten der Distanz- und Winkelmessungen σ_d und σ_α werden die entsprechenden Unsicherheiten der in Abschnitt 4.2 und 4.4 vorgestellten Laserscanner Valeo Scala und Velodyne VLP-16 verwendet. Da in diesem Abschnitt untersucht werden soll, wie genau eine Positionsbestimmung mit vorliegenden, hochgenauen Referenzdaten möglich ist, werden für die Genauigkeiten der extrahierten Stangen- und Ebenenparameter σ_{Stange} und σ_{Ebene} Werte von wenigen Zentimetern angenommen.

Tabelle 7.1.: In die Berechnung der Fehlerellipsen der Lokalisierung durch stangenförmige Objekte und Ebenen eingehende Unsicherheitsfaktoren.

Genauigkeitsfaktor	Standardabweichung
Distanzmessung Valeo Scala ($\sigma_{d,Valeo}$)	10 cm
Winkelmessung Valeo Scala ($\sigma_{\alpha,Valeo}$)	0,25 °
Distanzmessung Velodyne VLP-16 ($\sigma_{d,Velodyne}$)	3 cm
Winkelmessung Velodyne VLP-16 ($\sigma_{\alpha,Velodyne}$)	0,25 °
Stangen (σ_{Stange})	2 cm
Ebenenendpunkte (σ_{Ebene})	5 cm

7.1.1. Valeo Scala

Die zur Auswertung der Lokalisierung verwendeten Daten des horizontal montierten Valeo Scala wurden in einem Testgebiet mit einer Größe von 3 km² im Hannoveraner Stadtteil Badendstedt aufgenommen. Das Gebiet lässt sich in verschiedene Bereiche einteilen, die in Abbildung 7.2 mit den Ziffern 1-5 gekennzeichnet sind. Der Bereich 1 führt entlang einer breiten Hauptstraße, an der zum Teil Fahrzeuge am Seitenstreifen abgestellt sind. Die Bereiche 2 und 3 kennzeichnen Wohngebiete. Im Bereich 4 befindet sich ein Neubaugebiet, im Bereich 5 ein typischer Innenstadtbereich. Die Test-Trajektorie mit einer Länge von 6 km führt durch alle beschriebenen Bereiche des Gebietes.

Im Testgebiet wurden durch das in Brenner (2009a) vorgestellte Verfahren (siehe Abschnitt 2.1.6) über 10 000 stangenförmige Objekte und über 2000 Ebenen aus den LiDAR-Messungen des in Abschnitt 4.1 vorgestellten Riegl VMX-250 Mobile-Mapping-Systems extrahiert. Die extrahierten Landmarken sind in Abbildung 7.2 dargestellt und dienen im Folgenden als Referenzdaten.

Die Extraktion der stangenförmigen Objekte und Ebenen aus den Daten des Automotive-Laserscanners wird wie in Abschnitt 6.1 beschrieben durchgeführt. Die niedrige Einbauhöhe und der geringe vertikale Sichtbereich des Laserscanners (vergleiche Abbildung 4.3) führen dazu, dass häufig nur ein schmaler Bereich vom Sensor erfasst wird, der sich am unteren Ende der jeweiligen Objekte befindet. Dies wird in Abbildung 6.4b (Abschnitt 6.1) deutlich. Diese Einschränkung und das vergleichsweise hohe Rauschen des Laserscanners führen dazu, dass viele irrtümliche Detektionen auftreten. In diesem Fall ergibt sich auf einer Trajektorie mit einer Länge von etwa sechs Kilometern eine False-Positive-Rate von 68 %. Dies bedeutet, dass 68 % der vom Automotive-Scanner detektierten, vermeintlichen Landmarken keiner Referenz zugeordnet werden können. Hierbei wird angenommen, dass aus den Messungen des Mobile-Mapping-Systems alle auftretenden Stangen entlang der Trajektorie extrahiert werden konnten. Die online detektierten Landmarken werden der Referenz mit der geringsten euklidischen Distanz in 2D zugeordnet, wobei eine Maximaldistanz von 1 m angenommen wird. Für die Ebenen werden die Distanzen der Ebenenendpunkte zur Referenzebene sowie der Winkel der Ebenen zueinander betrachtet. Die maximale Distanz der Endpunkte zueinander beträgt 1 m, der maximale Winkel 30°. Insgesamt können entlang der Test-Trajektorie

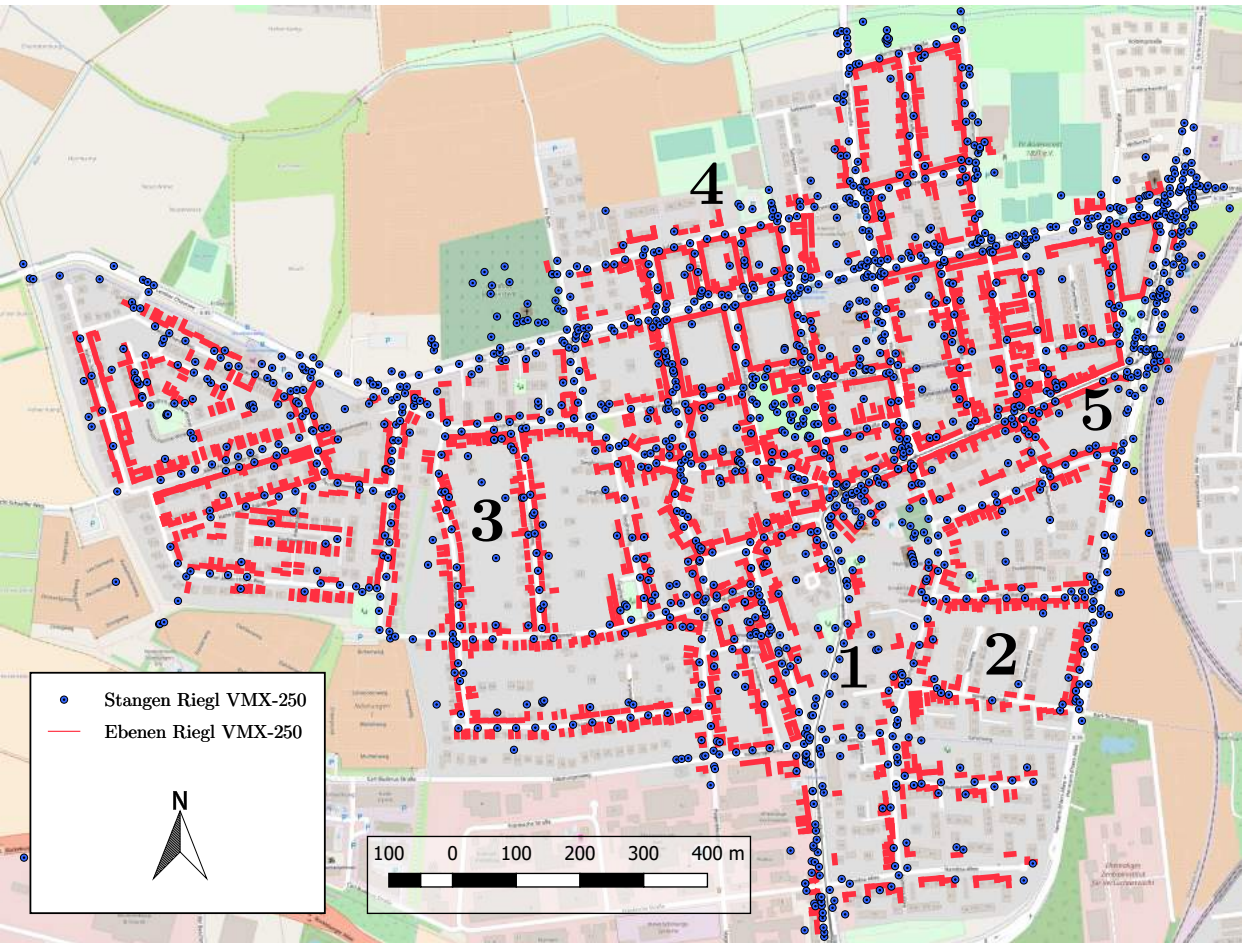


Abbildung 7.2.: Aus den Messungen des Mobile-Mapping-Systems extrahierte Stangen (blau) und Ebenen (rot) in Hannover-Badenstedt (Hintergrundkarte: OpenStreetMap contributors (2018)).

191 Stangen und 77 Ebenen detektiert und einer Referenz zugeordnet werden.

In der Auswertung der Ergebnisse wird zwischen der Fahrzeuglokalisierung mithilfe von ausschließlich detektierten stangenförmigen Objekten, ausschließlich detektierten Ebenen sowie stangenförmigen Objekten und Ebenen unterschieden. Die Anzahl der erfolgreichen Positionsbestimmungen und die jeweiligen mittleren Genauigkeiten der Standpunkte sind in Tabelle 7.2 aufgeführt. Die Fehlerellipsen entlang der Trajektorie werden in Abbildung 7.3 sowie in Abbildung A.1 und A.2 im Anhang dargestellt.

Tabelle 7.2.: Auswertung der erfolgreichen Positionsbestimmung über aus den simulierten Daten eines Einzeilenscanners extrahierte stangenförmige Objekte und Ebenen im Testgebiet.

	Stangen	Ebenen	Stangen und Ebenen
Vollständigkeit	64 %	35 %	74 %
Mittlere Standardabweichung	0,06 m	0,11 m	0,06 m

Tabelle 7.2 zeigt, dass eine Positionsbestimmung unter Verwendung von erfolgreich zugeordneten Stangen in 64 % der Fälle mit einer mittleren Standardabweichung von 0,06 m möglich ist. Werden Ebenen in die Lokalisierung hinzugenommen, steigt diese Rate auf 74 %, bei einer gleichbleibenden Standardabweichung. Werden lediglich segmentierte Ebenen verwendet, so sinkt die Erfolgsrate

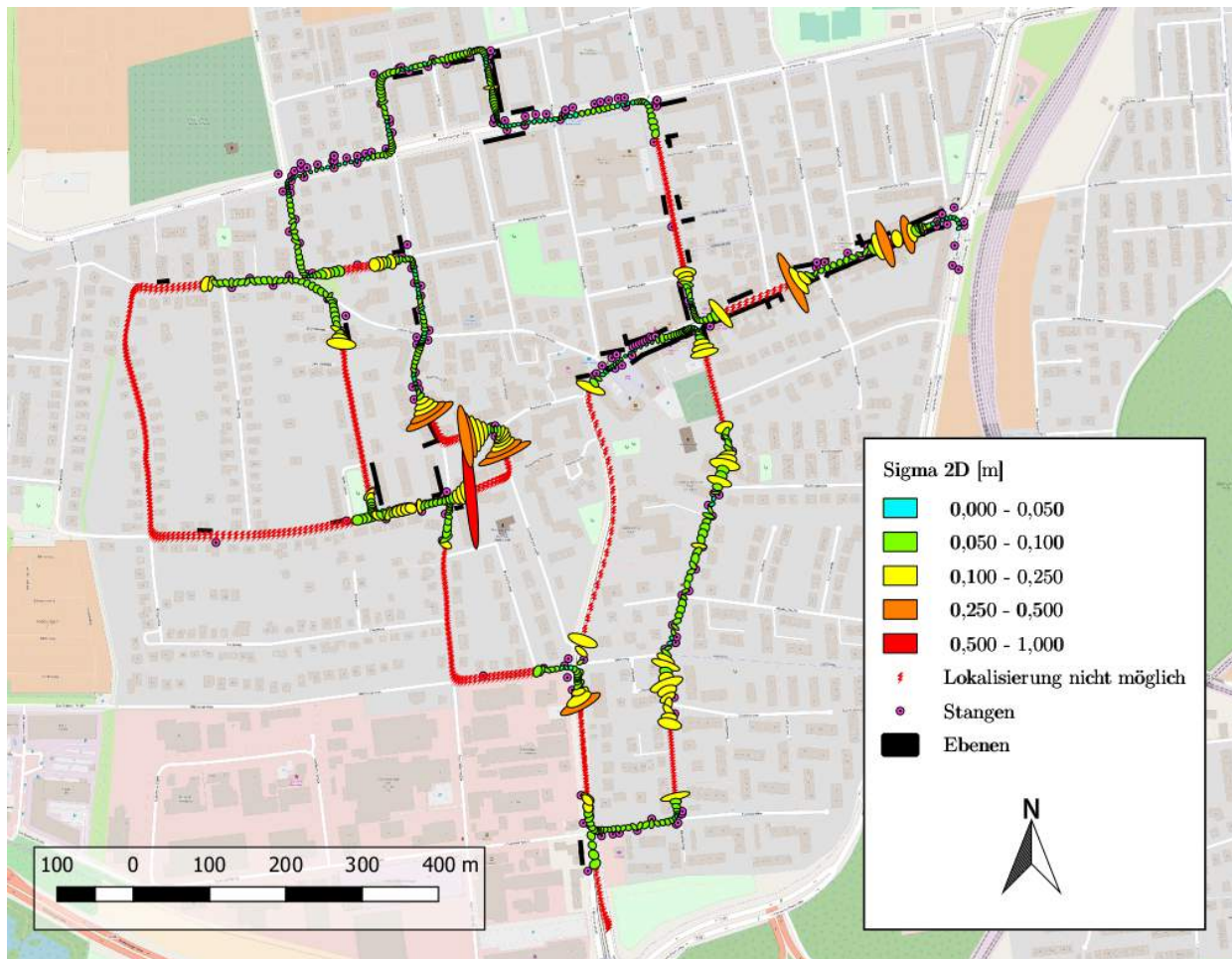
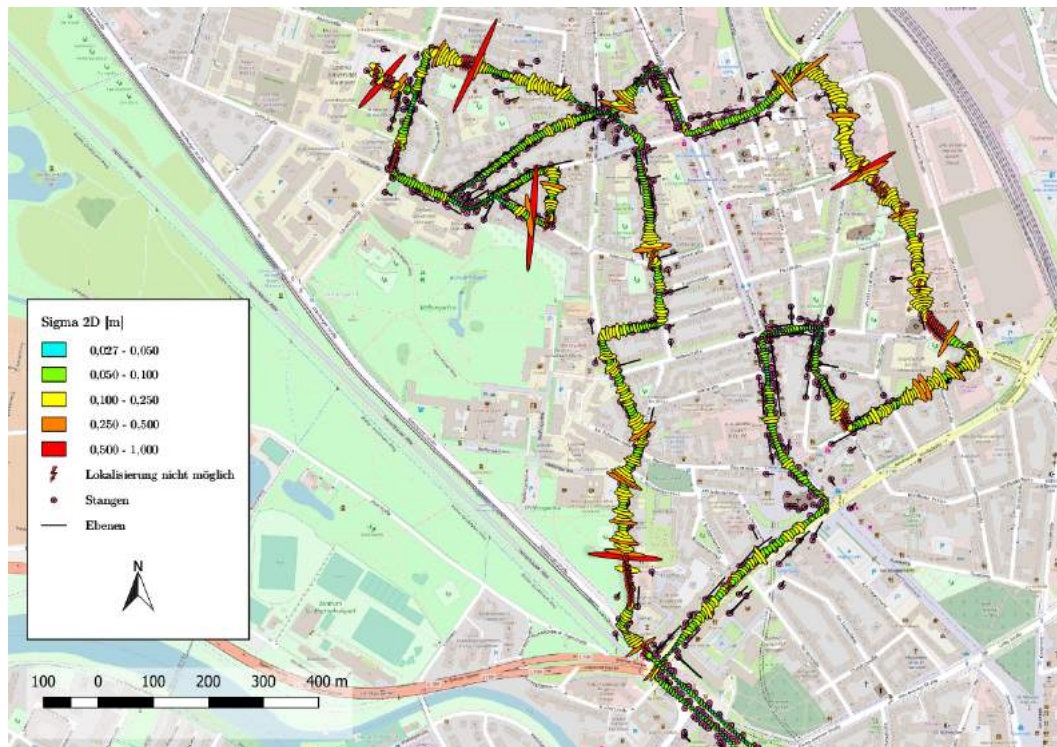


Abbildung 7.3.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der gemeinsamen Verwendung von Stangen und Ebenen, detektiert durch einen horizontal montierten Valeo Scala (Hintergrundkarte: OpenStreetMap contributors (2018)). Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

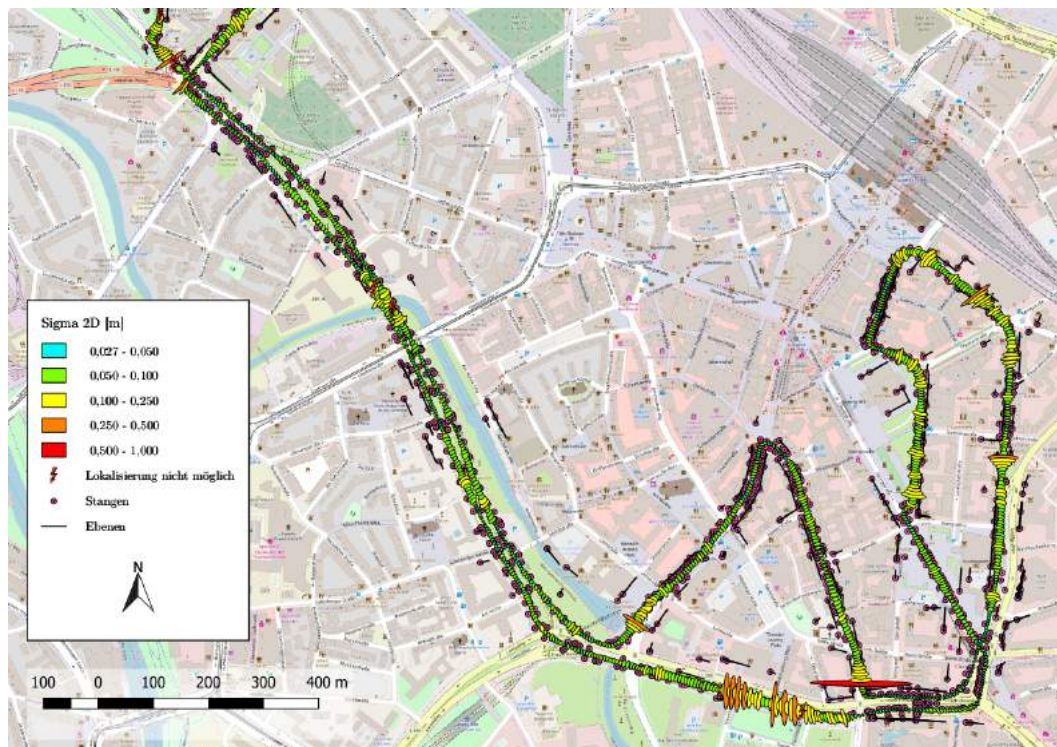
zu 35 %, wobei der mittlere Standardabweichung 0,11 m beträgt. Abbildung 7.3 zeigt, dass eine Positionsbestimmung aus stangenförmigen Objekten und Ebenen insbesondere in den Bereichen 1 und 3 nur selten möglich ist. Der Grund hierfür ist, dass es sich hierbei um Wohngebiete handelt, in denen vornehmlich Einzelhäuser mit durch Hecken oder Zäune begrenzten Grundstücken vorkommen. Dies hat hier zur Folge, dass stangenförmige Objekte, wie beispielsweise Straßenlaternen, verdeckt sind. Durch die niedrige Einbauhöhe sowie den geringen vertikalen Öffnungswinkel des Laserscanners können die nicht verdeckten Bereiche der Stangen ebenso nicht erfasst werden. Somit wird die entsprechende Landmarke in diesem Fall nicht detektiert. Da die Häuserfassaden ebenfalls verdeckt sind oder sich nicht im Sichtbereich des Laserscanners befinden, liegen in diesen Gebieten nur wenige Landmarken vor, die zur Lokalisierung verwendet werden können.

7.1.2. Velodyne VLP-16

Die in Abschnitt 6.1.3 vorgestellte Methode zur Segmentierung stangenförmiger Objekte und Ebenen aus den Daten eines Velodyne VLP-16 wird entlang einer Trajektorie mit einer Länge von 11 km, welche im Norden durch ein Wohngebiet, entlang einer großen Verkehrsstraße und durch einen Innenstadtbereich im Süden führt (siehe Abbildung 7.4), getestet. Zur Berechnung der Fehlerellipsen werden ebenfalls die in Tabelle 7.1 präsentierten Unsicherheitsfaktoren verwendet.



(a) Nördlicher Bereich.



(b) Südlicher Bereich.

Abbildung 7.4.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der gemeinsamen Verwendung von Stangen und Ebenen, detektiert durch einen vertikal montierten Velodyne VLP-16 (Hintergrundkarte: OpenStreetMap contributors (2018)). Die Orientierung des Sensors wird aus den Messungen der Inertialeinheit des Riegl VMX-250 bestimmt, die zurückgelegten Distanz beträgt ($s_{max} = 50$ m). Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

Tabelle 7.3.: Auswertung der erfolgreichen Positionsbestimmung durch aus den Daten eines Velodyne VLP-16 extrahierte stangenförmige Objekte und Ebenen entlang der Trajektorie unter Verwendung der Orientierung aus der Xsens MTi-G.

	Stangen	Ebenen	Stangen und Ebenen
Landmarken	1757	651	2408
Vollständigkeit ($s_{max} = 30\text{ m}$)	67 %	17 %	81 %
Mittlere Standardabweichung ($s_{max} = 30\text{ m}$)	0,12 m	0,13 m	0,10 m
Vollständigkeit ($s_{max} = 50\text{ m}$)	87 %	41 %	95 %
Mittlere Standardabweichung ($s_{max} = 50\text{ m}$)	0,15 m	0,13 m	0,08 m

Tabelle 7.4.: Auswertung der erfolgreichen Positionsbestimmung durch aus den Daten eines Velodyne VLP-16 extrahierte stangenförmige Objekte und Ebenen entlang der Trajektorie unter Verwendung der Orientierung aus der Inertialeinheit des Riegl VMX-250.

	Stangen	Ebenen	Stangen und Ebenen
Landmarken	1849	726	2575
Vollständigkeit ($s_{max} = 30\text{ m}$)	68 %	19 %	84 %
Mittlere Standardabweichung ($s_{max} = 30\text{ m}$)	0,12 m	0,13 m	0,10 m
Vollständigkeit ($s_{max} = 50\text{ m}$)	89 %	47 %	97 %
Mittlere Standardabweichung ($s_{max} = 50\text{ m}$)	0,11 m	0,13 m	0,08 m

Lediglich die Genauigkeit der Distanzmessung des Laserscanners wird auf 3 cm angepasst. Eine Übersicht über alle durch das Referenzsystem (Riegl VMX-250) extrahierten Stangen und Ebenen ist in Abbildung A.3 im Anhang zu finden.

Wie in Abschnitt 4.4.1 gezeigt, können sowohl die Winkelmessungen der Xsens MTi-G als auch die Winkelmessungen der Inertialeinheit des Riegl VMX-250 zur Registrierung der Messungen des Velodyne-Laserscanners verwendet werden. Die Position des Laserscanners zum Zeitpunkt der jeweiligen Messung wird aus den Daten des Riegl VMX-250 gewonnen. Tabelle 7.3 und 7.4 zeigen die resultierenden Lokalisierungsergebnisse. Hierbei werden die stangenförmigen Objekte und Ebenen zur Lokalisierung verwendet, welche in den zuletzt zurückgelegten $s_{max} = 30\text{ m}$ und $s_{max} = 50\text{ m}$ aus den Punktwolken der Messungen segmentiert und erfolgreich einer Referenz zugeordnet wurden. Zur Zuordnung wird, wie in Abschnitt 7.1.1 dargestellt, die nächstliegende Referenzlandmarke verwendet.

Es zeigt sich, dass bei einer Verwendung der aus den Messungen der Xsens MTi-G bestimmten Orientierung 10 % weniger Ebenen segmentiert und erfolgreich zugeordnet werden können. Für stangenförmige Objekte beträgt die Differenz 5 %. Ein wesentlicher Grund für die Unterschiede in den Ergebnissen der verschiedenen Sensoren ist, dass die Landmarken durch die Verwendung einer ungenauen Orientierung nicht senkrecht zur Bodenebene stehen. Somit können insbesondere bei der Segmentierung stangenförmiger Objekte und der Zuordnung der Ebenen Fehler auftreten. Für stangenförmige Objekte wird angenommen, dass diese vertikal stehen. Ein falscher Roll- oder

Nick-Winkel kann bewirken, dass diese Annahme nicht erfüllt wird. Werden für aus den Daten segmentierte Ebenen die Minimal- und Maximalwerte in x - und y -Richtung bestimmt, so bewirkt eine ungenaue Orientierung, dass diese Werte verfälscht werden, wobei der Fehler mit zunehmender relativer Höhe der Ebene steigt.

Insgesamt können unter Verwendung der Orientierung aus dem Riegl VMX-250 1849 stangenförmige Objekte und 726 Stangen detektiert und einer Referenz zugeordnet werden. Die Anzahl an False-Positive-Detektionen beträgt in diesem Fall 39 %. Die Wurzel der mittleren quadratischen Abweichung ist im Vergleich zu Abschnitt 7.1.1 höher. Dies ist in der Lage der Landmarken in Bezug zur jeweiligen Fahrzeugposition begründet, da durch die vertikale Einbauposition lediglich Landmarken aufgenommen werden können, welche sich in Fahrtrichtung hinter dem Fahrzeug befinden.

Abbildung 7.4 zeigt, dass bei der Verwendung von Stangen und Ebenen eine Lokalisierung nur in wenigen Bereichen nicht möglich ist. Insbesondere auf der großen Verkehrsstraße, welche den nördlichen Stadtteil (Abbildung 7.4a) sowie die Innenstadt im Süd-Osten ((Abbildung 7.4b) verbindet, ist die erzielbare Genauigkeit vergleichsweise hoch, da sich in diesem Bereich viele gut sichtbare stangenförmige Objekte befinden. Eine Darstellung der Ergebnisse für eine zurückgelegte Strecke von $s_{max} = 30 \text{ m}$ ist in Abbildung A.4 im Anhang zu finden.

7.1.3. Auswertung der Zuordnung über Landmarken-Muster

Für die in Abschnitt 7.1.1 und 7.1.2 ausgewerteten Messungen kann die Zuordnung der Landmarken durch eine Betrachtung der euklidischen Distanz erfolgen, da sowohl die Positionsgenauigkeit des Messfahrzeuges zur Erstellung der Automotive-Daten als auch des verwendeten Mobile-Mapping-Systems bei wenigen Dezimetern liegt. Es ist jedoch davon auszugehen, dass diese Genauigkeiten während der Online-Lokalisierung nicht gewährleistet werden können und somit eine Zuordnung über die euklidischen Distanzen zu Fehlern führen würde. Im Falle der Daten der Messungen des Valeo Scala wird die Zuordnung zusätzlich durch die niedrige Einbauhöhe sowie den geringen Sichtbereich des Laserscanners erschwert. Aus diesem Grund wird hier für die Zuordnung der Landmarken der in Abschnitt 6.1.4 vorgestellte Algorithmus auf die Daten des Valeo Scala angewendet.

Zur Bewertung der Zuordnungs-Ergebnisse werden zwei Indikatoren verwendet: die prozentuale Anzahl korrekter Zuordnungen im Vergleich zu fehlerhaften Zuordnungen sowie die resultierende Erfolgsrate der Lokalisierung entlang der Test-Trajektorie. Ein geringer Anteil an korrekten Zuordnungen würde bedeuten, dass die Lokalisierungsergebnisse durch viele fehlerhafte Zuordnungen verfälscht werden würden. Die Ergebnisse der Zuordnung über Landmarken-Muster werden mit einem Nearest-Neighbor-Ansatz verglichen, wie er in den vorherigen Abschnitten verwendet wurde. Zusätzlich werden die Messgenauigkeit der Ebenenwinkel zur X -Achse $e_{\alpha, \text{Ebenen}}$ sowie die Messgenauigkeit stangenförmiger Objekte e_{Stangen} variiert und die hieraus resultierenden Ergebnisse miteinander verglichen.

Da sowohl für die Automotive- als auch für die Referenzdaten die Genauigkeit der Positionierung hoch ist, wird erwartet, dass auch der Betrag $\sqrt{\Delta x^2 + \Delta y^2}$ des sich ergebenden Translationsvektors im Bereich von wenigen Dezimetern liegt. Es ist ebenfalls zu erwarten, dass innerhalb eines Radius von einem Meter um die Landmarken keine weiteren benachbarten Landmarken auftreten. Aus diesem Grund wurde eine Translation als korrekt angenommen, sobald dessen Betrag unter einem Meter liegt.

Die Ergebnisse der Zuordnung entlang der Test-Trajektorie für eine unterschiedliche Anzahl an verwendeten Landmarken sind in Abbildung 7.5 abgebildet, mit korrekten Zuordnungen in

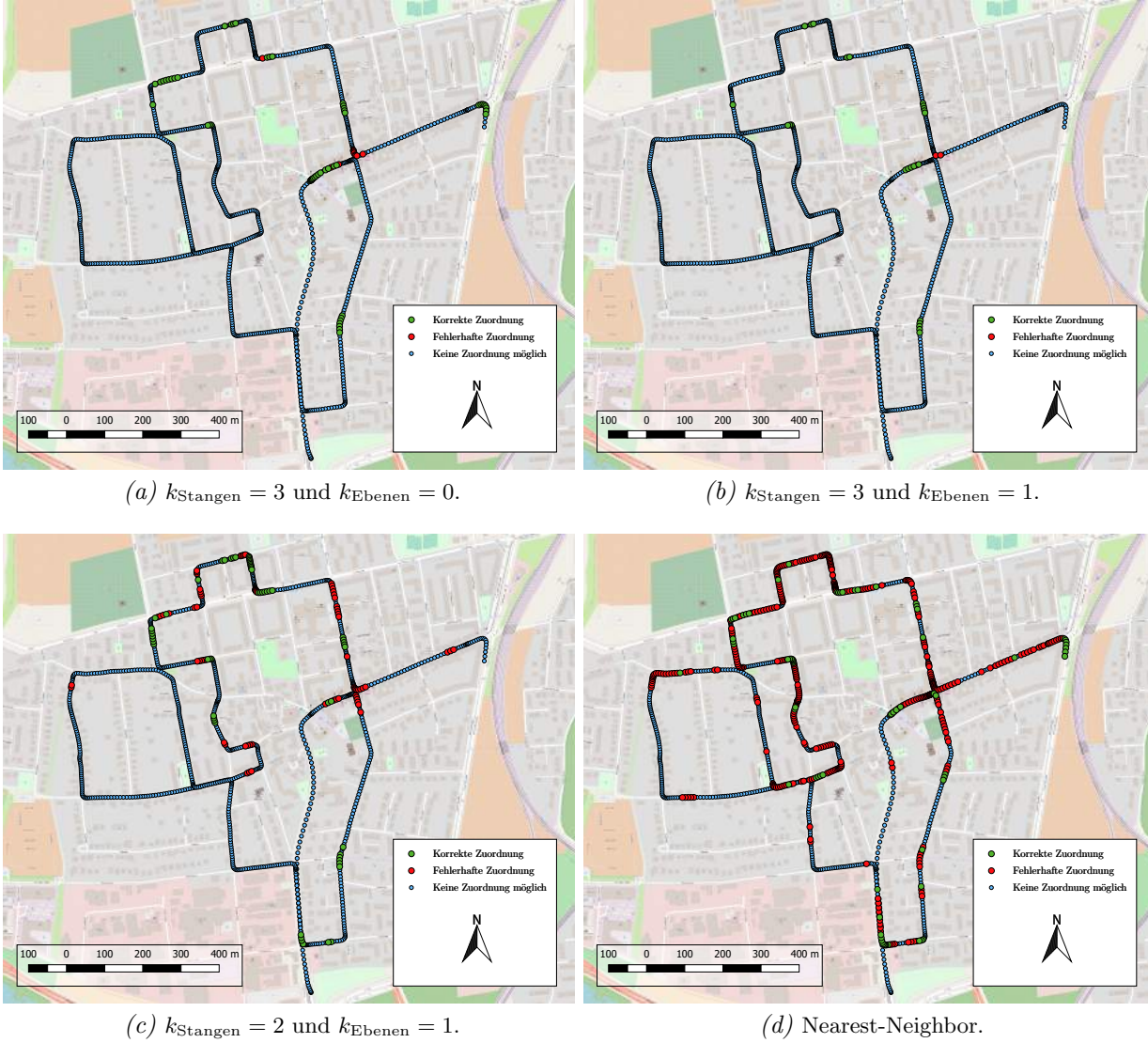


Abbildung 7.5.: Test-Trajektorie mit korrekten Zuordnungen über Landmarken-Muster in grün und fehlerhaften Zuordnungen in blau für $e_{\text{Stangen}} = 0,3\text{ m}$ und $e_{\alpha, \text{Ebenen}} = 15^\circ$ im Vergleich mit einem Nearest-Neighbor-Ansatz (Hintergrundkarte: OpenStreetMap contributors (2018)). Die blauen Symbole repräsentieren Positionen, an denen keine Zuordnung erfolgen konnte.

grün und fehlerhaften Zuordnungen in rot. Die Ergebnisse werden weiterhin in Tabelle 7.5 bis 7.7 für variierende Zuordnungsbedingungen aufgeführt. Es ist ersichtlich, dass die Erfolgsrate der Lokalisierung mit höheren Toleranzwerten geringfügig steigt. Somit bewirken höhere Toleranzwerte eine Steigerung der Vollständigkeit. Dementgegen sinkt die Korrektheit, also die relative Anzahl korrekter Zuordnungen, bei höheren Toleranzwerten.

Tabelle 7.5 zeigt die Ergebnisse für den Fall, dass drei Stangen und keine Ebene im Deskriptor berücksichtigt werden. Wird eine Stange durch eine Ebene ersetzt ($k_{\text{Stangen}} = 2$ und $k_{\text{Ebenen}} = 1$), so führt dies zu einer niedrigeren relativen Anzahl korrekter Zuordnungen bei einer vergleichbaren Lokalisierungsrate. Die besten Ergebnisse für die Korrektheit (89 %) ergeben sich für $e_{\text{Stangen}} = 0.30\text{ m}$ und $e_{\alpha, \text{Ebenen}} = 15^\circ$ bei $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 1$. Die Erfolgsrate der Lokalisierung beträgt hier jedoch nur 3 %. Die Erfolgsrate des Nearest-Neighbor-Ansatzes ist mit 10 % zwar deutlich höher,

jedoch sinkt hier der relative Anteil an korrekten Zuordnungen auf 18 %.

Die im Vergleich zu Abschnitt 7.1.1 deutlich geringere Vollständigkeit lässt darauf schließen, dass die hohe Anzahl an Ausreißern (68 %) sich auch stark auf die Zuordnungen durch Landmarken-Muster auswirkt.

Tabelle 7.5.: Ergebnisse der Zuordnung stangenförmiger Objekte und Ebenen über Landmarkenmuster für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 0$.

Parameter		Erfolgsrate der	Korrekte
e_{Stangen}	$e_{\alpha, \text{Ebenen}}$	Lokalisierung	Zuordnungen
0,20 m	10°	4 %	73 %
0,30 m	15°	6 %	83 %
0,50 m	20°	7 %	73 %
1,00 m	45°	7 %	56 %

Tabelle 7.6.: Ergebnisse der Zuordnung stangenförmiger Objekte und Ebenen über Landmarkenmuster für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 1$.

Parameter		Erfolgsrate der	Korrekte
e_{Stangen}	$e_{\alpha, \text{Ebenen}}$	Lokalisierung	Zuordnungen
0,20 m	10°	2 %	70 %
0,30 m	15°	3 %	89 %
0,50 m	20°	3 %	64 %
1,00 m	45°	3 %	55 %

Tabelle 7.7.: Ergebnisse der Zuordnung stangenförmiger Objekte und Ebenen über Landmarkenmuster für $k_{\text{Stangen}} = 2$ und $k_{\text{Ebenen}} = 1$.

Parameter		Erfolgsrate der	Korrekte
e_{Stangen}	$e_{\alpha, \text{Ebenen}}$	Lokalisierung	Zuordnungen
0,20 m	10°	6 %	52 %
0,30 m	15°	6 %	44 %
0,50 m	20°	6 %	38 %
1,00 m	45°	6 %	32 %

7.1.4. Laufzeitbetrachtung

Die mittlere Laufzeit für die Segmentierung und Parameterbestimmung der Landmarken aus den Daten des Valeo Scala innerhalb einer Sekunde (25 Scans) beträgt 28 ms für Stangen und 161 ms für Ebenen. Die Segmentierung der innerhalb von fünf Sekunden (100 Scans) mit dem Velodyne VLP-16 gemessenen Stangen benötigt eine mittlere Laufzeit von 660 ms, die Ebenendetektion durch einen RANSAC-Algorithmus 1307 ms. Die mittlere Laufzeit der Musteranalyse beträgt 7 ms für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 0$, 23 ms für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 1$ sowie 11 ms für $k_{\text{Stangen}} = 2$

und $k_{\text{Ebenen}} = 1$. Die Filterung der Landmarken, welche die Anzahl der False-Positive-Detektionen verringern soll, benötigt im Mittel 70 ms. Die Evaluierung wurde auf einem 64-Bit-System mit einem 3,70-GHz-Prozessor i5-3570K und 8 GB Arbeitsspeicher durchgeführt.

7.2. Sequenzanalyse

Die in Abschnitt 6.2 vorgestellte Methode der Sequenzanalyse zur globalen Positionsbestimmung in urbanen Gebieten wird anhand drei unterschiedlicher Datensätze mit unterschiedlichen Eigenschaften getestet: anhand der in Abschnitt 4.1 vorgestellten Daten eines Einzeilenscanners, welche aus den Messungen eines Mobile-Mapping-Systems erstellt wurden, der Daten eines tatsächlichen Einzeilenscanners des Oxford Robotcars (siehe Abschnitt 4.3) sowie der Daten eines Velodyne VLP-16 mit 16 Scanzeilen (siehe Abschnitt 4.4). In allen Fällen werden unterschiedliche Konfigurationen für das neuronale Netz zur Erstellung der die jeweiligen Scans repräsentierenden Merkmale verwendet.

7.2.1. Automotive-Daten eines Einzeilen-Laserscanners

Die Lokalisierung über eine Sequenzanalyse durch die aus den Daten eines Mobile-Mapping-Systems erstellten Automotive-Daten wird entlang einer 11 km langen Trajektorie durchgeführt. Die Trajektorie führt durch ein Wohngebiet mit vergleichsweise engen Straßen und hohen Häusern im Norden, einer großen Verkehrsstraße sowie einen Innenstadtbereich im Süden (siehe Abbildung 7.6). Die Länge der Referenz-Trajektorie, mit der die Sequenzanalyse durchgeführt wird, beträgt 26 km und führt zusätzlich durch ein weiteres Wohn- und Stadtgebiet. Die Referenzdaten wurden im Winter 2015 sowie Frühling 2017, die Online-Daten im Sommer 2017 aufgenommen, mit wesentlich mehr Vegetation sowie weiteren Veränderungen in den LiDAR-Daten. Die Positionsbestimmung wird für etwa 35 000 Positionen getestet, ohne jegliche Verwendung von Vorinformationen über GNSS.

Die Labels der Test- sowie der Referenz-Trajektorie sind für einen Ausschnitt beispielhaft in Abbildung 7.7 dargestellt. Es zeigt sich, dass die zugeordneten Labels größtenteils übereinstimmen. Insbesondere, wenn sich das Label in aufeinanderfolgenden Standpunkten nicht ändert, stimmen auch die Werte der Test- und Referenz-Trajektorie überein.

Die Lokalisierung wird anhand zweier Fehlermaße beurteilt: der als Wurzel der mittleren quadratischen Abweichung (RMSE) definierten Genauigkeit der resultierenden Fahrzeug-Positionen sowie der Vollständigkeit, welche sich aus der prozentualen Anzahl der korrekt bestimmten Positionen ergibt. Eine Positionsbestimmung wird als korrekt angenommen, sobald die euklidische 2D-Distanz zur tatsächlich Position weniger als 5 m beträgt.

Zunächst werden die Auswirkungen einer Variation der Parameter des Autoencoders, die Anzahl der fully connected Layer (*FCL*), die Länge des sich ergebenden Merkmalsvektors n_{Merkmale} sowie die Kernelgröße und der Anzahl an Merkmalskarten in den Convolutional-Layern untersucht. Zur Variation der Parameter wird hier ein Latin Hypercube Sampling (Stein, 1987) verwendet.

Tabelle 7.8 zeigt die Ergebnisse der verschiedenen Konfigurationen. Die Anzahl der Cluster k beträgt hierbei 200, die Sequenzlänge $n_{\text{Seq}} = 200$ Scans. Bei einer Fahrzeuggeschwindigkeit von 15 m/s (54 km/h) entspricht dies einer zurückgelegten Distanz von 60 m.

Es ist ersichtlich, dass die besten Ergebnisse für eine Konfiguration mit drei fully connected Layern, 224 Neuronen, einer Kernelgröße von 1 und 42 Merkmalskarten erzielt werden können. Hervorzuheben ist hierbei insbesondere, dass eine Filterung mit einer Kernelgröße von 1 die besten Ergebnisse erzielt, da dies keine Faltung im klassischen Sinne ergibt. Dennoch können 1×1 -Convolutions (in 2D) in einigen Netzwerkarchitekturen gefunden werden (Lin u. a., 2013). Für die entsprechende Konfiguration beträgt die Vollständigkeit *VS* 97,4 %. Der RMSE-Wert liegt hierbei bei 1,37 m. Die Parameter des Netzwerkes werden für die zukünftigen Experimente genutzt, in denen der Effekt von veränderten Werten für n_{Merkmale} , n_{Seq} und k analysiert wird.

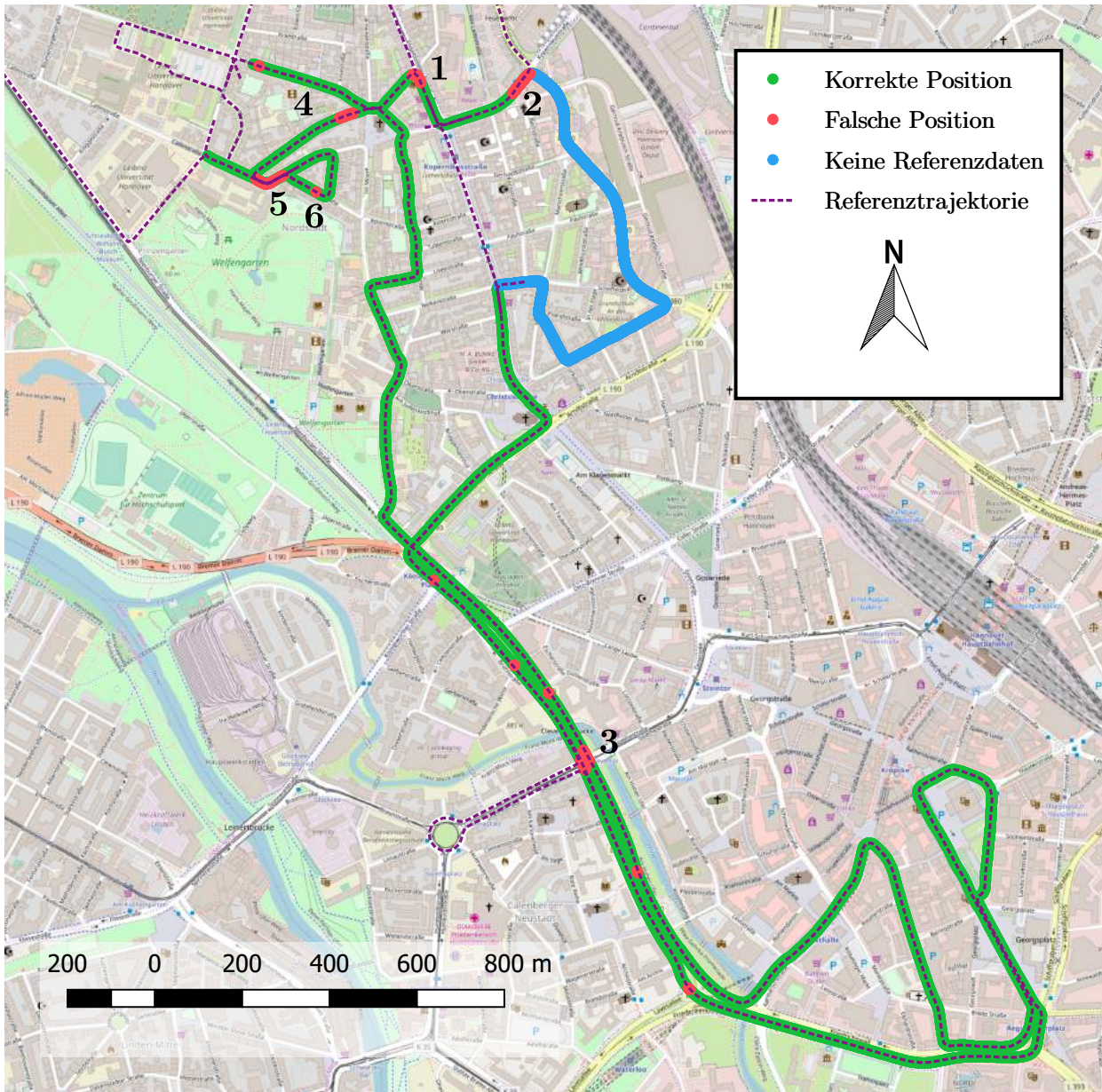


Abbildung 7.6.: Darstellung der korrekten (grün) und fehlerhaften (rot) Lokalisierungsergebnisse durch einer Sequenzanalyse der simulierten Daten eines Einzelenscanners. entlang der Trajektorie (Hintergrundkarte: OpenStreetMap contributors (2018)). Positionen, an denen keine Referenzdaten verfügbar sind, sind blau eingezeichnet. Die gestrichelte Linie stellt die Referenz-Trajektorie dar. In diesem Fall wurde n_{Merkmale} zu 224, k zu 200 und n_{Seq} zu 200 (60 m) gesetzt.

Abbildung 7.8 zeigt die Ergebnisse für eine Variation der Länge des Merkmalsvektors n_{Merkmale} sowie der Sequenzlänge n_{Seq} . Es lässt sich feststellen, dass die Vollständigkeit mit einer höheren Sequenzlänge steigt. Bei einer Länge von $n_{\text{Seq}} = 200$ (60 m) liegt die Vollständigkeit bei 98,9 % (für $n_{\text{Merkmale}} = 100$). Der RMSE beträgt an dieser Stelle 1,40 m. Ist $n_{\text{Merkmale}} \geq 50$, liefert die Verwendung von Merkmalen stets bessere Ergebnisse als die Verwendung der ursprünglichen Distanzmessungen. Wird eine Sequenzlänge $n_{\text{Seq}} \geq 200$ (≥ 60 m) betrachtet, so liefern bereits zehn Merkmale eine höhere Vollständigkeit als die Distanzmessungen. Dies ist insbesondere in Abbildung 7.9a zu erkennen, welche einen Ausschnitt von 7.8a zeigt.

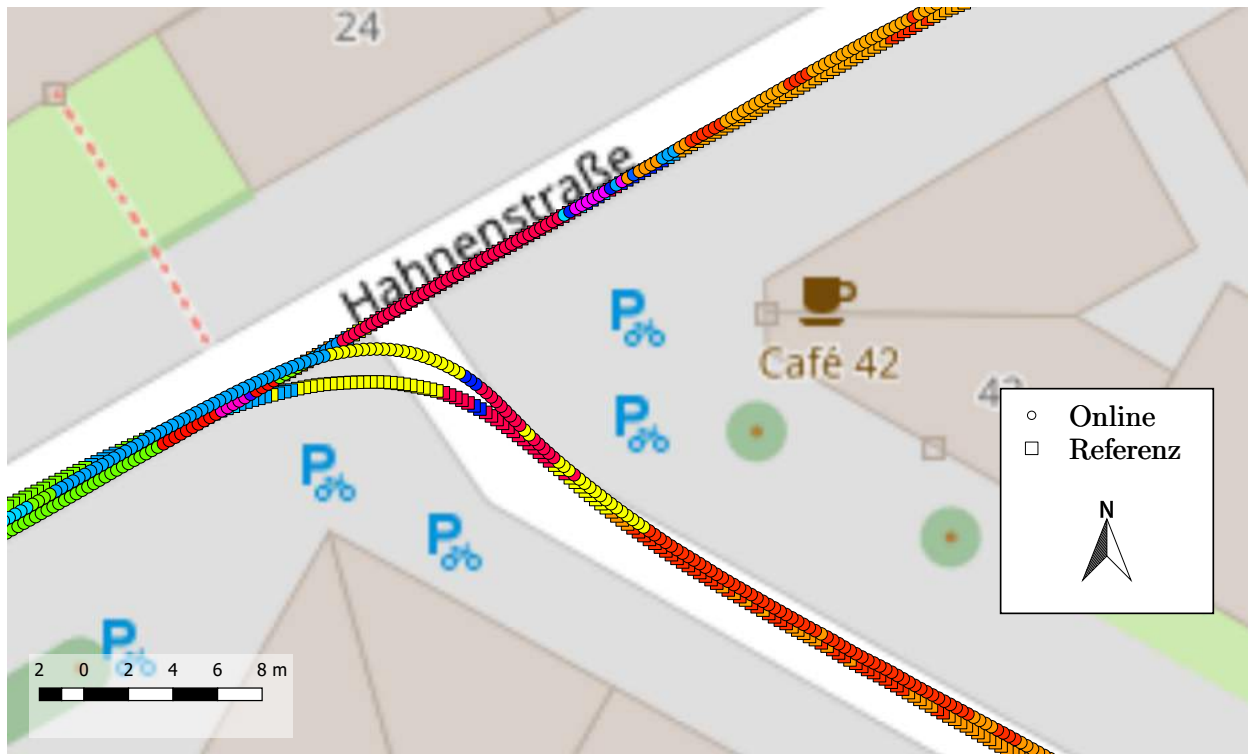


Abbildung 7.7.: Beispielhafte Ergebnisse der durch einen K-Means-Algorithmus mit $k = 100$ gelabelten Referenz- und Online-Trajektorie (Hintergrundkarte: OpenStreetMap contributors (2018)).

Tabelle 7.8.: Auswirkung der verschiedenen Parameter des neuronalen Netzes auf die Ergebnisse einer Sequenzanalyse der simulierten Daten eines Einzelenscanners bei einer Sequenzlänge von $n_{Seq} = 200$ (60 m) und einer Anzahl an Cluster-Mittelpunkten von $k = 200$ auf die Vollständigkeit.

FCL	n_{Merkmale}	Kernelgröße	Merkmalskarten	VS [%]
1	94	2	9	95,7
1	519	1	37	94,9
2	413	5	54	95,6
2	727	4	20	94,7
2	783	6	4	93,6
3	224	1	42	97,4
3	384	5	59	96,3
3	511	3	30	97,2

Neben n_{Merkmale} und n_{Seq} wurde auch die Anzahl der Cluster k variiert (siehe Abbildung 7.9b). Ab einer Clusteranzahl von $k = 20$ liefert der Algorithmus gute Ergebnisse, mit einer Vollständigkeit von 96,3 %. Ein Maximum wird bei $k = 100$ erreicht ($VS = 98,9$ %). Da der Einfluss von k auf die Rechenzeit sehr gering ist, kann hierfür ohne Bedenken ein hoher Wert gewählt werden.

Abbildung 7.6 zeigt eine Karte, in der die Positionen eingetragen sind, an denen die Lokalisierung erfolgreich (grün) und an denen die bestimmte Position fehlerhaft ist (rot). In den Bereichen 4 und 5 treten Fehler an kleinen Plätzen in der Stadt auf. In Bereich 3 könnten sich die Fehler durch das

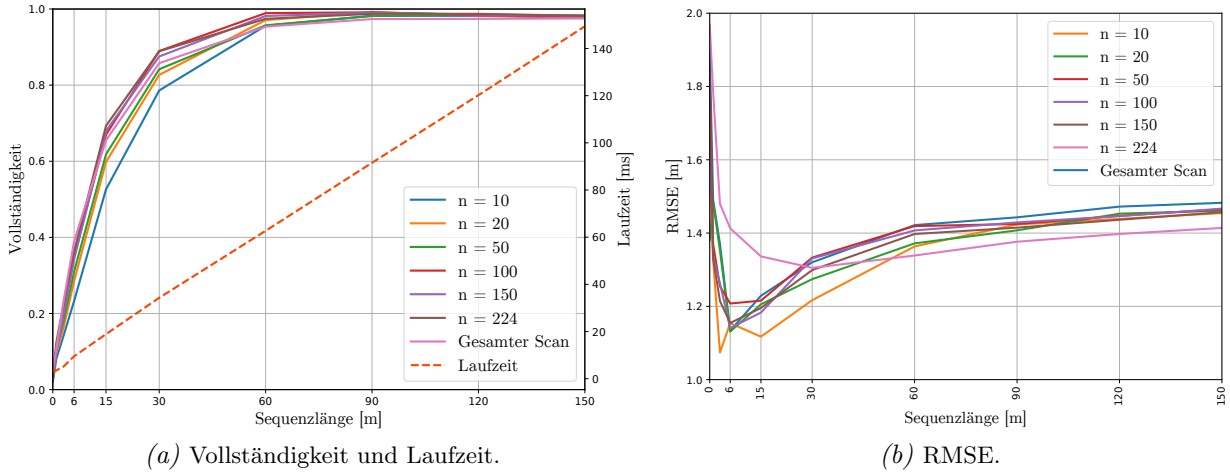


Abbildung 7.8.: Vollständigkeit, Laufzeit und RMSE für eine unterschiedliche Anzahl an Merkmalen n_{Merkmale} von 2-500 und einer Sequenzlänge von 1-150 m unter Verwendung von $k = 100$ Clustern.

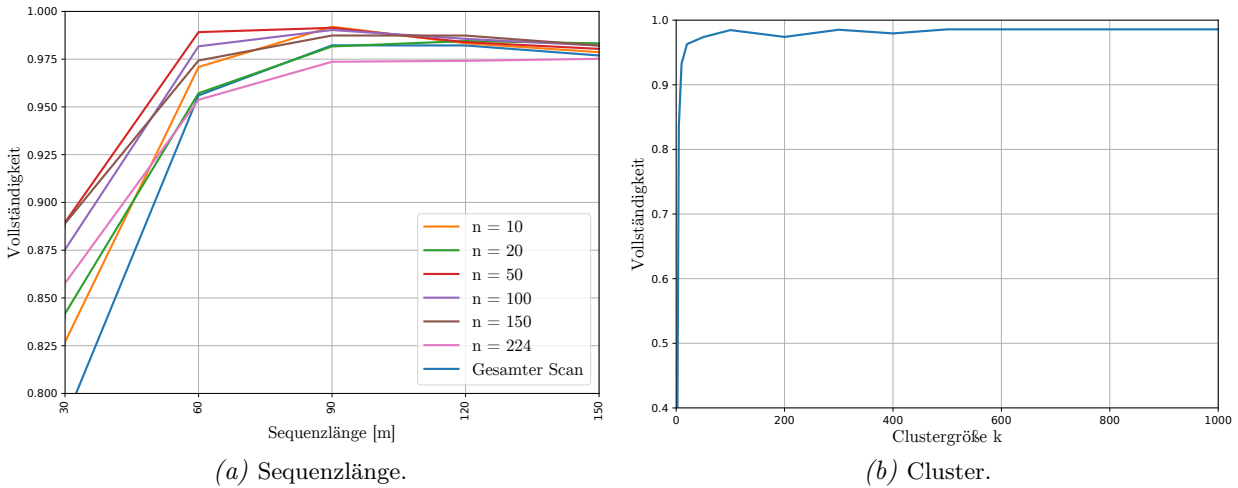


Abbildung 7.9.: Vollständigkeit für eine unterschiedliche Anzahl an Merkmalen n_{Merkmale} von 2-500 und einer Sequenzlänge von 30-150 m unter Verwendung von $k = 100$ Clustern (a) sowie einer Sequenzlänge von 60 m ($n_{\text{Seq}} = 200$) unter einer unterschiedlichen Anzahl an Clustern k von 2-1000 (b).

Straßendesign, mit vielen Bäumen, vier Fahrbahnen und wenig Strukturen am Straßenrand, ergeben.

Die Verwendung der Intensitätswerte anstelle der Distanzmessungen in demselben Netzwerk mit $n_{\text{Merkmale}} = 224$ Merkmalen sowie einer Sequenzlänge von $n_{\text{Seq}} = 200$ (60 m) und einer Clusteranzahl von $k = 200$ bewirkt hier, dass die Ergebnisse deutlich schlechter ausfallen. Die Vollständigkeit beträgt 78,3 % bei einem RMSE-Wert von 1,78 m. Werden die Ergebnisse zweier getrennter Netzwerke mit jeweils $n_{\text{Merkmale}} = 224$ Merkmalen, in denen die Intensitäts- und Distanzwerte verwendet werden, kombiniert, so liegt die Vollständigkeit bei 87,9 %, mit einem RMSE-Wert von 1,71 m.

In den Bereichen 1, 2 und 6 treten jeweils Baustellen und andere große Veränderungen im Straßenraum auf. Es ist anzunehmen, dass in diesen Bereichen die entsprechenden Scans unterschiedlichen Labels als an denselben Positionen in den Referenzdaten zugeordnet werden und die Fehler in diesen Bereichen hierdurch verursacht werden. Dies ist beispielhaft für den Bereich 2 in Abbildung

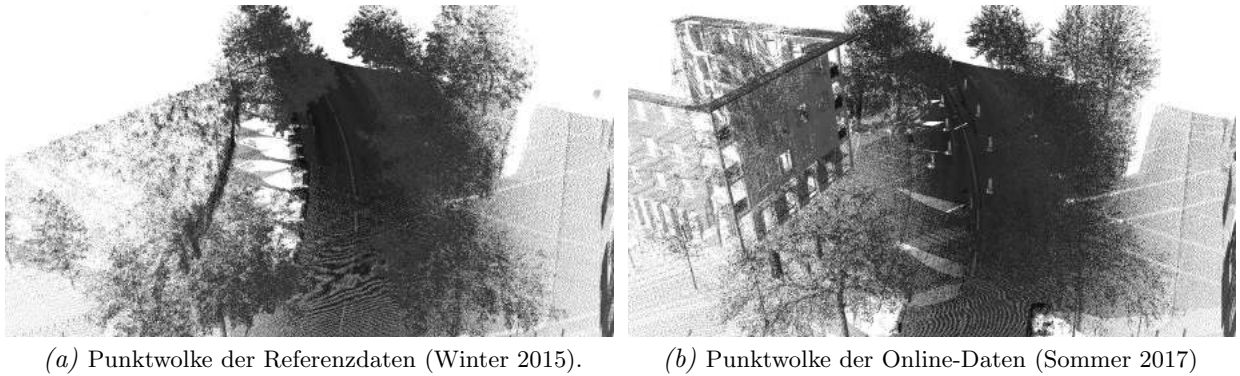


Abbildung 7.10.: Zwei Punktwolken des Bereiches 2 aus Abbildung 7.6. In den Referenzdaten aus dem Winter 2015 befindet sich an der Stelle, an der im Sommer 2017 ein neues Gebäude steht, noch ein unbebautes Grundstück.

7.10 aufgezeigt. Abbildung 7.10b zeigt die Online-Daten, in denen sich ein Gebäude am Straßenrand befindet, wohingegen in den Referenzdaten an derselben Stelle noch ein unbebautes Grundstück zu erkennen ist. Das Problem könnte behoben werden, indem eine Online-Änderungsdetektion der Daten durchgeführt wird. Werden starke Änderungen erkannt, so könnten diese Bereiche durch die Messungen einer Inertialeinheit überbrückt werden. Alternativ können die Ergebnisse der Positionsbestimmung, wie in Abschnitt 7.2.4 gezeigt, innerhalb eines Filters auch als Ausreißer detektiert und folglich nicht zur Lokalisierung verwendet werden.

7.2.2. Oxford Robotcar Dataset

Die Sequenzanalyse zur Lokalisierung wird zusätzlich auf den LiDAR-Messungen des Oxford Robotcars (siehe Abschnitt 4.3) entlang eine Trajektorie mit einer Länge von 3km in Oxford, Großbritannien angewandt. Die zugehörigen Daten wurden an zwei aufeinanderfolgenden Tagen im Sommer 2014 aufgenommen. Da in diesem Fall keine hochgenaue Referenz-Trajektorie zur Verfügung steht, wird eine Position als falsch angenommen, sofern die euklidische Distanz zur

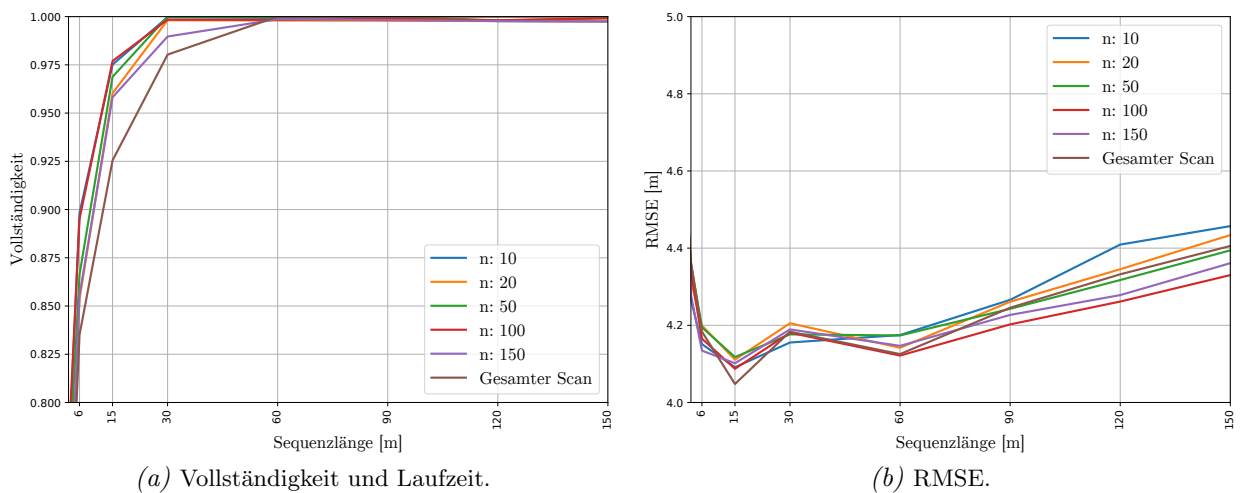


Abbildung 7.11.: Vollständigkeit und RMSE der Sequenzanalyse durch die Daten des Oxford Robotcars für eine unterschiedliche Anzahl an Merkmalen n_{Merkmale} von 10-150 und einer Sequenzlänge von 1-150 m unter Verwendung von $k = 100$ Clustern.

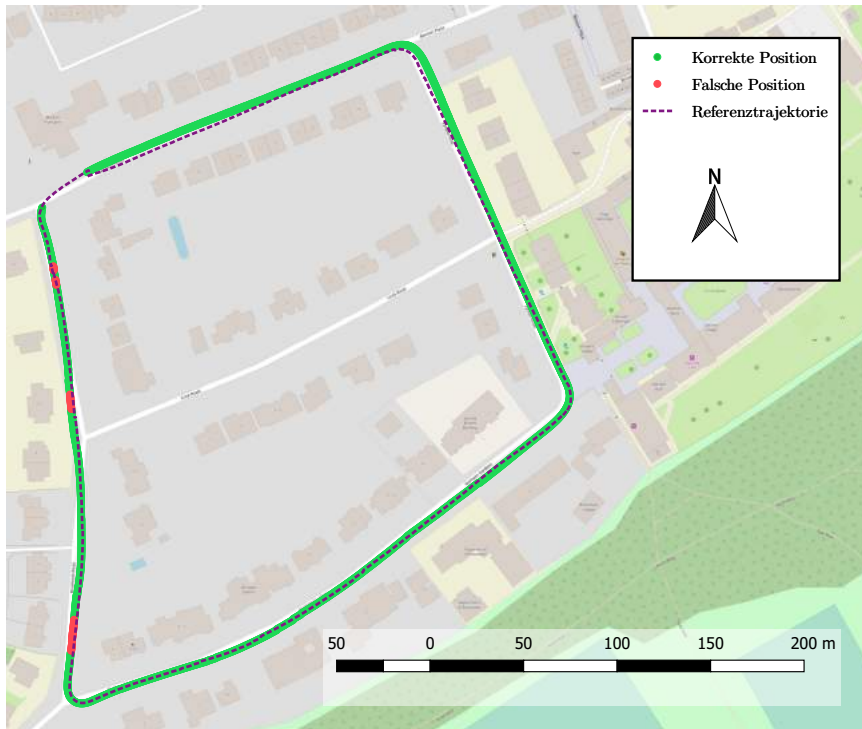


Abbildung 7.12.: Darstellung der korrekten (grün) und fehlerhaften (rot) Lokalisierungsergebnissen entlang der Trajektorie des Oxford Robotcars (Hintergrundkarte: OpenStreetMap contributors (2018)). Die gestrichelte Linie stellt die Referenz-Trajektorie dar. In diesem Fall wurden n_{Merkmale} zu 100, k zu 100 und n_{Seq} zu 50 (15 m) gesetzt.

Referenzmessung mehr als 10 m statt ursprünglich 5 m beträgt. Die Distanz der in Abschnitt 6.2.3 beschriebenen interpolierten Standpunkte beträgt 0,3 m.

Zur Bestimmung der Merkmale wird dasselbe Netzwerk wie in Abschnitt 7.2.1 gewählt. Abbildung 7.11 zeigt die Ergebnisse für eine variierende Sequenzlänge sowie eine unterschiedlichen Anzahl an Merkmalen. In diesem Fall beträgt die Vollständigkeit bereits nach 30 m ($n_{\text{Seq}} = 100$) 100 %. Werden statt der mithilfe des Autoencoders bestimmten Merkmale die ursprünglichen Distanzmessungen verwendet, fallen die Ergebnisse wiederum schlechter aus. In diesem Fall beträgt die Vollständigkeit nach 30 m 98 %. Auf Grund der fehlenden hochgenauen Referenzmessungen kann in diesem Fall keine Angabe zur Genauigkeit der Positionslösung gemacht werden.

Abbildung 7.12 zeigt die Ergebnisse für eine Sequenzlänge von 15 m. Bilder der Frontkamera des Oxford Robotcars der Regionen, in denen fehlerhafte Positionen auftreten, sind in Abbildung 7.13 dargestellt. Es ist anzunehmen, dass sich hier große Bäume negativ auf die Lokalisierung auswirken.

7.2.3. Velodyne VLP-16

Die Daten des Velodyne VLP-16 unterscheiden sich von den in den vorherigen Abschnitten verwendeten Daten darin, dass für jeden Scan 16 statt nur einer Zeile aufgenommen werden. Um die Scans durch aus einem Autoencoder gewonnene Merkmale zu repräsentieren, wird das Netz entsprechend angepasst. Ein hieraus resultierender beispielhafter Aufbau des Netzes ist in Abbildung B.1 im Anhang zu finden. Als Referenz-Trajektorie dient eine Trajektorie mit einer Länge von 69 km (siehe Abbildung B.2 im Anhang). Das Verfahren wird entlang der in Abschnitt 7.2.1 (Abbildung 7.6) vorgestellten Trajektorie mit einer Länge von 11 km getestet, deren Daten im April 2018 aufgenom-



Abbildung 7.13.: Bilder der Frontkamera des Oxford Robotcars in Bereichen der Trajektorie, in denen fehlerhafte Positionen auftreten (Quelle: Maddern u. a. (2017)).

men wurden. Der mit der Test-Trajektorie übereinstimmende Abschnitt wurde im Februar, die übrigen Abschnitte wurden im Februar und Mai 2018 aufgenommen.

Abbildung 7.14 zeigt die vom Encoder rekonstruierten Distanz- sowie Intensitätsbilder. Es ist ersichtlich, dass die Strukturen der Objekte trotz der Reduzierung der Dimension von 14 000 Pixeln zu in diesem Fall 200 Werte erhalten bleibt. Auch die einzelnen Distanzen und Intensitäten stimmen mit dem Input-Bild überein. Lediglich die Schärfe der Bilder nimmt ab.

Tabelle 7.9.: Auswirkung veränderter Parameter des neuronalen Netzes auf die Ergebnisse einer Sequenzanalyse der Daten eines Velodyne VLP-16 bei einer Sequenzlänge von $n_{Seq} = 200$ und einer Anzahl an Cluster-Mittelpunkten von $k = 1000$ auf die Lokalisierungs-Vollständigkeit. Als Pooling-Verfahren dient ein Max Pooling. In diesem Fall werden Convolutions mit 32 Merkmalskarten in der ersten und 64 Karten in der zweiten Convolution verwendet.

Convo- lution	FCL	Pooling	n_{Merkmale}	Kernel- größe	Merkmals- karten	Eingangsdaten	VS [%]
2	1	Nein	50	3	32	Dist.	96,6
2	1	Nein	200	3	32	Int.	75,1
2	1	Nein	100	3	32	Dist. & Int.	92,4
3	1	Nein	50	3	32	Dist.	87,9
2	2	Nein	50	3	32	Dist.	95,5
2	1	Nein	200	7	32	Dist.	96,6
2	1	Nein	100	3	8	Dist.	96,7
2	1	Ja	50	3	32,64	Dist.	96,8

Tabelle 7.9 listet die Ergebnisse der Sequenzanalyse für ausgewählte Netzkonfigurationen bei einer Sequenzlänge von 60 m ($n_{Seq} = 200$) und einer Clusteranzahl von $k = 1000$ auf. Eine ausführliche Übersicht über alle getesteten Konfigurationen ist in Tabelle B.1 im Anhang zu finden. Es zeigt sich, dass die höchste Vollständigkeit bei einem Netzwerk mit zwei convolutional Layern

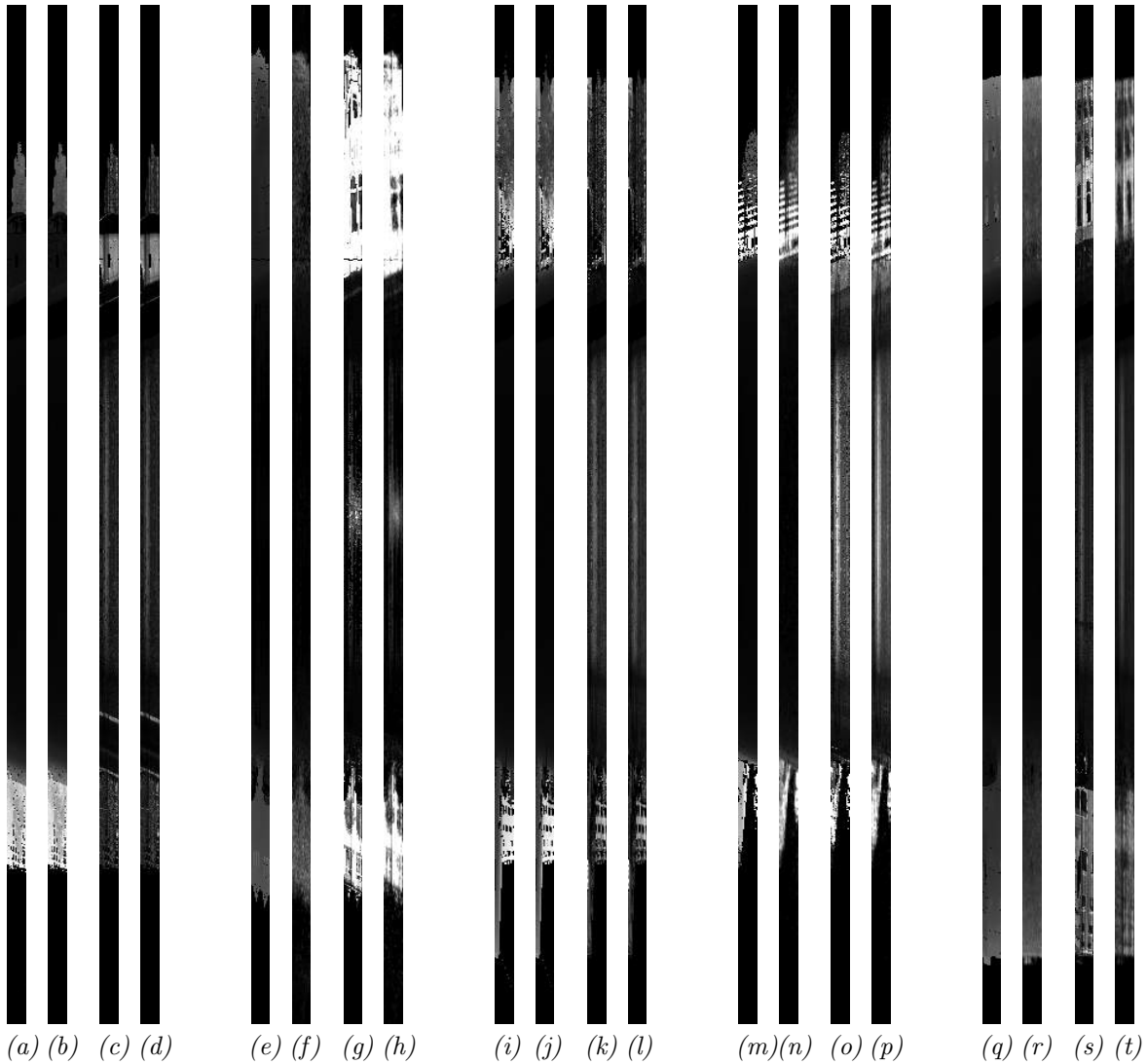


Abbildung 7.14.: Originale (a, c, e, g, i, k, m, o, q, s) sowie vom Encoder prädizierte (b, d, f, h, j, l, n, p, r, t) Scanbilder des Velodyne VLP-16, eingefärbt nach der gemessenen Distanz (a, b, e, f, i, j, m, n, q, r) sowie der Intensität des reflektierten Laserimpulses (c, d, g, h, k, l, o, p, s, t). Hierfür wurde ein Netzwerk mit 2 convolutional Layern mit je 900×16 Neuronen, 32 Merkmalskarten sowie einer Filtergröße von 3 und einem fully connected Layer mit 100 Merkmalen als letzten Layer des Encoders verwendet.

mit Max Pooling und einem fully connected Layer mit $n_{\text{Merkmale}} = 50$ Merkmalen erzielt werden kann. Die Vollständigkeit liegt hier bei 96,8 %, als Eingangsdaten dienen die Distanzmessungen des Laserscanners. Erneut führt die Verwendung der Intensitätswerten zu deutlich schlechteren Ergebnissen. Insgesamt lässt sich feststellen, dass die Auswirkungen verschiedener Parameter, wie der Anzahl der Merkmalskarten oder der Kernelgröße, relativ gering sind. Abbildung 7.15 zeigt die korrekten und fehlerhaften Positionen entlang der Testtrajektorie für eine Sequenzlänge von 60 m ($n_{\text{Seq}} = 200$). Erneut treten Fehler insbesondere an einer großen Straße sowie an Abbiegungen und Kreuzungen auf. Aufgrund der geringen Einbauhöhe des Scanners werden statt der Strukturen am Straßenrand Fahrzeuge erfasst, welche auf benachbarten Fahrbahnen fahren.

Neben der Verwendung verschiedener Netzarchitekturen werden zusätzlich die Auswirkungen einer unterschiedlichen Anzahl an Merkmalen n_{Merkmale} sowie einer variierender Sequenzlänge n_{Seq}

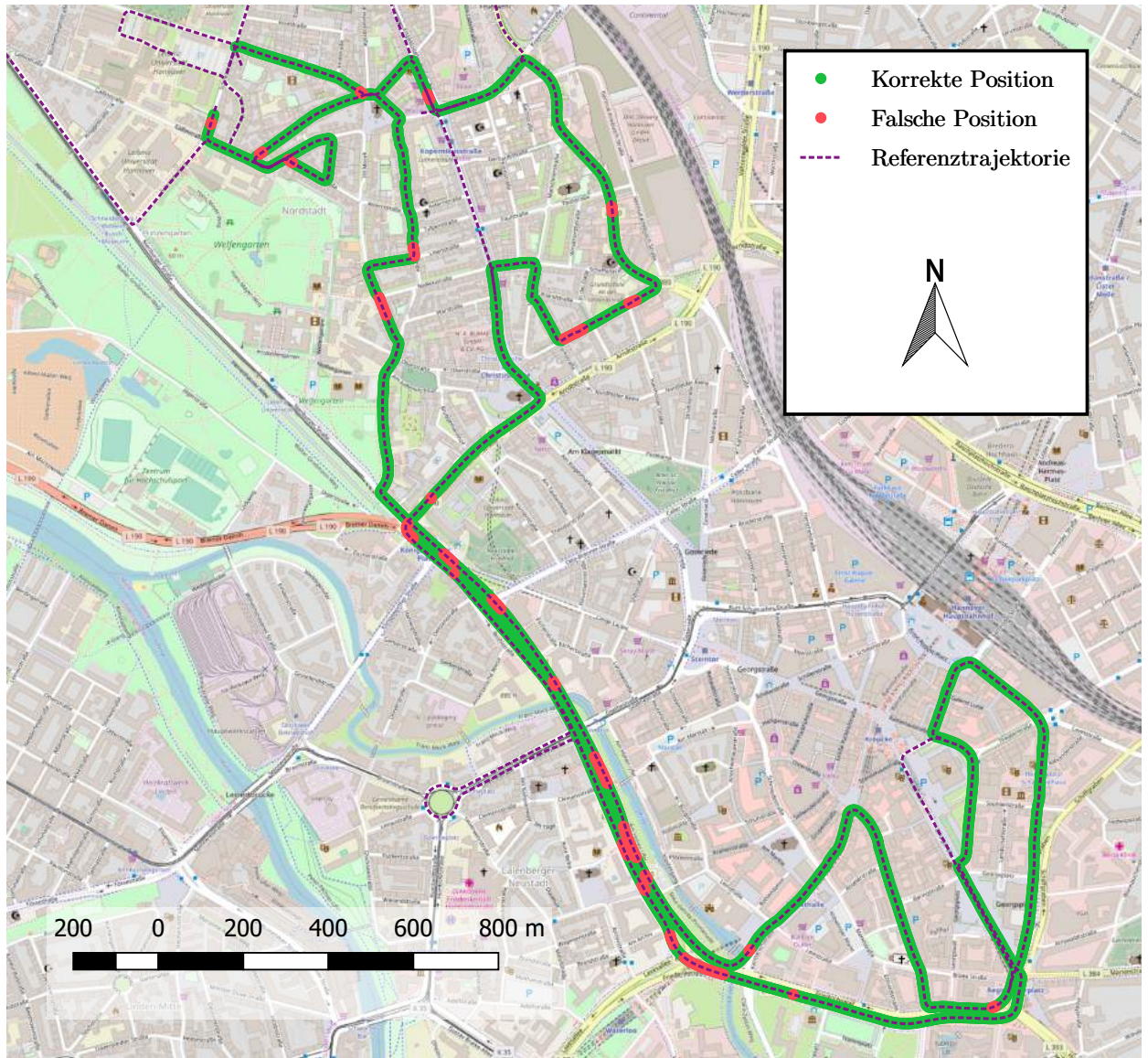


Abbildung 7.15.: Darstellung der korrekten (grün) und fehlerhaften (rot) Lokalisierungsergebnisse durch eine Sequenzanalyse der Daten des Velodyne VLP-16 entlang der Online-Trajektorie (Hintergrundkarte: OpenStreetMap contributors (2018)). Die gestrichelte Linie stellt die Referenz-Trajektorie dar. In diesem Fall wird ein Netzwerk mit 2 convolutional Layern mit je 900×16 Neuronen, 32 Merkmalskarten sowie einer Filtergröße von 3 und einem fully connected Layer mit $n_{\text{Merkmale}} = 200$ Merkmalen als letzten Layer des Encoders verwendet. Als Input-Daten dienen die Distanzwerte. Die Vollständigkeit der Sequenzanalyse unter der Verwendung von $k = 1000$ Clustern und einer Sequenzlänge von 60 m ($n_{\text{Seq}} = 200$) beträgt 94,3%.

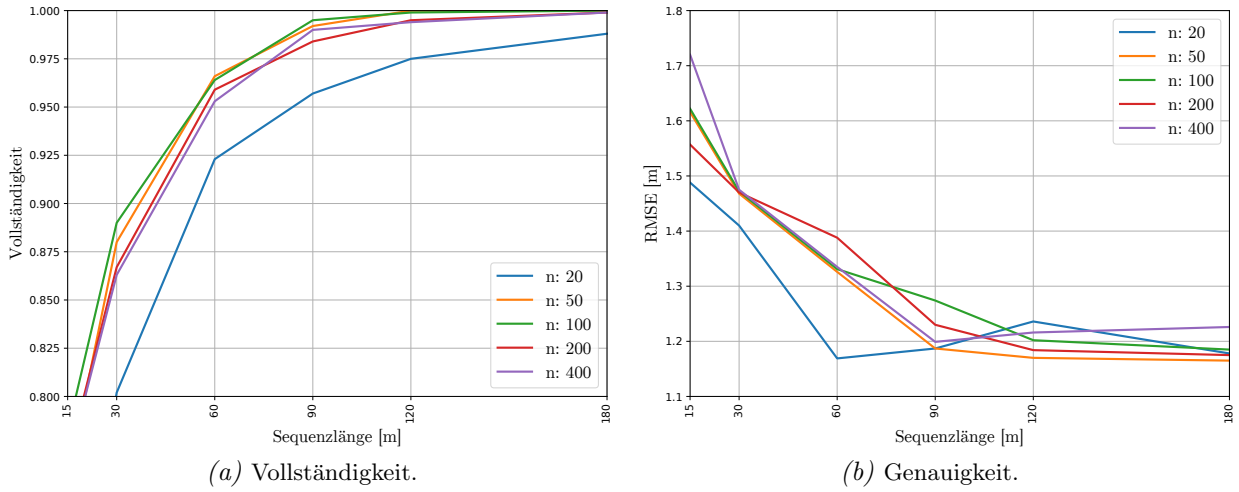


Abbildung 7.16.: Vollständigkeit und Genauigkeit der Sequenzanalyse durch die Daten des Velodyne VLP-16 für eine unterschiedliche Anzahl an Merkmalen n_{Merkmale} von 20-400 und einer Sequenzlänge von 15-180 m unter Verwendung von $k = 1000$ Clustern.

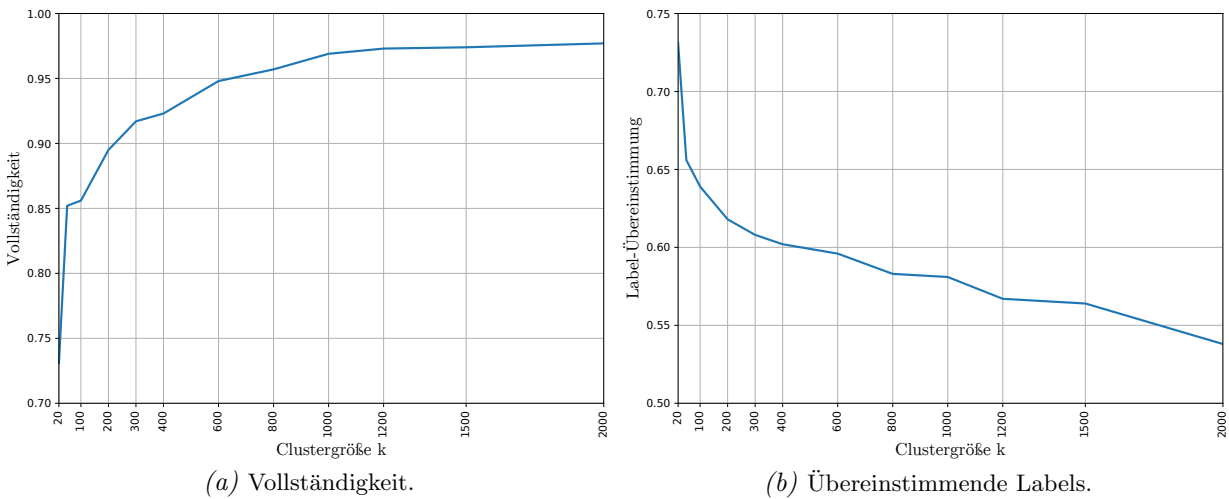


Abbildung 7.17.: Vollständigkeit und prozentualer Anteil übereinstimmender Labels der Sequenzanalyse durch die Daten des Velodyne VLP-16 für $n_{\text{Merkmale}} = 200$ und einer Sequenzlänge von 60 m ($n_{\text{Seq}} = 200$) bei einer variierenden Anzahl an Clustern k von 10-2000.

und Clusteranzahl k untersucht. Die Ergebnisse sind in Abbildung 7.16 und 7.17 dargestellt. Ab einer Sequenzlänge von 120 m ($n_{\text{Seq}} = 400$) kann bei einer Verwendung von $k = 1000$ Clustern für $n_{\text{Merkmale}} = 50$ eine Vollständigkeit von 100 % erreicht werden. Für eine geänderte Clusteranzahl verbessern sich die Ergebnisse mit einer steigenden Clusteranzahl. Vergleichsweise gute Ergebnisse mit einer Vollständigkeit von 96,8 % bei einer Sequenzlänge von 60 m ($n_{\text{Seq}} = 200$) ergeben sich für $k \geq 1000$ Cluster. Werden 2000 Clustermittelpunkte verwendet, so steigt die Vollständigkeit auf 97,7 %. Der Mittelwert der prozentual übereinstimmenden Label zwischen der Online- und der Referenzsequenz sinkt mit einer höheren Anzahl an Clustern, liegt für $k = 2000$ jedoch noch bei 54 %.

Die östliche Referenz- und Test-Trajektorie in Abbildung 7.18a zeigt eine Situation, in der das Fahrzeug während der Referenz- und Testfahrt im nördlichen Abschnitt auf unterschiedlichen

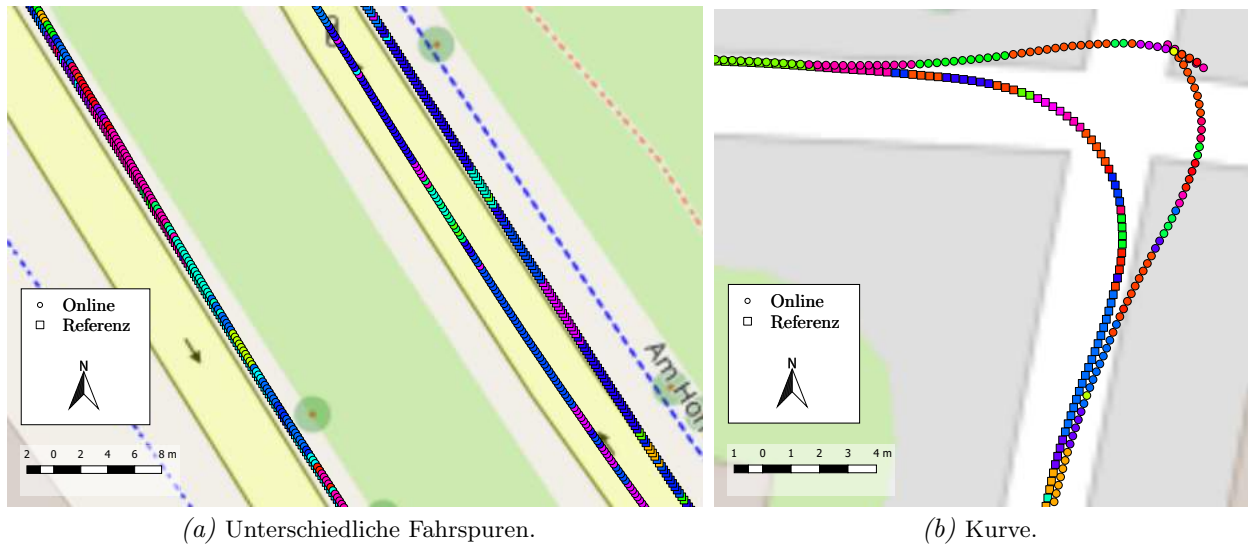


Abbildung 7.18.: Ergebnisse des K-Means-Algorithmus für $k = 100$ entlang unterschiedlicher Fahrspuren und einer Kurve, in der ein Wendemanöver durchgeführt wurde (Hintergrundkarte: OpenStreet-Map contributors (2018)).

Fahrspuren mit einem Abstand von etwa 3 m zueinander gefahren ist. Auch in diesem Fall stimmen die Labels größtenteils überein, sodass die Position weiterhin korrekt bestimmt werden kann.

Eine Situation, die zu Fehlern führen kann, ist in Abbildung 7.18b dargestellt. Dadurch, dass in diesem Fall eine Kurve in beiden Trajektorien unterschiedlich gefahren wird, unterscheidet sich die Länge der Trajektorien in diesem Abschnitt, was eine Verschiebung der ermittelten Position in der Sequenzanalyse bewirkt. Zusätzlich wurde in diesem Beispiel das Fahrzeug im nord-östlichen Bereich geparkt, was die Verlängerung der Trajektorie durch das durchgeführte Wendemanöver weiterhin erhöht. Insgesamt bewirken beide Verlängerungen in diesem Fall eine Positionsdivergenz von etwa 8 m zwischen der durch die Sequenzanalyse bestimmten und der tatsächlichen Position. Die hier betrachtete Test-Trajektorie wird nicht zur Auswertung der Genauigkeit und Vollständigkeit des Verfahrens verwendet. Sie wird lediglich genutzt, um potentielle Fehlerquellen des Verfahrens zu erfassen.

7.2.4. Verwendung eines Filteransatzes

Abbildung 7.15 zeigt die Ergebnisse einer Sequenzanalyse der Daten des Velodyne VLP-16 mit einer Sequenzlänge von 60 m ($n_{\text{Seq}} = 200$). Bereiche, in denen die Pose nicht korrekt bestimmt werden kann (rot), sind auf kurze Abschnitte beschränkt. Ist die Lokalisierung fehlerhaft, so können diese Bereiche durch die Verwendung einer Inertialeinheit überbrückt werden. Die entsprechenden Messungen werden durch die Xsens MTi-G durchgeführt und aus der Trajektorie des Riegl VMX-250 abgeleitet. Die Kombination mit den Ergebnissen der Sequenzanalyse erfolgt hier in einem EKF (siehe Abschnitt 2.3.3), mit dem Bewegungsmodell aus Gleichung 2.56. Die Frequenz der Prädiktion durch die Geschwindigkeit v und Drehrate ω beträgt 200 Hz. Mithilfe der Sequenzanalyse wird hier neben der Position weiterhin die Orientierung (Heading θ) des Fahrzeuges bestimmt und als Korrektur im Filter mit einer Frequenz von 5 Hz verwendet. Um zu verhindern, dass fehlerhafte Posenschätzungen der Sequenzanalyse ebenfalls in den Filter eingehen, findet eine Ausreißererkennung durch einen Vergleich zwischen dem aktuellen Zustandsvektor und der Pose aus der Sequenzanalyse statt. Ist die euklidische Distanz der Position größer als 50 m oder stimmt die Orientierung auf weniger als 90° überein, so wird die aus der Sequenzanalyse bestimmte Pose verworfen.

Tabelle 7.10.: Ergebnisse des EKF bei einer Kombination von aus der Sequenzanalyse bestimmten Werte für Position und Orientierung und der Geschwindigkeits- und Drehratenwerte aus der Xsens MTi-G sowie aus den Messungen des Riegl VMX-250.

Datenquelle Prädiktion	$RMSE_{xy}$	$RMSE_{\theta}$
Xsens MTi-G	3,53 m	14,9°
Riegl VMX-250	1,76 m	17,3°

Das Messrauschen, also die Varianz der aus der Sequenzanalyse bestimmten Position und Orientierung, wird zu $\sigma_{xy}^2 = (1 \text{ m})^2$ und $\sigma_{\theta}^2 = (1^\circ)^2$ gesetzt. Das Prozessrauschen, welches hier durch die Varianz der Geschwindigkeits- und Drehratenmessung bestimmt wird, zu $\sigma_v^2 = (10 \frac{\text{m}}{\text{s}})^2$ und $\sigma_{\omega}^2 = (1 \frac{^\circ}{\text{s}})^2$.

Werden v und ω aus der hochgenauen Trajektorie des Riegl VMX-250 abgeleitet, so ist zu erwarten, dass der Zustandsvektor im Bereich einer fehlerhaften Sequenzanalyse sehr genau prädiziert werden kann. Dies zeigt sich auch in den in Tabelle 7.10 aufgeführten Genauigkeitswerten des Zustandsvektors. Mit 1,76 m liegt der RMSE in derselben Größenordnung wie der RMSE der korrekt bestimmten Positionen durch eine Sequenzanalyse. Wird hingegen die Xsens MTi-G zur Prädiktion der Zustände herangezogen, so sinkt die Genauigkeit auf 3,53 m. Fehlerhafte Werte in der Geschwindigkeit und Drehrate können zu starken Abweichungen in der Position und Orientierung führen. Ein Beispiel eines fehlerhaft prädizierten Zustandes wird in Abbildung 7.19a dargestellt. Durch eine zu hohe Geschwindigkeit weicht die Position bereits nach einer Sekunde um mehr als 15 m von der tatsächlichen Position ab. Dieses Problem könnte reduziert werden, indem die

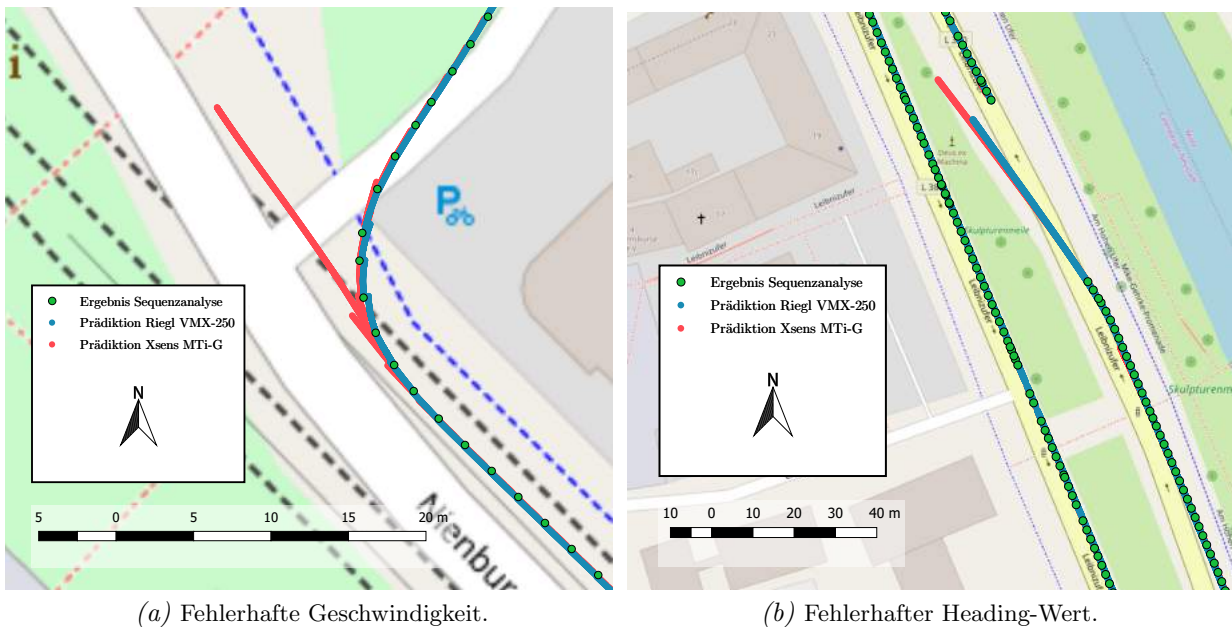


Abbildung 7.19.: Ergebnisse des EKF bei einer Kombination von aus der Sequenzanalyse bestimmten Werte für Position und Orientierung und der Geschwindigkeits- und Drehratenwerte aus der Xsens MTi-G (rot) sowie aus den Messungen des Riegl VMX-250 (blau) (Hintergrundkarte: OpenStreetMap contributors (2018)). Die zum Teil fehlerhaften Geschwindigkeitswerte der Xsens MTi-G führen zu Fehlern in der Prädiktion der Position (a). Ist der sich aus der Sequenzanalyse ergebende Heading-Wert fehlerhaft, wirkt sich dies direkt auf die in der Folge prädizierten Positionen aus (b).

Geschwindigkeit ebenfalls in den Zustandsvektor eingeht und somit gefiltert wird. Ein weiteres Problem des Filteransatzes ist in Abbildung 7.19b ersichtlich. Das Heading weicht nach der Korrektur durch das Ergebnis der Sequenzanalyse vom tatsächlichen Heading um wenige Grad ab. Diese Abweichung führt hier in der folgenden Prädiktion ebenfalls zu starken Fehlern.

7.2.5. Laufzeitbetrachtung

Bei einer Sequenzlänge von $n_{\text{Seq}} = 200$ beträgt die zur Bestimmung einer Position benötigte Rechenzeit auf einem Computer mit einem 2,6-GHz-Prozessor 62 ms. Die Rechenzeit für die Bestimmung der Merkmale aus dem in Abschnitt 7.2.1 verwendeten neuronalen Netz auf einer Titan X GPU liegt unter 1 ms.

7.3. Scanbildkorrelation

Die Lokalisierung über Intensitäts- und Höhenbilder wird entlang der bereits in Abschnitt 7.2.1 präsentierten 11 km langen Trajektorie mit den in Abschnitt 4.1 vorgestellten Automotive-Laser-scanning-Daten sowie mit den Daten des in Abschnitt 4.4 vorgestellten Velodyne VLP-16 durchgeführt. Als Referenzdaten dienen jeweils Scanbilder, deren Daten aus acht verschiedenen Messfahrten zu verschiedenen Jahres- und Tageszeiten stammen. Zur Extraktion dynamischer Objekte werden die Messungen in jeder Iteration mit den Daten fünf weiterer Messkampagnen verglichen.

Sowohl für die simulierten Automotive-Daten als auch für die Daten des Velodyne-Laserscanners wird stets die Transformation zu demjenigen Mobile-Mapping-Datensatz bestimmt und angebracht, welcher parallel zu den Automotive-Daten aufgenommen wurde. Im Fall der simulierten Daten sind dies dieselben Daten, aus denen die gefilterten Scanlinien erzeugt wurden. Für den Velodyne VLP-16 befand sich das Mobile-Mapping-System am selben Fahrzeug, an dem auch der VLP-16 montiert war. Die parallele Aufnahme der Daten hat zum Vorteil, dass die Transformation zwischen den Positionen der Scanbilder zu der aktuellen Position des Fahrzeuges im Lokalisierungsschritt bekannt ist. Für die simulierten Daten beträgt sie für die Translation in x - und y -Richtung sowie für die Orientierung 0. Für die Daten des Velodyne VLP-16 können die Werte um wenige Zentimeter und Grad durch Restfehler in der Kalibrierung abweichen.

Weiterhin ist zu beachten, dass die zeitgleich aufgenommenen Mobile-Mapping-Daten lediglich zur Registrierung aber nicht zur Erstellung der Intensitäts- und Höhenbilder verwendet werden, da zu erwarten ist, dass die zeitgleiche Aufnahme von Referenz- und Online-Daten die Ergebnisse verfälschen würde.

7.3.1. Automotive-Daten eines Einzeilen-Laserscanners

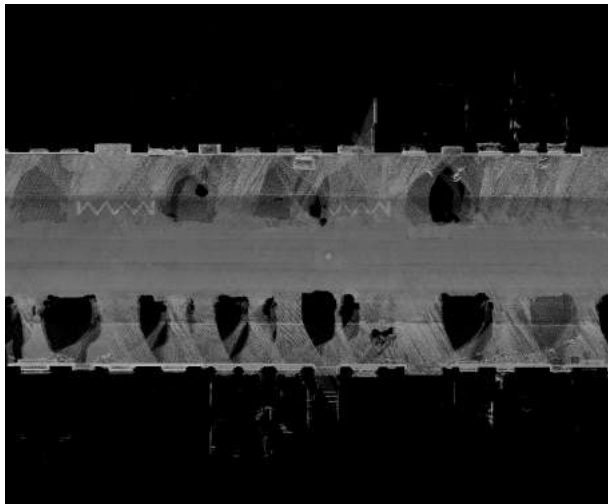
Um dynamische Objekte aus der Bestimmung der Korrelation auszuschließen, wird zunächst die in Abschnitt 5.2.2 vorgestellte Methode zur Detektion dynamischer Objekte in einzelnen Scans angewandt. Aus den Abweichungen zwischen den aus der Scanbildkorrelation bestimmten und den tatsächlichen Positionen Δx und Δy werden Fehlerellipsen für jeden Standpunkt erstellt, welche in Abbildung 7.21 eingezeichnet sind. Ist die 2D-Abweichung $\sqrt{\Delta x^2 + \Delta y^2}$ größer als 0,5 m, so wird das Ergebnis als falsch angenommen und fließt nicht in die Berechnung des RMSE mit ein. Gleichzeitig ergibt sich die Vollständigkeit der Lokalisierung entlang der Trajektorie aus der relativen Anzahl an Standpunkten mit einer Abweichung von unter 0,5 m. Die Ergebnisse sind in Tabelle 7.11 und 7.12 aufgeführt, wobei zwischen den Genauigkeiten mit und ohne Anwendung der Detektion dynamischer Objekte unterschieden wird.

Tabelle 7.11.: Ergebnisse der Lokalisierung über eine Scanbildkorrelation der simulierten Daten eines Einzeilenscanners mit und ohne Anwendung der Klassifizierung zur Detektion dynamischer Objekte für die Position x , y sowie die Orientierung θ .

	Keine Klassifizierung	Klassifizierung
$RMSE_x$	0,05 m	0,04 m
$RMSE_y$	0,02 m	0,02 m
$RMSE_{xy}$	0,05 m	0,05 m
$RMSE_\theta$	0,06°	0,06°
Vollständigkeit	88 %	93 %

Tabelle 7.12.: Ergebnisse der Lokalisierung über eine Scanbildkorrelation der simulierten Daten eines Einzelenscanners nach Anwendung der Klassifizierung zur Detektion dynamischer Objekte für die Position x , y sowie die Orientierung θ in einem Wohngebiet, entlang einer großen Verkehrsstraße sowie im Innenstadtbereichs.

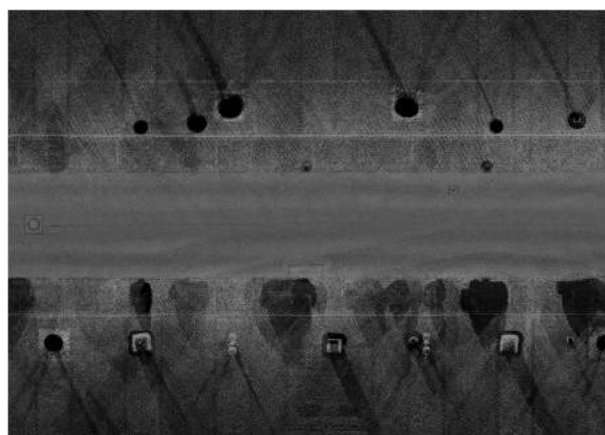
	Wohngebiet	Große Verkehrsstraße	Innenstadt
$RMSE_x$	0,05 m	0,05 m	0,02 m
$RMSE_y$	0,02 m	0,02 m	0,02 m
$RMSE_{xy}$	0,05 m	0,05 m	0,03 m
$RMSE_\theta$	0,07°	0,06°	0,07°
Vollständigkeit	97 %	91 %	89 %



(a) Wohngebiet.

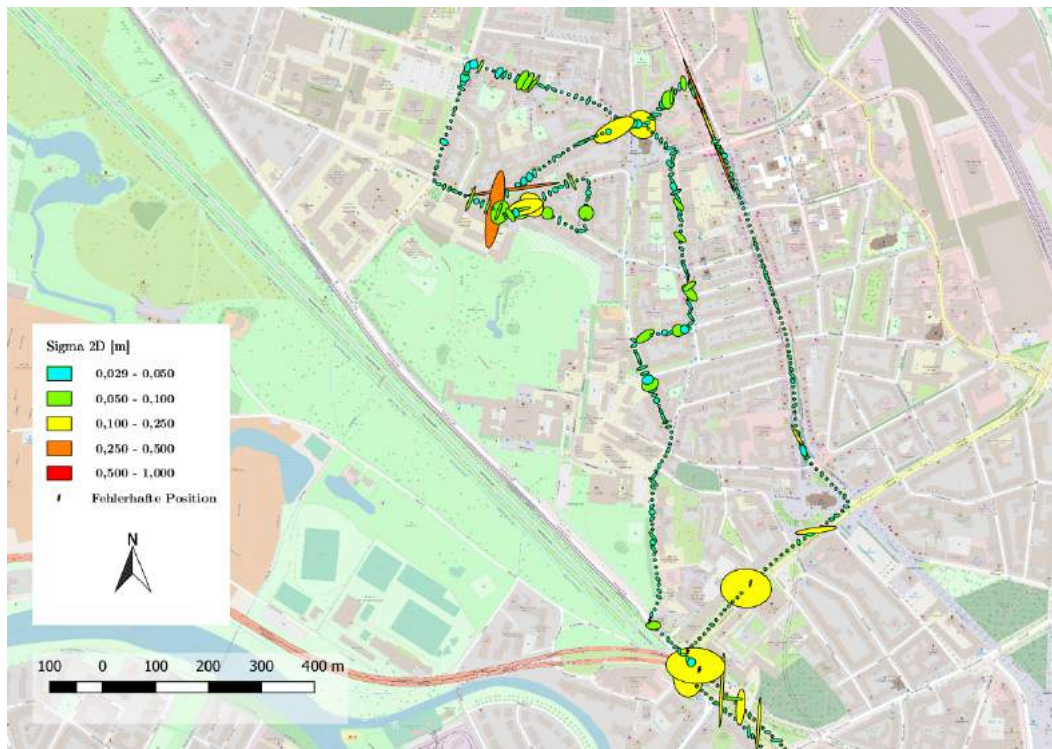


(b) Große Verkehrsstraße.

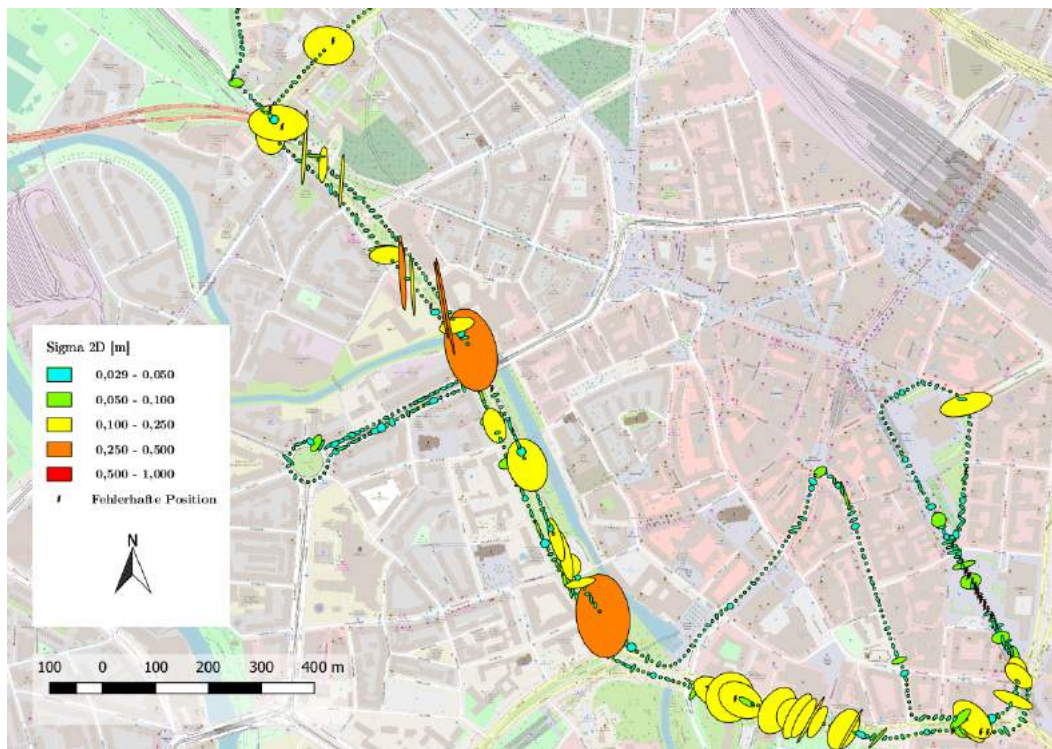


(c) Geringe Strukturen und Höhenunterschiede.

Abbildung 7.20.: Beispiele für nach der Intensität eingefärbte Scanbilder in einem Wohngebiet, entlang einer großen Verkehrsstraße und entlang einer Straße mit geringen Strukturen und Höhenunterschieden.



(a) Nördlicher Bereich.



(b) Südlicher Bereich.

Abbildung 7.21.: Fehlerellipsen der Lokalisierung durch Scanbilder durch die Messungen eines Einzelenscanners, welche aus den Daten des Riegl VMX-250 simuliert sind (Hintergrundkarte: OpenStreetMap contributors (2018)). Die Trajektorie führt durch ein Wohngebiet im Norden, entlang einer großen Verkehrsstraße sowie einen Innenstadtbereich im Süd-Osten. Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

Es zeigt sich, dass eine Genauigkeit von 0,05 m und 0,06° möglich ist. Insbesondere in Wohngebieten, mit engen Straßen und nahestehende Gebäuden, wie in Abbildung 7.20a zu sehen, lassen sich gute Ergebnisse erzielen. Die Vollständigkeit, also die Anzahl der Trajektorie-Punkte, an denen die Abweichung in der Lage weniger als 0,05 m beträgt, liegt hier bei 97 %. Zu fehlerhaften Lokalisierungen kann es insbesondere kommen, wenn Bäume nicht aus der Erstellung der Referenzbilder ausgeschlossen werden können oder innerhalb des von den Scanbildern umfassenden Bereiches wenig Strukturen und Höhendifferenzen auftreten, wie es im in Abbildung 7.20c gezeigten Bild der Fall ist. Dies ist hier in einem Bereich in der Innenstadt der Fall, siehe Abbildung 7.21b, wodurch die Vollständigkeit in diesem Gebiet nur bei 89 % liegt. Entlang einer großen Straße ist die Vollständigkeit mit 91 % ebenfalls niedrig. Als Grund hierfür werden in der Klassifizierung nicht detektierte Bäume vermutet (siehe Abbildung 7.20b).

7.3.2. Velodyne VLP-16

Die Auswertung der Anwendung des Verfahrens der Scanbildkorrelation auf die Daten des Velodyne VLP-16 wird in denselben Stadtgebieten von Hannover wie in Abschnitt 7.3.1 durchgeführt. Zur Erstellung der Punktwolken aus den Messungen des Laserscanners werden Orientierung und Position des Scanners aus den Daten des Riegl VMX-250 bestimmt.

Abbildung 7.22 zeigt die überlagerten Punkte des Riegl VMX-250 sowie des Velodyne VLP-16 nach der Anbringung der Transformation T_{max} auf die Messungen des Velodyne-Laserscanners, welche sich aus der maximalen Korrelation ergibt. Es ist sichtbar, dass der Algorithmus in diesem Fall die korrekte Transformation ergibt.

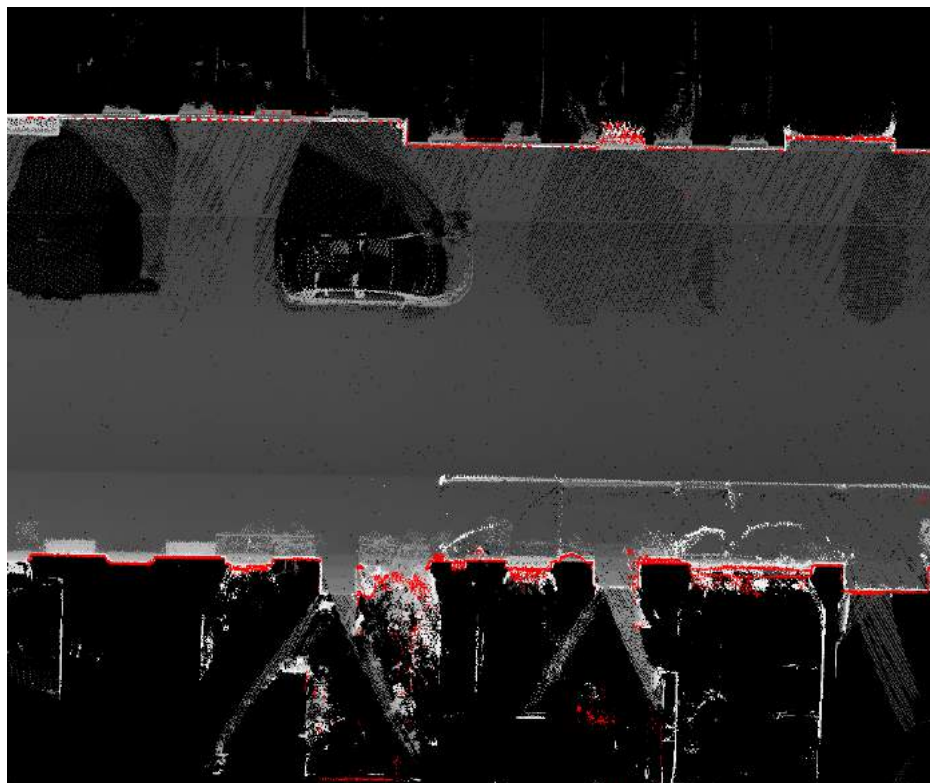
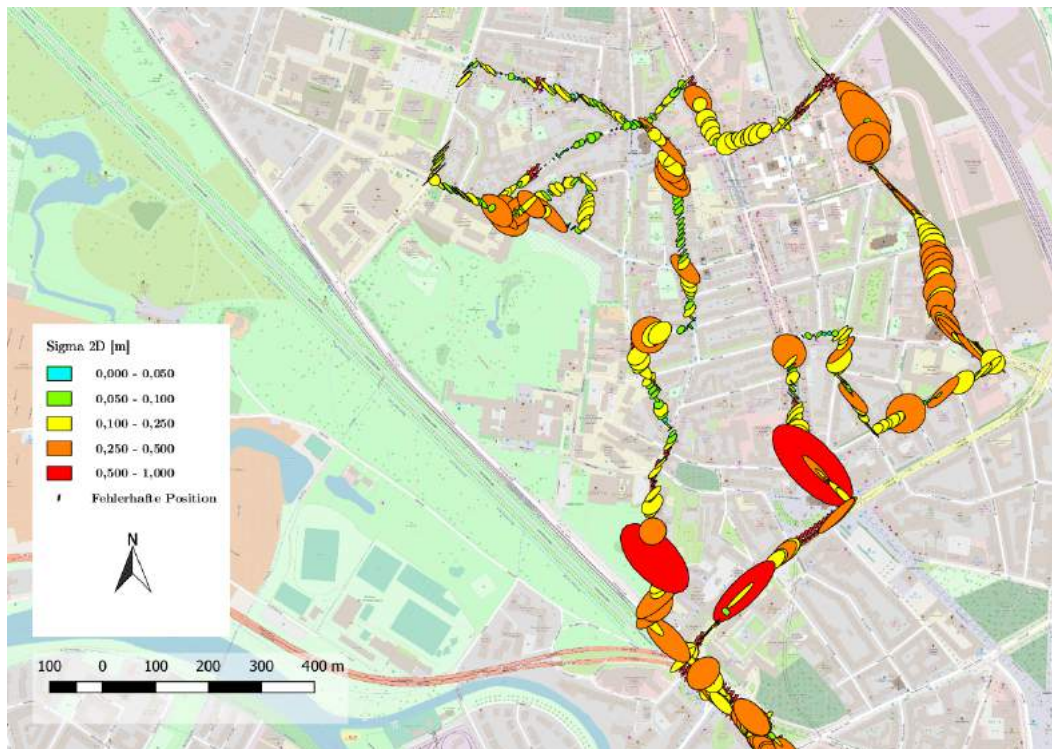
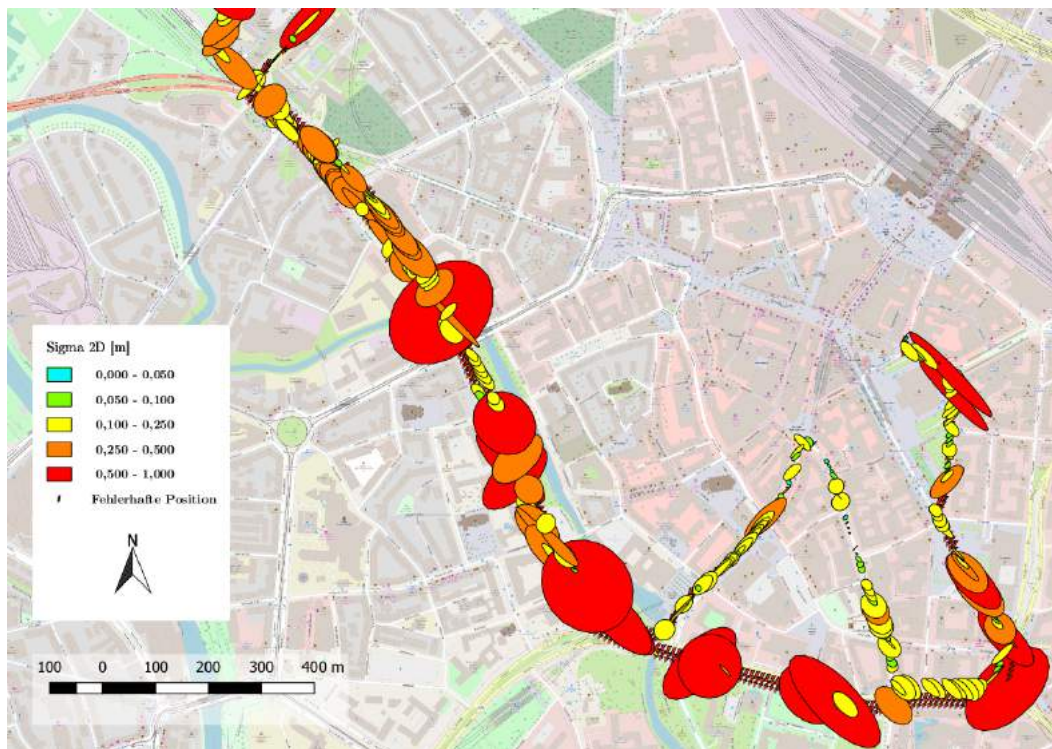


Abbildung 7.22.: Überlagerte auf die x/y -Ebene projizierte Punkte des Riegl-VMX-250-Referenzsystems sowie des Velodyne VLP-16. Die vom Referenzsystem aufgenommenen Punkte sind nach ihren Höhenwerten eingefärbt, die des Velodyne-Laserscanners rot. Zu Visualisierungszwecken werden hier lediglich Punkte mit einem Höhenwert von mindestens 2 m über der Straßenebene verwendet.



(a) Nördlicher Bereich.



(b) Südlicher Bereich.

Abbildung 7.23.: Fehlerellipsen der Lokalisierung durch Scanbilder aus Messungen des Velodyne VLP-16 (Hintergrundkarte: OpenStreetMap contributors (2018)). Die Trajektorie führt durch ein Wohngebiet im Norden, entlang einer großen Verkehrsstraße sowie einen Innenstadtbereich im Süd-Osten. Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

Tabelle 7.13.: Ergebnisse der Lokalisierung über eine Scanbildkorrelation der Daten des Velodyne VLP-16 für die Position x , y sowie die Orientierung ω .

	Keine Klassifizierung	Klassifizierung
$RMSE_x$	0,19 m	0,19 m
$RMSE_y$	0,08 m	0,09 m
$RMSE_{xy}$	0,20 m	0,21 m
$RMSE_\theta$	0,60°	0,62°
Vollständigkeit	65 %	69 %

Tabelle 7.14.: Ergebnisse der Lokalisierung über eine Scanbildkorrelation der Daten des Velodyne VLP-16 nach Anwendung der Klassifizierung zur Detektion dynamischer Objekte für die Position x , y sowie die Orientierung ω in einem Wohngebiet, entlang einer großen Verkehrsstraße sowie im Innenstadtbereich.)

	Wohngebiet	Große Verkehrsstraße	Innenstadt
$RMSE_x$	0,18 m	0,26 m	0,19 m
$RMSE_y$	0,05 m	0,13 m	0,08 m
$RMSE_{xy}$	0,19 m	0,29 m	0,21 m
$RMSE_\theta$	0,36°	1,00°	0,53°
Vollständigkeit	85 %	42 %	73 %

Die Ergebnisse der Test-Trajektorie mit einer Länge von 11 km sind in Tabelle 7.13 und Abbildung 7.23 aufgeführt. Es zeigt sich, dass wiederum die Daten, in denen mithilfe des neuronalen Netzes dynamische Punkte entfernt wurden, zu besseren Ergebnissen führen. Der RMSE zur Sollposition beträgt hier 0,21 m, die Vollständigkeit 69 %. Wie schon für die aus den Riegl-Messungen erstellten Automotive-Daten in Abschnitt 7.3.1 treten Fehler insbesondere auf großen Straßen auf. Zwar ist die Anzahl der Punkte für die Daten des Velodyne-Sensors höher, jedoch können inkonsistente Intensitätswerte, insbesondere im Straßenbereich, zu Fehlern führen. Da in diesem Fall lediglich Punkte mit einer maximalen Höhe von 3 m über der Straßenebene verwendet werden, ist der Fehlereinfluss nicht detektierter Bäume in den Referenzdaten als deutlich geringer einzuschätzen.

7.3.3. Laufzeitbetrachtung

Die Laufzeit der Bestimmung des Korrelationskoeffizientens und des Klassifizierungsschrittes ist im Wesentlichen von der Anzahl der betrachteten Scanpunkte sowie der Segmentlänge, also der zuletzt zurückgelegten Strecke, und somit der Anzahl der Scans abhängig. Um die Berechnung zu beschleunigen, wird untersucht, wie sich ein Sampling der Scanpunkte sowie eine Änderung der Segmentlänge auf die Genauigkeit und Laufzeit der Daten des Einzeilenscanners (Abschnitt 7.3.1) auswirkt. Die Ergebnisse sind in Abbildung 7.24 zu finden. Die Berechnungen wurden auf einem 64-Bit-System mit einem 3,70-GHz-Prozessor i5-3570K und 8 GB Arbeitsspeicher durchgeführt.

Es ist ersichtlich, dass eine höhere Sampling-Rate sowie eine kürzere zurückgelegte Strecke zu einer geringen Verschlechterung der Genauigkeit und einer starken Verschlechterung der Vollständigkeit führen, sich jedoch wie erwartet positiv auf die Laufzeit auswirken. Wird lediglich jeder fünfte Punkt verwendet und beträgt die Segmentlänge 20 m, so kann die Korrelationsanalyse in weniger als einer Sekunde durchgeführt werden.

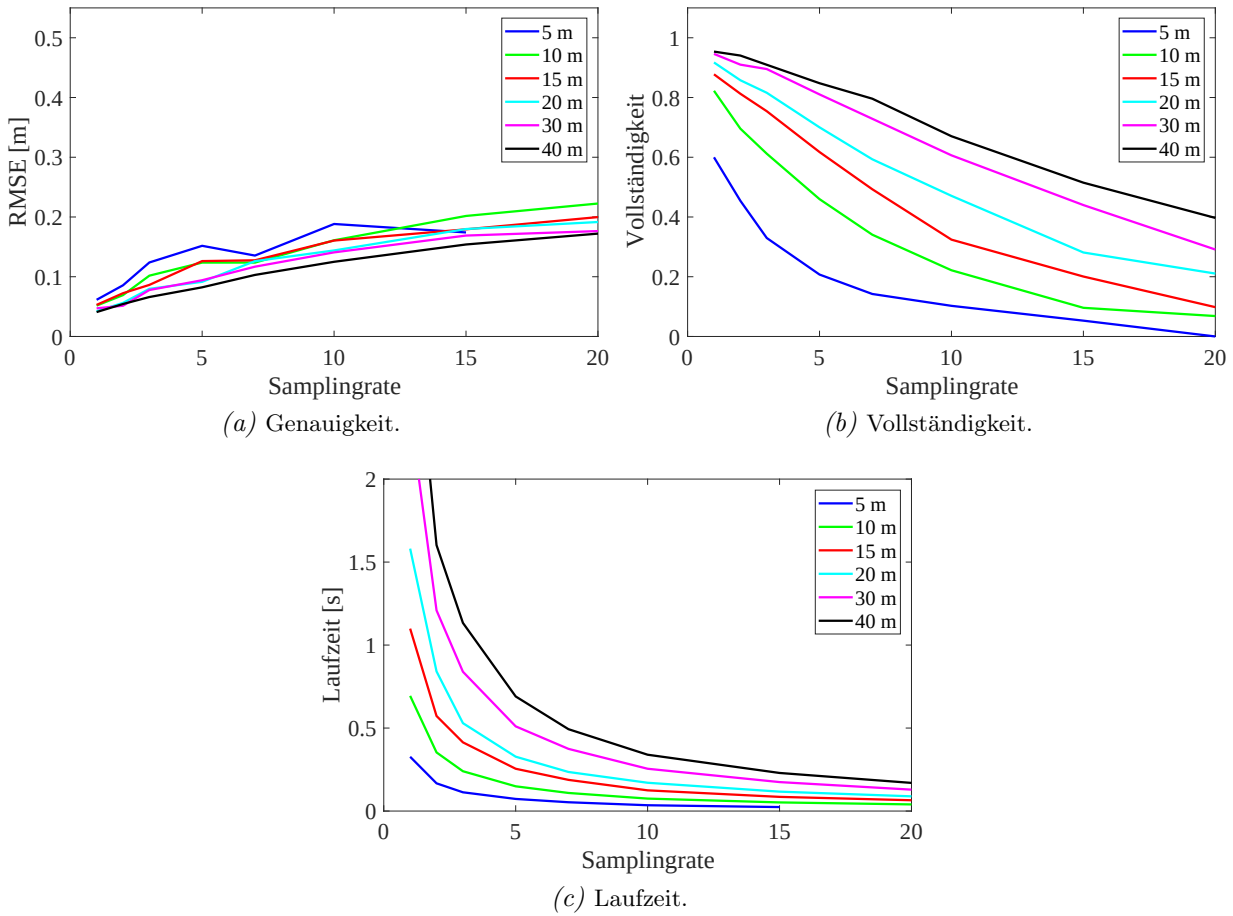


Abbildung 7.24.: Auswirkungen des Samplings während der Scanbildkorrelation auf die Genauigkeit, Vollständigkeit und Laufzeit des Verfahrens.

7.4. Diskussion der Ergebnisse

In den Abschnitten 7.1 bis 7.3 werden die Ergebnisse verschiedener Verfahren zur Fahrzeug-Lokalisierung durch Messungen von Automotive-Laserscannern ausgewertet. Hierbei werden für jedes Verfahren unterschiedliche Laserscanner verwendet. Die Ergebnisse der Methoden sind in Tabelle 7.15 zusammengefasst. Aufgrund der vergleichsweise kurzen Test- und Referenz-Trajektorie werden die Ergebnisse der Sequenzanalyse unter Verwendung der Daten des Oxford Robotcars hierbei nicht berücksichtigt.

Es zeigt sich, dass bei der Verwendung stangenförmiger Objekte und Ebenen zur Lokalisierung ein Laserscanner mit 16 Scanzeilen (Velodyne VLP-16), welcher vertikal am Fahrzeug platziert ist, deutlich bessere Ergebnisse liefert als ein horizontal montierter Laserscanner mit vier Scanzeilen (Valeo Scala). Wird die aktuelle Fahrzeugposition aus detektierten und einer Referenz zugeordneten Stangen und Ebenen bestimmt, so kann durch die Messungen des Valeo Scala zwar eine hohe Genauigkeit von 0,06 m erzielt werden, allerdings nur mit einer Vollständigkeit von 74 % bei Betrachtung aller Stangen in einem Radius von 30 m. Im Falle des vertikal montierten Velodyne VLP-16 steigt die Vollständigkeit auf 84 % bei einer mittleren Standardabweichung von 0,10 m. Hierbei werden Landmarken betrachtet, welche während einer zurückgelegten Strecke von 30 m detektiert wurden. Die niedrigere Genauigkeit kann dadurch erklärt werden, dass lediglich Landmarken gemessen werden können, welche sich in Fahrtrichtung hinter dem Fahrzeug befinden. Hierdurch kommt

Tabelle 7.15.: Erzielte Ergebnisse der Lokalisierung über detektierte stangenförmige Objekte und Ebenen, eine Sequenzanalyse sowie eine Scanbildkorrelation nach Anwendung der Klassifizierung zur Detektion dynamischer Objekte. Die Genauigkeit wird für die Verwendung stangenförmiger Objekte und Ebenen durch die mittlere Standardabweichung, für das Verfahren der Sequenzanalyse und die Scanbildkorrelation durch den RMSE-Wert in x/y-Richtung beschrieben.

	Vollständigkeit	Genauigkeit
Stangen und Ebenen (Valeo)	74 %	0,06 m
Stangen und Ebenen (Velodyne, 30 m)	84 %	0,10 m
Stangen und Ebenen (Velodyne, 50 m)	97 %	0,08 m
Sequenzanalyse (Einzeilenscanner)	99 %	1,40 m
Sequenzanalyse (Velodyne)	100 %	1,17 m
Scanbilder (Einzeilenscanner)	93 %	0,05 m
Scanbilder (Velodyne)	69 %	0,21 m

es zu einem schleifenden Schnitt, welcher sich negativ auf die Genauigkeit der Bestimmung der Koordinaten auswirkt. Wird statt der zuletzt zurückgelegten Strecke von 30 m eine Strecke von 50 m betrachtet, so steigt die Vollständigkeit auf 97 % bei einer Genauigkeit von 0,08 m.

Inbesondere für die Messungen des horizontal montierten Valeo Scala, mit einem geringen Öffnungswinkel von $3,2^\circ$, ist die Zuordnung der online detektierten Landmarken zu den entsprechenden Referenzen fehlerbehaftet. Dem Problem der fehlerhaften Zuordnung wird durch die Verwendung von Landmarken-Muster entgegengewirkt. Hierdurch kann die relative Anzahl an fehlerhaften Zuordnungen im Vergleich zu einem Nearest-Neighbor-Ansatz von 82 % auf 11 % reduziert werden. Allerdings wirkt sich dies auch negativ auf die Vollständigkeit aus, welche auf wenige Prozent sinkt. Ein weiterer Vorteil der Verwendung von Landmarken-Muster ist, dass durch den Abgleich des aktuellen Musters mit einer Referenzdatenbank die Bestimmung einer globalen Position ermöglicht wird.

Neben der Verwendung von 3D-Landmarken und einem entsprechenden Musterabgleich wird mit der Sequenzanalyse ein weiteres Verfahren zur globalen Lokalisierung vorgestellt. Für die Messungen eines Einzeilen-Laserscanners beträgt die Vollständigkeit nach einer zurückgelegten Strecke von 120 m 99 %. Mit dem Velodyne VLP-16 kann nach 120 m eine Vollständigkeit von 100 % erreicht werden. Der resultierende RMSE liegt jeweils bei weniger als 2 m. Bei einer niedrigeren zurückgelegten Strecke verringert sich auch der Wert für die Vollständigkeit. Fehlerhafte Positionsbestimmungen treten auf, wenn sich die Umgebung der Straße ändert, beispielsweise durch Baustellen oder wenn für einen Abschnitt nicht exakt dieselbe Trajektorie zur Aufnahme der Referenzdaten gefahren wurde. Der Einfluss der Anzahl an verwendeten Clustern sowie der Länge des Merkmalsvektors, welcher sich aus dem verwendeten neuronalen Netz ergibt, ist gering. Dennoch bewirkt die Verwendung der Merkmale neben einer Reduzierung der Dimension, dass die Vollständigkeit im Vergleich zur Verwendung der ursprünglichen Messungen des Laserscanners steigt. Dies kann dadurch begründet werden, dass lediglich relevante Merkmale zur Bestimmung der Clusterzugehörigkeit eines Scans verwendet werden.

Die Ergebnisse der Sequenzanalyse können als Initialwerte für eine Positionsbestimmung über Scanbilder dienen. Durch die Verwendung einer Scanbildkorrelation lassen sich sehr hohe Genauigkeiten von bis zu 0,05 m erzielen. Die Vollständigkeit des Verfahrens beträgt bei der Verwendung der Messungen eines Einzeilenscanners, die aus den Daten eines Mobile-Mapping-Systems erstellt

werden, 93 %. Hierbei wird eine Klassifizierung dynamischer Punkte vorgenommen, welche eine Steigerung der Genauigkeit von 5 % für die Daten eines Einzeilenlaserscanners und von 4 % für die Daten eines Velodyne VLP-16 bewirkt. Fehler treten insbesondere in Regionen mit einer geringen Bebauung und mit wenigen Intensitätsunterschieden, beispielsweise in Form von Fahrbahnmarkierungen, auf. Die Vollständigkeit der Messungen des Velodyne VLP-16 beträgt 75 % bei einer Genauigkeit von 0,10 m. Die schlechteren Ergebnisse für die Messungen des Velodyne-Sensors lassen sich vor allem auf die Verwendung zweier unterschiedlicher Scansysteme im Lokalisierungsschritt und während der Erstellung der Referenz-Scanbilder zurückführen. Während die Einzeilen-Laserscans mit demselben System wie zur Erstellung der Referenzen aufgenommen wurden, unterscheidet sich der Velodyne-Laserscanner in der Anbringung am Fahrzeug. Durch die Anbringung an der Anhängerkupplung des Fahrzeuges treten häufiger Verdeckungen auf, beispielsweise durch Fahrzeuge. Weiterhin sind die Intensitätswerte des reflektierten Laserimpulses nicht kalibriert und weisen im Gegensatz zu den Messungen des Riegl VMX-250 inkonsistente Werte auf.

Im Kontext des autonomen Fahrens wird eine Positionslösung mit einer Genauigkeit von etwa 0,1 m benötigt (Levinson u. a., 2007). Durch die Lokalisierung über stangenförmige Objekte und Ebenen sowie über Scanbilder kann diese Anforderung erreicht werden. Die Genauigkeiten liegen im Bereich des aktuellen Stands der Forschung (siehe Abschnitt 3.2). Im Vergleich zu den in Rohde u. a. (2016), Qu u. a. (2018), Levinson u. a. (2007) sowie Kim u. a. (2017) präsentierten Verfahren werden neben den LiDAR-Messungen lediglich die Messungen einer Inertialeinheit jedoch keines GNSS-Sensors benötigt. Stattdessen kann eine globale Position durch das Verfahren der Sequenzanalyse bereitgestellt werden. Die Vollständigkeit vergleichbarer Verfahren zur globalen Positionsbestimmung, wie in Tipaldi u. a. (2013) und Himstedt u. a. (2014) präsentiert, ist mit maximal 93 % deutlich geringer als die hier vorgestellte Methode. Der in Brubaker u. a. (2016) vorgestellte Ansatz zur Lokalisierung über einen Abgleich der gefahrenen Trajektorie mit OpenStreetMap-Daten benötigt etwa eine Minute Fahrzeit, um die Position in einem Gebiet mit einer Größe von 1 km² zu bestimmen. Bei einer Fahrtgeschwindigkeit von 15 m/s entspricht dies einer Strecke von 900 m. In dem hier vorgestellten Ansatz ist eine korrekte Positionsbestimmung bereits nach etwa 120 m mit einer Vollständigkeit von 100 % möglich, wobei die Größe des Referenzgebietes höher ist.

8. Zusammenfassung und Ausblick

8.1. Zusammenfassung

In der vorliegenden Arbeit werden Methoden zur Lokalisierung von Fahrzeugen in urbanen Gebieten durch Automotive-Laserscanner untersucht und miteinander verglichen. Um eine hochgenaue Positionslösung zuverlässig zu ermöglichen, bedarf es redundanter Systeme zur Positionsbestimmung. Aus diesem Grund werden hier Methoden aufgezeigt, die eine korrekte Position durch die Auswertung von LiDAR-Daten bestimmen können, auch wenn keine GNSS-Lösung bereitgestellt werden kann.

Die Funktionsweise von Laserscannern sowie die hierfür benötigten Algorithmen zur Verarbeitung der LiDAR-Daten wurden in Kapitel 2 vorgestellt. Hierbei wurde neben der Sensorik auf die Punktwolkenregistrierung, räumliche Datenstrukturen für einen effizienten Zugriff auf Punkte sowie auf Segmentierungsalgorithmen eingegangen. Eine Segmentierung wurde in dieser Arbeit zur Einteilung der Punktwolken in einzelne, freistehende Objekte sowie zur Detektion stangenförmiger Objekte und Ebenen verwendet. Weiterhin wurden verschiedene Methoden des maschinellen Lernens vorgestellt: der K-Means-Algorithmus, welcher dem Clustering von Daten dient, Random Forests zur Klassifizierung von Daten anhand bestimmter Merkmale sowie neuronale Netze, welche in der Arbeit sowohl zur Detektion dynamischer Objekte in Distanz- und Intensitätsbildern der Laserscans sowie zur Dimensionsreduzierung der Scans verwendet wurden.

Eine Übersicht über den aktuellen Stand der Forschung liefert Kapitel 3. Es wurden Arbeiten zur Detektion dynamischer Objekte in LiDAR-Daten anhand einer Änderungsdetektion sowie durch verschiedene Klassifizierungsalgorithmen aufgeführt. Weiter wurde auf Methoden zur Fahrzeuglokalisierung eingegangen. Hierbei zeigte sich, dass verschiedene Verfahren eine Lokalisierungsgenauigkeit von etwa 0,1 m erzielen, hierbei jedoch stets eine Kombination aus bildgebenden Sensoren mit GNSS-Messungen zum Einsatz kommt. GNSS-unabhängige Lösungen erzielen eine verhältnismäßig geringe Vollständigkeit von maximal 85 % oder benötigen eine Messung mit einer Dauer von mindestens einer Minute zur globalen Positionsbestimmung.

In der Arbeit wurden unterschiedliche Automotive-Laserscanner sowie ein Mobile-Mapping-System (Riegl VMX-250) verwendet. Neben der Erstellung von Referenzdaten diente das Riegl VMX-250 auch zur Erzeugung simulierter Daten eines Einzeilen-Laserscanners. Als weitere Automotive-Laserscanner kamen ein horizontal montierter Valeo Scala sowie ein vertikal montierter SICK LMS151 und Velodyne VLP-16 zum Einsatz. Eine detaillierte Beschreibung der Sensoren ist in Kapitel 4 zu finden.

Es wurden drei unterschiedliche Methoden zur Fahrzeug-Lokalisierung vorgestellt: eine Lokalisierung durch segmentierte stangenförmige Objekte und Ebenen, durch eine Sequenzanalyse aufeinanderfolgender Scans zur Bestimmung einer globalen Position und durch eine Scanbildkorrelation. Für jedes Verfahren wurden außerdem unterschiedliche Automotive-Laserscanner getestet. In der Arbeit wurden zur Bewertung der Verfahren die Genauigkeit sowie die Vollständigkeit analysiert. Die Vollständigkeit ergibt sich hierbei aus der prozentualen Anzahl an Standpunkten entlang einer Test-Trajektorie, an denen eine Position erfolgreich bestimmt werden kann.

Zur Lokalisierung über stangenförmige Objekte und Ebenen wurden die Landmarken zunächst in den Daten eines Mobile-Mapping-Systems detektiert und in eine Referenzkarte geschrieben.

Diese Objekte wurden ebenfalls aus Messungen eines Automotive-Laserscanners segmentiert und den Referenzen zugeordnet. Lagen in einem bestimmten Bereich um den aktuellen Standpunkt mindestens drei detektierte und einer Referenz zugeordneten Landmarken vor, so wurde die Fehlerellipse der Positionslösung berechnet. Zur Auswertung der Ergebnisse wurden zwei unterschiedliche Automotive-Laserscanner verwendet: ein auf einen Sichtbereich von $3.2 \times 125^\circ$ begrenzter horizontaler Laserscanner (Valeo Scala), welcher beispielsweise auch zur Detektion vorausfahrender Fahrzeuge dienen kann, sowie ein vertikal montierter Laserscanner (Velodyne VLP-16) mit einem Sichtbereich von $15 \times 360^\circ$. Es zeigte sich, dass die Anzahl an fehlerhaften Detektionen von Landmarken durch die Verwendung des vertikal montierten Scanners von 68 % auf 39 % reduziert werden kann. Die Genauigkeit der Positionslösung lag unter der Betrachtung aller Landmarken in einem Radius von 30 m für den horizontal montierten Valeo Scala bei 0,06 m, die Vollständigkeit bei 74 %. Für den Velodyne VLP-16 ergaben sich die Werte zu 0,10 m und 84 %, sofern Landmarken berücksichtigt werden, welche entlang der zuletzt zurückgelegten 30 m aufgenommen wurden. Wird die zurückgelegte Strecke auf 50 m erhöht, so steigern sich Genauigkeit und Vollständigkeit auf 0,08 m und 97 %.

Zusätzlich wurde zur globalen Lokalisierung in urbanen Gebieten eine Sequenzanalyse angewandt. Hierfür wurde eine Referenzkarte erstellt, indem jedem Scan eines vertikal montierten Laserscanners durch einen K-Means-Algorithmus entlang einer gefahrenen Referenztrajektorie ein Label zugeordnet wurde. Dasselbe Clustering wurde verwendet, um den Scans im Lokalisierungsschritt ebenfalls ein Label zuzuordnen. Durch einen Abgleich der zuletzt aufgenommenen Scans mit der Referenzkarte konnte anschließend die aktuelle Position bestimmt werden. Als Laserscanner dienten ein aus den Daten eines Mobile-Mapping-Systems (Riegl VMX-250) simulierter Einzeilen-Laserscanner, ein tatsächlicher Einzeilen-Laserscanner des Oxford Robotcars (SICK LMS151) sowie ein Velodyne VLP-16. Anstatt der gesamten Laserscans gingen durch ein neuronales Netz bestimmte Merkmale in das Clustering ein, was die Dimension der Scans deutlich reduziert. Die Experimente zeigten, dass nach einer gefahrenen Strecke von 120 m die korrekte Position auf weniger als 2 m genau mit einer Vollständigkeit von 99 % für die simulierten Messungen eines Einzeilenscanners und mit einer Vollständigkeit von 100 % für die Daten des Velodyne VLP-16 bestimmt werden kann. Das Verfahren setzt voraus, dass im Lokalisierungsschritt eine bereits in den Referenzdaten enthaltene Strecke befahren wird. Bei einer Änderung der Fahrspur liefert die Methode weiterhin korrekte Ergebnisse. Wird die Strecke innerhalb eines Abschnitts jedoch beispielsweise durch Wendemanöver verlängert, so kann dies zu einer fehlerhaften Positionsbestimmung führen. Eine weitere Fehlerquelle sind Veränderungen der Fahrzeugumgebung, beispielsweise verursacht durch Baustellen.

Zur Feinlokalisierung wurden mithilfe der Daten eines Mobile-Mapping-Systems Referenz-Scanbilder erzeugt, welche die Höhen- und Intensitätswerte der LiDAR-Messungen auf der x/y -Ebene aufrastern. Um lediglich statische Objekte zu berücksichtigen, wurden Dynamiken über eine Änderungsdetektion mehrerer Mobile-Mapping-Datensätze detektiert und schließlich aus den Referenzdaten extrahiert. Über eine Korrelation mit den Daten eines vertikal montierten Automotive-Laserscanners wurde entlang einer Test-Trajektorie die Position bestimmt. Um hierbei ebenfalls dynamische Objekte zu extrahieren, wurden Methoden des maschinellen Lernens angewandt. Für die simulierten Messungen eines Einzeilen-Laserscanners kam ein Random Forest zum Einsatz, im Falle der Messungen des Velodyne VLP-16 ein neuronales Netz. Es zeigte sich, dass für beide Automotive-Laserscanner eine Lokalisierung auf weniger als 0,1 m genau durchgeführt werden kann. Die Vollständigkeit des Verfahrens lag hierbei für den Einzeilen-Laserscanner bei 93 %, für den Velodyne VLP-16 bei 75 %. Wie bereits beim Verfahren der Sequenzanalyse können Veränderungen des Straßenraums zwischen dem Zeitpunkt der Aufnahme der Referenzdaten und dem Lokalisierungsschritt zu Fehlern in der Positionsbestimmung führen. Weitere Fehler können in Bereichen entstehen, in denen wenig Strukturen, wie Gebäude oder Fahrbahnmarkierungen, zu finden sind. Zusätzlich weisen die Messungen des verwendeten Velodyne VLP-16 inkonsistente Intensitätswerte

in einem etwa 2 m breiten Bereich entlang der Fahrzeug-Trajektorie auf, welche vermutlich auf die an dieser Stelle niedrige Distanz des Sensors zur Fahrbahn von 0,5 m zurückzuführen sind.

Die Ergebnisse zeigen, dass sich die Verfahren eignen, um zur Positionsbestimmung in zukünftigen Fahrerassistenzsystemen und im Kontext des autonomen Fahrens eingesetzt zu werden. Die Verwendung stangenförmiger Objekte und Ebenen sowie die Scanbildkorrelation können genutzt werden, um in Kombination mit bereits existierenden Methoden die Genauigkeit und Zuverlässigkeit der Fahrzeug-Lokalisierung zu erhöhen. Kann keine genaue GNSS-Lösung gewährleistet werden, beispielsweise aufgrund hoher Straßenschluchten im städtischen Bereich, so kann eine globale Positionsbestimmung durch eine Sequenzanalyse mit einer Genauigkeit von weniger als 2 m zuverlässig durchgeführt werden.

8.2. Ausblick

In dieser Arbeit werden die einzelnen Methoden hinsichtlich ihrer Genauigkeit und Zuverlässigkeit weitestgehend unabhängig voneinander analysiert. Eine Kombination aller drei Methoden in einem Filteransatz könnte zu einer weiteren Steigerung der Genauigkeit und Zuverlässigkeit führen. Um aus den Laserscans des Velodyne VLP-16, welcher für alle drei vorgestellten Methoden verwendet wird, stangenförmige Objekte und Ebenen zu segmentieren und eine Scanbildkorrelation durchzuführen, müssen die Änderungen in der Lage, Höhe und Orientierung jederzeit bekannt sein. Für eine von GNSS unabhängige Lösung kann die Änderung in der Pose zwischen zwei Punktmessungen beispielsweise aus den Daten einer Inertialeinheit gewonnen werden. In dieser Arbeit werden zur Bestimmung der Pose die Messungen eines Mobile-Mapping-Systems verwendet. Die Daten ergeben sich hierbei aus einem Filteransatz, in dem ein Differential-GNSS-System mit einer Inertialeinheit und einem DMI kombiniert sind. Die Rohdaten der einzelnen Sensoren stehen nicht zur Verfügung, sodass in diesem Fall keine von GNSS unabhängige Lösung vorliegt. Die ebenfalls an der Messplattform befestigte IMU Xsens MTi-G liefert zwar Werte für die aktuelle Geschwindigkeit und Orientierung, es zeigte sich allerdings, dass diese Messungen zu ungenau sind, um eine verwertbare Punktwolke aus den einzelnen Laserscans zu erstellen. Für die Kombination der Verfahren in einem Filteransatz bedarf es entsprechend einer Inertialeinheit einer höheren Genauigkeitsklasse im Vergleich zur hier verwendeten Xsens MTi-G, welche unabhängig von GNSS-Messungen die entsprechenden Messungen ausgibt.

Die Referenzdaten stammen im Falle der Lokalisierung über stangenförmige Objekte und Ebenen sowie der Scanbildkorrelation aus den Messungen eines hochgenauen Mobile-Mapping-Systems. Dies hat zum Nachteil, dass jeder Bereich, auf dem eine Positionsbestimmung möglich sein soll, mit diesem System befahren werden muss. Um Änderungen zu detektieren, müssen die Fahrten zudem regelmäßig erfolgen. Alternativ könnte die Gewinnung der Referenzdaten sowie deren Aktualisierung auch über einen Crowd-Sensing-Ansatz erfolgen. Hierbei müsste eine bestimmte Anzahl an Fahrzeugen, welche bereits mit Laserscannern ausgestattet sind, eine Kartierung des Raumes vornehmen und die entsprechenden Daten an einen zentralen Server senden. Zur gleichzeitigen Positionsbestimmung und Kartierung können SLAM-Verfahren (Simultaneous Localization and Mapping) verwendet werden. Die auf dem Server erstellten oder aktualisierten Referenzdaten können anschließend, beispielsweise über Mobilfunk, an die entsprechenden Fahrzeuge gesendet werden.

Die Sequenzanalyse setzt in dieser Arbeit voraus, dass die gefahrene Trajektorie der aktuellen Sequenz entlang derselben Route ebenfalls in den Referenzdaten vorliegt. Dies hat zur Folge, dass beispielsweise zur Lokalisierung nach einer Kreuzung alle möglichen Abbiegemöglichkeiten aus jeder Richtung befahren werden müssen, um die Referenzkarte zu erstellen. Der Aufwand zur

Erstellung der Referenzkarte kann verringert werden, wenn zum Abgleich der aktuellen Sequenz mit der Referenz ein graphenbasierter Ansatz verfolgt wird, in dem die einzelnen Teilstücke zu einer Referenzsequenz kombiniert werden können. Um das Problem der Verschiebung der Labels aufgrund einer Verlängerung oder Verkürzung der gefahrenen Strecke zu verringern, könnten die einzelnen Sequenzen benachbarter Kanten des Graphs unter Zunahme einer Puffersequenz mit einer variablen Länge oder mit einer entsprechenden Verkürzung der Sequenzen zusammengefügt werden.

Für eine weltweite, globale Positionierung durch eine Sequenzanalyse mit einer anschließenden Feinlokalisierung über stangenförmige Objekte und Ebenen oder Scanbilder würden sowohl der benötigte Speicherbedarf für die Referenzkarte als auch die Laufzeit der Verfahren stark ansteigen. Somit müsste die Referenzkarte auf einen Bereich von beispielsweise wenigen Kilometern beschränkt werden. In der Regel kann davon ausgegangen werden, dass bekannt ist, in welchem Bereich sich das Fahrzeug befindet. Hierzu kann die zuletzt bestimmte Position verwendet, die jeweilige Stadt oder der entsprechende Stadtteil vom Fahrer eingegeben werden oder die Positionsbestimmung über Mobilfunksignale erfolgen. Weiterhin werden sowohl die Parameter des neuronalen Netzes zur Bestimmung der Merkmale, die in den K-Means-Algorithmus eingehen, als auch deren Clustermittelpunkte anhand von Trainingsdaten eines speziellen Bereiches bestimmt. Es ist nicht davon auszugehen, dass diese auch für unterschiedliche Städte oder Länder zu ebenfalls guten Ergebnissen führen. Somit müssten für die jeweiligen Gebiete eigene Netze trainiert werden. Ebenso müssten anschließend die sich hieraus für einen Trainingsdatensatz ergebenden Clustermittelpunkte des K-Means-Algorithmus für diese Gebiete erstellt werden. Ist der entsprechende Bereich, in dem sich das Fahrzeug befindet, bekannt, so können die entsprechende Referenzkarte sowie die Parameter des neuronalen Netzes und des K-Means-Algorithmus geladen werden. Das Laden der Daten kann beispielsweise während der Fahrt über Mobilfunk oder regelmäßig von einem festen Standpunkt aus erfolgen. Somit ist anschließend eine auf wenige Meter genaue Positionsbestimmung mit anschließender Feinlokalisierung möglich.

Anhang

A. Ergebnisse der Lokalisierung über stangenförmige Objekte und Ebenen

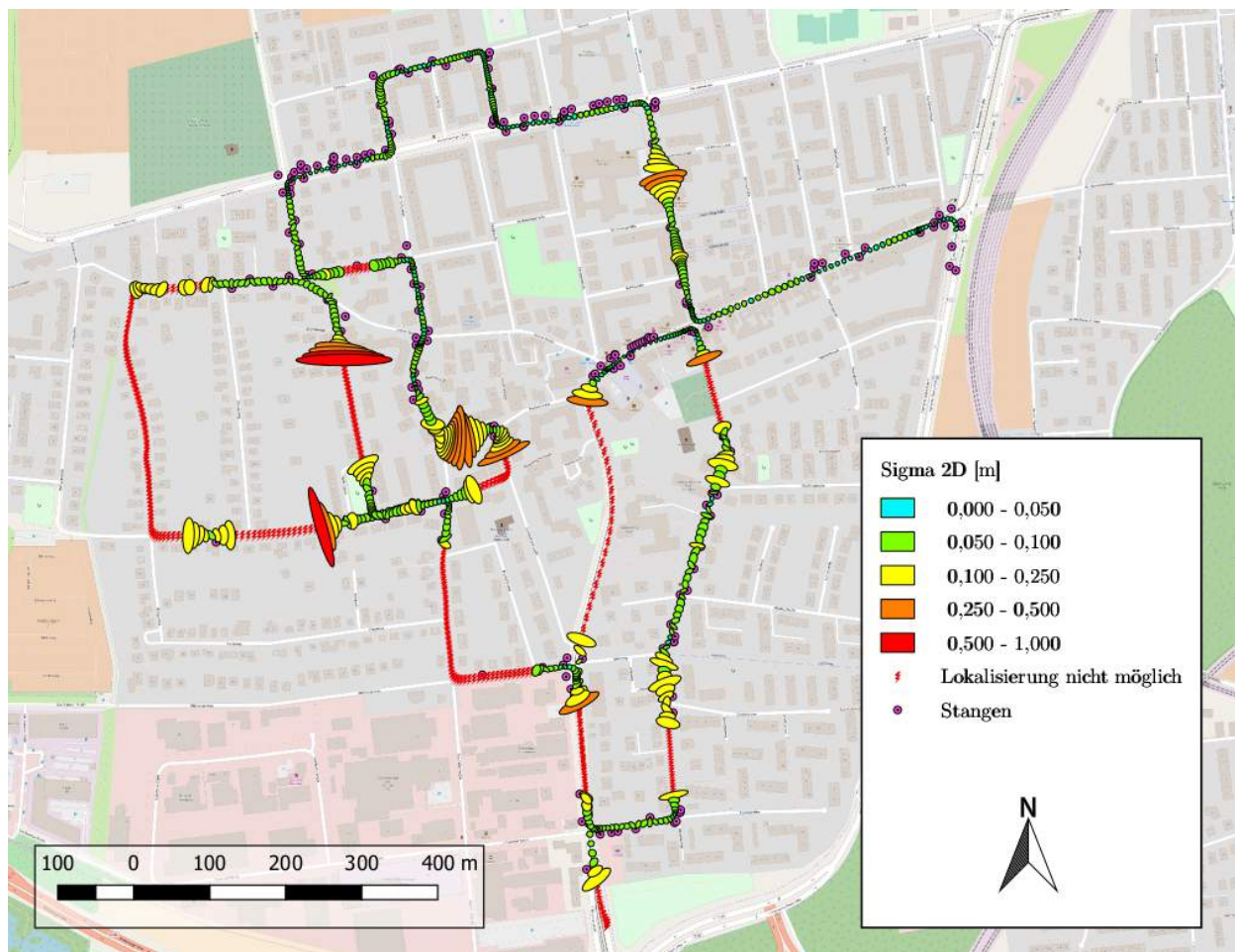


Abbildung A.1.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von **Stangen**, detektiert durch einen horizontal montierten **Valeo Scala** (Hintergrundkarte: OpenStreetMap contributors (2018)). Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

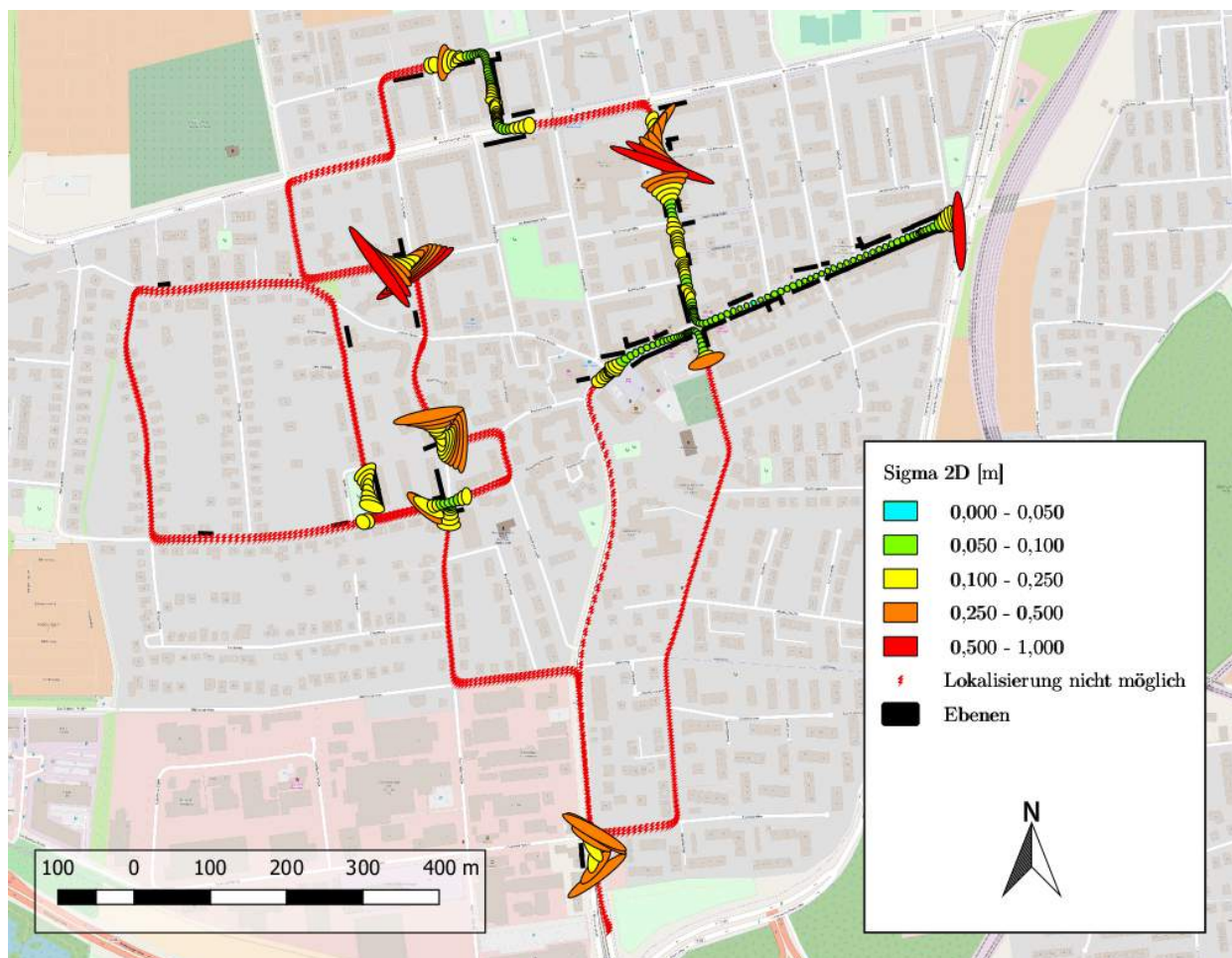


Abbildung A.2.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von **Ebenen**, detektiert durch einen horizontal montierten **Valeo Scala** (Hintergrundkarte: OpenStreetMap contributors (2018)). Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

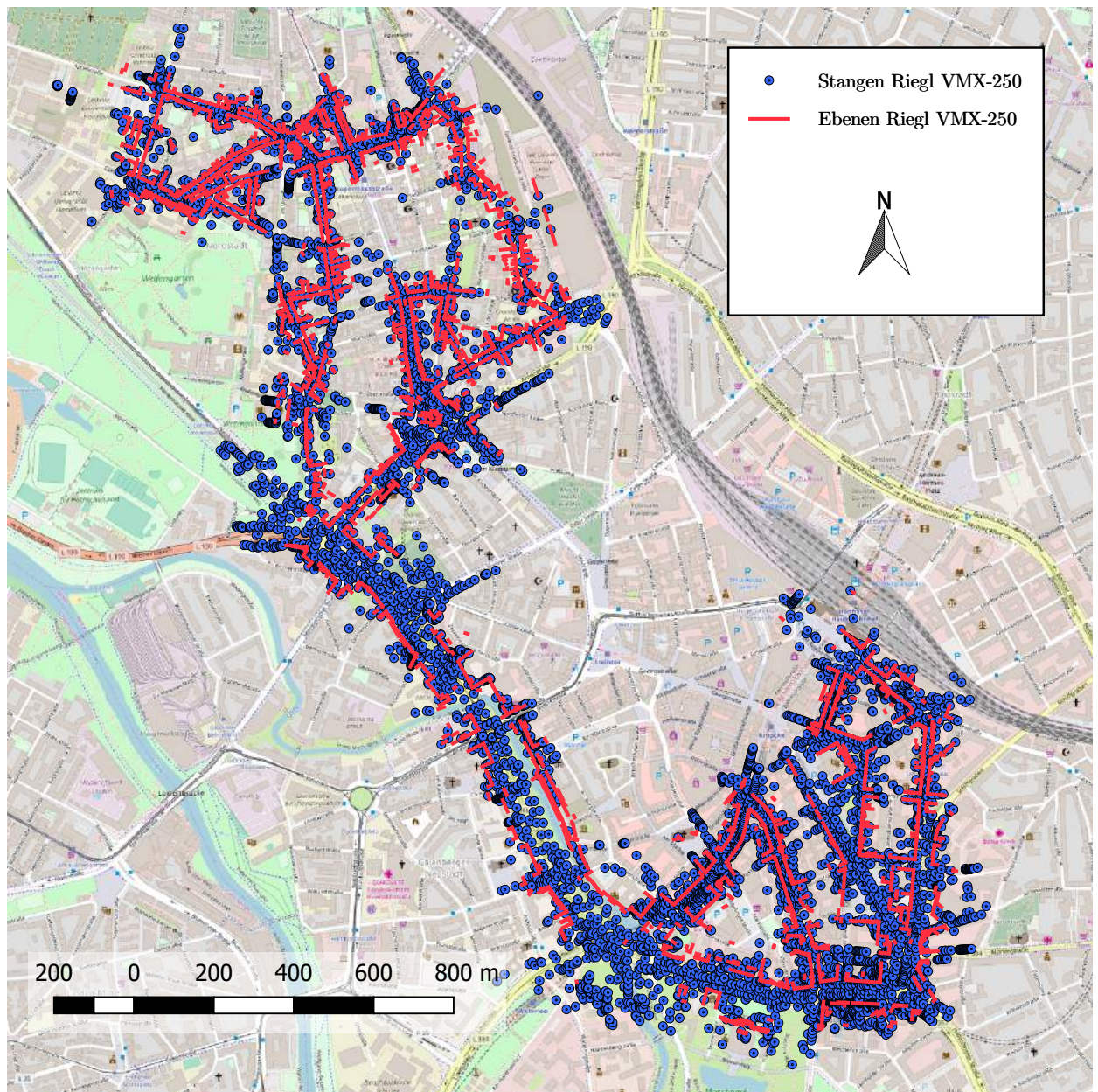
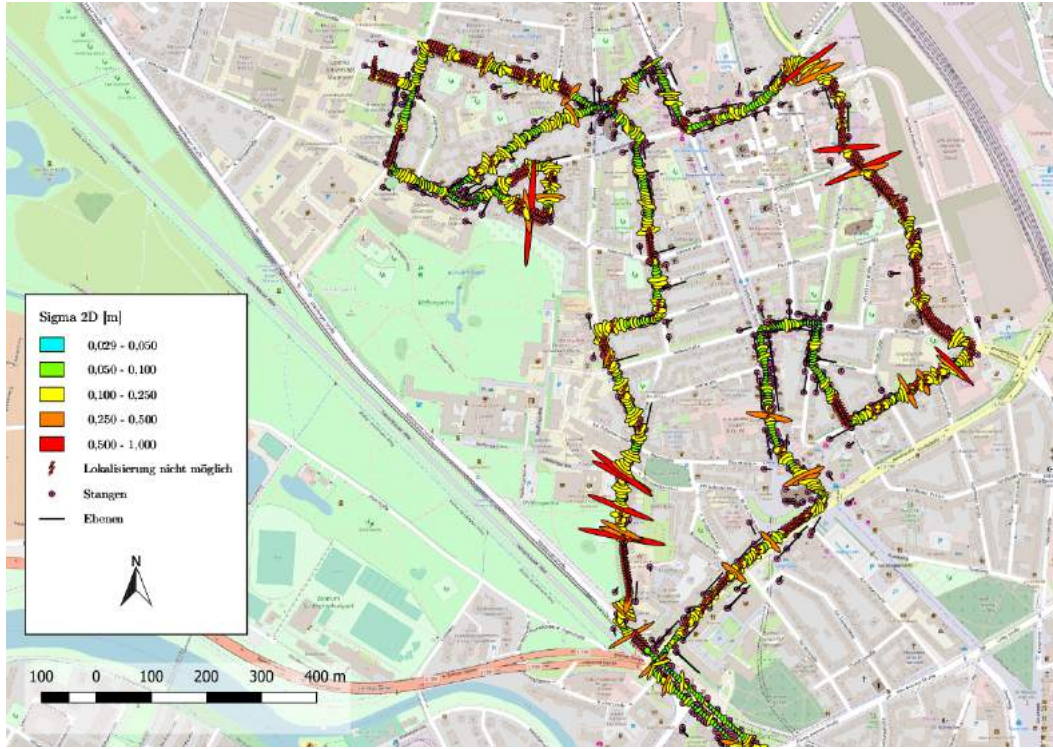
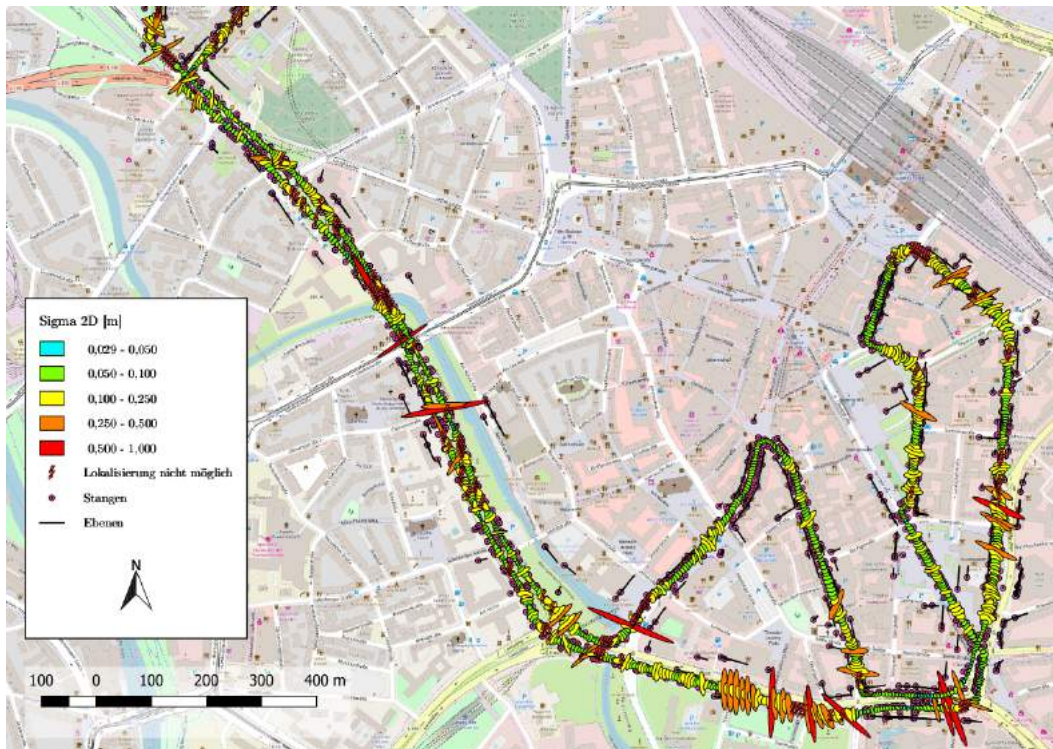


Abbildung A.3.: Aus den Messungen des Mobile-Mapping-Systems extrahierte stangenförmige Objekte (blau) und Ebenen (rot) in der Innenstadt von Hannover, entlang einer großen Verkehrsstraße sowie im Stadtteil Nordstadt (Hintergrundkarte: OpenStreetMap contributors (2018)).

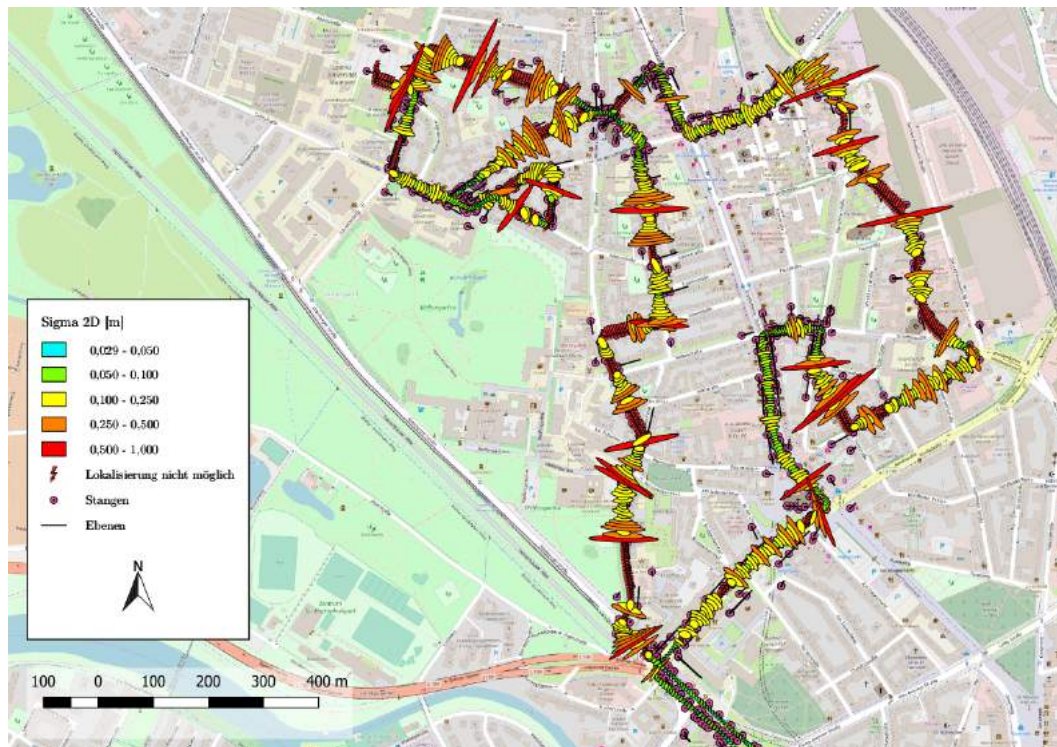


(a) Nördlicher Bereich.

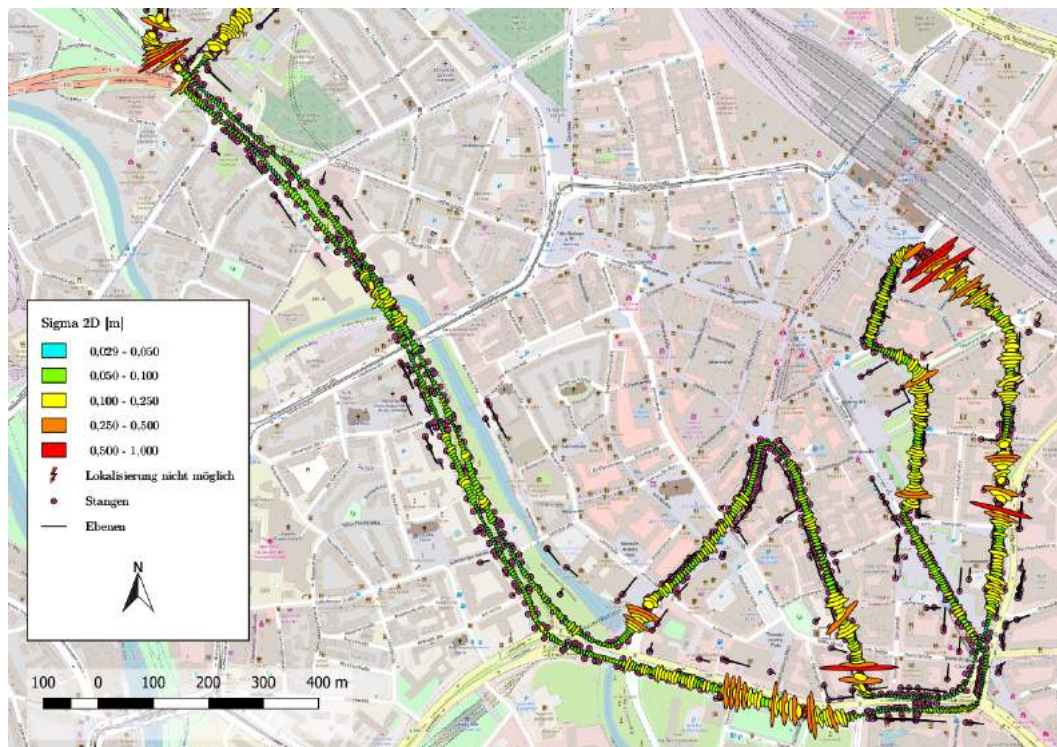


(b) Südlicher Bereich.

Abbildung A.4.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der gemeinsamen Verwendung von **Stangen** und **Ebenen**, detektiert durch einen vertikal montierten **Velodyne VLP-16**. Die Orientierung des Sensors wird aus den Messungen der **Inertialeinheit des Riegl VMX-250** bestimmt, die zurückgelegte Distanz beträgt $s_{\max} = 30 \text{ m}$. Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet (Hintergrundkarte: OpenStreetMap contributors (2018)).

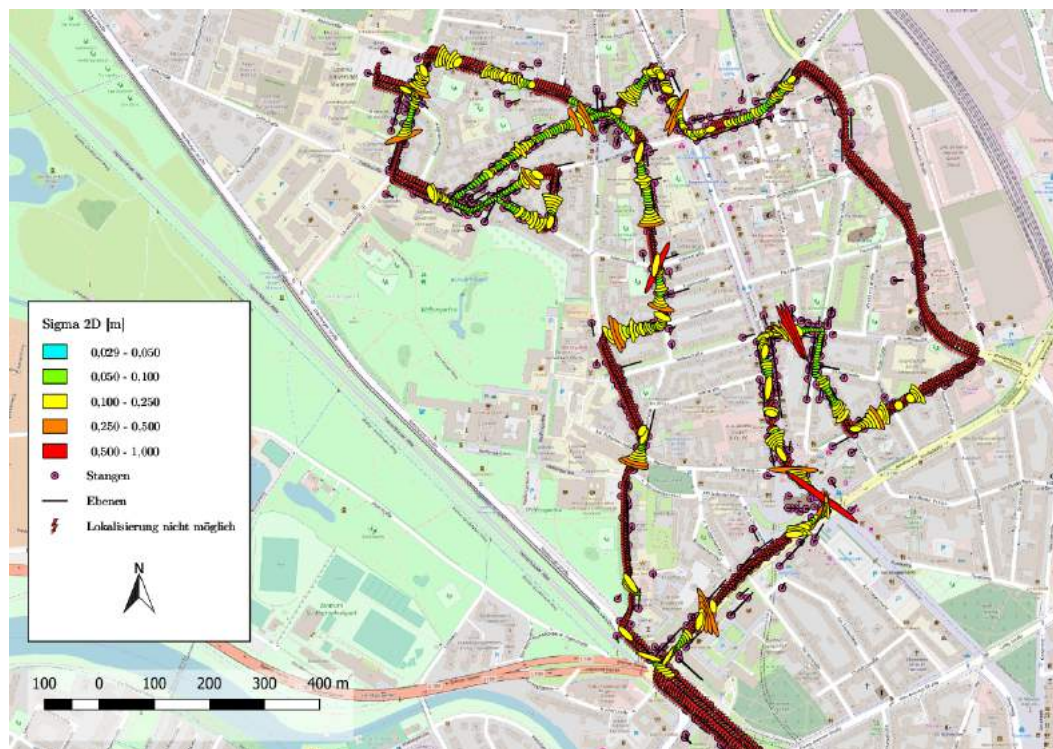


(a) Nördlicher Bereich.

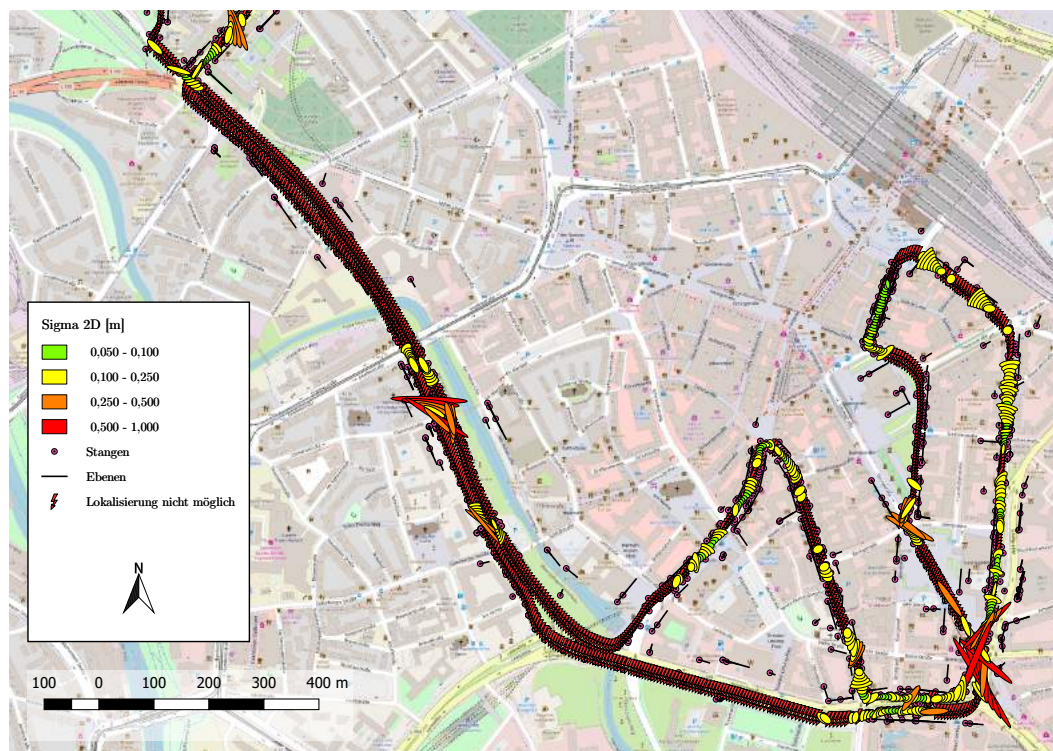


(b) Südlicher Bereich.

Abbildung A.5.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von **Stangen**, detektiert durch einen vertikal montierten **Velodyne VLP-16** (Hintergrundkarte: OpenStreetMap contributors (2018)). Die Orientierung des Sensors wird aus den Messungen der **Inertialeinheit des Riegl VMX-250** bestimmt, die zurückgelegten Distanz beträgt $s_{\max} = 50 \text{ m}$. Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

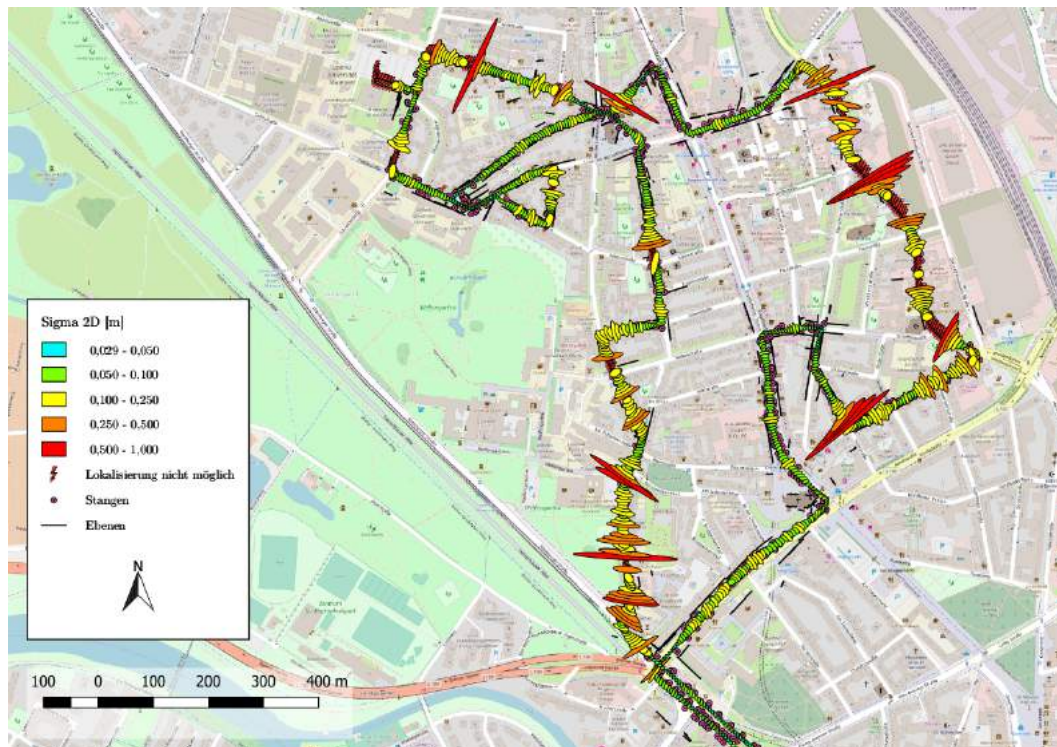


(a) Nördlicher Bereich.

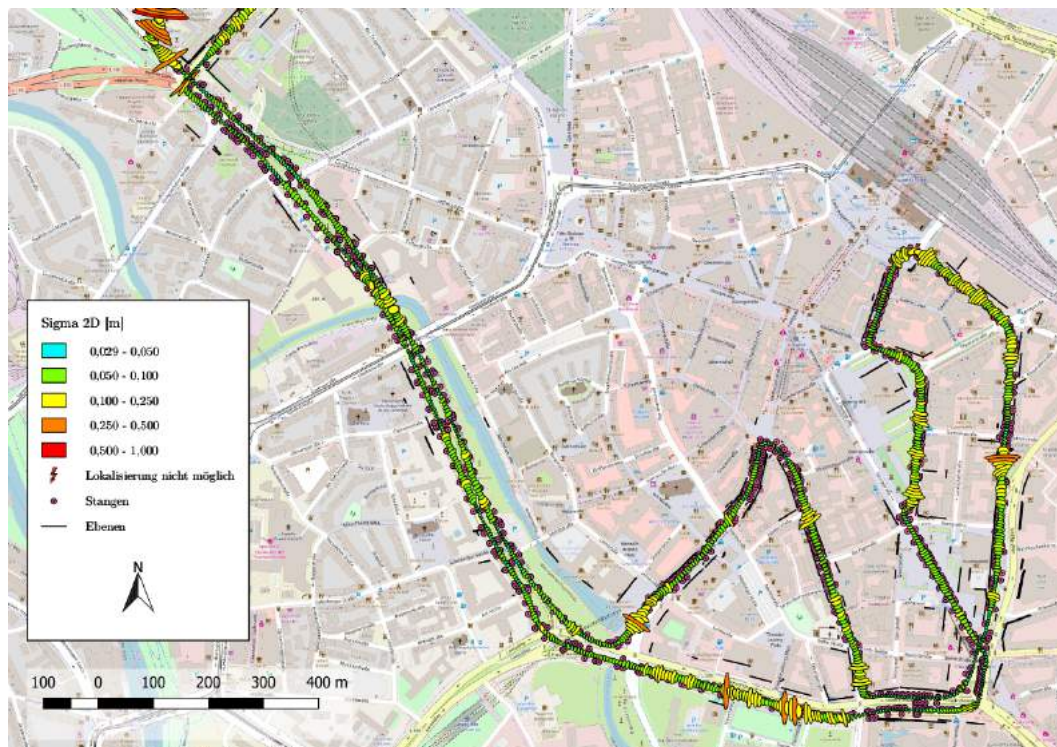


(b) Südlicher Bereich.

Abbildung A.6.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung **Ebenen**, detektiert durch einen vertikal montierten **Velodyne VLP-16** (Hintergrundkarte: OpenStreetMap contributors (2018)). Die Orientierung des Sensors wird aus den Messungen der **Inertialeinheit des Riegl VMX-250** bestimmt, die zurückgelegte Distanz beträgt $s_{\max} = 50 \text{ m}$. Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.



(a) Nördlicher Bereich.



(b) Südlicher Bereich.

Abbildung A.7.: Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der gemeinsamen Verwendung von **Stangen** und **Ebenen**, detektiert durch einen vertikal montierten **Velodyne VLP-16** (Hintergrundkarte: OpenStreetMap contributors (2018)). Die Orientierung des Sensors wird durch die **Xsens MTi-G** bestimmt, die zurückgelegte Distanz beträgt $s_{\max} = 50 \text{ m}$. Positionen, an denen keine Ergebnisse vorliegen, sind durch ein rotes Symbol gekennzeichnet.

B. Ergebnisse der Sequenzanalyse durch einen Velodyne VLP-16

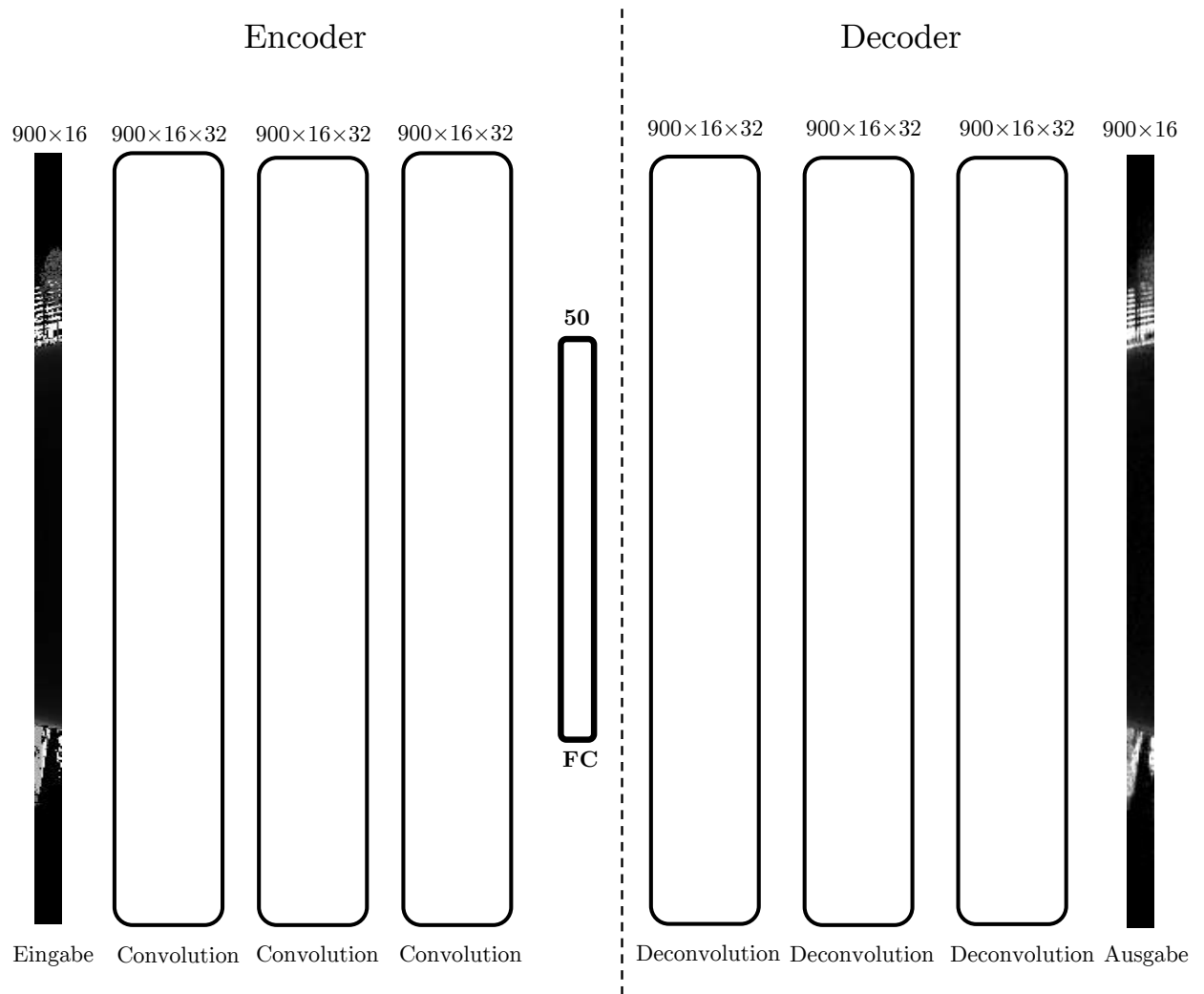


Abbildung B.1.: Beispielhafter Aufbau des Autoencoders für die LiDAR-Daten des Velodyne VLP-16. Auf die Eingangsdaten werden zwei Faltungen angewendet, wobei im Anschluss an jede Convolution gegebenenfalls ein Max Pooling durchgeführt wird. Anschließend wird die Dimension in dem folgenden fully connected Layer in diesem Fall auf 50 Merkmale reduziert. In den folgenden Layern des Decoders werden die ursprünglichen Eingangsdaten rekonstruiert.

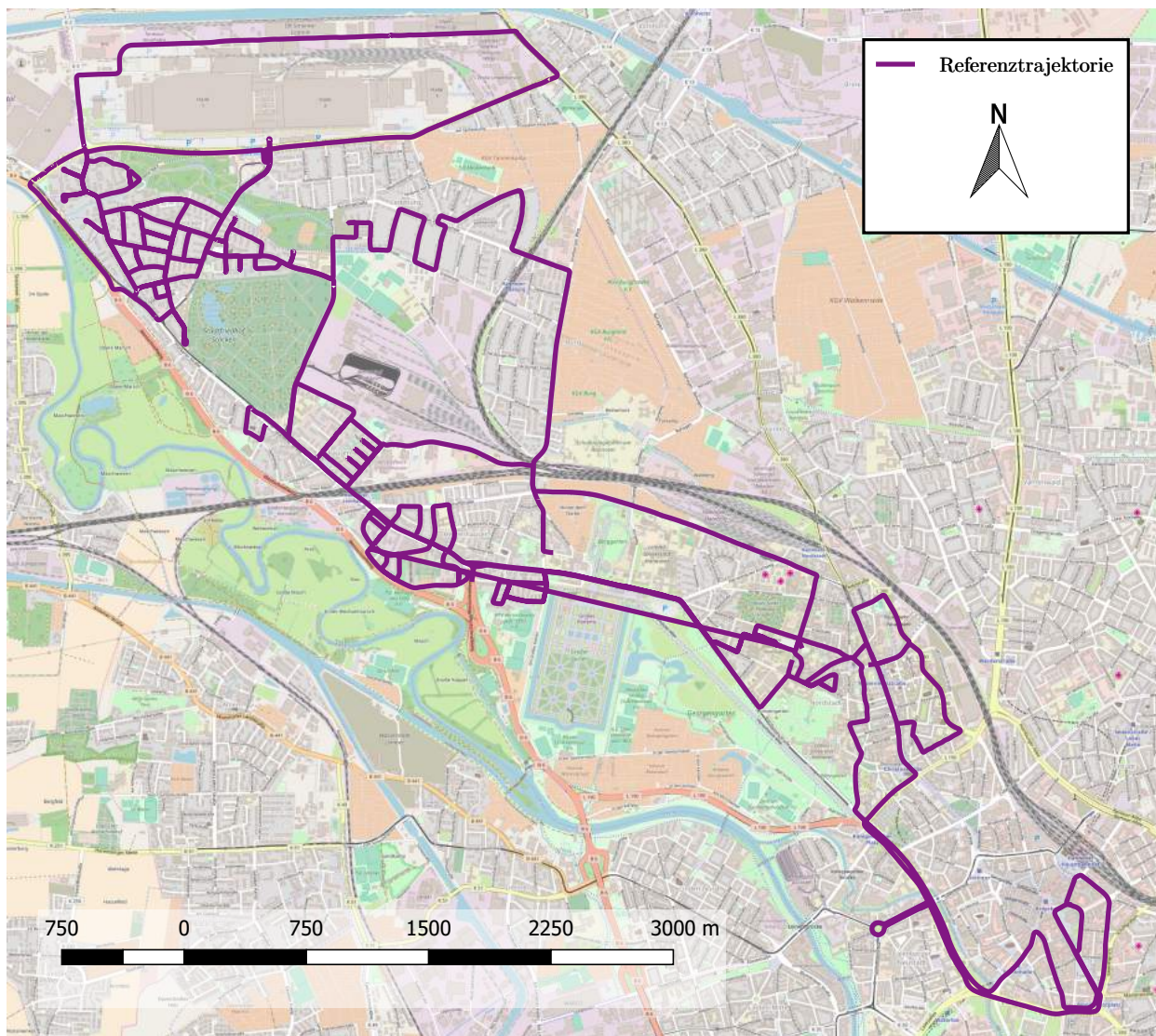


Abbildung B.2.: Referenztrajektorie der Sequenzanalyse für den Datensatz des Velodyne VLP-16 (Hintergrundkarte: OpenStreetMap contributors (2018)).

Tabelle B.1.: Vollständige Auflistung der Auswirkungen veränderter Parameter des neuronalen Netzes bei einer Sequenzlänge von $n_{seq} = 200$ und einer Anzahl an Cluster-Mittelpunkten von $k = 1000$ auf die Lokalisierungs-Vollständigkeit.

Convo- lution	FCL	Pooling	n_{Merkmale}	Kernel- größe	Merkmals- karten	Eingangsdaten	VS [%]
2	1	Nein	20	3	32	Dist.	93,1
2	1	Nein	50	3	32	Dist.	96,6
2	1	Nein	100	3	32	Dist.	95,9
2	1	Nein	200	3	32	Dist.	94,3
2	1	Nein	400	3	32	Dist.	82,0
2	1	Nein	20	3	32	Int.	35,9
2	1	Nein	50	3	32	Int.	52,4
2	1	Nein	100	3	32	Int.	68,7
2	1	Nein	200	3	32	Int.	75,1
2	1	Nein	400	3	32	Int.	38,3
2	1	Nein	20	3	32	Dist. & Int.	76,1
2	1	Nein	50	3	32	Dist. & Int.	84,5
2	1	Nein	100	3	32	Dist. & Int.	92,4
2	1	Nein	200	3	32	Dist. & Int.	91,6
2	1	Nein	400	3	32	Dist. & Int.	89,7
2	1	Nein	20	3	32	Dist.	93,1
2	1	Nein	50	3	32	Dist.	96,6
2	1	Nein	100	3	32	Dist.	95,9
2	1	Nein	200	3	32	Dist.	94,3
2	1	Nein	400	3	32	Dist.	82,0
3	1	Nein	20	3	32	Dist.	83,2
3	1	Nein	50	3	32	Dist.	87,9
3	1	Nein	100	3	32	Dist.	88,3
3	1	Nein	200	3	32	Dist.	84,6
3	1	Nein	400	3	32	Dist.	92,6
2	2	Nein	10	3	32	Dist.	92,0
2	2	Nein	25	3	32	Dist.	80,9
2	2	Nein	50	3	32	Dist.	95,5
2	2	Nein	100	3	32	Dist.	69,7
2	2	Nein	200	3	32	Dist.	69,3
2	2	Nein	10	3	32	Int.	38,7
2	2	Nein	25	3	32	Int.	51,8
2	2	Nein	50	3	32	Int.	57,4
2	2	Nein	100	3	32	Int.	50,4
2	2	Nein	200	3	32	Int.	38,4

Convo- lution	<i>FCL</i>	Pooling	n_{Merkmale}	Kernel- größe	Merkmals- karten	Eingangsdaten	<i>VS</i> [%]
3	2	Nein	10	3	32	Dist.	94,4
3	2	Nein	25	3	32	Dist.	94,1
3	2	Nein	50	3	32	Dist.	81,6
3	2	Nein	100	3	32	Dist.	92,2
3	2	Nein	200	3	32	Dist.	90,7
3	2	Nein	10	3	32	Dist. & Int.	72,2
3	2	Nein	25	3	32	Dist. & Int.	81,4
3	2	Nein	50	3	32	Dist. & Int.	59,6
3	2	Nein	100	3	32	Dist. & Int.	68,7
3	2	Nein	200	3	32	Dist. & Int.	70,4
2	1	Nein	20	1	32	Dist.	95,9
2	1	Nein	50	3	32	Dist.	96,6
2	1	Nein	100	5	32	Dist.	95,9
2	1	Nein	200	7	32	Dist.	96,6
2	1	Nein	400	9	32	Dist.	96,2
2	1	Nein	20	3	2	Dist.	96,5
2	1	Nein	50	3	4	Dist.	95,3
2	1	Nein	100	3	8	Dist.	96,7
2	1	Nein	200	3	16	Dist.	96,4
2	1	Nein	400	3	32	Dist.	94,8
2	1	Nein	400	3	64	Dist.	92,7
2	1	Nein	20	3	8	Dist.	95,7
2	1	Nein	50	3	8	Dist.	96,7
2	1	Nein	100	3	8	Dist.	95,8
2	1	Nein	200	3	8	Dist.	95,7
2	1	Nein	400	3	8	Dist.	93,1
2	1	Ja	20	3	32,64	Dist.	92,2
2	1	Ja	50	3	32,64	Dist.	96,8
2	1	Ja	100	3	32,64	Dist.	96,7
2	1	Ja	200	3	32,64	Dist.	96,6
2	1	Ja	400	3	32,64	Dist.	96,1

Abbildungsverzeichnis

1.1. Sensoren für Fahrerassistenzsysteme und deren Anwendung.	9
2.1. Ablenkmechanismen des Laserstrahls durch einen sich drehenden Spiegel.	18
2.2. Flash-LiDAR-System	19
2.3. Segmentierung stangenförmiger Objekte in Punktwolken.	28
2.4. Beispiel eines Entscheidungsbaum.	31
2.5. Beispiel eines neuronalen Netzes mit einem Input-Layer, einem hidden Layer und einem Output-Layer.	33
2.6. Verlauf der logistischen Sigmoid-Funktion.	34
2.7. Verlauf der ReLU.	35
2.8. Verlauf der Tangens-hyperbolicus-Funktion.	35
2.9. Beispiel einer valid Convolution.	37
4.1. Mobile-Mapping-System Riegl VMX-250.	49
4.2. Valeo Scala.	50
4.3. Schematische Darstellung der Scannerposition und -Orientierung des Valeo Scala am Messfahrzeug.	51
4.4. Schematische Zeichnung des Oxford Robotcars.	51
4.5. SICK LMS151.	52
4.6. Anbringung des Velodyne VLP-16 und der Xsens MTi-G am Messfahrzeug.	53
4.7. Ablauf der Kalibrierung des Velodyne VLP-16 zum Riegl VMX-250.	54
4.8. Vergleich der Punktwolken des Velodyne VLP-16 mit den Daten des Riegl VMX-250 nach Anbringung verschiedener Rotationen zur Registrierung.	55
5.1. Punktabstände einer Punktwolke eines Mobile-Mapping-Systems zu einem Referenzdatensatz.	58
5.2. Aufteilung der ursprünglichen Punktwolke in drei Punktmengen anhand des lokalen Normalenvektors und einer maximalen Krümmung von 0,05.	59
5.3. Vergleich der Segmente mehrerer Punktwolken.	60
5.4. Detektion dynamischer Objekte in den Daten eines Mobile-Mapping-Systems.	60
5.5. Klassifizierung von Vegetation in Punktwolken.	61
5.6. Ergebnisse der Objekt-Segmentierung in Punktwolken.	62
5.7. Einfärbung der Punktwolke anhand der Linear-, Planar- und Volumenelemente.	62
5.8. Punktwolke eines Zeilen-Laserscanners.	63
5.9. Ergebnis der Klassifizierung dynamischer Objekte durch eine Änderungsdetektion sowie durch ein CNN.	65
5.10. CNN zur Detektion dynamischer Objekte in Scanbildern des Velodyne VLP-16.	66
6.1. Beispielhaftes Profil eines Zeilenscans des Valeo Scala.	69
6.2. Reduzierung des False-Positive-Detektionen stangenförmiger Objekte und Ebenen.	70
6.3. Ergebnisse der Segmentierung stangenförmiger Objekte und Ebenen aus den Messungen eines Velodyne VLP-16.	71

6.4. Detektierte stangenförmige Objekte und Ebenen in den verschiedenen LiDAR-Daten.	72
6.5. Erstellung eines Landmarkenmusters.	73
6.6. 3D-Darstellung aller lokalen Referenzdeskriptoren eines Testgebietes.	74
6.7. Ablauf der Sequenzanalyse zur Bestimmung der globalen Positionen.	75
6.8. Aus Mobile-Mapping-Daten erstelltes Scanprofil und das Ergebnis der Rekonstruktion durch den Decoder eines Autoencoders.	76
6.9. Beispielhafter Aufbau des Autoencoders für die aus den Mobile-Mapping-Daten erstellten Automotive-Laserscans mit einer Scanzeile.	77
6.10. Ablauf der Sequenzanalyse.	78
6.11. Beispiel eines aus einem Datensatz erstellten Intensitäts- und Höhenbildes.	80
6.12. Beispiel eines aus acht Datensätzen erstellten Intensitäts- und Höhenbildes.	80
6.13. Korrelation zwischen den Messungen des Laserscanners und den Referenzbildern.	81
6.14. Auf die x/y -Ebene projizierte Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Xsens MTi-G.	81
6.15. Auf die x/y -Ebene projizierte Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Inertialeinheit des Riegl VMX-250.	82
6.16. Auf die x/y -Ebene projizierte Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Inertialeinheit des Riegl VMX-250 bei einer maximalen Höhe von 3 m über der Straßenebene.	82
6.17. Höhenwerte der auf die x/y -Ebene projizierten statischen Punkte des Velodyne VLP-16 unter Verwendung der Orientierung aus den Messungen der Inertialeinheit des Riegl VMX-250 bei einer maximalen Höhe von 3 m über der Straßenebene.	83
7.1. Ermittlung der Fehlerellipse des Standpunktes aus erfolgreich detektierten und zugeordneten stangenförmigen Objekten und Ebenen.	85
7.2. Aus den Messungen des Mobile-Mapping-Systems extrahierte Stangen und Ebenen in Hannover-Badenstedt.	87
7.3. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der gemeinsamen Verwendung von Stangen und Ebenen, detektiert durch einen horizontal montierten Valeo Scala.	88
7.4. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der gemeinsamen Verwendung von Stangen und Ebenen, detektiert durch einen horizontal montierten Velodyne VLP-16.	89
7.5. Ergebnisse der Zuordnung stangenförmiger Objekte und Ebenen über Landmarken-Muster für verschiedene Parameter und Vergleich mit einem Nearest-Neighbor-Ansatz.	92
7.6. Darstellung der korrekten (grün) und fehlerhaften (rot) Lokalisierungsergebnisse durch einer Sequenzanalyse der simulierten Daten eines Einzeilenscanners.	96
7.7. Beispielhafte Ergebnisse der durch einen K-Means-Algorithmus mit $k = 100$ gelabelten Referenz- und Online-Trajektorie.	97
7.8. Vollständigkeit, Laufzeit und RMSE für eine unterschiedliche Anzahl an Merkmalen und einer variierenden Sequenzlänge der simulierten Daten eines Einzeilenscanners.	98
7.9. Vollständigkeit, Laufzeit und RMSE für eine unterschiedliche Anzahl an Merkmalen und Custern sowie einer variierenden Sequenzlänge der simulierten Daten eines Einzeilenscanners.	98
7.10. Vergleich zweier Punktwolken desselben Gebietes, welche an unterschiedlichen Tagen aufgenommen wurden.	99
7.11. Vollständigkeit und RMSE für eine unterschiedliche Anzahl an Merkmalen und einer variierenden Sequenzlänge der Daten des Oxford Robotcars.	99

7.12. Darstellung der korrekten und fehlerhaften Lokalisierungsergebnissen entlang der Trajektorie des Oxford Robotcars.	100
7.13. Bilder der Frontkamera des Oxford Robotcars in Bereichen der Trajektorie, in denen fehlerhafte Positionen auftreten.	101
7.14. Originale sowie vom Encoder prädizierte Scanbilder des Velodyne VLP-16, eingefärbt nach der gemessenen Distanz sowie der Intensität des reflektierten Laserimpulses.	102
7.15. Darstellung der korrekten und fehlerhaften Lokalisierungsergebnisse durch eine Sequenzanalyse der Daten des Velodyne VLP-16.	103
7.16. Vollständigkeit und RMSE für eine unterschiedliche Anzahl an Merkmalen und einer variierenden Sequenzlänge der Daten des Velodyne VLP-16.	104
7.17. Vollständigkeit und prozentualer Anteil übereinstimmender Labels der Sequenzanalyse durch die Daten des Velodyne VLP-16 bei einer variierenden Anzahl an Clustern.	104
7.18. Ergebnisse des K-Means-Algorithmus entlang unterschiedlicher Fahrspuren und einer Kurve, in der ein Wendemanöver durchgeführt wurde.	105
7.19. Ergebnisse des EKF bei einer Kombination von aus der Sequenzanalyse bestimmten Werte für Position und Orientierung und der Geschwindigkeits- und Drehratenwerte aus der Xsens MTi-G sowie aus den Messungen des Riegl VMX-250.	106
7.20. Beispiele für nach der Intensität eingefärbte Scanbilder in verschiedenen Regionen des Testgebietes.	109
7.21. Fehlerellipsen der Lokalisierung durch Scanbilder durch die Messungen eines Einzeilenscanners, welche aus den Daten des Riegl VMX-250 simuliert sind.	110
7.22. Überlagerte auf die x/y -Ebene projizierte Punkte des Riegl-VMX-250-Referenzsystems sowie des Velodyne VLP-16.	111
7.23. Fehlerellipsen der Lokalisierung durch Scanbilder aus Messungen des Velodyne VLP-16.	112
7.24. Auswirkungen des Samplings während der Scanbildkorrelation auf die Genauigkeit, Vollständigkeit und Laufzeit des Verfahrens.	114
A.1. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von Stangen, detektiert durch einen horizontal montierten Valeo Scala.	123
A.2. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von Ebenen, detektiert durch einen horizontal montierten Valeo Scala.	124
A.3. Aus den Messungen des Mobile-Mapping-Systems extrahierte stangenförmige Objekte und Ebenen in der Innenstadt von Hannover, entlang einer großen Verkehrsstraße sowie im Stadtteil Nordstadt.	125
A.4. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der gemeinsamen Verwendung von Stangen und Ebenen, detektiert durch einen vertikal montierten Velodyne VLP-16. Die Orientierung des Sensors wird aus den Messungen der Inertialeinheit des Riegl VMX-250 bestimmt, die zurückgelegten Distanz beträgt 50 m.	126
A.5. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von Stangen, detektiert durch einen vertikal montierten Velodyne VLP-16. Die Orientierung des Sensors wird aus den Messungen der Inertialeinheit des Riegl VMX-250 bestimmt, die zurückgelegten Distanz beträgt 50 m.	127
A.6. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von Ebenen, detektiert durch einen vertikal montierten Velodyne VLP-16. Die Orientierung des Sensors wird aus den Messungen der Inertialeinheit des Riegl VMX-250 bestimmt, die zurückgelegten Distanz beträgt 50 m.	128

- A.7. Fehlerellipsen der Trajektorie-Punkte im gewählten Testgebiet unter der Verwendung von Stangen, detektiert durch einen vertikal montierten Velodyne VLP-16. Die Orientierung des Sensors wird durch die Xsens MTi-G bestimmt, die zurückgelegte Distanz beträgt 50 m. . . 129
- B.1. Beispielhafter Aufbau des Autoencoders für die LiDAR-Daten des Velodyne VLP-16. 131
- B.2. Referenztrajektorie der Sequenzanalyse für den Datensatz des Velodyne VLP-16. 132

Tabellenverzeichnis

2.1. Aufbau einer Konfusionsmatrix zur Beurteilung von Klassifizierungsergebnissen.	30
2.2. Beispiel-Trainingsdaten für einen Entscheidungsbaum.	31
4.1. Vergleich der in dieser Arbeit verwendeten LiDAR-Systeme.	56
5.1. Evaluierung der Detektion von Vegetation durch einen Random Forest.	61
5.2. Evaluierung der Detektion dynamischer Objekte in Scanzeilen durch einen Random Forest. . .	64
5.3. Evaluierung der Detektion dynamischer Objekte in Scanbilder eines Velodyne VLP-16 durch ein CNN.	65
6.1. Gewählte Parameter zur Segmentierung der Stangen aus den Referenzdaten.	68
6.2. Gewählte Parameter zur Segmentierung der Ebenen aus den Referenzdaten.	69
7.1. In die Berechnung der Fehlerellipsen der Lokalisierung durch stangenförmige Objekte und Ebenen eingehende Unsicherheitsfaktoren.	86
7.2. Auswertung der erfolgreichen Positionsbestimmung über aus den simulierten Daten eines Einzeilenscanners extrahierte stangenförmige Objekte und Ebenen im Testgebiet.	87
7.3. Auswertung der erfolgreichen Positionsbestimmung durch aus den Daten eines Velodyne VLP- 16 extrahierte stangenförmige Objekte und Ebenen entlang der Trajektorie unter Verwendung der Orientierung aus der Xsens MTi-G.	90
7.4. Auswertung der erfolgreichen Positionsbestimmung durch aus den Daten eines Velodyne VLP- 16 extrahierte stangenförmige Objekte und Ebenen entlang der Trajektorie unter Verwendung der Orientierung aus der Inertialeinheit des Riegl VMX-250.	90
7.5. Ergebnisse der Zuordnung stangenförmiger Objekte und Ebenen über Landmarkenmuster für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 0$	93
7.6. Ergebnisse der Zuordnung stangenförmiger Objekte und Ebenen über Landmarkenmuster für $k_{\text{Stangen}} = 3$ und $k_{\text{Ebenen}} = 1$	93
7.7. Ergebnisse der Zuordnung stangenförmiger Objekte und Ebenen über Landmarkenmuster für $k_{\text{Stangen}} = 2$ und $k_{\text{Ebenen}} = 1$	93
7.8. Auswirkung der verschiedenen Parameter des neuronalen Netzes auf die Ergebnisse einer Sequenzanalyse der simulierten Daten eines Einzeilenscanners.	97
7.9. Auswirkung der verschiedenen Parameter des neuronalen Netzes auf die Ergebnisse einer Sequenzanalyse der Daten eines Velodyne VLP-16.	101
7.10. Ergebnisse des EKF bei einer Kombination der Ergebnisse einer Sequenzanalyse mit den Messungen der Xsens MTi-G sowie der Inertialeinheit des Riegl VMX-250.	106
7.11. Ergebnisse der Lokalisierung über eine Scanbildkorrelation der simulierten Daten eines Ein- zeilenscanners mit und ohne Anwendung der Klassifizierung zur Detektion dynamischer Objekte.	108
7.12. Ergebnisse der Lokalisierung über eine Scanbildkorrelation der simulierten Daten eines Ein- zeilenscanners nach Anwendung der Klassifizierung zur Detektion dynamischer Objekte in verschiedenen Gebieten.	109

7.13. Ergebnisse der Lokalisierung über eine Scanbildkorrelation der Daten des Velodyne VLP-16. .	113
7.14. Ergebnisse der Lokalisierung über eine Scanbildkorrelation der des Velodyne VLP-16 nach Anwendung der Klassifizierung zur Detektion dynamischer Objekte in verschiedenen Gebieten.	113
7.15. Erzielte Ergebnisse der Lokalisierung über detektierte stangenförmige Objekte und Ebenen, eine Sequenzanalyse sowie eine Scanbildkorrelation nach Anwendung der Klassifizierung zur Detektion dynamischer Objekte. Die Genauigkeit wird für die Verwendung stangenförmiger Objekte und Ebenen durch die mittlere Standardabweichung, für das Verfahren der Sequenz- analyse und die Scanbildkorrelation durch den RMSE-Wert in x/y -Richtung beschrieben. . .	115
B.1. Vollständige Auflistung der Auswirkungen veränderter Parameter des neuronalen Netzes bei einer Sequenzlänge von $n_{\text{seq}} = 200$ und einer Anzahl an Cluster-Mittelpunkten von $k = 1000$ auf die Lokalisierungs-Vollständigkeit.	133
B.1. Vollständige Auflistung der Auswirkungen der verschiedenen Parameter des neuronalen Netzes	134

Literaturverzeichnis

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. <https://www.tensorflow.org/>, aufgerufen am 09.02.2018.
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., u. a., 2016. TensorFlow: A System for Large-Scale Machine Learning. In: *OSDI*. Bd. 16. S. 265–283.
- Adams, R., Bischof, L., 1994. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence* 16 (6), S. 641–647.
- Aijazi, A., Checchin, P., Trassoudaine, L., 2013. Detecting and updating changes in lidar point clouds for automatic 3d urban cartography. *ISPRS Annals of the Photogram, Remote Sens Spat Inf Sci II (5/W2)*, S. 7–12.
- Arras, K. O., Mozos, O. M., Burgard, W., 2007. Using boosted features for the detection of people in 2d range data. In: *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, S. 3402–3407.
- Asvadi, A., Garrote, L., Premebida, C., Peixoto, P., Nunes, U. J., 2017. DepthCN: vehicle detection using 3D-LIDAR and convnet. In: *IEEE ITSC*.
- Bae, K.-H., Lichti, D. D., 2008. A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 63 (1), S. 36–54.
- Ballard, D. H., Brown, C. M., 1982. Computer vision. Prentice Hall Professional Technical Reference.
- Barnea, S., Filin, S., 2007. Registration of terrestrial laser scans via image based features. In: *ISPRS Workshop on Laser Scanning*. S. 12–14.
- Benedikt Still, 2016. Laserscanner ebnet Web zum pilotierten Fahren. <https://blog.audi.de/laserscanner-ebnet-weg-zum-pilotierten-fahren/>, aufgerufen am 17.06.2018.
- Benjemaa, R., Schmitt, F., 1999. Fast global registration of 3D sampled surfaces using a multi-z-buffer technique. *Image and Vision Computing* 17 (2), S. 113–123.
- Bentley, J. L., 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18 (9), S. 509–517.
- Beraldin, J.-A., Blais, F., Lohr, U., 2010a. Laser Scanning Technology. In: Vosselman, G., Maas, H. (Hrsg.), *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.
- Beraldin, J.-A., Blais, F., Lohr, U., 2010b. Laser Scanning Technology. In: Vosselman, G., Maas, H. (Hrsg.), *Airborne and Terrestrial Laser Scanning*. Whittles Publishing, S. 17.
- Besl, P., McKay, N., 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence* 14(2) 14 (2), S. 239–256.
- Biber, P., Straßer, W., 2003. The normal distributions transform: A new approach to laser scan matching. In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Bd. 3. IEEE, S. 2743–2748.
- Bishop, C., 2007. Pattern Recognition and Machine Learning. *Springer, New York*.

- Bishop, G., Welch, G. u. w., 2001. An introduction to the Kalman filter. *Proc of SIGGRAPH, Course 8* (27599-23175), S. 41.
- Blais, G., Levine, M. D., 1995. Registering multiview range data to create 3D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (8), S. 820–824.
- BMW Group, 2015. Pressemitteilung: AUDI AG, BMW Group und Daimler AG einigen sich mit Nokia Corporation über gemeinsamen Kauf des digitalen Kartengeschäfts HERE. <https://www.press.bmwgroup.com/deutschland/article/detail/T0228722DE/audi-ag-bmw-group-und-daimler-ag-einigen-sich-mit-nokia-corporation-ueber-gemeinsamen-kauf-des-digitalen-kartengeschaefts-here?language=de>, aufgerufen am 17.06.2018.
- Bock, F., Eggert, D., Sester, M., 2015. On-street parking statistics using lidar mobile mapping. In: *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, S. 2812–2818.
- Böhm, J., Becker, S., 2007. Automatic marker-free registration of terrestrial laser scans using reflectance. In: *Proceedings of the Proceedings of 8th Conference on Optical 3D Measurement Techniques, Zurich, Switzerland*. S. 9–12.
- Bolles, R., Fischler, M., 1981. A ransac-based approach to model fitting and its application to finding cylinders in range data. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*.
- Bosse, M., Roberts, J., 2007. Histogram matching and global initialization for laser-only SLAM in large unstructured environments. In: *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, S. 4820–4826.
- Breiman, L., 2001. Random forests. *Machine learning* 45 (1), S. 5–32.
- Breiman, L., Friedman, J., Stone, C. J., Olshen, R. A., 1984. Classification and regression trees. CRC press.
- Brenner, C., 2009a. Extraction of features from mobile laser scanning data for future driver assistance systems. In: *Advances in GIScience*. Springer, S. 25–42.
- Brenner, C., 2009b. Global Localization of Vehicles Using Local Pole Patterns. In: *Pattern Recognition*. Springer, S. 61–70.
- Brenner, C., Dold, C., Ripperda, N., 2008. Coarse orientation of terrestrial laser scans in urban environments. *ISPRS journal of photogrammetry and remote sensing* 63 (1), S. 4–18.
- Brubaker, M. A., Geiger, A., Urtasun, R., 2016. Map-based probabilistic visual self-localization. *IEEE transactions on pattern analysis and machine intelligence* 38 (4), S. 652–665.
- Champion, N., Rottensteiner, F., Matikainen, L., Liang, X., Hyypä, J., Olsen, B., 2009. A test of automatic building change detection approaches. *Proceedings of CMRT09*, S. 03–04.
- Chen, Y., Medioni, G., 1992. Object modelling by registration of multiple range images. *Image and vision computing* 10 (3), S. 145–155.
- Choi, J., Maurer, M., 2015. Simultaneous localization and mapping based on the local volumetric hybrid map. In: *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, S. 561–566.
- Continental AG, 2018. 3D Flash Lidar. <https://www.continental-automotive.com/de-DE/Landing-Pages/CAD/Automated-Driving/GlobalHighlights/3D-Flash-Lidar>, aufgerufen am 20.01.2018.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Criminisi, A., Shotton, J., Konukoglu, E., u. a., 2012. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision* 7 (2–3), S. 81–227.

- Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Bd. 1. IEEE, S. 886–893.
- DARPA, 2018. Urban Challenge. <http://archive.darpa.mil/grandchallenge/>, aufgerufen am 14.02.2018.
- DEKRA, 2017. Dekra Verkehrssicherheitsreport 2017. <https://www.dekra-roadsafety.com/media/04archiv/pdf/dekra-vsr-2017-best-practices-de.pdf>, aufgerufen am 17.06.2018.
- Demantke, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3D lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (Part 5), S. W12.
- Dewan, A., Caselitz, T., Tipaldi, G. D., Burgard, W., 2016. Motion-based detection and tracking in 3d lidar scans. In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, S. 4508–4513.
- Ding, W., Wang, J., Mumford, P., Li, Y., Rizos, C., 2005. Time synchronization design for integrated positioning and georeferencing systems. In: *Proceedings of SSC*. S. 1265–1274.
- Dorai, C., Wang, G., Jain, A. K., Mercer, C., 1998. Registration and integration of multiple object views for 3D model construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1), S. 83–89.
- Douillard, B., Underwood, J., Vlaskine, V., Quadros, A., Singh, S., 2014. A pipeline for the segmentation and classification of 3D point clouds. In: *Experimental robotics*. Springer, S. 585–600.
- El-Sheimy, N., 2005. An overview of mobile mapping systems. In: *FIG Working Week*. S. 16–21.
- Elseberg, J., Magnenat, S., Siegwart, R., Nuchter, A., 2012. Comparison on nearest-neighbour-search strategies and implementations for efficient shape registration. 3, S. 2–12.
- Feng, Y., Schlichting, A., Brenner, C., 2016. 3D feature point extraction from LiDAR data using a neural network. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences- ISPRS Archives 41 (2016)* 41, S. 563–569.
- Förstner, W., Khoshelham, K., 2017. Efficient and Accurate Registration of Point Clouds with Plane to Plane Correspondences. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. S. 2165–2173.
- Gardner, C. S., 1992. Ranging performance of satellite laser altimeters. *IEEE Transactions on Geoscience and Remote Sensing* 30 (5), S. 1061–1072.
- Gehring, J., Hebel, M., Arens, M., Stilla, U., 2017. An Approach to Extract Moving Objects from MIs Data Using a Volumetric Background Representation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4, S. 107.
- Girardeau-Montaut, D., Roux, M., Marc, R., Thibault, G., 2005. Change detection on points cloud data acquired with a ground laser scanner. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (part 3), S. W19.
- GmbH, I. A. S., 2013. ibeo Lux. <http://www.ibeo-as.com>, aufgerufen am 17.06.2018.
- Godin, G., Rioux, M., Baribeau, R., 1994. Three-dimensional registration using range and intensity information. In: *Proceedings of SPIE videometric III*. Bd. 2350. S. 279–290.
- Golovinskiy, A., Kim, V. G., Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments. In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, S. 2154–2161.
- Gonzalez, R. C., Woods, R. E., u. a., 2002. Digital image processing.
- Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016a. Deep learning. Bd. 1. MIT press Cambridge, Kap. 6.

- Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016b. Deep learning. Bd. 1. MIT press Cambridge, Kap. 8.
- Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016c. Deep learning. Bd. 1. MIT press Cambridge, Kap. 9.
- Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016d. Deep learning. Bd. 1. MIT press Cambridge, Kap. 14.
- Gupta, S., Girshick, R., Arbeláez, P., Malik, J., 2014. Learning rich features from RGB-D images for object detection and segmentation. In: *European Conference on Computer Vision*. Springer, S. 345–360.
- Guttman, A., 1984. R-trees: A dynamic index structure for spatial searching. Bd. 14. ACM.
- Haala, N., Peter, M., Kremer, J., Hunter, G., 2008. Mobile LiDAR mapping for 3D point cloud collection in urban areas—A performance test. *The international archives of the photogrammetry, remote sensing and spatial information sciences* 37, S. 1119–1127.
- Hamming, R. W., 1950. Error detecting and error correcting codes. *Bell Labs Technical Journal* 29 (2), S. 147–160.
- Hastie, T., Tibshirani, R., Friedman, J., 2001. The elements of statistical learning. Bd. 1. Springer series in statistics New York.
- Hata, A. Y., Wolf, D. F., 2016. Feature detection for vehicle localization in urban environments using a multilayer LIDAR. *IEEE Transactions on Intelligent Transportation Systems* 17 (2), S. 420–429.
- Hebel, M., Stilla, U., 2012. Simultaneous calibration of ALS systems and alignment of multiview LiDAR scans of urban areas. *IEEE Transactions on Geoscience and Remote Sensing* 50 (6), S. 2364–2379.
- Himstedt, M., Frost, J., Hellbach, S., Böhme, H.-J., Maehle, E., 2014. Large scale place recognition in 2D LIDAR scans using geometrical landmark relations. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, S. 5030–5035.
- Hinton, G., Srivastava, N., Swersky, K., 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.
- Hoffman, R., Jain, A. K., 1987. Segmentation and classification of range images. *IEEE transactions on pattern analysis and machine intelligence* (5), S. 608–620.
- Höfle, B., Hollaus, M., Hagenauer, J., 2012. Urban vegetation detection using radiometrically calibrated small-footprint full-waveform airborne LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing* 67, S. 134–147.
- Hofmann, S., 2017. Potential von LiDAR Mobile Mapping für hochgenaue Karten. Dissertation, Leibniz Universität Hannover.
- Hofton, M. A., Minster, J.-B., Blair, J. B., 2000. Decomposition of laser altimeter waveforms. *IEEE Transactions on geoscience and remote sensing* 38 (4), S. 1989–1996.
- Horn, B. K., 1987. Closed-form solution of absolute orientation using unit quaternions. *JOSA A* 4 (4), S. 629–642.
- Hu, H., Wei, N., 2009. A study of GPS jamming and anti-jamming. In: *Power Electronics and Intelligent Transportation System (PEITS), 2009 2nd International Conference on*. Bd. 1. IEEE, S. 388–391.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093*.
- Jutzi, B., Meyer, F. J., Hinz, S., 2016. Aktive Fernerkundungssensorik - Technologische Grundlagen und Abbildungsgeometrie. In: Heipke, C. (Hrsg.), *Photogrammetrie und Fernerkundung*. Springer-Verlag.

- Jutzi, B., Stilla, U., 2006. Range determination with waveform recording laser systems using a Wiener Filter. *ISPRS Journal of Photogrammetry and Remote Sensing* 61 (2), S. 95–107.
- Jutzi, B., Stilla, U., 2008. Waveform analysis for small-footprint pulsed laser systems. In: *Topographic Laser Ranging and Scanning*. CRC Press, S. 233–252.
- Kaartinen, H., Hyypä, J., Kukko, A., Jaakkola, A., Hyypä, H., 2012. Benchmarking the performance of mobile laser scanning systems using a permanent test field. *Sensors* 12 (9), S. 12814–12835.
- Kahmen, H., 2006. Angewandte Geodäsie: Vermessungskunde. De Gruyter Lehrbuch. Walter de Gruyter.
- Kalman, R. E., 1960. A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82 (1), S. 35–45.
- Kang, Z., Li, J., Zhang, L., Zhao, Q., Zlatanova, S., 2009. Automatic registration of terrestrial laser scanning point clouds using panoramic reflectance images. *Sensors* 9 (4), S. 2621–2646.
- Kendall, A., Cipolla, R., 2017. Geometric loss functions for camera pose regression with deep learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Kendall, A., Grimes, M., Cipolla, R., 2015. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. .
- Kerns, A. J., Shepard, D. P., Bhatti, J. A., Humphreys, T. E., 2014. Unmanned aircraft capture and control via GPS spoofing. *Journal of Field Robotics* 31 (4), S. 617–636.
- Kidono, K., Miyasaka, T., Watanabe, A., Naito, T., Miura, J., 2011. Pedestrian recognition using high-definition LIDAR. In: *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, S. 405–410.
- Kim, H., Liu, B., Goh, C. Y., Lee, S., Myung, H., 2017. Robust Vehicle Localization Using Entropy-Weighted Particle Filter-based Data Fusion of Vertical and Road Intensity Information for a Large Scale Urban Area. *IEEE Robotics and Automation Letters* 2 (3), S. 1518–1524.
- Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kirschvink, J., 1980. The least-squares line and plane and the analysis of palaeomagnetic data. *Geophysical Journal International* 62 (3), S. 699–718.
- Kraftfahrt-Bundesamt, 2017. Gesamtkilometer steigen um 1,4 Prozent. https://www.kba.de/DE/Statistik/Kraftverkehr/VerkehrKilometer/verkehr_in_kilometern_node.html, aufgerufen am 17.06.2018.
- Kremer, J., 2007. Performance of the StreetMapper Mobile LIDAR Mapping System in Real World Projects. *“Photogrammetric Week 2007”*, S. 215–225.
- Kutterer, H., 2010. Mobile Mapping. In: Vosselman, G., Maas, H. (Hrsg.), *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.
- Lages, U., Spencer, M., Katz, R., 2013. Automatic scenario generation based on laserscanner reference data and advanced offline processing. In: *Intelligent Vehicles Symposium Workshops (IV Workshops), 2013 IEEE*. IEEE, S. 146–148.
- Lategahn, H., Beck, J., Stiller, C., 2014. DIRD is an illumination robust descriptor. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, S. 756–761.
- LeCun, Y., u. a., 1989. Generalization and network design strategies. *Connectionism in perspective*, S. 143–155.
- Lehtomäki, M., Jaakkola, A., Hyypä, J., Kukko, A., Kaartinen, H., 2010. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sensing* 2 (3), S. 641–664.
- Levinson, J., Montemerlo, M., Thrun, S., 2007. Map-Based Precision Vehicle Localization in Urban Environments. In: *Robotics: Science and Systems*.

- Li, B., Zhang, T., Xia, T., 2016a. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*.
- Li, L., Yang, M., Wang, C., Wang, B., 2016b. Road DNA based localization for autonomous vehicles. In: *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, S. 883–888.
- Lichti, D., Skaloud, J., 2010. Registration and Calibration. In: Vosselman, G., Maas, H. (Hrsg.), *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.
- Lin, J., 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37 (1), S. 145–151.
- Lin, M., Chen, Q., Yan, S., 2013. Network in network. *arXiv preprint arXiv:1312.4400*.
- Lin, Y., Hyypä, J., Jaakkola, A., 2011. Mini-UAV-borne LIDAR for fine-scale mapping. *IEEE Geoscience and Remote Sensing Letters* 8 (3), S. 426–430.
- Lowe, D. G., 1999. Object recognition from local scale-invariant features. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Bd. 2. Ieee, S. 1150–1157.
- Maas, A. L., Hannun, A. Y., Ng, A. Y., 2013. Rectifier nonlinearities improve neural network acoustic models. In: *Proc. icml*. Bd. 30. S. 3.
- MacQueen, J., u. a., 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Bd. 1. Oakland, CA, USA., S. 281–297.
- Maddern, W., Pascoe, G., Linegar, C., Newman, P., 2017. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)* 36 (1), S. 3–15.
- Maddern, W., Pascoe, G., Newman, P., 2015. Leveraging experience for large-scale LIDAR localisation in changing cities. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, S. 1684–1691.
- Masuda, T., Sakaue, K., Yokoya, N., 1996. Registration and integration of multiple range images for 3-D model construction. In: *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. Bd. 1. IEEE, S. 879–883.
- MathWorks, 2018. Neural Network Toolbox. Create, train, and simulate shallow and deep learning neural networks. <https://de.mathworks.com/products/neural-network.html>, aufgerufen am 09.02.2018.
- Meagher, D., 1982. Geometric modeling using octree encoding. *Computer graphics and image processing* 19 (2), S. 129–147.
- Michalski, R. S., Carbonell, J. G., Mitchell, T. M., 2013. Machine learning: An artificial intelligence approach. Springer Science & Business Media.
- Microsoft, 2018. The Microsoft Cognitive Toolkit. <https://www.microsoft.com/en-us/cognitive-toolkit/>, aufgerufen am 09.02.2018.
- Navarro-Serment, L. E., Mertz, C., Hebert, M., 2010. Pedestrian detection and tracking using three-dimensional ladar data. *The International Journal of Robotics Research* 29 (12), S. 1516–1528.
- Neugebauer, P. J., 1997. Geometrical cloning of 3D objects via simultaneous registration of multiple range images. In: *Shape Modeling and Applications, 1997. Proceedings., 1997 International Conference on*. IEEE, S. 130–139.
- Neumann, T., Ferrein, A., Kallweit, S., Scholl, I., 2014. Towards a mobile mapping robot for underground mines. In: *Proc. 2014 PRASA, RobMech and AfLaT Int. Joint Symposium, Cape Town, South Africa*.
- Nguyen, A., Le, B., 2013. 3D point cloud segmentation: A survey. In: *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*. IEEE, S. 225–230.

- Nielsen, M. A., 2015. Neural Networks and Deep learning. Determination Press, Kap. 3.
- Nievergelt, J., Hinterberger, H., Sevcik, K. C., 1984. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems (TODS)* 9 (1), S. 38–71.
- Nothdurft, T., Hecker, P., Ohl, S., Saust, F., Maurer, M., Reschka, A., Bohmer, J., 2011. Stadtpilot: First fully autonomous test drives in urban traffic. In: *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, S. 919–924.
- Novatel, 2016. SPAN CPT User Manual. <https://www.novatel.com/assets/Documents/Papers/SPAN-CPT.pdf>, aufgerufen am 17.06.2018.
- OpenStreetMap contributors, 2018. Kartenausschnitt aus <https://planet.osm.org>.
- Oxford Technical Solutions Ltd., 2013. RT3000 Inertial and GPS Navigation System - Datasheet. <http://www.oxts.com>, aufgerufen am 17.06.2018.
- Pabst, S., Koch, A., Straßer, W., 2010. Fast and scalable cpu/gpu collision detection for rigid and deformable surfaces. In: *Computer Graphics Forum*. Bd. 29. Wiley Online Library, S. 1605–1612.
- Petrovskaya, A., Thrun, S., 2009. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots* 26 (2-3), S. 123–139.
- Pfeifer, N., Mandlbürger, G., Glira, P., 2016. Laserscanning. In: Heipke, C. (Hrsg.), *Photogrammetrie und Fernerkundung*. Springer-Verlag.
- Point Cloud Library, 2017. Cylinder model segmentation.
- Polewski, P., Yao, W., Heurich, M., Krzystek, P., Stilla, U., 2016. Combining active and semisupervised learning of remote sensing data within a Renyi entropy regularization framework. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9 (7), S. 2910–2922.
- Pomerleau, D., Jochem, T., 1996. Rapidly adapting machine vision for automated vehicle steering. *IEEE expert* 11 (2), S. 19–27.
- Premebida, C., Carreira, J., Batista, J., Nunes, U., 2014. Pedestrian detection combining rgb and dense lidar data. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, S. 4112–4117.
- Premebida, C., Ludwig, O., Nunes, U., 2009. Exploiting lidar-based features on pedestrian detection in urban scenarios. In: *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*. IEEE, S. 1–6.
- Pu, S., Vosselmann, G., 2006. Automatic extraction of building features from terrestrial laser scanning.
- Pulli, K., Shapiro, L. G., 2000. Surface reconstruction and display from range and color data. *Graphical Models* 62 (3), S. 165–201.
- Qu, X., Soheilian, B., Paparoditis, N., 2015. Vehicle localization using mono-camera and geo-referenced traffic signs. In: *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, S. 605–610.
- Qu, X., Soheilian, B., Paparoditis, N., 2018. Landmark based localization in urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing* 140, S. 90–103.
- Reitberger, J., Krzystek, P., Stilla, U., 2008. Analysis of full waveform LIDAR data for the classification of deciduous and coniferous trees. *International journal of remote sensing* 29 (5), S. 1407–1431.
- RIEGL Laser Measurement Systems GmbH, 2012a. RIEGL VMX-250. http://www.riegl.com/uploads/tx_pxriegldownloads/10_DataSheet_VMX-250_20-09-2012.pdf, aufgerufen am 17.06.2018.
- RIEGL Laser Measurement Systems GmbH, 2012b. RIEGL VQ-250. http://www.riegl.com/uploads/tx_pxriegldownloads/10_DataSheet_VQ-250_rund_25-09-2012.pdf, aufgerufen am 17.06.2018.

- Rohde, J., Völz, B., Mielenz, H., Zöllner, J. M., 2016. Precise vehicle localization in dense urban environments. In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, S. 853–858.
- Rokach, L., Maimon, O., 2009. Classification trees. In: *Data mining and knowledge discovery handbook*. Springer, S. 149–174.
- Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986. Learning representations by back-propagating errors. *nature* 323 (6088), S. 533.
- Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm. In: *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, S. 145–152.
- Rusu, R. B., 2009. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. Dissertation, Computer Science department, Technische Universität München, Germany.
- Schachtschneider, J., Schlichting, A., Brenner, C., 2017. Assessing temporal behavior in lidar point clouds of urban environments. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42, S. 543.
- Schlemmer, H., 1996. Grundlagen der Sensorik: Eine Instrumentenkunde für Vermessungsingenieure. Wichmann.
- Schlichting, A., Brenner, C., 2014a. Genauigkeitsuntersuchung zur Lokalisierung von Fahrzeugen mittels Automotive-Laserscannern. In: *Publikation der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V. - Vorträge der 34. Wissenschaftlich-Technischen Jahrestagung der DGPF*. DGPF.
- Schlichting, A., Brenner, C., 2014b. Localization using automotive laser scanners and local pattern matching. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, S. 414–419.
- Schlichting, A., Brenner, C., 2016a. Generating a Hazard Map of Dynamic Objects Using Lidar Mobile Mapping. *Photogrammetric Engineering & Remote Sensing* 82 (12), S. 967–972.
- Schlichting, A., Brenner, C., 2016b. Vehicle localization by lidar point correlation improved by change detection. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences- ISPRS Archives* 41 (2016) 41, S. 703–710.
- Schlosser, J., Chow, C. K., Kira, Z., 2016. Fusing lidar and images for pedestrian detection using convolutional neural networks. In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, S. 2198–2205.
- Seif, H. G., Hu, X., 2016. Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry. *Engineering* 2 (2), S. 159–162.
- SICK AG, 2017. LMS151-10100, LMS1xx. https://www.sick.com/media/pdf/0/40/840/dataSheet_LMS151-10100_1047607_en.pdf, aufgerufen am 17.06.2018.
- Spangenberg, R., Goehring, D., Rojas, R., 2016. Pole-based localization for autonomous vehicles in urban scenarios. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, S. 2161–2166.
- Spinello, L., Arras, K. O., Triebel, R., Siegwart, R., 2010. A Layered Approach to People Detection in 3D Range Data. In: *AAAI*. Bd. 10. S. 1–1.
- Stein, M., 1987. Large sample properties of simulations using Latin hypercube sampling. *Technometrics* 29 (2), S. 143–151.
- Steinhauser, D., Ruepp, O., Burschka, D., 2008. Motion segmentation and scene classification from 3D LIDAR data. In: *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, S. 398–403.

- Stilla, U., Jutzi, B., 2008. Waveform analysis for small-footprint pulsed laser systems. In: *Topographic Laser Ranging and Scanning*. CRC Press, S. 233–252.
- Suhr, J. K., Jang, J., Min, D., Jung, H. G., 2017. Sensor fusion-based low-cost vehicle localization system for complex urban environments. *IEEE Transactions on Intelligent Transportation Systems* 18 (5), S. 1078–1086.
- Tesla Inc., 2016. A Tragic Loss. https://www.tesla.com/de_DE/blog/tragic-loss, aufgerufen am 17.06.2018.
- Thrun, S., Burgard, W., Fox, D., 2005a. Probabilistic robotics. MIT press, Kap. 4.
- Thrun, S., Burgard, W., Fox, D., 2005b. Probabilistic robotics. MIT press, Kap. 2.
- Thrun, S., Burgard, W., Fox, D., 2005c. Probabilistic robotics. MIT press, Kap. 3.
- Tipaldi, G. D., Spinello, L., Burgard, W., 2013. Geometrical FLIRT phrases for large scale place recognition in 2D range data. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, S. 2693–2698.
- Toth, C., Jutzi, B., 2016. Plattformen und Sensoren für die Fernerkundung und deren Geopositionierung. In: Heipke, C. (Hrsg.), *Photogrammetrie und Fernerkundung*. Springer-Verlag.
- Turk, G., Levoy, M., 1994. Zippered polygon meshes from range images. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, S. 311–318.
- Tuttas, S., Braun, A., Borrmann, A., Stilla, U., 2017. Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4D BIM. *PFG-Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 85 (1), S. 3–15.
- Valeo, 2017. Valeo Scala®. <https://www.valeo.com/en/valeo-scala/>, aufgerufen am 06.01.2018.
- Velodyne LiDAR, 2017. Velodyne LiDAR Puck, real-time LiDAR 3D sensor. <http://velodynelidar.com/vlp-16.html>, aufgerufen am 17.06.2018.
- Verband der Automobilindustrie e.V., 2018. Automatisiertes Fahren. Die Evolution von Fahrfunktionen - von Assistenten zu Automaten. <https://www.vda.de/de/themen/innovation-und-technik/automatisiertes-fahren/zukunft-des-automatisierten-fahrens.html>, aufgerufen am 12.05.2018.
- Vitányi, P. M., Balbach, F. J., Cilibrasi, R. L., Li, M., 2009. Normalized information distance. In: *Information theory and statistical learning*. Springer, S. 45–82.
- Vosselmann, G., Klein, R., 2010. Visualisation and Structuring of Point Clouds. In: Vosselman, G., Maas, H. (Hrsg.), *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.
- Vu, T. T., Matsuoka, M., Yamazaki, F., 2004. LIDAR-based change detection of buildings in dense urban areas. In: *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*. Bd. 5. IEEE, S. 3413–3416.
- Wagner, W., Ullrich, A., Melzer, T., Briese, C., Kraus, K., 2004. From single-pulse to full-waveform airborne laser scanners: potential and practical challenges. *Remote Sensing and Spatial Information Sciences* 35 (Part B3), S. 201–206.
- Wallace, L., Watson, C., Lucieer, A., 2014. Detecting pruning of individual stems using airborne laser scanning data captured from an unmanned aerial vehicle. *International Journal of Applied Earth Observation and Geoinformation* 30, S. 76–85.
- Wei, L., Cappelle, C., Ruichek, Y., 2013. Horizontal/vertical LRFs and GIS maps aided vehicle localization in urban environment. In: *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, S. 809–814.

- Weik, S., 1997. Registration of 3-D partial surface models using luminance and depth information. In: *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in.* IEEE, S. 93–100.
- Weinmann, M., 2016. Reconstruction and analysis of 3D scenes: from irregularly distributed 3D points to object classes. Dissertation, Karlsruher Institut für Technologie (KIT).
- Weinmann, M., Weinmann, M., Hinz, S., Jutzi, B., 2011. Fast and automatic image-based registration of TLS data. *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (6), S. S62–S70.
- West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M., Iverson, A. E., 2004. Context-driven automated target detection in 3D data. In: *Automatic Target Recognition XIV.* Bd. 5426. International Society for Optics and Photonics, S. 133–144.
- Wolcott, R. W., Eustice, R. M., 2014. Visual localization within lidar maps for automated urban driving. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on.* IEEE, S. 176–183.
- Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., Liu, H., 2013. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sensing* 5 (2), S. 584–611.
- Wu, F., Wen, C., Guo, Y., Wang, J., Yu, Y., Wang, C., Li, J., 2017. Rapid localization and extraction of street light poles in mobile LiDAR point clouds: A supervoxel-based approach. *IEEE Transactions on Intelligent Transportation Systems* 18 (2), S. 292–305.
- Xiao, W., Vallet, B., Brédif, M., Paparoditis, N., 2015. Street environment change detection from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 107, S. 38–49.
- Xsens Technologies B.V., 2012. MTi-G: Miniature AHRS with integrated GPS.
- Xu, D., Erdogmus, D., 2010. Renyi's entropy, divergence and their nonparametric estimators. In: *Information Theoretic Learning.* Springer, S. 47–102.
- Xu, Y., Tuttas, S., Hoegner, L., Stilla, U., 2018. Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor. *Automation in Construction* 85, S. 76–95.
- Yoneda, K., Tehrani, H., Ogawa, T., Hukuyama, N., Mita, S., 2014. Lidar scan feature for localization with highly precise 3-D map. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE.* IEEE, S. 1345–1350.
- Zeibak, R., Filin, S., 2008. Change detection via terrestrial laser scanning. Technion-Israel Institute of Technology, Faculty of Civil and Environmental Engineering.
- Zeiler, M. D., Krishnan, D., Taylor, G. W., Fergus, R., 2010. Deconvolutional networks. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* IEEE, S. 2528–2535.
- Zeisl, B., Sattler, T., Pollefeys, M., 2015. Camera pose voting for large-scale image-based localization. In: *Computer Vision (ICCV), 2015 IEEE International Conference on.* IEEE, S. 2704–2712.
- Ziegler, J., Latégahn, H., Schreiber, M., Keller, C. G., Knoppel, C., Hipp, J., Haueis, M., Stiller, C., 2014. Video based localization for bertha. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE.* IEEE, S. 1231–1238.
- Zlot, R., Bosse, M., 2014. Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine. In: *Field and service robotics.* Springer, S. 479–493.
- Zoller + Fröhlich GmbH, 2018. Z+F Imager 5010X. http://www.zf-laser.com/fileadmin/editor/Datenblaetter/Z_F_IMAGER_5010_Datenblatt_D.pdf, aufgerufen am 17.06.2018.

Lebenslauf

Persönliches

Name	Alexander Schlichting
Geburtsdatum	20.05.1986 in Hamburg

Schule

1993 - 1997	Grundschule Wiegensen
1997 - 1999	Orientierungsstufe Süd, Buxtehude
1999 - 2003	Gymnasium Süd, Buxtehude
2003 - 2006	Halepaghen-Schule, Buxtehude
2006	Abitur

Studium

2007 - 2012	Studium der Geodäsie und Geoinformatik, Leibniz Universität Hannover
2012	Abschluss als Master of Science (M. Sc.)

Beruf

seit 2012	Wissenschaftlicher Mitarbeiter am Institut für Kartographie und Geoinformatik, Leibniz Universität Hannover
-----------	---

Danksagung

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Kartographie und Geoinformatik an der Leibniz Universität Hannover. In dieser Zeit haben mich viele Personen unterstützt und somit zum Abschluss der Arbeit beigetragen.

Ein großer Dank gilt meinem Betreuer, Herrn apl. Prof. Dr.-Ing. Claus Brenner, der bereits meine Masterarbeit betreut und mich dazu ermutigt hat, den Weg zurück nach Hannover anzutreten und die Dissertation anzufertigen. Vielen Dank für die hilfreichen Hinweise, Ideen und auch für die Geduld.

Bedanken möchte ich mich auch bei Herrn Prof. Dr.-Ing Uwe Stilla und Herrn Prof. Dr.-Ing. Steffen Schön für die freundliche Übernahme des Korreferats.

Vielen Dank an Frau Prof. Dr.-Ing. habil. Monika Sester für die zahlreichen Möglichkeiten, die sich auch durch ihre Unterstützung für mich ergeben haben, wie die Teilnahme an weltweiten Konferenzen und Exkursionen oder den Auslandsaufenthalt an der University of Melbourne.

Auch meinen Kollegen am Institut sowie an der gesamten Fakultät für Bauingenieurwesen und Geodäsie sei für die tolle Arbeitsatmosphäre, die guten Gesprächen und für die Hilfe bei verschiedenen fachlichen Fragestellungen sowie während der Messungen gedankt.

Ganz herzlich bedanke ich mich bei meinen Freunden und meiner Familie für die gesamte Unterstützung seit Beginn meines Studiums in Hannover. Meinen Freunden vor allem für ihr Verständnis, dass ich nicht immer vor Ort war und insbesondere in den letzten Monaten vor der Abgabe der Arbeit wenig Zeit hatte aber auch für die ihre Hilfe während des Studiums und das Durchlesen dieser Arbeit. Meiner Familie danke ich insbesondere für die moralische Unterstützung und dafür, dass sie immer für mich da sind, wenn ich sie brauche.

Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover

(Eine vollständige Liste der Wiss. Arb. ist beim Geodätischen Institut, Nienburger Str. 1, 30167 Hannover erhältlich.)

Nr. 319	LIN, Miao:	Regional gravity field recovery using the point mass method (Diss. 2015)
Nr. 320	ZHANG, Lijuan:	Mining GPS-Trajectory Data for Map Refinement and Behavior Detection (Diss. 2015)
Nr. 321	ZADDACH, Sebastian:	Zum Beitrag Bayesscher Schätzverfahren in der Vergleichswertermittlung (Diss. 2016)
Nr. 322	SMYRNAIOS, Marios:	Carrier-phase Multipath in Satellite-based Positioning (Diss. 2016)
Nr. 323	MENZE, Moritz:	Object Scene Flow (Diss. 2016)
Nr. 324	WU, Hu:	Gravity field recovery from GOCE observations (Diss. 2016)
Nr. 325	XU, Xiangyang:	Terrestrial Laser Scanning for the Generation and Calibration of Finite Element Models (Diss. 2016)
Nr. 326	SAYYAD, Muhammad Naeem Shahzad:	Joint use and mutual control of terrestrial laser scans and digital images for accurate 3D measurements (Diss. 2016)
Nr. 327	SCHACK, Lukas:	Object-based matching of Persistent Scatterers to Optical Oblique Images (Diss. 2016)
Nr. 328	REICH, Martin:	Global Image Orientation from Pairwise Relative Orientations (Diss. 2016)
Nr. 329	KLINGER, Tobias:	Probabilistic multi-person localisation and tracking (Diss. 2016)
Nr. 330	SCHMIDT, Alena:	Markierte Punktprozesse für die automatische Extraktion von Liniennetzen in Rasterdaten (Diss. 2016)
Nr. 331	HOFMANN, Franz:	Lunar Laser Ranging - verbesserte Modellierung der Monddynamik und Schätzung relativistischer Parameter (Diss. 2017)
Nr. 332	BRIEDEN, Phillip:	Validierung von GOCE-Gravitationsgradienten in Kreuzungspunkten und Zukunftsperspektiven der Satellitengradiometrie (Diss. 2017)
Nr. 333	VON GÖSSELN, Ilka:	Simulationsbasierte Effizienzoptimierung von Messprozessen am Beispiel der tachymetrischen Netzmessung (Diss. 2017)
Nr. 334	HOFMANN, Sabine:	Potential von LiDAR Mobile Mapping für hochgenaue Karten (Diss. 2017)
Nr. 335	ALBERT, Lena:	Simultane Klassifikation der Bodenbedeckung und Landnutzung unter Verwendung von Conditional Random Fields (Diss. 2017)
Nr. 336	NIEMEYER, Joachim:	Verwendung von Kontext zur Klassifikation luftgestützter Laserdaten urbaner Gebiete (Diss. 2017)
Nr. 337	EGGERT, Daniel:	Effiziente Verarbeitung und Visualisierung von Mobile Mapping Daten (Diss. 2017)
Nr. 338	FITZNER, Daniel:	Estimation of Spatio-Temporal Moving Fields at High Resolution (Diss. 2017)
Nr. 339	CZIOSKA, Paul:	Meeting point locations for shared rides (Diss. 2018)
Nr. 340	FEUERHAKE, Udo:	Erfassung von Trajektorien und Erkennung von Bewegungsmustern (Diss. 2018)
Nr. 341	HUANG, Hai:	Bayesian Models for Pattern Recognition in Spatial Data (Habil. 2018)
Nr. 342	BOCK, Urs Fabian:	Dynamic Parking Maps from Vehicular Crowdsensing (Diss. 2018)
Nr. 343	KRAWINKEL, Thomas:	Improved GNSS Navigation with Chip-scale Atomic Clocks (Diss. 2018)
Nr. 344	GOSWAMI, Sujata:	Understanding the sensor noise in the GRACE range-rate observations by analyzing their residuals (Diss. 2018)
Nr. 345	ZHANG, Yin:	Expected Utilities for Decision Making and Steering of Measurement Processes (Diss. 2018)
Nr. 346	SCHLICHTING, Alexander:	Fahrzeuglokalisierung durch Automotive Laserscanner unter Verwendung statischer Merkmale (Diss. 2018)

Die Arbeiten werden im Rahmen des wissenschaftlichen Schriftenaustausches verteilt und sind nicht im Buchhandel erhältlich. Der Erwerb ist zu einem Stückpreis von € 25,00 bei den herausgebenden Instituten möglich.

