TECHNISCHE UNIVERSITÄT MÜNCHEN
TUM School of Engineering and Design

Detektion, Verfolgung und Posenschätzung von Personen im urbanen Straßenverkehr mit mobilen Multisensorsystemen

Björn Borgmann

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr.-Ing. Christoph Holst
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Uwe Stilla
	2. Prof. Dr.-Ing. habil. Alexander Reiterer Universität Freiburg

Die Dissertation wurde am 23.09.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 26.01.2023 angenommen.

TECHNISCHE UNIVERSITÄT MÜNCHEN
TUM School of Engineering and Design

Detektion, Verfolgung und Posenschätzung von Personen im urbanen Straßenverkehr mit mobilen Multisensorsystemen

Björn Borgmann

Dissertation

2022

Kurzfassung

Fußgänger sind insbesondere im urbanen Straßenverkehr eine besonders gefährdete Gruppe an Verkehrsteilnehmern. Es ist in den kommenden Jahren zu erwarten, dass zunehmend mehr Fahrzeuge im Straßenverkehr mit Fahrerassistenzsystemen oder Systemen zum autonomen Fahren ausgestattet sind. Wenn die durch diese Systeme erfassten Daten vieler Fahrzeuge zusammengeführt werden, wird es möglich den urbanen Verkehrsraum permanent zu erfassen. So kann das Bewegungsverhalten von Fußgängern und anderen Verkehrsteilnehmern aufgenommen, dokumentiert und ausgewertet werden. Weiterhin können Karten und Stadtmodelle ergänzt und Gefährdungsbereiche erkannt werden, die sich anschließend durch bauliche oder regulatorische Maßnahmen beseitigen lassen.

In dieser Arbeit werden die Daten von LiDAR-Sensoren und Kameras verwendet, um Fußgänger zu detektieren, ihre Bewegung zu verfolgen und Informationen über ihre Körperpose zu gewinnen. Dabei werden die beiden untersuchten Sensormodalitäten für unterschiedliche Aufgaben genutzt. Personen werden in den Daten von LiDAR-Sensoren detektiert. Hierfür wird eine Methode vorgeschlagen, die einen abstimmbasierten Ansatz mit einem neuronalen Netz kombiniert. Dieses extrahiert gelernte Merkmale aus lokalen Punktnachbarschaften der LiDAR-Punktwolken und wertet diese aus. Das Ergebnis dieser Auswertung wird genutzt, um den Stimmraum eines an *Implicit Shape Models* angelehnten Abstimmverfahrens zu füllen, durch welches dann die Personen detektiert werden.

Das Detektionsverfahrens wird um ein Trackingverfahren ergänzt, um einerseits zusätzliche Informationen, zu ermitteln und andererseits das Detektionsverfahren zu unterstützen, wenn eine Person vorübergehend nicht detektiert werden kann. Der dritte Schwerpunkt dieser Arbeit ist eine Untersuchung LiDAR-Sensoren und Kameras gemeinsam zu verwenden, um Personen nicht nur zu detektieren, sondern in einem bildbasierten Verfahren auch eine Schätzung ihrer Körperpose durchzuführen. Hier werden drei unterschiedliche Varianten vorgeschlagen und verglichen, die sich darin unterscheiden welche Aufgaben von welcher Sensormodalität erfüllt werden.

Für die experimentelle Untersuchung der Verfahren werden Daten verwendet, die mit dem Multisensorfahrzeug MODISSA des Fraunhofer IOSB in Rahmen von Messfahrten in Ettlingen aufgezeichnet wurden. Diese wurden manuell von einem Menschen annotiert um eine Grundwahrheit zu erhalten. Für die Experimente werden drei unterschiedliche Sequenzen genutzt. Zusätzliche Daten wurden annotiert, um sie für das Training des neuronalen Netzes zu verwenden.

Im Hinblick auf die Detektion von Personen wird untersucht, wie sich das Verfahren bei unterschiedlicher Punktdichte und unterschiedlicher Menge von Trainingsdaten verhält. Es hat sich hierbei u.a. gezeigt, dass bei einem Training mit nur 325 Punktwolken, bei einer Genauigkeit von etwa 0,8 noch eine Sensitivität von 0,71 erzielt werden kann und bei einem Training mit nur 100 Punktwolken eine Sensitivität von 0,61. Für das Tracking hat sich gezeigt, dass es das Detektionsverfahren im Hinblick auf Personen die verdeckt sind ergänzen kann und dann zu 18 % besseren Ergebnissen führt.

Abstract

Pedestrians are a particularly vulnerable group of road users, especially in urban road traffic. In the coming years, it is expected that more and more vehicles on the road will be equipped with driver assistance systems or systems for autonomous driving. If the data gathered by these systems of many vehicles is combined, it will be possible to permanently record urban traffic. In this way, the movement behavior of pedestrians and other road users can be recorded, documented and evaluated. In this way, the movement behavior of pedestrians and other road users can be recorded, documented and evaluated. Furthermore, maps and city models can be supplemented and hazardous areas can be identified. Areas which can then be eliminated by structural or regulatory measures.

In this thesis, data from LiDAR sensors and cameras are used to detect pedestrians, track their movement and gather information about their body pose. To achieve this, the two sensor modalities will be used for different tasks. Persons are detected in the data of LiDAR sensors. For this, a method is proposed which combines a voting based approach with a neural network. The neural network extracts learned features from local point neighborhoods of LiDAR point clouds and evaluates them. The result of the evaluation is used to fill the voting space of a method inspired by *Implicit Shape Models* by which the persons are detected.

The detection is supplemented with a tracking. On the one hand to determine additional information and on the other hand to supplement the detection if a person can not be detected temporarily. The third focus of this thesis is an examination of using LiDAR sensors and cameras together to not only detect persons, but to also estimate their body pose in an image-based procedure. Here, three different variants are proposed compared. They differ in which tasks are performed with which sensor modality.

For the experimental evaluation, data recorded in Ettlingen by the multi-sensor vehicle MODISSA of the Fraunhofer IOSB is used. The data was annotated by a human in order to obtain a ground truth. Three different sequences are used for the experiments. Additional data was annotated to be used for the training of the neural network.

With regard to the person detection, it is evaluated how the method behaves with different point densities and different amounts of training data. It has been shown that with a precision of about 0.8 a recall of 0.71 can be achieved for a training with 325 point clouds and a recall of 0.61 for a training with only 100 point clouds. It is also shown that the tracking is able to supplement the detection with regard to temporarily occluded persons and then lead to 18 % better results.

Inhaltsverzeichnis

Liste der verwendeten Abkürzungen

Abkürzung	Beschreibung	Seite
2D	Zweidimensional	20
3D	Dreidimensional	18
ALS	Airborne Laser Scanning	35
CNN	Convolutional Neural Network	28
DATMO	Detection and Tracking of Moving Objects	31
DBMS	Datenbankmanagementsystem	85
DMI	(Applanix) Distance Measurement Instrument	79
ECEF	Earth-Centered, Earth-Fixed	82
ENU	East-North-Up	82
FPGA	Field Programmable Gate Array	82
GNN	Global Nearest Neighbor	30
GNSS	Globales Navigationssatellitensystem	37
IOSB	Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung	77
IMU	Inertial Measurement Unit	37
INS	Inertiales Navigationssystem	37
ISM	Implicit Shape Model	28
JPDAF	Joint probabilistic data association filter	31
LiDAR	Light Detection and Ranging	17
LLA	Latitude, longitude, altitude	82
MLP	Multi-layer perceptron	25
MLS	Mobile Laser Scanning	35
MODISSA	Mobile Distributed Situation Awareness. Multisensorfahrzeug des Fraunhofer IOSB, welches u.a. als Experimentalsystem genutzt wird.	77
MOTA	Multiple Object Tracking Accuracy	92
MOTP	Multiple Object Tracking Precision	92
NMEA	National Marine Electronics Association	81
PCS	(Applanix) POS Computer System	79
PPS	Puls pro Sekunde	81
RGB	Red, Green and Blue (image)	23
RGB-D	Red, Green, Blue and Depth (image)	23
ROI	Region of Interest	26
ROS	Robot Operating System	82
RPN	Region Proposal Network	29
SLAM	Simultaneous Localization and Mapping	38
SNK	Schwenk-Neigekopf	79
SQL	Structured Query Language	85
TLS	Terrestrial Laser Scanning	35
UTC	Universal Time Coordinated, Koordinierte Weltzeit	80

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Fußgänger stellen eine besonders verwundbare Gruppe der Verkehrsteilnehmer dar. Anders als z.B. Autofahrer sind sie kaum durch technische Maßnahmen, wie Sicherheitsgurte oder Airbags geschützt und haben daher ein hohes Risiko schwer verletzt zu werden, wenn sie in einen Verkehrsunfall verwickelt werden. Im Jahr 2019 kamen in Deutschland 30243 Fußgänger bei Verkehrsunfällen zu Schaden. Die Mehrzahl dieser Unfälle (95,45 %) geschah innerhalb geschlossener Ortschaften. In geschlossenen Ortschaften sind 11,58 % der Verkehrsunfallopfer Fußgänger, gleichzeitig machen sie dort aber 33,05 % der Verkehrstoten aus. Dies macht ihr erhöhtes Risiko für schwere Verletzungen im Falle eines Verkehrsunfalls deutlich [Statistisches Bundesamt (Destatis), 2021].

Bei dem Design von Fahrerassistenzsystemen und autonomen Fahrzeugen sollte daher ein besonderes Augenmerk auf die Sicherheit von Fußgängern gelegt werden, was vor allem beim Einsatz solcher Systeme im urbanen Umfeld eine wichtige Rolle spielt. Um dies zu ermöglichen ist es hilfreich, dass Fußgänger durch diese Systeme auch als solche erkannt und von anderen Verkehrsteilnehmern und Hindernissen unterschieden werden. Hierdurch können dann auch ihre Besonderheiten, z.B. im Hinblick auf ihr Bewegungsverhalten berücksichtigt werden. Fußgänger können beispielsweise sehr viel plötzlicher und freier ihre Bewegungsrichtung ändern als andere Verkehrsteilnehmer.

Es ist zu erwarten, dass es in den kommenden Jahren zunehmend mehr Fahrzeuge mit leistungsfähigen Fahrerassistenzsystemen sowie autonome bzw. teilautonome Fahrzeuge geben wird. Der in Abbildung 1.1 gezeigte Kleinbus ist ein Beispiel für ein solches in Entwicklung befindliches autonomes System, welches zukünftig den öffentlichen Nahverkehr ergänzen soll. Fahrerassistenzsysteme und autonome Fahrzeuge nutzen oft eine Kombination aus verschiedenen Sensoren und Sensormodalitäten. Dabei sind Kameras, Radar- und LiDAR-Sensoren, neben den schon seit längeren vor allem in Einparkhilfen verwendeten Ultraschallsensoren, die wohl am häufigsten verwendeten Modalitäten. So ausgestattete Fahrzeuge können damit auch als mobile Multisensorsysteme betrachtet werden. Diese Eigenschaft macht es interessant ihre Daten noch auf andere Weise für die Verbesserung der Fußgänger- und Verkehrssicherheit zu nutzen: Die Erfassung des öffentlichen Verkehrsraums mit Multisensorsystemen erfolgt bisher meist in der Form von einmaligen oder selten wiederholten Messkampagnen. Durch die Sensoren von Fahrerassistenzsystemen oder autonomen Fahrzeugen ist, bei einer entsprechend hohen Dichte solcher Fahrzeuge, hingegen eine semi-permanente Erfassung des Verkehrsraums mit verschiedenen Sensormodalitäten möglich. Wenn die dabei erfassten Daten mehrerer Fahrzeuge aggregiert und kombiniert werden, lassen sich daraus ggf. Aussagen bezüglich Verkehrsströmen und -dichte, Unfall- und Gefahrenschwerpunkten sowie Unfallursachen treffen. Ergebnisse, die dann zu baulichen oder regulatorischen Maßnahmen und damit zur Verbesserung der Verkehrssicherheit führen können.

Die Speicherung der vollständigen Sensordaten für die oben genannten Zwecke ist aber aus verschiedener Sicht problematisch. Sie lässt zwar alle Möglichkeiten bei der nachträglichen Aus-

Abbildung 1.1: Prototyp eines autonomen Kleinbusses ausgestattet u.a. mit LiDAR-Sensoren und Kameras

wertung offen, macht diese jedoch auch sehr aufwendig. Insbesondere dann, wenn die auszuwertenden Daten von vielen verschiedenen Sensorsystemen stammen, da bei deren Auswertung oft die unterschiedlichen Charakteristika dieser Sensorsysteme berücksichtigt werden müssen. Auch wird für die Speicherung der vollständigen Sensordaten viel Speicherplatz benötigt. Weshalb sie sich auch nur aufwendig an eine zentrale Stelle transferieren lassen, um sie dort zu sammeln und auszuwerten. Zuletzt ist insbesondere bei Kameras die langfristige Speicherung aller Daten auch aus Sicht des Datenschutzes problematisch. Eine Alternative ist es, die Daten noch auf innerhalb der Sensorsysteme selbst auszuwerten, was häufig sowieso bereits als Teil ihrer primären Funktion passiert. Die Ergebnisse solcher Datenauswertungen, in verschiedenen Sensorsystemen, können dann zusammengetragen und wie oben beschrieben genutzt werden.

Die Auswertung der Daten in den Sensorsystemen kann von der Ausnutzung der verschiedenen Stärken der unterschiedlichen Sensormodalitäten profitieren. So können LiDAR-Sensoren gut dafür genutzt werden, die 3D-Position von Objekten in der Umgebung zu bestimmen. Verglichen mit Kameras sind die von ihnen erfassten Daten jedoch üblicherweise niedriger aufgelöst und sie umfassen keine Farbinformation. Daher sind weniger gut geeignet Details wie z.B. die Arm- und Beinposition oder die Blickrichtung eines Fußgängers zu erfassen. In Bezug auf Fußgänger könnte die Datenauswertung auf den Sensorsystemen z.B. folgende Ergebnisse liefern: Den Pfad, welchen Fußgänger im Erfassungsbereich des Sensorsystems zurücklegen sowie Informationen über deren Körperhaltung und Blickrichtung zu den verschiedenen Erfassungszeitpunkten.

1.2 Forschungsfragen

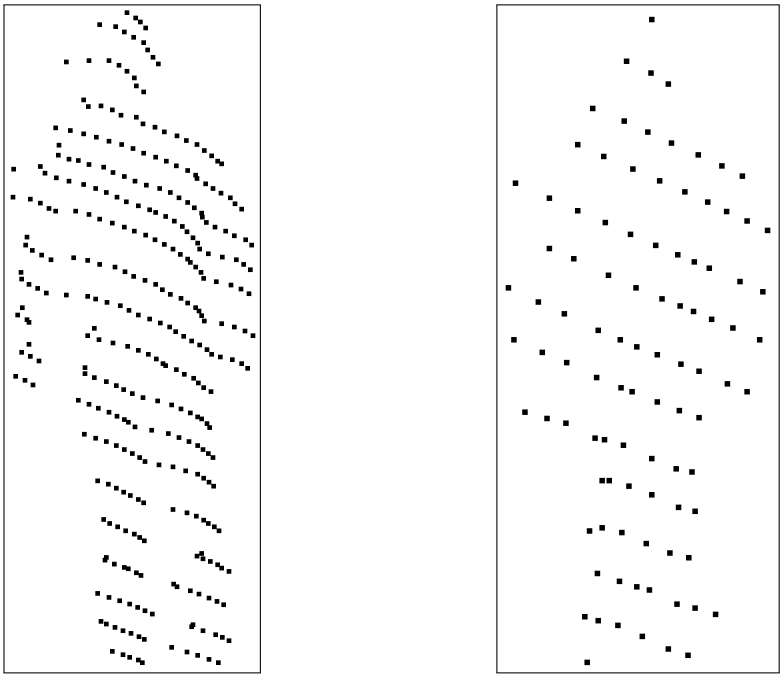
Diese Arbeit untersucht die Gewinnung von Informationen über Fußgänger aus den Daten mobiler Multisensorsysteme. Die grundsätzliche Idee dabei ist es, die verschiedenen Sensormodalitäten entsprechend ihrer individuellen Stärken für spezifische Teilaufgaben zu nutzen. Dabei wird vor-

geschlagen, LiDAR-Sensoren für die Detektion und das Tracking von Fußgängern im Umfeld des Systems zu nutzen. Kameras wiederum sollen gezielt genutzt werden, um Detailinformationen wie z.B. die Körperpose dieser Fußgänger zu bestimmen.

Die Arbeit soll folgende Forschungsfragen behandeln:

1. Welche Detektionsleistung von Personen kann in 3D-Punktwolken erreicht werden und wie stark hängt diese von der Menge an verwendeten Trainingsdaten, der Punktdichte bzw. der Entfernung der Personen zum Sensor ab?
2. Welche Trackinggenauigkeit kann erzielt werden und wie verändert sich die Detektionsleistung des Gesamtsystems durch das Tracking?
3. Welche Detektionsleistung erzielen unterschiedliche Varianten zur multimodalen Körperposenschätzung und Detektion von Personen und wie unterscheiden sich diese Varianten in Bezug auf die Verarbeitungszeit?

1.2.1 Detektion von Personen in 3D-Punktwolken



(a) (b)

Abbildung 1.2: Person in LiDAR-Punktwolke in unterschiedlichen Entfernungen [Borgmann et al., 2017a]. Aufnahme mit einem Velodyne HDL-64E bei 10 Umdrehungen des Scankopfes pro Sekunde. Zu sehen ist die gleiche Person aufgenommen in verschiedenen Entfernungen und davon abhängig mit verschiedener Punktdichte, a) Person in 7,20 m Entfernung, b) Die gleiche Person in 19,50 m Entfernung

Methoden zur Objektdetektion verwenden mittlerweile mehrheitlich Verfahren des maschinellen Lernens. Um diese zu nutzen, müssen sie zunächst trainiert werden. Ein hierfür häufig verwendetes Vorgehen ist das überwachte Lernen, bei welchem dem Verfahren des maschinellen Lernens Trainingsdaten präsentiert werden, für welche die gestellte Aufgabe bereits anderweitig gelöst wurde. Solche Trainingsdaten zu erstellen ist häufig sehr aufwendig, weswegen sie oft nur in begrenzter Menge verfügbar sind.

Bei der Detektion von Objekten in LiDAR-Punktwolken wird vor allem deren Daten- bzw. Punktdichte als begrenzender Faktor angesehen. Je geringer diese ist, umso schwieriger wird es Objekte noch korrekt zu erkennen. Die Punktdichte mit der ein Objekt erfasst wird, hängt üblicherweise, wie in Abbildung 1.2 dargestellt, von der Entfernung dieses Objekts zum verwendeten LiDAR-Sensor ab. Bei der MLS-Datenerfassung variiert diese stark. Ein Sensorsystem, das beispielsweise an einem Passanten vorbeifährt, nähert sich an diesen zunächst an und entfernt sich dann wieder vom ihm.

Es soll aufgezeigt werden, welche Sensitivität und Genauigkeit eine in dieser Arbeit vorgestellte Methode für die Personendetektion in LiDAR-Daten erreicht. Es soll außerdem beantwortet werden, wie stark diese von der Entfernung zwischen Objekt und Sensor und von der Menge an verfügbaren Trainingsdaten abhängen. Hierfür ist es erforderlich zunächst ein Detektionsverfahren zu entwickeln, welches in Hinblick auf diesen beiden Kriterien optimiert wurde.

1.2.2 Tracking von Personen in 3D-Punktwolken

Das Verfolgen von einmal detektierten Personen dient zwei Zwecken: Zum einen kann es dazu genutzt werden die Detektion zu unterstützen. Ein Trackingverfahren, welches in der Lage ist basierend auf vergangenen Detektionen die künftige Position einer Person zu antizipieren, kann helfen Phasen zu überbrücken, in denen eine Person z.B. durch Verdeckungen nicht mehr detektiert werden kann. Ein Tracking liefert aber auch zusätzliche Daten, da es möglich wird dieselbe Person über mehrere Zeitschritte zu verfolgen und so deren Bewegungspfad und Geschwindigkeit zu bestimmen.

Es soll beantwortet werden, welche Trackinggenauigkeit erzielt wird, also wie gut eine Objektidentität im Zeitverlauf verfolgt werden kann. Es soll außerdem dargestellt werden, wie sehr das Tracking bei der fortlaufenden Erkennung von Personen hilft, die durch starke Verdeckungen zeitweise nicht vom Detektionsverfahren erfasst werden.

1.2.3 Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern

Ein Vorteil eines Multisensorsystems ist es, verschiedene Sensormodalitäten kombinieren zu können, um ein umfassenderes Gesamtbild zu erhalten. Dies soll am Beispiel der Detektion und Körperposenschätzung von Personen im 3D-Raum untersucht werden. Hierbei liefern LiDAR-Sensoren die 3D-Information, während Kameras detaillierte Aufnahmen der Personen für die Posenschätzung zur Verfügung stellen, mit denen sich zunächst aber nur Merkmale in 2D-Bildkoordinaten bestimmen lassen. Es werden verschiedene Ansätze zur Datenfusion dieser Modalitäten untersucht, um ein umfassendes Gesamtbild mit 3D-Koordinaten der Personen sowie Informationen über deren Körperpose zu erhalten. Hierbei werden die Vor- und Nachteile der verschiedenen Ansätze im Hinblick auf die Laufzeit der Datenverarbeitung und die Qualität der Ergebnisse bestimmt.

1.3 Beiträge dieser Arbeit

Diese Arbeit leistet Beiträge zu dem publizierten Stand der Forschung in folgenden Bereichen:

- Entwicklung und Untersuchung eines Verfahrens zur Detektion von Personen bzw. Objekten in 3D-Punktwolken, welches diese anhand ihres Erscheinungsbilds und nicht ihres Kontexts detektiert und welches ein neuronales Netz mit einem an *Implicit Shape Models* orientierten Abstimmverfahren kombiniert.

- Entwicklung und Untersuchung eines neuronalen Netzes als Teil des Detektionsverfahrens, dessen Komplexität bewusst für ein einfacheres Training begrenzt wurde.
- Untersuchung wie ein ergänzendes Tracking ein Detektionsverfahren bei Verdeckungen der zu detektierenden Objekte unterstützen kann.
- Untersuchung mehrerer Varianten der gemeinsamen Verwendung von RGB-Kameras und LiDAR-Sensoren am Beispiel der Detektion und Körperposenschätzung von Personen.

Teile dieser Arbeit wurden in den folgenden Veröffentlichungen vorgestellt:

- [Borgmann et al., 2017b] Borgmann B, Hebel M, Arens M, Stilla U (2017b) Konzept zur Gefährdungserkennung im städtischen Verkehrsraum durch Personendetektion in MLS-Punktwolken. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., 26: 262–275.
- [Borgmann et al., 2017a] Borgmann B, Hebel M, Arens M, Stilla U (2017a) Detection of persons in MLS point clouds using implicit shape models. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W7: 203–210.
- [Borgmann et al., 2018a] Borgmann B, Hebel M, Arens M, Stilla U (2018a) Fußgängerbezogene Informationsgewinnung zur Situationsanalyse mit einem mobilen Multisensorsystem. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., 27: 363–375.
- [Borgmann et al., 2018b] Borgmann B, Hebel M, Arens M, Stilla U (2018b) Usage of multiple LiDAR sensors on a mobile system for the detection of persons with implicit shape models. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2: 125–131.
- [Borgmann et al., 2018c] Borgmann B, Schatz V, Kieritz H, Scherer-Klöckling C, Hebel M, Arens M (2018c) Data processing and recording using a versatile multi-sensor vehicle. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1: 21–28.
- [Borgmann et al., 2019] Borgmann B, Hebel M, Arens M, Stilla U (2019) Using neural networks to detect objects in MLS point clouds based on local point neighborhoods. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W7: 17–24.
- [Borgmann et al., 2020] Borgmann B, Hebel M, Arens M, Stilla U (2020) Pedestrian detection and tracking in sparse MLS point clouds using a neural network and voting-based approach. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, V-2-2020: 187–194.
- [Borgmann et al., 2021a] Borgmann B, Hebel M, Arens M, Stilla U (2021a) Information acquisition on pedestrian movements in urban traffic with a mobile multi-sensor system. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2021: 131–138.
- [Borgmann et al., 2021b] Borgmann B, Schatz V, Hammer M, Hebel M, Arens M, Stilla U (2021b) MODISSA: A multipurpose platform for the prototypical realization of vehicle-related applications using optical sensors. Applied Optics, 60 (22): F50–F65.

1.4 Aufbau der Arbeit

Nach dieser Einleitung ist die weitere Arbeit wie folgt aufgebaut: Kapitel 2 gibt den Stand der Forschung in verschiedenen thematisch verwandten Bereichen wieder. Diese umfassen neuronale Netze für die Verarbeitung von 3D-Punktwolken, sowie die Detektion und das Tracking von Objekten in 3D-Punktwolken.

In Kapitel 3 werden für den weiteren Verlauf der Arbeit relevante Grundlagen vorgestellt. Diese umfassen das Mobile Laser Scanning, welches für die Erfassung der Daten genutzt wird, die von den meisten in dieser Arbeit vorgestellten Methoden verarbeitet werden. Außerdem werden künstliche neuronale Netze und die Verwendung von Kalman-Filtern zum Tracking erläutert. Diese bilden jeweils die technische Grundlage, für eine der in den nachfolgenden Kapiteln vorgestellten Methoden. Abschließend wird in Abschnitt 3.4 eine kurze Übersicht über verschiedene Verfahren zur Körperposenschätzung in Bilddaten gegeben, was die Grundlage für die spätere Auswahl eines solchen Verfahrens für Zwecke dieser Arbeit ist.

Kapitel 4 stellt den ersten methodischen Schwerpunkt dieser Arbeit vor und erläutert das entworfene Verfahren zur Detektion von Personen in 3D-Punktwolken. Dieses umfasst eine Bodenextraktion zur Vorverarbeitung der Daten, die anschließend in lokale Punktnachbarschaften zerlegt und in dieser Form von einem neuronalen Netz verarbeitet werden. Das Ergebnis des neuronalen Netzes wird dann in einem Abstimmverfahren verwendet, um Personen zu detektieren.

Kapitel 5 stellt den zweiten methodischen Schwerpunkt dieser Arbeit dar und beschreibt ein Trackingverfahren, welches die Detektion ergänzt, um einerseits zusätzliche Informationen über die Objekte zu gewinnen und es andererseits erlaubt auch dann noch eine Position für bekannte Personen zu bestimmen, wenn diese z.B. aufgrund von Verdeckungen vorübergehend nicht mehr detektiert werden können.

Kapitel 6 stellt den dritten methodischen Schwerpunkt dar. Hier werden drei verschiedene Varianten einer Methode beschrieben, welche die 3D-Punktwolken gemeinsam mit RGB-Bildern nutzen um Personen nicht nur zu detektieren, sondern auch um zusätzliche Informationen über diese zu gewinnen. Dies wird am Beispiel einer Körperposenschätzung untersucht.

In Kapitel 7 werden zunächst das genutzte Experimentalsystem und die mit diesem für die Experimente erfassten Daten erläutert. Anschließend werden die Experimente selbst sowie das Vorgehen beim Bewerten ihrer Ergebnisse dargestellt.

Die Ergebnisse der Experimente werden in Kapitel 8 in bildlicher und tabellarischer Form vorgestellt. Sie sind entsprechend der drei methodischen Schwerpunkte dieser Arbeit aufgegliedert.

In Kapitel 9 werden die vorgestellten methodischen Schwerpunkte sowie die Ergebnisse der zu ihrer Untersuchung durchgeführten Experimente umfassender diskutiert, interpretiert und bewertet.

Diese Arbeit schließt mit Kapitel 10 ab, in dem die Arbeit abschließend zusammengefasst wird und die Forschungsfragen beantwortet werden. Es werden außerdem Ansätze für weitere Arbeiten erläutert.

2 Stand der Forschung

In diesem Kapitel wird der Stand der Forschung in verschiedenen Themenbereichen beschrieben, die für die in dieser Arbeit behandelte Thematik relevant sind.

2.1 Neuronale Netze für die Verarbeitung von 3D-Punktwolken

Neuronale Netze sind ein wichtiges Mittel zur Lösung einer Vielzahl von Problemstellungen in der Datenauswertung. Sie werden beispielsweise in der Bildauswertung, für die Objektdetektion und -klassifizierung aber auch für das semantische Labeling verwendet. Hierbei sind vor allem *Convolutional Neural Network* (CNN) besonders erfolgreich, die in der Lage sind Daten ohne vom Menschen definierte Merkmale zu verstehen und stattdessen, entsprechend ihrer jeweiligen Aufgabe, passende Merkmale selbst lernen. CNN verwenden diskrete Faltungen als Basis und nutzen dabei das Pixelraster, welches Bilder üblicherweise haben, um solche Faltungen auf jeweils einen Bereich benachbarter Pixel anzuwenden. Obwohl es naheliegend ist neuronale Netze auch für die Verarbeitung von 3D-Punktwolken zu verwenden, ist dies nicht ohne weiteres möglich, da diese entweder gar keine inhärente Rasterstruktur haben oder keine, die alle vorhandenen räumlichen Dimensionen abdeckt. Es wurden daher bereits verschiedene Strategien entwickelt, um LiDAR-Punktwolken mit neuronalen Netzen zu verarbeiten.

Überführen der Punktwolken in Pixelraster

Eine Methode ist es, Punktwolken in ein Pixelraster zu überführen. Dabei bilden zwei der Raumdimensionen die Pixelkoordinaten, während die Dritte einen „Farbkanal“ des Pixels bildet. Daraus resultieren dann Tiefenbilder, bei denen der Farbwert der Pixel die Entfernung zum Sensor darstellt. Ein Problem bei der Erstellung dieser Tiefenbilder ist, dass die Punktwolken nicht immer eine ausreichend hohe Punktdichte haben, um für alle Pixel des Bildes direkt einen Tiefenwert zu liefern. Um diese Schwierigkeit zu umgehen, werden Verfahren eingesetzt, um die Tiefenwerte der einzelnen Pixel aus den Koordinaten mehrerer Punkte zu interpolieren. Asvadi et al. [2017] nutzen hierfür eine Delaunay-Triangulation, um aus der Punktwolke ein Dreiecksnetz zu erstellen. Der Tiefenwert für die Pixel innerhalb eines solchen Dreiecks ergibt sich dann aus den Tiefenwerten der Eckpunkte des Dreiecks. Die daraus resultierenden Tiefenbilder klassifizieren sie dann mithilfe eines CNN.

Eine Stärke bei der Verwendung von Tiefenbildern ist, dass diese eine direkte Möglichkeit darstellen eine Datenfusion zwischen Kameras und LiDAR-Sensoren durchzuführen, indem die Pixel sowohl die drei RGB-Farbkanäle als auch einen Tiefenkanal haben. Die sich so ergebenen Bilder werden auch als RGB-D (*Red, Green, Blue and Depth*) Bilder bezeichnet. Gao et al. [2018] nutzen einen solchen Ansatz und verarbeiten RGB-D Bilder in einem CNN für die Klassifikation von Fußgängern, Radfahrern, Autos und LKWs in den Daten eines Multisensorsystems, ausgestattet mit LiDAR-Sensoren und Kameras. Das RGB-D Tiefenbild wird bei ihrem Ansatz erzeugt, indem das RGB-Bild einer Kamera als Basis genommen wird. In dieses RGB-Bild werden die Punkte

der LiDAR-Punktwolke projiziert. Da die Punktwolke weniger dicht ist als das RGB-Bild und aufgrund der unterschiedlichen Position und Ausrichtung von Kamera und LiDAR-Sensor, landen die Punkte nicht exakt auf den Pixeln des RGB-Bildes. Die Tiefenwerte für die einzelnen Pixel ergeben sich daher aus mehreren benachbarten Punkten, deren Tiefenwert abhängig von der jeweiligen Distanz zwischen Punkt und Pixel berücksichtigt wird. Socher et al. [2012] verarbeiten ebenfalls RGB-D Bilder um diese zu klassifizieren, nutzen aber zwei weitestgehend getrennte neuronale Netze für die RGB-Kanäle und den Tiefenkanal des Bildes. Deren Ausgabe wird dann erst später zu einem gemeinsamen Ergebnis fusioniert.

Überführen der Punktwolken in Voxelraster

Die Nutzung von Tiefenbildern hat den Nachteil, dass die drei räumlichen Dimensionen nicht gleichbehandelt werden. Die unterschiedliche Betrachtung von Höhe und Breite auf der einen Seite und Tiefe auf der anderen führt dazu, dass sich ein Objekt, je nach Position und Ausrichtung, in den Daten stark unterschiedlich darstellt. Eine Objekteigenschaft, die von vorne betrachtet zu einer Änderung in dem Tiefenwert eines Pixels führt, führt von der Seite betrachtet zu einer Änderung der Pixelkoordinaten. Eine Alternative, die diese Schwierigkeit nicht hat, ist es statt ein 2-dimensionales Pixelraster ein 3-dimensionales Voxelraster zu verwenden. Voxel sind das 3-dimensionale Äquivalent zu Pixeln und daher die naheliegende Alternative zur Verarbeitung von 3-dimensionalen Daten. In einem CNN können dann dementsprechend statt diskrete 2D-Faltungen diskrete 3D-Faltungen verwendet werden.

Maturana & Scherer [2015] nutzen einen solchen Ansatz für die Objektklassifizierung. Sie erzeugen zunächst ein *Occupancy Grid*, in welchem die Voxel den Zustand „belegt“, „frei“ oder „unbekannt“ haben können. Die Unterscheidung zwischen freien und unbekanntem Raum stellt dabei eine zusätzliche Information dar, die vom neuronalen Netz genutzt werden kann. Sie ergibt sich aus der Annahme, dass der Raum, den ein Laserstrahl eines LiDAR-Sensors durchquert hat, bevor er auf eine Oberfläche getroffen ist, frei sein muss. Der Raum, in dem keine Oberfläche gemessen wurde, den aber auch kein Laserstrahl durchquert hat, kann hingegen entweder frei oder belegt sein. Er wird daher als unbekannt betrachtet. Hierbei handelt es sich um die Bereiche, welche entweder außerhalb des Sichtbereichs des Sensors liegen oder die durch andere Oberflächen verdeckt sind. Maturana & Scherer [2015] haben die Verwendung von unterschiedlichen Varianten eines solchen Occupancy Grids untersucht, bei denen die Zellen entweder einen binären oder einen kontinuierlichen Dichtewert haben. Letzterer gibt eine Wahrscheinlichkeit an, mit der ein bestimmtes Voxel von einem Laserstrahl durchquert werden kann. Er erlaubt es also besser mit Zellen umzugehen, die nur von einem Teil der Strahlen durchquert werden konnten, also teilweise aber nicht vollständig von einer Oberfläche belegt sind. Maturana & Scherer [2015] nehmen an, dass ihr Verfahren zuvor generierte Punktwolkensegmente verarbeitet, die ein einzelnes Objekt enthalten. Sie nehmen außerdem an, dass diese Segmente ein definiertes Koordinatensystem haben, dessen z-Achse entsprechend der Höhenrichtung ausgerichtet ist. Im Bezug auf Rotationen um die z-Achse sind die verwendeten Koordinatensysteme jedoch undefiniert. Da das von ihnen verwendete neuronale Netz gegenüber derartigen Rotationen um die z-Achse nicht invariant ist, augmentieren sie stattdessen die Trainingsdaten und erzeugen von jeder Objektinstanz mehrere Kopien, die jeweils unterschiedlich stark um die z-Achse rotiert sind. Einen ähnlichen Ansatz, ebenfalls zur Objektklassifizierung, verwenden Garcia-Garcia et al. [2016].

Verwendung von neuronalen Netzen ohne Diskretisierung der Punktwolken

Sowohl die Verwendung von Pixelrastern als auch von Voxelrastern hat den Nachteil, dass die Daten der Punktwolke in die Zellen des Rasters (Pixel bzw. Voxel) diskretisiert werden müssen. Eine solche Diskretisierung hat mehrere Schwierigkeiten: Zum einen muss die richtige Größe für

die Rasterzellen gewählt werden. Diese passt im Idealfall zu der vorliegenden Punktdichte, damit einerseits kein Informationsverlust durch das Zusammenfassen vieler Punkte in einer Zelle stattfindet, gleichzeitig aber auch genügend Punkte vorhanden sind, um jede Zelle sinnvoll zu füllen. Die Schwierigkeit dabei ist, dass die Punktdichte in einer LiDAR-Punktewolke üblicherweise von der Entfernung zum Sensor abhängt. Sie kann also in der aufgenommenen Umgebung, abhängig von dieser Entfernung, stark variieren. Eine im Nahbereich optimale Zellengröße ist daher in größerer Entfernung evtl. nicht mehr passend. Um mit diesen Schwierigkeiten umzugehen, wurden verschiedene Ansätze entwickelt. So lassen sich, mit verschiedenen bereits erläuterten Methoden zur Interpolation aus mehreren der Zelle nahe liegenden Punkten, Informationen für diese ermitteln, wenn kein Punkt direkt in eine Zelle fällt [Asvadi et al., 2017; Gao et al., 2018]. Wenn stattdessen mehrere Punkte mit widersprüchlichen Informationen in eine Zelle fallen, kann der daraus resultierende Informationsverlust durch einen kontinuierlichen anstatt binären Wert für die Zelle minimiert werden [Maturana & Scherer, 2015]. Trotz dieser Maßnahmen tritt durch die Diskretisierung jedoch ein Verlust an Informationen auf, da die Oberflächenstruktur innerhalb einer Zelle auch so nicht berücksichtigt werden kann. Auch wird entsprechend der Zellengröße die genaue Position der Punkte und damit die gemessene Oberfläche verwischt. Methoden, die nicht auf ein solches Raster angewiesen sind und so alle diese Schwierigkeiten vermeiden, können daher Vorteile haben.

Der von Qi et al. [2017a] vorgestellte *PointNet*-Ansatz ist ein neuronales Netz, welches in der Lage ist eine unstrukturierte Reihe von 3D-Punkten zu verarbeiten. Es kann dementsprechend jede Art von 3D-Punktwolken oder auch Segmente solcher Punktwolken verarbeiten, ohne diese vorher in ein Pixel- oder Voxelraster zu überführen. PointNet lernt dabei eine symmetrische Funktion, deren Ergebnis nicht von der Reihenfolge der Eingabedaten abhängt. Hierfür verwendet das neuronale Netz ein *Multi-layer perceptron* (MLP), welches die Punkte der Eingabe individuell verarbeitet, aber von allen dieser Punkte geteilt wird. Anschließend werden die Ergebnisse, der Verarbeitung der individuellen Punkte mithilfe eines *Max-poolings* zu einem gemeinsamen globalen Merkmal zusammengefasst. Dieses kann dann z.B. in einem weiteren MLP verarbeitet werden, um die gesamten Daten zu klassifizieren oder mit Informationen auf Einzelpunktebene aus einer früheren Schicht des Netzes kombiniert werden, um ein Merkmal auf Punktebene zu erzeugen, welches sowohl den Kontext der gesamten Daten als auch Eigenschaften des einzelnen Punktes berücksichtigt. Basierend darauf kann dann z.B. ein semantisches Labeling erfolgen, bei dem jedem einzelnen Punkt eine Klasse zugeordnet wird. PointNet ist invariant im Hinblick auf die Reihenfolge der verarbeiteten Daten. Um auch invariant dagegen zu sein, wo und mit welcher Ausrichtung diese Daten im zugrundeliegenden Koordinatensystem liegen, verfügt PointNet über ein Subnetz, welches lernt eine Transformationsmatrix zu erzeugen, die die Daten in eine Standardlage im Koordinatensystem überführt.

PointNet wurde später um eine hierarchische Komponente zu *PointNet++* erweitert. Hier werden zunächst lokale Subsets der Punktwolke erstellt. Diese sind über einen ausgewählten Mittelpunkt definiert und umfassen die umliegenden Punkte. Für sie wird dann mit einem PointNet ein Merkmalsvektor bestimmt. Benachbarte Subsets und ihre Merkmale werden anschließend zusammengefasst, um wiederum mit einem PointNet einen gemeinsamen Merkmalsvektor zu bilden. Dieser Schritt kann über mehrere hierarchische Ebenen wiederholt werden, um zum letztlichen gemeinsamen Merkmalsvektor für die Daten zu kommen. Dieses Vorgehen bildet das Verhalten eines CNN nach, welches ebenfalls über mehrere Ebenen einen immer größeren Ausschnitt der Daten betrachtet, also die Merkmale für einen größeren Ausschnitt der Daten jeweils aus denen eines kleineren bildet. [Qi et al., 2017b]

Liu et al. [2019] verwenden in ihrem *DensePoint*-Ansatz einen speziell entworfenen Faltungs-Operator *PConv*, der invariant gegenüber der Reihenfolge der Daten ist. Mit diesem Operator wird

für die Punkte und ihre Nachbarn ein Merkmalsvektor ermittelt. Bei PConv handelt es sich dabei um ein *Single-layer perceptron*. Dieser Prozess wird dann in mehreren Stufen, mit einem immer größeren Radius der Nachbarschaft, wiederholt. Hierbei wird jeweils die Ausgabe der vorherigen Stufe als Input für die nachfolgende verwendet. Das jeweilige Ergebnis von PConv auf einer Ebene wird dann jedoch noch mit dem Ergebnis aller vorherigen Ebenen konkateniert. Der auf einer bestimmten Ebene resultierende Merkmalsvektor wird dementsprechend mit jeder Ebene größer. Auch hier werden wieder aus Merkmalen für einen lokalen Ausschnitt der Daten die Merkmale für einen größeren Ausschnitt gebildet.

Zhou & Tuzel [2018] verwenden eine Mischform aus den Ansätzen die Punktwolken als ungeordnete Gruppe von Punkten oder als Voxelraster zu verarbeiten. Sie unterteilen die Daten in Voxel, betrachten dann aber zunächst die Punkte innerhalb dieser Voxel. Bei Voxeln, in denen die lokale Datendichte hoch ist und in denen dementsprechend viele Punkte liegen, wenden sie eine zufällige Auswahl an, um die Menge der weiter betrachteten Punkte zu reduzieren. Mithilfe dieser Punkte werden dann unter Nutzung eines neuronalen Netzes Merkmale für die Voxel gebildet. Diese Merkmale berücksichtigen die Einzelpunkte und die lokale Struktur innerhalb der Voxel. Diese gehen also durch das Bilden der Voxel nicht als Informationsquelle verloren. Die daraus resultierenden Merkmale und ihre Voxel werden dann als Voxelraster in einem CNN weiter verarbeitet.

Einordnung dieser Arbeit

In dieser Arbeit wird ein neuronales Netz als Teil eines Verfahrens zur Objektdetektion in LiDAR-Punktwolken verwendet. Dieses greift das Konzept des PointNet-Ansatzes von [Qi et al., 2017a] auf, nutzt diesen aber um kleinere Segmente der Punktwolken sog. lokale Punktnachbarschaften zu verarbeiten. Anders als bei PointNet++ werden deren Merkmale anschließend jedoch nicht in mehreren Stufen vom neuronalen Netz für immer größere Raumbereiche zusammengefasst. Das neuronale Netz wird also gezielt dafür genutzt, Informationen aus kleinen lokalen Bereichen der Punktwolke zu gewinnen. Diese werden dann von dem weiteren Detektionsverfahren verwendet. Ziel ist es mit einem neuronalen Netz auszukommen, welches eine vergleichsweise geringe Komplexität hat und mit relativ wenig Aufwand trainiert werden kann.

2.2 Detektion von Objekten in 3D-Punktwolken

Es wurde bereits eine Vielzahl an Methoden entwickelt, um Personen oder allgemein *Objekte* in den Daten von LiDAR-Sensoren zu detektieren. Diese Daten liegen dabei meist zunächst in der Form von 3D-Punktwolken vor. Die Methoden können anhand ihres generellen Vorgehens in zwei Gruppen eingeteilt werden: „Segmentierung mit anschließender Klassifikation“ und „Detektion ohne Segmentierung“.

Segmentierung von 3D-Punktwolken mit anschließender Klassifikation

Die Segmentierung mit anschließender Klassifikation zerlegt die Aufgabe der Objektdetektion in die beiden Teilaufgaben Segmentierung und Klassifikation der Segmente. Hierbei werden die zu verarbeitenden Daten zunächst in Segmente eingeteilt, bei denen man davon ausgeht, dass sie ein einzelnes Objekt enthalten. Dieser Schritt wird auch als Hypothesengenerierung bezeichnet. Anschließend werden die erzeugten Segmente klassifiziert, um zu entscheiden, welche Art von Objekt sie enthalten. Manchmal wird vor der Klassifizierung noch eine Heuristik angewendet, um Segmente, die gegen bestimmte Kriterien verstoßen und daher keines der gesuchten Objekte enthalten können, bereits vor der eigentlichen Klassifizierung auszufiltern. Auch kann vor der Segmentierung bereits eine *Region of Interest* (ROI) definiert werden, um die Verarbeitung der

Daten auf den Bereich zu beschränken, in dem die gesuchten Objekte erwartet werden. Wenn z.B. nur bodengebundene Objekte detektiert werden sollen, hat es ggf. wenig Nutzen auch Daten, die weit über dem Boden aufgenommen wurden, zu verarbeiten. Auch der Boden selbst wird oft bereits vor der Segmentierung mithilfe eines Bodenextraktionsverfahrens aus den Daten entfernt.

Für die Segmentierung wird von Velizhev et al. [2012] zur Objektdetektion im urbanen Umfeld und von Wang et al. [2017] zur Personendetektion das sog. *Region Growing* verwendet. Bei diesem werden alle Elemente der zu verarbeitenden Daten (z.B. Punkte einer Punktwolke) zunächst als ein eigenes Segment angesehen. Anschließend werden zwei dieser Segmente zusammengefasst, wenn ihr Abstand einen gewissen Schwellwert unterschreitet. Dieser Schritt wiederholt sich bis es keine zwei Segmente mehr gibt, deren Abstand den gewählten Schwellwert unterschreitet. Beide eben zitierte Verfahren enthalten eine Bodenextraktion, die der eigentlichen Segmentierung vorangestellt wird und die notwendig ist, da ansonsten alle bodengebundenen Objekte über den Boden miteinander verbunden wären. Während Velizhev et al. [2012] direkt die Punktwolke segmentieren, überführen Wang et al. [2017] diese zunächst in ein 2D *Occupancy Grid*. Wang et al. [2017] filtern nach der Segmentierung Segmente aus, die aufgrund ihrer geometrischen Ausdehnung nicht als Person infrage kommen. Velizhev et al. [2012] setzen einen ähnlichen Filter ein, entfernen aber zusätzlich Segmente, die zu hoch über dem Boden sind oder die zu wenige Punkte umfassen.

Asvadi et al. [2017] verwenden hingegen in ihrer Methode für die Detektion von Fahrzeugen in LiDAR-Daten *DBSCAN* für die Segmentierung. Hierbei handelt es sich um ein Clustering-Verfahren, bei dem Punkte zusammengefasst werden, wenn sie als dichte-verbunden gelten. Bei DBSCAN werden Punkte bis zu einer bestimmten Entfernung voneinander als Nachbarn betrachtet. Zusätzlich wird zwischen Kern- und Randpunkten eines Clusters unterschieden. Ein Parameter *MinPts* definiert dabei, ob ein Punkt ein Kernpunkt ist: Jeder Punkt, der mindestens *MinPts* Nachbarn hat, gilt als Kernpunkt. Jedes Paar an Punkten, die über eine Kette von Kernpunkten verbunden sind, gelten als dichte-verbunden und sind Teil desselben Clusters [Ester et al., 1996].

Zhao et al. [2019] verwenden ebenfalls DBSCAN für die Segmentierung. Sie passen den *MinPts* Parameter jedoch in mehreren Schritten abhängig von der Entfernung zum LiDAR-Sensor an. Hierdurch können sie die von der Entfernung zum Sensor abhängige variierende Punktdichte berücksichtigen und für die jeweilige mögliche Punktdichte optimale Parameter für DBSCAN wählen. Sie verwenden sowohl eine ROI als auch ein Ausfiltern des Hintergrundes, um die Menge der zu verarbeitenden Daten bereits vor der Segmentierung zu reduzieren. Sie können dabei den Hintergrund gut erkennen, da ihr Verfahren für stationäre an Straßenkreuzungen platzierte Sensoren entworfen wurde. Als Hintergrund angesehen werden die Teile der aufgenommenen Daten, die sich über längere Zeit nicht verändern.

Ein häufiges Problem bei einer Segmentierung besteht darin, dass es zu einer Unter- bzw. Übersegmentierung kommen kann. Bei einer Untersegmentierung sind mehrere Objekte Teil desselben Segments. Bei einer Übersegmentierung hingegen wird ein Objekt in mehrere Segmente aufgeteilt. Beides führt zu Schwierigkeiten bei der weiteren Verarbeitung und muss entsprechend behandelt werden. Behley et al. [2013] nutzen eine hierarchische Segmentierung, um dieses Problem zu umgehen. Bei dieser werden über mehrere Stufen immer feinere Segmente gebildet. Die Segmente der unterschiedlichen Hierarchiestufen werden dann klassifiziert. Anschließend wird bei widersprüchlichen Klassifizierungen auf mehreren Hierarchiestufen nur diejenige verwendet, welche die höchste Konfidenz aufweist.

Auch für die Klassifizierung werden eine Reihe verschiedener Verfahren verwendet. Ein mögliches Verfahren zur Klassifikation sind *Support Vector Machines*. Diese setzen auf eine zuvor erfolgte Merkmalsextraktion. In einem Trainingsverfahren werden Hyperebenen im Merkmalsraum ermittelt, welche die verschiedenen Klassen voneinander trennen. Diese werden dann für die Klassifizierung verwendet. Wang et al. [2017] nutzen eine Support Vector Machine für die

Klassifizierung der Segmente nach einem Region Growing. Diese nutzt als Eingabe eine Reihe von geometrischen Merkmalen, die speziell für den Einsatzzweck der Fußgängerdetektion ausgewählt wurden. Navarro-Serment et al. [2010] nutzen zwei Support Vector Machines für die Detektion von Fußgängern in LiDAR-Daten im Anschluss an eine zuvor erfolgten Segmentierung dieser Daten. Die erste erhält dabei eine Reihe geometrischer Merkmale als Input. Die zweite den Output der ersten sowie mehrere Bewegungsmerkmale, die durch ein Trackingverfahren gewonnen werden.

Ein anderes genutztes Verfahren für die Klassifikation ist *Bag-of-Words*. Auch dieser Ansatz basiert auf einer zuvor erfolgten Merkmalsextraktion. Hierbei werden üblicherweise lokale Merkmale verwendet, im Fall von Punktwolken bzw. Punktwolkensegmenten beispielsweise für jeden einzelnen Punkt. In einem Trainingsverfahren wird ein geometrisches Wörterbuch mit Einträgen für bestimmte Ausprägungen dieser Merkmale gefüllt. Diesen Wörtern wird dabei eine Klasse zugeordnet. Die Klasse eines Punktwolkensegments ergibt sich dann aus den diesem Segment zugeordneten Wörtern. Behley et al. [2013] nutzen eine Bag-of-Words Klassifizierung für ihre hierarchischen Segmente. Als Merkmale verwenden sie hierbei sog. *Spin-Images* [Johnson & Hebert, 1999]. Diese beschreiben die lokale Oberfläche im Umkreis eines Punktes. Um Spin-Images zu erzeugen, wird zunächst der Normalenvektor aller Punkte einer Umgebung bestimmt. Anschließend wird um den gerade betrachteten Punkt basierend auf diesem Vektor ein Zylinder gelegt, der dann radial und vertikal in Volumen unterteilt wird. Die Anzahl an Nachbarpunkten innerhalb dieser Volumen wird gezählt und das Ergebnis davon lässt sich als zweidimensionales Bild darstellen, wobei sich die Koordinaten der Pixel aus den radialen und vertikalen Koordinaten der Volumen ergeben.

Asvadi et al. [2017] nutzen ein *Convolutional Neural Network* (CNN) für die Klassifizierung. Hierfür konvertieren sie die Punktwolkensegmente zunächst in Dense-depth Maps, welche dann vom neuronalen Netz klassifiziert werden. Bei Dense-depth Maps handelt es sich um eine 2D-Bildrepräsentation der Punktwolken, bei der sich die Farbwerte der einzelnen Pixel aus den dort gemessenen Entfernungs- bzw. Tiefenwerten ergeben. Maturana & Scherer [2015] nutzen ebenfalls ein CNN zur Objektdetektion in Punktwolken, die zuvor segmentiert wurden. Ihr CNN verarbeitet die Segmente in Form von 3D-Occupancy Grids. Hierbei handelt es sich um eine Voxelrepräsentation der Punktwolkensegmente, bei der zwischen belegten, freien und unbekannten Voxeln unterschieden wird. Der Vorteil bei der Verwendung dieser neuronalen Netze ist, dass keine zuvor vom Menschen explizit definierten Merkmale benötigt werden, sondern das Netz selbst die für die jeweilige Aufgabe optimalen Merkmale lernt. Erfahrungen bei der Objekterkennung und Detektion insbesondere in Bildern haben gezeigt, dass vom Menschen definierte Merkmalstypen oft keine optimalen Ergebnisse liefern. Auf die Besonderheiten bei der Verarbeitung von Punktwolken in neuronalen Netzen wird in Abschnitt 2.1 eingegangen.

Zhao et al. [2019] nutzen eine andere Art von neuronalem Netz, welches auf vom Menschen definierte und für die Segmente extrahierte Merkmale aufsetzt. Sie verwenden dabei drei Merkmale: Die Anzahl an Punkten im Segment, die Distanz zwischen Sensor und Segment und die grundsätzliche Ausrichtung des Segments. Das neuronale Netz klassifiziert die Segmente dann in die Klassen Fußgänger oder Fahrzeug. Da ihr System stationär im urbanen Umfeld eingesetzt wird und zuvor den Hintergrund und stationäre Objekte ausfiltert, reicht die Unterscheidung zwischen diesen zwei Klassen für ihren Anwendungsfall aus.

Velizhev et al. [2012] benutzen *Implicit Shape Models* (ISM) als Klassifikationsverfahren. Hierbei handelt es sich um eine Weiterentwicklung des Bag-Of-Words Ansatzes. Den Wörtern wird nicht mehr nur eine Klasse, sondern auch ein Richtungsvektor zugeordnet, der üblicherweise auf den Mittelpunkt eines Objektes dieser Klasse deutet. Das Verfahren betrachtet also nicht nur, welche Merkmale an einem Objekt einer bestimmten Klasse zu erwarten sind, sondern auch implizit wo sich diese an diesem Objekt befinden. Bei Personen sind z.B. Merkmale, die auf Beine

hindeuten, üblicherweise unten am Körper zu finden. Es handelt sich bei ISM um ein abstimmungsbasiertes Verfahren, in dem die Merkmale und die daraus abgeleiteten Wörter über die mögliche Position von Objekten abstimmen. Die Annahme dabei ist, dass es an Orten, an denen ein solches Objekt tatsächlich vorhanden ist, zu einer Häufung solcher Stimmen kommt. Üblicherweise haben die Stimmen dabei auch ein Gewicht. Velizhev et al. [2012] bestimmen dieses Gewicht basierend darauf, wie distinktiv die jeweilige Merkmalsausprägung ist. Merkmale, die besonders eindeutig auf ein Objekt einer bestimmten Klasse hindeuten, erhalten also ein größeres Gewicht als andere. Zusätzlich nutzen sie Faktoren, welche das Stimmgewicht sowohl der Wörter als auch der Objektklassen im Wörterbuch normalisieren. Als Merkmalstyp verwenden sie ebenfalls die bereits erwähnten Spin-Images. Velizhev et al. [2012] nutzen diesen abstimmungsbasierten Ansatz, um mit einer Untersegmentierung umzugehen. Wenn nach der Segmentierung mehrere Objekte in demselben Segment landen, befinden sich in diesen Segmenten auch mehrere Orte, an denen sich Stimmen häufen. In dem Segment können also korrekt mehrere separate Objekte erkannt werden, obwohl die ursprüngliche Segmentierung sie fälschlicherweise zusammengefasst hat.

Detektion von Objekten in 3D-Punktwolken ohne Segmentierung

Es gibt jedoch auch Verfahren, welche nicht auf eine zuvor erfolgte Segmentierung aufsetzen, sondern die aufgenommenen Daten als ganzes bzw. die ganze ROI verarbeiten. Hierdurch lassen sich die mit einer Segmentierung verbundenen Probleme komplett vermeiden. Es geht jedoch auch die Möglichkeit verloren, nach einer Segmentierung bereits Teile dieser Segmente vor der eigentlichen Klassifizierung durch eine Heuristik auszufiltern.

Zhou & Tuzel [2018] nutzen ein *Region Proposal Network* (RPN) für die Detektion von Autos, Fußgängern und Radfahrern. Ein RPN ist ein neuronales Netz, welches Objekte detektiert, indem es einen Rahmen bestimmt, der diese umschließt. Der eigentliche RPN-Teil des neuronalen Netzes setzt dabei auf eine CNN-Komponente auf. Zhou & Tuzel [2018] realisieren dieses CNN, indem sie ein Voxelgrid verwenden. Dabei betrachten sie zunächst jedes Voxel individuell und bestimmen für die Punkte innerhalb dieser Voxel mithilfe mehrere sog. *Voxel feature encoding layer* ein Merkmal, welches die geometrische Anordnung der Punkte innerhalb des Voxels beschreibt. Das eigentliche CNN setzt dann auf den Voxeln und deren Merkmalen auf.

Abstimmungsbasierte Verfahren, wie das bereits erwähnte ISM, können ebenfalls auch ohne vorherige Segmentierung der Daten für die Detektion von Objekten verwendet werden. Knopp et al. [2011] nutzen ein solches Verfahren, um Objekte in Punktwolken zu erkennen, welche zuvor in ein Mesh überführt wurden. Sie unterteilen dieses Mesh dann erst nach der Objektdetektion und erzeugen Segmente, welche die zuvor detektierten Objekte umfassen. Als Merkmalstyp verwenden sie *3D-SURF*, eine für die Nutzung im dreidimensionalen Raum angepasste Variante des auch in der Bildauswertung verwendeten *SURF* Merkmals. Qi et al. [2019] nutzen ein abstimmungsbasiertes Verfahren, welches auf einem neuronalen Netz aufbaut. Dabei verwenden sie das von ihnen entworfene *PointNet* [Qi et al., 2017a] bzw. *PointNet++* [Qi et al., 2017b]. Dieses neuronale Netz verarbeitet dabei die Punktwolke als ganzes und generiert die Stimmen. Diese werden anschließend ebenfalls mithilfe des neuronalen Netzes ausgewertet.

Einordnung dieser Arbeit

In dieser Arbeit wird ein abstimmungsbasiertes Verfahren ohne vorherige Segmentierung für die Detektion von Personen in 3D-Punktwolken verwendet. Dieses orientiert sich an klassischen ISM Ansätzen [Knopp et al., 2011; Velizhev et al., 2012] nutzt jedoch anders, als diese keine vom Menschen definierten Merkmale, sondern ein neuronales Netz, um die notwendigen Informationen für die Generierung von Stimmen zu gewinnen. Es hat also Ähnlichkeiten zu dem Verfahren von Qi et al. [2019]. Anders als bei diesem verwendet das Verfahren in dieser Arbeit aber nur für

Extraktion von Informationen für die Stimmgenerierung ein neuronales Netz. Welches zudem weniger komplex ist als das von Qi et al. [2019] genutzte. Der eigentliche Abstimmprozess wird hingegen mit klassischen Methoden, außerhalb eines neuronalen Netzes realisiert. Das Verfahren ist daher gewissermaßen ein hybrid aus den Verfahren von Knopp et al. [2011], Velizhev et al. [2012] und Qi et al. [2019].

2.3 Tracking von Objekten in 3D-Punktwolken

Als Tracking von Objekten wird das Verfolgen von Objekten in mehreren aufeinanderfolgenden Sensoraufnahmen bezeichnet. Ein Trackingverfahren ist dabei in der Lage, den Objekten im Zeitverlauf immer wieder dieselbe Identität zuzuordnen. Hierdurch lassen sich ggf. auch Informationen ableite. So kann beispielsweise von einem Objekt, dessen Position im Zeitverlauf verfolgt wird, ggf. auch die Geschwindigkeit und Beschleunigung bestimmt werden.

Tracking basierend auf Detektionen

Eine Gruppe von Verfahren verwendet einen Ansatz, der als „Tracking basierend auf Detektionen“ bezeichnet werden kann. Sie kombinieren die eigentliche Komponente zum Tracking von Objekten mit einer zum Detektieren von diesen. Hierbei entfällt auf den Detektor die Aufgabe, neu auftauchende Objekte zu finden und das Verschwinden von Objekten zu registrieren. Das Tracking selbst kann dabei das Gesamtverfahren unterstützen und z.B. Zeiträume überbrücken, in denen ein Objekt z.B. aufgrund von Verdeckungen vorübergehend nicht detektiert werden kann. Es erlaubt außerdem die Objektidentität im Zeitverlauf zu verfolgen und dementsprechend die oben erwähnte Ableitung von weiteren Informationen über die Objekte. Hierbei handelt es sich um Informationen, die durch das Detektionsverfahren alleine nicht verfügbar wären. Wenn ein aufwändiges Detektionsverfahren genutzt wird, kann außerdem die Laufzeitperformance des Gesamtverfahrens durch das Tracking verbessert werden, indem die eigentliche Detektion nicht für jede Sensoraufnahme durchgeführt wird, sondern nur für eine Auswahl von diesen.

Sato et al. [2010] detektieren und tracken Fußgänger in LiDAR-Daten. Für die Detektion verwenden sie einen sehr einfachen Ansatz, der Fußgänger als sich bewegende Objekte mit einer gewissen geometrischen Ausdehnung detektiert. Das Tracking erfolgt in 2D unter der Annahme, dass sich Fußgänger nur entlang der Bodenebene bewegen. Sie verwenden für das Tracking einen Kalman-Filter mit einem *Constant Velocity* Bewegungsmodell für die Fußgänger. Die Verwendung von Kalman-Filtern für das Tracking von Objekten wird auch in Abschnitt 3.3 erläutert. Es handelt sich bei ihnen um ein Verfahren, um den Zustand eines Systems, in diesem Fall das getrackte Objekt, basierend auf ggf. fehlerbehafteten Messungen von ggf. nur einigen Größen dieses Systems zu bestimmen und in zukünftigen Zeitschritten vorherzusagen. Wenn ein *Constant Velocity* Modell verwendet wird, umfasst der modellierte Systemzustand die Position und Geschwindigkeit der Objekte. Durch den Detektor gemessen wird dabei aber nur deren Position. Kalman-Filter modellieren Unsicherheiten in den Messungen in dem Vorhersagemodell durch Normalverteilungen. Für die Assoziation zwischen den vom Kalman-Filter vorhergesagten Fußgänger-Positionen und den vom Detektor gemessenen Positionen verwenden Sato et al. [2010] in ihrem Ansatz das *Global Nearest Neighbor* (GNN) Verfahren [Konstantinova et al., 2003]. Sie initialisieren einen neuen Track, wenn ein Fußgänger in 8 aufeinanderfolgenden Scans detektiert wird und dabei keinem vorhandenen Track zugeordnet werden kann. Sie entfernen Tracks, wenn ein Fußgänger in 5 aufeinanderfolgenden Scans nicht detektiert wird. Wang et al. [2017] haben ihren, bereits im vorherigen Abschnitt erwähnten, auf einer Support Vector Machine basierenden Detektor ebenfalls um ein ähnliches Tracking-Verfahren ergänzt. Sie tracken Fußgänger ebenfalls auf der 2D-Bodenebene und nutzen dafür auch einem Kalman-Filter mit GNN für die Assoziation.

Manghat & El-Sharkawy [2020] verwenden einen multimodalen Ansatz, der sowohl Bilddaten als auch LiDAR-Daten für die Detektion und das Tracking verwendet. Im Bezug auf das Tracking verwenden sie getrennte Vorhersagemodelle für die Bild- und LiDAR-Daten. In den LiDAR-Daten tracken sie die 3D-Bounding Box der Objekte. Deren Zustand wird in einem *Constant Velocity* Modell durch ihre Koordinaten auf der Bodenebene sowie ihre Ausdehnung mit den dazugehörigen Geschwindigkeiten modelliert und vorhergesagt. Hierfür wird wiederum ein Kalman-Filter verwendet. Die Assoziation zwischen vom Tracker vorhergesagten und detektierten Objektpositionen erfolgt in ihrer Methode unter Berücksichtigung der Überschneidung zwischen der vorhergesagten und der detektierten Bounding Box. Sie initialisieren Tracks für neu hinzugekommene Objekte sofort und löschen sie, nachdem diese in 3 aufeinanderfolgenden Frames nicht detektiert wurden.

Zhang et al. [2020] detektieren und Tracken Fahrzeuge in den Daten von stationären LiDAR-Sensoren. Die Detektion erfolgt bei ihnen mithilfe einer Support Vector Machine, die zuvor generierte Segmente klassifiziert. Für das Tracking verwenden sie anstatt eines einfachen Kalman-Filters einen *Unscented Kalman-Filter* [Wan & Van Der Merwe, 2000]. Hierbei handelt es sich um eine Erweiterung des Kalman-Filters, der nicht auf ein lineares Bewegungsmodell für die getrackten Objekte beschränkt ist. Für die Assoziation nutzen sie einen *Joint probabilistic data association filter* (JPDAF) [Bar-Shalom et al., 2009]. Anders als bei den bisher genannten Verfahren erfolgt die Assoziation zwischen Track und Detektion hier nicht rein anhand der nächsten Nachbarn, sondern mithilfe eines probabilistischen Ansatzes, der auch die Wahrscheinlichkeit der Vorhersagen des Trackers und die Konfidenz der Detektionen berücksichtigt. Dieses Vorgehen bei den Assoziationen reduziert falsche Zuordnungen in Situationen, in denen viele Objekte nahe beieinander sind.

Benedek [2014] nutzen, ähnlich wie mehrere der zuvor genannten Ansätze, auch einen Kalman-Filter für das Tracking von Personen und führen die Assoziationen zunächst ebenfalls basierend auf der Entfernung zwischen Track und Detektion durch. Hierfür nutzen sie die Ungarische Methode [Kuhn, 1955; Munkres, 1957]. Sie ergänzen dies jedoch um ein zweites Assoziierungsverfahren, welches zum Einsatz kommt, wenn Personen längere Zeit nicht mehr beobachtet wurden. Dieses ist in der Lage die Personen wiederzuerkennen und sie so mit einem älteren Track zu assoziieren. Es nutzt hierfür die Größe der Personen und die durch den LiDAR-Sensor gemessenen Intensitätswerte. Diese geben wieder, wie stark die an einem Punkt gemessene Oberfläche das Laserlicht reflektiert. Frossard & Urtasun [2018] haben ein Verfahren zum Detektieren und Tracking beschrieben, welches sowohl Kameras als auch LiDAR-Sensoren verwendet. Es nutzt dabei ein neuronales Netz, in einem Ende-zu-Ende Ansatz, sowohl für die Detektion als auch für das Tracking.

Modellfreie Trackingverfahren

Eine andere Gruppe von Trackingverfahren sind sog. Modellfreie Verfahren. Sie verwenden kein Erscheinungsmodell zur expliziten Objektdetektion. In vielen dieser Verfahren werden Objekte stattdessen anhand ihrer Bewegung getrackt und detektiert. Ein Vorgang der auch als DATMO *Detection and Tracking of Moving Objects* bezeichnet wird. Dewan et al. [2016] verarbeiten LiDAR-Punktwolken und verwenden RANSAC [Fischler & Bolles, 1981], um Bewegungsmodelle zunächst für den LiDAR-Sensor selbst und dann für dynamische Objekte in der aufgenommenen Umgebung zu schätzen. Anschließend wird ein Bayesscher Ansatz verwendet, um zu entscheiden, welche Punkte der Punktwolke einem dieser Bewegungsmodelle folgen. Es werden so Segmente der Punktwolke gebildet, die zu den einzelnen dynamischen Objekten gehören. Moosmann & Stiller [2013] segmentieren die Punktwolke hingegen direkt und sehen jedes Segment als Objekthypothese an. Ihr Vorgehen ist hier also ähnlich zu vielen Detektionsverfahren. Die Detektion selbst erfolgt dann aber mithilfe eines auf einem Kalman-Filter basierenden Trackings. Mit diesem wird entschieden, ob sich ein Segment bewegt, es sich bei diesem also um ein sich bewegendes Objekt handelt.

Wang et al. [2015] verarbeiten Daten eines LiDAR-Sensors mit nur einer Scanzeile zur modellfreien Detektion und Tracking von Objekten im urbanen Straßenverkehr. Ihr Ansatz verwendet einen erweiterten Kalman-Filter und einen gemeinsamen Systemzustand, der die Sensorpose, die Position und Geschwindigkeit dynamischer Objekte, Randpunkte von statischen Hintergrundobjekten und Randpunkte von dynamischen Objekten umfasst. Ihr Verfahren verwendet also eine Karte des statischen Hintergrunds, die laufend aktualisiert wird. Hierfür werden über neu auftauchende Objekte zunächst so lange Informationen gesammelt, bis entschieden werden kann, ob sie statisch oder dynamisch sind.

Eine Reihe von DATMO-Verfahren setzen auf eine Rasterrepräsentation der Umgebung, um zwischen statischem Hintergrund und ggf. individuellen dynamischen Objekten zu unterscheiden. Die Verwendung einer Rasterrepräsentation statt einer direkten Betrachtung der Punkte hilft hier dabei, den Verarbeitungsaufwand zu reduzieren. Sie ist außerdem ggf. weniger anfällig gegen ein gewisses Rauschen in den Messungen der LiDAR-Sensoren. Es können sowohl 2D- [Baig et al., 2014], 2,5D- [Asvadi et al., 2015] aber auch 3D- [Azim & Aycard, 2012] Raster verwendet werden. Azim & Aycard [2012] verwenden ein sog. 2,5D-Raster, welches für jede Zelle in einem 2D-Raster die Höhe von Objekten über dem Boden in dieser Zelle speichert. Diese wird bestimmt, indem die durchschnittliche Höhe von Punkten innerhalb dieser Zellen berechnet wird. Durch Betrachten des Rasters von mehreren verschiedenen Zeitpunkten kann zwischen Rasterzellen mit statischen Objekten und solchen mit dynamischen unterschieden werden. Für das eigentliche Tracking der dynamischen Objekte wird ein 2D Kalman-Filter mit einem *Constant Velocity* Modell verwendet.

Neben den DATMO-Verfahren gibt es auch modellfreie Trackingverfahren, die darauf basieren, dass sie zunächst einmalig mit den zu trackenden Objekten initialisiert werden und diese dann im Anschluss selbständig verfolgen. Diese einmalige Initialisierung kann dabei händisch erfolgen. In der Praxis wird hierfür aber häufig ein Detektionsverfahren verwendet. Diese Trackingverfahren unterscheiden sich also von den auf Detektionen basierenden darin, dass sie nur einmalig durch ein Detektionsverfahren initialisiert werden müssen und anschließend komplett unabhängig operieren. Ein Vertreter solcher Verfahren wurde von Du et al. [2018] vorgestellt. Ihr Ansatz verarbeitet LiDAR-Punktwolken und führt zunächst eine Bodenextraktion durch. Anschließend werden die verbleibenden Punkte mit einem modifizierten Region Growing Verfahren, welches auch die Punktdichte berücksichtigt, segmentiert. Dabei werden die bekannten Positionen von Objekten als Ursprungspunkte für das Region Growing genutzt. Für das eigentliche Tracking wird ein Partikelfilter verwendet, der die Position und Geschwindigkeit der bekannten Objekte im nächsten Zeitschritt vorhersagt und die tatsächliche Position der ihnen dann zugeordneten Segmente berücksichtigt, um diese Vorhersage zu korrigieren. Bei Partikelfiltern handelt es sich um eine Alternative zu Kalman-Filtern, die besser mit nichtlinearem Rauschen umgehen können. Sie erfordern dafür jedoch aufwendigere Berechnungen. Pang et al. [2021] tracken Objekte, deren äußeres Erscheinungsbild sich nicht verändert, in LiDAR-Punktwolken. Sie modellieren den Objektstatus als eine 3D-Bounding Box mit Länge, Höhe und Breite sowie mit einer 3D-Koordinate und Ausrichtung. Diese Bounding Box beschreibt dabei das äußere Erscheinungsbild des Objektes. Sie sagen die Position der Bounding Boxen vorher, indem sie auf ihre vorherige Position den gleitenden Durchschnitt der vorherigen Bewegungen addieren. Diese Vorhersage wird anschließend korrigiert, wobei die Punkte berücksichtigt werden, die an der vorhergesagten Position in der um einen gewissen Faktor vergrößerten Bounding Box des Objektes gefunden werden. Die Bounding Boxen selbst werden in einem nachfolgenden Schritt ebenfalls aktualisiert, wobei wiederum die Punkte in ihnen bzw. ihrem nahen Umfeld berücksichtigt werden.

Einordnung dieser Arbeit

In dieser Arbeit wird ein Verfahren zum Tracking von Personen in 3D-Punktwolken als Ergänzung zu einem Verfahren zur Personendetektion genutzt. Dieses Trackingverfahren verwendet einen Kalman-Filter und ist so ein typischer Vertreter des „Tracking basierend auf Detektionen“ Vorgehens. Im Unterschied zu vielen der hier vorgestellten Verfahren, erfolgt das Tracking aber im 3D-Raum und nicht auf einer 2D-Ebene. Auch basiert die Entscheidung darüber, wann ein getracktes Objekt, welches nicht mehr in den Detektionen auftaucht, entfernt wird auf der Kovarianzmatrix des Kalman-Filters, also darauf wie sicher die Vorhersagen dieses Filters sind.

3 Grundlagen

Dieses Kapitel gliedert sich in drei Abschnitte. Zunächst werden Grundlagen im Bezug auf das Mobile Laser Scanning erläutert, welches die Quelle für den Hauptteil der in dieser Arbeit verwendeten Daten ist. Danach werden Grundlagen im Bezug auf künstliche neuronale Netze vorgestellt und ein kurzer Überblick über Verfahren für die bildbasierte Posenschätzung von Personen gegeben. Dieser bildet die Grundlage für eine in Abschnitt 6.2 erfolgte Auswahl eines solchen Verfahrens für die Zwecke dieser Arbeit.

3.1 Mobile Laser Scanning

Als MLS bzw. *Mobile Laser Scanning* (deutsch: Mobile Laser-Abtastung) wird die Umgebungserfassung mit einem LiDAR-Sensor bezeichnet, der Teil eines mobilen bodengebundenen Sensorsystems ist. Es kann also vom ALS (*Airborne Laser Scanning*), bei dem das Sensorsystem fliegt und TLS (*Terrestrial Laser Scanning*), bei dem es stationär ist, unterschieden werden. Im Ergebnis liefert eine Datenerfassung via Laser Scanning zunächst eine Reihe von Entfernungsmessungen und Informationen darüber, in welche Richtung, ausgehend vom Sensor, diese Entfernungen gemessen wurden. Dies wird üblicherweise im ersten Schritt genutzt, um eine 3D-Punktwolke zu erzeugen, welche die Daten repräsentiert und für die weitere Verarbeitung verwendet wird. Hierbei handelt es sich um eine Ansammlung von 3D-Punkten in einem gemeinsamen Koordinatensystem. Je nach Fähigkeiten des Sensors, des Sensorsystems und der Verarbeitungsschritte zur Erzeugung der Punktwolke, können die 3D-Punkte in einer solchen neben ihren 3D-Koordinaten noch weitere Attribute haben. Beispielsweise die Intensität, mit der die gemessene Oberfläche das Laserlicht reflektiert hat. Auch kann das von der Punktwolke verwendete Koordinatensystem entweder im Bezug auf den Sensor oder dem Sensorsystem sein, aber es kann sich dabei auch um ein Weltkoordinatensystem handeln. Abbildung 3.1 zeigt als Beispiel einen Ausschnitt einer Punktwolke, die in einem urbanen Gebiet durch ein MLS-Sensorsystem aufgenommen wurde.

Im Unterschied zu einer ALS-Erfassung einer Umgebung sind bei einer MLS-Erfassung geringere Entfernungen zwischen Sensor und den aufgenommenen Oberflächen üblich. Diese betragen bei ALS oft mehrere hundert bis einige tausend Meter. Beim MLS hingegen sind es selten über 200 m und meist deutlich unter 100 m. Allerdings ist diese Entfernung beim MLS auch deutlich variabler. Beim ALS werden der Boden und Objekte auf dem Boden üblicherweise aus einer relativ konstanten Flughöhe erfasst. Die Entfernungen zwischen dem Sensor und der von ihm aufgenommenen Oberflächen sind dabei verhältnismäßig konstant. Bei der MLS-Datenerfassung kann sich hingegen, in derselben Szene mal eine Oberfläche direkt vor dem Sensor befinden und mal eine weit von diesem entfernt sein. Dies hat zur Folge, dass die Anzahl an einzelnen Entfernungsmessungen, mit denen ein Objekt erfasst wird, abhängig davon wie weit Sensor und Objekt bei der Aufnahme voneinander entfernt sind, stark variiert. TLS ist in dieser Hinsicht ähnlich wie MLS, muss jedoch nicht mit der Eigenbewegung des Sensorsystems umgehen. Es wird außerdem häufig für die Erfassung unbeweglicher Objekte (z.B. die Erfassung der Architektur eines Gebäudes) genutzt. In solchen Fällen gibt es dann überhaupt keine zu berücksichtigende Bewegung, was u.a.

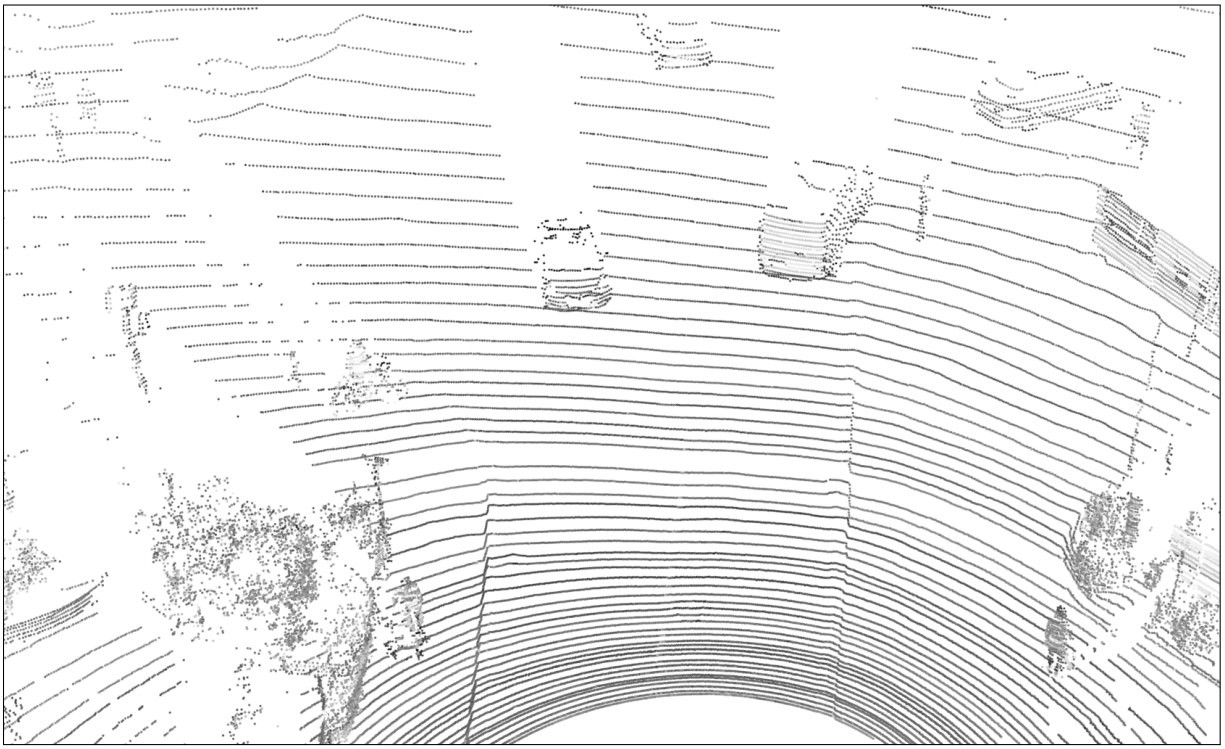


Abbildung 3.1: Ausschnitt einer Punktwolke, die von einem MLS-System im urbanen Umfeld aufgenommen wurde

die Verwendung von Sensoren mit einem langsameren Aufnahmetempo erlaubt, die dafür typischerweise genauere Messungen durchführen. Aufgrund dieser unterschiedlichen Charakteristiken unterscheiden sich also die aus MLS-, ALS- und TLS-Messungen resultierenden Daten sowie das Anforderungsprofil, welches an die verwendeten LiDAR-Sensoren gestellt wird.

3.1.1 LiDAR-Sensoren

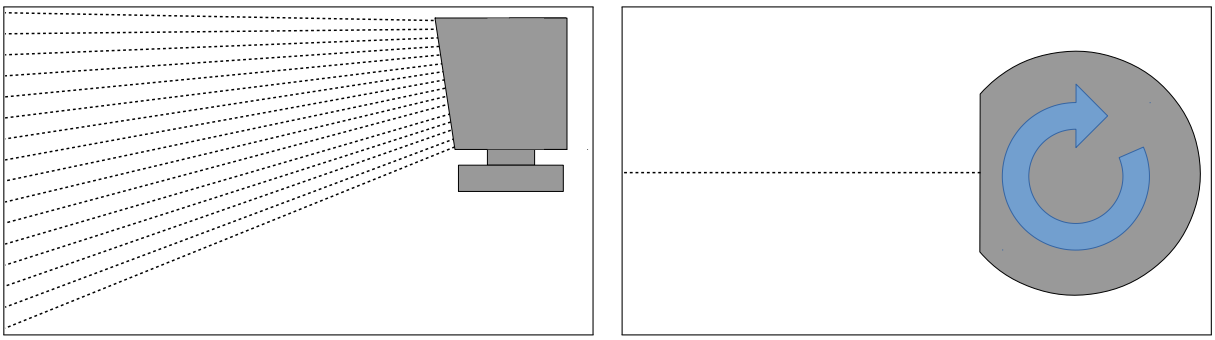
LiDAR-Sensoren verwenden Laserlicht, um die Entfernung zwischen Sensor und einer Oberfläche zu messen. Hierfür wird durch die Laserquelle des Sensors ein möglichst kleiner Fleck auf der Oberfläche beleuchtet, die dann einen Teil des Laserlichts reflektiert. Das reflektierte Licht wird von dem Detektor des LiDAR-Sensors registriert und genutzt, um die Entfernung zu bestimmen. Dabei werden zwei verschiedene Messprinzipien verwendet: phasenbasiert und laufzeitbasiert.

Bei der phasenbasierten Messung wird ein Signal mit einer verhältnismäßig großen Wellenlänge auf das Laserlicht aufmoduliert. Es wird dann die Phasenlage des ausgesendeten und des reflektierten Laserlichts verglichen. Hierdurch lässt sich unter Berücksichtigung der bekannten aufmodulierten Wellenlänge ein Hinweis auf die Entfernung bestimmen. Dabei muss berücksichtigt werden, dass bei Entfernungen, die größer sind als die Wellenlänge, so keine eindeutige Entfernungsbestimmung möglich ist. Es ist daher notwendig mehrere verschiedene Wellenlängen nacheinander zu verwenden, damit eine eindeutige Entfernung bestimmt werden kann. Die phasenbasierte Messung wird oft im TLS verwendet und zeichnet sich durch eine hohe Entfernungsauflösung und Genauigkeit aus.

Bei der laufzeitbasierten Messung wird direkt die Zeit zwischen Aussenden eines Laserpulses und dem Registrieren von dessen Reflexion gemessen. Diese Laufzeit kann dann unter Berücksichtigung der Lichtgeschwindigkeit in die Strecke umgerechnet werden, die das Laserlicht zurückgelegt

hat. Die tatsächliche Entfernung zum Punkt der Reflexion ist dann dementsprechend die Hälfte dieser Strecke. Eine Schwierigkeit hierbei ist, dass die Lichtgeschwindigkeit sehr hoch ist und daher bereits kleine Messfehler bei der Bestimmung der Laufzeit, zu signifikanten Ungenauigkeiten bei der gemessenen Entfernung führen. Aktuell werden beim MLS meist laufzeitbasierte LiDAR-Sensoren eingesetzt.

Abbildende LiDAR-Sensoren führen eine große Anzahl Entfernungsmessungen in unterschiedliche Richtungen durch. Meist wird hierfür ein scannender bzw. abtastender Ansatz genutzt, bei dem diese Messungen in kurzer Folge nacheinander durchgeführt werden. Hierfür kommen entweder Ablenkvorrichtungen (z.B. ein Spiegel) zum Einsatz, um einen einzelnen oder eine kleine Zahl von Lasern in verschiedene Richtungen abzulenken, oder der ganze Sensor bzw. der Kopf des Sensors wird bewegt. Auch Kombinationen aus beiden Ansätzen sind möglich. Es gibt jedoch auch sog. Flash-LiDAR Sensoren, bei denen statt einzelner Punkte ein größerer Bereich der Umgebung gleichzeitig mit dem Laserlicht beleuchtet wird. Bei den im Rahmen dieser Arbeit verwendeten LiDAR-Sensoren handelt es sich um welche, bei denen der Sensorkopf rotiert. Hierdurch wird ein 360°-großer horizontaler Erfassungsbereich erzielt. Der Sensorkopf selbst deckt mit mehreren Lasern, die mit verschiedenen Abstrahlwinkeln vertikal angeordnet sind, einen gewissen vertikalen Erfassungsbereich ab. Dieses im Bereich der MLS-LiDAR-Sensoren verbreitete Prinzip wird in Abbildung 3.2 dargestellt.



(a) (b)

Abbildung 3.2: Funktionsprinzip eines MLS-Laserscanners mit rotierendem Sensorkopf. a) Seitenansicht, der vertikale Erfassungsbereich ergibt sich aus dem Winkel zwischen oberstem und unterstem Laser. b) Draufsicht, der horizontale Erfassungsbereich ergibt sich aus der Rotation des Kopfes.

3.1.2 Sensorsystem

Neben den LiDAR-Sensoren gehören zu einem MLS-Sensorsystem noch weitere Komponenten. Definitionsgemäß ist ein solches Sensorsystem mobil. Dies bedeutet meist, dass es sich um Fahrzeug- oder ggf. auch roboterbasierte Systeme handelt. Es gibt aber auch Systeme, die von Personen getragen werden. Für die meisten Anwendungsfälle ist eine Referenzierung der im Zeitverlauf aufgenommenen Daten auf ein gemeinsames Koordinatensystem erforderlich. In vielen Fällen auch eine Georeferenzierung, also eine Bezugnahme auf ein globales Koordinatensystem. Für die Referenzierung ist es notwendig, dass das System seine Eigenbewegung bestimmen kann. Für die Georefenzierung muss es zusätzlich bestimmen, wo es sich auf der Welt befindet. Dies kann mit einer direkten Referenzierung oder einer datengetriebenen Referenzierung erreicht werden.

Für die direkte Referenzierung werden zusätzliche Komponenten wie z.B. ein INS (Inertiales Navigationssystem) genutzt, welches sich üblicherweise aus einem GNSS-Empfänger (Globales Navigationssatellitensystem) und einer IMU (*Inertial Measurement Unit*) zusammensetzt. Bei der

datengetriebenen Referenzierung wird hingegen z.B. auf sog. SLAM (*Simultaneous Localization and Mapping*) Verfahren zurückgegriffen. Diese nutzen die von den LiDAR-Sensoren aufgenommenen Daten, um das Sensorsystem in einer Karte zu lokalisieren, die laufend aktualisiert und erweitert wird. Auch hierbei kann zur Ergänzung eine IMU verwendet werden, welche das SLAM-Verfahren durch das Bereitstellen einer zusätzlichen Datenbasis unterstützen kann.

Ein MLS-Sensorsystem kann über mehr als einen LiDAR-Sensor sowie diverse weitere Komponenten verfügen. In diesem Fall ist es hilfreich, wenn die geometrische Anordnung der verschiedenen Komponenten zueinander bekannt ist. Dies erlaubt es, die Daten der verschiedenen Sensoren effektiver gemeinsam zu nutzen und z.B. eine Punktwolke zu erzeugen, die Daten mehrerer LiDAR-Sensoren enthält. Wenn ein Sensorsystem über mehr als einen LiDAR-Sensor bzw. über weitere Sensormodalitäten verfügt, wird es zu einem Multisensorsystem. Üblich sind z.B. Systeme, die neben LiDAR für die Erfassung der 3-dimensionalen Geometrie auch über Kameras für die Erfassung von Farbe und Oberflächenbeschaffenheit der Umgebung verfügen. So lassen sich z.B. auch 3D-Punktwolken erzeugen, bei denen jeder Punkt neben einer 3D-Koordinate auch einen RGB-Farbwert erhält.

Eine detaillierte Beschreibung des im Rahmen dieser Arbeit verwendeten Sensorsystems befindet sich in Abschnitt 7.1.

3.2 Künstliche Neuronale Netze

Bei künstlichen neuronalen Netzen handelt es sich um ein naturinspiriertes Verfahren des maschinellen Lernens. Hierbei wird versucht die Prinzipien von natürlichen neuronalen Netzen, wie sie im Gehirn bzw. dem zentralen Nervensystem von Lebewesen vorkommen, mathematisch nachzubilden. Sie verwenden künstliche Neuronen, die grob nach dem Verhalten natürlicher Neuronen modelliert sind. Diese Neuronen verarbeiten ihre Eingabe mithilfe einer sog. Aktivierungsfunktion und erzeugen so ein Signal, welches sie über Verbindungen an andere Neuronen weitergeben. Diese Verbindungen werden auch Kanten genannt und ihnen ist ein Gewicht zugeordnet.

Bei heutigen künstlichen neuronalen Netzen sind die Neuronen, wie in Abbildung 3.3 dargestellt, meist in Schichten angeordnet. Hierbei gibt es eine Eingabeschicht, deren Neuronen genutzt werden um die zu verarbeitenden Daten in das Netz zu geben. Im Hinblick auf ein neuronales Netz zur Bildauswertung könnte es z.B. für jedes Pixel der zu verarbeitenden Bilder ein Neuron in der Eingabeschicht geben. Es gibt außerdem eine Ausgabeschicht, an welcher das Ergebnis der Verarbeitung abgelesen werden kann. Für ein Netz, welches Daten klassifizieren soll, könnte es hier z.B. Neuronen für jede zu unterscheidende Klasse geben, an denen die jeweilige Konfidenz für diese Klassen abgelesen werden kann. Neben diesen beiden besonderen Schichten kann es noch beliebig viele sog. verborgene Schichten geben. Die Mächtigkeit des neuronalen Netzes ergibt sich dabei aus der Menge, Breite und Art dieser verborgenen Schichten sowie aus der Art der Verbindungen zwischen ihnen.

Es kann zwischen Feedforward-Netzen und rekurrenten Netzen unterschieden werden. Bei ersteren gibt es eine definierte Verarbeitungsrichtung. Die Ausgabeseite von Neuronen ist dementsprechend nur mit der Eingabeseite von Neuronen auf nachfolgenden Schichten des Netzes verbunden und es gibt dementsprechend keine Zyklen im neuronalen Netz. Rekurrente Netze hingegen können Kanten haben, welche in die entgegengesetzte Richtung laufen. Es können so Zyklen im neuronalen Netz entstehen. Bei den im Rahmen dieser Arbeit genutzten neuronalen Netzen handelt es sich um Feedforward-Netze.

Neuronale Netze müssen vor der Verwendung trainiert werden. In der Praxis werden während dieses Trainings üblicherweise die Gewichte der Verbindungen zwischen den Neuronen verändert.

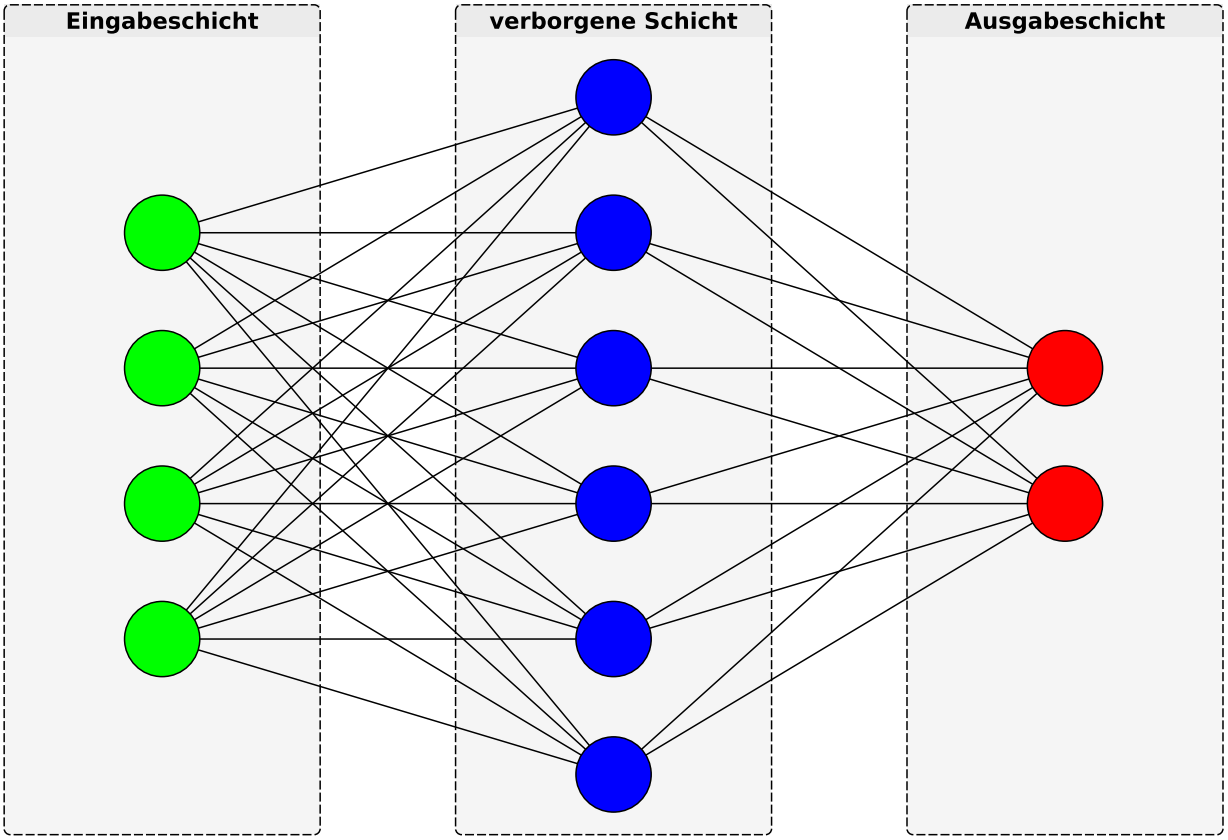


Abbildung 3.3: Beispiel eines einfachen künstlichen neuronalen Netzes mit 4 Eingabeneuronen, 2 Ausgabeneuronen und einer verborgenen Schicht mit 6 Neuronen. Das Netz ist voll vernetzt.

Hierfür werden die entsprechenden Parameter zunächst initialisiert, wofür z.B. Zufallswerte verwendet werden. Die wohl am meisten verwendete Methode für das Training ist das sog. überwachte Lernen mit der Backpropagation. Dafür sind passende Trainingsdaten erforderlich, welche sich aus der Eingabe des Netzes und der dazu passenden gewünschten Ausgabe zusammensetzen. Diese Trainingsdaten werden mit dem Netz verarbeitet. Anschließend wird die tatsächliche Ausgabe des Netzes mit der gewünschten Ausgabe mithilfe einer Verlustfunktion verglichen. Die Verlustfunktion ermittelt eine numerische Abweichung bzw. einen Fehler zwischen gewünschter und tatsächlicher Ausgabe. Mithilfe der Backpropagation wird für ein einzelnes Trainingsbeispiel der Gradient der Verlustfunktion im Hinblick auf die einzelnen Parameter des Netzes bestimmt. Dieser wird dann genutzt, um den Fehler in einem Gradientenabstieg zu minimieren, indem die Parameter angepasst werden. Das Training kann also als Optimierungsproblem gesehen werden, bei dem es darum geht, den von der Verlustfunktion ausgewiesenen Fehler des Netzes zu minimieren. Um lokale Minima zu vermeiden und möglichst effizient ein globales Minimum dieses Optimierungsproblems zu finden, wird ein Optimierer verwendet. Dessen Aufgabe ist es, die Parameter des Netzes effizient so anzupassen, dass mit möglichst wenig Training die Konfiguration der Parameter des Netzes gefunden wird, die den von der Verlustfunktion ausgewiesenen Fehler minimiert. Die Schwierigkeit dabei sind die Vielzahl an Parametern und die unbekannte Beschaffenheit des Lösungsraums. Die Optimierer verändern und beeinflussen dabei u.a. die Rate, mit der die verschiedenen Parameter angepasst werden. Diese wird auch Lernrate genannt. Für viele neuronale Netze wird ein Adam-Optimierer [Kingma & Ba, 2015] verwendet. Dieser bestimmt für jeden Parameter eine eigene Lernrate und passt diese laufend an.

Beim Training werden die Trainingsdaten wiederholt vom Netz verarbeitet und währenddessen die Parameter entsprechend angepasst. Jeder Durchlauf der Trainingsdaten wird dabei als Trainingsepoche bezeichnet. Während des Trainings sollte sich das Netz immer mehr einem optimalen Zustand annähern, bei dem die Parameter so gewählt sind, dass es die gestellte Aufgabe am besten erfüllen kann. Eine Gefahr dabei ist, dass das Netz irgendwann so weit trainiert wird, dass die Ergebnisse zwar noch für die Trainingsdaten besser werden, das Netz aber nicht mehr gut für andere Daten generalisiert. Man spricht dann von einer sog. Überanpassung (engl. *Overfitting*). Um eine solche zu vermeiden, sollte das Training rechtzeitig beendet werden. Um diesen Zeitpunkt besser zu bestimmen, werden häufig neben den Trainingsdaten auch sog. Validierungsdaten verwendet. Diese werden nicht für das eigentliche Training genutzt, sondern um zu bestimmen, ob der Fehler nur noch für die Trainingsdaten, jedoch nicht mehr für andere Daten geringer wird. Wenn dies für längere Zeit der Fall ist, liegt meist eine Überanpassung vor. Das Training sollte dementsprechend dann beendet werden.

Durch Methoden zur Regularisierung kann eine Überanpassung verhindert oder zumindest verzögert werden. Eine bei neuronalen Netzen häufig verwendete Methode zur Regularisierung ist der Einsatz von sog. Dropout-Schichten. Bei diesen wird während des Trainings zufällig ein gewisser Teil der Neuronen abgeschaltet bzw. ignoriert. Sie erzeugen dann also keine Ausgabe. Die Konfiguration des Netzes verändert sich während des Trainings also fortwährend ein wenig. Dies sorgt dafür, dass selbst beim wiederholten Verarbeiten von immer wieder denselben Trainingsdaten über mehrere Epochen, die Neuronen des Netzes jedes Mal ein wenig anders reagieren müssen, um dasselbe Ergebnis zu erzielen.

3.2.1 Aktivierungsfunktionen

In diesem Abschnitt werden einige häufig verwendete Aktivierungsfunktionen für künstliche Neuronen vorgestellt. Diese werden von den Neuronen genutzt um ihre Eingabe zu verarbeiten und eine Ausgabe zu erzeugen. Sie sind in Abbildung 3.4 auch grafisch dargestellt.

Lineare Aktivierungsfunktion

Bei der linearen Aktivierung ist die Ausgabe des Neurons proportional zu dessen Input. Sie lässt sich mathematisch wie folgt darstellen:

$$f(x) = x \tag{3.1}$$

Die lineare Aktivierungsfunktion ist für mehrschichtige neuronale Netze nicht geeignet, da mehrere hintereinander ausgeführte lineare Funktionen mathematisch auch durch eine einzige lineare Funktion ersetzt werden können. Ein mehrschichtiges Netz hat also effektiv nur noch eine Schicht.

Sigmoid

Die Sigmoidfunktion erzeugt für alle Eingabewerte eine Ausgabe im Intervall von 0 bis 1. Diese Eigenschaft macht sie zu einem guten Kandidaten, wenn ein neuronales Netz eine Wahrscheinlichkeit schätzen soll. Die Ausgabe hat dabei einem S-förmigen Verlauf mit einem großen Gradient im Bereich um $x = 0$, der sich dann zu beiden Seiten hin abflacht. Die Funktion kann mathematisch wie folgt dargestellt werden:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

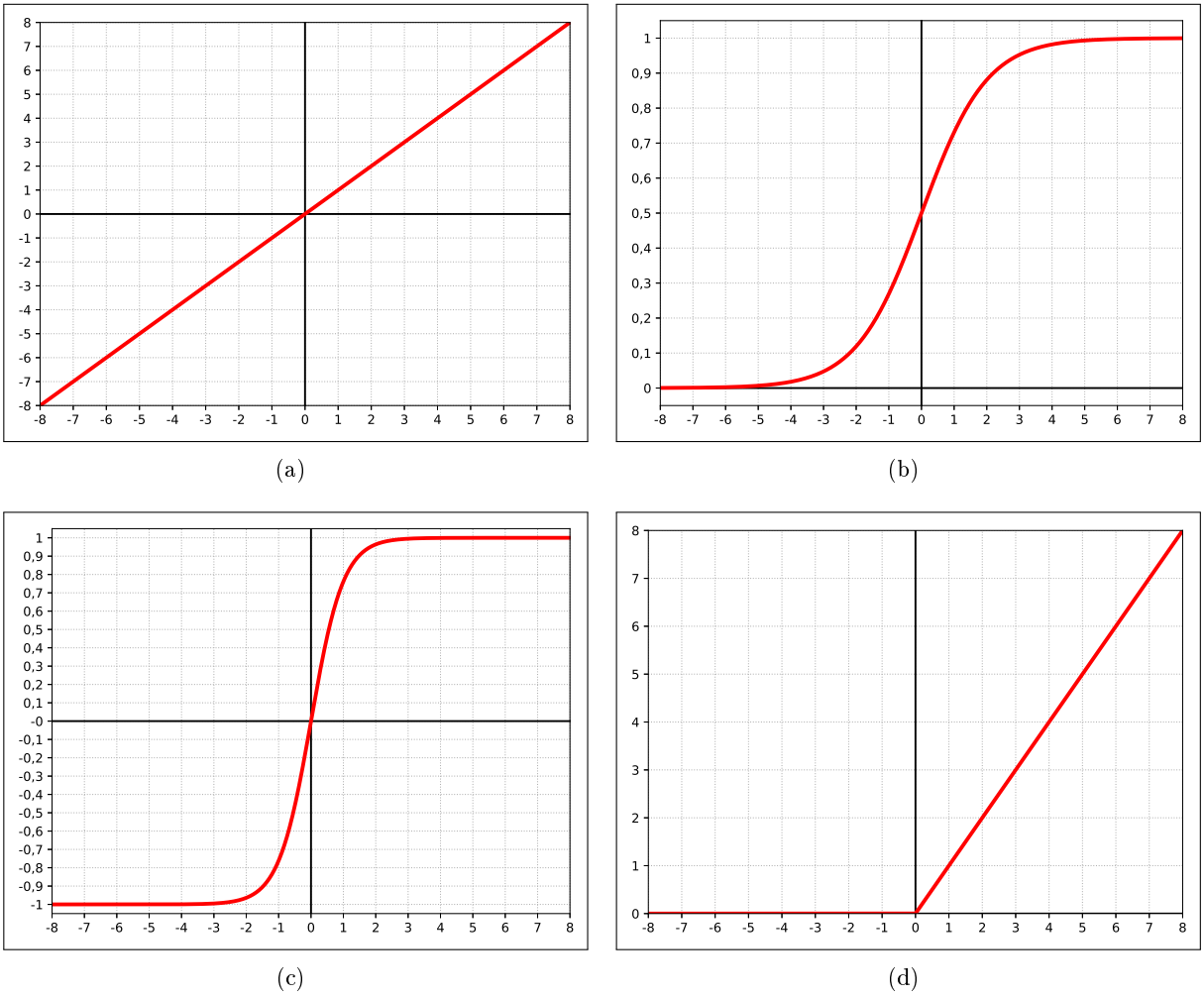


Abbildung 3.4: Verschiedene Aktivierungsfunktionen für künstliche Neuronen. a) Linear, b) Sigmoid, c) Tanh, d) ReLU

Ein Problem der Sigmoidfunktion ist der kleine Gradient bei größeren Eingabewerten. In solchen Situationen ist daher das lernen für das neuronale Netz erschwert.

Tanh

Die Tanh-Funktion hat einen ähnlichen Verlauf wie die Sigmoidfunktion, erzeugt aber Werte im Intervall von -1 bis 1. Sie kann wie folgt dargestellt werden:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.3}$$

Sie hat gegenüber der Sigmoidfunktion den Vorteil, dass negative Eingangswerte einen ggf. starken negativen Output und positive Eingangswerte einen ggf. starken positiven Output erzeugen. Um 0 herum hat sie außerdem einen größeren Gradienten. Für verborgene Schichten eines neuronalen Netzes wird sie daher üblicherweise gegenüber der Sigmoidfunktion bevorzugt.

ReLU

ReLU steht für *Rectified Linear Unit*, diese Aktivierungsfunktion wird dementsprechend auch *Rectifier* genannt. Es handelt sich dabei um eine Aktivierungsfunktion, die sich linear für positive Werte verhält aber für negative 0 ausgibt. Sie ist daher nicht linear und kann sinnvoll für mehrschichtige Netze verwendet werden. Sie lässt sich so darstellen:

$$f(x) = max(0, x) \tag{3.4}$$

ReLU ist eine einfache und effiziente Aktivierungsfunktion, die heutzutage in den versteckten Schichten vieler neuronaler Netze genutzt wird.

Softmax

Die Softmax-Aktivierungsfunktion ist eine Ergänzung zu der Sigmoid-Aktivierungsfunktion und wird meist in der Ausgabeschicht von Klassifizierungsproblemen genutzt. Wenn mehrere Klassen unterschieden werden, können die Ergebnisse der Sigmoidfunktionen, der verschiedenen Ausgabeneuronen für die verschiedenen Klassen, nicht direkt als Wahrscheinlichkeiten angesehen werden, da sie sich in der Summe nicht auf 1 aufaddieren. Die Softmax-Funktion löst diesen Umstand, indem sie eine Normierung für die Aktivierung der Ausgabeneuronen aller zu unterscheidenden Klassen durchführt. Mathematisch lässt sie sich so beschreiben:

$$\sigma(Z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.5}$$

wo $\sigma(Z)_i$ = Ergebnis der Softmax-Funktion für das i-te Element des Eingabevektors Z
z_i = i-tes Element des Eingabevektors
$\sum_{j=1}^{K} e^{z_j}$ = Normierungsterm
K = Anzahl der Klassen die unterschieden werden

3.2.2 Verlustfunktionen

Eine Verlustfunktion wird genutzt um zu ermitteln, wie groß der Fehler bzw. die Abweichung der Verarbeitung im neuronalen Netz zwischen gewünschtem und tatsächlichem Ergebnis ist. Abhängig von der Aufgabenstellung an das neuronale Netz sind verschiedene Verlustfunktionen geeignet. Aufgrund der Menge an verschiedenen Verlustfunktionen wird sich hier auf diejenigen konzentriert, die auch in dieser Arbeit verwendet werden.

Kategoriale Kreuzentropie

Die kategoriale Kreuzentropie ist eine Verlustfunktion, die für Netze genutzt werden kann, welche Daten klassifizieren. Sie betrachtet für jede Klasse, die unterschieden wird, wie sehr die vom Netz vorhergesagte Wahrscheinlichkeit für diese Klasse mit der tatsächlichen Wahrscheinlichkeit übereinstimmt. In der Praxis ist es dabei oft so, dass in den Trainingsdaten die tatsächliche Wahrscheinlichkeit für eine Klasse entweder 0 oder 1 ist, da die Daten beim Erstellen der Trainingsdaten eindeutig einer bestimmten Klasse zugeordnet wurden. Sie wird wie folgt berechnet:

$$CE = -\sum_{i=1}^{n} t_i \log(p_i) \tag{3.6}$$

wo t_i = Tatsächliche Wahrscheinlichkeit für i-te Klasse.
p_i = Vom neuronalen Netz vorhergesagte Wahrscheinlichkeit für i-te Klasse.
n = Anzahl an Klassen, die unterschieden werden.

Mittlere quadratische Abweichung

Die mittlere quadratische Abweichung ist eine häufig verwendete Verlustfunktion, wenn es um Regressionsverfahren geht. Wie der Name andeutet, wird hier der Durchschnitt der quadrierten Abweichungen zwischen vorhergesagtem und tatsächlichem Wert wie folgt berechnet:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (t_i - p_i)^2 \tag{3.7}$$

wo t_i = Tatsächlicher i-ter Wert.
p_i = Vom neuronalen Netz vorhergesagter i-ter Wert.
n = Anzahl an Werten.

3.3 Kalman-Filter zum Tracking von Objekten

Bei Kalman-Filtern [Kalman, 1960] handelt es sich um ein mathematisches Verfahren, mit dem der Zustand eines Systems basierend auf ggf. fehlerbehafteten Messungen bestimmt wird. Der Kalman-Filter schätzt dabei auch nicht direkt messbare Größen des Systemzustands, auf Basis der messbaren. So lässt sich beispielsweise die Position und Geschwindigkeit eines Objekts basierend auf einer Reihe von mit einem Rauschen versehenden Messungen, nur der Position des Objektes bestimmen. Der Filter basiert auf mehrdimensionalen Normalverteilungen. Kalman-Filter haben vielfältige Anwendungsgebiete z.B. im Bereich der Messtechnik, Signalauswertung, Satellitennavigation oder dem Tracken von Objekten.

Kalman-Filter operieren in einem Zyklus, der aus zwei Schritten besteht und der üblicherweise für diskrete Zeitpunkte wiederholt wird. Die Zeitintervalle zwischen diesen Zeitpunkten können sich z.B. aus dem Messzyklus eines Sensorsystems ergeben und müssen nicht zwangsläufig gleich lang sein. Die beiden Schritte des Zyklus sind die Prädiktion, in welcher der neue Systemzustand basierend auf dem vorherigen Systemzustand und der seitdem vergangenen Zeit vorhergesagt wird und das Update bzw. die Korrektur des Systemzustands, in dem neue Messungen in diesen integriert werden.

Ein Kalman-Filter verwendet eine Reihe von Variablen um das System und dessen Zustand zu modellieren:

- F_k = Übergangsmatrix, die den Systemzustand von Zeitpunkt t_{k-1} zu Zeitpunkt t_k propagiert.
- H_k = Beobachtungsmatrix zum Zeitpunkt t_k, bildet die beobachteten Werte auf die Werte des Systemzustands ab.
- Q_k = Kovarianzmatrix für das sog. Prozessrauschen. Dieses beschreibt Fehler die aufgrund der Modellierung oder der sich ändernden Bedingungen auftreten.

- R_k = Kovarianzmatrix für das Messrauschen.

Für das Tracking von Objekten werden sowohl *Constant Velocity* als auch *Constant Acceleration* Modelle für den Systemzustand verwendet. Bei ersteren modelliert der Zustand sowohl die Position als auch die Geschwindigkeit eines Objektes. Die Beschleunigung wird hingegen als eine unbekannte Größe angesehen, die auf das System einwirkt. Bei zweiteren wird zusätzlich die Beschleunigung des Objektes modelliert, wobei wiederum die Änderung dieser Beschleunigung zur unbekannten Größe wird. In einem *Constant Velocity* Modell für das Tracking von Objekten im 3D-Raum lässt sich der Systemzustand wie folgt beschreiben:

$$
\begin{aligned}
x_k &= \begin{pmatrix} x \\ x' \\ y \\ y' \\ z \\ z' \end{pmatrix} \\
F_k &= \begin{pmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
G_k &= \begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \\ \frac{1}{2}\Delta t^2 \\ \Delta t \\ \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix} \\
x_k &= F_k x_{k-1} + G_k a_k
\end{aligned}
\tag{3.8}
$$

wo x_k = Der Systemzustand zum Zeitpunkt t_k.
x_{k-1} = Der Systemzustand zum vorherigen Zeitpunkt t_{k-1}.
Δt = Länge des Zeitintervalls t_{k-1} bis t_k.
a_k = Unbekannte Beschleunigung im Zeitintervall von t_{k-1} bis t_k.

Da a_k unbekannt ist, sehen die Berechnungen während der Prädiktion wie folgt aus:

$$
\begin{aligned}
Q_k &= \begin{pmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 & 0 & 0 & 0 & 0 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 & 0 & 0 \\ 0 & 0 & \frac{1}{2}\Delta t^3 & \Delta t^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ 0 & 0 & 0 & 0 & \frac{1}{2}\Delta t^3 & \Delta t^2 \end{pmatrix} \sigma_a^2 \\
\hat{x}_{k|k-1} &= F_k \hat{x}_{k-1} \\
P_{k|k-1} &= F_k P_{k-1} F_k^T + Q_k
\end{aligned}
\tag{3.9}
$$

wo $\hat{x}_{k|k-1}$ = Vorhergesagter Systemzustand zum Zeitpunkt t_k (a-priori).

$\hat{x}_{k-1}$ = Der Systemzustand zum vorherigen Zeitpunkt t_{k-1}.
$P_{k|k-1}$ = Vorhergesagte Kovarianzmatrix der Fehler des Systemzustands (a-priori).
P_{k-1} = Kovarianzmatrix zum vorherigen Zeitpunkt t_{k-1}.
σ_a^2 = Varianz des Prozessrauschens.

In der Praxis wird oft mit Zeitintervallen gearbeitet, die alle die gleiche Länge haben. In solchen Fällen werden sowohl F als auch Q zu Konstanten, sie müssen folglich nur einmal bestimmt werden.

Der Zustand, also die Position und Geschwindigkeit, eines getrackten Objekts kann auch ohne neue Beobachtung für mehrere Zeitpunkte hintereinander vorhergesagt werden. Dabei wird der in der Kovarianzmatrix P ausgedrückte Fehler aufgrund des in Q ausgedrückten Prozessrauschens jedoch immer größer. Sollte eine neue Messung (also eine Detektion) für ein getracktes Objekt vorliegen, kann diese im Korrekturschritt des Kalman-Filters wie folgt berücksichtigt werden:

$$
\begin{aligned}
H_k &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \\
R_k &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \sigma_z^2 \\
\tilde{y}_k &= z_k - H_k \hat{x}_{k|k-1} \\
S_k &= H_k P_{k|k-1} H_k^T + R_k \\
K_k &= P_{k|k-1} H_k^T S_k^{-1} \\
\hat{x}_k &= \hat{x}_{k|k-1} + K_k \tilde{y}_k \\
P_k &= (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T
\end{aligned}
\tag{3.10}
$$

wo
z_k = Messung zum Zeitpunkt t_k. Beim Tracking also die gemessene Objektposition.
$\tilde{y}_k$ = Residual, also Differenz zwischen vorhergesagter und gemessener Objektposition.
S_k = Kovarianz des Residual.
K_k = Kalman-Gain zur Korrektur des Systemzustands.
$\hat{x}_k$ = Systemzustand nach Berücksichtigung der Messung z_k (a-posteriori).
P_k = Kovarianzmatrix des Zustands nach Berücksichtigung der Messung (a-posteriori).
I = Einheitsmatrix, Größe abhängig vom Systemzustand. Hier 6x6.
σ_z^2 = Varianz des Messrauschens.

3.4 Körperposenschätzung in Bildern

Die Bestimmung der Körperpose von Personen kann Informationen zum Verständnis einer Situation und zur Aktionserkennung einer Person liefern. Sie ist daher ein wichtiges Themenfeld in der Bildauswertung. Es wurden bereits eine ganze Reihe von Verfahren entwickelt, um eine bildbasierte Posenschätzung durchzuführen. Ihr Ergebnis ist üblicherweise die Position von bestimmten Körper-Schlüsselpunkten und ggf. die Verbindung zwischen diesen Schlüsselpunkten. Die Positionen liegen dabei meist nur in Bildkoordinaten vor. Beispiele für solche Schlüsselpunkte sind: die Position des Kopfes, der Gelenke oder der Extremitäten.

Die Verfahren für die Posenschätzung lassen sich im Hinblick darauf, wie viele Personen in den verarbeiteten Bilddaten abgebildet sind, in zwei Gruppen einteilen. Bei Verfahren für

die Einpersonenposenschätzung wird davon ausgegangen, dass sich nur eine einzelne Person im Bild befindet. Dies vereinfacht die Aufgabenstellung, da keine Zuordnung von erkannten Körper-Schlüsselpunkten zu einzelnen Personen erforderlich ist. Solche Verfahren funktionieren aber nicht mehr wenn die Grundannahme, dass nur eine Person zu sehen ist, verletzt wird. Beispiele für Verfahren der Einpersonenposenschätzung sind Yang & Ramanan [2013]; Dantone et al. [2013]; Ke et al. [2018].

Verfahren der Mehrpersonenposenschätzung können damit umgehen, dass mehrere Personen in den Bilddaten abgebildet sind. Bei diesen wird zwischen zwei grundsätzlichen Vorgehensmodellen unterschieden. Dem Top-Down Ansatz und dem Bottom-Up Ansatz. Bei dem Top-Down Ansatz findet zunächst eine Personendetektion statt, um Bildbereiche zu selektieren, die nur eine einzelne Person zeigen. Anschließend erfolgt in den so selektierten Bildbereichen eine separate Posenschätzung, was es erlaubt ähnliche Verfahren wie bei der Einpersonenposenschätzung zu nutzen [He et al., 2017; Fang et al., 2017]. Verfahren, die den Bottom-Up Ansatz verfolgen, detektieren hingegen direkt die einzelnen Körperteile von Personen. Diese werden dann erst in einem zweiten Schritt einzelnen Personen zugeordnet.

In dieser Arbeit wird ein Mehrpersonenposenschätzungsverfahren auf Basis des Bottom-Up Ansatzes verwendet. Hierbei handelt es sich um OpenPose [Cao et al., 2017; Cao et al., 2019]. Openpose nutzt ein *Convolutional Neural Network* (CNN), um sowohl Körperteile als auch sog. *Part Affinity Fields* (PAFs) zu bestimmen. Bei letzteren handelt es sich um 2D-Vektorfelder, die die Position und Ausrichtung der Gliedmaßen modellieren, mit denen die verschiedenen Körperteile verbunden sind. Diese werden dann verwendet um Körperteile, welche zu derselben Person gehören, miteinander zu verbinden. Hierbei wird ein sog. Greedy-Algorithmus verwendet, also in jedem Schritt jeweils die Verbindung ausgewählt, die den besten Fortschritt beim Zusammenfügen aller Körperteile verspricht. Es findet hierbei also keine globale Minimierung eines Fehlers statt.

4 Detektion von Personen in 3D-Punktwolken

In diesem Kapitel wird die entwickelte und untersuchte Methode zur Personendetektion in MLS-Punktwolken vorgestellt. Obwohl diese Methode im Rahmen dieser Arbeit nur für die Personendetektion verwendet wird, kann sie theoretisch auch für die Detektion anderer Objektklassen genutzt werden. Es wird davon ausgegangen, dass die Methode Punktwolken einzelner Scans eines LiDAR-Sensors verarbeitet. Ein Scan ist dabei üblicherweise die Abtastung der Umgebung in einer einzelnen Rotation eines LiDAR-Sensors mit rotierendem Kopf. Die Methode kann auch die Daten der Punktwolken mehrerer LiDAR-Sensoren verarbeiten, sofern diese gleichzeitig erfasst wurden und ein gemeinsames Koordinatensystem verwenden. Hierbei kann es sich um ein globales Koordinatensystem handeln, was jedoch für die reine Detektion nicht zwingend erforderlich ist. Bezüglich des Koordinatensystems der Punktwolken wird die Annahme getroffen, dass eine der Achsen parallel zur Gravitationsrichtung ausgerichtet ist, also die Höhe wiedergibt. Innerhalb dieser Arbeit wird diese als z-Achse bezeichnet. Es wird außerdem davon ausgegangen, dass die Information wo sich der Sensor zum Zeitpunkt der Datenaufnahme in Relation zu den aufgenommenen Daten befunden hat, nach er Erstellung der Punktwolken erhalten bleibt und verfügbar ist. Diese Information kann dabei entweder für eine Punktwolke als ganzes oder für jeden Punkt individuell vorliegen.

Die Methode kombiniert einen abstimmungsbasierten von ISM inspirierten [Velizhev et al., 2012; Knopp et al., 2011] Ansatz mit einem neuronalen Netz. Dieses wird genutzt, um lokale Punktnachbarschaften zu verarbeiten und den Stimmraum zu füllen. Die abschließende Auswertung dieses Stimmraums und der darin befindlichen Stimmen zur Personendetektion findet hingegen mit klassischen Methoden außerhalb des neuronalen Netzes statt. Die Methode unterscheidet sich daher von der von Qi et al. [2019] verwendeten. Anstatt nur lokale Punktnachbarschaften betrachten diese bei der Stimmgenerierung, über mehrere hierarchische Ebenen die Punktwolken als Ganzes und werten den resultierenden Stimmraum ebenfalls mit einem neuronalen Netz aus. Dies hat zwar den inhärenten Vorteil, dass ihre Methode den Kontext eines Objektes besser zu dessen Detektion verwenden kann, es erfordert aber auch ein komplexeres neuronales Netz, welches aufwendiger zu trainieren ist.

Im Folgenden wird zunächst ein Gesamtüberblick über die Methode gegeben. Anschließend werden einzelne Komponenten näher erläutert. Diese sind die Bestimmung einer Bodenebene, welche Teil der Vorverarbeitung der Daten ist, das für die Stimmgenerierung verwendete neuronale Netz und die abschließende Auswertung der Stimmen im Stimmraum.

4.1 Überblick zur Methode für die Personendetektion

Die Verarbeitungsschritte der Personendetektion werden in Abbildung 4.1 dargestellt. Diese beginnt mit einer Bestimmung der Bodenebene und dem Entfernen des Bodens. Das dabei verwendete Verfahren ist in Abschnitt 4.2 erläutert. Die Bodenextraktion ist optional, kann aber die

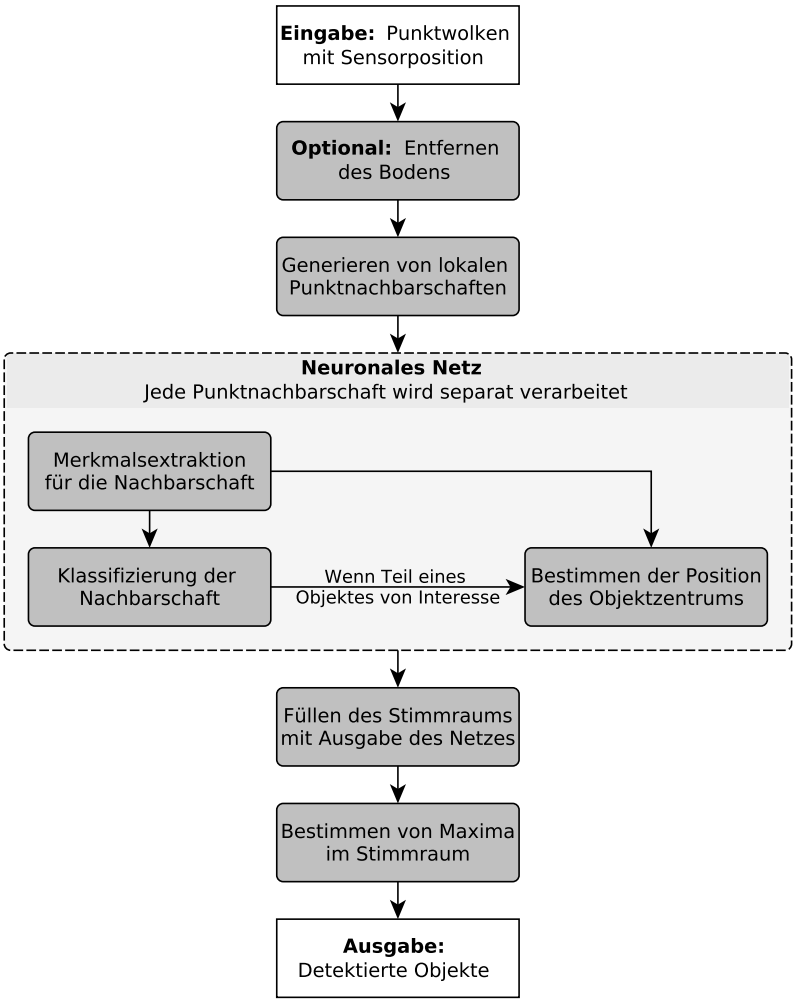


Abbildung 4.1: Verarbeitungsschritte der Methode zur Personendetektion. Das Entfernen des Bodens ist optional und kann genutzt werden, um die Verarbeitungsgeschwindigkeit zu verbessern.

Datenmenge reduzieren, die in den folgenden Schritten verarbeitet werden muss. Das Bestimmen der Bodenebene kann außerdem dazu genutzt werden, die Höhe der verbleibenden Punkte über diesem Boden zu ermitteln. Hierbei handelt es sich um eine Information, die vom neuronalen Netz genutzt werden kann, was in Abschnitt 4.3.2 erläutert wird.

Der zweite Verarbeitungsschritt ist die Erzeugung von lokalen Punktnachbarschaften. Diese bilden die Eingabe für das neuronale Netz. In dieser Arbeit wird eine lokale Punktnachbarschaft definiert als ein Ursprungspunkt und dessen Nachbarn in einem bestimmten Radius. Im Rahmen der Experimente wurden verschiedene Radien untersucht. Ein typischer Wert bei der Personendetektion ist jedoch 50 cm. Die Verwendung eines fixen Radius im Gegensatz zu den k-nächsten Nachbarn hat den Vorteil, dass dieses Vorgehen besser mit der variierenden Datendichte in MLS-Punktwolken umgehen kann. Eine Nachbarschaft aus k-nächsten Nachbarn würde in Regionen mit einer hohen Datendichte einen relativ kleinen Raumbereich abdecken und in Regionen mit einer geringen Datendichte einen großen. Dies ist im Hinblick auf die Generalisierbarkeit der Verarbeitung im neuronalen Netz nicht wünschenswert.

Die lokalen Nachbarschaften unterteilen die Daten und sollen ihre lokale Struktur beschreiben. Das neuronale Netz lernt also Objekte anhand ihres lokalen Erscheinungsbildes zu erkennen und

nicht anhand ihrer Umgebung. Dies ist eine bewusste Entscheidung in dieser Arbeit, welche es erlauben soll mit weniger komplexen Netzen zu operieren, die sich mit weniger Aufwand trainieren lassen. Wenn sich im Radius einer Nachbarschaft nur sehr wenige Punkte befinden, ist in der daraus resultierenden Nachbarschaft nicht genügend Information für das neuronale Netz vorhanden, um nützliche Verarbeitungsergebnisse zu erzielen. Solche Nachbarschaften werden daher ignoriert und nicht weiter verarbeitet. Sie treten vor allem in Regionen mit einer sehr geringen Datendichte auf, also in großer Entfernung zum Sensor. Es gibt außerdem eine Obergrenze für die Anzahl an Punkten in einer Nachbarschaft. Sollte es mehr Punkte im Radius geben, erfolgt eine zufällige Auswahl der maximal zugelassenen Anzahl. Es wird davon ausgegangen, dass eine solche Auswahl die Struktur der Nachbarschaft ausreichend gut wiedergibt.

Es ist anzunehmen, dass nahe beieinanderliegende Punkte große Teile ihrer Nachbarschaft teilen und dementsprechend als Ursprungspunkt zu ähnlichen lokalen Punktnachbarschaften führen. Zur Verbesserung der Gesamtlaufzeit kann es daher von Vorteil sein, solche Nachbarschaften nur für einen Teil der Punkte als Ursprung zu erzeugen. Dieses Sub-Sampling wird in den Experimenten untersucht.

Die erzeugten lokalen Punktnachbarschaften verfügen über ein wohldefiniertes Koordinatensystem. Das neuronale Netz muss daher nicht invariant gegenüber der Lage und Ausrichtung der Daten im zugrundeliegenden Koordinatensystem sein. Der Ursprungspunkt der Nachbarschaft bildet hierbei auch den Ursprung ihres Koordinatensystems. Die z-Achse der Nachbarschaft ist aufwärts parallel zur Gravitationsrichtung ausgerichtet, entspricht also der Richtung der z-Achse in der ursprünglichen Punktwolke. Die x-Achse steht senkrecht zur z-Achse und ist an der Linie zwischen der Position des Sensors und dem Ursprungspunkt der Nachbarschaft orientiert, wobei sie von der Sensorposition weg zeigt. Die y-Achse steht senkrecht zur x- und zur z-Achse mit der Ausrichtung, die in einem rechtshändigen Koordinatensystem resultiert.

Die lokalen Nachbarschaften werden durch das neuronale Netz verarbeitet. Dieses kann für sie bis zu zwei Ausgaben erzeugen. Die erste ist eine Klassifizierung, welche aussagt zu welcher Art von Objekt die Nachbarschaft gehört. Wenn es sich dabei um eine Objektklasse von Interesse handelt, im Rahmen dieser Arbeit also um eine Person, gibt es zusätzlich die zweite Ausgabe des neuronalen Netzes. Diese ist eine Schätzung der Position des Objektzentrums im Koordinatensystem der Nachbarschaft. Diese beiden Informationen sowie die Konfidenz des Klassifizierungsergebnisses werden genutzt, um den Stimmraum des Abstimmverfahrens zu füllen. Anschließend werden Maxima in diesem Stimmraum gesucht an denen ausreichend Stimmen für ein Objekt einer bestimmten Klasse zusammenfallen. Diese werden dann als die Position eines detektierten Objektes angesehen. Die Details der Stimmgenerierung und diese Suche von Maxima sind in Abschnitt 4.4 beschrieben. Während der Suche von Maxima und der letztendlichen Detektion von Objekten wird verfolgt, welche Punkte in einer lokalen Nachbarschaft resultieren, die zur Detektion eines Objektes beigetragen haben. Diese Punkte werden dann genutzt, um eine Bounding Box für das Objekt zu generieren. Diese Box umschließt alle Punkte, deren Nachbarschaften dessen Detektion unterstützt haben.

4.2 Bestimmung der Bodenebene und Bodenextraktion

Wie bereits beschrieben ist der Hauptzweck der Bodenbestimmung und -extraktion eine Datenreduktion zu erzielen und daraus resultierend eine Verbesserung der Laufzeit. Das eingesetzte Verfahren detektiert Personen anhand ihres lokalen Erscheinungsbildes und nicht basierend auf ihrer Umgebung. Diese bewusste Entscheidung bedeutet auch, dass der Boden wenig relevante Kontextinformationen zur Verfügung stellt. Diesen in der Personendetektion nicht weiter zu verarbeiten, sollte deren Ergebnisse daher kaum beeinflussen. In Abschnitt 4.3.2 wird ein Mecha-

-	-	-	-	2,80	1,30	2,78	1,30	1,31	1,35	1,33	1,34	4,40	4,50	4,45
2,78	2,79	2,81	2,80	1,28	1,29	1,29	1,30	1,30	1,32	1,32	4,60	1,32	4,55	-
1,26	1,25	1,27	1,26	1,27	1,28	1,29	1,30	1,29	1,30	1,31	1,30	1,31	1,32	1,33
1,07	1,08	1,08	1,08	1,08	1,09	1,09	1,10	1,09	1,10	1,11	1,11	1,12	1,12	1,13
1,08	1,08	1,09	1,09	1,09	1,09	1,10	1,11	1,10	1,10	1,11	1,12	1,12	1,12	1,13
1,09	1,10	1,09	1,10	1,10	1,11	1,11	1,12	1,11	1,12	1,12	1,13	1,13	1,14	1,14
1,08	1,09	1,09	1,09	1,08	1,10	1,09	1,10	1,10	1,11	1,11	1,11	1,12	1,13	1,13
1,08	1,08	1,08	1,07	1,07	1,09	1,08	1,09	1,09	1,10	1,10	1,11	1,11	1,12	1,12
1,09	1,08	1,08	1,09	1,08	1,09	1,09	1,08	1,09	1,10	1,11	1,11	1,20	1,21	1,22
-	1,26	1,27	1,28	1,30	1,29	1,28	1,29	2,60	-	-	3,50	3,55	-	-

(a)

-	-	-	-	2,80	1,30	2,78	1,30	1,31	1,35	1,33	1,34	4,40	4,50	4,45
2,78	2,79	2,81	2,80	1,28	1,29	1,29	1,30	1,30	1,32	1,32	4,60	1,32	4,55	-
1,26	1,25	1,27	1,26	1,27	1,28	1,29	1,30	1,29	1,30	1,31	1,30	1,31	1,32	1,33
1,07	1,08	1,08	1,08	1,08	1,09	1,09	1,10	1,09	1,10	1,11	1,11	1,12	1,12	1,13
1,08	1,08	1,09	1,09	1,09	1,09	1,10	1,11	1,10	1,10	1,11	1,12	1,12	1,12	1,13
1,09	1,10	1,09	1,10	1,10	1,11	1,11	1,12	1,11	1,12	1,12	1,13	1,13	1,14	1,14
1,08	1,09	1,09	1,09	1,08	1,10	1,09	1,10	1,10	1,11	1,11	1,11	1,12	1,13	1,13
1,08	1,08	1,08	1,07	1,07	1,09	1,08	1,09	1,09	1,10	1,10	1,11	1,11	1,12	1,12
1,09	1,08	1,08	1,09	1,08	1,09	1,09	1,08	1,09	1,10	1,11	1,11	1,20	1,21	1,22
-	1,26	1,27	1,28	1,30	1,29	1,28	1,29	2,60	-	-	3,50	3,55	-	-

(b)

-	-	-	-	-	1,30	-	1,30	1,31	1,35	1,33	1,34	-	-	-
-	-	-	-	1,28	1,29	1,29	1,30	1,30	1,32	1,32	-	1,32	-	-
1,26	1,25	1,27	1,26	1,27	1,28	1,29	1,30	1,29	1,30	1,31	1,30	1,31	1,32	1,33
1,07	1,08	1,08	1,08	1,08	1,09	1,09	1,10	1,09	1,10	1,11	1,11	1,12	1,12	1,13
1,08	1,08	1,09	1,09	1,09	1,09	1,10	1,11	1,10	1,10	1,11	1,12	1,12	1,12	1,13
1,09	1,10	1,09	1,10	1,10	1,11	1,11	1,12	1,11	1,12	1,12	1,13	1,13	1,14	1,14
1,08	1,09	1,09	1,09	1,08	1,10	1,09	1,10	1,10	1,11	1,11	1,11	1,12	1,13	1,13
1,08	1,08	1,08	1,07	1,07	1,09	1,08	1,09	1,09	1,10	1,10	1,11	1,11	1,12	1,12
1,09	1,08	1,08	1,09	1,08	1,09	1,09	1,08	1,09	1,10	1,11	1,11	1,20	1,21	1,22
-	1,26	1,27	1,28	1,30	1,29	1,28	1,29	-	-	-	-	-	-	-

(c)

Abbildung 4.2: Bestimmung der Bodenebene und Bodenextraktion. a) Initiales Befüllen des Bodenrasters: Ermittelte Höhenwerte basieren auf den z-Koordinaten der Punkte in der jeweiligen Rasterzelle. Die ursprüngliche Punktwolke wird in einer Draufsicht dargestellt. b) Validieren des Bodenrasters: Ausgehend von einer Ursprungszelle (dunkelgrau) wird überprüft welche Nachbarzellen erreichbar sind (hellgrau), ohne ein Kriterium für die maximale Steilheit des Bodens zu verletzen. Der Prozess wird dann für die erreichbaren Nachbarzellen wiederholt. c) Aus der Validierung resultierendes Bodenraster und Punktwolke ohne Bodenpunkte.

nismus erläutert, mit dessen Hilfe dem Netz zudem ein Teil der aus dem Boden resultierenden Kontextinformationen wieder zur Verfügung gestellt werden kann.

Damit aus der Bodenextraktion überhaupt ein Laufzeitgewinn resultieren kann, ist es erforderlich, dass diese Extraktion selbst entsprechend effizient ist. Ein einfacher Ansatz wäre es, eine komplett flache Bodenebene z.B. mit RANSAC [Fischler & Bolles, 1981] zu bestimmen. Die Annahme, dass der Boden komplett flach ist, ist allerdings häufig nicht zutreffend. Im urbanen Umfeld gibt es z.B. Treppen, Bordsteinkanten oder Schrägen. Es wurde daher eine Bodenextraktion entwickelt, die mit solchen Unebenheiten umgehen kann.

Die Bodenextraktion basiert auf den in den Punktwolken in Form der z-Koordinaten vorhandenen Höhenwerte. Sie erzeugt in der x-, y-Dimension ein 2D-Bodenraster, in welchem für die einzelnen Zellen die Höhe des dort vorhandenen Bodens abgelegt wird. Die Schritte zur Erzeugung dieses Rasters sind in Abbildung 4.2 dargestellt. Zunächst wird das Bodenraster initial befüllt. Hierfür werden die Punkte der Punktwolke entsprechend ihrer x-, und y-Koordinate den Rasterzellen zugeordnet. Es wird dann davon ausgegangen, dass die niedrigsten Punkte einer jeden Zelle Bodenpunkte sind. Dementsprechend ergibt sich die Bodenhöhe der Zelle aus deren z-Koordinate. Da es jedoch z.B. durch Artefakte bei der Datenerfassung zu Ausreißern bei den gemessenen Punkten kommen kann, wird nicht die niedrigste in einer Zelle vorhandene z-Koordinate als deren Höhenwert verwendet, sondern diejenige an der Grenze des 5 % Perzentils. Am Ende dieses Schritts wurde für jede Zelle des Rasters, in die Punkte gefallen sind, ein Höhenwert ermittelt. Das Ergebnis hiervon wird von Abbildung 4.2 a) dargestellt.

Es kann Zellen geben, die gar keinen Boden enthalten, beispielsweise eine Zelle in der alle erfassten Punkte Teil eines Baumwipfels sind. Im ersten Schritt der Erzeugung des Bodenrasters wurde auch für diese Zellen ein Wert für die Höhe des Bodens bestimmt. Diese falschen und eigentlich nicht zum Boden gehörenden Werte müssen daher aus dem Raster entfernt werden. Hierfür wird ein Validierungsverfahren verwendet, bei dem ausgehend von einer Startzelle versucht wird Nachbarzellen zu erreichen. Hierbei wird ein Kriterium für die maximale Steilheit des Bodens verwendet. Um eine Nachbarzelle zu erreichen, darf der Höhenunterschied des Bodens von der Ursprungszelle zur Nachbarzelle dementsprechend nicht zu groß sein, dies ist auch in Abbildung 4.2 b) dargestellt. Ausgehend von den erreichbaren Nachbarzellen wird anschließend ebenfalls wieder versucht deren noch nicht besuchte Nachbarzellen zu erreichen. Dieser Prozess nach dem Prinzip des *Region Growing* (Regionenwachstums) wird solange fortgesetzt, bis keine noch nicht besuchte Zelle mehr erreicht werden kann. Bei Zellen, die während dieses Prozesses nicht erreicht wurden, wird davon ausgegangen, dass sie keinen Boden enthalten. Sie werden dementsprechend aus dem Bodenraster entfernt. Das Ergebnis ist in Abbildung 4.2 c) dargestellt.

Der Erfolg des Validierungsprozesses hängt von der Wahl der ursprünglichen Startzelle ab, bei der die Validierung begonnen wird. Hierbei sollte es sich um eine Zelle handeln, die Boden enthält und die möglichst zentral in dem Bereich der Punktwolke liegt, der Boden umfasst. Sie wird anhand von drei Kriterien ausgewählt:

1. Der Höhenwert der Startzelle muss zwischen der Grenze des 10 % und 25 % Perzentil der Höhenwerte aller Zellen liegen. Hiermit soll zum einen erreicht werden, dass die Startzelle tatsächlich Boden enthält und zum anderen Zellen vermieden werden, deren Höhenwert aus Ausreißern unter die Bodenebene zustande gekommen sind. Es gilt also:

$$H_S > 10\,\%\text{ - Perzentil} \wedge H_S < 25\,\%\text{ - Perzentil} \tag{4.1}$$

wo H_S = Höhenwert der gewählten Startzelle S

2. Zellen mit möglichst vielen erreichbaren Nachbarn werden als Startzelle bevorzugt. Als erreichbare Nachbarn gelten die Zellen der 8 direkt angrenzenden, die erreicht werden können, ohne das erläuterte Kriterium für die maximale Steilheit des Bodens zu verletzen. Es gilt:

$$N_S = max(N_1, N_2, ..., N_n) \tag{4.2}$$

wo N_S = Anzahl erreichbarer Nachbarn der gewählten Startzelle S
N_i = Anzahl der erreichbaren Nachbarn der i-ten Zelle.

Der für N_S resultierende Wert liegt bei maximal 8, kann aber je nach Beschaffenheit der Zellen im initialen Bodenraster niedriger sein, wenn keine Zelle im Raster 8 erreichbare Nachbarn hat.

3. Die Suche nach einer Startzelle beginnt im Zentrum des Bodenrasters und arbeitet sich von dort nach außen vor. Startzelle wird die erste gefundene Zelle, die die beiden genannten Kriterien erfüllt. Dies sorgt implizit dafür, dass Zellen näher im Zentrum des Rasters Zellen weiter außen im Raster vorgezogen werden. Die Nähe zum Zentrum des Rasters ist entsprechend das dritte Auswahlkriterium für die Startzelle.

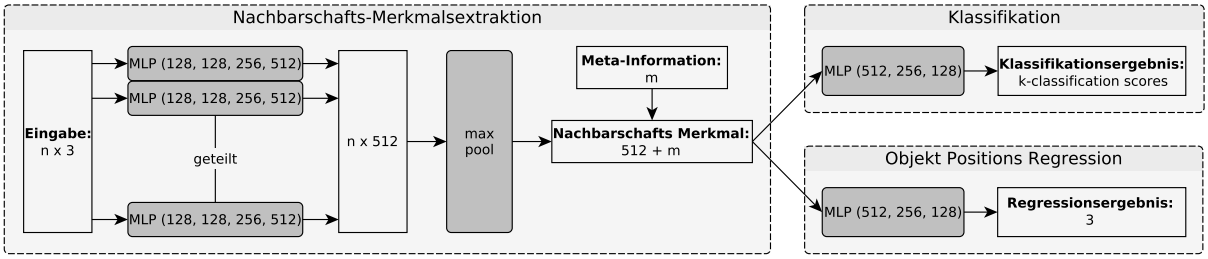
Zum Entfernen von Bodenpunkten wird über die Punktwolke iteriert. Für jeden Punkt werden die drei am nächsten liegenden Mittelpunkte von Zellen des Bodenrasters bestimmt. Mithilfe der Koordinaten dieser Rasterzellen und deren Werte für die Bodenhöhe wird dann ein Ebenenstück erzeugt. Anhand der Entfernung des gerade betrachteten Punktes zu diesem Ebenenstück wird entschieden, ob es sich bei diesem um einen Bodenpunkt handelt.

4.3 Neuronales Netz für die Personendetektion

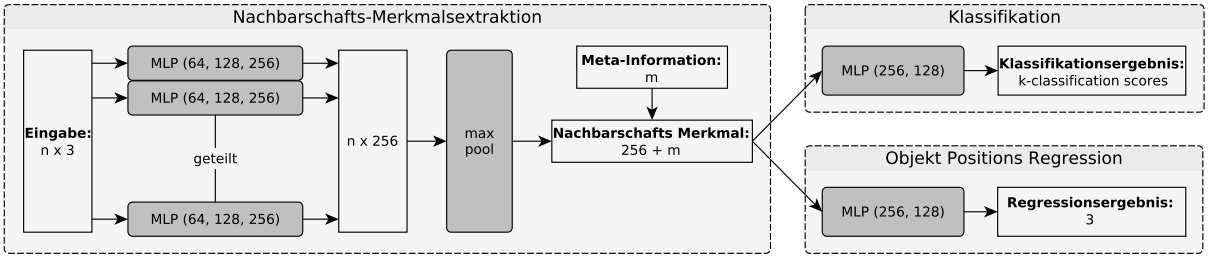
In diesem Abschnitt wird das für die Personendetektion entworfene neuronale Netz vorgestellt. Zunächst wird ein Überblick über die Struktur des Netzes gegeben. Anschließend wird erläutert wie die primäre Eingabe des neuronalen Netzes, d.h. die 3D-Koordinaten der Punkte der zuvor generierten lokalen Punktnachbarschaften, um bestimmte Metainformationen ergänzt werden können, um die Leistung des Netzes potenziell zu verbessern. Abschließend wird erläutert, wie das neuronale Netz trainiert wird.

4.3.1 Struktur des neuronalen Netzes

Die Struktur des entworfenen neuronalen Netzes ist in Abbildung 4.3 dargestellt. Es gliedert sich in drei Subnetze: Die Merkmalsextraktion, die Klassifikation der verarbeiteten Daten und die Regression der Objektposition. Das neuronale Netz ist von dem PointNet [Qi et al., 2017a] Ansatz inspiriert und verarbeitet wie dieses direkt die 3D-Koordinaten der Eingabe. Es verwendet ebenfalls eine Reihe von *Multi-layer Perceptron* (MLP) Schichten, die zwischen allen Punkten der Eingabe geteilt und später mithilfe eines *Max-poolings* zu einem gemeinsamen Merkmalsvektor zusammengefasst werden. Dieser beschreibt dabei als globales Merkmal die gesamte Eingabe. Da es sich bei dieser Eingabe um eine lokale Punktnachbarschaft handelt, wird also diese Nachbarschaft beschrieben. Dadurch dass die MLP-Schichten der Merkmalsextraktion auf alle Punkte der Eingabe gleichermaßen angewendet werden und durch das Zusammenfassen von deren individuellen Ergebnissen mit einem *Max-pooling*, ist das neuronale Netz invariant gegenüber der Reihenfolge der verarbeiteten Punkte. Dies ist erforderlich, da die verarbeiteten unstrukturierten Punktwolken und die daraus resultierenden lokalen Nachbarschaften keine feste Ordnung und Reihenfolge haben. Die lokalen Punktnachbarschaften verfügen wie beschrieben über ein wohldefiniertes Koordinatensystem. Das neuronale Netz muss daher anders als PointNet nicht invariant gegenüber



(a)



(b)

Abbildung 4.3: Die Struktur des neuronalen Netzes für die Personendetektion. Zwei Varianten: a) Zeigt die Basiskonfiguration des Netzes. b) Stellt eine vereinfachte Variante dar, die über weniger Schichten verfügt. Diese sollte weniger Trainingsdaten benötigen und kann verwendet werden, wenn diese nicht in ausreichender Menge verfügbar sind. Die primäre Eingabe sind die n 3D-Punkte einer lokalen Punktnachbarschaft. Zusätzlich werden m Werte als Metainformation für die Punktnachbarschaft verarbeitet. Die Ausgabe sind die Klassifizierungsscores für k Objektklassen und die geschätzten relativen 3D-Koordinaten des Objektzentrums. Die Zahlenangaben bei den MLP (*muli-layer perceptron*) Blöcken geben die Größe und Anzahl der MLP-Schichten an. Batch Normalization wird für alle MLP-Schichten verwendet, mit Ausnahme von denen die direkt vor einer Ausgabeschicht sind. Während des Trainings folgt auf die letzte Batch Normalization jeder Ausgabe eine Dropout-Schicht mit einer Dropout Rate von 0,2.

Unsicherheiten im Bezug auf dieses Koordinatensystem sein. Am Ende des Subnetzes für die Merkmalsextraktion kann der resultierende Merkmalsvektor noch um sog. Metainformationen ergänzt werden. Dies wird in Abschnitt 4.3.2 näher erläutert.

Das Ergebnis der Merkmalsextraktion dient als Eingabe für die beiden anderen Subnetze. Eines klassifiziert diese Eingabe. Es versucht also zu ermitteln, zu welcher Art von Objekt die verarbeitete lokale Punktnachbarschaft gehört. Aufgrund der begrenzten Größe dieser Nachbarschaften ist zu erwarten, dass die Mehrzahl von ihnen ausschließlich oder überwiegend Punkte eines einzelnen Objektes umfassen. In den meisten Fällen sollte sich eine lokale Nachbarschaft dementsprechend eindeutig zu einem bestimmten Objekt zuordnen lassen. Die Ausgabe dieses Subnetz sind die Klassifikationsscores für jede Klasse, die das neuronale Netz versucht zu unterscheiden. Es wird außerdem eine Restklasse verwendet für Nachbarschaften, die sich keiner der bekannten Klassen zuordnen lassen.

Das dritte Subnetz schätzt die Position des Objektmittelpunktes. Von diesem Netz wird für jede Objektklasse von Interesse eine eigene Instanz trainiert. Die Verarbeitung in diesem Netz findet nur für Nachbarschaften statt, die zuvor mit einem ausreichend hohen Score als Teil einer solchen Objektklasse klassifiziert wurden. Das Ergebnis dieses Subnetzes ist die 3D-Koordinate des Objektmittelpunktes im Koordinatensystem der lokalen Nachbarschaft.

Mit Ausnahme der Ausgabeschicht verwenden alle Schichten des neuronalen Netzes die Rectified Linear Unit (ReLU) Aktivierungsfunktion. In der Ausgabeschicht des Subnetzes für die

Klassifizierung wird eine SoftMax-Aktivierungsfunktion verwendet. Im Subnetz für das Schätzen der Objektposition wird in der Ausgabeschicht eine lineare Aktivierungsfunktion verwendet.

Das neuronale Netz verarbeitet die lokalen Punktnachbarschaften individuell. Abhängig von deren Ergebnis bei der Klassifikation gibt es für jede lokale Nachbarschaft daher folgende Ausgaben am Ende der Verarbeitung im neuronalen Netz:

- Für jede verwendete Objektklasse einen Score, der angibt wie wahrscheinlich es ist, dass die Nachbarschaft Teil eines Objektes dieser Klasse ist.
- Die geschätzte 3D-Koordinate im Koordinatensystem der lokalen Nachbarschaft für jede Objektklasse, in welcher der Score einen bestimmten Schwellwert überschreitet.

Diese werden anschließend genutzt um den Stimmraum des Abstimmverfahrens zu füllen, was in Abschnitt 4.4 erläutert wird.

In dieser Arbeit wurden zwei Varianten des so entworfenen Netzes untersucht, die in Abbildung 4.3 a) und b) dargestellt sind. Die Grundkonfiguration wird in a) dargestellt. Die in b) dargestellte Variante ist gegenüber dieser Konfiguration vereinfacht. Sie verfügt über weniger MLP-Schichten und hat dementsprechend weniger Parameter, die im Verlauf des Trainings gelernt werden müssen. Sie sollte daher Vorteile haben, wenn nur wenige Trainingsdaten zur Verfügung stehen.

Das neuronale Netz verwendet für die MLP-Schichten eine Batch Normalization mit Ausnahme von den Ausgabeschichten, sowie den Schichten direkt vor einer Ausgabeschicht. Den Empfehlungen von Li et al. [2019] für die Kombination von Batch Normalization und Dropout folgend wird außerdem während des Trainings für jede Ausgabe des Netzes eine einzelne Dropout-Schicht verwendet, die nach der letzten Batch Normalization-Schicht dieser Ausgabe folgt. Hierdurch soll ein Overfitting des Netzes während des Trainings vermieden werden.

4.3.2 Integration von Metainformationen

Eine Idee hinter der Verwendung von lokalen Punktnachbarschaften ist es, Objekte primär anhand ihres äußeren Erscheinungsbildes und nicht so sehr anhand ihrer Umgebung zu detektieren. Die Annahme dabei ist, dass so eine geringere Menge gelabelter Daten für das Trainieren des neuronalen Netzes erforderlich sind, da die verschiedenen Umgebungskontexte, in denen Objekte einer bestimmten Klasse vorkommen können, nicht gelernt werden müssen. Es kann jedoch gewisse Kontextinformationen geben, die leicht zu ermitteln sind und die besonders nützlich für die weitere Verarbeitung sein können. In dem verwendeten neuronalen Netz können solche Metainformation am Ende der Merkmalsextraktion dem resultierenden Merkmalsvektor hinzugefügt werden. Sie können so für die Klassifikation und die Schätzung der Objektposition verwendet werden.

In bisherigen eigenen Arbeiten [Borgmann et al., 2020] wurde die Nutzung von zwei Arten dieser Metainformationen untersucht. Zum einen die Entfernung zwischen dem Sensor und der verarbeiteten Nachbarschaft. Die Idee hierbei ist, dass diese Entfernung auch ein Ausdruck für die vorhandene lokale Datendichte darstellt. Insbesondere bei sehr niedriger Datendichte kann das Erscheinungsbild eines Objektes in einer Punktwolke anders sein, da Details nicht mehr durch diese wiedergegeben werden. Wenn das neuronale Netz also eine Information über diese Datendichte hat, kann es diese mit berücksichtigen.

Die andere untersuchte Metainformation ist die Höhe der lokalen Nachbarschaft über der Bodenebene. Für ihre Ermittlung wird auf die zuvor erläuterte Bestimmung der Bodenebene zurückgegriffen (vgl. Abschnitt 4.2). Sollte es dabei zu einer Situation kommen, in der sich eine

Punktnachbarschaft in einem Gebiet befindet, welches nicht im Bodenraster enthalten ist, wird anders als bei der Bodenextraktion der Höhenwert der Zelle verwendet, die dieser Nachbarschaft am nächsten liegt. Die Idee hinter dieser Metainformation ist, dass bestimmte lokale Merkmale vor allem in bestimmten Höhen an einem Objekt auftauchen. Bezogen auf einem Fußgänger befindet sich z.B. der Kopf üblicherweise recht weit oben, während die Beine und Füße weiter unten sind. Da in dieser Arbeit die Detektion von Personen untersucht wird, die sich üblicherweise auf dem Boden bewegen, stellt die Höhe einer Nachbarschaft über dem Boden eine gute Näherung dar um festzustellen, wo sie sich bezogen auf ein zu detektierendes Objekt befindet. Diese Information kann daher dem neuronalen Netz während der Klassifizierung und dem Schätzen der Objektposition helfen die lokale Nachbarschaft besser zu bewerten.

4.3.3 Training des neuronalen Netzes

Für das Training des neuronalen Netzes werden zuvor gelabelte Punktwolken verwendet. Die Label umfassen dabei die Klasse und Position von Objekten. Als Position wird hierbei der Mittelpunkt des Objektes genommen. Zusätzlich liegt für jedes Objekt eine Information vor, welche Punkte der Punktwolke Teil des Objektes sind. Basierend auf diesen Trainingsdaten werden lokale Punktnachbarschaften für die Objektklassen generiert, die das neuronale Netz detektieren soll. Hierbei ist zu bedenken, dass für ein einzelnes gelabeltes Objekt eine ganze Reihe von Nachbarschaften für das Training gebildet werden können. Die generierten Trainingsnachbarschaften umfassen folgende Informationen:

- Die Koordinaten der zur Nachbarschaft gehörenden Punkte im Koordinatensystem der Nachbarschaft.
- Wenn diese verwendet werden, die Metainformationen der Nachbarschaft.
- Die Klasse des Objektes, zu der die Nachbarschaft gehört.
- Die Position des Objektes im Koordinatensystem der Nachbarschaft.

Zusätzlich werden zufällige Punkte ausgewählt, die keiner der Objektklassen von Interesse zugeordnet sind, um Trainingsdaten für eine Restklasse zu bilden. Diese soll vor allem Hintergrund und Störpunkte umfassen. Für lokale Punktnachbarschaften, die dieser Restklasse zugeordnet werden, liegt keine Information für die Objektposition vor. Diese wird bei diesen aber auch nicht benötigt, da für sie nur die Subnetze zur Merkmalsextraktion und Klassifikation trainiert werden müssen. Bei der Verarbeitung der Trainingspunktwolken kann optional wie auch bei der eigentlichen Personendetektion zuvor eine Bodenextraktion erfolgen.

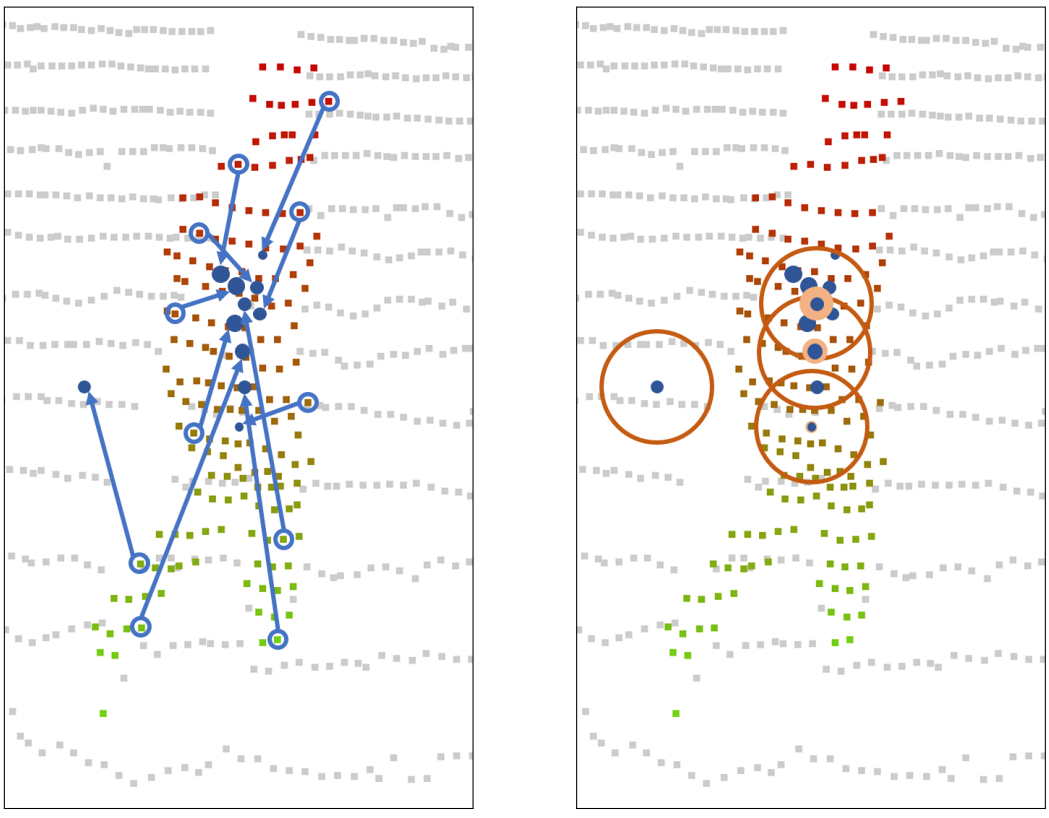
In jeder Epoche des Trainings wird für jeden Punkt eines gelabelten Objekts von Interesse in den Trainingsdaten eine lokale Nachbarschaft erzeugt. Optional kann die Menge an generierten Nachbarschaften für eine Trainingsepoche auch auf einen bestimmten Wert limitiert werden. Ist dies der Fall, erfolgt in jeder Epoche erneut eine zufällige Auswahl der verwendeten Nachbarschaften. Wie beschrieben werden zusätzlich zufällige Punkte ausgewählt, um Nachbarschaften für eine Restklasse zu bilden. Und zwar so viele, dass ihre Anzahl der Anzahl an Nachbarschaften der größten Objektklasse von Interesse entspricht. Die hierfür verwendete zufällige Auswahl wird in jeder Trainingsepoche wiederholt. So wird eine gewisse Varianz bei den in den individuellen Trainingsepochen verwendeten Daten erzielt.

Das eigentliche Training findet in mehreren Phasen statt. In der ersten Phase werden zunächst das Subnetz für die Merkmalsextraktion und das für die Klassifikation anhand der Klassifikationsergebnisse gemeinsam trainiert. Dabei wird als Verlustfunktion die kategoriale Kreuzentopie

verwendet. Anschließend werden die Gewichte für das Subnetz zur Merkmalsextraktion fixiert. Dieses wird dann genutzt, um für jede Objektklasse von Interesse einzeln die Subnetze für die Schätzung der Objektposition zu trainieren. Hierbei werden nur Trainingsdaten der jeweils gerade trainierten Objektklasse verwendet. Als Verlustfunktion wird hier die mittlere quadratische Abweichung genutzt.

Während des Trainings wird ein *Adam*-Optimierer [Kingma & Ba, 2015] mit einer initialen Lernrate von 0.0004 verwendet. Um eine Überanpassung des Netzes an die Trainingsdaten zu vermeiden, erfolgt eine Validierung des Trainingsfortschritts nach jeder Epoche, wofür separate Validierungsdaten verwendet werden. Eine Trainingsphase wird basierend darauf beendet, wenn in fünf aufeinanderfolgenden Epochen keine Verbesserung der Performanz des Netzes mit den Validierungsdaten mehr erzielt werden konnte. Es wird dann jeweils der Stand des Netzes verwendet, welcher die besten Ergebnisse in diesen Daten erzielt hat.

4.4 Abstimmverfahren



(a) (b)

Abbildung 4.4: Beispielhafte grafische Darstellung des Abstimmverfahrens. Bodenpunkte werden grau dargestellt, die Farbe der anderen Punkte ist durch ihre Höhe über dem Boden definiert. a) Erzeugen von Stimmen für Objektkandidaten basierend auf lokalen Nachbarschaften bestimmter Ursprungspunkte. Die Größe der blauen Punkte symbolisiert das Stimmgewicht der erzeugten Objektkandidaten. Aus Gründen der Übersichtlichkeit wurde dieser Vorgang hier nur für eine kleine Anzahl an Nachbarschaften dargestellt. b) Bewerten der erzeugten Objektkandidaten durch Aufschlagen eines von der Entfernung abhängigen Anteils des Stimmgewichts anderer Objektkandidaten in einem bestimmten Radius. Dieser Radius ist in dunklem Orange dargestellt, das aufgeschlagene Stimmgewicht in hellem Orange. Dieser Prozess wird hier nur für 4 Objektkandidaten dargestellt, findet aber eigentlich für alle statt.

Der erste Schritt des Abstimmverfahrens ist es, den Stimmraum mit Objektkandidaten zu füllen. Dies wird in Abbildung 4.4 a) beispielhaft dargestellt. Hierfür wird das Ergebnis der Verarbeitung der lokalen Punktnachbarschaften im neuronalen Netz genutzt, wenn die jeweilige Nachbarschaft hierbei mit einer ausreichenden Konfidenz als Teil eines Objektes von Interesse klassifiziert wurde. Bei dem Stimmraum handelt es sich um einen dreidimensionalen Raum, dessen Koordinatensystem identisch mit dem der verarbeiteten Punktwolke ist. Es wird dementsprechend direkt über die Position von Objekten in diesem Koordinatensystem abgestimmt. Ein Objektkandidat verfügt über drei Attribute:

1. Die Position des Objekts.
2. Die Klasse des Objekts.
3. Ein Stimmgewicht.

Die Position ergibt sich dabei aus der Schätzung der Objektposition des neuronalen Netzes. Diese zunächst relative Position im Koordinatensystem der lokalen Punktnachbarschaft wird in das Koordinatensystem der Punktwolke und des Stimmraums transformiert. Die beiden anderen Attribute ergeben sich aus dem Klassifizierungsergebnis des neuronalen Netzes. Die Basis des Stimmgewichts ist dabei die Konfidenz der Klassifizierung. Das Stimmgewicht wird hieraus mit folgender Formel berechnet:

$$W_c = \frac{P(c)}{n} \tag{4.3}$$

wo W_c = Stimmgewicht für Objektkandidat der Klasse c
$P(c)$ = Konfidenz des neuronalen Netzes für Klasse c
n = Anzahl an Punkten im Radius der verarbeiteten lokalen Punktnachbarschaft

Der Parameter n sorgt dabei dafür, dass das gesamte Stimmgewicht in einer Region der Punktwolke entsprechend der dort vorliegenden lokalen Punktdichte normiert wird. Hierdurch wird sichergestellt, dass in Regionen der Punktwolke mit vielen Punkten nicht insgesamt Stimmen mit einem in Summe größeren Stimmgewicht erzeugt werden, als in Regionen mit wenigen Punkten. Abbildung 4.5 ist ein reales Beispiel des Ergebnisses der Erzeugung von Objektkandidaten für die Objektklasse „Person“ in einer im urbanen Umfeld aufgenommenen Punktwolke. Zu sehen sind auch einige inkorrekte Kandidaten, die aus einem Mast mit einem daran montierten Verkehrsschild resultierten.

Der zweite Schritt des Abstimmverfahrens ist es, Maxima im Stimmraum zu bestimmen. Das Ziel hierbei ist zwischen Objektkandidaten, bei denen es sich wirklich um ein gesuchtes Objekt handelt und solchen, die aus einem fehlerhaften Auswertungsergebnis des neuronalen Netzes resultieren, zu unterscheiden. Die Annahme hierbei ist, dass tatsächliche Objekte zu einer Häufung von Stimmgewicht an ungefähr derselben Position führen sollten. Um die Unterscheidung zu treffen wird das Gewicht der erzeugten Objektkandidaten neu bewertet, wobei die Gewichte von anderen Objektkandidaten für dieselbe Objektklasse in ihrer Umgebung berücksichtigt werden. Ein Teil des Stimmgewichts dieser anderen Objektkandidaten wird dem Stimmgewicht des gerade bewerteten Objektkandidaten aufgeschlagen. Abbildung 4.4 b) stellt dies beispielhaft dar.

Wie viel des Stimmgewichts aufgeschlagen wird, ergibt sich aus der Normalverteilung. Dies folgt der Annahme, dass sich korrekte Objektkandidaten entsprechend dieser um die tatsächliche Position eines Objektes verteilen sollten. Um das letztliche Stimmgewicht eines Kandidaten zu ermitteln wird daher folgende Formel verwendet:

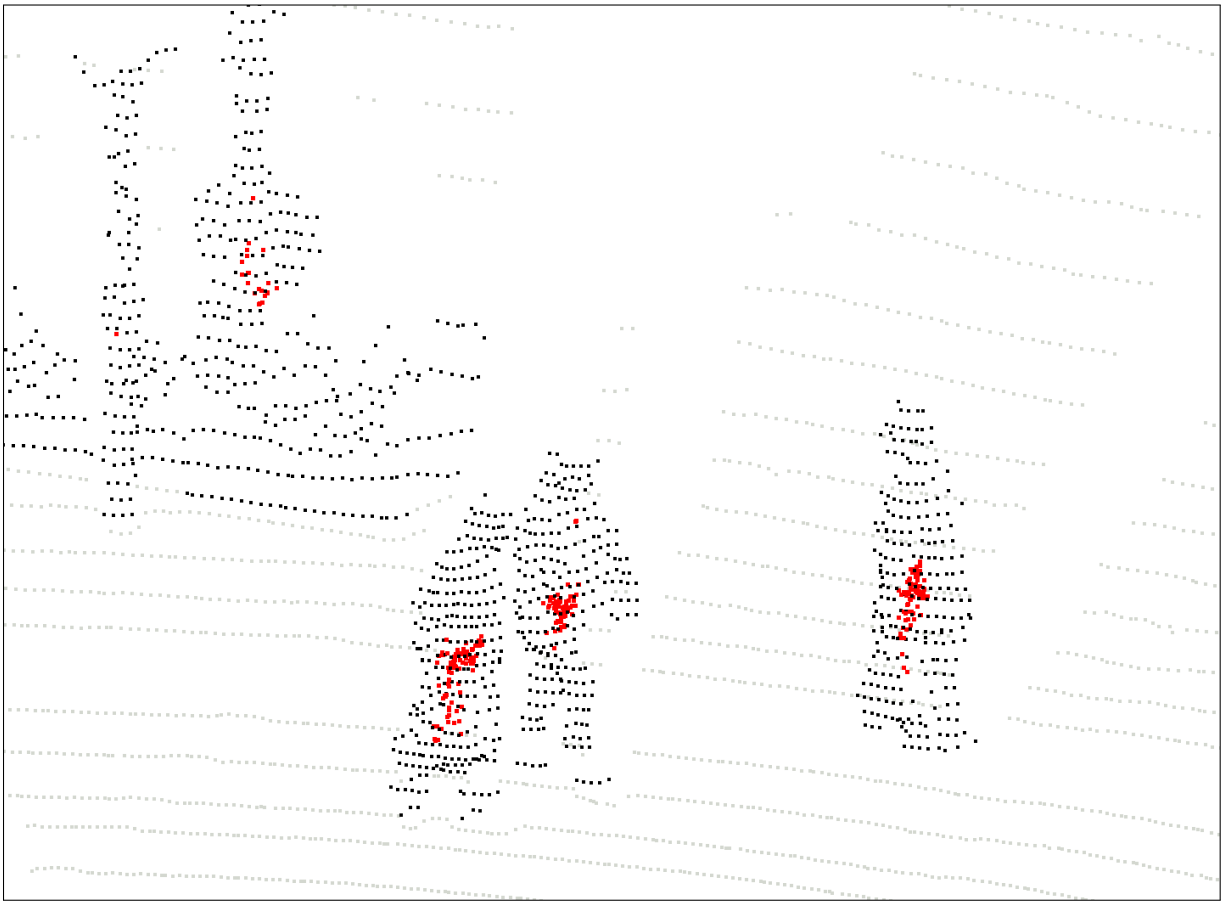


Abbildung 4.5: Beispiel für Ergebnis der Erzeugung von Objektkandidaten für die Klasse „Person“. Bodenpunkte sind in Grau dargestellt, sonstige Punkte in Schwarz und erzeugte Objektkandidaten in Rot. Neben einer ganzen Reihe von nahe an der tatsächlichen Position einer Person befindlichen Kandidaten gibt es auch einige, die aus einem Mast mit einem daran montierten Verkehrsschild resultierten. Diese gilt es im weiteren Verlauf des Abstimmverfahrens zu entfernen.

$$R_c = W_c + \sum_{k \in K} W_k \cdot e^{-\frac{{D_{ck}}^2}{2\sigma^2}} \tag{4.4}$$

wo
R_c = Bewertetes Stimmgewicht des Objektkandidaten c
W_c = Originales Stimmgewicht des Objektkandidaten c
K = Objektkandidaten mit derselben Klasse wie c im Radius 2σ
W_k = Originales Stimmgewicht des Objektkandidat k
D_{ck} = Euklidische Distanz zwischen Kandidaten c und k
σ bestimmt die Breite der genutzten Normalverteilung

Die Beschränkung von K auf Objektkandidaten bis zu einer Entfernung von 2σ begrenzt hierbei die Menge an zu berücksichtigenden anderen Objektkandidaten und hat nur wenig Einfluss auf das Ergebnis, da jenseits von 2σ der Anteil des aufgeschlagenen Stimmgewichts nur sehr klein wäre. Die Größe von σ ist ein wichtiger Parameter für das Verfahren. Dessen Größe wird in den Experimenten näher untersucht. Hierbei ist zu berücksichtigen, dass dieser auch Auswirkungen auf die Erzeugung von Bounding Boxen für die detektierten Objekte hat. Diese umfassen die Ursprungspunkte aller Nachbarschaften, welche die Detektion eines Objektes unterstützt haben,

die also zu einen Objektkandidaten geführt haben, der letztlich als tatsächliches Objekt betrachtet wurde oder der Stimmgewicht zu einem solchen Kandidaten beigetragen hat.

Nach der Neubewertung der Objektkandidaten wird ein Schwellwert angewendet und alle Kandidaten entfernt, deren resultierendes Gewicht unter diesem Schwellwert liegt. Da für ein tatsächliches Objekt eine ganze Reihe ähnlicher Kandidaten nahe der tatsächlichen Objektposition erzeugt werden, bleiben nach der Anwendung des Schwellwerts oft mehrere von ihnen für dasselbe Objekt übrig, die nahe beieinander liegen. Diese Cluster an verbleibenden Kandidaten werden daher zusammengefasst und als ein detektiertes Objekt angesehen. Hierbei wird die Position des Kandidaten mit dem höchsten Stimmgewicht als Objektposition genutzt. Dieses detektierte Objekt erhält außerdem eine Bounding Box, die wie oben beschrieben ermittelt wird und einen Detektions-Score, bei dem es sich um das Stimmgewicht des auch für die Position genutzten Objektkandidaten handelt.

5 Tracking von Personen in 3D-Punktwolken

In diesem Kapitel wird die untersuchte Methode zum Tracking von Personen vorgestellt. Ähnlich wie bei der Detektion von Personen wird auch diese im Rahmen dieser Arbeit ausschließlich für Personen verwendet, kann aber auch für andere Arten von Objekten genutzt werden. Das Tracking erfolgt im 3D-Raum und berücksichtigt die 3D-Koordinaten der Personen. Es ist also nicht wie in einigen anderen in Abschnitt 2.3 vorgestellten Ansätzen auf eine 2D-Ebene beschränkt. Dies berücksichtigt den Umstand, dass sich Personen im urbanen Umfeld zwar meist auf der Bodenebene bewegen, es aber trotzdem vorkommen kann, dass dies z.B. bei Fußgängerbrücken nicht der Fall ist.

Das Tracking wird als Teil des Gesamtsystems gesehen und operiert in Ergänzung zu dem Detektionsverfahren. Es soll dabei u.a. sicherstellen, dass auch eine Position für Personen bestimmt werden kann, die z.B. durch Verdeckungen vorübergehend nicht detektiert werden können. Was auch ein Fokus der experimentellen Untersuchung ist. Aufgrund dieser vorgesehenen Kombination wurde ein „Tracking basierend auf Detektionen“ Ansatz gewählt.

Im Folgenden wird zunächst ein Gesamtüberblick über das verwendete Tracking gegeben. Anschließend wird das Vorgehen bei der Assoziation der vorhergesagten Positionen von getrackten Personen mit den Positionen der detektierten Personen und das damit verbundene Trackmanagement erläutert.

5.1 Überblick zur Methode des Trackings

Abbildung 5.1 gibt einen Überblick über die zum Tracking gehörenden Verarbeitungsschritte und die Integration des Trackings mit der Objektdetektion. Der Trackingzyklus wird für jede neue Punktwolke im Anschluss an das Detektionsverfahren durchgeführt. Für das eigentliche Tracking wird ein Kalman-Filter mit *Constant Velocity* Modell verwendet, wie er in Abschnitt 3.3 vorgestellt wurde. Es wird angenommen, dass die Punktwolken entweder in gleichbleibenden Zeitschritten aufgenommen werden oder über einen Zeitstempel verfügen. Im ersten Fall ist Δt eine Konstante, im zweiten Fall ergibt es sich aus der Differenz der Zeitstempel von der vorherigen und der aktuellen Punktwolke.

Die Wahl für ein *Constant Velocity* anstatt eines *Constant Acceleration* Modells ist in dieser Arbeit gefallen, weil davon ausgegangen wird, dass Fußgänger üblicherweise keine langen Beschleunigungs- und Abbremsphasen haben. Diese also oft innerhalb der Zeitperiode von nur wenigen oder gar einer einzelnen Punktwolke stattfinden. Da sich die Beschleunigung daher nicht mit ausreichender Sicherheit bestimmen lässt, wird sie stattdessen als unbekannte Größe angesehen und von dem gewählten Kalman-Filter nicht als Teil des Systemzustands der getrackten Objekte modelliert.

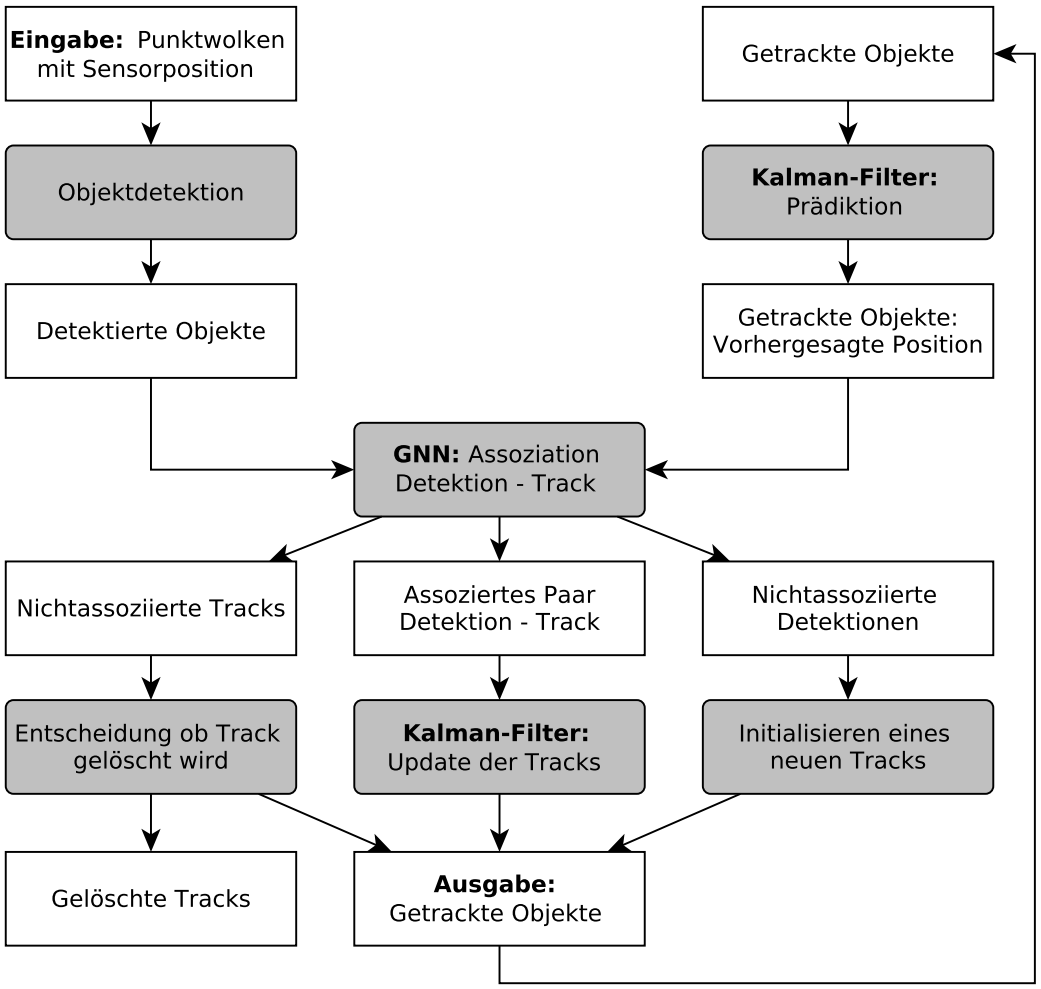


Abbildung 5.1: Verarbeitungsschritte des Trackings

Neben den Punktwolken und den darin detektierten Objekten wird in jedem Zyklus des Trackings auch der Bestand an bereits aus vorherigen Zeitschritten bekannten und aktuell getrackten Objekten als Eingabe angesehen. Für jedes dieser Objekte wird der Zustand separat von einem Kalman-Filter modelliert. Zusätzlich liegt die Information vor, um welche Art von Objekt es sich dabei handelt. Im ersten Schritt des Trackingzyklus wird die neue Position dieser Objekte, basierend auf ihrem bekannten Systemzustand, im Prädiktionsschritt des Kalman-Filters vorhergesagt.

Die getrackten Objekte mit ihrer vorhergesagten Position sowie die detektierten Objekte mit ihrer detektierten Position werden dann miteinander assoziert. Daraus können drei Arten von Objekten resultieren.

- Getrackte Objekte, denen kein detektiertes Objekt zugeordnet werden konnte: Diese werden ggf. aus dem Bestand an getrackten Objekten entfernt.

- Getrackte Objekte, denen ein detektiertes Objekt zugeordnet werden konnte: Deren Position wird im Update-Schritt des Kalman-Filters mit der Position der Detektion aktualisiert.

- Detektierte Objekte, die keinem getrackten Objekt zugeordnet werden konnten: Für diese wird ein neuer Track initialisiert, wodurch sie dem Bestand an getrackten Objekten hinzugefügt werden.

Am Ende des Trackingzyklus steht ein aktualisierter Bestand an getrackten Objekten. Diese werden ausgegeben und bilden gleichzeitig die Eingabe für den nächsten Zyklus.

5.2 Assoziation der Vorhersage mit den Detektionen und Trackmanagement

Ziel bei der Assoziation der detektierten Objekte zu den bereits bekannten getrackten Objekten ist es, jedes detektierte Objekt genau einem getrackten Objekt und jedes getrackte Objekt genau einem detektierten Objekt zuzuordnen. Dabei ist zu beachten, dass es vorkommen kann, dass für ein detektiertes Objekt keine Zuordnung erfolgt. In einem idealen Szenario, wenn bei der Assoziation selbst keinerlei Fehler passieren, ist dies der Fall, wenn ein bisher noch nie erfasstes Objekt erstmals detektiert wird, es also dementsprechend noch keinen Track für dieses Objekt geben kann. Es kann auch vorkommen, dass einem getrackten Objekt kein detektiertes Objekt zugeordnet werden kann. Dies sollte idealerweise nur dann passieren, wenn dieses den Erfassungsbereich des Sensorsystems entweder dauerhaft oder zeitweise verlassen hat und daher nicht mehr detektiert wird.

Im Rahmen dieser Arbeit wird die Position der Objekte im 3D-Raum als alleiniges Kriterium für die Zuordnung genutzt. Wenn ein getracktes Objekt nahe bei einem detektierten Objekt derselben Objektklasse ist, soll also eine solche Zuordnung erfolgen. Eine Zuordnung anhand eines Erscheinungsmodells der Objekte kann im Hinblick auf Personen in LiDAR-Daten schwierig sein, da z.B. die Farbe der Kleidung nicht ohne Weiteres als Kriterium genutzt werden kann und sich die reine Geometrie von Personen laufend verändert, wenn sich diese bewegen. Während der Zuordnung wird ein Schwellwert für die maximale Distanz zwischen dem detektierten und dem getrackten Objekt verwendet. Dieses berücksichtigt die erwartbare Bewegungsgeschwindigkeit von Personen und schließt Zuordnungen aus, die basierend darauf nicht realistisch sind.

Für die eigentliche Assoziation wird das von Konstantinova et al. [2003] vorgestellte *Global nearest neighbor* Verfahren verwendet. In diesem werden Cluster für die getrackten und detektierten Objekte gebildet. Dabei bildet zunächst jedes getrackte Objekt ein eigenes Cluster. Diesem Cluster werden dann alle detektierten Objekte derselben Objektklasse hinzugefügt, die sich in dem durch die maximale Zuordnungsdistanz definierten Radius befinden. Wenn ein detektiertes Objekt hierbei dem Cluster von zwei oder mehr getrackten Objekten hinzugefügt wird, werden diese und ihre Cluster zu einem gemeinsamen großen Cluster zusammengeführt. Im Ergebnis kann es dann vier Situationen geben.

- Ein Cluster, dass nur ein einzelnes getracktes Objekt und ein oder mehrere detektierte Objekte umfasst: Bei diesem erfolgt die Zuordnung des getrackten Objekts zu dem detektierten Objekt, zu dem die Distanz am geringsten ist. Die anderen detektierten Objekte des Clusters gelten als nicht zugeordnet.

- Ein Cluster, welches mehrere getrackte Objekte und mindestens ein detektiertes Objekt umfasst: Die Zuordnung innerhalb des Clusters erfolgt mit der ungarischen Methode [Kuhn, 1955; Munkres, 1957]. Dabei können am Ende detektierte oder getrackte Objekte verbleiben, für die keine Zuordnung erfolgt ist. Diese gelten als nicht zugeordnet.

- Ein Cluster ohne detektierte Objekte: Das getrackte Objekt in einem solchen Cluster gilt als nicht zugeordnet.
- Ein detektiertes Objekt, welches zu keinem Cluster gehört: Dieses gilt als nicht zugeordnet.

Nach dem Assoziationsprozes kann es sowohl getrackte als auch detektierte Objekte ohne Zuordnung geben. Für die detektierten Objekte wird in solchen Fällen ein neuer Kalman-Filter initialisiert und sie werden den getrackten Objekten als neue, bisher unbekannte Objekte hinzugefügt.

Bei den getrackten Objekten ohne Zuordnung ist eine Entscheidung notwendig, ob diese gelöscht werden oder weiter im Bestand der getrackten Objekte verbleiben, unter der Annahme, dass sie durch Verdeckungen oder einem Fehler im Detektionsverfahren nur vorübergehend nicht detektiert werden. Als Basis für diese Entscheidung wird die Varianz der durch den Kalman-Filter vorhergesagten Objektposition verwendet. Diese kann an der Diagonalen, der Kovarianzmatrix P des Systemzustands (vgl. Abschnitt 3.3) abgelesen werden. Diese Varianz sinkt, wenn ein Objekt im Zeitverlauf wiederholt in der Nähe der vorhergesagten Position detektiert wird, die Vorhersagen des Trackers sich also als korrekt erweisen. Sie steigt, wenn das Objekt nicht mehr detektiert wird oder der Fehler der Vorhersagen größer wird. Die Idee ist also Objekte, bei denen die Vorhersagen des Trackers vermutlich zuverlässiger sind, längere Zeit ohne Detektion weiter zu tracken und ihre Position wiederholt vorherzusagen. Im Umkehrschluss werden Objekte, bei denen die Vorhersagen des Trackers ein großes Potenzial für Fehler aufweisen schneller gelöscht, wenn sie nicht mehr detektiert werden.

6 Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern

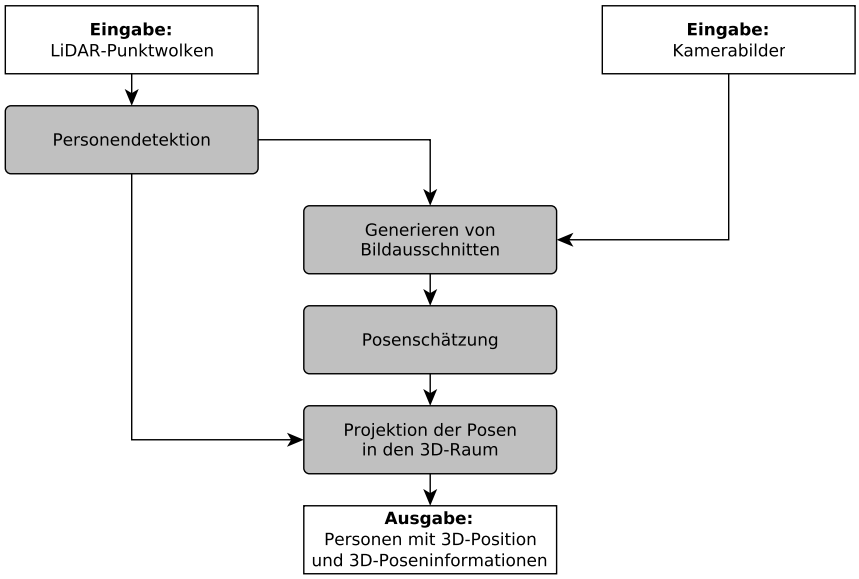
Dieses Kapitel beschäftigt sich mit der gemeinsamen Nutzung mehrerer Sensormodalitäten, um die Position und Körperpose von Personen im 3D-Raum zu bestimmen. Es werden Kamerabilder für die eigentliche Posenschätzung und LiDAR-Sensoren zum Bereitstellen der 3D-Information genutzt. Dabei können ggf. mehrere Kameras verwendet werden, um den damit abgedeckten Erfassungsbereich zu vergrößern. Es ist also eine Form der Datenfusion sowohl innerhalb als auch zwischen den Sensormodalitäten erforderlich. Hierbei wird angenommen, dass die geometrische Anordnung der Sensoren am Sensorsystem sowie die intrinsischen Kalibrierparameter der Kameras bekannt sind und dass die Sensoren eine gemeinsame Zeitbasis verwenden. Ihre Aufnahmen können also in einem zeitlichen Zusammenhang zueinander gebracht werden. Es wird außerdem angenommen, dass die Aufnahmen mit den beiden Sensormodalitäten in etwa zeitgleich erfolgen.

6.1 Varianten zur multimodalen Körperposenschätzung und Detektion von Personen

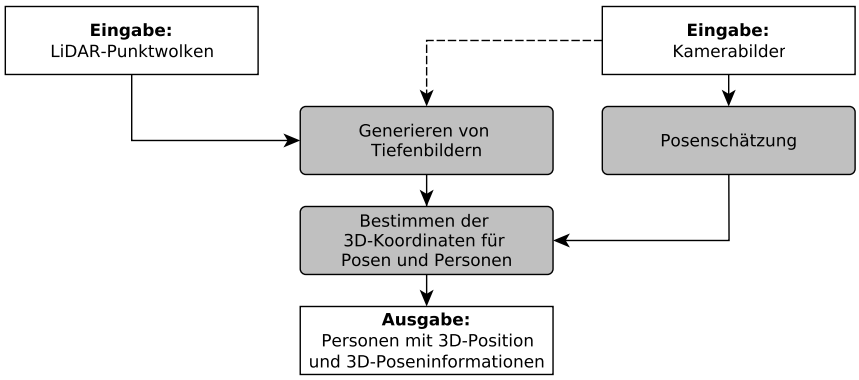
In dieser Arbeit werden drei unterschiedliche Varianten einer Methode entworfen und untersucht, mit der Personen im 3D-Raum detektiert und Informationen über deren Körperpose bestimmt werden. Diese unterscheiden sich darin, welche Aufgaben durch welche Sensormodalität erfüllt werden und ob die verschiedenen Modalitäten dafür genutzt werden, sich gegenseitig zu verifizieren. Die drei Varianten sind in Abbildung 6.1 schematisch dargestellt.

In der ersten Variante, dargestellt von Abbildung 6.1 a), wird die in Kapitel 4 erläuterte Personendetektion verwendet, um die Personen zunächst in den LiDAR-Daten zu detektieren. Anschließend werden basierend auf den Detektionsergebnissen Bildausschnitte generiert, die jeweils möglichst nur eine detektierte Person zeigen. Diese Bildausschnitte bilden dann die Eingabe für die bildbasierte Körperposenschätzung. Deren Ergebnisse liegen zunächst in 2D-Bildkoordinaten vor, welche dann als Strahl in den 3D-Raum übertragen werden. Diese Strahlen werden zusammen mit den 3D-Punkten der Personen genutzt, um 3D-Koordinaten für die Posenschlüsselpunkte zu bestimmen. Im Ergebnis dieser Variante der Methode sind alle Personen mit ihrer 3D-Position enthalten, die in den Punktwolken detektiert wurden. Personen, für die zusätzlich eine Posenschätzung durchgeführt werden konnte, umfassen außerdem die dabei bestimmten Posenschlüsselpunkte.

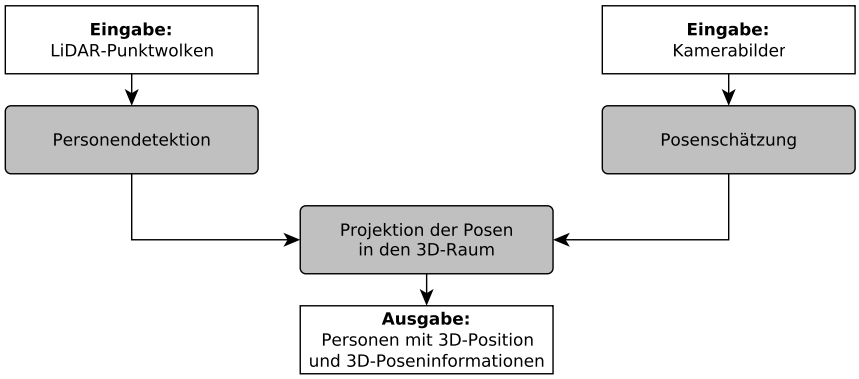
Die zweite Variante wird von Abbildung 6.1 b) dargestellt. Sie führt die Körperposenschätzung auf den vollständigen Kamerabildern aus. Diese fungiert hierbei dann auch als Personendetektion, da die Posenschlüsselpunkte einer Person auch dafür geeignet sind, die Position der Person selbst zu



(a)



(b)



(c)

Abbildung 6.1: Varianten zur multimodalen Körperposenschätzung und Detektion von Personen a) LiDAR als führender Sensor. Personendetektion in LiDAR-Daten und anschließend Generieren von Bildausschnitten für diese detektierten Personen zur Körperposenschätzung. b) Kameras als führender Sensor. Personendetektion erfolgt als Teil der bildbasierten Körperposenschätzung auf den vollständigen Bildern. LiDAR liefert 3D-Informationen für die zunächst nur in 2D-Bildkoordinaten vorliegenden Ergebnisse. c) Kein führender Sensor. LiDAR zur Personendetektion und vollständige Kamerabilder für die Posenschätzung. Anschließend Zusammenführen der Ergebnisse, dabei gegenseitige Verifikation der Sensormodalitäten.

bestimmen. Die LiDAR-Daten werden genutzt, um zu den Kamerabildern passende Tiefenbilder zu erzeugen. Diese werden dann verwendet, um die zu den Bildkoordinaten der Posenschlüsselpunkte passenden 3D-Koordinaten zu erhalten. Im Ergebnis dieser Variante sind alle Personen enthalten, die während der Posenschätzung erkannt wurden. Für diese sind dann sowohl die Position als auch die Posenschlüsselpunkte verfügbar.

Die dritte Variante wird von Abbildung 6.1 c) dargestellt. Sie ist ähnlich zur ersten Variante, verzichtet aber auf das Generieren der Bildausschnitte und führt stattdessen die Posenschätzung auf den vollständigen Kamerabildern aus. Die Detektion in den LiDAR-Punktwolken sowie die Posenschätzung in den Bildern werden dann gemeinsam genutzt und die Ergebnisse beider Verfahren gegeneinander verifiziert. Im Ergebnis sind dann nur Personen enthalten, die von beiden Verfahren detektiert wurden. Für diese liegen die Position sowie auch die Posenschlüsselpunkte vor.

Im weiteren Verlauf dieses Kapitels werden zunächst die Auswahl einer geeigneten und dem Stand der Technik entsprechenden Methode zur bildbasierten Posenschätzung sowie die an diese gestellten Anforderungen erläutert. Anschließend werden die drei Varianten der Methode in jeweils eigenen Abschnitten näher erklärt.

6.2 Auswahl eines Verfahrens zur Körperposenschätzung

Für diese Arbeit wurde ein geeignetes, dem Stand der Technik entsprechendes Verfahren zur bildbasierten Körperposenschätzung benötigt. Um ein solches auszuwählen wurde auf die in Abschnitt 3.4 vorgestellte Kategorisierung solcher Verfahren zurückgegriffen.

Da mit mehreren Personen in den Kameraaufnahmen gerechnet werden muss, wird zumindest bei den Varianten der Methode, bei denen die vollständigen Kamerabilder verarbeitet werden, ein Verfahren für die Körperposenschätzung benötigt, welches mit mehreren Personen im verarbeiteten Bild umgehen kann. Da es im öffentlichen Verkehrsraum zudem oft vorkommt, dass mehrere Fußgänger eng beisammen laufen, kann auch bei der Generierung von personenbezogenen Bildausschnitten nicht sichergestellt werden, dass diese wirklich nur eine Person zeigen. Zumindest Teile einer zweiten Person sind in solchen Fällen oft sichtbar. Auch kann es vorkommen, dass eine Person durch eine andere weiter im Vordergrund teilweise verdeckt wird. Es ist daher in allen Varianten der Methode notwendig, ein Verfahren zur Körperposenschätzung zu verwenden, welches mit mehreren Personen umgehen kann.

Bei den Mehrpersonenverfahren zur Körperposenschätzung wird eines mit Bottom-Up-Ansatz bevorzugt, da diese nicht darauf basieren, Personen zunächst in den RGB-Bildern zu detektieren, um dann anschließend deren Pose und verschiedenen Körperteile zu erkennen. Stattdessen detektieren sie direkt die Körperteile und die Verbindungen zwischen diesen, was letztlich in den Personen resultiert. Dieser Ansatz ist potenziell robuster in Situationen, in denen sich mehrere Personen im Bild gegenseitig verdecken. Für die Zwecke dieser Arbeit wurde daher OpenPose [Cao et al., 2017; Cao et al., 2019] als ein dem Stand der Technik entsprechendes Bottom-Up Mehrpersonenverfahren zur Körperposenschätzung ausgewählt. Es wird wie in den folgenden Abschnitten beschrieben in die verschiedenen untersuchten Varianten der Methode integriert.

6.3 LiDAR als führender Sensor

Die erste untersuchte Variante (vgl. Abbildung 6.1 a)) nutzt die LiDAR-Sensoren und die mit ihnen erzeugten Punktwolken als führende Datenquelle. Dafür wird auf die in Kapitel 4 erläuterte Personendetektion zurückgegriffen, um Personen zunächst in diesen Punktwolken zu detektieren.

Anschließend erfolgt eine zielgerichtete Bildauswertung zur Körperposenschätzung, basierend auf den dann bekannten Positionen der Personen im Umfeld. Die Idee hinter dieser Variante der Methode ist es, den Aufwand für die Bildverarbeitung zu minimieren, indem nur die Bereiche der Bilder verarbeitet werden von denen bekannt ist, dass sie Personen enthalten.

6.3.1 Generieren von Bildausschnitten und Körperposenschätzung

Zunächst werden die Datenströme der Sensoren so gut wie möglich zeitlich synchronisiert. Das heißt, es werden die Kameraaufnahmen gesucht, deren Zeitstempel am besten zu dem der Punktwolke passt. Bei scannenden LiDAR-Sensoren ist eine vollständige Zeitsynchronisation nicht möglich, da diese eine Punktwolke nicht zu einem bestimmten Zeitpunkt, sondern in einem Zeitverlauf aufnehmen. Bei einem LiDAR-Sensor mit einem rotierenden Kopf, der 10 Rotationen pro Sekunde macht, umfasst die Punktwolke eines Einzelscans beispielsweise Messungen im Zeitraum von 0,1 s. Es verbleibt also meist zwangsläufig eine gewisse zeitliche Differenz zwischen der Erfassung einer bestimmten Person mit einem LiDAR-Sensor und der zeitlich am besten dazu passenden Erfassung derselben Person durch eine RGB-Kamera. Aufgrund der relativ niedrigen Bewegungsgeschwindigkeit von Personen wird jedoch davon ausgegangen, dass diese mögliche Ungenauigkeit akzeptabel ist.

Die folgende Verarbeitung erfolgt für jede in der Punktwolke detektierte Person individuell. Es werden aus den zeitlich passenden Bildern Ausschnitte generiert, die möglichst nur diese Person zeigen. Hierfür wird die 3D-Bounding Box der Person als Basis genommen, deren Eckpunkte in die Bildkoordinatensysteme der verschiedenen Kameras des Sensorsystems projiziert werden. Dies erfolgt in einem zweistufigen Prozess. Zunächst werden die Koordinaten der 8 Eckpunkte der Bounding Box, vom Punktwolkenkoordinatensystem in das Kamerakoordinatensystem transformiert:

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{xx} & a_{xy} & a_{xz} & a_{xt} \\ a_{yx} & a_{yy} & a_{yz} & a_{yt} \\ a_{zx} & a_{zy} & a_{zz} & a_{zt} \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{6.1}$$

wo (X, Y, Z) = Ursprüngliche Koordinaten im Punktwolkenkoordinatensystem
(X', Y', Z') = Resultierende Koordinaten im Kamerakoordinatensystem

Anschließend werden die Kameramatrix sowie die Verzeichnungsparameter der Kamera verwendet, um die 3D-Koordinaten in das Bildkoordinatensystem zur projizieren:

$$
\begin{aligned}
x &= X'/Z' \\
y &= Y'/Z' \\
r^2 &= x^2 + y^2 \\
x' &= x\frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 xy + p_2(r^2 + 2x^2) \\
y' &= y\frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1(r^2 + 2y^2) + 2p_2 xy \\
u &= f_x * x' + c_x \\
v &= f_y * y' + c_y
\end{aligned}
\tag{6.2}
$$

wo (X', Y', Z') = Koordinaten im Kamerakoordinatensystem
(u, v) = Resultierende Bildkoordinaten in Pixel
(c_x, c_y) = Bildhauptpunkt
f_x, f_y = Brennweite in Pixel
$k_1, k_2, k_3, k_4, k_5, k_6$ = Radiale Verzeichnungsparameter
p_1, p_2 = Tangentiale Verzeichnungsparameter

Basierend auf den resultierenden Eckpunkten der 3D-Bounding Box in Bildkoordinaten wird ein an den Bildachsen ausgerichteter Bildausschnitt erzeugt, der alle Eckpunkte der Bounding Box umschließt. Dabei werden Bounding Boxen ignoriert, die zu ungültigen Bildkoordinaten führen. Wenn mehrere Kameras verfügbar sind, wird dieser Schritt für alle Kameras wiederholt. Es ist also denkbar, dass es für dieselbe Person einen gültigen Bildausschnitt in den Bildern von mehr als einer Kamera gibt.

Die erzeugten Bildausschnitte werden anschließend mit OpenPose für die eigentliche Körperposenschätzung verarbeitet. Dieses liefert als Ergebnis Posenschlüsselpunkte, die nach den im Bildausschnitt erkannten Personen gruppiert sind. Bei diesen Schlüsselpunkten handelt es sich um die Position gewisser Körperteile, wie z.B. die Extremitäten oder Gelenke. Sie verfügen dementsprechend über einen Typ, der angibt welchen Körperteil sie repräsentieren. Die Schlüsselpunkte liegen in Bildkoordinaten vor und sind außerdem mit einem Wert für die Konfidenz versehen. Schlüsselpunkte, deren Konfidenz unter einem gewissen Schwellwert liegt, werden verworfen. Die verbleibenden werden als valide angesehen.

6.3.2 Bestimmen von 3D-Koordinaten und Zuordnung der Ergebnisse der Körperposenschätzung zu den Detektionen

Es ist nun erforderlich, die richtigen Posenschlüsselpunkte der im 3D-Raum detektierten Person zuzuordnen und eine 3D-Koordinate für diese zu bestimmen. Dabei gibt es zwei Schwierigkeiten: Zum einen kann es vorkommen, dass auch in den gezielt erzeugten Bildausschnitten mehr als eine Person zu sehen ist. Dies tritt vor allem dann auf, wenn mehrere Personen nahe beieinander sind bzw. eine Person von einer anderen teilweise verdeckt wird. In solchen Fällen liefert die Körperposenschätzung Ergebnisse für alle im Bildausschnitt sichtbaren Personen. Es ist also erforderlich festzustellen, welche dieser Ergebnisse zu der Person gehören, für die der Bildausschnitt generiert wurde. Eine andere Schwierigkeit ergibt sich, wenn die Daten von mehr als einer Kamera genutzt werden und es Überschneidungen im Sichtbereich dieser Kameras gibt. In einem solchen Fall können Ergebnisse für den gleichen Typ Schlüsselpunkt von mehr als einer Kamera vorliegen.

Wenn es für dieselbe Person Ergebnisse der Körperposenschätzung von mehr als einer Kamera gibt, werden diese zunächst separat betrachtet. Die Bildkoordinaten der Posenschlüsselpunkte wer-

den als 3D-Strahlen in das Koordinatensystem der Punktwolken überführt. Dafür werden die auch in Abschnitt 6.3.1 verwendeten intrinsischen und extrinsischen Kalibrierparameter der Kamera genutzt. Die resultierenden Strahlen haben ihren Ursprung im Projektionszentrum der jeweiligen Kamera und die 3D-Koordinaten ihres jeweiligen Posen-Schlüsselpunkts befinden sich irgendwo auf ihnen. Da es wie beschrieben sein kann, dass die Posenschätzung Ergebnisse für mehr als eine im Bildausschnitt sichtbare Person geliefert hat, sind nun zwei Probleme zu lösen: Zum einen muss entschieden werden, bei welcher in den Ergebnissen der Körperposenschätzung vorhandenen Person es sich um die handelt, für die der Bildausschnitt generiert wurde. Zum anderen müssen für deren Posenschlüsselpunkte 3D-Koordinaten auf den Strahlen bestimmt werden.

Zur Lösung beider Probleme wird auf dem Strahl jedes Posenschlüsselpunkts die Position bestimmt, an der dieser Strahl einem 3D-Punkt der Punktwolke, welcher laut Personendetektion zur Person gehört, am nächsten kommt. Hierbei werden dieselben Punkte berücksichtigt, die auch bei der Erzeugung der 3D-Bounding Box genutzt wurden und ursprünglich zu einer lokalen Punktnachbarschaft geführt haben, welche die Detektion unterstützt hat (vgl. Abschnitt 4.4). Für jeden dieser Punkte wird berechnet, wie nah der Strahl ihm kommt und an welchen Koordinaten auf dem Strahl dies der Fall ist:

$$
\begin{aligned}
L &= \frac{\begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} \cdot \left(\begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} - \begin{pmatrix} U_x \\ U_y \\ U_z \end{pmatrix} \right)}{\begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} \cdot \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix}} \\
\begin{pmatrix} K_x \\ K_y \\ K_z \end{pmatrix} &= \begin{pmatrix} U_x \\ U_y \\ U_z \end{pmatrix} + \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} L \\
D &= \left\| \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} - \begin{pmatrix} K_x \\ K_y \\ K_z \end{pmatrix} \right\|
\end{aligned}
\tag{6.3}
$$

wo
(P_x, P_y, P_z) = Koordinaten des untersuchten Punkts P
(U_x, U_y, U_z) = Ursprungspunkt des Strahls
(R_x, R_y, R_z) = Einheitsvektor, der die Richtung des Strahls angibt
(K_x, K_y, K_z) = Punkt P am nächsten liegende Koordinaten auf dem Strahl
D = Kürzeste Distanz zwischen Strahl und Punkt P

Für jeden Posenschlüsselpunkt wird anschließend das Ergebnis für den 3D-Punkt verwendet, bei dem die resultierende Distanz D am geringsten ist. Die anderen werden verworfen. Die daraus resultierenden 3D-Koordinaten für den jeweiligen Posenschlüsselpunkt werden als die am besten passenden angenommen. Wenn es wie beschrieben in der Körperposenschätzung Ergebnisse für mehrere Personen gibt, wird zusätzlich die resultierende durchschnittliche Entfernung aller Posenschlüsselpunkte der Personen aus der Posenschätzung bestimmt. Die Annahme ist nun, dass diese Distanz dann am geringsten ist, wenn es sich um die gesuchte Person handelt, für die auch der verarbeitete Bildausschnitt generiert wurde. Es werden also nur deren Ergebnisse weiter verwendet und die anderen verworfen.

Nachdem das Zuordnungsproblem für die Ergebnisse von Bildern einer Kamera gelöst und 3D-Koordinaten für die Posenschlüsselpunkte bestimmt wurden, muss noch berücksichtigt werden,

dass es Ergebnisse der Posenschätzung für dieselbe Person von mehreren Kameras geben kann. Diese werden zusammengeführt. Das Ziel dabei ist es, von jeden Typ von Posenschlüsselpunkt pro Person nur einen in das Ergebnis zu übernehmen. Posenschlüsselpunkte von einem Typ, der für die Person nur einmal gefunden wurde, werden direkt in das Endergebnis übernommen. Wenn ein Typ mehrmals in den Bildern verschiedener Kameras gefunden wurde, findet eine Auswahl statt. Hierbei gibt eine zweistufige Priorisierung: Schlüsselpunkte von Kameras, deren Bildausschnitte nur eine einzelne Person zeigen, werden gegenüber solchen vorgezogen, die mehrere Personen zeigen. So soll erreicht werden, dass Fehler im vorherigen Zuordnungsschritt möglichst nicht in das Endergebnis übernommen werden. Wenn es nach dieser Priorisierung für dieselbe Person immer noch mehrere konkurrierende Schlüsselpunkte vom selben Typ gibt, werden die mit der höheren Konfidenz in der Posenschätzung genutzt. Das resultierende Endergebnis sind dann die ursprünglich in den Punktwolken detektierten Personen, ergänzt um die für sie bestimmten Posenschlüsselpunkte, welche ebenfalls mit einer 3D-Koordinate versehen wurden.

6.4 Kameras als führende Sensoren

Die zweite Variante der Methode, dargestellt von Abbildung 6.1 b), verarbeitet die Kamerabilder vollständig mit der Körperposenschätzung. Diese wird dann nicht nur für die Bestimmung der Posenschlüsselpunkte genutzt, sondern implizit auch für die eigentliche Detektion der Personen. Die LiDAR-Sensoren werden dann nur genutzt, um für die zunächst in Bildkoordinaten vorliegenden Ergebnisse 3D-Koordinaten zu bestimmen.

Wie auch bei der ersten Variante, werden die Punktwolken der LiDAR-Sensoren sowie die Bilder der Kameras zunächst zeitlich so gut wie möglich zueinander synchronisiert. Die beiden danach folgenden Verarbeitungsschritte erfolgen parallel. Diese sind zum einen die eigentliche Körperposenschätzung auf den vollständigen Kamerabildern und zum anderen ein Erzeugen von ggf. nicht vollständig gefüllten Tiefenbildern für jede verwendete Kamera.

6.4.1 Erzeugen von Tiefenbildern

Die Erzeugung von Tiefenbildern erfolgt für jede verwendete Kamera separat. Es werden zunächst alle 3D-Punkte der LiDAR-Punktwolke in das Bildkoordinatensystem der jeweiligen Kamera projiziert. Hierbei wird dasselbe Vorgehen verwendet wie beim Projizieren der 3D-Bounding Boxen, welches von Variante 1 verwendet wird (vgl. Abschnitt 6.3.1). Punkte, die dabei zu ungültigen Bildkoordinaten führen, werden verworfen. Für die verbleibenden Punkte wird nun ein Tiefenwert bestimmt, wobei es sich um die Distanz zwischen dem Projektionszentrum der Kamera und der 3D-Koordinate des Punktes handelt.

Mithilfe der für die 3D-Punkte bestimmten Bildkoordinaten und Tiefenwerte wird nun ein Tiefenbild erzeugt. Dieses hat dieselbe Auflösung wie die Bilder der Kamera, für die es erzeugt wird. Dabei ist zu berücksichtigen, dass aufgrund der üblicherweise deutlich geringeren Datendichte in den Punktwolken ein Tiefenbild, bei dem nur die Pixel an den direkt errechneten Bildkoordinaten der 3D-Punkte gefüllt werden, überwiegend leer bleibt. Um dieses Problem zu reduzieren, werden nicht nur die Pixel, die direkt den errechneten Bildkoordinaten entsprechen von einem 3D-Punkt gefüllt, sondern in einem gewissen Radius auch benachbarte Pixel. Alle diese Pixel erhalten den zuvor errechneten Tiefenwert im Tiefenbild. Sollte es vorkommen, dass ein Pixel im Tiefenbild einen Tiefenwert von mehreren verschiedenen 3D-Punkten erhält, wird dem niedrigeren Wert der Vorzug gegeben. Oberflächen im Vordergrund werden also gegenüber welchen im Hintergrund bevorzugt. Das resultierende Tiefenbild ist immer noch nicht zwangsläufig vollständig gefüllt. Dies wird aber in der nachfolgenden Verarbeitung berücksichtigt und stellt für diese kein Problem dar. Abbildung 6.2 b) zeigt ein Beispiel für ein erzeugtes Tiefenbild.

(a)

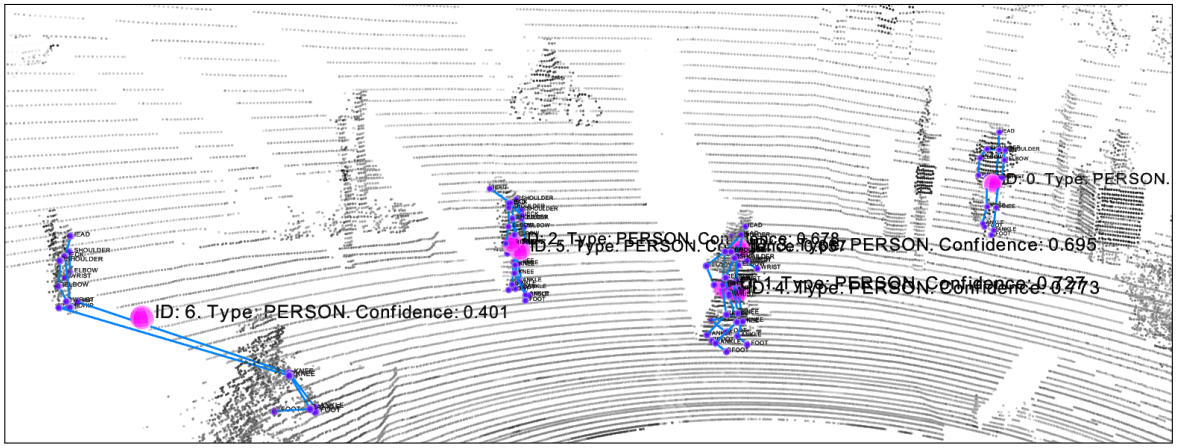
(b)

Abbildung 6.2: Ergebnisse der beiden ersten Verarbeitungsschritte der zweiten Variante der Methode. a) Ergebnis der bildbasierten Posenschätzung. Die Posenschlüsselpunkte und deren Verbindungen werden auf das Bild projiziert dargestellt. b) Ein erzeugtes Tiefenbild. Je heller ein Pixel ist, umso weiter ist er von der Kamera entfernt. Komplett schwarze Pixel haben keinen Tiefenwert erhalten.

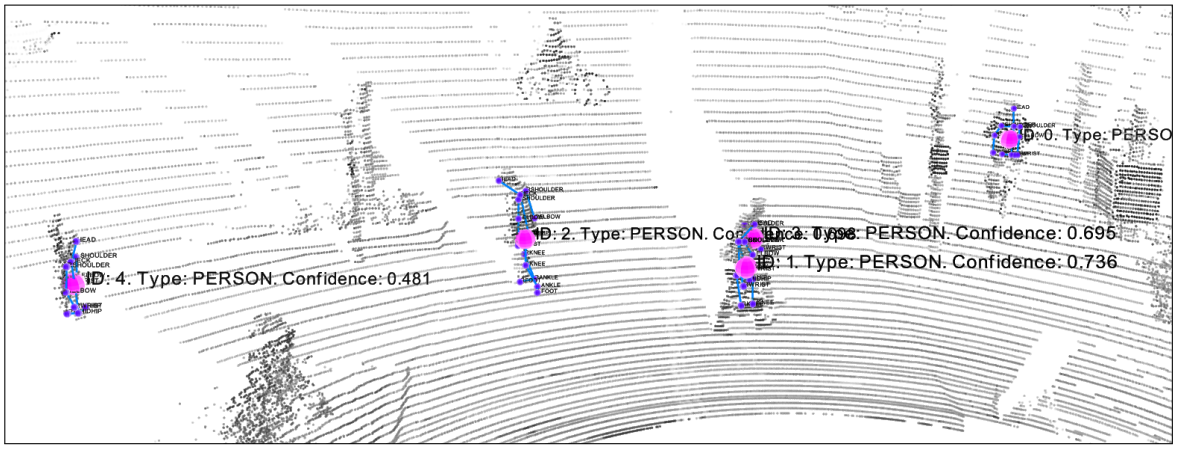
6.4.2 Bestimmen von 3D-Koordinaten für Posenschlüsselpunkte und Personen

Dieser Verarbeitungsschritt hat zwei Eingaben: Die Ergebnisse der Körperposenschätzung auf den Kamerabildern und die erzeugten Tiefenbilder. Diese Eingaben werden auch anhand eines Beispiels in Abbildung 6.2 dargestellt. Es gilt nun die Ergebnisse der Körperposenschätzung mit 3D-Koordinaten zu versehen. Da berücksichtigt wird, dass die Tiefenbilder nicht zwangsläufig vollständig gefüllt sind, wird hierfür ähnlich wie in Abschnitt 6.3.2 zunächst ein 3D-Strahl für die Posenschlüsselpunkte bestimmt. Um zu entscheiden, wo auf diesem Strahl die 3D-Koordinate des Posenschlüsselpunkts liegt, wird auf das Tiefenbild zurückgegriffen. Da dieses nicht vollständig

gefüllt ist, wird der Tiefenwert des gefüllten Pixels genommen, das am nächsten an den Bildkoordinaten des Posenschlüsselpunkts dran ist. Hierbei wird außerdem ein Schwellwert für den maximalen Abstand zwischen Bildkoordinaten des Schlüsselpunkts und dem nächsten gefüllten Tiefenbild Pixel angewendet. Dieser ist erforderlich, da die Sichtbereiche der Kameras nicht vollständig von den LiDAR-Sensoren abgedeckt werden. Für größere Bildbereiche, die gar nicht von den LiDAR-Sensoren erfasst werden, aber auch für welche, die durch Verdeckungen von diesen temporär nicht eingesehen werden können, kann kein sinnvoller Tiefenwert bestimmt werden. Posenschlüsselpunkte, die in solchen Bereichen liegen, werden dementsprechend nicht weiter verarbeitet und verworfen. Um neben den Posenschlüsselpunkten auch eine 3D-Koordinate für die Person als ganzes zu haben, wird zusätzlich die durchschnittliche 3D-Position aller Posenschlüsselpunkte bestimmt. Ähnlich wird ihre durchschnittliche Konfidenz als Konfidenz der Person genutzt.



(a)



(b)

Abbildung 6.3: Beispielergebnisse für die zweite Variante der Methode, dargestellt wird die Punktwolke mit den detektieren Personen und deren Posenschlüsselpunkten. a) Ergebnisse ohne Clustering zur Validierung. 3D-Koordinaten der Posenschlüsselpunkte werden ggf. durch Verdeckungen falsch bestimmt. b) Ergebnisse mit Validierung und zusätzlichem Zusammenführen von Detektionen aus unterschiedlichen Bildern mit überlappendem Sichtbereich.

Abbildung 6.3 a) zeigt beispielhaft das Ergebnis nach der Bestimmung der 3D-Koordinaten. An der Person mit der ID 6, welche sich im linken Bereich der Abbildung befindet, ist ein Problemfall des Verfahrens zu erkennen: Es kann vorkommen, dass im Bild Posenschlüsselpunkte in Bereichen detektiert werden, die in den Punktewolken von anderen Objekten weiter im Vordergrund

verdeckt sind. In solchen Fällen, wird unter Umständen der Tiefenwert von diesen Objekten im Vordergrund genommen, um die 3D-Position der betroffenen Posenschlüsselpunkte im Tiefenbild zu bestimmen, was zu fehlerhaften Ergebnissen führt. Um solche zu erkennen und zu entfernen, wird ein Clustering der Posenschlüsselpunkte im 3D-Raum eingesetzt, um die 3D-Koordinaten der Posenschlüsselpunkte zu validieren.

Das Clustering erfolgt für jede Person separat. Es wird ein hierarchisches *Single-Linkage* Clustering verwendet. Bei einem solchen werden zunächst alle Posenschlüsselpunkte als ein eigenes Cluster angesehen. Diese werden dann in aufsteigender Reihenfolge ihrer Distanz zueinander miteinander vereint. Die Distanz zwischen zwei Clustern definiert sich dabei als die geringst mögliche Distanz zwischen zwei Elementen der beiden Cluster. Für den Zweck der Validierung der Posenschlüsselpunkte, wird ein Schwellwert für die maximale Distanz zwischen den nächsten zu vereinenden Clustern als Abbruchkriterium verwendet. Wenn die Posenschlüsselpunkte einer Person im 3D-Raum zu weit auseinander liegen, verbleiben am Ende des Clusterings also mehrere Cluster. In solchen Fällen werden nur die zum größten Cluster gehörenden Schlüsselpunkte als korrekte angesehen und behalten. Die anderen werden verworfen.

Im mittleren Bereich von Abbildung 6.3 a) zeigt sich ein weiteres Problem, welches entsteht, wenn mehrere Kameras mit überlappendem Sichtbereich verwendet werden. Die Personen dort wurden mehrmals in verschiedenen Bildern detektiert, was zu doppelten Ergebnissen führt. Um dieses Problem zu beheben, werden Personen aus verschiedenen Bildern anhand ihrer Position im 3D-Raum zusammengefasst, wenn sie nah genug beieinander sind. Im Ergebnis werden dann für die Posenschlüsselpunkte von jedem Typ die Ergebnisse mit der höchsten Konfidenz genutzt. Die Position der Person als ganzes sowie ihre Konfidenz wird entsprechend neu als Durchschnitt der entsprechenden Werte von den letztlich verwendeten Posenschlüsselpunkten berechnet. Nach diesem Zusammenfassen und der Validierung ergibt sich ein bereinigtes Ergebnis, dieses wird von Abbildung 6.3 b) beispielhaft dargestellt.

6.5 Beide Sensoren gleichwertig

Die dritte Variante der Methode wird von Abbildung 6.1 c) dargestellt. In ihr werden beide Sensoren gleichwertig verwendet und dafür genutzt, ihre Ergebnisse gegenseitig zu bestätigen. Auch hier werden wieder die Daten der verschiedenen Sensoren soweit wie möglich zeitlich synchronisiert. Es erfolgt dann, wie in der ersten Variante, eine Detektion der Personen in den LiDAR-Punktwolken aber auch, wie in der zweiten Variante, eine Posenschätzung in den vollständigen Kamerabildern. Die Ergebnisse von beidem werden anschließend zusammengeführt. In das Endergebnis werden nur Personen übernommen, die in der Punktwolke detektiert wurden und für die zusätzlich eine Körperpose bestimmt werden konnte.

Die Fusion zwischen den Sensormodalitäten erfolgt ähnlich wie bei der ersten Variante. Es liegen die Ergebnisse der Personendetektion in den Punktwolken als 3D-Koordinaten vor. Außerdem die aus den Bilddaten ermittelten Posenschlüsselpunkte in Bildkoordinaten, die nach Personen und Bildern gruppiert sind. Diese Bilder werden im Folgenden individuell verarbeitet. Anhand der Bildkoordinaten der Posenschlüsselpunkte werden, wie auch in den anderen Varianten, 3D-Strahlen im Punktwolkenkoordinatensystem bestimmt. Da die Posenschlüsselpunkte aber, anders als in der ersten Variante, nicht aus einem für eine spezifische detektierte Person erzeugten Bildausschnitt stammen, werden in dem Prozess, 3D-Koordinaten anhand dieser Strahlen zu bestimmen nun alle Personen in den Punktwolken berücksichtigt.

Für jede Person aus dem gerade verarbeiteten Bild wird also anhand jeder Person aus den Punktwolken ein Satz von 3D-Koordinaten für die Posenschlüsselpunkte bestimmt, wofür Gleichung 6.3 verwendet wird. Es wird dabei außerdem ein Durchschnitt für die minimale Distanz

zwischen den 3D-Strahlen und zur Person gehörenden 3D-Punkten bestimmt. Also der Durchschnitt von D. Wenn dieser unter einem gewissen Schwellwert liegt, wird die Zuordnung von Person aus den Bilddaten zur Person aus den Punktwolken als valide angenommen. Konnte für eine Person aus den Bilddaten gar keine valide Zuordnung gefunden werden, werden sie und ihre Posenschlüsselpunkte verworfen. Wenn mehrere valide Zuordnungen für dieselbe Person aus den Bilddaten gefunden werden, wird diejenige mit der geringsten durchschnittlichen Distanz als korrekt angesehen und genutzt. Die Methode ist hierbei „gierig“. Dies bedeutet, wenn es mehrere Personen in den Posendaten eines Bildes gibt, werden Personen aus den Punktwolken, die bereits einer anderen Person aus dem Bild zugeordnet wurden, nicht mehr bei der Zuordnung der nachfolgenden Personen aus diesem Bild berücksichtigt.

Wenn mehrere Bilder verarbeitet werden, kann es bei überschneidenden Sichtbereichen der Kameras dazu kommen, dass einer Person aus den Punktwolken mehrere Personen aus verschiedenen Bildern zugeordnet werden. In einem solchen Fall wird wieder angenommen, dass es sich dabei eigentlich um dieselbe Person handelt. Deren Posenschlüsselpunkte werden daher ähnlich wie in Variante 2 zusammengefasst. Allerdings wird für die Position der Person als ganzes sowie der dieser Person zugeordneten Konfidenz immer der Wert aus der Detektion in den LiDAR-Punktwolken verwendet. Personen aus den Punktwolken, denen keine Posenschlüsselpunkte zugeordnet wurden, die also in den Bilddaten nicht erkannt wurden, werden verworfen und nicht in das Endergebnis übernommen. In diesem hat dementsprechend jede Person auch Posenschlüsselpunkte.

7 Experimente

In diesem Kapitel wird die experimentelle Bewertung der in den vorherigen Kapiteln vorgestellten Verfahren erläutert. Es wird zunächst das Experimentalsystem vorgestellt, welches zur Datenerfassung für die Experimente genutzt wurde. Anschließend werden die damit aufgenommenen und für die Experimente genutzten Daten erläutert. Danach folgt eine Erklärung, wie die Ergebnisse der Experimente bewertet werden, woran sich die Erläuterung der durchgeführten Experimente selbst anschließt.

7.1 Eingesetztes Experimentalsystem MODISSA

Für die Experimente dieser Arbeit wurden Daten des mobilen Multisensorsystems MODISSA verwendet, welches am Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB) in Ettlingen auf Basis eines VW-Transporters aufgebaut wurde. MODISSA wird sowohl verwendet für die reine Aufzeichnung von unverarbeiteten Rohdaten als auch für die direkte Datenverarbeitung auf dem Sensorsystem. Die Datenaufzeichnung dient dabei u.a. dazu Daten bereitzustellen, die für eine spätere experimentelle Untersuchung verschiedener Verfahren genutzt werden können. Die Datenverarbeitung auf dem Fahrzeug selbst erlaubt es hingegen solche Verfahren direkt in einem Live-Betrieb zu nutzen. Hierdurch wird es auch möglich interaktive Verfahren zu untersuchen, die beispielsweise direkt beeinflussen welche Bereiche das Fahrzeug mit schwenkbaren Sensoren erfasst. Der Live Betrieb wird auch für Vorführungen verwendet. Das Sensorsystem wurde in zwei Veröffentlichungen beschrieben [Borgmann et al., 2018c, 2021b].

In den folgenden Abschnitten wird zunächst die Hardware des Sensorsystems vorgestellt. Anschließend erfolgt eine Erläuterung, wie die verschiedenen Sensoren synchronisiert werden. Danach wird das Softwaresystem vorgestellt, welches für die direkte Datenverarbeitung auf dem System genutzt wird und u.a. für diese Arbeit erstellt wurde. Zuletzt wird ein Überblick über eine Geodatenbank gegeben, welche im Rahmen dieser Arbeit als Beispiel für die langfristige Ablage der Auswertungsergebnisse sowie für die spätere Zusammenführung dieser Ergebnisse mehrerer Fahrzeuge verwendet wurde.

7.1.1 Hardware von MODISSA

Das Multisensorsystem MODISSA ist auf Basis eines VW-Transporters aufgebaut. Die genaue Ausstattung des Fahrzeugs unterliegt einem gewissen Wandel, da es als erweiterbares und anpassungsfähiges Sensorsystem ausgelegt wurde. Im Folgenden wird der Aufbau des Fahrzeugs beschrieben, wie er zum Zeitpunkt der Datenerfassung für die Experimente in dieser Arbeit bestand.

Abbildung 7.1 zeigt das Fahrzeug und dessen Sensoren von außen. Diese Sensoren sind primär auf dem Dach des Fahrzeugs untergebracht. Sie werden im Folgenden erläutert. Anschließend wird der Innenausbau des Fahrzeugs kurz beschrieben. Dieser umfasst die Rechner-Hardware, welche die zweite Sitzreihe ersetzt sowie die Stromversorgung, die im Kofferraum untergebracht ist.

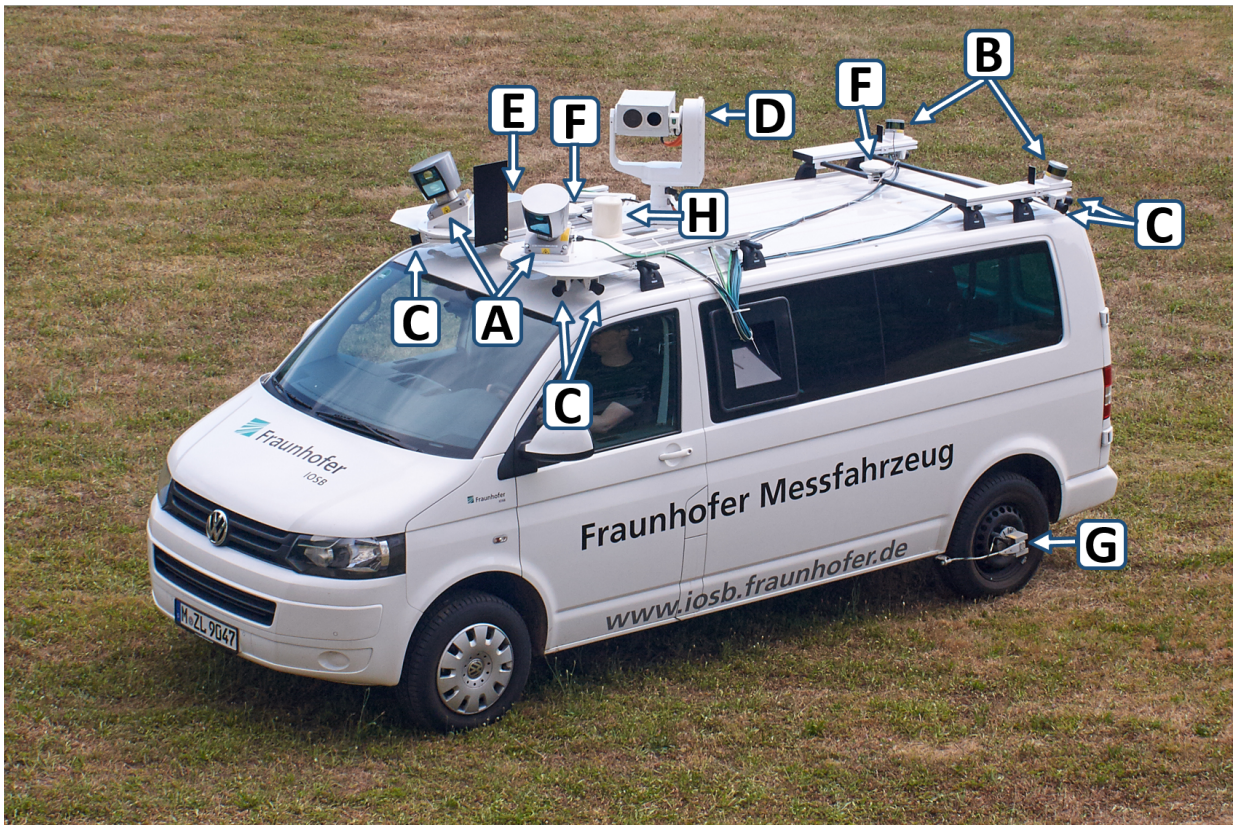


Abbildung 7.1: Das Multisensorfahrzeug MODISSA, das für die Experimente genutzt wurde. Die verschiedenen Hardwarekomponenten sind markiert.
A: 2x Velodyne HDL-64E LiDAR
B: 2x Velodyne VLP-16 LiDAR
C: 8x Baumer VLG-20C.I. Kameras, 2 Kameras pro Ecke zur 360° Abdeckung des Umfelds
D: Schwenk-Neigekopf ausgestattet mit Jenoptik IR-TCM 640 LWIR-Kamera, JAI CM 200-MCL Kamera und Jenoptik DLEM 20 Laserentfernungsmesser
E, F, G: Applanix POS LV V5 520 INS mit **E:** IMU, **F:** 2x GNSS-Empfänger und **G:** Raddrehgeber
H: WiFi-Antenne

LiDAR-Sensoren

Das Fahrzeug verfügt insgesamt über vier LiDAR-Sensoren. Vorne links und rechts des Dachaufbaus befindet sich jeweils ein Velodyne HDL-64E. Hierbei handelt sich um einen LiDAR-Sensor, der mithilfe eines rotierenden Kopfes einen horizontalen Erfassungsbereich von 360° abdeckt. Der vertikale Erfassungsbereich geht von +2° bis −24,8°. Er ist also hauptsächlich nach unten gerichtet. Dies ist konsistent mit dem primären Einsatzbereich dieser Sensoren, welche für die Nutzung im Rahmen des autonomen Fahrens entwickelt wurden. Dieser vertikale Erfassungsbereich ist in 64 Laserscanzeilen unterteilt, was zu einer vertikalen Winkelauflösung von etwa 0,4° führt. Die Sensoren führen 1.300.000 Messungen pro Sekunde durch. Die Rotationsgeschwindigkeit des Kopfes kann zwischen 5 und 20 Umdrehungen pro Sekunde frei gewählt werden. Dies führt zu einer von der Rotationsgeschwindigkeit abhängigen horizontalen Winkelauflösung zwischen ca. 0,09° und 0,35°. Für die Zwecke dieser Arbeit wurde eine Rate von 10 Umdrehungen pro Sekunde gewählt, welche in einer horizontalen Winkelauflösung von ca. 0,17° resultiert. Die Genauigkeit der durchgeführten Entfernungsmessungen liegt bei ±2 cm und Messungen sind bis zu einer Entfernung von 120 m möglich. Die Sensoren sind an dem Fahrzeug auf einem austauschbaren Sockel montiert, sie können so in verschiedene Richtungen gekippt werden. Dies erlaubt es abhängig vom

Einsatzzweck den Erfassungsbereich zu verschieben, beispielsweise für Anwendungen im Bereich des Mobile Mappings. Im Rahmen dieser Arbeit werden Daten von diesen Sensoren verwendet.

An den beiden hinteren Ecken des Dachaufbaus befindet sich jeweils ein Velodyne VLP-16. Hierbei handelt es sich um technisch mit den HDL-64E verwandte aber kleinere Sensoren. Sie verfügen nur über 16 statt 64 Scanzeilen, die einen vertikalen Erfassungsbereich von 30° abdecken. Die daraus resultierende vertikale Winkelauflösung ist dementsprechend signifikant geringer und liegt bei nur 2°. Die Verwendung dieser Sensoren wurde für diese Arbeit in Erwägung gezogen aber aufgrund dieser geringen vertikalen Auflösung verworfen.

Rundumkameras

An allen vier Ecken des Dachaufbaus befinden sich jeweils zwei Baumer VLG-20C.I. Kameras. Diese Paare sind jeweils horizontal in einem Winkel von 66° zueinander angeordnet, um es zu ermöglichen das Fahrzeugumfeld horizontal in alle Richtungen zu erfassen, was bereits ab einer Entfernung von weniger als 1 m der Fall ist. Vertikal sind die Kameras um 15° nach unten geneigt. Diese Farbkameras erstellen Bilder in einer Auflösung von 1624x1228, welche im Bayer-Pattern ausgegeben werden. Für die Datenerfassung werden die Kameras mit einem externen Triggersignal gesteuert, was in Abschnitt 7.1.2 näher erläutert wird. Sie wurden im Rahmen dieser Arbeit mit 10 Aufnahmen pro Sekunde getriggert und ihre Daten werden für die bildbasierten Auswertungen genutzt.

Schwenk-Neigekopf

MODISSA verfügt über einen Schwenk-Neigekopf (SNK) , an dem mehrere Sensoren untergebracht sind. Ein Jenoptik IR-TCM 640 Mikrobolometer als Kamera für das thermische Infrarot und eine JAI CM 200-MCL Kamera für Graustufenbilder. Diese beiden Sensoren haben außerdem eine im Vergleich zu den Rundumkameras längere Brennweite. Der SNK wird daher genutzt um detailliertere Aufnahmen für Gebiete von Interesse zu machen. Er verfügt zusätzlich über einem Jenoptik DLEM 20 Laserentfernungsmesser. Dieser kann Entfernungen bis zu 5 km messen. Er wird u.a. dazu genutzt, die Entfernung zu einem Objekt zu bestimmen, welches gerade von den Kameras des SNK beobachtet wird. Für die Auswertungen im Rahmen dieser Arbeit wurden die Sensoren des SNK nicht verwendet.

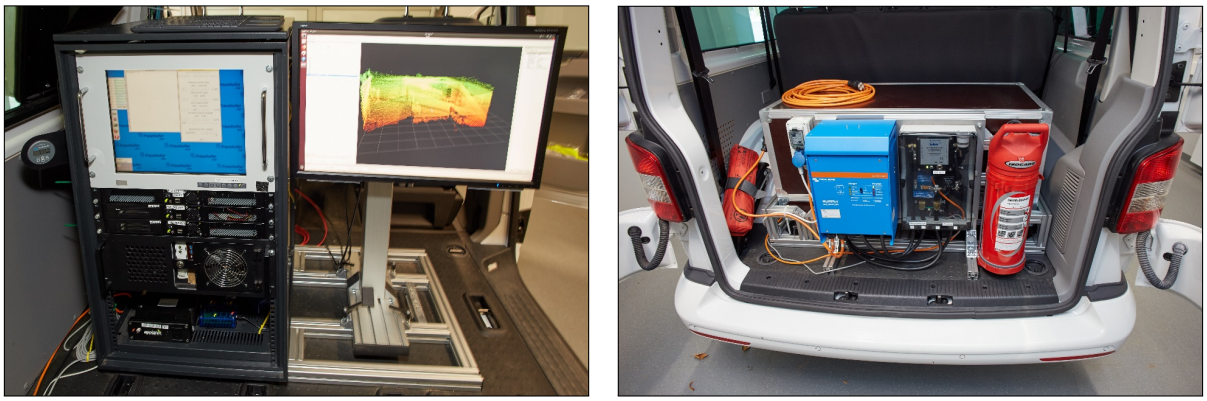
INS

Das Fahrzeug verfügt über ein Applanix POS LV V5 520 inertiales Navigationssystem (INS), welches aus mehreren Einzelkomponenten besteht. Diese umfassen auf dem Dach montiert eine *Inertial Measurement Unit* (IMU) zur Messung von Rotation und Beschleunigung in allen drei Freiheitsgraden sowie zwei GNSS (Globales Navigationssatellitensystem) Antennen für die Bestimmung der Position und horizontalen Ausrichtung des Fahrzeugs. Zusätzlich verfügt das INS über einen Raddrehgeber (auch *Distance Measurement Indicator* - DMI genannt), mit dem die zurückgelegte Strecke erfasst werden kann. Die Daten der verschiedenen Sensoren des INS werden innerhalb eines im Fahrzeuginneren montierten PCS (POS Computer System) zusammengeführt, welcher Teil des INS ist. Dieser bestimmt eine Navigationslösung und stellt die Daten des INS den anderen Komponenten des Fahrzeugs zur Verfügung.

Die primäre Funktion des INS ist es die Position und Ausrichtung des Fahrzeugs in der Welt zu bestimmen. Es ist dementsprechend eine zentrale Komponente für die direkte Georeferenzierung der erfassten Daten aller Sensoren. Das INS stellt außerdem eine von allen Komponenten des Sensorsystems gemeinsam genutzte Referenzzeit bereit. Diese basiert auf der über GNSS emp-

fangenen UTC-Zeit. Für diese Aufgaben wurde das INS auch im Rahmen der Datenerfassung für diese Arbeit verwendet.

Innenausbau



(a) (b)

Abbildung 7.2: Der Innenraum von MODISSA. a) Rechner-Rack und Monitore, welche die zweite Sitzreihe ersetzen. b) Stromversorgung im Heck.

Abbildung 7.2 zeigt Komponenten des Sensorsystems, welche im Innenraum des Fahrzeugs montiert sind. Der VW-Transporter verfügt normalerweise über drei Sitzreihen. Die mittlere davon wurde entfernt und durch in einem Rack montierte Hardware sowie Monitore ersetzt. Die hintere Reihe wird von den Bedienern des Systems als Sitzplatz genutzt. Das Rack ist zu sehen in Abbildung 7.2 a) und umfasst insgesamt vier Computer, die aus normaler Server-Hardware aufgebaut sind, den PCS des INS sowie weitere Komponenten für die Synchronisierung und Ansteuerung der Sensoren (siehe auch Abschnitt 7.1.2). Die Datenflüsse zwischen den Sensoren und diesen Komponenten sind in Abbildung 7.3 schematisch dargestellt.

Einer der Computer ist für die Steuerung der Datenaufzeichnung zuständig. Einer der Monitore des Fahrzeugs ist mit ihm verbunden und er erlaubt es den Bedienern, die Datenerfassung zu starten und zu stoppen sowie bestimmte Einstellungen zu tätigen. Er steuert auch den SNK an und ist zusätzlich für die Aufzeichnung der Daten des INS und der LiDAR-Sensoren zuständig. Jeweils einer der Computer übernimmt die Aufzeichnung der Daten der thermischen Infrarotkamera sowie der Graustufenkamera des SNK. Der vierte Computer ist der leistungsstärkste des Sensorsystems. Er wird zum einen für die Aufzeichnung der 8 Rundumkameras verwendet, zum anderen aber auch für die Datenverarbeitung direkt auf dem Fahrzeug. Mit ihm ist der zweite Monitor verbunden, der auch das Interface für diese Datenverarbeitung darstellt. Alle Computer sind untereinander mit einem Netzwerk verbunden, über das sie Daten austauschen. So können auch Daten aller Sensoren für die Datenverarbeitung auf dem Fahrzeug genutzt werden. Im Abschnitt 7.1.2 wird die Interaktion zwischen den verschiedenen Computern und sonstigen Komponenten erläutert, um eine Synchronisation der Datenaufzeichnung zwischen den verschiedenen Sensoren zu erzielen. In Abschnitt 7.1.3 wird erläutert, wie die Verarbeitung auf dem Fahrzeug auf die Daten aller Sensoren zugreift.

Die in Abbildung 7.2 b) zu sehende Stromversorgung des Sensorsystems ist unabhängig vom Motor des Fahrzeugs. Sie wird über Akkus sichergestellt, die über einen Wechselrichter bis zu 2000 W bei 230 V als Wechselstrom zur Verfügung stellen. Es wird also normaler Netzstrom be-

reitgestellt, was die Flexibilität bei der Auswahl der Komponenten erhöht. Die Akkus haben dabei genügend Energie für einen mehrstündigen Betrieb, wobei die genaue Laufzeit von der Auslastung des Systems abhängt.

7.1.2 Sensorsynchronisation

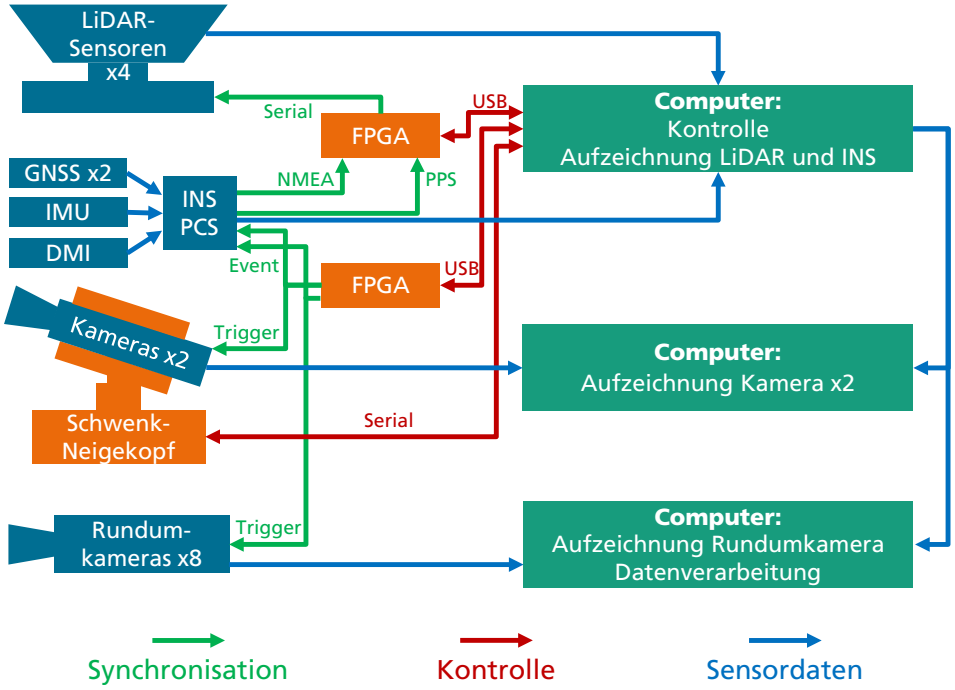


Abbildung 7.3: Datenflüsse im Multisensorfahrzeug MODISSA für die Synchronisation und Kontrolle der Sensoren sowie für ihre Daten

Für die effektive Nutzung der verschiedenen Sensoren eines Multisensorsystems ist es notwendig deren Daten in einen räumlichen und zeitlichen Zusammenhang zu bringen. Der räumliche Zusammenhang ergibt sich aus der extrinsischen Kalibrierung der verschiedenen Sensoren. Die kalibrierten Parameter geben wieder, wo und mit welcher Ausrichtung die Sensoren im Bezug auf ein gemeinsames Koordinatensystem des Gesamtsystems montiert sind. Für MODISSA wurde eine solche extrinsische Kalibrierung mit verschiedenen Verfahren für die unterschiedlichen Sensormodalitäten durchgeführt [Borgmann et al., 2021b; Diehm et al., 2020].

Für den zeitlichen Zusammenhang ist eine Synchronisierung der Datenaufzeichnung aller Sensoren auf eine gemeinsame Referenzzeit erforderlich. Auf MODISSA wird diese von dem INS bereitgestellt, welches sich wiederum per GNSS gegenüber der UTC-Zeit synchronisiert. Dies ist von Vorteil z.B. auch für Messkampagnen, bei denen mehrere verschiedene Sensorsysteme eingesetzt werden, da es so möglich ist auch zwischen diesen einen zeitlichen Zusammenhang herzustellen. Abhängig von den technischen Möglichkeiten der verwendeten Sensoren werden verschiedene Methoden für die Synchronisierung ihrer Datenaufzeichnung mit dieser gemeinsamen Referenzzeit verwendet. Dies ist in Abbildung 7.3 dargestellt und wird im Folgenden erläutert.

Die hier verwendeten LiDAR-Sensoren verfügen über eine eigene interne Uhr und betten deren Zeitstempel in ihren Datenstrom ein. Sie erlauben es diese interne Uhr mithilfe von Nachrichten im sog. NMEA-Format (National Marine Electronics Association) und einem PPS (Puls pro Sekunde) Signal mit einer externen Zeitquelle zu synchronisieren. Auf MODISSA wird dieser Mechanismus genutzt um die interne Uhr der LiDAR-Sensoren mit der Referenzzeit des INS zu synchronisieren.

Obwohl das INS sowohl ein PPS Signal als auch die Nachrichten im NMEA-Format direkt bereitstellen kann, ist aus technischen Gründen eine zusätzliche Konvertierung nötig, da das Ausgabeformat des INS nicht direkt mit dem erwarteten Eingabeformat der LiDAR-Sensoren kompatibel ist. Für diese Konvertierung wird ein FPGA (*Field Programmable Gate Array*) verwendet.

Mithilfe der Zeitstempel im Datenstrom der LiDAR-Sensoren ist es möglich, den genauen Zeitpunkt von jeder ihrer Messungen zu bestimmen. Wenn dieser Zeitstempel mit der Referenzzeit des INS synchronisiert ist, kann dann mithilfe des Datenstroms vom INS die genaue Pose des Sensorsystems in der Welt zum Zeitpunkt der Messung bestimmt werden. Dies wird bei der Erstellung von Punktwolken aus den LiDAR-Messungen genutzt, indem für jeden Punkt der Punktwolke die genaue Pose des LiDAR-Sensors in der Welt berücksichtigt wird. Hierdurch werden Verzerrungen in den resultierenden Punktwolken vermieden, die ansonsten aus der Eigenbewegung des LiDAR-Sensors während der Datenaufnahme entstehen würden. Bei diesem Prozess ist zu berücksichtigen, dass die LiDAR-Sensoren 1.300.000 Messungen (vordere Sensoren) pro Sekunde durchführen. Das INS liefert hingegen nur 200 Navigationslösungen pro Sekunde. Es wird daher eine lineare Interpolation verwendet, um für jede LiDAR-Messung eine Navigationslösung zu bestimmen. Die erstellten Punktwolken verwenden ein ENU (*East-North-Up*) Koordinatensystem mit einem im bzw. in der Nähe vom Messgebiet befindlichen Ursprung. Dieses erlaubt niedrige Werte für die Koordinaten und hat eine klar definierte Höhenachse (z-Achse). Um diese ENU-Koordinaten zu bestimmen, werden die in LLA (*latitude, longitude, altitude*) Koordinaten vorliegenden Navigationslösungen zunächst in ECEF (*Earth-Centered, Earth-Fixed*) Koordinaten projiziert. Anschließend können sie entsprechend des gewählten Ursprungs des ENU-Koordinatensystems in dieses transformiert werden.

Bei den Kameras wird ein anderer Mechanismus für die Synchronisation verwendet, da diese über keine interne Uhr und keine Möglichkeit verfügen direkt einen Zeitstempel in ihren Datenstrom einzubetten. Sie werden im sog. *Frame Trigger Modus* betrieben. Hierbei werden ihre Aufnahmen durch externe Signale ausgelöst. Diese Signale generiert ein weiteres FPGA. Sie gehen dann sowohl an die Kameras und lösen dort eine Bildaufnahme aus, als auch über einen sog. Eventeingang an das INS. Dieses erzeugt auf ein solches Signal hin einen Eventdatensatz, welcher einen Zeitstempel und eine Navigationslösung umfasst und über den Datenstrom des INS ausgegeben wird. Die Aufzeichnungscomputer für die Kameras erhalten über das Datennetzwerk eine Kopie des Datenstroms des INS und ordnen die darin enthaltenen Events den jeweiligen Kameraaufnahmen zu. Die den Events zugeordneten Daten, also der Zeitstempel und die Navigationslösung, werden den Kameraaufnahmen anschließend als Metadaten hinzugefügt. Bei diesem Prozess wird auch die zuvor ermittelte Auslöseverzögerung der Kameras berücksichtigt [Schatz, 2017].

7.1.3 Softwareumgebung von MODISSA

Das Multisensorsystem MODISSA verfügt über eine Softwareumgebung für die direkte Datenverarbeitung auf dem System. Diese Umgebung soll eine Abstraktionsschicht zwischen der Hardware und der forschungs- und anwendungsspezifischen Datenverarbeitung darstellen. Mit dem Ziel, dass auf die Daten unterschiedlicher Sensoren und auf unterschiedliche Hardwarekomponenten über standardisierte Schnittstellen und Datenformate zugegriffen werden kann. Hierdurch wird erreicht, dass für die Nutzung des Fahrzeugs keine detaillierte Kenntnis der dort verbauten Hardware erforderlich ist und dass Änderungen an dieser Hardware keine oder nur geringe Änderungen an der Software erforderlich machen.

Die Softwareumgebung ist auf Basis des *Robot Operating System* (ROS)* aufgebaut. Hierbei handelt es sich um ein Open-Source Software-Framework, welches ursprünglich für die Verwen-

*www.ros.org

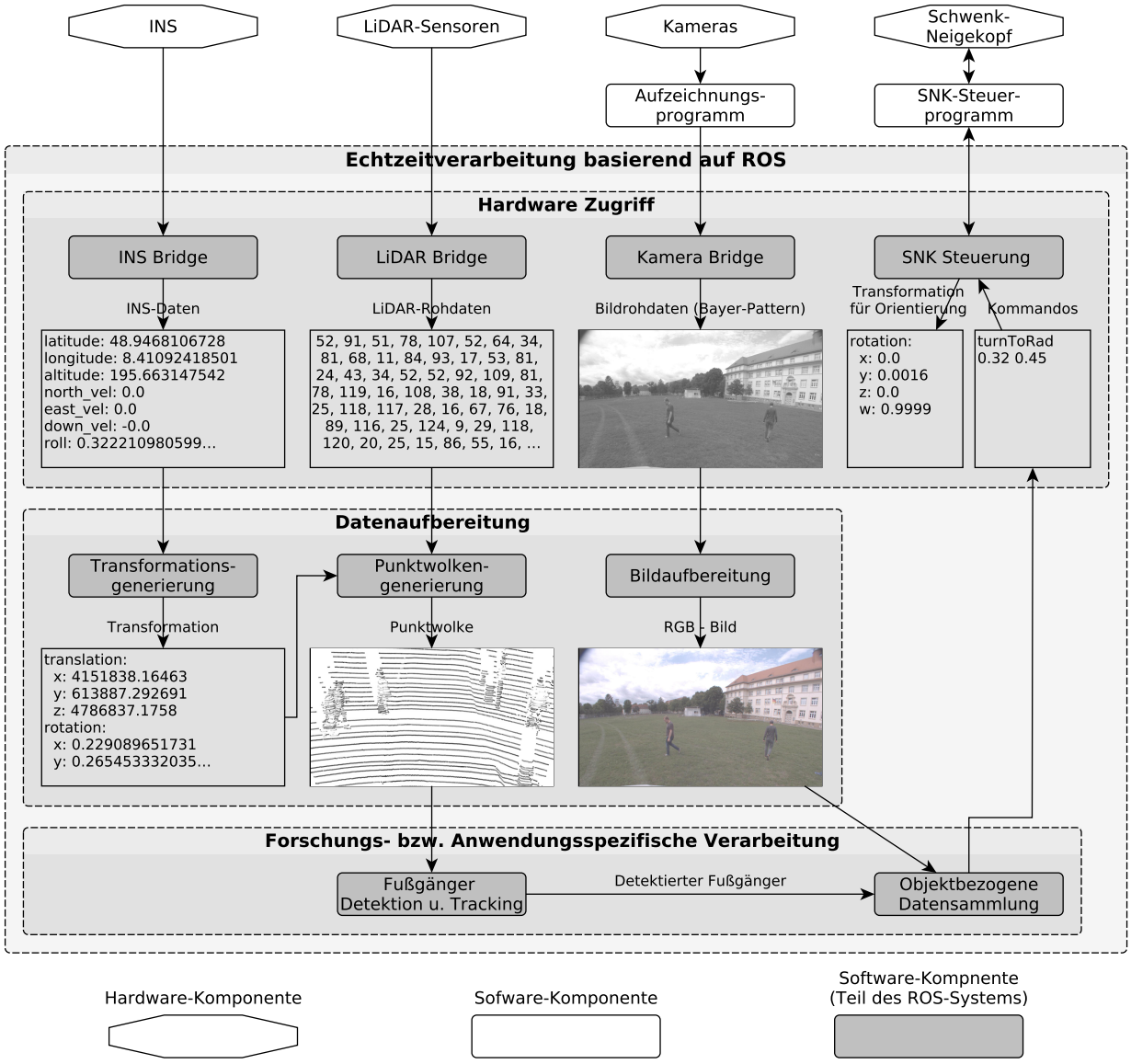


Abbildung 7.4: Die Softwareumgebung des Multisensorfahrzeugs MODISSA für die Datenverarbeitung im Fahrzeug

dung im Zusammenhang mit Robotern entwickelt wurde. Es hat aber viele Eigenschaften, welche es auch für die Verwendung in anderen Arten von Multisensorsystemen nützlich machen. ROS bietet Möglichkeiten für den Datenaustausch zwischen verschiedenen Softwareprozessen und unterstützt dabei eine Reihe von unterschiedlichen Programmiersprachen. Dieser Datenaustausch erfolgt über standardisierte, einfach zu nutzende Schnittstellen und ist für die verwendete Software weitestgehend transparent. Er kann sowohl lokal auf einem Computer als auch über mehrere mit einem Netzwerk verbundene Computer stattfinden. Für den Datenaustausch werden definierte Datentypen genutzt. Viele dieser Datentypen bringt ROS bereits standardmäßig mit und deckt damit z.B. den Austausch der Daten verschiedener Sensortypen ab. Es können jedoch bei Bedarf auch zusätzliche Datentypen definiert werden. ROS stellt außerdem einen Transformationsbaum bereit, der sich mit Informationen über die verschiedenen extrinsischen Transformationen zwischen den Komponenten eines Multisensorsystems füllen lässt. Auf diesen kann von überall im ROS-System zugegriffen werden. Ein ROS-System besteht üblicherweise aus einer Reihe von speziellen

Programmen, die im Kontext von ROS auch als *Nodes* bezeichnet werden. Für MODISSA wurden im Rahmen dieser Arbeit mehrere solcher Nodes entwickelt, um die im Folgenden erläuterte Softwareumgebung zu schaffen.

Abbildung 7.4 gibt einen Überblick über das auf MODISSA verwendete Softwaresystem. Dieses lässt sich in drei Schichten aufteilen, die ebenfalls dargestellt sind. Die erste Schicht innerhalb des ROS-Systems ist für die Interaktion mit der Hardware des Fahrzeugs zuständig. Sie ist die einzige, die mit Komponenten außerhalb des ROS-Systems interagiert und setzt sich aus Programmen zusammen, die spezifisch für diese Interaktion entwickelt wurden. Es wurde dabei Wert darauf gelegt, dass die reine Datenaufzeichnung und die direkte Datenverarbeitung auf dem Sensorsystem parallel betrieben werden können. Dies hat je nach Sensorart bestimmte Maßnahmen erfordert. Sowohl bei dem INS als auch bei den LiDAR-Sensoren kann das ROS-System direkt auf diese Sensoren zugreifen, da diese in der Lage sind ihre Daten parallel an dieses System sowie an eine gleichzeitig ablaufenden Datenaufzeichnung zu liefern. Bei den Kameras ist dies hingegen nicht möglich. Hier interagiert zunächst nur die unabhängig von ROS betriebene Software zur Datenaufzeichnung mit den Sensoren. Diese übernimmt auch die in Abschnitt 7.1.2 beschriebene Ergänzung ihrer Daten mit einem Zeitstempel, welcher mit der gemeinsamen Referenzzeit synchronisiert ist. Das Aufzeichnungsprogramm reicht die Daten dann per Netzwerkverbindung an das ROS-System weiter. Ein ähnliches Vorgehen wird im Fall der Ansteuerung des SNK verwendet, wobei hier eine Kommunikation in beide Richtungen stattfindet, um einerseits Informationen über die aktuelle Ausrichtung des SNK zu erhalten und diesen andererseits anzusteuern um diese Ausrichtung zu ändern.

Die Hardware-Zugriffsschicht stellt die Daten der Sensoren und sonstigen Hardware des Sensorsystems dem restlichen ROS-System als weitestgehend unverarbeitete Rohdaten zur Verfügung. Diese sind also oft noch nicht in einer Form, die für eine weitere Verwendung gut geeignet ist. Es gibt daher eine Datenaufbereitungsschicht, welche diese Rohdaten in eine besser nutzbare Form umwandelt. Im Fall der LiDAR-Daten werden hier z.B. aus den Entfernungsmessungen der Sensoren Punktwolken generiert. Wobei auch wie in Abschnitt 7.1.2 beschrieben die Eigenpose des Sensorsystems berücksichtigt wird. Für die Kameras findet in dieser Schicht die Umwandlung vom Bayer-Pattern-Format ins RGB-Format statt. Auch die Daten des INS werden aufbereitet, um mit ihnen den Transformationsbaum zu aktualisieren, welcher im Fall von MODISSA die Transformationen zwischen allen Komponenten des Sensorsystems sowie zwischen dem Sensorsystem und einem ENU- sowie ECEF-Weltkoordinatensystem enthält.

Die eigentliche forschungs- bzw. anwendungsspezifische Datenverarbeitung findet in der dritten Schicht statt. Die dort angesiedelten Nodes nutzen die Daten aus den vorherigen Schichten. Nodes in dieser Schicht tauschen jedoch ggf. auch untereinander Daten aus. Für die Zwecke dieser Arbeit sind dort zwei Nodes angesiedelt. Einer führt die Detektion sowie das Tracking von Fußgängern in den LiDAR-Daten durch. Der andere umfasst die bildbasierte Körperposenschätzung.

7.1.4 Geodatenbank für die Organisation der Auswertungsergebnisse

Für ein System, wie es in Abschnitt 1.1 beschrieben wurde, ist nicht nur die Datenerfassung und -auswertung relevant, sondern auch die langfristige Speicherung der Ergebnisse dieser Auswertung in einer Form, die eine effektive weitere Nutzung erlaubt. Für die Untersuchungen auf dem MODISSA-System wurde eine entsprechende Lösung beispielhaft implementiert. Hierbei wurden mehrere Grundüberlegungen berücksichtigt: Es sollte möglich sein, die aus der Datenauswertung resultierenden objekt- bzw. fußgängerbezogenen Merkmale und Informationen abzulegen, sowie die mit diesen Fußgängern verknüpften Sensordaten. Im Hinblick auf eine breitere Nutzung des Systems ist es zudem wünschenswert, auch eine allgemeine Ablage und Verwaltung der erfassten Sensordaten zu ermöglichen, auch wenn eine solche für den Fokus dieser Arbeit keine besondere

Relevanz hat. Für einige der beschriebenen Einsatzszenarien z.B. im Bereich der Unfallforensik ist es erforderlich, in dieser Datenhaltung orts- aber auch zeitbezogen nach Daten zu suchen und auf diese zuzugreifen.

Die beschriebenen Anforderungen lassen sich technisch gut mit einer Geodatenbank erfüllen. Diese unterscheiden sich von klassischen Datenbanken, wie sie in der Informationsverarbeitung seit langem üblich sind, darin dass sie über spezielle räumliche Daten-, Index- und Abfragetypen verfügen. Diese sind insbesondere für den ortsbezogenen Datenabruf erforderlich, da Datenbanken Daten effizient mithilfe von Indizes suchen und abrufen. Damit dies auch für räumliche Daten und damit für den Datenabruf anhand einer räumlichen Koordinate funktioniert, braucht die Datenbank spezielle Daten- und dazugehörige Indextypen für solche Daten. Normale Zahlendatentypen würden zwar auch eine Ablage von Koordinaten erlauben, können aber nicht effizient für den Abruf von Daten anhand dieser Koordinaten und Umkreissuchen genutzt werden. Ein Beispiel für einen solche Suche ist: „Gib mir alle Daten über Fußgänger im Umkreis von 100 m um N49.013504° E8.404412°“.

Auswahl des verwendeten Datenbankmanagementsystems

Die Software, die für die Realisierung einer Datenbank genutzt wird, wird als Datenbankmanagementsystem (DBMS) bezeichnet und umfasst u.a. Komponenten für die Erzeugung der eigentlichen Datenbank, den Zugriff auf diese, für die Datendefinition, zur Sicherstellung der Erfüllung definierter Regeln zur Datenkonsistenz und die Rechtverwaltung. Es gibt mehrere existierende DBMS, die für eine Geodatenbank entsprechend der oben genannten Anforderungen genutzt werden können. Es wurde daher eine Auswahl getroffen, bei der verschiedene relationale DBMS: Oracle, IBM-DB2, SQLite mit SpetiaLite und PostgreSQL mit PostGIS, aber auch sog. NoSQL DBMS: MongoDB, BigTable, Cassandra und CouchDB in Betracht gezogen wurden.

NoSQL-Datenbanken sind als Alternative zu den schon älteren relationalen Datenbanken entstanden und haben in den letzten Jahren eine große Verbreitung erfahren. Der Begriff NoSQL umfasst dabei eine ganze Reihe unterschiedlicher Konzepte zur Realisierung einer Datenbank. Relationale Datenbanken verwalten Daten in Relationen, welche als Tabellen dargestellt werden. Sie verwenden eine definierte relationale Algebra, um auf diese Daten zuzugreifen und sie zu manipulieren. NoSQL-Datenbanken verwenden hingegen z.B. ein dokumentenorientiertes (MongoDB, CouchDB), ein auf Schlüsselwert Paaren basierendes (BigTable) oder ein spaltenorientiertes (Cassandara, BigTable) Konzept, wobei es möglich ist, dass eine NoSQL-Datenbank in mehrere Kategorien fällt. NoSQL-Datenbanken haben Stärken im Bereich der Skalierung über mehrere Datenbankserver und sind außerdem je nach konkreter Ausprägung besser dazu geeignet, mit weniger stark strukturierten Daten umzugehen: Relationale Datenbanken definieren und benötigen eine klare Struktur der Daten, die in ihnen abgelegt werden sollen. Dies ist bei vielen NoSQL-Datenbanken nicht unbedingt erforderlich. Sie können aber Nachteile haben, wenn solche klaren Strukturen der Daten und ihrer Beziehungen zueinander vorhanden sind. Diese lassen sich dann gut in relationalen Datenbanken abbilden. NoSQL-Datenbanken setzen außerdem einen weniger starken Fokus auf eine permanent vorhandene Datenkonsistenz. Diese spielt bei relationalen Datenbanken eine besondere Rolle, was sich u.a. darin ausdrückt, dass Transaktionen auf einer relationalen Datenbank die sog. ACID (*atomicity, cosistency, isolation, durability*) Eigenschaften erfüllen. Da NoSQL Datenbanken anders als relationale Datenbanken meist nicht auf SQL (*Structured Query Language*) als standardisierte Abfragesprache setzen und auch sonst weniger standardisiert sind, lassen sie sich auch nicht so gut wie relationale Datenbanken gegeneinander austauschen. Für die genannten Anforderungen in dieser Arbeit sind die Stärken von NoSQL-Datenbanken nicht von besonderer Bedeutung. Es wurde daher auf ein relationales DBMS gesetzt.

Der Funktionsumfang der betrachteten relationalen DBMS mit Geo-Funktionalität ist zumindest im Hinblick auf die genannten Anforderungen zu großen Teilen vergleichbar. Ausschlaggebend waren daher auch Faktoren wie die benötigten Lizenzen und den damit verbundenen Kosten. Hier haben sowohl PostgreSQL mit PostGIS als auch SQLite mit SpetiaLite Vorteile, da es sich bei beiden um frei verfügbare Open-Source Software handelt. Im direkten Vergleich ist dann die Entscheidung für PostgreSQL mit PostGIS gefallen. Dieses unterstützt nicht nur geometrische Datentypen mit einer Entfernungsberechnung, die von einem kartesischen Koordinatensystem ausgeht, sondern auch geographische Datentypen, welche ein ellipsoidisches Koordinatensystem basierend auf dem WGS84-Ellipsoiden verwenden.

Modell der Datenbank

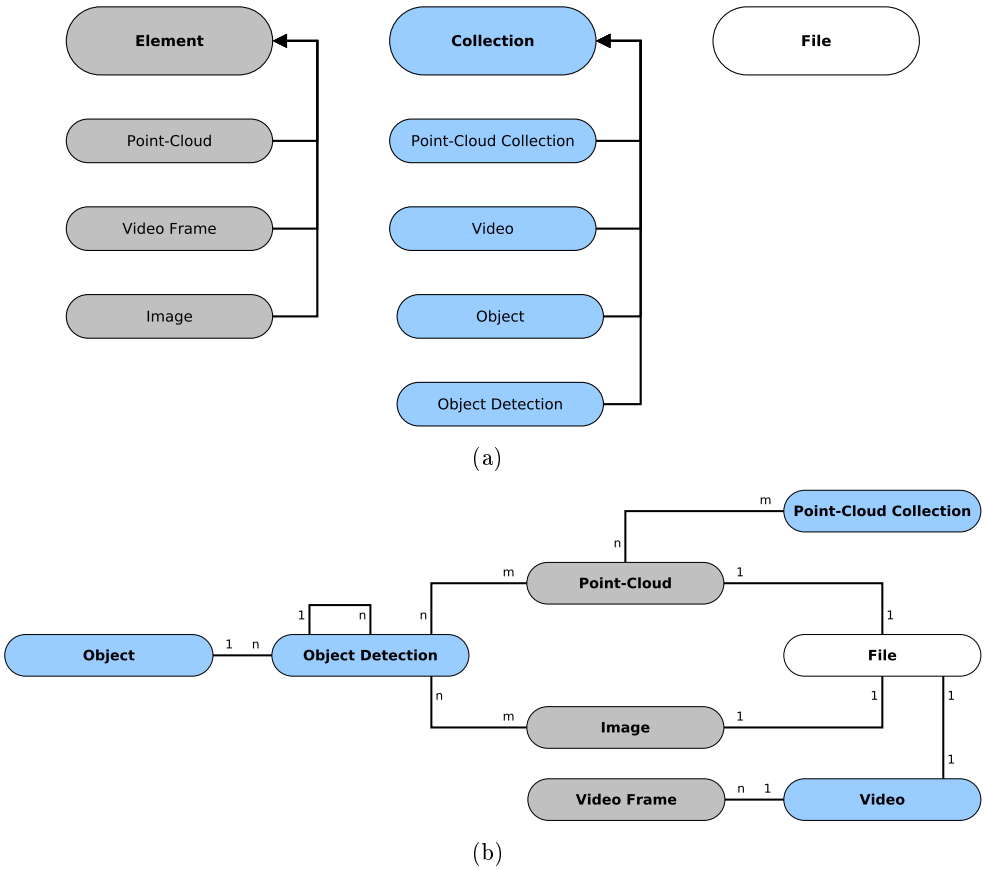


Abbildung 7.5: Modell der Datenbank zur Ablage der Auswertungsergebnisse und von Sensorrohdaten. a) Logische Übersicht der verwalteten Datentypen. Diese sind in die Gruppen Element, Collection und File eingeteilt. Davon sind Spezialisierungen abgeleitet, die gegenüber dem Grundtyp über zusätzliche Attribute verfügen. Diese Grundtypen sind in beiden Grafiken auch farblich hervorgehoben. b) Die Beziehungen der verschiedenen Datentypen untereinander und deren Kardinalitäten. Die Grundtypen werden hier nicht dargestellt.

Neben der Auswahl eines DBMS ist es auch erforderlich ein Modell bzw. eine Struktur zu entwerfen, mit der die Daten entsprechend der Anforderungen verwaltet und abgelegt werden können. Das Ergebnis hiervon ist in Abbildung 7.5 dargestellt. Es wird zwischen drei Grunddatentypen unterschieden: Element, Sammlung (engl. *Collection*) und Dateien (engl. *File*). Dateien stellen dabei eine Besonderheit dar. Sie repräsentieren tatsächliche Dateien die im Dateisystem abgelegt sind und sind verschiedenen anderen Datentypen zugeordnet. Dies wird z.B. für die Verwaltung von

Sensorrohdaten verwendet, die letztlich als Datei vorliegen. Die Datenbank speichert hier primär die Metadaten dieser Sensordaten und erlaubt es, sie anhand dieser Metadaten zu suchen.

Bei Elementen handelt es sich um atomare nicht mehr weiter unterteilte Einträge in der Datenbank. Von ihnen gibt es verschiedene Typen für Punktwolken, einzelnen Frames eines Videos und einzelnen Bildern. Sammlungen werden genutzt um mehrere Elemente oder auch andere Sammlungen zu gruppieren. Bei ihnen gibt es zum einen eine generische Sammlung von Punktwolken, die dazu genutzt wird die Daten einer Messkampagne zusammenzufassen. Videos wiederum setzen sich aus einzelnen Frames zusammen. Objekte und Objektdetektionen werden im Kontext dieser Arbeit genutzt, um die Auswertungsergebnisse in der Datenbank abzulegen. Eine Sammlung vom Typ Objekt fasst dabei alles zusammen, was einer bestimmten Objektinstanz zugeordnet werden kann. Beispielsweise ein Fußgänger, der im Zeitverlauf in mehreren Punktwolken detektiert und getrackt wird und von dem ggf. in diesem Zeitraum mehrere Kameraaufnahmen existieren. Objektdetektionen repräsentieren die konkrete Position und Detektion eines Objektes zu einem bestimmten Zeitpunkt. Diesen Detektionen können wiederum dazu passende Sensoraufnahmen wie Bildausschnitte oder Punktwolken zugeordnet werden. Also beispielsweise die Punktwolke, in der der Fußgänger detektiert wurde. Objektdetektionen können zusätzlich aber auch andere Detektionen enthalten. Dieser Mechanismus wird genutzt um die Schlüsselpunkte der Körperpose eines Fußgängers abzulegen. Aus Sicht der Datenbank gibt es also keinen Unterschied zwischen der Detektion eines Fußgängers und der Detektion eines Körperteils dieses Fußgängers. Beides wird in der Datenbank als Objektdetektion abgelegt und die Detektion des Körperteils ist der Detektion der Person hierarchisch zugeordnet.

Software-API zum Zugriff auf die Datenbank

Um die Nutzung der Datenbank in verschiedenen Software-Programmen zu erleichtern sowie um zukünftige Strukturänderungen der Datenbank und auch einen Austausch des verwendeten DBMS zu ermöglichen ohne diese Programme anpassen zu müssen, wurde eine Software-API für den Zugriff auf die Datenbank implementiert. Diese spiegelt die beschriebene Struktur von Elementen, Sammlungen und Dateien mit ihren jeweiligen Spezialisierungen wieder und setzt diese als eine Objekthierarchie um. Sie werden in der API durch sog. Deskriptoren repräsentiert. Diese können erzeugt und verändert werden. Neben den Deskriptoren gibt es eine zentrale Klasse um mit der Datenbank zu interagieren. Diese kann über entsprechende Methoden die zu den Deskriptoren gehörenden Datensätze in der Datenbank erzeugen, ändern und löschen und bietet eine Reihe von Methoden an, um Datensätze in der Datenbank anhand verschiedener Kriterien zu finden und abzurufen. Aus technischer Sicht werden hierbei jeweils entsprechende SQL-Kommandos erzeugt und auf der Datenbank ausgeführt. Bei einer Datenabfrage werden entsprechend des Ergebnis, welches die Datenbank ausgibt, Deskriptoren angelegt und an die aufrufende Methode zurückgegeben.

7.2 Aufgenommene Testszenen

Für die Experimente werden eine Gruppe von Datensätzen genutzt, die mit dem Multisensorsystem MODISSA aufgezeichnet wurden. Es wird dabei zwischen Daten unterschieden, die zum trainieren des neuronalen Netzes verwendet werden und denen, die für die eigentlichen Experimente genutzt werden. Zwischen diesen beiden Gruppen gibt es keine Überlappungen. Die Daten stammen dabei aus zwei separaten Messfahrten. Die erste wurde im Stadtgebiet von Ettlingen durchgeführt. Sequenzen aus dieser Messfahrt werden sowohl für das Training als auch für die Experimente verwendet. Die zweite Messfahrt wurde auf dem Institutsgelände des Fraunhofer IOSB in Ettlingen durchgeführt. Sie umfasst gestellte Szenen, in denen sich mehrere Personen gezielt im

Umfeld des Sensorsystems befunden haben. Daten dieser Messfahrt werden nur für das Training, nicht jedoch für die eigentlichen Experimente genutzt.

Während der Messfahrten wurden die LiDAR-Sensoren des Fahrzeugs mit 10 Umdrehungen pro Sekunde betrieben. Für die Experimente werden daher die in 0,1 s, also während einer Umdrehung gemessenen Daten als *eine Punktwolke* angesehen. Dies kann auch als eine Abstastung oder ein Scan der Umgebung bezeichnet werden. Da es sich bei Personen um bewegte Objekte handelt, wird die Kombination der Daten aus mehreren Umdrehungen in eine gemeinsame Punktwolke nicht als zielführend angesehen. Aus der Datenrate der Sensoren ergibt sich, dass eine solche Umdrehung ca 130.000 Messungen umfasst. Üblicherweise führt ein Teil dieser Messungen zu keinem Ergebnis, da beispielsweise in eine Richtung gemessen wird, in der sich keine Oberfläche befindet. Die resultierenden Punktwolken umfassen daher, abhängig von der Umgebung, üblicherweise zwischen 85.000 und 95.000 tatsächliche 3D-Punkte. Diese bewahren neben ihren Punktkoordinaten auch die Koordinaten des Sensors während der Messung als Zusatzinformation. Alle Koordinaten aller Punkte in allen Punktwolken verwenden dabei dasselbe Koordinatensystem. Bei diesem handelt es sich um ein ENU-Koordinatensystem, dessen Ursprung in der Nähe des Messgebiets liegt. Für die bildbasierten Auswertungen werden zusätzlich Bilddaten verwendet, die mit den Rundumkameras des Sensorsystems aufgezeichnet wurden. Diese Aufnahmen erfolgten mit je 10 Bildern pro Sekunde pro Kamera.

Objekte von Interesse in den verwendeten Punktwolken wurden interaktiv, manuell von einem Menschen annotiert. Diese Annotationen umfassen eine Information darüber, welche Punkte der Punktwolke zu dem entsprechenden Objekt gehören. Hieraus wurde zusätzlich ein Objektmittelpunkt als Durchschnittskoordinate der zum Objekt gehörenden Punkte sowie eine Bounding Box erzeugt. Es wird zwischen vollständig sichtbaren Personen, Personen die teilweise verdeckt sind und vorübergehend vollständig verdeckten Personen unterschieden. Zusätzlich sind Radfahrer mit derselben Unterscheidung im Hinblick auf Verdeckungen annotiert. Für die Experimente im Bezug auf das Tracking verfügen die annotierten Objekte über IDs, welche über die gesamte Sequenz konsistent sind. Für die Experimente, bei denen auch die Kameras verwendet werden, wurde außerdem überprüft, ob sich eine Person in den Punktwolken auch im Sichtbereich der Rundumkameras des Fahrzeugs befindet oder nicht. Es hat sich jedoch herausgestellt, dass die Sichtbereiche der LiDAR-Sensoren und Kameras im Bodenbereich so weit übereinstimmend sind, dass sich alle in den LiDAR-Daten sichtbaren Personen auch im Sichtbereich der Kameras befinden.

Die Überprüfung, ob die Objekte auch im Sichtbereich der Kameras sind, das Erstellen der IDs für das Tracking und die Annotierung von vollständig verdeckten Objekten, wurden nicht für die Daten durchgeführt, die ausschließlich für das Training des Verfahrens zur Personendetektion verwendet werden, da diese Informationen hierfür nicht benötigt werden.

7.2.1 Daten für das Training

Die für das Training verwendeten Daten umfassen eine Reihe von kurzen Sequenzen aufeinanderfolgender Punktwolken, die jeweils unterschiedliche Länge haben. Sie stammen aus beiden Messfahrten. Es wird zwischen den Daten für das eigentliche Training und denen für die Validierung des Trainingsfortschritts unterschieden. Letztere werden nicht selbst beim Training genutzt, sondern dafür zu entscheiden, wann die aktuelle Trainingsphase beendet werden sollte (vgl. Abschnitt 4.3.3). Die Eigenschaften und Zusammensetzung der genutzten Daten ist in Tabelle 7.1 dargestellt.

7.2.2 Daten für die Experimente

Für die Experimente wurden drei längere Sequenzen (30 s bzw. 40 s) genutzt. Sie stammen alle aus der Messfahrt im Ettlinger Stadtgebiet. Die erste Sequenz wurde vor, an und nach einer belebten

Anzahl	Trainingsdaten	Validierungsdaten
Punktwolken	1300	226
Vollständig sichtbare Personen	2572	597
Teilweise verdeckte Personen	1424	689
Vollständig sichtbare Radfahrer	24	104
Teilweise verdeckte Radfahrer	121	26

Tabelle 7.1: Zusammensetzung der Trainings- und Validierungsdaten

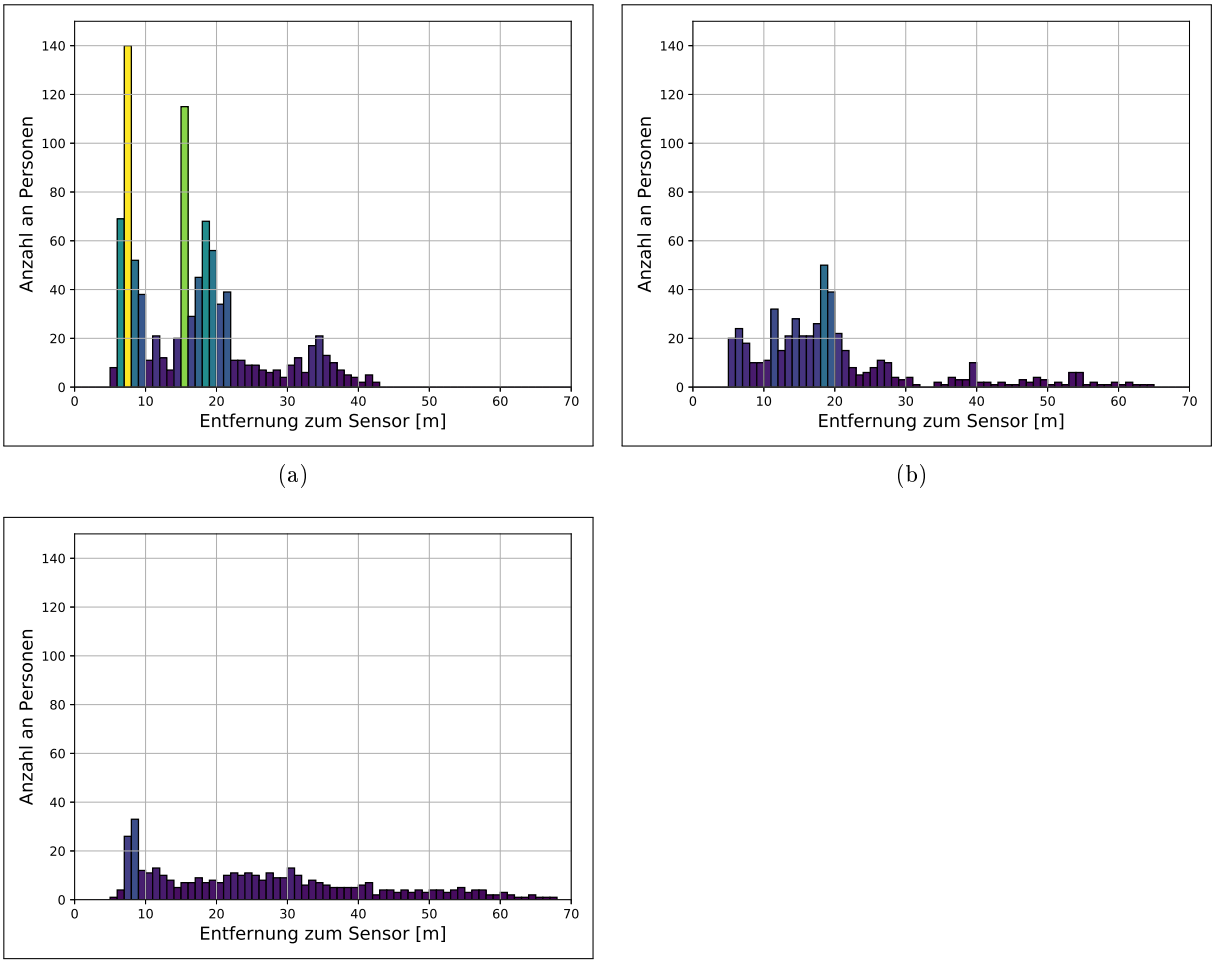


Abbildung 7.6: MODISSA-Datensätze: Histogramme der Entfernungen zwischen vollständig sichtbaren Personen und Sensor. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3

Kreuzung mit vielen Fußgängern aufgenommen, welche teilweise die Straße überqueren. Das Sensorfahrzeug hat hierbei zunächst an der Kreuzung gewartet und diese dann geradeaus durchquert. Die zweite Sequenz wurde an einer anderen Kreuzung mit weniger Fußgängern aufgenommen. Hier ist das Fahrzeug abgebogen. Die dritte Sequenz umfasst erst eine längere Fahrt geradeaus, in der mehrere Fußgänger passiert werden. Anschließend durchfährt das Fahrzeug einen Kreisel, wo sich ebenfalls Passanten auf dem Bürgersteig befinden. Eine Besonderheit der dritten Sequenz ist, dass

es sich in dieser bei einigen der Passanten um junge Kinder handelt, die deutlich kleiner sind als Erwachsene. Die Daten haben die in Tabelle 7.2 dargestellte Zusammensetzung.

Anzahl	**Ettlingen 1**	**Ettlingen 2**	**Ettlingen 3**
Punktwolken	300	300	400
Vollständig sichtbare Personen	942	516	414
Teilweise verdeckte Personen	1105	820	1187
Vorübergehend verdeckte Personen	51	118	77
Vollständig sichtbare Radfahrer	307	57	0
Teilweise verdeckte Radfahrer	790	232	58
Vorübergehend verdeckte Radfahrer	63	60	8

Tabelle 7.2: Zusammensetzung der Sequenzen Ettlingen 1, 2 und 3

Abbildung 7.6 zeigt Histogramme für die Entfernung zwischen den in den Sequenzen vorhandenen vollständig sichtbaren Personen und dem LiDAR-Sensor. Diese sind dabei insbesondere im ersten Datensatz überwiegend weniger als 20 m vom Sensor entfernt. Dies ist auf das dort vorhandene Warten des Sensorfahrzeugs an einer Kreuzung zurückzuführen. Während dieser Zeit haben mehrere Passanten vor dem Fahrzeug die Straße überquert. Abbildung 7.7 zeigt Ausschnitte aus der Grundwahrheit der drei Datensätze.

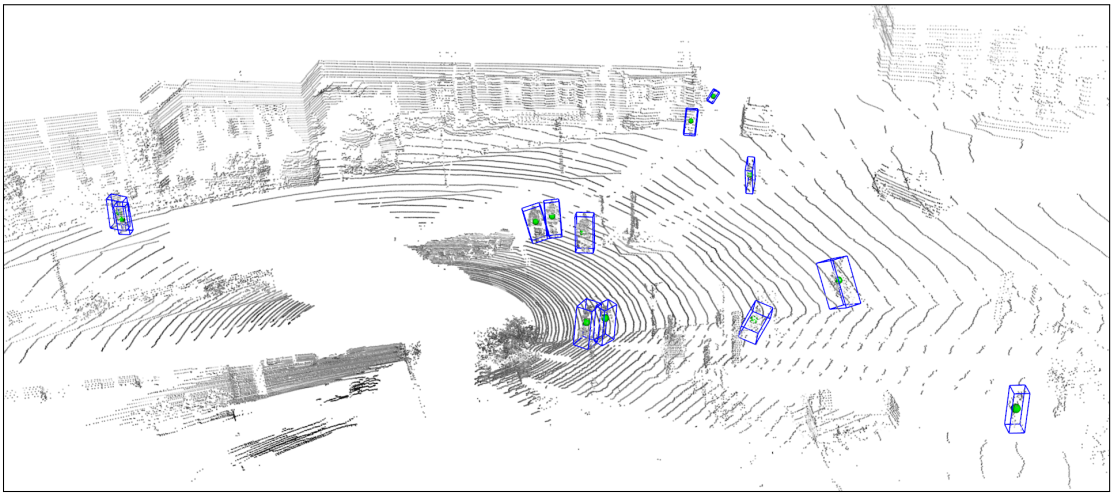
7.3 Vorgehen beim Bewerten der Ergebnisse

In diesem Abschnitt wird erläutert, wie die Ergebnisse der untersuchten Verfahren bewertet werden und wie sich die dafür verwendeten Kennzahlen zusammensetzen. Dabei wird zwischen der Bewertung einer Detektionsleistung und einer Trackingleistung unterschieden.

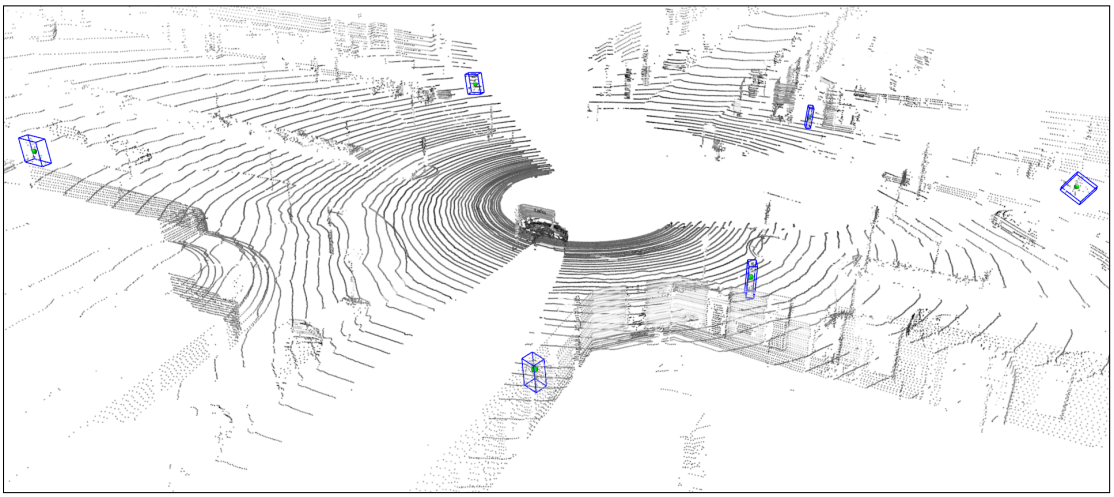
7.3.1 Detektionsleistung

Um die Ergebnisse der Personendetektion zu bewerten, werden diese mit den Daten der Grundwahrheit verglichen. Hierbei wird untersucht, welche Detektionen in den Ergebnissen einem entsprechenden Objekt in der Grundwahrheit zugeordnet werden können. Ob eine solche Zuordnung möglich ist, wird basierend auf der Distanz zwischen Detektion und Objekt in der Grundwahrheit entschieden. Sie gilt als möglich, wenn diese Distanz $<= 0{,}5\,\text{m}$ ist. Sollte eine Detektion entsprechend diesem Kriterium mehreren Objekten in der Grundwahrheit zugeordnet werden können, wird das Objekt ausgewählt, bei dem die Distanz am geringsten ist. Detektionen, die zugeordnet werden können, werden als *richtig positiv* gezählt. Detektionen, die nicht zugeordnet werden können, als *falsch positiv*. Zusätzlich wird betrachtet, welchen Objekten in der Grundwahrheit keine Detektion zugeordnet werden, diese gelten als *falsch negativ*.

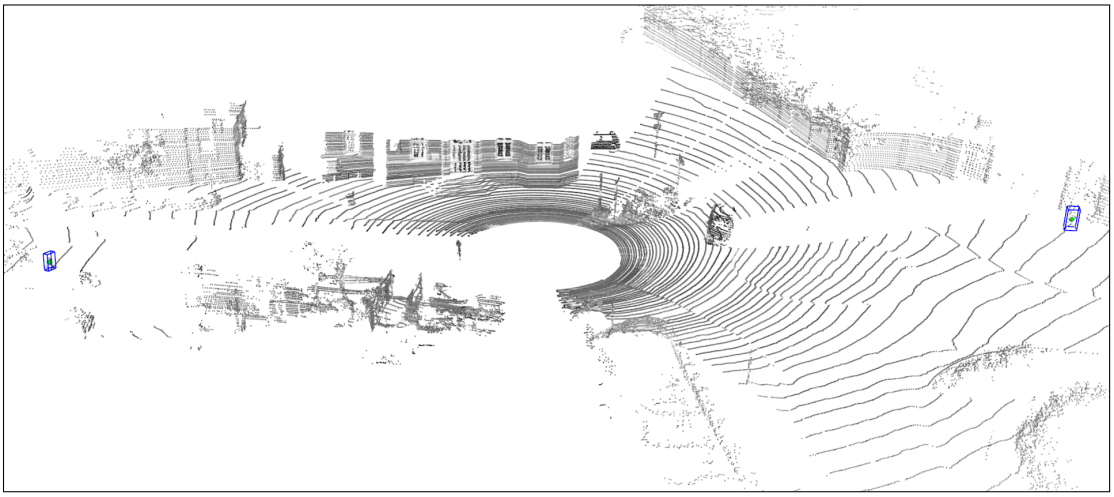
Für die Datensätze des MODISSA-Systems war eine Besonderheit im Umgang mit Radfahrern notwendig. Diese sind nicht in ausreichender Menge in den Trainingsdaten vorhanden, um das neuronale Netz sinnvoll mit ihnen zu trainieren. Da es sich bei ihnen jedoch im Grunde um Fußgänger auf einem Fahrrad handelt, werden sie ohne entsprechendes Training oft als Fußgänger erkannt. Um ihren Einfluss auf die Ergebnisse zu eliminieren, wurde daher folgendes Vorgehen gewählt: Radfahrer, die als Fußgänger detektiert werden, zählen weder als richtig positiv noch als falsch positiv. Radfahrer in der Grundwahrheit werden dementsprechend auch nicht als falsch negativ gezählt. Je nach Untersuchung wurde ein ähnliches Vorgehen für stark oder vollständig verdeckte Objekte gewählt. Bei Untersuchungen, in denen es nicht speziell um die Leistung des



(a)



(b)



(c)

Abbildung 7.7: Beispiele der MODISSA-Datensätze. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3

Verfahrens im Hinblick auf Verdeckungen ging, wurden diese ebenfalls nicht als richtig positiv, falsch positiv oder falsch negativ gezählt.

Basierend auf richtig positiv (rp), falsch positiv (fp) und falsch negativ fn werden *Genauigkeit* und *Sensitivität* mit folgenden Formeln errechnet:

$$Genauigkeit = \frac{rp}{rp + fp} \tag{7.1}$$

$$Sensitivität = \frac{rp}{rp + fn} \tag{7.2}$$

Durch verschiedene Schwellwerte für die Detektion können diese beiden Werte gegeneinander verschoben werden. Ein niedriger Schwellwert für die Detektion führt zu vielen falsch positiven Ergebnissen. Hierdurch sinkt die Genauigkeit. Gleichzeitig steigt jedoch die Sensitivität, da die Menge an falsch negativen Ergebnissen ebenfalls reduziert wird. Wenn verschiedene Schwellwerte durchprobiert werden, ergibt sich für die beiden Werte daher eine Kurve, die dargestellt werden kann. Ebenfalls lässt sich die Fläche unter dieser Kurve bestimmen, die möglichst maximiert werden sollte.

7.3.2 Trackingleistung

Um die Leistung des Trackingverfahrens zu bewerten, wurde auf die von Bernardin & Stiefelhagen [2008] vorgestellten Kennzahlen *multiple object tracking precision* ($MOTP$) und *multiple object tracking accuracy* ($MOTA$) zurückgegriffen. Diese beiden Kennzahlen wurden entworfen, um die Leistung eines Verfahrens in einem Szenario mit mehreren zu trackenden Objekten zu bewerten.

Die Ermittlung dieser Kennzahlen setzt eine Zuordnungsprozedur voraus, in der Paare aus Objekten in den Ergebnissen des Verfahrens und der Grundwahrheit gebildet werden. Hierfür wird ein Entfernungsschwellwert definiert, bis zu dem eine solche Zuordnung als valide gilt. Für die Zwecke dieser Arbeit wurde dieser Schwellwert auf 50 cm festgelegt. Der Zuordnungsprozess erfolgt für jeden Frame bzw. Aufnahmezeitpunkt der Daten. Es werden Zuordnungen bevorzugt, die nicht denen aus dem vorherigen Aufnahmezeitpunkt widersprechen, sofern diese ohne Verletzung des Entfernungsschwellwerts weiterhin möglich sind. So nicht zugeordnete Objekte werden mithilfe der Ungarischen Methode [Kuhn, 1955; Munkres, 1957] in einer Art und Weise zugeordnet, die die Gesamtdistanz zwischen Objekten der Grundwahrheit und den Ergebnissen des Verfahrens minimiert, wobei ebenfalls wieder der definierte Schwellwert berücksichtigt wird. Fälle in denen sich die Zuordnung gegenüber dem vorherigen Aufnahmezeitpunkt geändert hat, werden als *Mismatches* bezeichnet. Objekte der Grundwahrheit, die nicht zugeordnet werden konnten, als *Misses* und Objekte der Verarbeitungsergebnisse, die nicht zugeordnet werden konnten, als *false positives*.

$MOTP$ legt dabei den Fokus auf die Genauigkeit der Objektpositionen und vergleicht diese mit den Objekten in den Daten der Grundwahrheit. Die Kennzahl gibt die durchschnittliche Abweichung dieser Position für alle gebildeten Zuordnungen wieder. Sie ist wie folgt definiert:

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{7.3}$$

wo d_t^i = Distanz zwischen den Objekten des i-ten Zuordnungspaares zum Zeitpunkt t
c_t = Anzahl an gebildeten Zuordnungspaaren zum Zeitpunkt t

$MOTA$ beschreibt Fehler des Trackingverfahrens bei der Assoziation von Track zu Objekt sowie falsch negative und falsch positive Ergebnisse und setzt diese ins Verhältnis zu der Anzahl

von Objekten in den Daten der Grundwahrheit. Sie leitet sich also aus der Falsch-positiv-Rate, Falsch-negativ-Rate (*Miss*) und *Mismatch*-Rate ab. Die Kennzahl ist wie folgt definiert:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{7.4}$$

wo m_t = Misses (falsch negative) zum Zeitpunkt t
fp_t = falsch positive zum Zeitpunkt t
mme_t = Mismatches (Zuordnungswechsel) zum Zeitpunkt t
g_t = Anzahl an Objekten in der Grundwahrheit zum Zeitpunkt t

Anders als bei der Bewertung der Dektionsleistung werden bei der Bewertung der Trackingsleistung, falls nicht anders angegeben, auch die Personen berücksichtigt, die teilweise oder vollständig verdeckt sind. Im Bezug auf die Radfahrer wurde jedoch ein ähnliches Vorgehen wie bei der Detektion gewählt und diese wurden bei der Bestimmung der Kennzahlen nicht mitgezählt.

7.3.3 Laufzeit

Bei einigen der durchgeführten Untersuchungen wird auch die Laufzeit der Verfahren bestimmt und verglichen. Da die absolute Laufzeit immer stark von der eingesetzten Hardware abhängt, wird hierbei auf einen Vergleich der relativen Laufzeit unterschiedlicher Varianten bzw. Konfigurationen der Verfahren gesetzt. Da für alle Experimente dieselbe Hardware verwendet wurde, ist deren Einfluss bei einem solchen Vergleich minimiert. Die Laufzeit wird direkt anhand entsprechender Logausgaben in den Programmen gemessen, die die untersuchten Verfahren implementieren. Dabei werden Aspekte, die nicht zu dem eigentlichen Verfahren gehören, wie z.B. das Laden von Daten und das Speichern der Ergebnisse, aus der gemessenen Laufzeit herausgerechnet. Da die Laufzeit abhängig von den verarbeiteten Daten variieren kann, wird ein Durchschnitt über die kompletten, im jeweiligen Experiment genutzten, Datensätze berechnet.

7.4 Durchgeführte Untersuchungen

Im Folgenden werden die Experimente beschrieben, die durchgeführt wurden, um die verschiedenen in dieser Arbeit vorgestellten Teilverfahren zu untersuchen.

7.4.1 Detektion von Personen in 3D-Punktwolken

Parametrisierung des Verfahrens

Für die Personendetektion wurden zunächst eine Reihe von Untersuchungen durchgeführt, um den Einfluss verschiedener Parameter auf das Verfahren zu beurteilen und um gute Werte für diese Parameter zu bestimmen. Sie lassen sich im Hinblick darauf, wie und wo sie im Verfahren berücksichtigt werden müssen, in zwei Gruppen einteilen. Die erste Gruppe umfasst Parameter, die u.a. einen Einfluss auf das Training des neuronalen Netz haben, weil sie entweder Parameter des neuronalen Netzes selbst oder der Generierung der lokalen Punktnachbarschaften sind, die deren Beschaffenheit beeinflussen. Um sie zu untersuchen, müssen jeweils neuronale Netze gesondert mit ihnen trainiert werden. Die zweite Gruppe an Parametern sind unabhängig vom neuronalen Netz und beeinflussen das sonstige Verfahren. Zur ersten Gruppe gehören:

- Der Radius der lokalen Punktnachbarschaften. Also die Entfernung, bis zu der zwei 3D-Punkte bei der Erzeugung der lokalen Punktnachbarschaften für die Verarbeitung im neuronalen Netz als Nachbarn gesehen werden. Dieser Radius hat zwei Effekte: Wenn er kleiner

wird, wird es unwahrscheinlicher, dass ausreichend Punkte in einer Nachbarschaft sind, um dem neuronalen Netz genügend Informationen für ein aussagekräftiges Ergebnis zu geben. Wenn der Radius größer wird, steigt jedoch die Wahrscheinlichkeit, dass die Nachbarschaft Punkte von vielen verschiedenen Objekten enthält. Im Idealfall wird sie nur von Punkten eines einzelnen Objektes klassifiziert. Sowohl ein zu kleiner als auch ein zu großer Radius kann also Schwierigkeiten erzeugen.

- Die minimale Anzahl an Punkten in einer lokalen Punktnachbarschaft. Wenn zu wenige Punkte zu einer Nachbarschaft gehören, enthält diese nicht genügend Informationen für ein sinnvolles Verarbeitungsergebnis im neuronalen Netz. Gleichzeitig begrenzt dieser Parameter jedoch zusammen mit dem Radius der Nachbarschaft, wie klein die Punktdichte sein darf, um überhaupt noch Nachbarschaften zu erzeugen. Eine passende Untergrenze für die Menge an Nachbarn ist daher erforderlich.

- Die maximale Anzahl an Punkten in einer lokalen Punktnachbarschaft. Unter Annahme, dass in Bereichen mit hoher Punktdichte eine Punktnachbarschaft viele redundante Punkte umfasst, wird die Menge der Nachbarpunkte durch zufällige Selektion auf einen gewählten Maximalwert begrenzt. Dieser Parameter soll u.a. die Laufzeit des Verfahrens verbessern, indem die Verarbeitung redundanter Informationen vermieden wird. Es soll ein Wert gefunden werden, der hierfür ideal ist.

Die zweite Gruppe umfasst:

- Der Wert für σ bei der Neubewertung des Stimmgewichts der Objektkandidaten im Abstimmverfahren. Dieser beeinflusst, wie viel des Stimmgewichts eines Nachbarkandidaten in einer gegebenen Entfernung dem gerade bewerteten Kandidaten aufgeschlagen wird und bis zu welcher Entfernung (2σ) diese Nachbarn überhaupt berücksichtigt werden.

- Die Rate des Sub-Samplings bei der Generierung der lokalen Nachbarschaften. Eine höhere Rate des Sub-Samplings erhöht die Verarbeitungsgeschwindigkeit. Zu hohe Werte könnten aber die Qualität der Ergebnisse beeinträchtigen. Die Rate des Sub-Sampling gibt dabei an, wie viele potenzielle lokale Nachbarschaften übersprungen werden. Ein Wert von bedeutet z.B., dass jeder dritte Punkt der verarbeiteten Daten als Ursprungspunkt für eine lokale Nachbarschaft genutzt wird.

Um die Auswirkung dieser Parameter zu untersuchen und um optimale Werte zu ermitteln, wurden die drei Datensätze des MODISSA-Systems mit den dazugehörigen Daten für das Training und die Validierung des Trainingsprozesses genutzt. Die Untersuchungen der Parameter, die Bezug auf die Konfiguration und das Training des neuronalen Netzes haben, wurden für die Standardvariante und die vereinfachte Variante dieses Netzes durchgeführt. Die anderen Parameter wurden nur mit der Standardvariante des Netzes untersucht, da die Ergebnisse sich hier auf die vereinfachte Variante übertragen lassen. Der gerade untersuchte Parameter wurde bei den Untersuchungen variiert, während die anderen konstant blieben. Die hierbei für beide Varianten des neuronalen Netzes genutzte Grundkonfiguration ist in Tabelle 7.3 dargestellt.

Optimierte Konfiguration

Das Ergebnis der ersten Reihe von Experimenten wurde genutzt, um die Parametrisierung des Verfahrens für die weiteren Experimente zu optimieren. Die dabei resultierenden Konfigurationen für beide Varianten des neuronalen Netzes befinden sich in Tabelle 7.4.

Parameter	Wert
Nachbarschaftsradius	0,5 m
Minimale Nachbarschaftsgröße	15 Punkte
Maximale Nachbarschaftsgröße	150 Punkte
σ	0,4 m
Sub-Sampling Rate	3

Tabelle 7.3: Grundkonfiguration der Personendetektion

Parameter	Wert Standardvariante	Wert vereinfachte Variante
Nachbarschaftsradius	0,6 m	0,6 m
Minimale Nachbarschaftsgröße	10 Punkte	3 Punkte
Maximale Nachbarschaftsgröße	150 Punkte	150 Punkte
σ	0,4 m	0,4 m
Sub-Sampling Rate	3	3

Tabelle 7.4: Optimierte Konfiguration der Personendetektion für beide Varianten des neuronalen Netzes

Verhalten in Abhängigkeit zur Menge an Trainingsdaten

Das Verfahren zur Detektion von Personen verwendet ein neuronales Netz, welches zuvor trainiert werden muss. Die Qualität dieses Trainings und des Trainingsergebnisses hängt u.a. von der Menge an verwendeten Trainingsdaten ab. Um zu untersuchen, wie sich diese Menge auf die beiden untersuchten Varianten des neuronalen Netzes sowie das Detektionsverfahren als ganzes auswirken, wurden diese mit einer verminderten Menge an Trainingsdaten trainiert. Hierfür wurden zunächst aus den insgesamt 1300 Punktwolken der Trainingsdaten 650 zufällig ausgewählt, die Menge an Punktwolken für das Training wurde also halbiert. Aus diesen 650 Punktwolken wurden dann nochmal 325 zufällig ausgewählt, die ursprüngliche Menge wurde als geviertelt. In zwei weiteren Schritten wurden aus den verbleibenden Punktwolken zunächst 100 und davon dann nochmal 50 Punktwolken zufällig ausgewählt. Um zu untersuchen, wie stark die Auswahl der Punktwolken eine Rolle für das Training spielt, wurde zudem aus aus Gesamtmenge von 1300 Punktwolken alternativ noch eine zusammenhängende Sequenz von 100 Punktwolken ausgewählt.

Beide Varianten des neuronalen Netzes, in ihrer jeweils optimierten Konfiguration, wurden dann jeweils mit den unterschiedlich erzeugten Sets an Trainingsdaten trainiert. Diese umfassen im einzelnen:

- Die vollständigen **1300** Punktwolken
- **650** Punktwolken zufällig ausgewählt aus den 1300
- **325** Punktwolken zufällig ausgewählt aus den 650
- **100** Punktwolken zufällig ausgewählt aus den 325
- **50** Punktwolken zufällig ausgewählt aus den 100
- **100** Punktwolken in einer zusammenhängenden Sequenz ausgewählt aus den 1300

Für die Validierung des Trainingsfortschritts wurden dabei in allen Fällen die vollen 226 Punktwolken der Validierungsdaten verwendet. Die Leistung des Detektionsverfahrens mit den

so unterschiedlich trainierten Instanzen der beiden Varianten des neuronalen Netzes wurden dann miteinander verglichen.

Verhalten in Abhängigkeit zur Entfernung zum Sensor

Die Annahme ist, dass die Leistung des Verfahrens abnimmt, wenn die lokal beim Objekt vorliegende Punktdichte kleiner wird. Dies ergibt sich daraus, dass bei einer geringen Punktdichte nicht mehr genügend geometrische Informationen über das Objekt in der Punktwolke vorhanden sind, um dieses anhand seines Aussehens zu detektieren. Da sich die Punktdichte bei den verwendeten Sensoren aus der Entfernung zwischen Sensor und Objekt ergibt, wurden die Ergebnisse der Experimente anhand dieser Entfernung ausgewertet. Für diese Untersuchung wurde die optimierte Konfiguration von beiden Varianten des neuronalen Netzes genutzt (vgl. Tabelle 7.4).

Als relevante Entfernung zwischen Objekt und Sensor wurde bei richtig positiven und falsch negativen Ergebnissen die Entfernung zwischen Sensor und Objektmittelpunkt in der Grundwahrheit genommen. Bei falsch positiven Ergebnissen liegt diese nicht vor, weswegen hier die Entfernung zwischen Sensor und Objektmittelpunkt in den Ergebnissen der Detektion genutzt wurde. Um die Menge von Personen in den jeweils untersuchten Entfernungsbereichen zu erhöhen, wurden bei diesem Experiment immer alle drei Datensätze gemeinsam genutzt.

7.4.2 Tracking von Personen in 3D-Punktwolken

Trackingleistung

Ein Fokus der Experimente im Bezug auf das Tracking von Personen in 3D-Punktwolken war es, die Leistungsfähigkeit des gewählten Verfahrens zum Tracking zu bestimmen. Diese Experimente wurden auch dazu genutzt, gute Werte für einige der Parameter des Verfahrens zu finden. Die Festlegung der Konfiguration der anderen Parameter, sowie die initiale Konfiguration der experimentell optimierten Parameter, erfolgte anhand von theoretischen Überlegungen. Ein Teil dieser theoretischen Überlegungen war die maximale Geschwindigkeit, mit der sich eine Person zu Fuß fortbewegt. Dabei wurde mit etwa 7 m/s (ca. 25 km/h) eine großzügige Obergrenze gewählt, die nur von Sprintern überschritten wird und für den normalen Fußgängerverkehr im urbanen Umfeld mehr als ausreichend ist. Das Trackingverfahren hat folgende Konfigurationsparameter:

- Varianz des Messrauschens des Kalman-Filters: In Voruntersuchungen hat sich gezeigt, dass die Genauigkeit der durch das Detektionsverfahren ermittelten Objektpositionen hoch ist. Dieser Wert wurde daher auf 0,25 festgelegt und war nicht Teil der empirischen Untersuchungen.
- Varianz des Prozessrauschens des Kalman-Filters: Der Wert dieses Parameters wurde im Rahmen der Experimente festgelegt.
- Maximale Distanz für die Assoziation zwischen vorhergesagter und gemessener Objektposition: Ausgehend von der angenommenen maximalen Geschwindigkeit der Personen (7 m/s) sowie der durchschnittlichen Zeit zwischen der Aufnahme von zwei aufeinanderfolgenden Punktwolken (0,1 s), wurde dieser Wert auf 0,7 m festgelegt. So ist sichergestellt, dass eine Assoziation auch möglich ist, bevor der Kalman-Filter die Geschwindigkeit einer Person bestimmen konnte, um ihre Position korrekt vorherzusagen.
- Maximale Geschwindigkeit der getrackten Objekte: Dieser Wert wird auch bei der Initialisierung des Kalman-Filters einer neuen getrackten Person verwendet, um die Geschwindigkeitsanteile der Kovarianzmatrix des Objektzustands zu initialisieren. Wie bereits erwähnt wurde er auf 7 m/s festgelegt.

- Maximale Varianz der vorhergesagten Objektposition: Beim verwendeten Trackingverfahren werden getrackte Objekte, die nicht mehr länger detektiert werden, entfernt, wenn die Varianz der für sie vorhergesagten Position zu groß wird (vgl. Abschnitt 5.2). Dieser Parameter definiert den dabei verwendeten Schwellwert. Der Wert dieses Parameters wurde anhand der Ergebnisse der Experimente festgelegt.

Neben diesen Parametern der Trackingsverfahrens hat aber auch das Detektionsverfahren Einfluss auf die Trackingleistung des Gesamtverfahrens. Für die Experimente im Bezug auf die Trackingleistung, wurde die Standardvariante des neuronalen Netzes in der optimierten Konfiguration verwendet (vgl. Abschnitt 7.4.1), zunächst mit einem Schwellwert für die Detektion, der zu einer Gesamtgenauigkeit über alle drei Datensätze von 0,9 führt. Nachdem das Trackingverfahren durch die Experimente vollständig parametrisiert war, wurde auch dieser Schwellwert nochmals variiert, um dessen Einfluss auf das Tracking zu bestimmen. Tabelle 7.5 gibt die unterschiedlichen während der Experimente genutzten Konfigurationen des Trackingverfahrens wieder.

Parameter	**Prozessrauschen Experiment**	**Max. Varianz Experiment**	**Optimierte Konfiguration**
Varianz Messrauschen	0,25	0,25	0,25
Varianz Prozessrauschen	-	1,6	1,6
Maximale Assoziationsdistanz	0,7 m	0,7 m	0,7 m
Maximale Geschwindigkeit	7 m/s	7 m/s	7 m/s
Maximale Varianz der Position	0,25	-	0,425

Tabelle 7.5: Verschiedene für die Experimente genutzte Konfigurationen des Trackings

Im Rahmen dieser Experimente wurden die Kennzahlen $MOTA$ und $MOTP$ bestimmt. Dabei wurden auch Personen in den Daten der Grundwahrheit berücksichtigt, die stark oder sogar vorübergehend vollständig verdeckt sind. Deren Position durch das Detektionsverfahren alleine also ggf. vorübergehend überhaupt nicht bestimmt werden kann.

Unterstützung der Personendetektionsleistung durch das Tracking

Das Tracking hat zwei Aufgaben. Zum einen stellt es zusätzliche Informationen bereit, in Form einer konsistenten Objektidentität und dem Bestimmen der Objektgeschwindigkeit. Zum anderen hilft es aber auch dabei mit Objekten umzugehen, die vorübergehend verdeckt sind. Wenn große Teile eines Objekts nicht mehr zu sehen sind, wird es für jedes Detektionsverfahren schwieriger diese noch zu detektieren. Und bei vollständig verdeckten Objekten ist eine solche Detektion unmöglich. Es ist jedoch dennoch wünschenswert weiterhin eine Position für welche Objekte zu bestimmen, wenn diese nur vorübergehend verdeckt sind. Um zu bestimmen, in welchem Umfang das Tracking bei dieser Aufgabe helfen kann, wurde die Detektionsleistung des Verfahrens mit und ohne Tracking miteinander verglichen. Anders als bei den Experimenten in Abschnitt 7.4.1, wurden für dieses Experiment auch Personen mit starken Verdeckungen sowie solche, die vollständig verdeckt sind, berücksichtigt. Für das Experiment wurde die optimierte Konfiguration des Trackings sowie die optimierte Konfiguration des Detektionsverfahrens mit der Standardvariante des neuronalen Netzes genutzt.

7.4.3 Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern

In diesem Abschnitt werden die Experimente zur gemeinsamen Nutzung von LiDAR-Sensoren und RGB-Kameras für die Detektion und Körperposenschätzung von Personen beschrieben. Da keine Grundwahrheit für die Körperposen von Personen im 3D-Raum vorhanden war, lag der Fokus der quantitativen Auswertung dieser Experimente bei dem Vergleich der Detektionsleistung der drei untersuchten Varianten. Eine umfangreiche Untersuchung des genutzten Verfahrens zur Posenschätzung wurde aber von Cao et al. [2019] durchgeführt, deren bereits trainiertes „Body+foot“ Modell wurde für diese Experimente verwendet.

Wie in Kapitel 6 beschrieben, unterscheiden sich die drei untersuchten Varianten darin, welche Sensormodalität welche Aufgabe übernimmt und wie die Modalitäten zusammengeführt werden. Variante 1 nutzt LiDAR-Sensoren als führende Sensoren. Personen werden in deren Daten detektiert, wofür das mit den in Abschnitt 7.4.1 beschriebenen Experimenten untersuchte Verfahren zur Personendetektion verwendet wird. Anschließend findet eine gezielte Bildauswertung zur Gewinnung von Daten über die Körperpose dieser Personen statt. Im Hinblick auf die reine Detektion von Personen erzielt diese Variante dieselbe Leistung, die auch die Detektion von Personen in 3D-Punktwolken alleine erzielen würde. Variante 2 nutzt die Kameras als führende Sensoren und nutzt die Posenschätzung in RGB-Bildern zusätzlich für die Detektion der Personen. Die LiDAR-Daten werden dann genutzt, um 3D-Koordinaten für diese detektierten Personen zu bestimmen. Im Hinblick auf die Detektion hängt die Leistung dieser Variante alleine von der Leistung des Verfahrens zur Körperposenschätzung ab. Variante 3 nutzt keine der beiden Sensormodalitäten als führend. Stattdessen bestätigen sich diese gegenseitig. Eine detektierte Person fließt nur dann in das Ergebnis ein, wenn sie von beiden Verfahren detektiert wurde. Dies sollte falsch positive Ergebnisse reduzieren, da diese dann nur auftreten können, wenn in beiden Sensormodalitäten an derselben Stelle fälschlicherweise eine Person detektiert wird. Im Umkehrschluss wird jedoch die Zahl der falsch negativen Ergebnisse größer. Variante 1 und Variante 3 nutzen auch das Verfahren zur Detektion von Personen in 3D-Punktwolken. Für die Experimente wurde die optimierte Konfiguration von diesem verwendet. Bei den Untersuchungen zur Variante 1 wird sowohl das Standardmodell als auch das vereinfachte Modell genutzt. Bei der Variante 3 wurde hingegen nur das Standardmodell berücksichtigt.

Neben einem Vergleich der Detektionsleistung der drei Varianten wird auch ihre Laufzeit miteinander verglichen. Dies soll u.a. dazu dienen festzustellen, ob die zielgerichtete Auswertung von Bildausschnitten anhand der zuvor in LiDAR-Daten detektierten Personen einen Vorteil gegenüber der Verarbeitung der vollständigen Bilder darstellt. Dieser mögliche Vorteil hängt vor allem auch davon ab, wie viele Bildausschnitte generiert und verarbeitet werden, also wie viele richtig positive und falsch positive Detektionen es in den Punktwolken gibt. Es ist daher für diese Auswertung auch relevant, wie der Schwellwert für das Detektionsverfahren gewählt wird. Für den Vergleich der Laufzeiten wird bei der Variante 1 daher ein Schwellwert genutzt, der über alle Datensätze zu einer Genauigkeit von 0,9 führt.

8 Ergebnisse

In diesem Kapitel werden die Ergebnisse der im vorherigen Kapitel beschriebenen Experimente wiedergegeben. Eine umfassende Diskussion dieser Ergebnisse befindet sich im folgenden Kapitel.

8.1 Detektion von Personen in 3D-Punktwolken

In diesem Abschnitt befinden sich die Ergebnisse der Experimente im Bezug auf die Detektion von Personen in 3D-Punktwolken. Zunächst werden die Ergebnisse der Untersuchungen, die zur optimalen Parametrisierung des Verfahrens durchgeführt wurden, vorgestellt. Anschließend folgen die Ergebnisse zu den Untersuchungen im Bezug auf eine reduzierte Menge an Trainingsdaten und zuletzt die im Bezug auf die Punktdichte bzw. die Distanz zwischen Sensor und zu detektierender Person.

8.1.1 Parametrisierung des Verfahrens

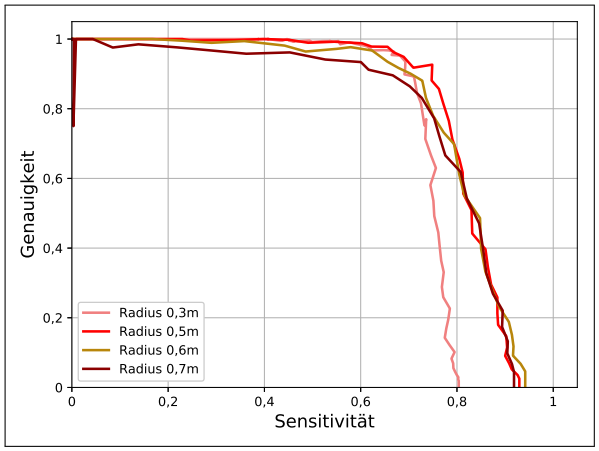
In diesem Abschnitt werden die Ergebnisse der Experimente im Hinblick auf die Parametrisierung des Verfahrens zur Personendetektion in LiDAR-Punktwolken vorgestellt. Diese sind in einzelne Abschnitte für die jeweils untersuchten Parameter unterteilt.

Radius der Punktnachbarschaft

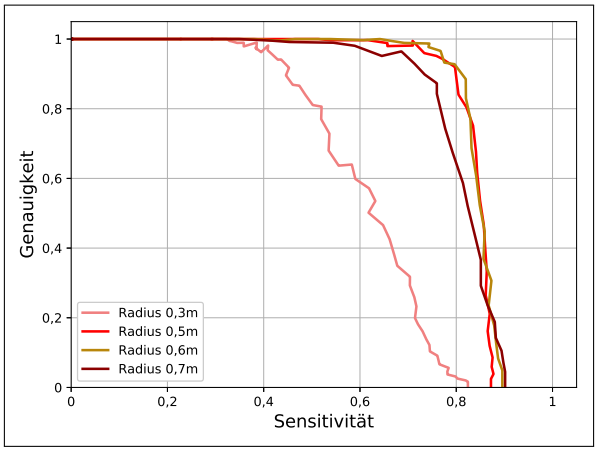
Abbildung 8.1 zeigt die Ergebnisse, für unterschiedlichen Radien der Punktnachbarschaft unter Verwendung der Standardvariante des neuronalen Netzes, Abbildung 8.2 zeigt die entsprechenden Ergebnisse, mit der vereinfachten Variante des neuronalen Netzes. Bei beiden Varianten des neuronalen Netzes fällt auf, dass ein Radius von 0,3 m einen starken negativen Einfluss auf die Ergebnisse hat. Er kann also als zu klein angesehen werden.

Die Ergebnisse für 0,5 m, 0,6 m und 0,7 m liefern ein weniger eindeutiges Bild. Hier fällt auf, dass 0,5 m zu einer besseren Genauigkeit, also zu einem kleineren Anteil an Falschdetektionen führt. 0,7 m hingegen führt zu einer besseren Sensitivität, es wird also ein größerer Anteil der tatsächlich vorhandenen Personen detektiert. Dieser Effekt ist deutlicher im „Ettlingen 2“-Datensatz zu sehen und noch stärker im „Ettlingen 3“-Datensatz. In diesen machen Personen, die weiter vom Sensor entfernt sind, einen zunehmend größeren Teil der vorhandenen Personen aus (vgl. Abbildung 7.6). Die Ergebnisse für 0,6 m wiederum liegen erwartungsgemäß zwischen den Ergebnissen für 0,5 m und 0,7 m. Insbesondere im „Ettlingen 1“-Datensatz, wo Personen in kleinerer Entfernung zum Sensor dominieren, ist die Genauigkeit bei 0,6 m verglichen zu der bei 0,5 m schlechter. Was sich jedoch beim „Ettlingen 2“- und „Ettlingen 3“- Datensatz ein wenig ausgleicht. Auch ist die Sensitivität in allen Fällen besser.

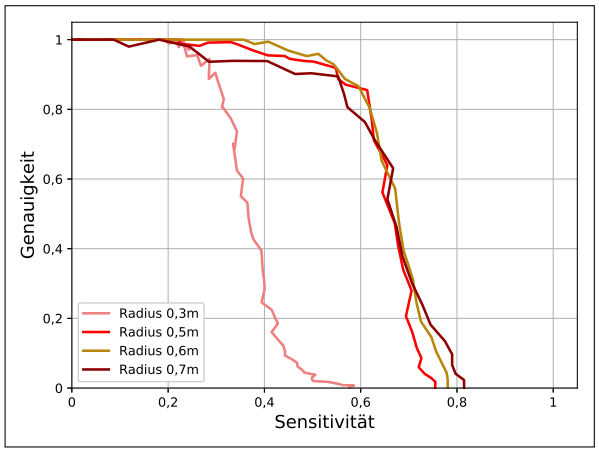
Die Ergebnisse lassen sich wie folgt erklären: Die Genauigkeit sinkt, wenn der Radius der Nachbarschaften zu groß wird, da es häufiger vorkommt, dass mehrere verschiedene Objekte Teil einer lokalen Nachbarschaft sind. Es wird also schwieriger für das neuronale Netz diese korrekt



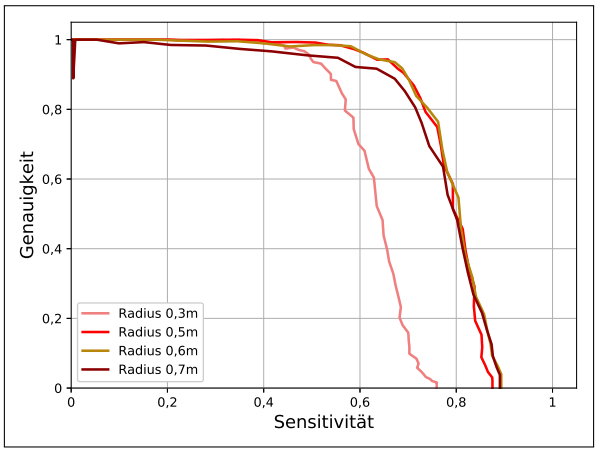
(a)



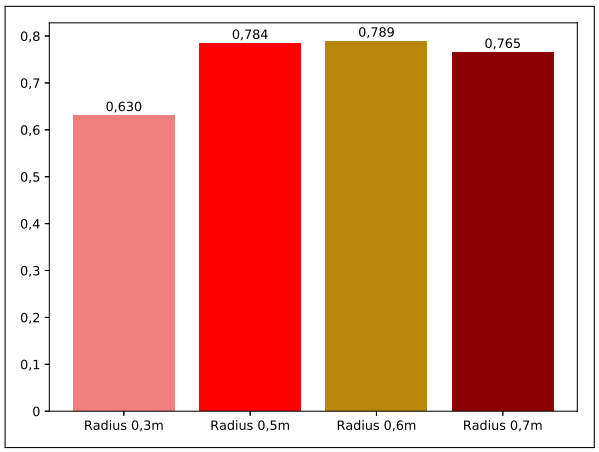
(b)



(c)

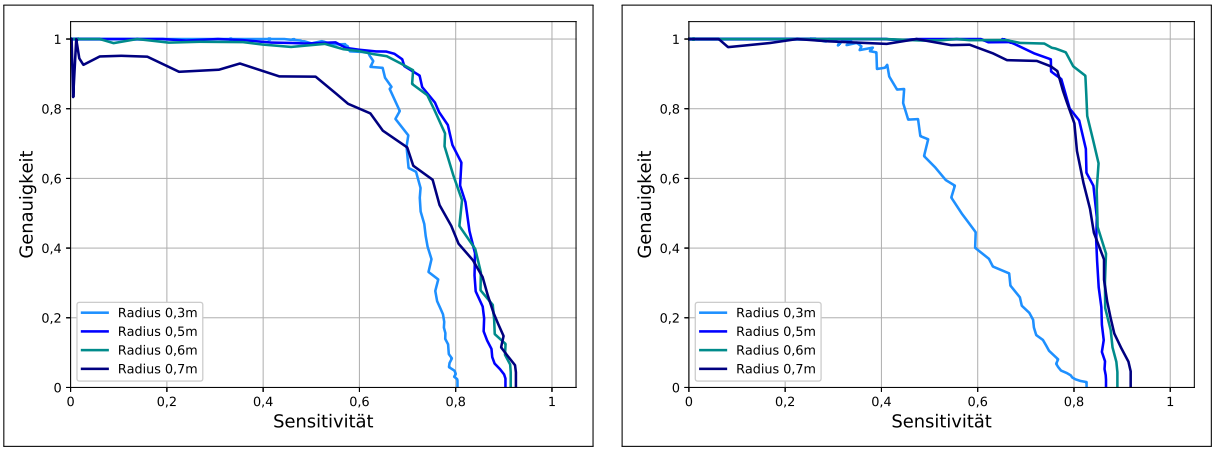


(d)



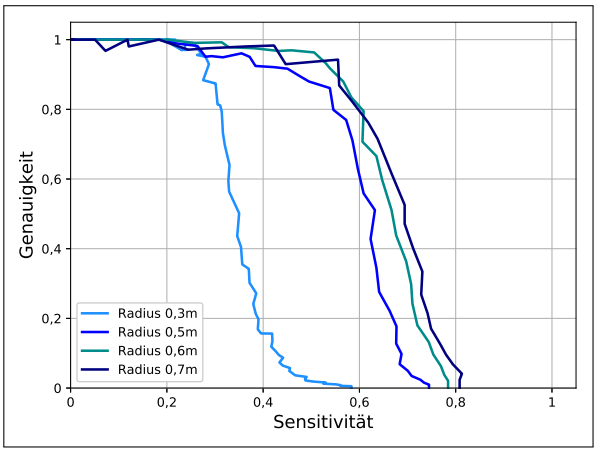
(e)

Abbildung 8.1: Ergebnisse mit unterschiedlichem Radius für lokale Punktnachbarschaften und der Standardvariante des neuronalen Netzes. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3.

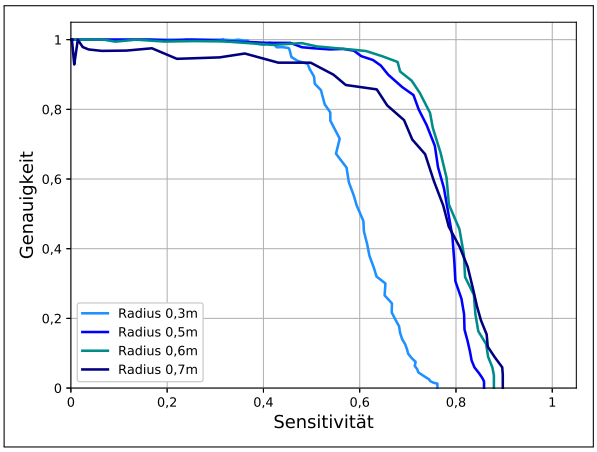


(a)

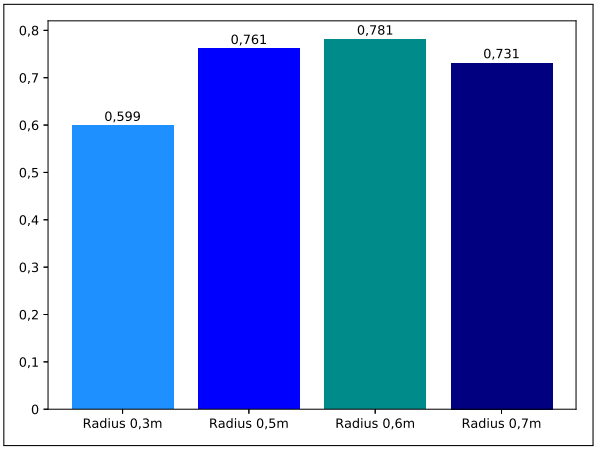
(b)



(c)



(d)



(e)

Abbildung 8.2: Ergebnisse mit unterschiedlichem Radius für lokale Punktnachbarschaften und der vereinfachten Variante des neuronalen Netzes. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3.

zu klassifizieren. In Regionen mit geringer Punktdichte, also weit vom Sensor entfernt, steigt durch einen größeren Radius aber die Chance, dass überhaupt noch genügend Punkte in einer Nachbarschaft sind, um diese sinnvoll zu verarbeiten. Zusätzlich spielt hier die im Folgenden untersuchte Untergrenze für die Anzahl Punkte eine Rolle, die eine Nachbarschaft mindestens haben muss, um für die weitere Verarbeitung infrage zu kommen.

Aufgrund dieser Ergebnisse wurde entschieden, in späteren Untersuchungen einen Radius 0,6 m für die lokalen Punktnachbarschaften zu verwenden.

Minimale Anzahl an Punkten in einer lokalen Punktnachbarschaft

Die Auswirkungen von unterschiedlichen Grenzen für die minimale Anzahl an Punkten in einer lokalen Punktnachbarschaft werden von Abbildung 8.3 für die Standardvariante des neuronalen Netzes gezeigt. Abbildung 8.4 zeigt sie für die vereinfachte Variante. Es zeigt sich, dass der für die ursprüngliche Grundkonfiguration genutzte Wert von 15 zu groß gewählt wurde. Beide Varianten des neuronalen Netzes liefern bessere Ergebnisse, wenn dieser Wert reduziert wird. Vergleichbar mit einem größeren Radius für die Nachbarschaft, scheint auch eine niedrigere Grenze, für die minimale Anzahl an Punkten in einer Nachbarschaft bei der Sensitivität zu helfen, wenn Personen mit geringer Punktdichte detektiert werden sollen. Es lässt sich jedoch auch beobachten, dass die Genauigkeit abnimmt, wenn die Untergrenze zu klein gesetzt wird. Hier zeigt sich, dass das vereinfachte neuronale Netz anscheinend besser mit sehr dünn besetzten Nachbarschaften umgehen kann als die Standardvariante des neuronalen Netzes.

Es wurde entschieden, für die späteren Untersuchungen bei der Standardvariante des neuronalen Netzes eine Untergrenze von 10 Punkten zu verwenden. Bei der vereinfachten Variante wurde die Untergrenze auf 3 Punkte festgelegt.

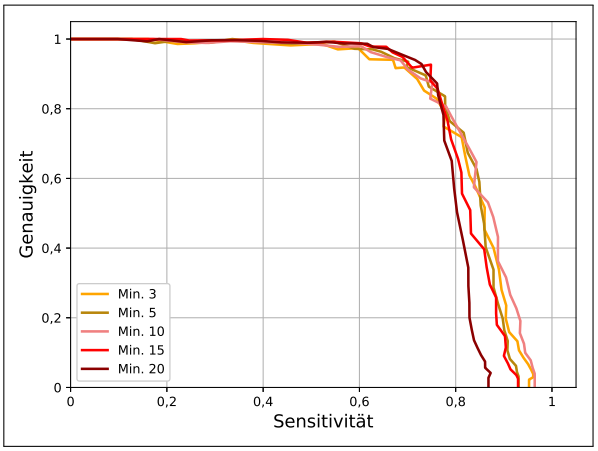
Maximale Anzahl an Punkten in einer lokalen Punktnachbarschaft

Abbildung 8.5 und Abbildung 8.6 zeigen die Ergebnisse von unterschiedliche Werten für den Parameter für die maximale Anzahl an Punkten in einer lokalen Punktnachbarschaft, und zwar unter Verwendung der Standard- und der vereinfachten Variante des neuronalen Netzes. Dieser Parameter beschränkt die maximale Größe der vom neuronalen Netz verarbeiteten lokalen Punktnachbarschaften durch zufällige Auswahl von Punkten. Verglichen mit den beiden anderen untersuchten Parametern der lokalen Punktnachbarschaften zeigt sich, dass dieser Parameter einen geringeren Einfluss auf die Ergebnisse hat. Signifikante Effekte sind nur bei einem sehr geringen Wert für diesen Parameter zu beobachten. Dies ist sicherlich zum Teil auf die Tatsache zurückzuführen, dass dieser Parameter nur dann eine Auswirkung haben kann, wenn die lokale Punktdichte sehr hoch ist, was häufig nicht der Fall ist.

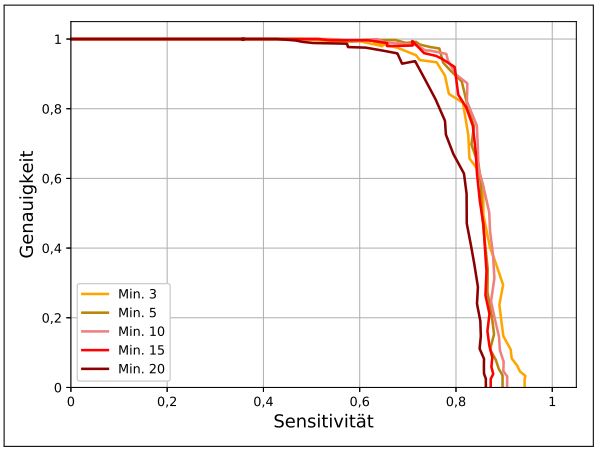
Es wurde entschieden diesen Parameter für die späteren Untersuchungen bei dem ursprünglich gewählten Wert von 150 Punkten zu belassen.

Parameter σ bei der Neubewertung des Stimmgewichts

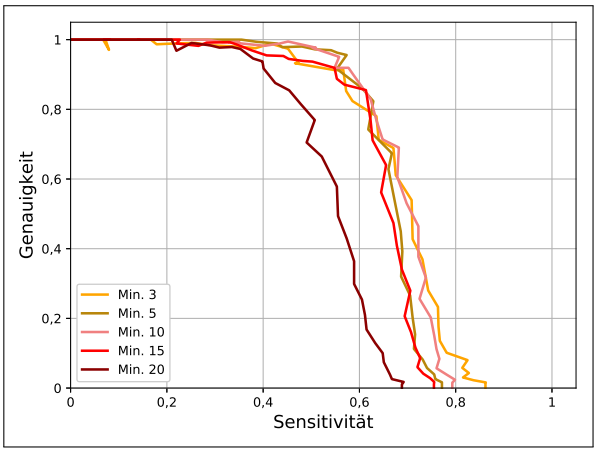
Die Ergebnisse in Bezug auf die Auswirkung von unterschiedlichen Werten für den Parameter σ werden in Abbildung 8.7 dargestellt. Der Parameter spielt bei der Bewertung der Objektkandidaten während des Abstimmverfahrens eine Rolle und beeinflusst, bis zu welcher Entfernung benachbarte Kandidaten bei der Ermittlung des Gewichts eines Objektkandidaten berücksichtigt werden und wie stark ihr jeweiliger Einfluss ist.



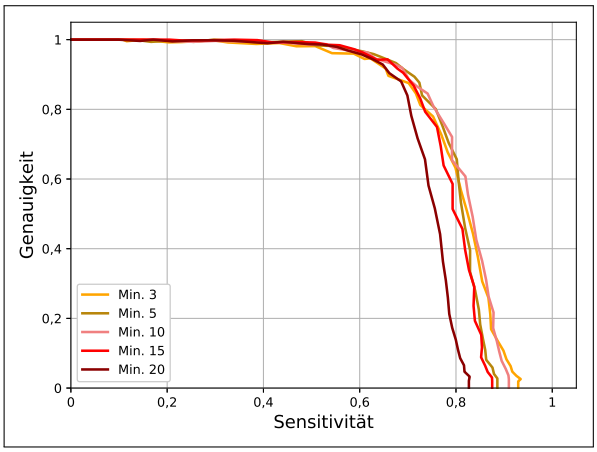
(a)



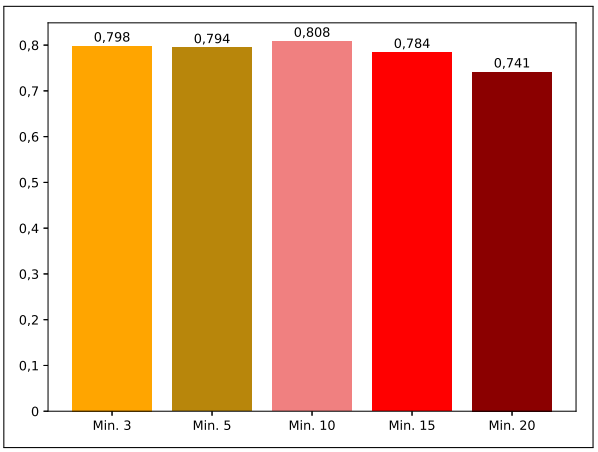
(b)



(c)

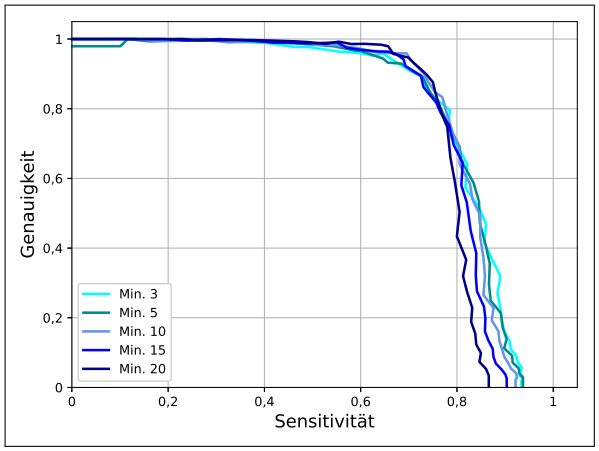


(d)

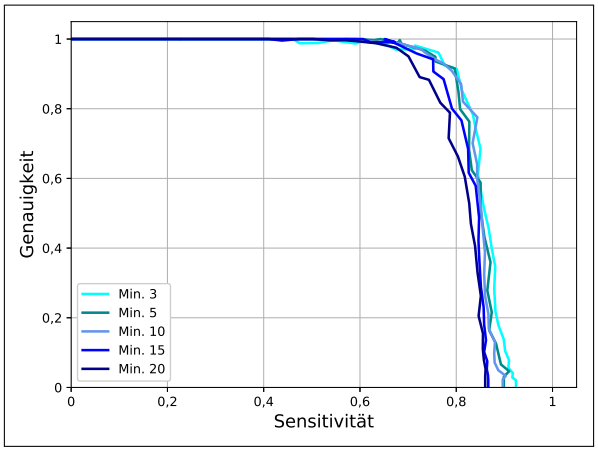


(e)

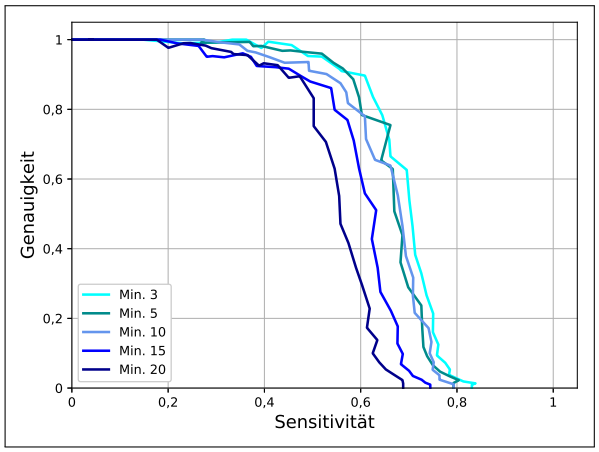
Abbildung 8.3: Ergebnisse mit unterschiedlicher Untergrenze an Punkten in einer lokalen Nachbarschaft und der Standardvariante des neuronalen Netzes. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3.



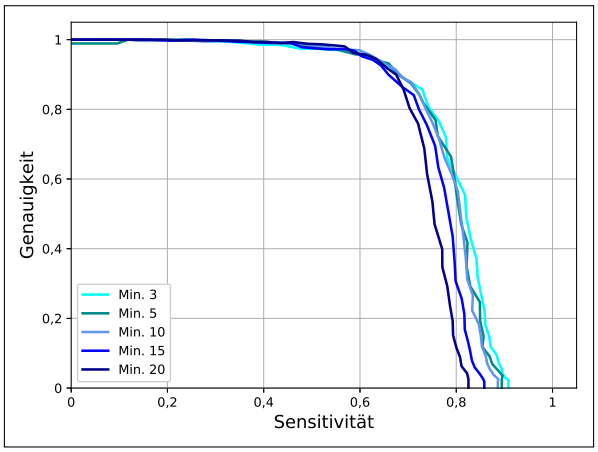
(a)



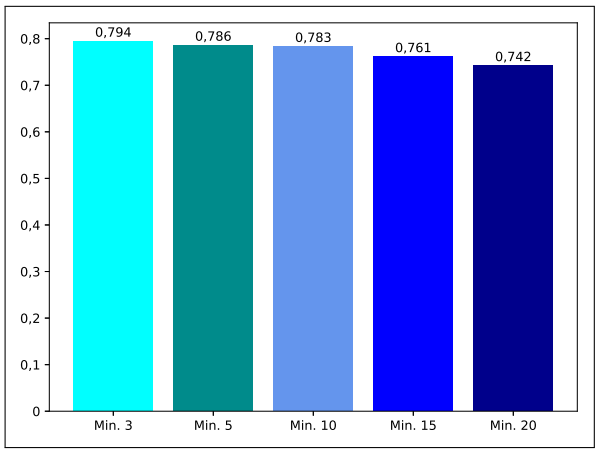
(b)



(c)

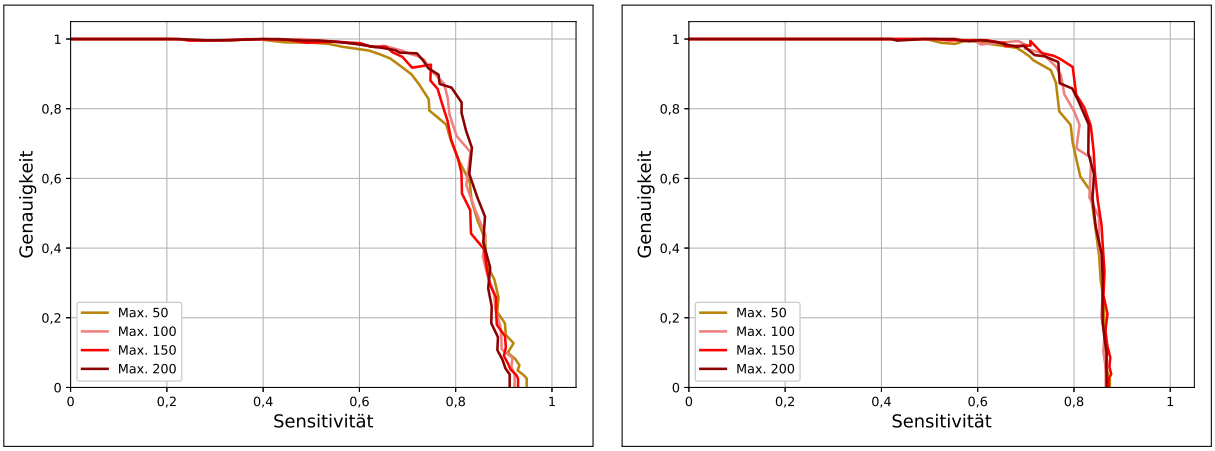


(d)



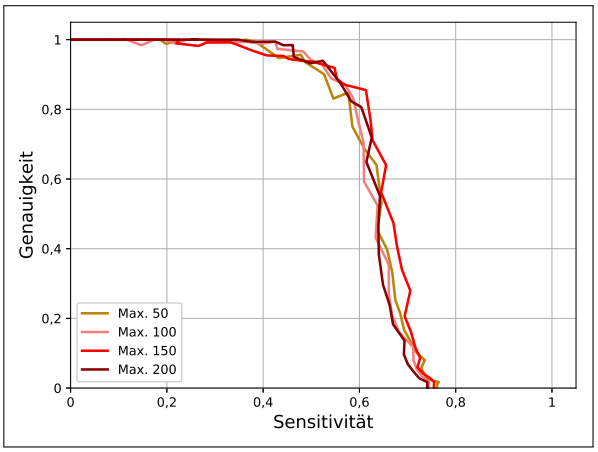
(e)

Abbildung 8.4: Ergebnisse mit unterschiedlicher Untergrenze an Punkten in einer lokalen Nachbarschaft und der vereinfachten Variante des neuronalen Netzes. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3.

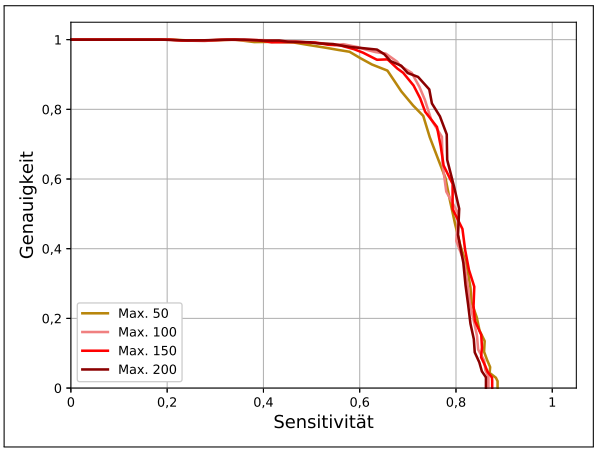


(a)

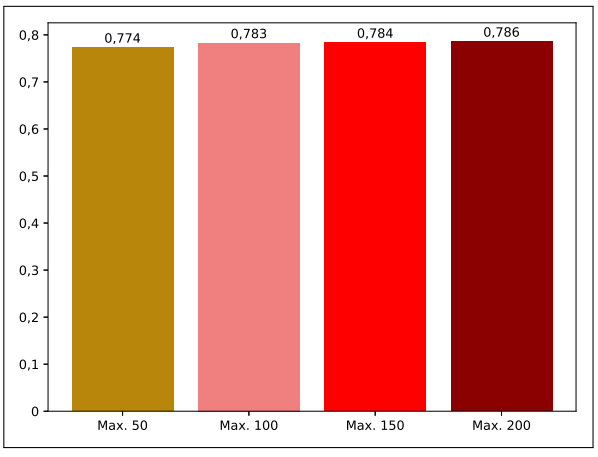
(b)



(c)

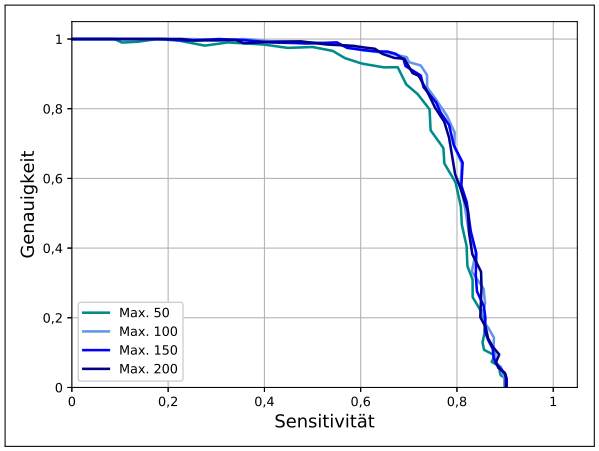


(d)

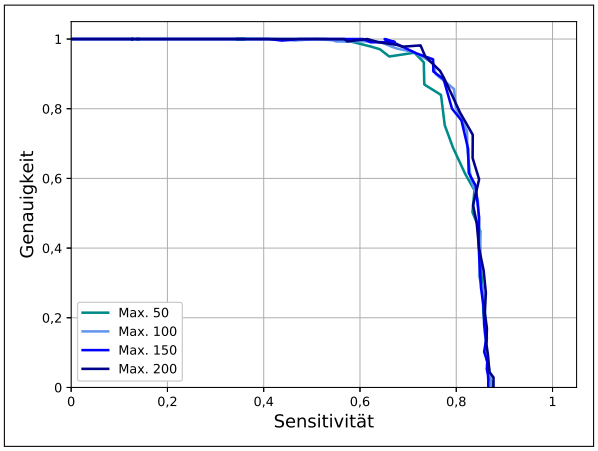


(e)

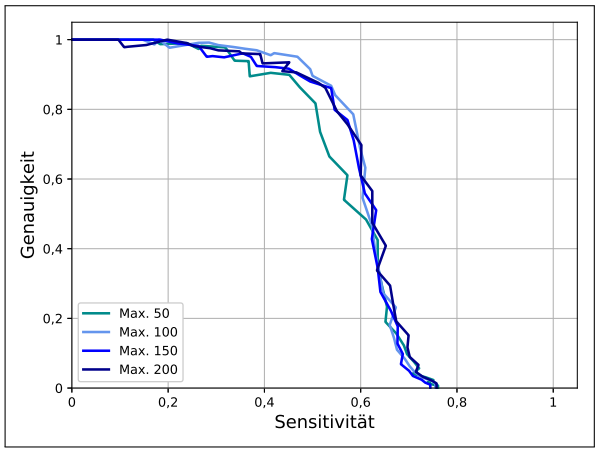
Abbildung 8.5: Ergebnisse mit unterschiedlichem Maximum an Punkten in einer lokalen Nachbarschaft und der Standardvariante des neuronalen Netzes. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3.



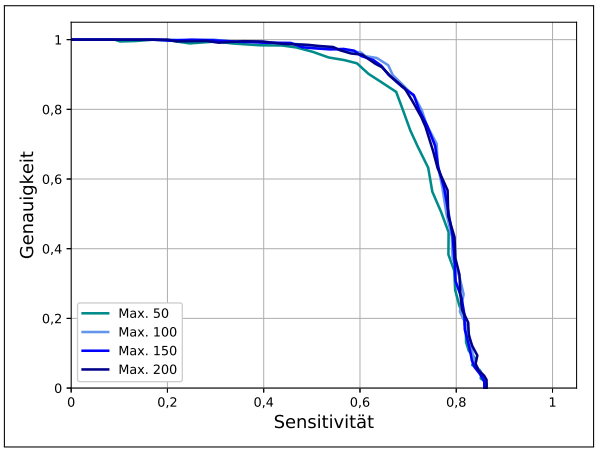
(a)



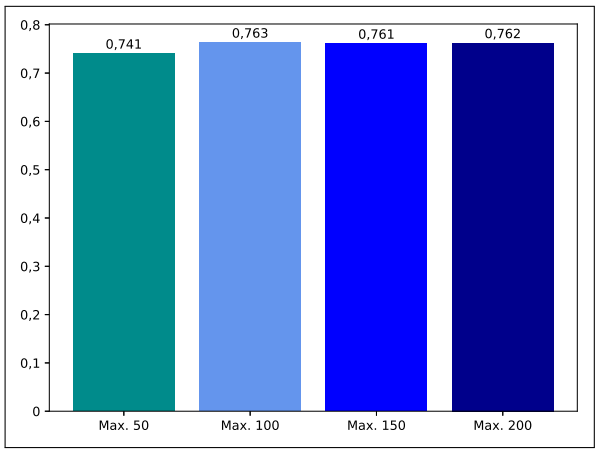
(b)



(c)

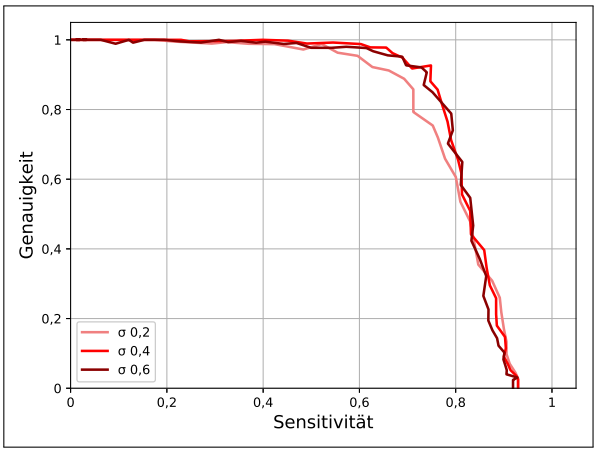


(d)

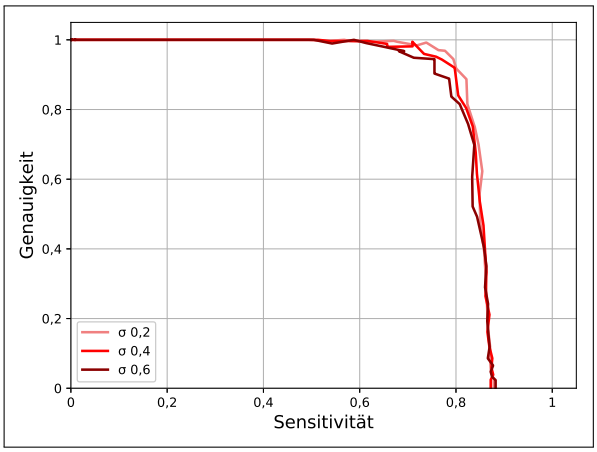


(e)

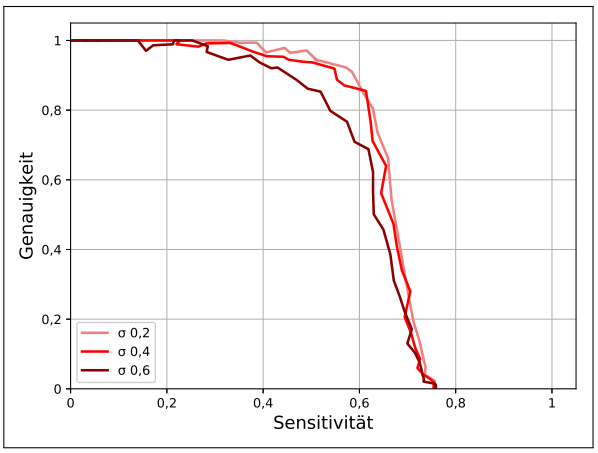
Abbildung 8.6: Ergebnisse mit unterschiedlichem Maximum an Punkten in einer lokalen Nachbarschaft und der vereinfachten Variante des neuronalen Netzes. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3.



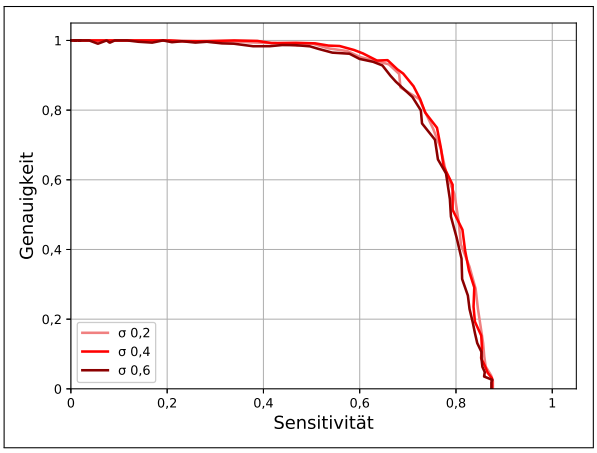
(a)



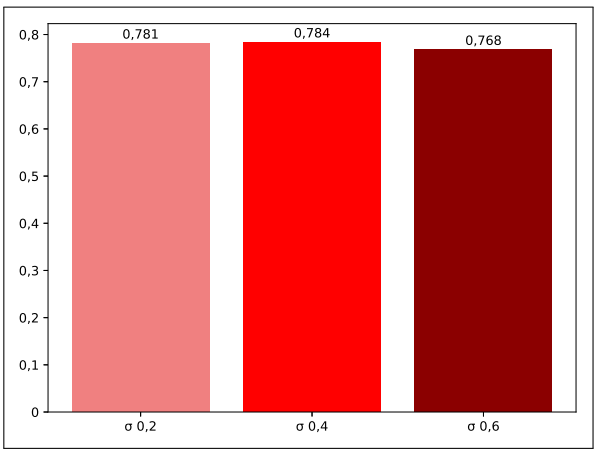
(b)



(c)



(d)



(e)

Abbildung 8.7: Ergebnisse mit unterschiedlichen Werten für den Parameter σ bei der Neubewertung des Stimmgewichts. Für die Untersuchung wurde die Standardvariante des neuronalen Netzes genutzt. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3.

Es zeigt sich, dass bei dem „Ettlingen 1“-Datensatz eher ein größerer Wert von Vorteil ist, bei dem „Ettlingen 3“-Datensatz hingegen eher ein größerer Wert. Für die weiteren Untersuchungen wurde daher entschieden, den ursprünglich gewählten Wert von 0,4 als Kompromiss beizubehalten.

Sub-Sampling

Die Ergebnisse für unterschiedliche Sup-Sampling Raten sind in Abbildung 8.8 dargestellt. Das Sup-Sampling ist so implementiert, dass entweder jede mögliche Nachbarschaft verarbeitet wird oder einige von ihnen übersprungen werden. Erwartungsgemäß wird die Qualität der Ergebnisse durch das Sup-Sampling schlechter, es ist jedoch dafür gedacht die Laufzeit des Verfahrens zu verbessern. Dies wird in Abbildung 8.8 f) dargestellt, wo die Laufzeit der unterschiedlichen Konfigurationen in Relation zueinander gesetzt werden.

In den Ergebnissen ist zu sehen, dass das Sub-Sampling einen kleinen aber durchaus messbaren Einfluss auf die Qualität der Ergebnisse hat. Insbesondere der Schritt vom Verarbeiten jeder Nachbarschaft zum Verarbeiten von nur noch jeder dritten Nachbarschaft verbessert aber signifikant die Laufzeit. Hier gilt es also abzuwägen, auf welches Kriterium ein größerer Wert gelegt wird. Für die weiteren Experimente wurde der ursprüngliche Wert einer Sub-Sampling Rate von 3 beibehalten.

Optimierte Konfiguration

Abbildung 8.9 zeigt die Ergebnisse für beide Varianten des neuronalen Netzes in der optimierten Konfiguration. Diese wurde für alle weiteren Experimente genutzt. Es fällt auf, dass die Ergebnisse sich gegenüber den besten zuvor untersuchten Varianten nicht mehr verbessert haben, sondern im Rahmen erwartbarer Schwankungen ähnlich sind. Da die gegenüber der zuvor genutzten Basiskonfiguration geänderten Parameter (Vergrößerung des Radius der Punktnachbarschaften und Verringerung der minimal erlaubten Anzahl Punkte einer Nachbarschaft) beide vor allem die Leistung des Verfahrens in Bereichen mit geringer Punktdichte verbessern, kann dies als Hinweis darauf gesehen werden, dass für solche Bereiche jetzt die mit dem Verfahren bestmögliche Leistung erreicht wird. Es muss auch bedacht werden, dass es im Trainingsprozess selbst gewisse Zufallseffekte gibt, die die Leistung des Verfahrens in beide Richtungen beeinflussen können.

Beim Vergleich beider Varianten des neuronalen Netzes in dieser optimalen Konfiguration fällt auf, dass die Standardvariante des Netzes für den „Ettlingen 1“-Datensatz klar die besseren Ergebnisse liefert, während die Ergebnisse der beiden Varianten bei den anderen Datensätzen, näher beieinander liegen. Eine mögliche Erklärung dafür ist, dass die Standardvariante mehr davon profitiert, wenn eine hohe Punktdichte bei den erfassten Personen vorliegt, also wenn diese näher beim Sensor sind. Wie zuvor erläutert, ist dies vor allem im „Ettlingen 1“-Datensatz der Fall. Dies wird in den folgenden Experimenten aber noch näher untersucht. Im Hinblick auf die Laufzeit fällt auf, dass die gesteigerte Komplexität des neuronalen Netzes in der Standardvariante diese nahezu verdoppelt.

8.1.2 Detektionsleistung in Abhängigkeit zur Menge an Trainingsdaten

Abbildung 8.10 stellt die Ergebnisse der Untersuchungen zur Menge und Art der verwendeten Trainingsdaten dar. Es zeigt sich, dass insbesondere die Halbierung der Trainingsdaten keinen großen Einfluss auf die Leistung des Verfahrens hat. Für die Standardvariante des neuronalen Netzes nahm diese sogar leicht zu, was vermutlich an Zufallseffekten beim Training und im Detektionsprozess liegt. Auch die Nutzung von nur einem Viertel der ursprünglich 1300 Punktwolken für das Training hatte nur geringen Einfluss auf die Leistung der Detektion von Personen. Die weitere Reduktion der Menge an Trainingsdaten auf 100 und 50 Punktwolken hat die Standardvariante

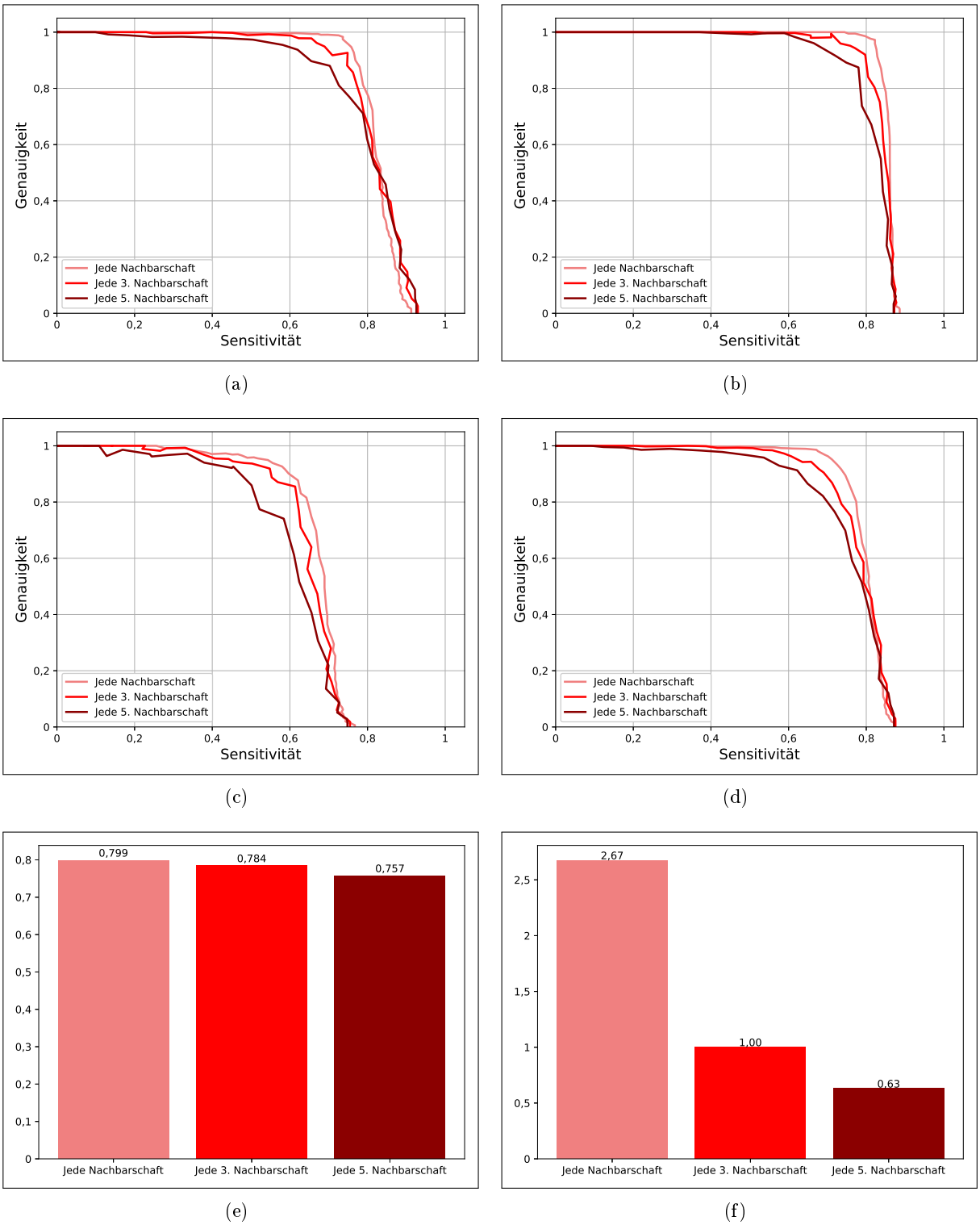


Abbildung 8.8: Ergebnisse mit unterschiedlichem Sub-Sampling. Für die Untersuchung wurde die Standardvariante des neuronalen Netzes genutzt. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3, f) Durchschnittliche Laufzeit relativ zur Laufzeit mit einer Sup-Sampling Rate von 3

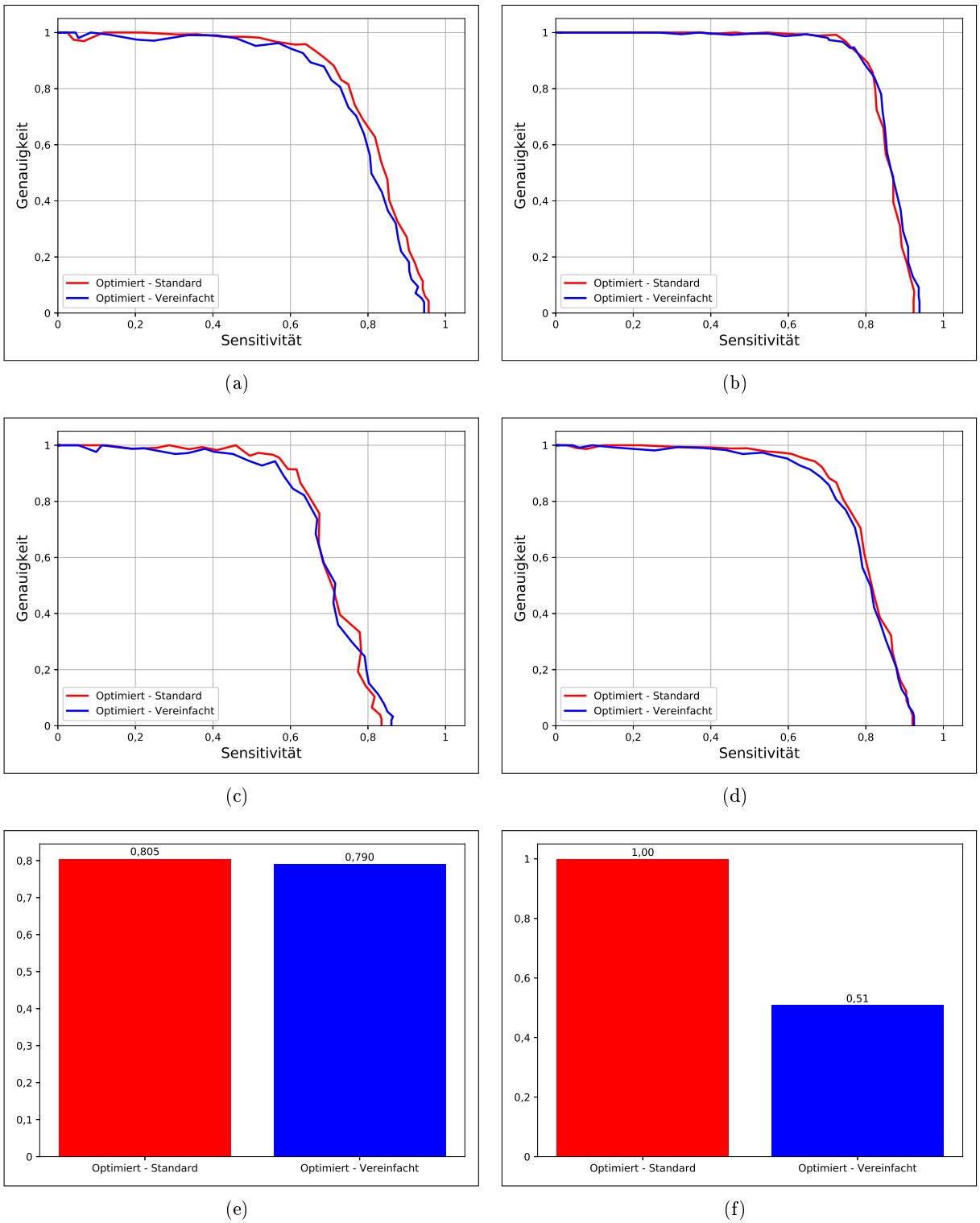


Abbildung 8.9: Ergebnisse unter Nutzung der optimierten Konfiguration für beide Varianten des neuronalen Netzes. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert, e) Fläche unter den Kurven für Ettlingen 1 - 3, f) Durchschnittliche Laufzeit relativ zur Laufzeit der Standardvariante des neuronalen Netzes.

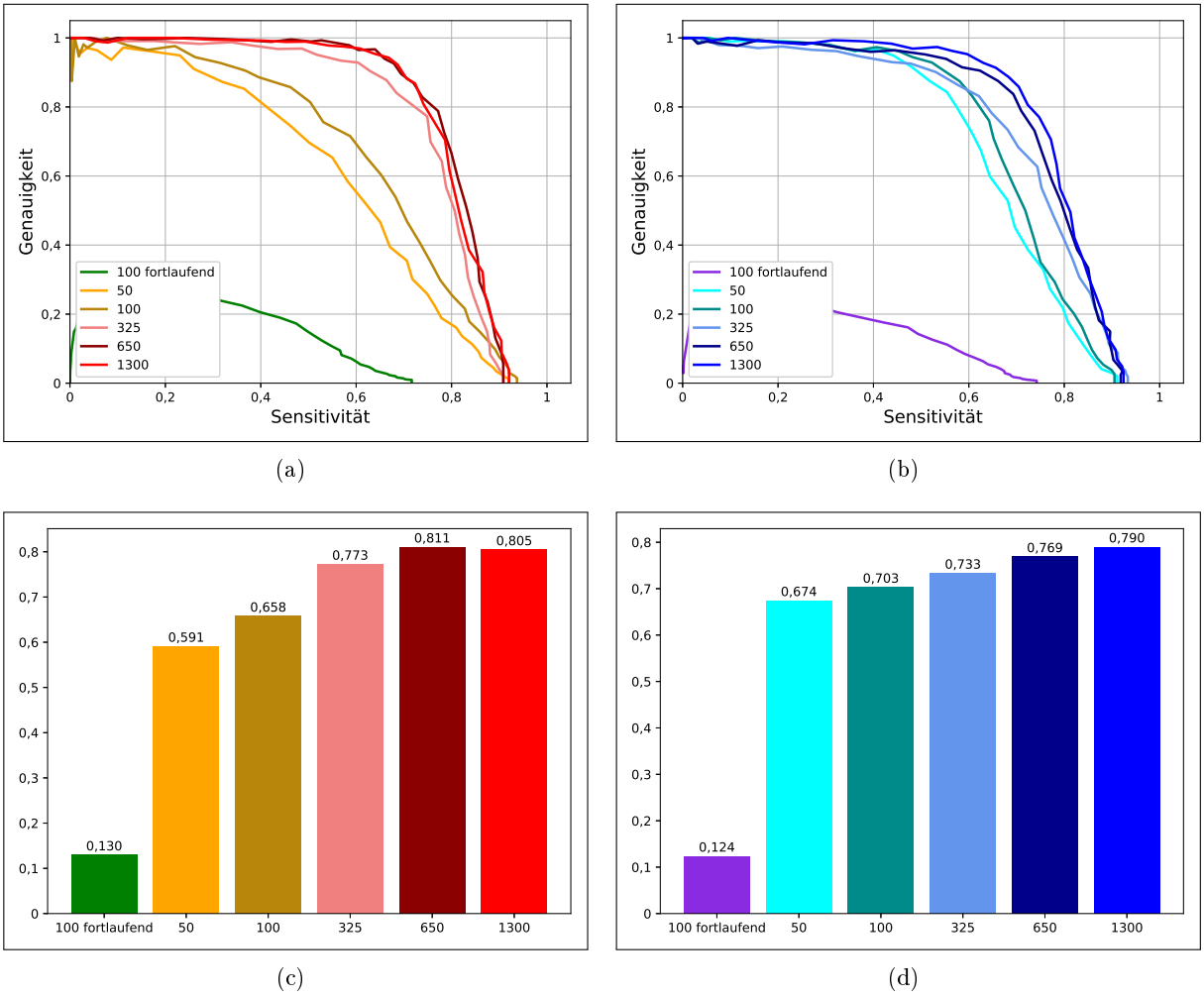


Abbildung 8.10: Ergebnisse in Abhängigkeit von Menge und Art der verwendeten Trainingsdaten. a) Standardvariante des neuronalen Netzes mit unterschiedlicher Menge und Art der Trainingsdaten für Datensätze Ettlingen 1 - 3. b) Vereinfachte Variante des neuronalen Netzes mit unterschiedlicher Menge und Art der Trainingsdaten für Datensätze Ettlingen 1 - 3. c) Fläche unter den Kurven zu a). d) Fläche unter den Kurven zu b).

des neuronalen Netzes jedoch stärker beeinträchtigt. Bei der vereinfachten Variante hingegen blieb der Leistungsverlust gering. Dies führte dazu, dass diese bei so wenigen Trainingsdaten insgesamt eine besseres Ergebnis abliefert als die Standardvariante.

Die Untersuchung des Trainings mit 100 Punktwolken in einer fortlaufenden Sequenz, anstatt zufällig ausgewählt, hat hingegen bei beiden Varianten des neuronalen Netzes, zu einem signifikant schlechteren Ergebnis geführt. Es kann gesagt werden, dass das Verfahren mit solchen Trainingsdaten nicht mehr funktioniert, was sich auch an dem erratischen Verlauf der Kurven zeigt. Das Trainingsergebnis und die Leistung des Detektionsverfahrens wird hier von Zufallseffekten dominiert.

8.1.3 Detektionsleistung in Abhängigkeit zur Entfernung zum Sensor

In Abbildung 8.11 sind die Ergebnisse der durchgeführten Untersuchungen, im Bezug auf die Abhängig des Verfahrens zur Detektion von Personen in 3D-Punktwolken, von der Entfernung

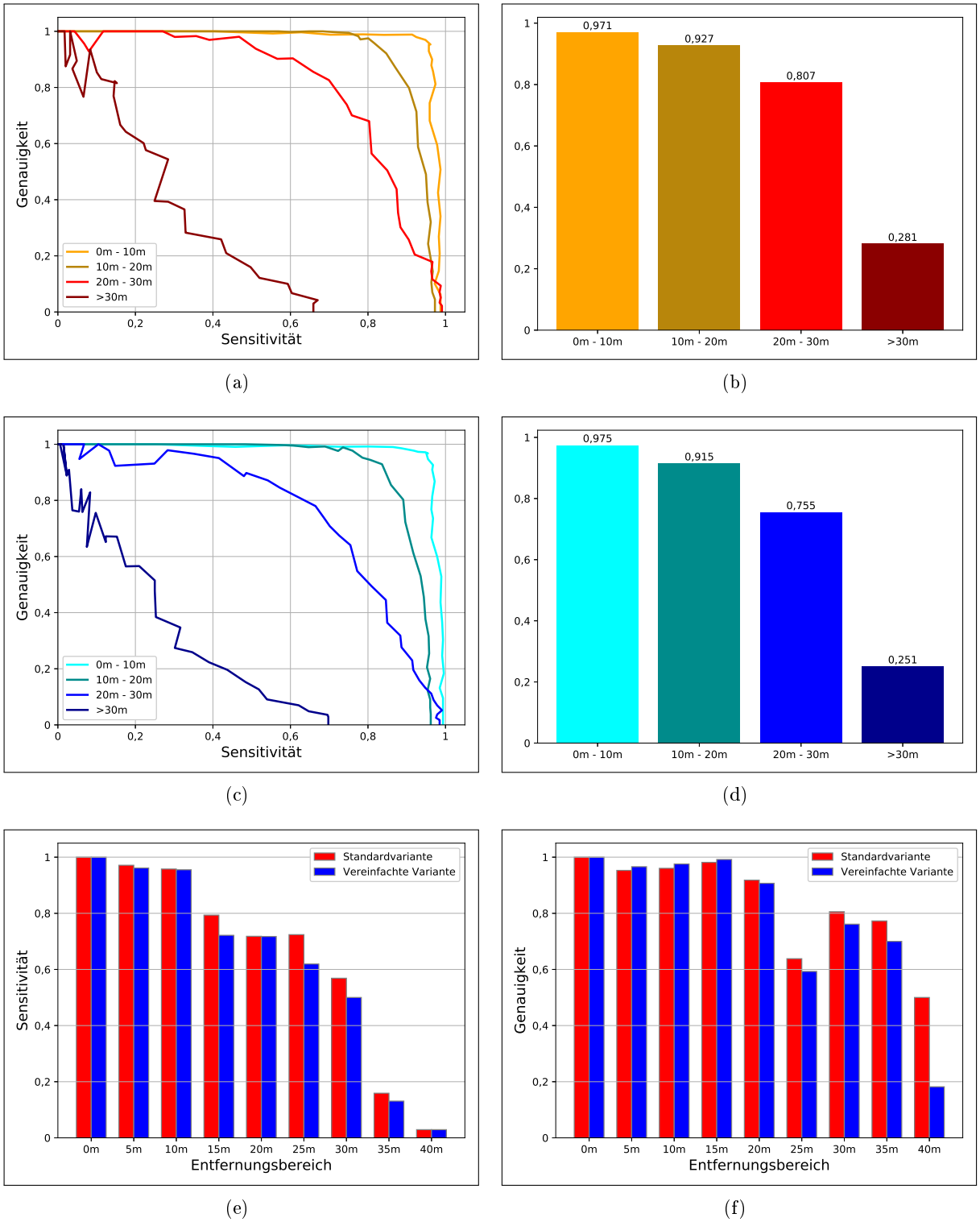


Abbildung 8.11: Ergebnisse in Abhängigkeit von der Entfernung zwischen Sensor und Person. a) Standardvariante des neuronalen Netzes in unterschiedlichen Entfernungsbereichen, b) Fläche unter den Kurven von a), c) Vereinfachte Variante in unterschiedlichen Entfernungsbereichen, d) Fläche unter den Kurven von c), e) Sensitivität in unterschiedlichen gerundeten Entfernungen bei einer Gesamtgenauigkeit von 0,9, f) Genauigkeit in unterschiedlichen gerundeten Entfernungen bei einer Gesamtgenauigkeit von 0,9.

zwischen Sensor und Person auf verschiedene Art und Weise dargestellt. a), b), c) und f) geben die Performanz des Detektionsverfahrens mit der Standardvariante und der vereinfachten Variante des neuronalen Netzes in unterschiedlichen Entfernungsbereichen wieder. Es zeigt sich, dass die Gesamtleistung bei geringer Entfernung zwischen 0 m und 10 m, also bei einer hohen Punktdichte, sehr gut und bei beiden Varianten des neuronalen Netzes nahezu identisch ist. Auch in dem nachfolgenden Entfernungsbereich von 10 m bis 20 m ist dieses Bild ähnlich und die Leistung sinkt gegenüber dem vorherigen Entfernungsbereich nur ein wenig.

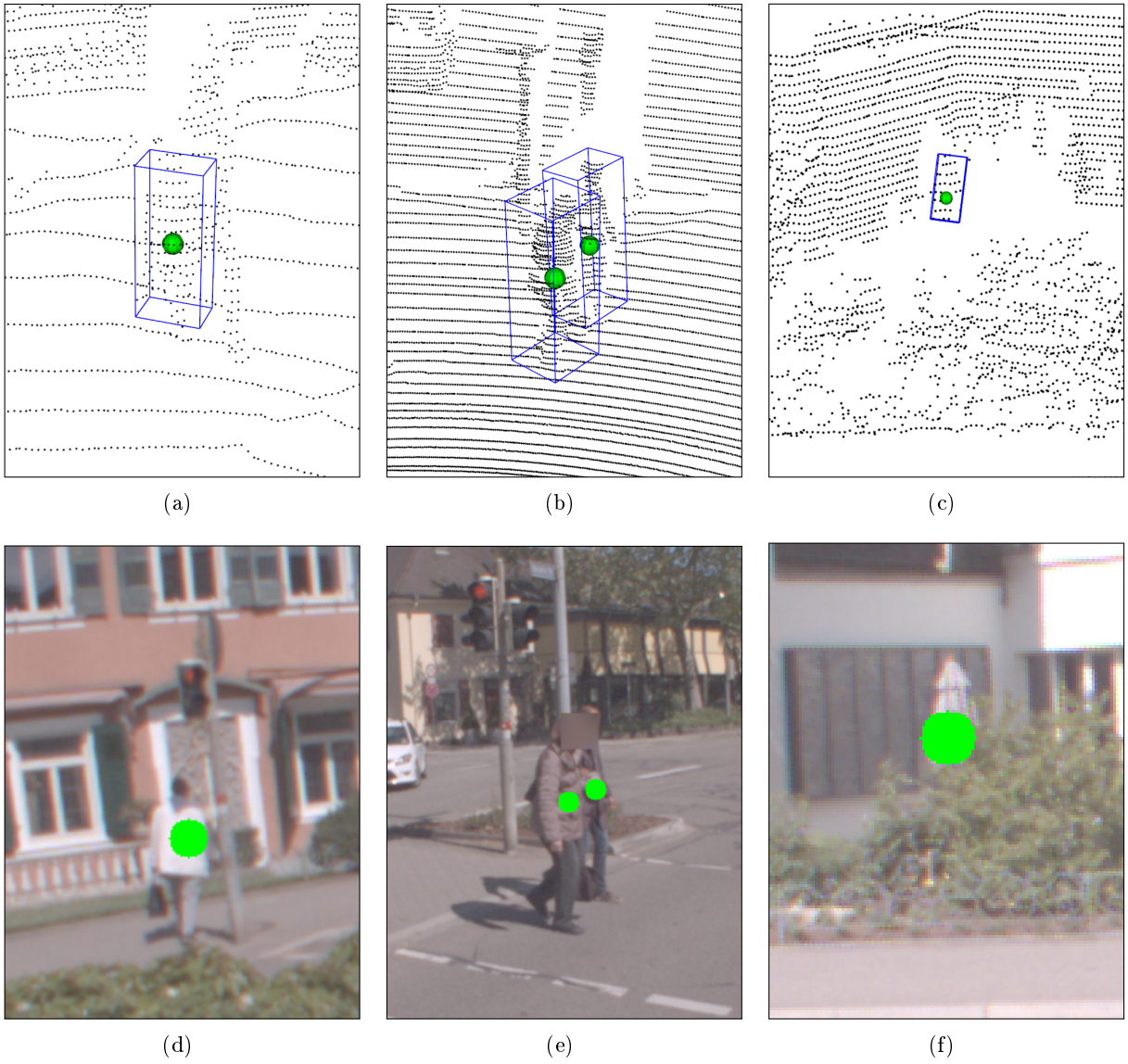
Im Entfernungsbereich von 20 m bis 30 m ist die Leistung des Verfahrens hingegen bereits deutlich geringer. Auch zeigt sich hier, dass die vereinfachte Variante des neuronalen Netzes schlechter mit der geringen Punktdichte in diesem Entfernungsbereich umgehen kann als die Standardvariante. Dies ist interessant, wenn man sich die unterschiedlichen optimierten Konfigurationen von beiden Varianten des Netzes anschaut. Der Parameter „minimale Nachbarschaftsgröße“ ist so gesetzt, dass bei der Standardvariante mindestens 10 Punkte im Radius der Nachbarschaft sein müssen, bei der vereinfachten Variante hingegen nur 3. Es werden also bei geringer Punktdichte bei der vereinfachten Variante mehr Nachbarschaften gebildet als bei der Standardvariante. Diese zusätzlichen Nachbarschaften umfassen allerdings nur sehr wenige Punkte. Obwohl diese kleinen Nachbarschaften der vereinfachten Variante des neuronalen Netzes, wie in den Untersuchungen zur Parametrisierung des Verfahrens festgestellt, insgesamt helfen, scheint dies nicht dafür zu reichen, Schwächen dieser Variante im Umgang mit einer geringen Punktdichte vollständig auszugleichen. Im Entfernungsbereich von mehr als 30 m fällt die Leistung von beiden Varianten stark ab und kann als nicht mehr ausreichend angesehen werden.

Abbildung 8.11 e) und f) zeigen die Sensitivität und Genauigkeit in unterschiedlichen Entfernungen bei einem festgelegten Schwellwert zur Detektion, der bei beiden Varianten jeweils zu einer Gesamtgenauigkeit über alle Entfernungen von ungefähr 0,9 führt. Die Entfernungen zwischen Sensor und Person wurden bei dieser Darstellung jeweils auf 5 m Schritte gerundet. Im Bezug auf die Sensitivität ist zu beobachten, dass diese zunächst langsam und dann ein wenig schneller bis ungefähr 30 m abfällt. Danach sinkt sie schnell und jenseits von rund 40 m werden überhaupt keine Personen mehr detektiert. Bei der Genauigkeit gibt es eine Besonderheit im Bereich um 25 m. Aus nicht genau bekanntem Grund ist die Genauigkeit hier deutlich geringer. Dies liegt vermutlich an einer Anomalie in den verwendeten Daten, die vor allem in diesem Entfernungsbereich auftritt und zu einer Häufung von falsch positiven Detektionen führt. Es könnte sich dabei um den in Abbildung 8.12 c) und f) dargestellten Sonnenschirm handeln, der wiederholt fälschlicherweise als Person detektiert wurde.

8.1.4 Beispielergebnisse der Personendetektion

Abbildung 8.12 zeigt Beispielergebnisse der Detektion von Personen in 3D-Punktwolken. Die detektierten Personen werden jeweils in der Punktwolke als auch projiziert auf Bilder dargestellt. Abbildung 8.12 a) und d) zeigen eine Person, die sich in der Nähe eines anderen Objektes befindet. Dabei handelt es sich um eine Ampel. Es ist zu erkennen, dass die Person erkannt wird und eine akkurate Bounding Box erzeugt wird. Abbildung 8.12 b) und e) zeigt zwei Personen, die nahe beieinander sind und beide erfolgreich erkannt werden. In dem Beispiel haben sie auch beide eine korrekte Bounding Box erhalten. Es hat sich jedoch gezeigt, dass in ähnlichen Situationen häufiger zwar die Positionen der Personen korrekt bestimmt werden, die erzeugten Bounding Boxen aber neben der korrekten auch Teile der anderen Person umfassen. Dies kommt vor, wenn eine Stimme im Abstimmverfahren (vgl. Abschnitt 4.4) beiden oder nur der falschen Person zugeordnet wird.

Ein Beispiel für eine falsch positive Detektion wird von Abbildung 8.12 c) und f) dargestellt. Es handelt sich um einen eingeklappten Sonnenschirm, der häufiger fälschlicherweise als Person detektiert wird. Der Sonnenschirm befindet sich in größerer Entfernung zum Sensor und die Punktdichte



(a) (b) (c)

(d) (e) (f)

Abbildung 8.12: Beispielergebnisse der Personendetektion. Die Ergebnisse werden sowohl als Punktwolke als auch als Projektion auf die Bilder der Rundumkameras dargestellt. a) und d) Detektion einer Person in der Nähe eines anderen Objekts. b) und e) Detektion von zwei Personen die nah beieinander sind. c) und f) Falsch positives Ergebnis, ein Sonnenschirm wird als Person detektiert.

auf dem Objekt ist daher gering. Der untere Bereich ist außerdem durch eine Hecke verdeckt. Da insbesondere die Arme einer Person bei geringer Punktdichte nicht mehr vom Oberkörper unterscheidbar sind, verbleibt als geometrisches Merkmal für den Oberkörper in solchen Situationen vor allem die generelle Körperform, die leicht gerundet ist und sich nach oben verjüngt. Dies wird durch den Sonnenschirm so gut repliziert, dass er gelegentlich zu falsch-positiven Ergebnissen führt.

8.2 Tracking von Personen in 3D-Punktwolken

In diesem Abschnitt befinden sich die Ergebnisse der Experimente im Bezug auf das in Kapitel 5 vorgestellte Verfahren zum Tracking von Personen. Es folgen zunächst Ergebnisse, welche

die Trackingleistung des Verfahrens untersuchen. Anschließend wird dargestellt, wie hilfreich ein solches Tracking ist, um mit Situationen umzugehen, in denen Objekte vorübergehend teilweise oder vollständig verdeckt sind.

8.2.1 Trackingleistung

Abbildung 8.13 stellt die Ergebnisse der verschiedenen Experimente im Bezug auf die Trackingleistung grafisch dar. Gezeigt werden jeweils die Kennzahlen $MOTP$ und $MOTA$. Da die erreichten Werte für $MOTP$ bei allen Experimenten und Konfigurationen als ausreichend angesehen werden, wurden im Bezug auf die Optimierung der Konfiguration vor allem die Ergebnisse für die Kennzahl $MOTA$ berücksichtigt.

Die Varianz des Prozessrauschens hat nur einen minimalen Einfluss auf die $MOTA$-Ergebnisse. Insgesamt kann jedoch ein Optimum bei 1,6 festgestellt werden, wie Abbildung 8.13 b) zeigt. Bei weiterer Steigerung dieser Varianz sinkt der erreichte $MOTA$-Wert jedoch stetig, weswegen 1,6 als Wert für die weiteren Experimente genommen wurde.

Die maximale Varianz der Positionsschätzung entscheidet bei dem verwendeten Trackingverfahren darüber, wie lange ein Objekt, welches nicht mehr beobachtet wird, weiter getrackt wird. Wenn dieser Wert zu groß gewählt wird, können bereits vereinzelte falsch positive Detektionen zu dann ebenfalls falschen Tracks führen, die dann relativ lange erhalten bleiben. Daher nimmt bei einem zu hohen Wert $MOTA$ immer weiter ab, wie von Abbildung 8.13 d) gezeigt wird. Ein zu niedriger Wert hingegen führt dazu, dass Tracks von nur kurzfristig verdeckten Objekten zu schnell gelöscht werden. Als Optimum wurde hier der Wert 0,425 bestimmt. Wobei davon auszugehen ist, dass dieses Optimum u.a. davon abhängig ist, wie oft und wie lange Objekte in den untersuchten Daten verdeckt sind.

Der bei der Detektion verwendete Schwellwert ist zwar kein Parameter des Trackingverfahrens selbst, hat jedoch als Teil des Gesamtverfahrens aus Detektor und Tracker dennoch einen großen Einfluss auf die Trackingleistung. Dies liegt daran, dass ein zu hoher Schwellwert dazu führt, dass viele Objekte nicht detektiert werden. Es kommt also oft zu *Misses*. Ein zu niedriger Schwellwert hingegen führt zu vielen falsch positiven Detektionen. Beides hat, wie von Abbildung 8.13 f) dargestellt, einen negativen Einfluss auf das $MOTA$-Ergebnis.

Tabelle 8.1 schlüsselt die Kennzahlen zur Leistungsfähigkeit des Trackings näher auf. Hierbei wird die Konfiguration dargestellt, die zu dem besten $MOTA$-Ergebnis über alle Datensätze und unter Berücksichtigung von verdeckten Personen geführt hat. Diese nutzt eine Varianz für das Prozessrauschen von 1,6, eine maximale Varianz der Positionsschätzung von 0,425 und einen Schwellwert für die Detektion von 0,35. Die Tabelle umfasst auch die Einzelkennzahlen aus denen sich $MOTA$ ergibt. Es zeigt sich, dass es in der Konfiguration zu keinen *Mismatches* gekommen ist. Die falsch positiv Rate war beim „Ettlingen 1“-Datensatz am höchsten während die falsch negativ Rate beim „Ettlingen 3“-Datensatz am höchsten war. Da die Personen im „Ettlingen 3“-Datensatz im Durchschnitt am weitesten vom Sensor entfernt sind, scheint die falsch negativ Rate hier u.a. die in Abschnitt 8.1.3 dargestellte begrenzte Leistungsfähigkeit des Detektionsverfahrens in größeren Entfernungen vom Sensor widerzuspiegeln.

8.2.2 Unterstützung der Personendetektionsleistung durch das Tracking

Abbildung 8.14 gibt die Ergebnisse zur Untersuchung der Detektionsleistung mit Unterstützung durch das Tracking wieder. Im Vergleich dazu werden auch die Ergebnisse ohne Nutzung des Trackings wiedergegeben. Dargestellt werden sowohl die Ergebnisse unter Berücksichtigung von nur den vollständig sichtbaren Personen, als auch die unter Berücksichtigung von allen Personen

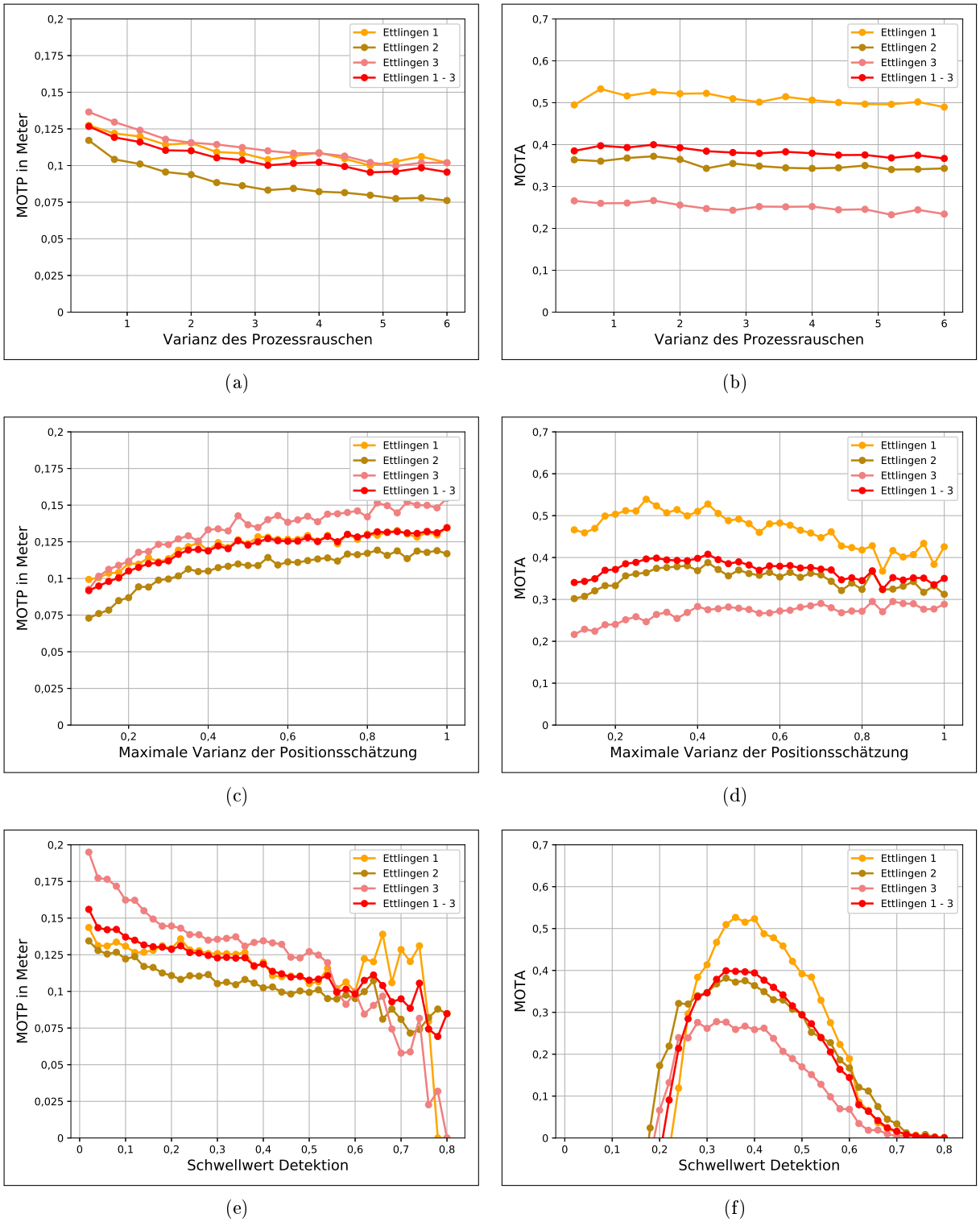
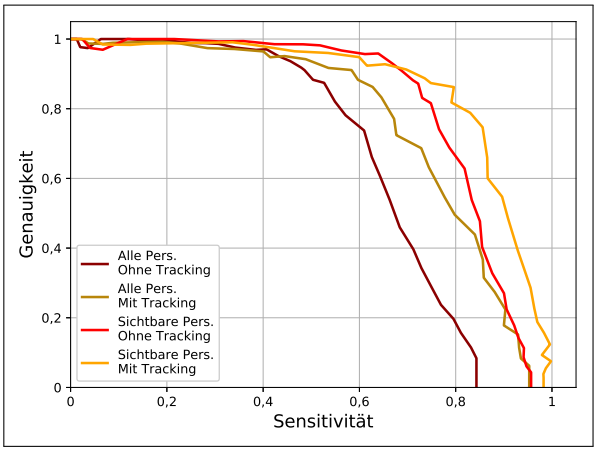
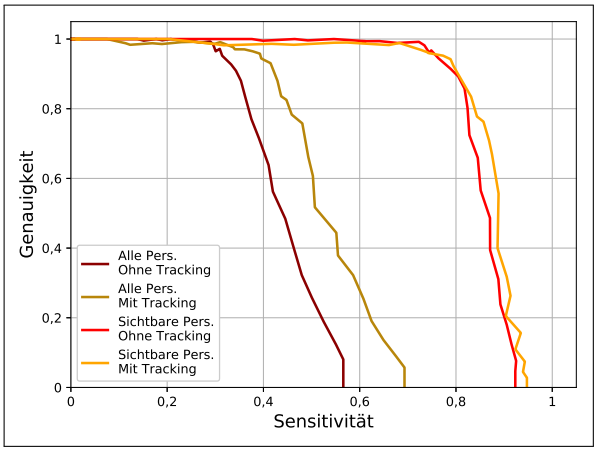


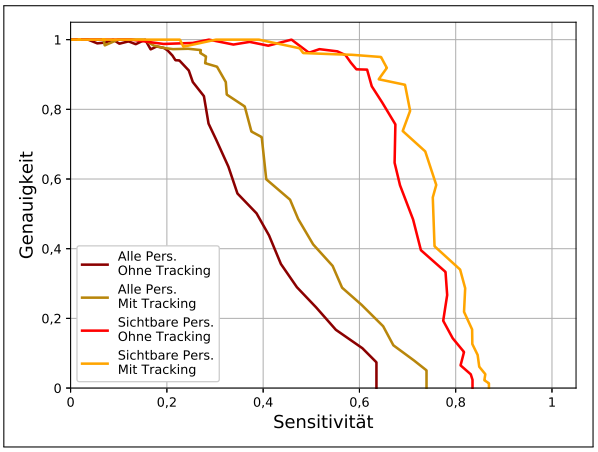
Abbildung 8.13: Ergebnisse der Experimente zur Trackingleistung. a) und b) $MOTP$ und $MOTA$ bei unterschiedlichen Werten für die Varianz des Prozessrauschens. c) und d) $MOTP$ und $MOTA$ bei unterschiedlichen Werten für die maximale Varianz der Positionsschätzung bevor ein Track entfernt wird. e) und f) $MOTP$ und $MOTA$ bei unterschiedlichen Schwellwerten im Detektionsverfahren.



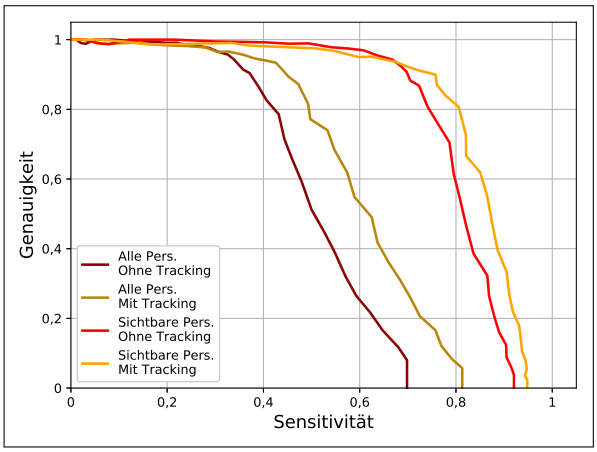
(a)



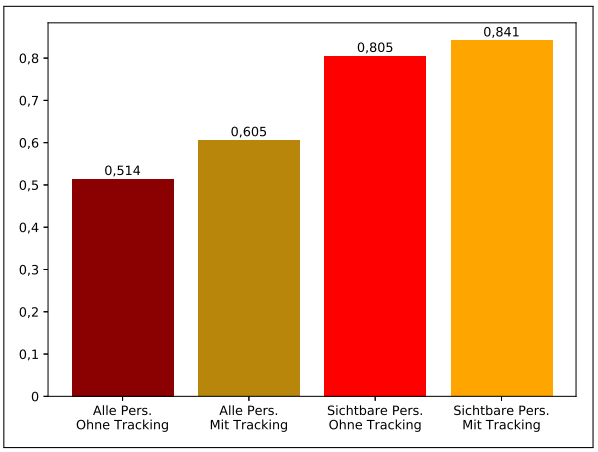
(b)



(c)



(d)



(e)

Abbildung 8.14: Ergebnisse des Vergleichs der Detektionsleistung mit und ohne Tracking. Jeweils für alle Personen im Datensatz als auch für nur die gut sichtbaren Personen. a) Ettlingen 1, b) Ettlingen 2, c) Ettlingen 3, d) Ettlingen 1 - 3 kombiniert e) Fläche unter den Kurven zu d).

Alle Personen				
Kennzahl	**Ettlingen 1**	**Ettlingen 2**	**Ettlingen 3**	**Ettlingen 1 - 3**
MOTP	0,12 m	0,11 m	0,13 m	0,12 m
MOTA	0,528	0,388	0.275	0,408
Falsch negativ Rate	35,84 %	58,46 %	70,14 %	53,14 %
Falsch positiv Rate	11,39 %	2,75 %	2,32 %	6,08 %
Mismatch Rate	0,00 %	0,00 %	0,00 %	0,00 %
Vollständig sichtbare Personen				
Kennzahl	**Ettlingen 1**	**Ettlingen 2**	**Ettlingen 3**	**Ettlingen 1 - 3**
MOTP	0,10 m	0,07 m	0,09 m	0,09 m
MOTA	0,583	0,734	0.582	0,624
Falsch negativ Rate	16,35 %	18,80 %	32,37 %	20,57 %
Falsch positiv Rate	25,37 %	7,75 %	9,42 %	16,99 %
Mismatch Rate	0,00 %	0,00 %	0,00 %	0,00 %

Tabelle 8.1: Kennzahlen zur Leistung des Trackings in der Konfiguration, die über alle Datensätze zu dem besten *MOTA*-Wert geführt hat.

in dem jeweiligen Datensatz. Letzteres umfasst also auch Personen, die vorübergehend vollständig verdeckt sind und daher gar nicht vom Detektionsverfahren alleine erfasst werden können.

Es zeigt sich, dass das Tracking die Ergebnisse auch dann verbessert, wenn nur gut sichtbare Personen berücksichtigt werden, da es vereinzelte falsch negative Detektionen ausgleicht. Der Vorteil des Trackings wird größer, wenn auch die schlecht oder gar nicht sichtbaren Personen berücksichtigt werden. Zwar ist es immer noch erforderlich, diese Personen zumindest für einige Zeit detektiert zu haben, aber wenn sie z.B. kurzfristig hinter einem Hindernis verschwinden, kann ihre Position für eine kurze Zeit weiterhin bestimmt werden.

8.3 Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern

Die Ergebnisse der Experimente zur Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern befinden sich in diesem Abschnitt. Der Fokus liegt zunächst auf der Detektionsleistung der drei untersuchten Varianten. Anschließend wird ihre Laufzeit untersucht. Zuletzt folgt eine qualitative Betrachtung der ermittelten Körperposen.

8.3.1 Detektionsleistung der drei untersuchten Varianten

Abbildung 8.15 zeigt die Ergebnisse zur Detektionsleistung der drei Varianten. Bei den Varianten 2 und 3, bei denen die Leistung der Detektion von Personen in den Bildern mithilfe des Verfahrens zur Körperposenschätzung relevant ist, fällt auf, dass diese mit Personen in größerer Entfernung zum Sensorsystem schlechter umgehen kann als das Verfahren zur Detektion von Personen in 3D-Punktwolken, was auch durch den Vergleich von Abbildung 8.15 a) und b) mit c) und d) herausgestellt wird. Dies ist natürlich auch von den verwendeten Kameras abhängig. Für die Experimente wurden Bilder der Rundumkameras des Experimentalsystems verwendet. Diese sind reich weitwinkelig, um bereits in kurzer Entfernung eine Erfassung des gesamten Fahrzeugsumfelds zu erlauben. Dadurch umfassen Personen in größerer Entfernung aber auch nur noch wenige Pixel, was das Verfahren zur Posenschätzung dann an seine Grenzen bringt.

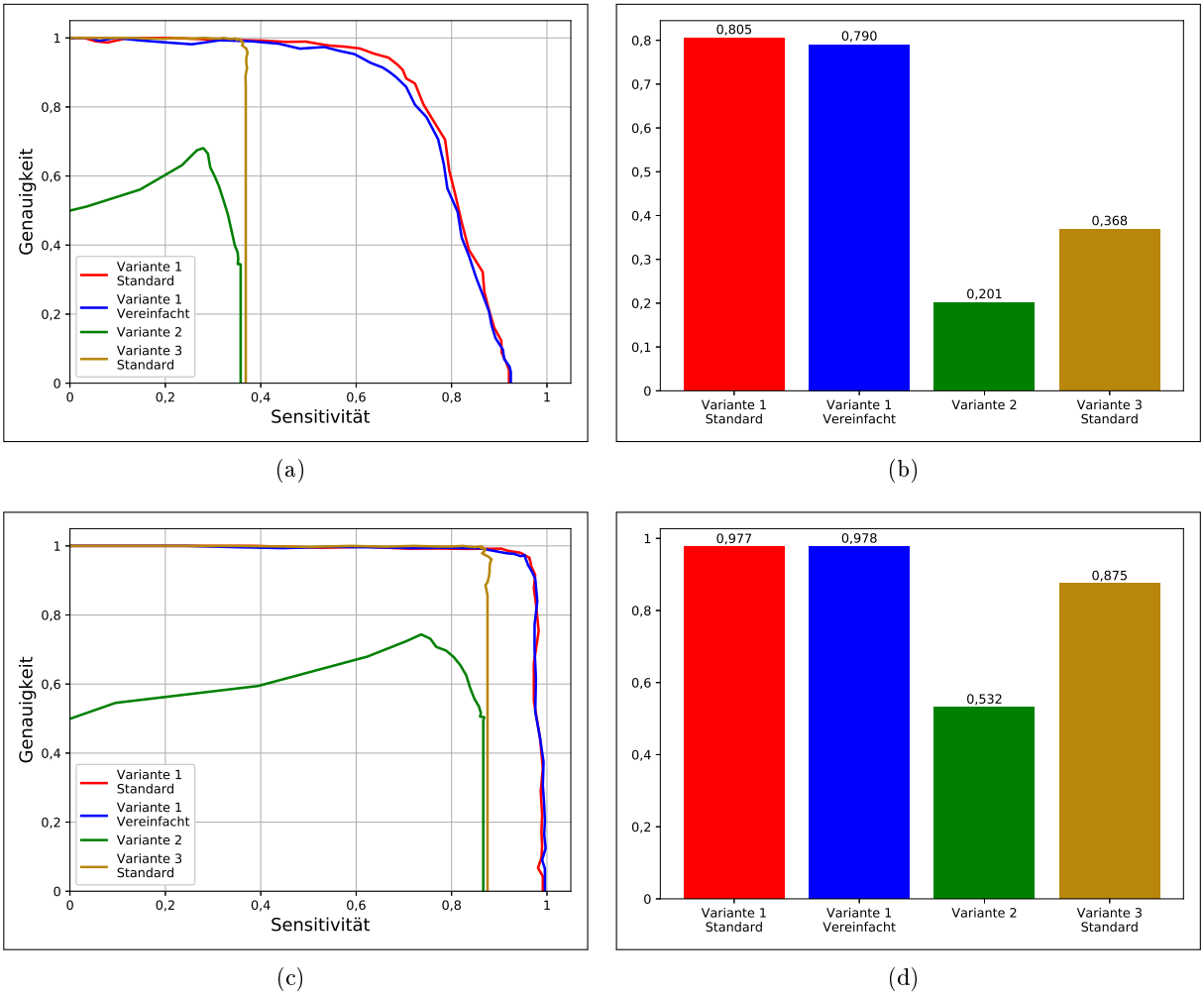
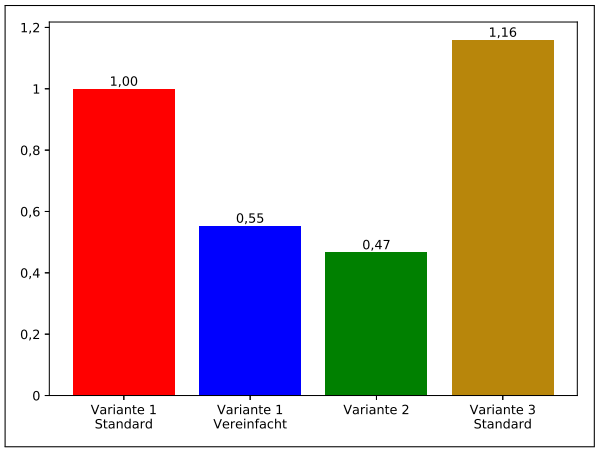


Abbildung 8.15: Detektionsleistung der drei Varianten zur Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern. Bei der Variante 1 wurden außerdem beide Varianten des neuronalen Netzes verglichen. a) und b) Ergebnisse für Personen in allen Entfernungen, c) und d) Ergebnisse für Personen bis zu 15 m vom Sensor entfernt.

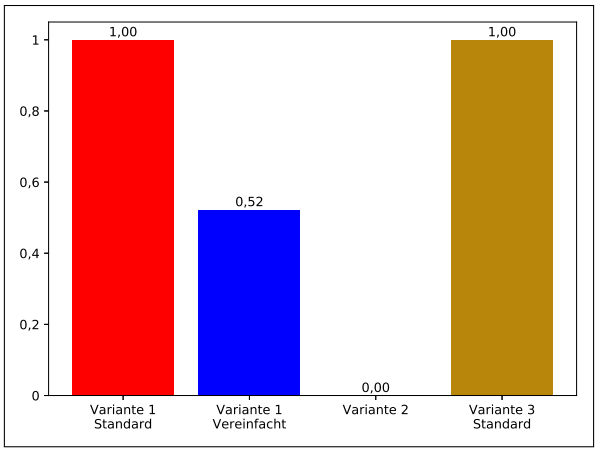
Variante 3 führt zu einer nahezu perfekten Genauigkeit, was mit den Erwartungen übereinstimmt. Dadurch, dass sich die beiden Verfahren hier gegenseitig bestätigen, kann es nur dann zu einer Fehldetektion kommen, wenn beide Verfahren an derselben Stelle ein falsch positives Ergebnis liefern. Dies ist aufgrund der Andersartigkeit der genutzten Verfahren und Daten sehr unwahrscheinlich. Im Umkehrschluss ist die Sensitivität aber auch durch beide Verfahren begrenzt.

Interessant ist hier noch, dass Variante 3 eine leicht höhere maximale Sensitivität hat als Variante 2. Dies ist unerwartet, da die maximale Sensitivität von Variante 3 eigentlich nicht höher sein kann als die von Variante 2. Die Erklärung hierfür liegt in der Bestimmung der 3D-Koordinaten für die in den Bildern detektierten Posen und Personen. Variante 2 nutzt dafür ein aus der Punktwolke generiertes Tiefenbild und berücksichtigt so die komplette Punktwolke. Variante 3 hingegen berücksichtigt nur die Punkte, für die zuvor vom Verfahren zur Detektion von Personen in Punktwolken festgestellt wurde, dass sie zu einer Person gehören. Diese Methode ist weniger fehleranfällig in Situationen, in denen sich zwischen Person und Sensor noch andere Objekte befinden. Bei Variante 2 wird in solchen Situationen für eine in den Bildern korrekt detektierte Person manchmal eine falsche 3D-Koordinate bestimmt.

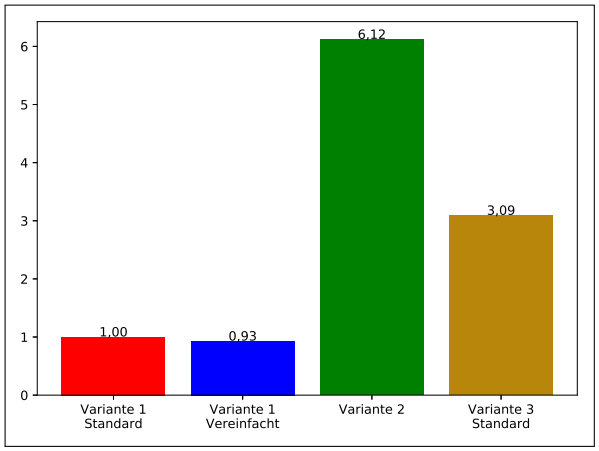
8.3.2 Laufzeit der drei untersuchten Varianten



(a)



(b)



(c)

Abbildung 8.16: Laufzeit der drei Varianten zur Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern. Laufzeit wird relativ zur Laufzeit der Variante 1 mit dem Standard neuronalen Netz angegeben. a) Durchschnittliche Gesamtlaufzeit, b) Durchschnittliche Laufzeit der Detektion in 3D-Punktwolken, c) Durchschnittliche Laufzeit der Posenschätzung und des Bestimmens von 3D-Koordinaten für die Posen.

Abbildung 8.16 vergleicht die Laufzeit der drei untersuchten Varianten zur Detektion und Körperposenschätzung. Variante 2, bei der sowohl die Posenschätzung als auch die Detektion rein in den Bildern erfolgt und die Punktwolken nur dazu dienen, 3D-Koordinaten für die Ergebnisse zu bestimmen, ist insgesamt am schnellsten. Variante 1 unter Nutzung des vereinfachten Modells ist ein wenig langsamer. Bei Variante 1 und 3 setzt sich die Laufzeit aus der Detektion der Personen in den Punktwolken und aus der Körperposenschätzung in den Bilddaten zusammen. Letztere umfasst dabei auch das Bestimmen von 3D-Koordinaten für die Ergebnisse der Posenschätzung. Variante 2 nutzt hingegen das Detektionsverfahren in den Punktwolken nicht, weswegen sich hier die Laufzeit allein aus der Körperposenschätzung ergibt.

Abbildung 8.16 b) und c) schlüsselt die Laufzeiten der beiden Komponenten einzeln auf. Bei Variante 1 erfolgt die Posenschätzung nur auf generierten Bildausschnitten für zuvor detektierte Personen. Bei den Varianten 2 und 3 hingegen erfolgt sie auf den vollständigen Bildern. Wie in Abbildung 8.16 b) zu sehen ist, führt die Nutzung dieser Bildausschnitte zu einer Zeitersparnis bei

der Posenschätzung. Auch zeigt sich im Vergleich der Varianten 2 und 3, dass das Bestimmen von 3D-Koordinaten für die Ergebnisse der Posenschätzung signifikant aufwendiger ist, wenn hierbei die komplette Punktwolke berücksichtigt werden muss, anstatt nur die Punkte zu berücksichtigen von denen bekannt ist, dass sie zu einer Person gehören.

8.3.3 Qualität der Posenschätzungsergebnisse

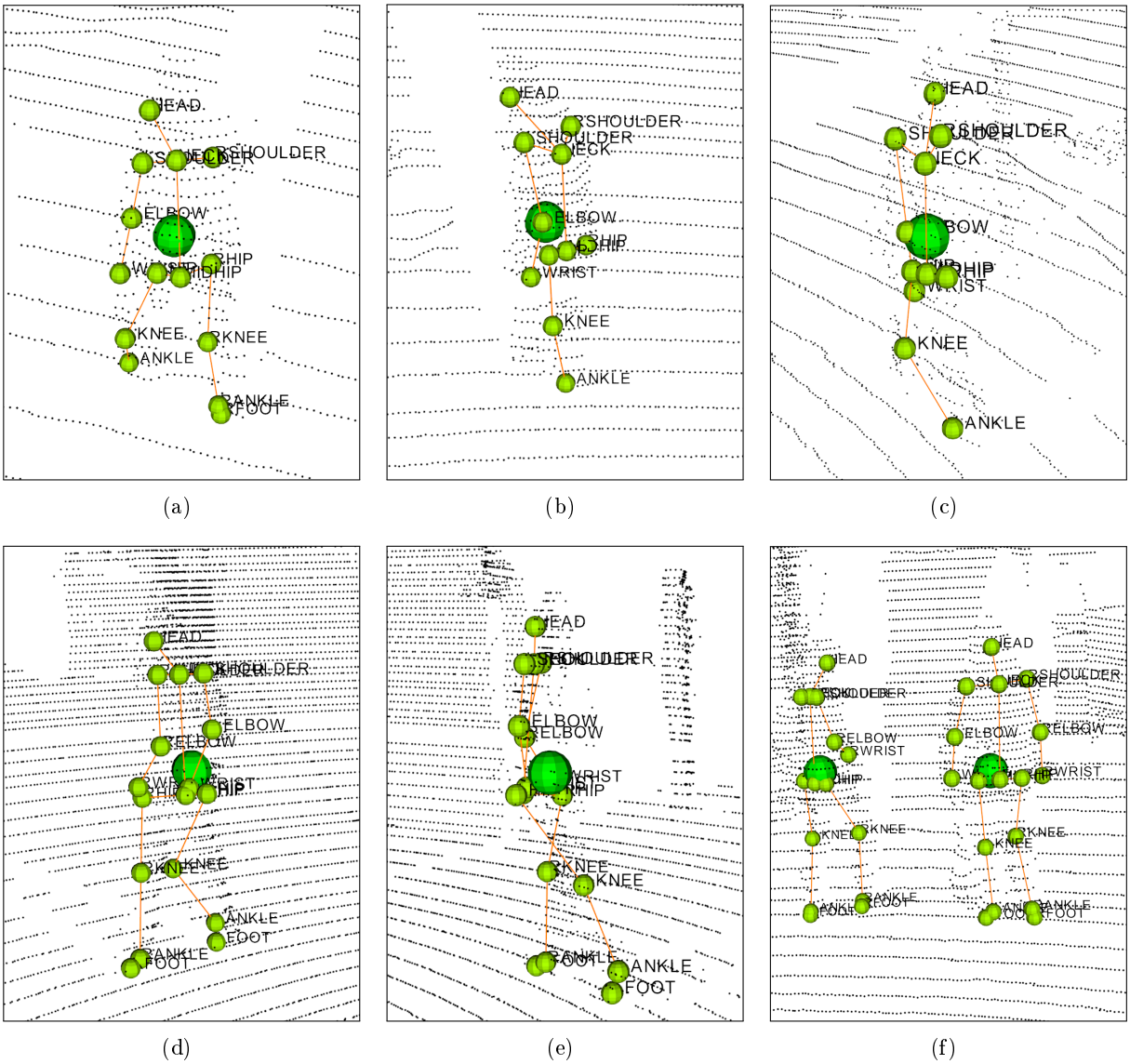


Abbildung 8.17: Beispielergebnisse der Posenschätzung mit einigen Problemfällen. Genutzt wurde die Variante 1. a) Gutes Ergebnis. b) und c) dieselbe Person aus unterschiedlichen Perspektiven da die Person in der Punktwolke nur nahezu senkrecht von einer Seite erfasst wurde, landen die 3D-Koordinaten der Posenschlüsselpunkte fälschlicherweise ebenfalls alle auf dieser Seite. d) und e) Ähnliches Problem wie bei b) und c), außerdem passt die Beinposition in der Posenschätzung nicht ganz zur Position in der Punktwolke. f) Gutes Ergebnis mit 2 Personen.

Die Qualität der Ergebnisse der Posenschätzung übertragen in den 3D-Raum wurde betrachtet, um mögliche qualitative Probleme zu entdecken. Abbildung 8.17 zeigt Beispiele der dabei gemachten Feststellungen. Es zeigt sich, dass die Genauigkeit der ermittelten Posenpunkte im

3D-Raum u.a. vom Blickwinkel, den der LiDAR-Sensor auf die Person hat, abhängt. Aufgrund des angewendeten Prinzips zur Bestimmung dieser 3D-Koordinaten landen diese immer in der Nähe der durch die Punktwolke erfassten Oberfläche. Je nach Blickwinkel des LiDAR-Sensors führt dies zu unterschiedlich starken Abweichungen von der tatsächlichen Position des durch den Posenschlüsselpunkt repräsentierten Körperteils. Abbildung 8.17 b) und c) sowie d) und e) zeigen jeweils dieselbe Person zum selben Zeitpunkt aus unterschiedlichen Blickwinkeln, um dieses Problem zu verdeutlichen.

Abbildung 8.17 d) zeigt beim genauen Betrachten der Position des Knies noch ein anderes Problem. Der Posenschlüsselpunkt des einen Knies passt nicht genau zu der Punktwolke. Dies liegt daran, dass es nicht möglich ist eine Punktwolke, die über einen Zeitraum von 0,1 s aufgenommen wurde, perfekt zeitlich mit einem Bild zu synchronisieren. Es gibt daher meist eine kleine zeitliche Diskrepanz zwischen der Erfassung der Person in der Punktwolke und im Bild. In dieser Zeit hat sich das Bein der dargestellten Person bereits ein wenig weiterbewegt, weswegen die Positionen des Knies in Bild und Punktwolke nicht übereinstimmen.

9 Diskussion

In diesem Kapitel werden die im vorherigen Kapitel vorgestellten Ergebnisse der Experimente sowie die Leistung der in diesen Experimenten untersuchten Methoden diskutiert. Es gliedert sich in vier Abschnitte. Zunächst werden die Methoden und die für sie durchgeführten Experimente einzeln erläutert. Anschließend gibt es eine Diskussion, in der die Ergebnisse in den Gesamtkontext eines Systems gestellt werden, das die in der Einleitung dieser Arbeit motivierten Aufgaben erfüllen kann.

9.1 Detektion von Personen in 3D-Punktwolken

Das im Rahmen dieser Arbeit entworfene und vorgestellte Verfahren zur Detektion von Personen in 3D-Punktwolken hängt von einer Reihe von Parametern ab, die im Rahmen der Experimente zunächst optimiert wurden. Dabei hat sich gezeigt, dass die Parameter den größten Einfluss auf das Verfahren haben, die dessen Fähigkeit beeinflussen mit einer geringen Datendichte umzugehen. Dies sind im Einzelnen: Der Radius der generierten lokalen Punktnachbarschaften, die minimale Anzahl an Punkten in einer validen Nachbarschaft und die Sub-Sampling-Rate. Sowohl der Radius als auch die minimale Anzahl an Punkten in einer validen Nachbarschaft können dafür sorgen, dass bei geringer Punktdichte keine validen Nachbarschaften generiert werden, da diese zu wenig Punkte umfassen. Das Sup-Sampling hingegen verringert die Anzahl an gebildeten Nachbarschaften insgesamt. Dies hat weniger Auswirkungen in Bereichen der Punktwolke, wo die Punktdichte hoch ist und daher sowieso eher mehr Nachbarschaften als nötig gebildet werden. In Bereichen mit geringer Punktdichte ist dies hingegen anders. Dieser Effekt kann vermieden werden, wenn statt einer zufälligen Auswahl für das Sup-Sampling eine Auswahl erfolgt, welche die lokale Punktdichte berücksichtigt. Eine solche ist zwar aufwändiger und verringert daher ggf. den Laufzeitgewinn, der durch das Sub-Sampling erzielt wird, erlaubt im Gegenzug aber vielleicht eine durchschnittlich höhere Sup-Sampling-Rate, ohne die Leistung des Verfahrens zu stark zu beeinträchtigen.

Im Hinblick auf die beiden untersuchten Varianten des neuronalen Netzes fällt auf, dass die komplexere Standardvariante zwar im Vergleich zur vereinfachten Variante eine etwas bessere Detektionsleistung hat, dies aber durch eine nahezu verdoppelte Laufzeit erkauft wird. Wenn nur wenig Trainingsdaten zur Verfügung stehen, verschwindet zudem auch der Vorteil der Standardvariante bei der Detektionsleistung. Für eine praktische Anwendung muss daher entschieden werden, ob die Nachteile der Standardvariante den kleinen Vorteil bei der Detektionsleistung rechtfertigen. Allgemeiner gesprochen sollte bei neuronalen Netzen darauf geachtet werden, ob der Preis, der für eine höhere Komplexität des Netzes in Form von Laufzeit und höheren Aufwand beim Training gezahlt wird durch die dadurch verbesserte Leistung gerechtfertigt ist. Im Umkehrschluss kann untersucht werden, ob die Reduktion der Komplexität eines neuronalen Netzes ohne zu starken Leistungsverlust möglich ist, wenn ein Verfahren, welches ein solches Netz nutzt beschleunigt werden soll oder wenn nur wenig Daten für das Training zur Verfügung stehen.

Eine Fragestellung dieser Arbeit führte zu der Untersuchung wie stark die Leistung eines Verfahrens zur Detektion von Personen in 3D-Punktwolken von der Menge der verwendeten Trainings-

daten abhängt. Die in Abschnitt 8.1.2 vorgestellten Ergebnisse zeigen, dass selbst eine deutliche Reduktion dieser Menge insbesondere bei der vereinfachten Variante des neuronalen Netzes die Leistung des Gesamtverfahrens nicht übermäßig stark beeinträchtigt. Ein Grund hierfür ist, dass das in dieser Arbeit genutzte Verfahren Punktwolken nicht als Ganzes in einem neuronalen Netz verarbeitet, sondern diese in relativ kleine lokale Nachbarschaften zerlegt. Aus einer einzelnen für das Training annotierten Punktwolke können sehr viele dieser lokalen Nachbarschaften gebildet werden. Auch benötigt das neuronale Netz eine geringere Komplexität, um diese relativ kleinen Nachbarschaften anstatt einer ganzen Punktwolke zu verarbeiten und dabei sinnvolle Ergebnisse zu liefern. Neuronale Netze, die eine geringere Komplexität also weniger zu trainierenden Parameter haben, lassen sich üblicherweise auch mit weniger Daten trainieren. Dies erklärt auch, warum das Vorhandensein von nur sehr wenigen Trainingsdaten die vereinfachte Variante des neuronalen Netzes weniger stark beeinträchtigen als die Standardvariante.

Die Verwendung von nur sehr wenigen Trainingsdaten stellt jedoch dann ein Problem dar, wenn diese nicht mehr zufällig aus dem gesamten Pool der für das Training zur Verfügung stehenden Daten ausgewählt werden. Wird die gleiche Menge an Trainingsdaten aus einer fortlaufenden Sequenz genutzt, führt dies zu signifikant schlechteren Ergebnissen. Der Unterschied ist hier, dass die zufällige Auswahl an Trainingsdaten immer noch zu einer guten Repräsentation von unterschiedlichen Personen in unterschiedlichem Kontext führt. Die fortlaufende Sequenz hingegen umfasst viele sehr ähnliche Punktwolken. Der beim Training des neuronalen Netzes hieraus gewonnene Informationsgehalt ist entsprechend deutlich geringer. Aus diesem Umstand können Schlüsse für das Vorgehen beim Erzeugen solcher Trainingsdaten gezogen werden. Dieser Prozess erfordert oft viel menschliche Arbeit, da Daten von Hand annotiert werden müssen. Statt lange fortlaufende Sequenzen zu annotieren, ist es vorteilhaft kleinere Sequenzen oder gar einzelne Punktwolken aus einem größeren Satz an aufgezeichneten Daten zu annotieren. Diese können hierfür z.B. zufällig ausgewählt werden. Noch besser ist es vermutlich, gezielt möglichst unterschiedliche Daten für das Annotieren auszuwählen. Mit dem gleichen Aufwand beim Annotieren von Daten kann so das erreichbare Trainingsergebnis verbessert werden.

Ein anderer untersuchter Aspekt ist, wie abhängig das Verfahren zur Detektion von der Dichte der Punktwolke ist. Diese Dichte ist üblicherweise von der Entfernung zum Sensor abhängig, weswegen der Fokus der Experimente auf der Detektionsleistung im Verhältnis zu dieser Entfernung lag. Bei diesen Experimenten hat sich gezeigt, dass die Detektionsleistung des Verfahrens in dem Bereich von 0 m bis 10 m sehr gut ist. Auch danach bleibt sie zunächst gut, fällt aber dann bei etwa 30 m schnell ab. In der in den Experimenten verwendeten Konfiguration haben die genutzten LiDAR-Sensoren eine vertikale Winkelauflösung von etwa 0,4° und eine horizontale von etwa 0,17°. Daraus ergibt sich, dass die Punkte in 30 m Entfernung vertikal ca. 0,21 m und horizontal ca. 0,09 m auseinander liegen. Bei einer 1,80 m großen Person, ergeben sich also horizontal bestenfalls noch 8,57 Scanzeilen, was sich weiter reduziert, wenn sich die Person nicht genau im senkrechten Winkel zum Sensor befindet. Gerade kleinere Strukturen des menschlichen Körpers, wie z.B. der Kopf oder die Extremitäten, werden bei zunehmender Entfernung jenseits von 30 m immer häufiger nicht mehr richtig vom Sensor erfasst, weswegen eine Detektion von Personen anhand ihres Erscheinungsbildes schwieriger und letztlich unmöglich wird.

Jenseits des Entfernungsbereichs, in dem eine Person anhand ihres äußeren Erscheinungsbilds in einer Punktwolke erkannt werden kann, sind alternative Detektionsverfahren denkbar. Eine Möglichkeit wäre hier ein DATMO-Verfahren, welches sich bewegende Objekte detektiert. Eine Schwierigkeit ist hier, eine Unterscheidung zwischen verschiedenen Arten von sich bewegenden Objekten zu treffen. Auch lassen sich unbewegliche Objekte so nicht detektieren. Durch eine Kombination von beiden Ansätzen, nämlich der Detektion anhand des Erscheinungsbildes und der Detektion als bewegliches Objekt, könnten ihre jeweiligen Stärken kombiniert werden.

Abschließend kann gesagt werden, dass das vorgestellte Verfahren zur Detektion von Personen in der Lage ist, die eingangs formulierte Aufgabenstellung zumindest in Bereichen mit ausreichender Punktdichte zu erfüllen. Dies ist bei der in den Experimenten eingesetzten Sensorik bis etwa 20 m und im leicht verminderten Umfang bis etwa 30 m der Fall. Es kann davon ausgegangen werden, dass leistungsfähigere LiDAR-Sensoren mit einer höheren Winkelauflösung diesen Entfernungsbereich steigern werden. Es hat sich auch gezeigt, dass durch die Kombination eines neuronalen Netzes mit einer relativ geringen Komplexität und einem nachgelagerten Abstimmverfahren gute Ergebnisse bei der Objektdetektion erzielt werden können, während es verhältnismäßig leicht ist, das neuronale Netz zu trainieren. Dies ist insbesondere der Fall, wenn die einfachere der beiden untersuchten Varianten des neuronalen Netzes genutzt wird. Dies dürfte auch im Hinblick auf die signifikant bessere Laufzeit dieser Variante in vielen Fällen empfehlenswert sein.

9.2 Tracking von Personen in 3D-Punktwolken

In dieser Arbeit wird ein klassisches Trackingverfahren basierend auf einen Kalman-Filter verwendet. Das Tracking erlaubt es zusätzliche Informationen zu erfassen, die durch ein Detektionsverfahren alleine nicht vorhanden wären. Diese umfassen eine für den Zeitraum der Erfassung konsistente Objektidentität, die Objektgeschwindigkeit und die Bewegungsrichtung. Neben diesen zusätzlichen Informationen steigert das Tracking aber auch die Detektionsleistung des Verfahrens. Dies wurde in den Experimenten herausgestellt. Das Tracking erlaubt es, für z.B. durch Verdeckungen vorübergehend nicht sichtbare Personen eine Position zu bestimmen. Hierfür wird Wissen über das vorherige Bewegungsverhalten dieser Personen genutzt. Dadurch ist das Tracking zwar nicht in der Lage, immer oder beliebig lange eine annähernd korrekte Position für solche Personen zu bestimmen, aber da für vollständig verdeckte Personen durch ein Detektionsverfahren alleine überhaupt keine Position bestimmt werden kann, stellt es dennoch eine Verbesserung dar. Auch bei teilweise verdeckten Personen hilft das Tracking die Detektionsleistung zu verbessern. Diese werden zwar teilweise auch noch durch das Detektionsverfahren erfasst, aber abhängig davon wie stark die Verdeckung ist, sinkt auch bei ihnen die Leistung die das Verfahren zur Detektion alleine erreicht.

Die Experimente haben auch gezeigt, dass das Tracking sogar bei gut sichtbaren Personen das Verfahren zur Detektion so unterstützt, dass sich die Gesamtleistung des Systems verbessert. Es kann daher gesagt werden, dass zumindest wenn Objekte erfasst werden sollen, die sich ggf. auch bewegen, ein Detektionsverfahren durch ein Trackingverfahren ergänzt werden sollte. Es lassen sich dadurch zusätzliche Informationen gewinnen und die Leistung des Gesamtsystems im Hinblick auf die Detektion von Objekten ist besser als durch ein Detektionsverfahren alleine.

9.3 Körperposenschätzung und Detektion von Personen in 3D-Punktwolken und RGB-Bildern

Im Hinblick auf die Körperposenschätzung ist die Idee in dieser Arbeit, LiDAR-Sensoren und Kameras zu kombinieren, um mehr und bessere Informationen über die Umgebung zu erfassen. Konkret wurde das Verfahren zur Detektion von Personen in Punktwolken mit einem dem Stand der Technik entsprechenden Verfahren zur Körperposenschätzung in Bildern kombiniert. Das dabei gewählte Vorgehen ist jedoch nicht auf die beiden konkreten genutzten Verfahren beschränkt und kann auf andere ähnliche Kombinationen von Auswertungsverfahren für Bilder und Punktwolken übertragen werden.

Es wurden drei Varianten zur Kombination der beiden Sensormodalitäten vorgestellt und experimentell untersucht. Diese unterscheiden sich darin, welche Aufgaben von welcher Sensormodalität

übernommen wird und darin, wie die Datenfusion zwischen den beiden Modalitäten durchgeführt wird. Variante 1 nutzt die LiDAR-Sensoren als Sensorik für die Detektion von Personen, um dann basierend darauf Bildausschnitte für die gezielte Körperposenschätzung dieser Personen zu bilden und auszuwerten. Im Hinblick auf die reine Detektion erreicht diese Variante daher dieselben Ergebnisse wie das Detektionsverfahren alleine. Es hat sich in den Experimenten außerdem gezeigt, dass der Zeitaufwand für die Körperposenschätzung in den Bildern reduziert werden kann, wenn nicht die vollständigen Bilder, sondern nur die generierten Bildausschnitte verarbeitet werden. Dies gilt aber vermutlich nicht mehr, wenn eine sehr große Zahl an Personen in der Umgebung sind, was aber im vorgesehenen Einsatzgebiet, dem urbanen Verkehrsraum, nur selten vorkommen sollte.

Die zweite untersuchte Variante verarbeitet die vollständigen Bilder in der Körperposenschätzung und nutzt deren Ergebnisse auch für die Detektion der Personen. Die Ergebnisse werden dann in den 3D-Raum übertragen, wobei die Punktwolken der LiDAR-Sensoren genutzt werden, um 3D-Koordinaten für die Posen-Schlüsselpunkte zu bestimmen. Im Hinblick auf die Detektionsleistung ist dieses Vorgehen auf die Leistung der Körperposenschätzung beschränkt. Es hat sich gezeigt, dass diese bei den verwendeten Kameras noch mehr Probleme mit Personen in größerer Entfernung hat als die Detektion in den Punktwolken. Die Detektionsleistung war daher insbesondere in diesem Entfernungsbereich signifikant schlechter als bei der Variante 1. Dies kann aber nicht als grundsätzliches Argument gegen auf Bildern basierende Detektionsverfahren gesehen werden. Das genutzte Verfahren ist primär für die Körperposenschätzung gedacht und es ist davon auszugehen, dass ein spezialisiertes Verfahren zur Personendetektion in Bildern bessere Ergebnisse erzielen würde. Auch hängt die Leistung der Verfahren natürlich auch von den verwendeten Sensoren ab. Bilder, welche die Personen auch in größerer Entfernung höher aufgelöst und besser darstellen, würden zu besseren Ergebnissen führen. Dies könnte leicht durch andere Kameras oder Objektive erreicht werden. Wobei dann ggf. mehr Kameras für eine vollständige Erfassung des Fahrzeugsumfelds notwendig wären. Im direkten Vergleich zwischen den drei Varianten zeigt sich auch, dass das Bestimmen von 3D-Koordinaten für die zunächst nur in 2D-Bildkoordinaten vorliegenden Posenschlüsselpunkte effizienter und weniger fehleranfällig ist, wenn Vorwissen darüber, wo sich Personen in den Punktwolken befinden, genutzt wird. Da dieses Wissen ein Resultat der Detektion von Personen in diesen Punktwolken ist, ist es bei der Variante 2 nicht verfügbar.

Die Variante 3 nutzt wie die Variante 1 ebenfalls das Verfahren um Personen in den LiDAR-Daten zu detektieren. Es verzichtet aber auf die Bildausschnitte und verarbeitet stattdessen die vollständigen Bilder für die Körperposenschätzung. In der untersuchten Konfiguration wird beim Zusammenführen der beiden Datenströme zudem darauf geachtet, dass eine Person in beiden Sensormodalitäten gefunden wird. Ist dies nicht der Fall, wird sie nicht in das Endergebnis übernommen. Die beiden Sensormodalitäten bestätigen sich also gegenseitig. Hierdurch kann eine sehr gute Genauigkeit bei der Detektion erzielt werden, auch wenn in den beiden Einzelverfahren Schwellwerte für die Detektion genutzt werden, die zu vielen falsch positiven Ergebnissen führen. Die erreichbare Sensitivität sinkt im Umkehrschluss aber. Diese Variante hat im Hinblick auf die Laufzeit auch die schlechtesten Ergebnisse erzielt, da hier neben der aufwändigen Detektion von Personen in den Punktwolken zusätzlich noch die vollständigen Bilder verarbeitet werden müssen. Grundsätzlich ist es jedoch möglich die Varianten 1 und 3 zu kombinieren, sodass Bildausschnitte genutzt werden und die Sensormodalitäten sich dennoch gegenseitig bestätigen.

Es muss auch berücksichtigt werden, dass die beiden Sensormodalitäten LiDAR und Kamera unterschiedliche prinzipielle Einschränkungen haben. LiDAR-Sensoren sind aktive Sensoren und nicht auf zusätzliche Lichtquellen angewiesen. Sie sind daher auch in der Lage Personen bei Dunkelheit zu erfassen. Kameras sind in solchen Situationen hingegen auf zusätzliche Lichtquellen angewiesen. Die übliche Beleuchtung von PKWs würde zwar auch bei Dunkelheit das Erfassen

von Personen vor dem Fahrzeug weiterhin problemlos erlauben, aber insbesondere seitlich vom Fahrzeug wären sie auf eine zusätzliche Straßenbeleuchtung angewiesen. Die Variante 1 hat hier bei Dunkelheit einen inhärenten Vorteil im Hinblick auf die Detektion von Personen.

Die Qualität der Ergebnisse der Posenschätzung wurde in dieser Arbeit aufgrund von fehlenden Daten der Grundwahrheit nicht quantitativ ausgewertet. Für das reine Posenschätzungsverfahren wurde dies aber bereits von Cao et al. [2019] untersucht. Im Hinblick auf das Überführen der Ergebnisse in den 3D-Raum mit den vorgestellten Verfahren hat sich ein gemischtes Bild ergeben. Ein prinzipielles Problem ist, dass die resultierenden 3D-Koordinaten immer auf der durch die Punktwolke beschriebenen Oberfläche einer Person liegen. Dies entspricht nicht unbedingt der tatsächlichen Position des jeweiligen Körperteils. Dies ist insbesondere dann ein Problem, wenn eine Person von der Seite durch den LiDAR-Sensor erfasst wird. Eine weitere Schwierigkeit ergibt sich aus der Zeitsynchronisation zwischen den Sensormodalitäten. Diese kann bei LiDAR-Sensoren, die die Szene nach und nach abtasten, nie komplett korrekt sein. Die resultierenden Zeitdifferenzen führen zu einer Abweichung der Position eines Körperteils in den Bildern und den Punktwolken. Eine Lösung ist es diese Differenzen zu minimieren, was erreicht werden kann, wenn die Aufnahmerate bei einer oder beiden Sensormodalitäten erhöht wird. Die in den Experimenten für die Erfassung der Bilddaten verwendeten Kameras wurden beispielsweise nur mit 10 Aufnahmen pro Sekunde betrieben, um die bei der durchgeführten längeren Messfahrt anfallende Datenmenge zu beschränken. Sie wären allerdings zu höheren Aufnahmeraten in der Lage. Ideal wäre es jedoch, wenn beide Sensormodalitäten die Umgebung komplett synchron aufnehmen würden. Dazu wäre jedoch eine andere Art von LiDAR-Sensor notwendig, z.B. sog. Flash-LiDAR-Sensoren.

9.4 Gesamtsystem

Das in der Einleitung vorgeschlagene System soll Informationen über Fußgänger im urbanen Verkehrsraum sammeln. Dabei sollte genutzt werden, dass es sich bei künftigen Fahrzeugen in einem zunehmenden Maß um mobile Multisensorsysteme handelt. Deren unterschiedliche Sensormodalitäten haben individuelle Stärken und sollten daher so kombiniert werden, dass möglichst von allen dieser Stärken profitiert wird. Diese Arbeit beschränkt sich auf die Untersuchung der Nutzung von LiDAR-Sensoren und RGB-Kameras, um die Daten über die Fußgänger von einem mobilen Multisensorsystem aus zu erfassen. Dabei hat sich gezeigt, dass die Kombination aus dem vorgeschlagenen Verfahren zur Detektion von Personen in 3D-Punktwolken sowie dem zum Tracking dieser Personen im 3D-Raum dazu geeignet ist, diese in nahen und mittleren Entfernungen vom Sensor zu detektieren und zu verfolgen. Es ist außerdem möglich, sie mit einer, für den Zeitraum in dem sie im Erfassungsbereich des Sensorsystems verbleiben, konsistenten ID zu versehen und ihre Bewegungsgeschwindigkeit und Richtung zu bestimmen. Es lassen sich so also Bewegungsströme von Personen erfassen und z.B. in einer Datenbank für eine spätere weitergehende Auswertung speichern.

Die Erfassung von Detailinformationen über die detektierten Personen mithilfe von Kameras wurde anhand der Körperposenschätzung untersucht. Dabei hat sich gezeigt, dass das Vorgehen zwar grundsätzlich funktioniert und teilweise gute Ergebnisse liefert, es aber dennoch einige Schwächen gibt. Mit den verwendeten Kameras ist es auf den Nahbereich beschränkt und die für die Posenschlüsselpunkte ermittelten 3D-Koordinaten sind nicht immer vollständig korrekt. Für einige dieser Probleme gibt es Lösungsansätze, die künftig untersucht werden können. Es wurde auch gezeigt, dass die Kombination der beiden Sensormodalitäten dazu geeignet ist, falsch positive Detektionen zu reduzieren und so die Genauigkeit der Detektion von Personen zu verbessern. Dies kann in Szenarien genutzt werden, wo falsch positive Detektionen in jedem Fall vermieden werden sollen.

10 Zusammenfassung und Ausblick

In diesem Kapitel wird die vorliegende Arbeit zunächst zusammengefasst und die gestellten Forschungsfragen beantwortet anschließend werden mögliche weitere Arbeiten vorgeschlagen, welche die vorliegende Arbeit logisch fortsetzen.

10.1 Zusammenfassung und Beantwortung der Forschungsfragen

Forschungsfrage 1: Welche Detektionsleistung von Personen kann in 3D-Punktwolken erreicht werden und wie stark hängt diese von der Menge an verwendeten Trainingsdaten, der Punktdichte bzw. der Entfernung der Personen zum Sensor ab?

Für die Detektion von Personen in 3D-Punktwolken wurde ein Verfahren entwickelt und untersucht, welches ein künstliches neuronales Netz mit einem Abstimmverfahren kombiniert, um Objekte zu detektieren. Das neuronale Netz verarbeitet die Punktwolken dabei in Form von lokalen Punktnachbarschaften. Diese werden durch einen Ursprungspunkt definiert und umfassen alle Punkte in einem gewissen Radius um diesen Punkt. Sie sind dadurch wesentlich kleiner als vollständige Punktwolken oder große Punktwolkensegmente. Sie erfordern daher auch ein im Vergleich zu anderen Ansätzen weniger komplexes neuronales Netz, welches mit weniger Aufwand trainiert werden kann. Das neuronale Netz entscheidet, ob die Nachbarschaft zu einem Objekt von Interesse gehört und um welche Art von Objekt es sich dabei handelt. Bei Objekten von Interesse, versucht es außerdem die Position des Objektmittelpunktes zu schätzen. Diese Informationen werden dann genutzt, um den Stimmraum des Abstimmverfahrens zu füllen, welches anschließend in diesem Raum nach Stimmschwerpunkten sucht, um zu entscheiden, wo sich Objekte von Interesse in den verarbeiteten Punktwolken befinden.

Mit diesem Detektionsverfahren kann über alle drei untersuchten Datensätze und unter Berücksichtigung von nur gut sichtbaren Personen in allen Entfernungsbereichen eine Sensitivität von 0,74 erreicht werden, wenn mindestens eine Genauigkeit von 0,8 angestrebt wird. Wird stattdessen eine Genauigkeit von mindestens 0,9 angestrebt, liegt die erreichbare Sensitivität bei 0,69. Diese Ergebnisse werden mit der Standardvariante der beiden untersuchten Varianten des neuronalen Netzes erreicht.

Wenn anstatt aller 1300 zum Training zur Verfügung stehenden Punktwolken nur 325 zufällig ausgewählte für das Training verwendet werden, verringert sich die Gesamtleistung des Detektionsverfahrens bei der Standardvariante des neuronalen Netzes nur um ca. 4 %. Bei nur 100 zufällig ausgewählten Punktwolken verringert sie sich bei dieser Variante des neuronalen Netzes jedoch um ca. 18 %. Bei so wenig Trainingsdaten hat sich die vereinfachte Variante des neuronalen Netzes, mit weniger zu trainierenden Parametern als geeigneter herausgestellt. Hier werden bei einem Training mit nur 100 anstatt 1300 Punktwolken nur ca. 12 % schlechtere Ergebnisse erzielt. Auch ist diese Variante des neuronalen Netzes bei so wenig Trainingsdaten insgesamt leistungsfähiger als die Standardvariante. Wichtig bei diesen Betrachtungen ist jedoch die Auswahl der Trainingsdaten. Diese sollten möglichst unterschiedlich zueinander sein. Wenn anstatt einer zufälligen Auswahl ei-

ne fortlaufende Sequenz an Punktwolken für das Training verwendet wird, kommt es zu signifikant schlechteren Trainingsergebnissen.

Im Hinblick auf die Punktdichte bzw. die Entfernung zum Sensor, aus der sich diese bei den verwendeten LiDAR-Sensoren ableitet, wurde festgestellt, dass bis zu 10 m sehr gute und bis 30 m zufriedenstellende Ergebnisse erzielt werden können. In dieser Entfernung liegen bei den verwendeten LiDAR-Sensoren zwei Punkte in der Punktwolke vertikal ca. 0,21 m und horizontal ca. 0,09 m auseinander. In größeren Entfernungen nimmt die Leistung des Verfahrens dann schnell ab. In zukünftigen Untersuchungen wäre es wünschenswert Sensoren zu verwenden, die horizontal und vertikal dieselbe Winkelauflösung haben, um festzustellen, ob bei einer homogeneren Verteilung der 3D-Punkte auch bei einer insgesamt geringeren Punktdichte bessere Ergebnisse erzielt werden können.

Forschungsfrage 2: Welche Trackinggenauigkeit kann erzielt werden und wie verändert sich die Detektionsleistung des Gesamtsystems durch das Tracking?

In dieser Arbeit wurde ein Verfahren zum Tracking der detektierten Objekte im 3D-Raum untersucht. Es verwendet einen Kalman-Filter mit einem *Constant Velocity* Bewegungsmodell. Für die Assoziation zwischen bekannten getrackten und im aktuellen Zeitschritt detektierten Objekten, wird die *global nearest neighbor* Methode verwendet. Die Entscheidung einen Track zu entfernen wenn das dazugehörige Objekt nicht mehr länger detektiert wird, erfolgt auf Basis der Unsicherheit, die der genutzte Kalman-Filter bei der Positionsschätzung hat. Diese wird aus der Kovarianzmatrix des vom Filter modellierten Objektzustands abgelesen und der Track wird entfernt, wenn sie einen Schwellwert überschreitet.

In Hinblick auf die Genauigkeit des Trackings (*MOTA*) kann, wenn gut sichtbare, teilweise verdeckte und vollständig verdeckte Personen berücksichtigt werden ein Wert von ca. 0,408 erreicht werden. Wenn stattdessen nur gut sichtbare Personen berücksichtigt werden, liegt die Trackgenauigkeit hingegen bei 0,624. Das Tracking ist außerdem in der Lage, wenn auch verdeckte Personen berücksichtigt werden, die Detektionsleistung des Gesamtsystems bestehend aus Detektion und Tracking um ca. 18 % gegenüber dem zu verbessern, was das Detektionsverfahren in solchen Situationen alleine erreichen kann.

Forschungsfrage 3: Welche Detektionsleistung erzielen unterschiedliche Varianten zur multimodalen Körperposenschätzung und Detektion von Personen und wie unterscheiden sich diese Varianten in Bezug auf die Verarbeitungszeit?

Der dritte Untersuchungsgegenstand dieser Arbeit lag in der kombinierten Auswertung von 3D-Punktwolken und RGB-Bildern, um neben der reinen Detektion von Personen auch weitere Informationen über diese zu gewinnen. Dies wurde am Beispiel einer Körperposenschätzung in Bildern untersucht, durch die die Körperhaltung von Personen bestimmt werden kann. Die Ergebnisse dieser Körperposenschätzung werden unter Nutzung der 3D-Punktwolken in den 3D-Raum übertragen. Es wurden drei unterschiedliche Varianten einer Methode für eine solche gemeinsame Nutzung der beiden Sensormodalitäten untersucht.

In der ersten Variante steuert die Detektion von Personen in den Punktwolken die anschließende Körperposenschätzung in den Bildern. Es werden gezielt nur die Teile der Bilder verarbeitet, über die bekannt ist, dass sie Personen enthalten. Die zweite Variante verzichtet komplett auf die Detektion von Personen in den Punktwolken und nutzt für diese Detektion stattdessen ebenfalls das Verfahren zur Körperposenschätzung in Bildern. Die Punktwolken werden dann anschließend nur genutzt, um 3D-Koordinaten für die Ergebnisse zu bestimmen. Die dritte Variante ähnelt der

ersten, verzichtet jedoch auf das Generieren von Bildausschnitten und verarbeitet stattdessen, wie die zweite Variante die vollständigen Bilder.

Bei der Betrachtung der Detektionsleistung hat sich herausgestellt, dass die dritte Variante, in der sich die Auswertungen der Punktwolken und der Bilddaten gegenseitig bestätigen, eine nahezu perfekte Genauigkeit erzielt gleichzeitig wird hier aber auch die geringste Sensitivität erreicht. Dies zeigt, dass es bei den beiden Sensormodalitäten selten an derselben Stelle zu einem falschen Ergebnis kommt.

Im Hinblick auf die Verarbeitungszeit hat sich gezeigt, dass die Variante 2, in der hauptsächlich die Bilddaten verarbeitet werden und nur eine einfache Verarbeitung der Punktwolken stattfindet, am schnellsten ist. Variante 1 mit dem vereinfachten neuronalen Netz ist jedoch nur geringfügig langsamer. Hier erfolgt zwar sowohl eine umfangreiche Verarbeitung der Punktwolken als auch der Bilder, bei den Bildern ist diese aber auf einzelne Bereiche beschränkt. Die Variante 3 hingegen ist am langsamsten, da beide Modalitäten umfangreich verarbeitet werden und es keine Einschränkung auf einzelne Bildbereiche gibt.

10.2 Weitere Arbeiten

Die aus dieser Arbeit gewonnen Erkenntnisse lassen Erweiterungs- und Verbesserungsmöglichkeiten der vorgestellten Methoden erkennen, sowie Ansätze zur weiteren experimentellen Untersuchung von deren Leistungsfähigkeit. Diese werden in diesem Abschnitt erläutert.

Andere Objektklassen

Das vorgestellte Verfahren zur Detektion von Objekten in Punktwolken wurde im Rahmen dieser Arbeit ausschließlich für die Detektion von Personen verwendet. Es ist jedoch nicht darauf beschränkt Personen zu detektieren, sondern so ausgelegt, dass es auch für andere Objektklassen verwendet werden kann. Die dahingehenden Fähigkeiten des Verfahrens und mögliche Schwierigkeiten dabei können in künftigen Arbeiten untersucht werden.

DATMO für Objekte in großer Entfernung

Ein begrenzender Faktor beim Detektieren von Objekten in 3D-Punktwolken ist die Punktdichte. Diese nimmt bei üblichen LiDAR-Sensoren in größerer Entfernung zum Sensor immer weiter ab. Es gibt einen Entfernungsbereich, in dem Objekte zwar grundsätzlich noch vom Sensor erfasst werden, aber nur noch so wenige Punkte in der resultierenden Punktwolke zu diesen Objekten gehören, dass ihr Erscheinungsbild nicht mehr ausreichend gut repräsentiert ist, um sie anhand von diesem zu detektieren. In solchen Bereichen kann ein sog. DATMO (*Detection and Tracking of Moving Objects*) Verfahren verwendet werden, um zumindest sich bewegende Objekte anhand ihrer Bewegung zu detektieren. Solche Verfahren haben aber wiederum andere Nachteile und können z.B. nur schwer zwischen verschiedenen Klassen von Objekten unterscheiden und keine stehenden Objekte erkennen. Es wäre daher wünschenswert, das in dieser Arbeit verwendete Verfahren zur Detektion von Objekten anhand ihres Erscheinungsbilds um ein DATMO-Verfahren zu ergänzen, was dann in größeren Entfernungen verwendet wird, in denen das vorgestellte Detektionsverfahren nicht mehr funktioniert.

Posenschätzung in LiDAR-Daten

Für die Körperposenschätzung wird in dieser Arbeit ein Verfahren verwendet, welches Bilder auswertet. Obwohl dieses bei ausreichend gut von den Kameras erfassten Personen gute Ergebnisse liefert, ist das Übertragen von dessen Ergebnissen in den 3D-Raum nicht ohne Schwierigkeiten.

Eine Posenschätzung in den Punktwolken wäre aufgrund von deren geringer Datendichte zwar ebenfalls schwierig, würde diese Schwierigkeiten aber umgehen und kann einen interessanten Gegenstand für weitere Untersuchungen darstellen. Das in dieser Arbeit verwendete Verfahren zur Detektion von Personen in 3D-Punktwolken könnte z.B. durch das zusätzliche Generieren von Stimmen im Bezug auf die Körperpose ggf. um eine solche Posenschätzung ergänzt werden. Eine Schwierigkeit hierbei wäre aber das Bereitstellen geeigneter Trainingsdaten.

Blickrichtung

Das Bestimmen der Blickrichtung einer Person kann interessante Hinweise auf deren Intentionen liefern. Ein Bildauswertungsverfahren um diese zu ermitteln, dass dann ähnlich wie das Verfahren zur Körperposenschätzung in die Gesamtmethode integriert wird, wäre eine wertvolle Erweiterung.

Gerichtete Sensorik

Ein Problem bei den in dieser Arbeit untersuchten Komponenten zur Bildauswertung ist, dass die dafür verwendeten Aufnahmen der Rundumkameras des Experimentalsystems Personen in größerer Entfernung nur sehr klein darstellen. Dies liegt u.a. daran, dass sie zum Erreichen der Rundumabdeckung weitwinklige Objektive verwenden. Es wäre denkbar stattdessen schwenkbare Kameras zu verwenden, die gezielt auf in den Punktwolken detektierte Personen ausgerichtet werden. Diese könnten dann besser aufgelöste Bilder dieser Personen erzeugen, was zu besseren Ergebnissen bei der Körperposenschätzung in größerer Entfernung führen würde.

Literaturverzeichnis

Asvadi A, Garrote L, Premebida C, Peixoto P, Nunes UJ (2017) DepthCN: Vehicle detection using 3D-LIDAR and ConvNet. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC): 1–6.

Asvadi A, Peixoto P, Nunes U (2015) Detection and tracking of moving objects using 2.5D motion grids. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems: 788–793.

Azim A, Aycard O (2012) Detection, classification and tracking of moving objects in a 3D environment. In: 2012 IEEE Intelligent Vehicles Symposium: 802–807.

Baig Q, Perrollaz M, Laugier C (2014) A robust motion detection technique for dynamic environment monitoring: A framework for grid-based monitoring of the dynamic environment. IEEE Robotics Automation Magazine, 21 (1): 40–48.

Bar-Shalom Y, Daum F, Huang J (2009) The probabilistic data association filter. IEEE Control Systems Magazine, 29 (6): 82–100.

Behley J, Steinhage V, Cremers AB (2013) Laser-based segment classification using a mixture of Bag-of-Words. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems: 4195–4200.

Benedek C (2014) 3D people surveillance on range data sequences of a rotating LiDAR. Pattern Recognition Letters, 50: 149–158. Depth Image Analysis.

Bernardin K, Stiefelhagen R (2008) Evaluating multiple object tracking performance: The CLEAR MOT metrics. EURASIP Journal on Image and Video Processing, 2008 (1): 246309.

Borgmann B, Hebel M, Arens M, Stilla U (2017a) Detection of persons in MLS point clouds using implicit shape models. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W7: 203–210.

Borgmann B, Hebel M, Arens M, Stilla U (2017b) Konzept zur Gefährdungserkennung im städtischen Verkehrsraum durch Personendetektion in MLS-Punktwolken. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., 26: 262–275.

Borgmann B, Hebel M, Arens M, Stilla U (2018a) Fußgängerbezogene Informationsgewinnung zur Situationsanalyse mit einem mobilen Multisensorsystem. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., 27: 363–375.

Borgmann B, Hebel M, Arens M, Stilla U (2018b) Usage of multiple LiDAR sensors on a mobile system for the detection of persons with implicit shape models. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2: 125–131.

Borgmann B, Hebel M, Arens M, Stilla U (2019) Using neural networks to detect objects in MLS point clouds based on local point neighborhoods. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W7: 17–24.

Borgmann B, Hebel M, Arens M, Stilla U (2020) Pedestrian detection and tracking in sparse MLS point clouds using a neural network and voting-based approach. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, V-2-2020: 187–194.

Borgmann B, Hebel M, Arens M, Stilla U (2021a) Information acquisition on pedestrian movements in urban traffic with a mobile multi-sensor system. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2021: 131–138.

Borgmann B, Schatz V, Hammer M, Hebel M, Arens M, Stilla U (2021b) MODISSA: A multipurpose platform for the prototypical realization of vehicle-related applications using optical sensors. Applied Optics, 60 (22): F50–F65.

Borgmann B, Schatz V, Kieritz H, Scherer-Klöckling C, Hebel M, Arens M (2018c) Data processing and recording using a versatile multi-sensor vehicle. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1: 21–28.

Cao Z, Hidalgo Martinez G, Simon T, Wei S, Sheikh YA (2019) OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Cao Z, Simon T, Wei SE, Sheikh Y (2017) Realtime multi-person 2D pose estimation using part affinity fields. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 1302–1310.

Dantone M, Gall J, Leistner C, Van Gool L (2013) Human pose estimation using body parts dependent joint regressors. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 3041–3048.

Dewan A, Caselitz T, Tipaldi GD, Burgard W (2016) Motion-based detection and tracking in 3D LiDAR scans. In: 2016 IEEE International Conference on Robotics and Automation (ICRA): 4508–4513.

Diehm AL, Gehrung J, Hebel M, Arens M (2020) Extrinsic self-calibration of an operational mobile LiDAR system. In: Proceedings of the conference "Laser Radar Technology and Applications XXV", 11410: 46–61.

Du Y, ShangGuan W, Chai L (2018) Particle filter based object tracking of 3D sparse point clouds for autopilot. In: 2018 Chinese Automation Congress (CAC): 1102–1107.

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining: 226–231.

Fang H, Xie S, Tai Y, Lu C (2017) RMPE: Regional multi-person pose estimation. In: 2017 IEEE International Conference on Computer Vision (ICCV): 2353–2362.

Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24 (6): 381–395.

Frossard D, Urtasun R (2018) End-to-end learning of multi-sensor 3D tracking by detection. In: 2018 IEEE International Conference on Robotics and Automation (ICRA): 635–642.

Gao H, Cheng B, Wang J, Li K, Zhao J, Li D (2018) Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. IEEE Transactions on Industrial Informatics, 14 (9): 4224–4231.

Garcia-Garcia A, Gomez-Donoso F, Garcia-Rodriguez J, Orts-Escolano S, Cazorla M, Azorin-Lopez J (2016) PointNet: A 3D convolutional neural network for real-time object class recognition. In: 2016 International Joint Conference on Neural Networks (IJCNN): 1578–1584.

He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV): 2980–2988.

Johnson AE, Hebert M (1999) Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21 (5): 433–449.

Kalman RE (1960) A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82 (1): 35–45.

Ke L, Chang MC, Qi H, Lyu S (2018) Multi-scale structure-aware network for human pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV): 713–728.

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: Bengio Y, LeCun Y (eds) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings

Knopp J, Prasad M, Gool LV (2011) Scene cut: Class-specific object detection and segmentation in 3D scenes. In: 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission: 180–187.

Konstantinova P, Udvarev A, Semerdjiev T (2003) A study of a target tracking algorithm using global nearest neighbor approach. In: Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'03): 290–295.

Kuhn HW (1955) The hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2 (1-2): 83–97.

Li X, Chen S, Hu X, Yang J (2019) Understanding the disharmony between dropout and batch normalization by variance shift. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR): 2677–2685.

Liu Y, Fan B, Meng G, Lu J, Xiang S, Pan C (2019) DensePoint: Learning densely contextual representation for efficient point cloud processing. In: IEEE International Conference on Computer Vision (ICCV): 5239–5248.

Manghat SK, El-Sharkawy M (2020) A multi sensor real-time tracking with LiDAR and camera. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC): 0668–0672.

Maturana D, Scherer S (2015) VoxNet: A 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 922–928.

Moosmann F, Stiller C (2013) Joint self-localization and tracking of generic objects in 3D range data. In: 2013 IEEE International Conference on Robotics and Automation (ICRA): 1146–1152.

Munkres J (1957) Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics, 5 (1): 32–38.

Navarro-Serment LE, Mertz C, Hebert M (2010) Pedestrian detection and tracking using three-dimensional LADAR data. The International Journal of Robotics Research, 29 (12): 1516–1528.

Pang Z, Li Z, Wang N (2021) Model-free vehicle tracking and state estimation in point cloud sequences. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 8075–8082.

Qi CR, Litany O, He K, Guibas LJ (2019) Deep hough voting for 3D object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV): 9276–9285.

Qi CR, Su H, Mo K, Guibas LJ (2017a) PointNet: Deep learning on point sets for 3D classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 77–85.

Qi CR, Yi L, Su H, Guibas LJ (2017b) PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in Neural Information Processing Systems 30 (pp. 5099–5108). Curran Associates, Inc.

Sato S, Hashimoto M, Takita M, Takagi K, Ogawa T (2010) Multilayer LiDAR-based pedestrian tracking in urban environments. In: 2010 IEEE Intelligent Vehicles Symposium: 849–854.

Schatz V (2017) Measurement of the timing behaviour of off-the-shelf cameras. Measurement Science and Technology, 28 (4): 045905.

Socher R, Huval B, Bath B, Manning CD, Ng AY (2012) Convolutional-recursive deep learning for 3D object classification. In: Advances in Neural Information Processing Systems: 656–664.

Statistisches Bundesamt (Destatis) (2021) Verunglückte: Deutschland, Jahre, Geschlecht, Altersgruppen, Art der Verkehrsbeteiligung, Ortslage, Schwere der Verletzung. [abgerufen am 17.08.2021] `https://www-genesis.destatis.de/genesis//online?operation=table&code=46241-0007&bypass=true&levelindex=0&levelid=1629195006955`.

Velizhev A, Shapovalov R, Schindler K (2012) Implicit shape models for object detection in 3D point clouds. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, I-3: 179–184.

Wan E, Van Der Merwe R (2000) The unscented kalman filter for nonlinear estimation. In: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373): 153–158.

Wang DZ, Posner I, Newman P (2015) Model-free detection and tracking of dynamic objects with 2D LiDAR. The International Journal of Robotics Research, 34 (7): 1039–1063.

Wang H, Wang B, Liu B, Meng X, Yang G (2017) Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. Robotics and Autonomous Systems, 88: 71–78.

Yang Y, Ramanan D (2013) Articulated human detection with flexible mixtures of parts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (12): 2878–2890.

Zhang J, Xiao W, Coifman B, Mills JP (2020) Vehicle tracking and speed estimation from roadside LiDAR. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13: 5597–5608.

Zhao J, Xu H, Liu H, Wu J, Zheng Y, Wu D (2019) Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors. Transportation Research Part C: Emerging Technologies, 100: 68–87.

Zhou Y, Tuzel O (2018) VoxelNet: End-to-End learning for point cloud based 3D object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition: 4490–4499.

Danksagung

Mit der Fertigstellung dieser Dissertation und dem Abschluss meiner Promotion geht für mich ein herausfordernder und spannender Lebensabschnitt zu Ende. Eine Phase meines Lebens, in der ich viele neue Erfahrungen gemacht und vieles gelernt habe. Wenn ich auf diese Zeit zurückblicke, kommen mir die Menschen in den Kopf, die mich bei meinem Vorhaben begleitet und unterstützt haben und denen ich hier danken möchte.

Zunächst möchte ich Prof. Dr.-Ing. Uwe Stilla danken, der meine Promotion an der Technischen Universität München betreut hat. Seine vielen Ratschläge im Hinblick auf das Thema meiner Promotion, wissenschaftliche Veröffentlichungen, dem Aufbau dieser Dissertation und zu verschiedenen fachlichen und inhaltlichen Aspekten waren sehr wertvoll für mich und haben mir in meiner Arbeit bedeutend geholfen. Ebenfalls möchte ich ihm dafür danken, mir zum richtigen Zeitpunkt mit motivierenden Worten ein wenig Antrieb gegeben zu haben, um diese Dissertation fertigzustellen. Danken möchte ich auch Prof. Dr.-Ing. Alexander Reiterer, der diese Dissertation neben Prof. Dr.-Ing. Uwe Stilla begutachtet hat und Prof. Dr.-Ing. Christoph Holst, der den Vorsitz der Prüfungskommission übernommen hat.

Ich möchte auch den vielen anderen Personen an der Technischen Universität München danken, die ich im Verlauf meiner Promotion kennenlernen durfte und mit denen ich mich austauschen konnte. Viele von ihnen befanden sich ihrerseits in einem Promotionsvorhaben oder hatten ein solches bereits abgeschlossen. Namentlich sei hier stellvertretend für alle Dr.-Ing. Ludwig Högner erwähnt, der immer Zeit für uns externe Doktoranden hatte und mit dem ich viele interessante fachliche, aber auch weniger fachliche Gespräche geführt habe.

Diese Dissertation ist im Rahmen meiner Tätigkeit am Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB entstanden. Von den vielen Kollegen, neben denen ich dort gearbeitet und geforscht habe und die mich auf verschiedene Art unterstützt haben, möchte ich hier nur einige nennen: Zunächst sind da mein Abteilungsleiter Dr. rer. nat. Michael Arens und mein Gruppenleiter sowie Mentor Dr. rer. nat. Marcus Hebel. Die beiden haben mir den nötigen Rahmen, die Freiheiten und den Antrieb gegeben, meine Promotion zu verfolgen und standen mir mit fachlichem Rat während des gesamten Promotionsprozesses zur Seite. Marcus Hebel war außerdem einer der Korrekturleser dieser Dissertation. Ebenfalls sollen hier meine Kollegen Dr. rer. nat. Marcus Hammer, Dr.-Ing. Joachim Gehrung, Axel Diehm, Dr. rer. nat. Volker Schatz, Ann-Kristin Grosselfinger und Dr.-Ing. Stefan Becker nicht unerwähnt bleiben, mit denen ich ein Büro bzw. viele gemeinsame Mittags- und Kaffeepausen geteilt habe und die mich so oft im lockeren Rahmen unterstützen konnten.

Aus meiner Familie möchte zunächst meinen Onkel Dirk Borgmann nennen, der ein weiterer Korrekturleser dieser Dissertation gewesen ist. Mein besonderer Dank gebührt außerdem meinen Eltern Claudia und Gerd Borgmann, die mich mein ganzes Leben in allem unterstützt haben und mich so überhaupt erst in die Lage brachten ein Promotionsvorhaben zu beginnen und nun erfolgreich abzuschließen.