Monokulare Visuelle Odometrie auf Multisensorplattformen für autonome Fahrzeuge

Zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

von der Fakultät für Maschinenbau
des Karlsruher Instituts für Technologie (KIT)
genehmigte

Dissertation

von

DIPL.-ING. JOHANNES GRÄTER

aus Schwäbisch Hall

Tag der mündlichen Prüfung:	28. Februar 2019
Hauptreferent:	Prof. Dr.-Ing. Christoph Stiller
Korreferent:	Prof. Dr.-Ing. Michael Heizmann

Danksagung

Diese Arbeit ist ein Produkt meiner Tätigkeit am Institut für Mess- und Regelungstechnik und wäre niemals möglich gewesen ohne die vielseitige Unterstützung meiner Kollegen, Familie und Freunde. Mein erster Dank gilt hierbei Herrn Prof. Dr.-Ing. Christoph Stiller, der die wohl besten Rahmenbedingungen einer Promotion geschaffen hat, die man sich wohl wünschen kann. Mein weiterer Dank gilt Herrn Prof. Dr.-Ing. Heizmann für die Übernahme des Korreferats und das damit verbundene Interesse an meiner Arbeit.

Ohne die hervorragende fachliche Betreuung von Dr. Martin Lauer wäre ich nicht zu diesem Punkt gekommen, vielen Dank für die tolle Projektleitung im Abalid-Projekt, die wöchentlichen Treffen und die vielen anregenden Diskussionen. An dieser Stelle möchte ich auch Henning Lategahn danken, dessen kurze aber sehr produktive Betreuung zu Anfang des Projekts Abalid mich meine Doktorandenzeit über inspiriert hat.

Unser Institut lebt von seinen Doktoranden. Ohne die vielen konstruktiven und unkonstruktiven Kaffeerunden, Sommerseminare, Social-Tuesdays, Konferenzbesuche und Treffen darüber hinaus hätte ich wohl schon früh das Handtuch geworfen. Es ist mir unmöglich, alle Kollegen und Kolleginnen aufzuzählen, denen ich dankbar bin, da jeder seinen Teil beitrug und ich von allen etwas gelernt habe. Darum beschränke ich mich darauf, meine Korrekturleser explizit zu nennen: Johannes Beck, Johannes Janosovits, Tilman Kühner, Jan-Hendrik Pauls, Fabian Poggenhans, Jannik Quehl, Marc Sons, Wei Tian und Sascha Wirges.

Auch dem Sekretariat, das stets ungeliebte Aufgaben von mir fern hielt, gilt mein Dank. Ohne die Unterstützung der Werkstätten wären meine Versuchsaufbauten wohl nie funktionstüchtig geworden.

Ganz besonderer Dank gilt meinem Bruder Sebastian Gräter und meinen Eltern Ulrike und Friedrich Gräter, die den Grundstein zu dem legten, was ich heute bin. Mein größter Dank gilt jedoch meiner Frau Andische. Ohne ihre Unterstützung, ihren Zuspruch, ihre Liebe und ihr kritisches Hinterfrage, welches mir half, Probleme objektiv zu betrachten, hätte ich diese Arbeit nicht anfertigen können.

Karlsruhe, im November 2018 *Johannes Gräter*

Abstract

Automated Vehicles need to establish a model of their surroundings in order to understand and interact with their environment. For that purpose, the motion of the vehicle has to be obtained in order to incorporate observations from different time instances into one single model. The estimation of this so called *odometry* is therefore the base of any autonomous platform.
Cameras are standard sensors for state of the art automated vehicles — and since their measurements resemble the human perception, they can be easily applied for advanced driver assistance systems. Moreover, their low weight, size and cost make them a very attractive sensor for odometry estimation. Therefore, this work aims at identifying and pushing the limits of state of the art monocular visual odometry in order to estimate the egomotion of the vehicle over long distances robustly and accurately.
The contributions presented are based on a well established methodology, which tracks points in a temporal image sequence and estimates the structure of the surroundings and the motion of the camera simultaneously. Due to the high density of information in an image, a huge amount of image points can be extracted. This results in high accuracy but also high computation time.
For online odometry estimation the computational cost has to be reduced. In this work a measurement selection strategy is proposed in order to choose appropriate measurements without lowering the accuracy of the estimation. For a fast and accurate solution of the optimization problem which lies beneath visual odometry, a novel prior estimation method is developed, which is not only applicable to monocular setups but also to multi-camera-platforms.
The main problem of monocular visual odometry is a consequence of the camera's measurement principle — by projecting the three dimensional world on a two dimensional imager, the depth-information is lost. While rotation and the direction of the translation can be obtained by a monocular system, the distance traveled, the so called *scale*, remains unobserved. Estimating the scale precisely and efficiently is therefore the focus of this work. For that task, two sources of information are utilized: First, the knowledge about the surrounding's geometry is used to determine the scale. Three different approaches are presented:

1. Estimation from two images, stabilized by vanishing points.

2. Estimation using the unscaled reconstruction of the scene.

3. Simultaneous estimation of the reconstruction of the scene, the motion and the scale.

The second source of scale information investigated is an additional Light-Detection-And-Ranging sensor (LIDAR).
For using the information from the LIDAR, the transformation between LIDAR and camera has to be known. To this end, a novel method is presented to estimate this transformation, which enables the usage of LIDARs with very low resolution.
The proposed methods are evaluated on the KITTI dataset [1] demonstrating their high accuracy. Additionally, the multi-camera variant of the prior estimation is assessed on the test vehicle of the Institute of Measurement and Control Systems (MRT) at KIT. Finally, to show the applicability of these algorithms on a real system, a driver assistance system that aims to avoid collisions between trucks and cyclists is designed, built and tested.

Kurzfassung

Automatisierte Fahrzeuge müssen ihre Umgebung präzise modellieren, um diese erfassen und mit ihr interagieren zu können. Essentiell hierfür ist die Schätzung der Fahrzeugbewegung mittels propriozeptiver Sensorik, um Messungen aus einzelnen Zeitpunkten im Umgebungsmodell zu vereinen. Eine Ausprägung davon ist die sogenannte *Odometrie*, welche somit einen Grundstein für das autonome Fahren darstellt.
Kameras werden vielfältig auf aktuellen automatisierten Fahrzeugen eingesetzt, da sie der menschlichen Wahrnehmung sehr ähneln und somit leicht für Fahrerassistenzsysteme eingesetzt werden können. Darüber hinaus sind sie leicht, klein und kostengünstig, was auch die Bewegungsschätzung mithilfe von Kameras sehr attraktiv macht.
Die Zielsetzung dieser Arbeit besteht darin, die Grenzen aktueller Odometrieschätzung mithilfe monokularer Kamerasysteme zu identifizieren und zu erweitern, sodass die Fahrzeug-Eigenbewegung über lange Strecken hinweg akkurat geschätzt werden kann.
Die in dieser Arbeit vorgestellten Weiterentwicklungen basieren auf einer etablierten Methode, in welcher Punkte in zeitlichen Bildsequenzen verfolgt werden und mit ihrer Hilfe die Umgebungsstruktur rekonstruiert sowie die Bewegung der Kamera gleichzeitig geschätzt werden. Aufgrund der hohen Dichte an Information im Bild können viele Punktmessungen extrahiert werden, wodurch die Genauigkeit dieser Bewegungsschätzung sehr hoch ausfällt. Um diese jedoch zur Laufzeit auf dem Fahrzeug durchzuführen, muss der dadurch entstehende Rechenaufwand gesenkt werden. Diese Arbeit widmet sich daher vertieft der Frage, wie Messungen ausgewählt werden können, um den Rechenaufwand zu verringern, ohne die Genauigkeit der Bewegungsschätzung zu reduzieren. Außerdem ist für die schnelle und akkurate Lösung des Optimierungsproblems, welches der Bewegungsschätzung zugrunde liegt, ein guter Startwert von großer Wichtigkeit. Hierfür wird eine Methode vorgestellt, welche diesen Startwert effizient und robust schätzt. Zudem wird die Methodik auf Multikamerasysteme erweitert und differenziert evaluiert.
Das zentrale Problem der kamerabasierten Bewegungsschätzung folgt aus deren Messprinzip: Durch die Projektion auf die Bildebene geht die Tiefeninformation der Umgebung verloren. Die Rotation kann zwar somit bestimmt werden, jedoch kann die Translation nur bis auf einen Skalierungsfaktor bestimmt werden — die sogenannte *Skale*. Ein besonderer Fokus

dieser Arbeit liegt daher auf der Frage, wie die Skale effizient und genau bestimmt werden kann. Hierzu wird unter anderem Wissen über die Geometrie der Umgebung genutzt. Insbesondere die Einbauhöhe der Kameras über Grund wird genutzt, um die Skale zu extrahieren. Um die Bodenoberfläche zu schätzen, werden drei Varianten untersucht:

- Fluchtpunktstabilisierte Schätzung aus zwei aufeinanderfolgenden Zeitpunkten.
- Schätzung mithilfe der Umgebungsrekonstruktion aus der unskalierten Bewegungsschätzung.
- Gleichzeitige Schätzung der Umgebungsrekonstruktion, der Bewegung und der Skale.

Zusätzlich wird in dieser Arbeit untersucht, wie die Skalenschätzung mithilfe eines Light-Detection-And-Ranging-Sensors (LIDAR) umgesetzt werden kann, welcher auf den meisten aktuellen autonomen Plattformen vorhanden ist.
Die Nutzung von LIDAR- und Kamera-Messungen setzt allerdings das Wissen über die Transformation zwischen LIDAR- und Kamerakoordinatensystem voraus. Darum wird zudem eine Methode vorgestellt, um diese Transformation zu erhalten. Diese zeichnet sich durch die Fähigkeit aus, auch LIDAR-Sensoren verwenden zu können, welche nur über eine sehr geringe Auflösung verfügen.
Die Algorithmen werden auf dem KITTI-Datensatz [1] evaluiert und verglichen. Eine sehr hohe Genauigkeit der Methoden konnte hierbei gezeigt werden. Des Weiteren wird die Multikamera-Variante der Startwertschätzung auf einem Versuchsträger des Instituts für Mess- und Regelungstechnik des KIT evaluiert. Um die Anwendbarkeit dieser Algorithmen in einem realen System zu demonstrieren, wird abschließend ein Advanced-Driver-Assistance-System vorgestellt, welches zur Kollisionsvermeidung zwischen Lastkraftwägen und Fahrradfahrern konzipiert, umgesetzt sowie in einer Probandenstudie getestet wurde.

Inhaltsverzeichnis

Notationen und Symbole . . . . . .

1 Einleitung **1**
1.1 Stand der Technik und Zielsetzung 2
1.2 Überblick 4

2 Übersicht **7**

3 Grundlagen **11**
3.1 Projektion auf die Bildebene 11
3.2 Epipolargeometrie 12
3.3 Methode der kleinsten Quadrate 15
3.3.1 Methode der kleinsten getrimmten Fehlerquadrate 16
3.3.2 M-Schätzer 17
3.4 RANdom SAmple Consensus (RANSAC) 18
3.5 Merkmalsextraktion im Bildraum 18

4 Frame-zu-Frame-Bewegungsschätzung **23**
4.1 Stand der Technik 24
4.2 Formulierung als nicht-lineares Optimierungsproblem . . 24
4.3 Nutzung von Bewegungsmodellen 26
4.4 Erweiterung auf Multikamerasysteme 29
4.5 Auswahl der Fehlermetrik durch simulierte Eingangsdaten 30

5 Zeitliche Inferenz **35**
5.1 Bewegungsschätzungsproblem als Graph 35
5.2 Erweiterung des Graphen auf mehrere Zeitpunkte 37
5.3 Keyframe-Auswahl 39
5.3.1 Nomenklatur 39
5.3.2 Fensterlänge des Optimierungsproblems 41
5.4 Landmarkenauswahl 41

5.4.1 Landmarkenkategorisierung 42
5.4.2 Auswahl durch semantische Informationen . . . 44
5.5 Robustifizierung und Problemformulierung 44

6 Skalenschätzung 47
6.1 Skalenschätzung anhand geometrischer Informationen . 47
6.1.1 Bodenebenenschätzung für Frame-zu-Frame Visuelle Odometrie mit Fluchtpunkten 48
6.1.1.1 Orientierungsschätzung der Bodenebene 51
6.1.1.2 Verfeinerung der Ebenenschätzung . . 52
6.1.2 Bodenebenenschätzung für Visuelle Odometrie mit zeitlicher Inferenz 53
6.1.2.1 A-Posteriori-Ebenenschätzung 54
6.1.2.2 Integration lokaler Bodenebenen in das Optimierungsproblem 54
6.2 Skalenschätzung mit Hilfe eines LIDAR 56
6.2.1 Vorgehen 57
6.2.2 Auswahl benachbarter Messungen 59
6.2.3 Segmentierung des Vordergrunds 59
6.2.4 Lokale Ebenenschätzung 60
6.2.5 Spezialfall: Punkte auf der Bodenebene 60
6.2.6 Integration der Tiefenmessungen in den Graphen 61
6.3 Skalenschätzung aus Bewegungsrelation der Kamera zum Bewegungsmodell 64

7 Extrinsische Kalibrierung von LIDAR und Kamera . . . 69
7.1 Stand der Technik 69
7.2 Methode 71
7.2.1 Problemformulierung 71
7.2.2 Unterstützung verschiedener Kameramodelle . . 72
7.2.3 Ausreißerbehandlung und Merkmalsassoziation . 75
7.2.4 Merkmalsextraktion 75
7.3 Ergebnisse 76
7.4 Evaluation 79

8 Evaluation 83
8.1 Frame-zu-Frame-Schätzung 84
8.1.1 Evaluation auf dem KITTI-Datensatz 85
8.1.2 Ergebnisse der Skalenschätzung mit Fluchtpunkten 88
8.2 Methoden mit zeitlicher Inferenz 92

8.2.1 Monokulare Skalenschätzung durch A-Posteriori-Bodenebenenschätzung 92
8.2.2 Monokulare Skalenschätzung durch integrierte Bodenebenenschätzung 93
8.2.3 Skalenschätzung durch LIDAR-Information . . . 96
8.2.4 Kombinierte Schätzung der Skale durch LIDAR und Bodenebenenannahme 100
8.3 Rechenzeiten 100
8.4 Fazit 104

9 Anwendungsfall ABALID 109
9.1 Einführung Projekt ABALID 109
9.2 Systemstruktur und Modulübersicht 110
9.3 Ergebnisse aus der Probandenstudie 112

10 Fazit 117
A.1 Parameter für LIMO 121
A.2 Zusätzliche Schaubilder für die Evaluation 122
A.3 Nutzung semantischer Information 122
A.4 Fahrradfahrer-Bewegungsschätzung 125

Notationen und Symbole

Akronym

$(\cdot)_i$	i-ter Eintrag eines Vektors
$\times$	Vektorprodukt
$\hat{(\cdot)}$	Schätzungen

2D/3D	**2**-**d**imensional/**3**-**d**imensional
LIDAR	**LI**ght **D**etection **A**nd **R**anging
BRIEF	**B**inary **R**obust **I**ndependent **E**lementary **F**eatures
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
IMU	**I**nertial **M**easurement **U**nit
GPS	**G**lobal **P**ositioning **S**ystem
ORB	**O**riented FAST and **r**otated **B**RIEF
RANSAC	**Ra**ndom **Sa**mpling **C**onsensus
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalization **a**nd **M**apping
SURF	**S**peeded **u**p **R**obust **F**eatures
VO	**V**isuelle **O**dometrie
MOMO	**MO**nocular **MO**tion Estimation on Manifolds
LIMO	**LI**DAR **MO**nocular Visual Odometry
LIVIODO	**LI**DAR **VI**sual **ODO**metry

Allgemeine Notationen

Skalare	Kursive (griechische) Minuskeln	a, b, c, σ, λ
Vektoren	Fette (griechische) Minuskeln	$\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\boldsymbol{\sigma}$, $\boldsymbol{\lambda}$
Matrizen	Fette Majuskeln	$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$
Mengen	Kaligrafische Majuskeln	$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$

Kapitel 1

Einleitung

Allgegenwärtige Mobilität hat die Welt verändert. War vor einem Jahrhundert das Pferd noch das Standardfortbewegungsmittel, sind heute Automobil, Flugzeug und Bahn unsere täglichen Begleiter. 625,2 Milliarden Kilometer legten in Deutschland zugelassene PKW allein 2016 zurück, rund 14000 km pro Fahrzeug [2]. Die Möglichkeit, in kurzer Zeit große Distanzen zurückzulegen, war der Grundstein für unsere heutige kulturell wie wirtschaftlich stark vernetzte Welt.
Zeitgenössische Fortbewegungsmittel sind dabei fast ausschließlich vom Menschen gesteuert. Die Hauptursache dafür ist leicht ersichtlich: Bewegung mit hoher Geschwindigkeit birgt zwangsläufig hohes Gefahrenpotenzial. Allein 2016 gab es 3206 Getötete sowie 396666 Verletzte auf deutschen Straßen [3]. Insbesondere das Verständnis der Umgebung und die angemessene Reaktion auf diese sind kognitiv äußerst herausfordernde Aufgaben, denen Maschinen heute noch nicht vollständig gewachsen sind. Zudem sind die Anforderungen an automatisierte Fortbewegungsmittel sehr hoch und die Menschen hinter dem Steuer daher nicht so einfach zu ersetzen. Aktuelle Entwicklungen bringen das autonome Kraftfahrzeug jedoch in greifbare Nähe und die stets weiter steigende Nachfrage an Fortbewegungsdienstleistungen macht ein solches System wirtschaftlich höchst attraktiv. So ist es nicht verwunderlich, dass große Firmen wie Google, Intel, Uber sowie alle namhaften Automobilhersteller an solchen Systemen arbeiten, um den nächsten Technologiesprung der Mobilität zu ermöglichen.
Anders als der Mensch sind aktuelle technische Systeme noch nicht in der Lage, Verkehrssituationen allumfassend zu erfassen und zu verstehen. Da-

her ist die Informationsintegration der Fahrzeugumgebung über größere Zeiträume hinweg notwendig. Um Informationen aus mehreren Zeitpunkten zu einem Modell zu vereinen, muss die Eigenbewegung des Fahrzeugs bekannt sein. Daher ist hochgenaue Trajektorienschätzung ein grundlegender Bestandteil aktueller Advanced Driver Assistance Systems (ADAS) und autonomer Fahrzeuge. Als Sensor für diese Anwendung bieten sich Kameras an, da sie günstig und wartungsarm sind und ihre Messungen sehr hohen Detailreichtum aufweisen. Mit ihnen ist es möglich, die Umgebung über lange Zeiträume zu verfolgen und somit hohe Trajektoriengenauigkeit zu erreichen. **Das Ziel dieser Arbeit ist die Umsetzung und Evaluation eines solchen Systems zur lokalen Trajektorienschätzung.** Hierbei liegt der Fokus auf dem wohl entscheidenden Nachteil von Kameras: Bei der Bildaufnahme wird die dreidimensionale Umgebung auf einen zweidimensionalen Bildsensor projiziert, wodurch die Tiefe der Umgebung verloren geht. Daher liegt hier besonderes Augenmerk auf der genauen Schätzung der durch die Kamera verlorengegangenen, Tiefeninformation, die sogenannte Skale.
Autonome Fahrzeuge haben hohe Anforderungen an Robustheit und Selbstabsicherung. Dies wird unter anderem mithilfe einer Vielzahl von Sensoren umgesetzt. Einerseits kann beispielsweise durch die Nutzung mehrerer Kameras der Messbereich erweitert werden, andererseits können komplementäre Sensoraufbauten Schwachstellen kompensieren, wie zum Beispiel LIDAR und Kamera. Hierbei wird insbesondere auf das Potential von Multikamerasystemen für die Rotationsschätzung sowie die Nutzung von LIDAR zur Skalengewinnung näher eingegangen.

1.1. Stand der Technik und Zielsetzung

Der wohl populärste Ansatz, um die Bewegung von Kameras zu schätzen, stammt aus der Photogrammetrie und wird als Bündelausgleich bezeichnet. Jeder Punkt in der dreidimensionalen Umgebung sendet Lichtstrahlen aus, welche in der Kamera beobachtet werden. Da sowohl die Position und die Orientierung der Kamera (die sogenannte Kamera-Pose) als auch die Position des beobachteten Punktes unbekannt sind, werden diese Strahlen so gebündelt, dass sie sich möglichst gut an einem Ort (der sogenannten Landmarke) schneiden. Hierbei sind also alle Posen und alle Landmarken Variablen desselben Optimierungsproblems. Theoretisch liefert diese Methode hervorragende Ergebnisse: Bei gegebenem normalverteiltem, mittelwertfreiem Bildrauschen ist der Bündelausgleich der Maximum-Likelihood-Schätzer für Posen und Landmarken [4]. In der Praxis steht diese Methodik jedoch vor wesentlichen Problemen:

1. Durch die Struktur des Problems verfängt sich der Optimierer schnell in lokalen Minima.

2. Durch Ausreißer wird das Ergebnis stark verfälscht.

3. Durch Tausende von Freiheitsgraden wird das Problem schnell sehr rechenaufwändig.

Diese Arbeit geht diese Probleme mit dem Gesamtziel an, ein **genaues und zugleich recheneffizientes Verfahren zur Eigenbewegungsschätzung aus monokularen Videosequenzen für autonome Fahrzeuge** zu entwickeln.

Für die Initialisierung des Bündelausgleichs ist ein präziser Startwert für die Optimierung nötig. Insbesondere der Rotationsanteil des Startwertes ist hier von besonderer Wichtigkeit, wie von Carlone et al. [5] erläutert wird. Ist der Startwert weit von der Lösung entfernt, konvergiert das Problem langsam und das Ergebnis ist ungenau. Wünschenswert ist hierfür, dass eine andere Fehlermetrik als für den Bündelausgleich verwendet wird. Hierfür bieten sich Fehlerfunktionen an, welche auf der Epipolargeometrie beruhen [6]. Für monokulare Systeme ist die Schätzung der Kamerabewegung aus der Epipolargeometrie eine Standardmethode. In realen Szenarien wie dem Straßenverkehr sind monokulare Systeme jedoch fehleranfällig — andere Fahrzeuge und Schmutz können die Kamera verdecken oder Sonne blendet den Bildsensor. Darum lautet das erste Ziel dieser Arbeit: **Rotationsschätzung des Fahrzeugs mit mehreren Kameras unterschiedlicher Perspektiven, in aufeinanderfolgenden Zeitschritten mithilfe von Epipolargeometrie.** Hierzu wird die Rotationsschätzung aus der Epipolargeometrie auf mehrere Kameras erweitert und ein Bewegungsmodell hinzugefügt. Simulativ werden verschiedene auf der Epipolargeometrie basierende Fehlermetriken ausgewertet und die Methodik wird anhand eines öffentlichen Datensatzes und auf Daten eines Testfahrzeugs evaluiert.

Über die Startwertschätzung hinaus ist das wohl dominanteste Thema der bündelausgleich-basierten Bewegungsschätzung die Auswahl der Messungen, welche dem Optimierungsproblem hinzugefügt werden (siehe hierzu zum Beispiel [7], [8], [9], [10]). Hierbei sollen Ausreißer erkannt und die Anzahl der Messungen auf ein Minimum reduziert werden, um das Optimierungsproblem beherrschen zu können. Darum ist das zweite Ziel dieser Arbeit, **Heuristiken zu erarbeiten, um Landmarken und Posen für das Bündelausgleichsproblem auszuwählen und Ausreißer zu reduzieren**. Insbesondere die Vorteile der *Methode der kleinsten getrimmten Fehler-*

quadrate sollen hierbei hervorgehoben werden.

Das letzte und dominante Ziel dieser Arbeit ist die **Schätzung der Skale**. Ausgehend von einem ersten untersuchten Ansatz mit einem monokularen Kamerasystem soll die Skale über geometrisches Vorwissen über die Umgebung gewonnen werden. Das am besten zu beobachtende Merkmal ist hierbei die Bodenoberfläche. Über die Jahre wurde hierzu eine Vielzahl von Arbeiten veröffentlicht, welche den Boden als Ebene modellieren und diese explizit in einem Nachbearbeitungsschritt rekonstruieren ([11], [12], [13]). Für die Ebenen-Rekonstruktion wird der Stand der Technik um drei Methoden erweitert: die Stützung der Bodenebene durch Fluchtpunkte, die Bodenrekonstruktion mithilfe der Methode der kleinsten getrimmten Fehlerquadrate nach der Umgebungsrekonstruktion sowie die in das Rekonstruktionsproblem integrierte Bodenoberflächenschätzung.
Zuletzt wird das System um einen LIght Detection And Ranging (LIDAR) Sensor erweitert. Die Kombination aus LIDAR und Kamera verbindet das Beste aus beiden Welten: hochgenaue, aber dünn besetzte Tiefenschätzung des LIDARs mit hoher Informationsdichte aus der Kamera, jedoch ohne Skaleninformation. Aufgrund zusätzlicher Herausforderungen wie die LIDAR-zu-Kamera-Kalibrierung ist dieses Feld noch weitgehend unerforscht. Neben Arbeiten zur Lokalisierung mit Kameras in einer Karte aus LIDAR-Punktmengen von Caselitz et al. [14] ist die einzige, jedoch sehr erfolgreiche Publikation zu Trajektorienschätzung mit Kamera und LIDAR bei Zhang et al. [15] zu finden. Dieser nutzt die Rotationsschätzung der Kamera jedoch lediglich als Initialisierung für klassische LIDAR-Bewegungsschätzung. In dieser Arbeit soll der Fokus auf die Bewegungsschätzung der Kamera gelegt werden, wobei nur die Tiefen aus dem LIDAR gewonnen werden, um die Grenzen von monokularem Bündelausgleich mit dünn besetzter Tiefe aus LIDAR auszuloten. Auf die Nachbereitung mit klassischen LIDAR-Trajektorienschätzungsalgorithmen wird daher explizit verzichtet.
Die Voraussetzung für die Fusion der Kamera und des LIDARs ist das Wissen um ihre relative Pose. Um diese zu schätzen, wird zusätzlich eine neuartige photometrische Methode zur LIDAR-Kamera-Kalibrierung vorgestellt.

1.2. Überblick

Diese Arbeit beleuchtet die verschiedenen Facetten von Visueller Odometrie. Hierzu wird das Visuelle Odometrie-System abstrahiert erläutert, um eine Übersicht über dessen einzelne Komponenten zu geben (Kapi-

tel 2). Nachdem einige fachliche Grundlagen der Bildverarbeitung in Kapitel 3 dargestellt wurden, werden die Einzelkomponenten des Systems anschließend fokussiert und vertiefend erläutert, beginnend mit der Frame-zu-Frame-Bewegungsschätzung aus einem Multikamerasystem in Kapitel 4. In Kapitel 5 wird die Darstellung des der Bewegungsschätzung zu Grunde liegenden Schätzproblems als Graph eingeführt und anhand dessen der Bündelausgleich formuliert. Heuristiken zur Landmarkenauswahl werden vorgeschlagen. Kapitel 6 stellt den umfassendsten Teil dieser Arbeit dar. Hierin werden verschiedene Methoden zur Wiedererlangung der Skale vorgestellt. Zuerst werden dabei drei Möglichkeiten vorgestellt, um Informationen aus der Umgebung, nämlich die Grundebene, zu nutzen:

1. Für Frame-zu-Frame-Methoden unter Zuhilfenahme von Fluchtpunkten.

2. Als Nachbearbeitungsschritt nach dem Bündelausgleich.

3. Integriert in das Optimierungsproblem des Bündelausgleichs.

Des Weiteren wird die Schätzung der Skale mithilfe von Informationen aus einem zusätzlichem Sensor, eines LIDARs, vorgestellt. Um diesen zusätzlichen Sensor nutzen zu können, wird eine neuartige Methode zur Laser-zu-Kamera-Kalibrierung in Kapitel 7 vorgestellt. Diese Methoden werden in Kapitel 8 anhand eines öffentlichen Benchmark-Datensatzes (KITTI-Datensatz [16]) und anhand von Sequenzen des Testfahrzeugs des Instituts für Mess- und Regelungstechnik des Karlsruher Instituts für Technologie quantitativ ausgewertet. Abschließend wird die Anwendung der vorgestellten Methoden in einem Totwinkel-Assistent für Lastkraftfahrzeuge illustriert und die Arbeit in Kapitel 10 zusammengefasst.

Kapitel 2

Übersicht

Die in Abbildung 2.1 gezeigte Struktur ist die allgemeinste Form für Bewegungsschätzung aus Kamerabildern. Als Eingang wird eine zeitliche Bildsequenz einer einzelnen Kamera entgegengenommen. Um die große Informationsmenge der Bilder handhabbar zu machen, werden Merkmale, wie zum Beispiel Ecken oder Kreise, aus den Bildern extrahiert und über die Sequenz verfolgt (Block A). Diese verfolgten Merkmale werden in Block B vorverarbeitet, wodurch grob fehlerhafte Daten zurückgewiesen werden können.
Im Zentrum der Methodik dieser Arbeit steht der sogenannte SLAM (Simultaneous Localization And Mapping, Block D). Diese Methodik rekonstruiert die zurückgelegte Trajektorie und die Umgebung des Fahrzeugs zur gleichen Zeit. Die verwendete Methode zur Lösung des SLAM-Problems ist der sogenannte Bündelausgleich. Mit der in Abschnitt 5 vorgestellten Problem-Modellierung als Graph kann der Bündelausgleich auch als Inferenz des Graphen in der Zeit-Dimension gesehen werden und wird hier zeitliche Inferenz genannt. Dies ist der rechenaufwendigste Block der Methodik.
Da die Bewegungsschätzung zwischen zwei Bildern (Frame-zu-Frame-Schätzung, Block C) den SLAM initialisiert, ist diese von großer Wichtigkeit für eine akkurate und schnelle Bewegungsschätzung. Daher wird eine Methode hierfür in Abschnitt 4 vorgestellt, welche zusätzlich mehrere Kameras verwenden kann, um die Robustheit zu steigern.
Die die in Block S verwendete Skaleninformation kann aus vielen Quellen gewonnen werden. Die wohl am besten untersuchte Quelle ist eine zweite Kamera, welche in einem bekannten Abstand zur ersten Kamera montiert

ist und in die gleiche Richtung zeigt (Beispiele hierfür sind [17], [18], [19], [20], [21]). Mit dieser sogenannten Stereokamera kann zu jedem Merkmal ein weiteres in der zweiten Kamera assoziiert werden. Durch den bekannten Abstand zwischen den Kameras kann die Skale bestimmt werden. Ein weitreichendes Problem hierbei ist die Kalibrierung der Kameras. Die geschätzte Tiefe der Merkmale ist sehr störungsanfällig gegenüber Winkelfehlern zwischen den Kameras. Schon kleine Änderungen der Kalibrierung können so die Tiefenschätzung stark verschlechtern und in der Praxis muss die Kalibrierung daher oft wiederholt oder korrigiert werden. Daher fokussiert sich diese Arbeit auf andere Skaleninformationsquellen:

1. Größenverhältnisse in der Umgebung.
2. Bewegungsrelation zwischen Sensoren und Fahrzeug.
3. Tiefeninformation aus LIDAR.

Die Methoden hierzu werden in Abschnitt 6 vorgestellt. Diese Skaleninformation wird zusammen mit der monokularen Bewegungsinformation aus dem Kamerabild in den Block der zeitlichen Inferenz (Block D) eingespeist. Die wesentlichen Bestandteile dessen und Verbesserungsvorschläge werden in Abschnitt 5 dargestellt. Die Schleifendetektion (Block E) ist zwar fester Bestandteil für Bewegungsschätzungs-Algorithmen, um zum Beispiel eine Strecke zu kartieren, ist für Visuelle Odometrie auf Fahrzeugen jedoch unnötig, da im Gebrauch eines Fahrzeugs eine wiederholte Befahrung vermieden wird. Daher wird diese hier nicht näher betrachtet. Für etablierte Algorithmen sei auf Galvez et al. [22] verwiesen.

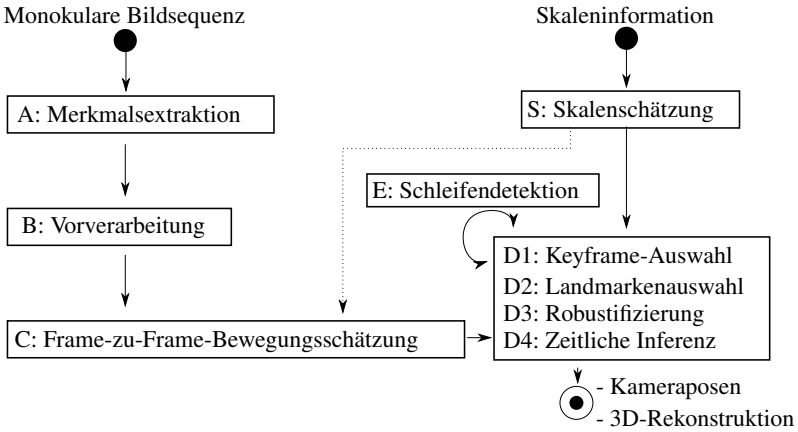


Abbildung 2.1.: Struktur der Bewegungsschätzung aus einer monokularen Bildsequenz.

Kapitel 3

Grundlagen

3.1. Projektion auf die Bildebene

Die dreidimensionale Umgebung wird in Kameras auf einem zweidimensionalen Bild beobachtet. Wie ein Punkt $\mathbf{x}$ der Umgebung auf einen Punkt $\mathbf{p}$ im Bild abgebildet wird, hängt dabei von vielen physikalischen Faktoren ab, wobei die Eigenschaften der Linse und des Sensorchips maßgeblich sind. Um diese komplexen Systeme zu modellieren, werden Kameramodelle eingeführt, welche möglichst präzise und einfach eine Beziehung zwischen dreidimensionaler Umgebung und Bildpunkt geben sollen. In dieser Arbeit wird diese Abbildung mit

$$\mathbf{p} = \pi(\mathbf{x}) \tag{3.1}$$

bezeichnet. Üblicherweise wird $\pi(\mathbf{x})$ im sogenannten Kamerakoordinatensystem beschrieben, in welchem die x-y-Ebene parallel zur Kamerabildebene ist, und somit die z-Achse die Tiefe von $\mathbf{x}$ beschreibt. Dieses Koordinatensystem ist in Abbildung 3.1 zu sehen. Für die einfachere mathematische Handhabbarkeit ist $\mathbf{p}$ hierbei in homogenen Koordinaten angegeben. Eine Beschreibung dieser ist zum Beispiel bei Szeliski et al. [8] zu finden. Zudem ist eine weitere notwendige Bedingung für die Bewegungsschätzung, dass auch $\mathbf{x}' = \pi^{-1}(\mathbf{p})$ definiert ist. Aufgrund der projektiven Eigenschaften ist das jedoch nur bis auf einen Skalierungsfaktor möglich, was durch die Notation $\mathbf{x}'$ angedeutet ist. Im Allgemeinen gilt $\mathbf{x}' = Aufpunkt + \frac{Richtung}{\|Richtung\|_2} \cdot s$, wobei s die unbekannte Tiefe von $\mathbf{x}$ bezeichnet.

Viele Kameras können durch eine Lochkamera angenähert werden. Für diese lautet die Projektionsgleichung:

$$\mathbf{p} = \mathbf{K}\frac{\hat{\mathbf{x}}}{z}, \tag{3.2}$$

mit der intrinsischen Matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$, dem dreidimensionalen Punkt $\hat{\mathbf{x}} = (x,y,z)^T$ und dem projizierten Punkt auf der Bildebene $\mathbf{p}$ in homogenen Koordinaten. $\frac{\hat{\mathbf{x}}}{z}$ beschreibt hierbei den Sichtstrahl zu $\hat{\mathbf{x}}$ mit Normalisierung zu homogenen Koordinaten.[1] Für mehr Informationen über Kameramodelle sei auf Szeliski et al. [8] verwiesen.

3.2. Epipolargeometrie

Die Epipolargeometrie ist eine geometrische Beschreibung des Zusammenhangs zweier Kameras, welche eine Szene aus unterschiedlichen Blickwinkeln betrachten. Sie wird in vielen Bereichen der visuellen Perzeption verwendet und ist ein klassisches Werkzeug der Visuellen Odometrie. Eine die Epipolargeometrie erklärende Skizze ist in Abbildung 3.1 zu sehen. Ein Punkt $\mathbf{x}$ wird darin aus zwei Kameraperspektiven beobachtet. Somit kann ein Dreieck durch den Brennpunkt der ersten Kamera $\mathbf{o}_\tau$, den Punkt $\mathbf{x}$ und den Brennpunkt der zweiten Kamera $\mathbf{o}_\upsilon$ aufgespannt werden. Epipolargeometriebasierte Fehlermetriken bestrafen Messungen, welche von diesem Dreieck abweichen. Eine perfekte Erfüllung dieser sogenannten Epipolarbedingung [6] lässt sich in Gleichung

$$\hat{\mathbf{p}}_\tau^T \mathbf{E} \hat{\mathbf{p}}_\upsilon = 0 \tag{3.3}$$

formulieren, wobei $\hat{\mathbf{p}}_\tau$ und $\hat{\mathbf{p}}_\upsilon$ die Sichtstrahlen der zu $\mathbf{x}$ korrespondierenden Messungen bezeichnen. Ein Sichtstrahl ist dabei als der normierte Richtungsvektor zwischen dem Kameraursprung und dem beobachteten Punkt definiert. Somit gilt

$$\hat{\mathbf{p}}_i = \frac{\mathbf{x} - \mathbf{o}_i}{\|\mathbf{x} - \mathbf{o}_i\|_2}. \tag{3.4}$$

[1] Befindet sich der Punkt $\mathbf{x}$ nicht im Kamerakoordinatensystem, so muss dieser zuerst mithilfe einer Transformation mit den Parametern $\mathbf{P}$ in das Kamerakoordinatensystem überführt werden. Die Projektionsgleichung ist dann mit $\pi(\mathbf{x}, \mathbf{P})$ bezeichnet.

Die sogenannte Essentielle Matrix $\mathbf{E}$ beschreibt die Ebene, welche durch die Bewegung

$$\mathbf{M} = \begin{pmatrix} r_{0,0} & r_{0,1} & r_{0,2} & t_0 \\ r_{1,0} & r_{1,1} & r_{1,2} & t_1 \\ r_{2,0} & r_{2,1} & r_{2,2} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.5}$$

zwischen den Bildern und einem Sichtstrahl aufgespannt wird. $\mathbf{R}(\mathbf{t} \times \hat{\mathbf{p}}_v)$ ist die Richtung des Normalvektors dieser sogenannten Epipolarebene (Dreieck $\mathbf{o}_\tau \mathbf{x} \mathbf{o}_v$ in Abbildung 3.1) und die Essentielle Matrix wird als

$$\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times \tag{3.6}$$

definiert, wobei $[\cdot]_\times$ die Matrixform des Kreuzprodukts darstellt. Der Fehler

$$\epsilon = \frac{\hat{\mathbf{p}}_\tau^T \mathbf{E} \hat{\mathbf{p}}_v}{\|\mathbf{E}\hat{\mathbf{p}}_v\|_2} \tag{3.7}$$

ist somit der Cosinus des Winkels zwischen Sichtstrahl und der Normalen der Epipolarebene und ist ähnlich zur Epipolarbedingung (Gleichung 3.3). Hierbei ist zu beachten, dass die Länge des Translationsvektors (die sogenannte Skale) durch dieses Fehlermaß nicht bewertet werden kann, da diese durch die Normalisierung keinen Einfluss hat:

$$\epsilon = \frac{\hat{\mathbf{p}}_\tau^T \mathbf{R}[\mathbf{t}]_\times \hat{\mathbf{p}}_v}{\|\mathbf{R}[\mathbf{t}]_\times \hat{\mathbf{p}}_v\|_2} = \frac{\hat{\mathbf{p}}_\tau^T \mathbf{R}\left[\frac{\mathbf{t}}{\|\mathbf{t}\|_2}\right]_\times \hat{\mathbf{p}}_v \cdot \|\mathbf{t}\|_2}{\|\mathbf{R}\left[\frac{\mathbf{t}}{\|\mathbf{t}\|_2}\right]_\times \hat{\mathbf{p}}_v\|_2 \cdot \|\mathbf{t}\|_2} = \frac{\hat{\mathbf{p}}_\tau^T \mathbf{R}\left[\frac{\mathbf{t}}{\|\mathbf{t}\|_2}\right]_\times \hat{\mathbf{p}}_v}{\|\mathbf{R}\left[\frac{\mathbf{t}}{\|\mathbf{t}\|_2}\right]_\times \hat{\mathbf{p}}_v\|_2}. \tag{3.8}$$

Bis hierhin sind noch keine Annahmen an das Kameramodell gestellt. Somit ist Gleichung 3.7 auch für komplexe Kameramodelle gültig, welche zum Beispiel für Fischaugenkameras und katadioptrische Kameras nötig sind.
Eine gebräuchlichere Form der Gleichung 3.3 ist der Spezialfall für ein Lochkameramodell. Für ein solches gilt die Projektionsgleichung 3.2. Invertiert man diese, erhält man $\hat{\mathbf{p}} = z\mathbf{K}^{-1}\mathbf{p}$ und mit Gleichung 3.3 folgt ${z_\tau \mathbf{p}_\tau}^T \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \mathbf{p}_v z_v = 0$. Teilt man durch die Tiefen z_τ und z_v, erhält man somit

$${\mathbf{p}_\tau}^T \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \mathbf{p}_v = 0. \tag{3.9}$$

Hierbei wird die Fundamentalmatrix als

$$\mathbf{F} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \tag{3.10}$$

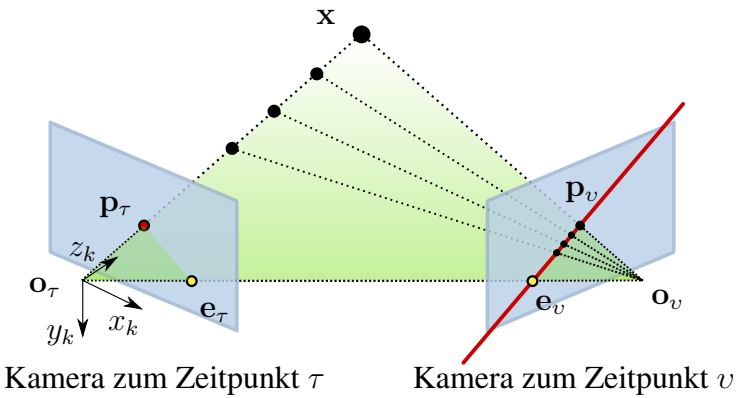


Abbildung 3.1.: Skizze der Epipolargeometrie zweier Kamerabilder (Quelle: [23], Notation verändert). Der Punkt $\mathbf{x}$ wird aus zwei Kameras zu Zeitpunkten τ und υ in Bildpunkten $\mathbf{p}_\tau$, $\mathbf{p}_\upsilon$ beobachtet. Die Epipolarlinie (rot) verläuft durch den Bildpunkt $\mathbf{p}_\upsilon$ und den zugehörigen Epipol $\mathbf{e}_\upsilon$. Das Kamerakoordinatensystem der ersten Kamera $\mathbf{o}_\tau$, welches zudem in ihrem Brennpunkt liegt, ist exemplarisch durch (x_k,y_k,z_k) dargestellt.

definiert. Der in Gleichung 3.7 beschriebene Winkelfehler definiert sich somit zu

$$\zeta' = \frac{z_\tau \mathbf{p}_\tau^T \mathbf{F} \mathbf{p}_\upsilon z_\upsilon}{\|\mathbf{E}\mathbf{K}^{-1}\mathbf{p}_\upsilon\|_2 z_\upsilon}. \tag{3.11}$$

Damit unterliegt ζ' einer zusätzlichen Skalierung mit der Tiefe des Punktes $\mathbf{x}$. Die Schnittgerade der Epipolarebene mit der Bildebene wird als Epipolarlinie $\mathbf{l} = \mathbf{F}\mathbf{p}$ bezeichnet. Für Lochkameras ist der Fehler der Epipolarlinie zum Bildpunkt gebräuchlich:

$$\zeta = \frac{\mathbf{p}_\tau^T \mathbf{F} \mathbf{p}_\upsilon}{\|\mathbf{F}\mathbf{p}_\upsilon\|_2}. \tag{3.12}$$

Der Nenner fungiert hierbei als eine Normalisierung im Bildraum. Dies ist also ζ' unter Vernachlässigung der Skalierung durch die Tiefe des Punktes $\mathbf{x}$.
Diese Fehlermetrik ist im Allgemeinen nicht-linear. Für eine leichtere Optimierung des Epipolarfehlers schlugen Hartley und Zisserman [6] vor, die Kostenfunktion in eine lineare Darstellung zu transformieren. Dies geschieht, indem die durch den Faktor $\sqrt{(\mathbf{F}p_\upsilon)_I^2 + (\mathbf{F}p_\upsilon)_{II}^2}$ ausgeführte Normalisierung nicht in der Kostenfunktion geschieht. Stattdessen wer-

den die Messungen $\mathbf{p}_{\tau,\upsilon}$ oder die Fundamentalmatrix $\mathbf{F}$ direkt normalisiert ([6], [7]). Die lineare Fehlermetrik ist somit als

$$\zeta_{\text{Linear}} = {\mathbf{p}_\tau}^T \mathbf{F} \mathbf{p}_\upsilon \tag{3.13}$$

formuliert.
Für eine umfassende Studie über die Epipolargeometrie und ihre Anwendungen sei auf Hartley und Zissermann [6] verwiesen.

Anmerkung In diesem Kapitel sind die zwei Kameras durch Indizes $\ldots_\tau$ und $\ldots_\upsilon$ für dieselbe Kamera zu unterschiedlichen Zeitpunkten dargestellt. Die Epipolargeometrie ist jedoch auf vielerlei Kameraaufbauten anwendbar. Zum Beispiel spielt diese eine wichtige Rolle in der Stereobildverarbeitung, für welche $\ldots_\tau$ und $\ldots_\upsilon$ die linke und die rechte Kamera darstellen würden.

3.3. Methode der kleinsten Quadrate

Die Methode der kleinsten Quadrate (Least-Squares, LS) ist eine der bedeutendsten Schätzmethoden in der Statistik. Simultan von Gauß und Legendre entwickelt, wurde diese erstmalig im frühen 19. Jahrhundert verwendet, um das Wiederauftauchen des Jupitermonds Ceres zu berechnen. Ihr Prinzip ist simpel: die Parameter einer Modellfunktion f werden so an die Messungen m_i angepasst, dass die quadratische Abweichung von f zu jeder Messung minimal wird. Dies ist ein Optimierungsproblem, welches wie folgt formuliert wird:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_i \left(f(x_i, \boldsymbol{\alpha}) - m_i\right)^2. \tag{3.14}$$

Hierbei bezeichnet x_i den Punkt der Modellfunktion, an welchem m_i beobachtet wurde und $\boldsymbol{\alpha} = \alpha_0, \alpha_1, \ldots$ deren zu schätzende Parameter. Für ein lineares Modell wären das zum Beispiel Aufpunkt und Richtung der Geraden.
Insbesondere für lineare Probleme hat die Methode der kleinsten Fehlerquadrate große Bedeutung. So besagt das Gauß-Markov-Theorem, dass das LS-Verfahren ein minimalvarianter, linearer, erwartungstreuer Schätzer ist, falls die Störgrößen des Systems unkorreliert sind und zudem einer Normalverteilung mit konstanter Varianz und Erwartungswert 0 gehorchen. In diesem Fall ist der LS-Schätzer, als der Schätzer mit minimaler Kovarianz, der bestmögliche Schätzer.
Diese Eigenschaften sind gültig für normalverteilte Messungen. In realer

Anwendung ist dies nicht der Fall — insbesondere verzerren Ausreißer das Schätzergebnis stark. Darum sind Methoden zur Robustifizierung dieser Methode entscheidend für ihre Anwendbarkeit. Die für diese Arbeit wichtigsten Robustifizierungsmethoden sind in den Abschnitten 3.3.1 und 3.3.2 beschrieben.

3.3.1. Methode der kleinsten getrimmten Fehlerquadrate

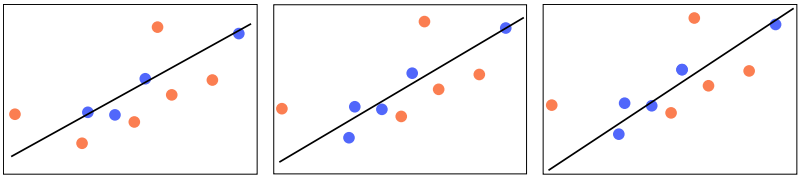


Abbildung 3.2.: Beispiel für einen LTS-Schätzer mit fixem Schwellwert. Nach der Startschätzung links (schwarz) werden Messungen bestimmt, welche nicht zu dieser Schätzung passen (orange). Mit den restlichen Messungen, den sogenannten *Inliern*, wird eine weitere Schätzung gemacht (Mitte) und eine neue Inliermenge bestimmt, welche zu einer weiteren Schätzung führt (rechts). Dies wird iterativ durchgeführt, bis sich die Schätzung nur noch wenig ändert.

Dieses Verfahren ist gemeinhin bekannt als Least-Trimmed-Squares-Methode (LTS). Hierin wird der Einfluss von Ausreißern entfernt, indem iterativ die Menge der Messungen gesucht wird, welche dem Modell am besten entspricht. Messungen, welche dem Modell nicht entsprechen, werden entfernt (*getrimmt*). Ein Beispiel ist in Abbildung 3.2 gezeigt. Zu Beginn wird eine grobe Schätzung als Startwert angenommen. Damit werden stark abweichende Messungen detektiert. Dies kann zum Beispiel durch einen fixen Schwellwert geschehen oder indem ein Anteil der Messungen entfernt wird (zum Beispiel 10% der Messungen mit den größten Abweichungen.). Die übrige Menge gültiger Messungen wird *Inlier* genannt. Mithilfe der Inlier wird eine neue LS-Schätzung erstellt und es werden wiederum Messungen entfernt. Um die Konvergenz des Algorithmus zu verbessern, werden hierbei auch Messpunkte untersucht, welche in einer vorherigen Iteration ausgeschlossen wurden. Dies wird solange wiederholt, bis die Änderung der Schätzung klein wird. Um zu verhindern, dass die Schätzung in einen Bereich außerhalb des Einflusses der Messungen wandert, können konstante Strafterme für jede entfernte Messung eingefügt werden.

3.3.2. M-Schätzer

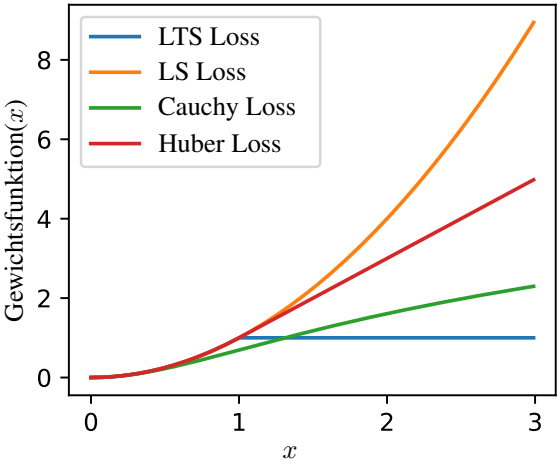


Abbildung 3.3.: Verschiedene populäre Gewichtsfunktionen in Abhängigkeit des Funktionswertes x für M-Schätzer. LTS Loss (blau) ist die Gewichtsfunktion, mit welcher ein M-Schätzer in eine spezielle Form des LTS überführt werden kann. Für LS Loss (orange) ist das Schätzproblem der nicht robuste LS-Schätzer.

In Abschnitt 3.3.1 wurde die LTS-Methode vorgestellt, welche Ausreißer aus dem Schätzproblem entfernt. Sind Ausreißer schwierig von Inliern zu unterscheiden, so kann es besser sein, den Einfluss von Ausreißern nicht komplett zu entfernen, sondern diesen abzuschwächen. Hierzu wird Gleichung 3.14 um einen Term $\rho(x)$ erweitert, welcher den Einfluss großer Residuen abschwächt:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_i \rho((f(x_i, \boldsymbol{\alpha}) - m_i)). \tag{3.15}$$

Hierbei wird $\rho(x)$ so gewählt, dass das Schätzproblem nicht von großen Residuen dominiert wird, sondern Messungen weniger gewichtet werden, je weiter sie von der aktuellen Schätzung abweichen. Gegeben einem guten Startwert kann so der Einfluss von Ausreißern reduziert werden, ohne diese ganz zu entfernen. Der nicht robuste LS-Schätzer kann durch $\rho(x) = x^2$ formuliert werden. Die Wahl von $\rho(x)$ ist entscheidend für das Konvergenzverhalten des Schätzproblems. Insbesondere deren Parametrisierung ist schwierig, da für eine zu starke Reduktion des Gewichts das Problem sich schnell in lokalen Minima verfängt, wohingegen bei zu geringer Re-

duktion des Residuengewichts Ausreißer Einfluss auf den Schätzwert haben können.

Daher existiert eine Vielzahl von Funktionen, welche hierfür eingesetzt werden. Klassische Beispiele hierfür sind der sogenannte Huber-Loss oder der Tukey-Loss. In letzter Zeit wurde die Cauchy-Funktion hierfür immer beliebter, welche eine starke Reduktion des Einflusses großer Residuen bewirkt und ohne Fallunterscheidung ableitbar ist. Beispielfunktionen für $\rho(x)$ sind in Abbildung 3.3 gezeigt. Darin ist auch eine Funktion $\rho(x)$ gezeigt, für welche der M-Schätzer gleich dem LTS-Schätzer mit fixem Schwellwert und konstantem Fehlerterm pro entfernter Messung ist.

3.4. RANdom SAmple Consensus (RANSAC)

RANSAC ist eine häufig genutzte Robustifizierungsmethode für eine Vielzahl von Schätzproblemen. Diese Methode wurde durch Fischler et al. [24] das erste Mal beschrieben. Ähnlich der in Abschnitt 3.3.1 beschriebenen Methode der kleinsten getrimmten Fehlerquadrate (LTS) entfernt sie Messungen aus dem Schätzproblem, anstatt sie geringer zu gewichten. Im Kontrast zu LTS werden die Ausreißer jedoch nicht mit einem iterativ verfeinerten Modell gewonnen, sondern zufällig gezogen. Das Verfahren wird in Algorithmus 1 beschrieben. Durch die zufällige Ziehung der Messungen kommt diese Methode sehr gut mit einer hohen Anzahl an Ausreißern zurecht und benötigt zudem keinen Startwert für das Modell. Ihr größter Nachteil ist jedoch, das durch das zufällige Ziehen der Messungen viele Iterationen benötigt werden, um das globale Minimum mit hoher Wahrscheinlichkeit zu finden. Eine Formel für die Anzahl an Iterationen für ein bekanntes stochastisches Problem sowie Details zu dieser Methode sind bei Szeliski et al. [8] zu finden.

3.5. Merkmalsextraktion im Bildraum

Die Basis für bildbasierte Bewegungsschätzungsalgorithmen ist der optische Fluss, welcher die Bewegung des Bildinhalts zwischen zwei Bildern beschreibt. Da es sehr rechenaufwändig ist, diesen für jedes Pixel im Bild zu schätzen, wird in dieser Arbeit nicht-dichter Fluss verwendet. Dabei wird im Allgemeinen wie folgt vorgegangen:

1. Es werden charakteristische Pixel, sogenannte *Keypoints*, im Bild identifiziert, die über eine Bildfolge gut wiederzufinden sind. Klassischerweise werden dafür *Ecken*, zum Beispiel mit dem Harris-

Data : Daten $\mathbb{D}$;
Modell $modell$;
Schwellwert s;
Iterationen n;
Result : Menge minimaler Datenpunkte für bestes Modell $\mathbb{M}_{Best}$;
Bestes Modell $\mathbb{M}_{Best}$;
Beste Anzahl Inlier N_{Best};
foreach $i \in n$ **do**
  Ziehe zufällig Menge minimaler Datenpunkte für Modell $\mathbb{M}$ aus $\mathbb{D}$;
  Berechne Modellhypothese $m = modell(\mathbb{M})$;
  Anzahl Inlier $N = 0$;
  foreach $d \in \mathbb{D}$ **do**
    Berechne Residuum $r = m(d)$;
    if $r < s$ **then**
      $N = N + 1$;
  end
  if $N > N_{Best}$ **then**
    $\mathbb{M}_{Best} = \mathbb{M}$;
    $N_{Best} = N$;
end

Algorithmus 1 : RANSAC-Algorithmus.

Eckendetektor [25] oder kreisförmige Strukturen, sogenannte *Blobs* genutzt.

2. Die Bildinformation der Region um das charakteristische Pixel wird mithilfe eines sogenannten *Deskriptors* charakterisiert.

3. Schritt 1 und Schritt 2 werden auf einem zweiten Bild ausgeführt.

4. Die Deskriptoren beider Bilder werden auf Ähnlichkeit verglichen und eine optimale Zuordnung von Keypoints wird gefunden.

Somit kann die Bewegung der Keypoints zwischen den Bildern, also der optische Fluss, wie in Abbildung 3.4 gezeigt, berechnet werden und in der folgenden Bewegungsschätzung verwendet werden. Keypoints und Deskriptoren werden zusammen als *Features* bezeichnet.

Gegensätzlich zur Keypoint-Identifizierung mit Ecken und Blobs, die allgemeiner wissenschaftlicher Konsens ist, gibt es keinen Standard für die Wahl des Deskriptors. Zur Berechnung der Keypoints wird die Antwort von Filtermasken ausgewertet, welche in den Abbildungen 3.5a und 3.5b

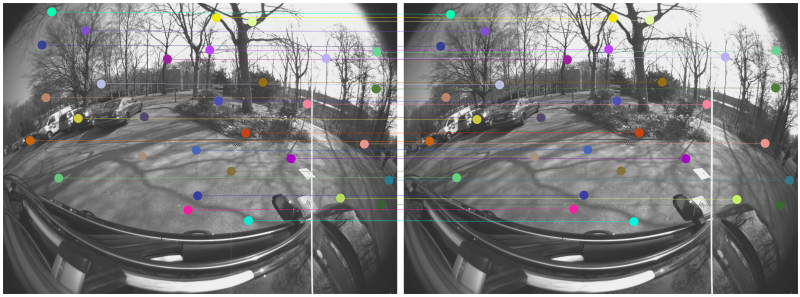


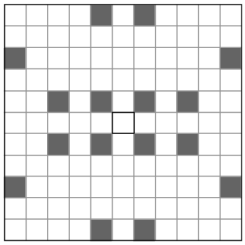
Abbildung 3.4.: Schätzung des Flusses aus zwei aufeinanderfolgenden Bildern. Hier sind, der Übersichtlichkeit wegen, nur einige assoziierte Features gezeigt. Für die Bewegungsschätzung werden typischerweise knapp 1000 dieser assoziierten Features benötigt.

-1	-1	-1	-1	-1
-1	+1	+1	+1	-1
-1	+1	+8	+1	-1
-1	+1	+1	+1	-1
-1	-1	-1	-1	-1

(a) Blob-Detektor

-1	-1	0	+1	+1
-1	-1	0	+1	+1
0	0	0	0	0
+1	+1	0	-1	-1
+1	+1	0	-1	-1

(b) Ecken-Detektor



(c) Vergleichsmuster des verwendeten Deskriptors

Abbildung 3.5a, 3.5b: Filtermasken zur Erkennung von Blobs und Ecken [17].
Abbildung 3.5c: Vergleichsmuster des in dieser Arbeit verwendeten Deskriptors von Geiger et al. [17]. Für einen Bildbereich von 11×11-Pixel werden Sobelfilter-Antworten in horizontaler und vertikaler Richtung verglichen. Um Rechenaufwand zu sparen, wird dies auf dem hier gezeigten Muster getan.

zu sehen sind. Wird ein Deskriptor benötigt, welcher rotationsinvariant ist, so wird die Hauptrichtung des Grauwertgradienten im Keypoint codiert und die Deskriptoren danach ausgerichtet. Skaleninvarianz wird durch die Ausführung der Extraktion auf Bildpyramiden implementiert. Der lange Zeit erfolgreichste Deskriptor ist SIFT [26], welcher die Bildregion um den Keypoint mithilfe von Histogrammen in Polarkoordinaten charakterisiert. SIFT kann die Merkmalszuordnung mit sehr guter Zuverlässigkeit lösen, ist jedoch verhältnismäßig rechenaufwendig. Von SIFT inspiriert, wurde der auch heute noch sehr häufig eingesetzte SURF-Deskriptor ent-

wickelt [27], welcher die Information im Bildausschnitt mit *Haar wavelets* approximiert und damit den Abgleich zweier Deskriptoren beschleunigt. Ein aktuell populärer Feature-Extraktor ist ORB [28]. Dieser nutzt einen sehr schnellen Eckendetektor, genannt FAST, in Verbindung mit dem BRIEF-Deskriptor [29]. Für BRIEF wird zuerst der Bildausschnitt um den Keypoint in kleinere Abschnitte unterteilt, für welche der geglättete Mittelwert der Grauwertintensitäten gebildet wird. Diese Abschnitte werden binär abgeglichen und zugeordnet, was sehr effizient durchgeführt werden kann. Das verwendete Muster für den Binärvergleich ist ausschlaggebend für die Wirksamkeit dieses Deskriptors. Weitere Einzelheiten sind der zugehörigen Veröffentlichung zu entnehmen [29].
Von diesen Standards abgesehen gibt es Deskriptoren, welche für spezifische Aufgaben erstellt wurden. Einer davon ist DIRD [30], der auf robuster Wiederkennung eines Keypoints zu verschiedenen Tageszeiten angelernt wurde.
Der in dieser Arbeit verwendete Deskriptor wurde von Geiger et al. vorgestellt [17]. Er besticht durch seine Einfachheit und die dadurch erreichte enorme Geschwindigkeit. Hierfür werden auf einem Bildbereich von 11×11-Pixel die Sobelfilter-Antworten in horizontaler und vertikaler Richtung für den Vergleich gebildet. Dieser denkbar einfache Deskriptor wird über die Summe der absoluten Differenzen abgeglichen. Um die Geschwindigkeit zu erhöhen, wird der Abgleich nicht auf den gesamten Block angewendet, sondern auf ein gleichmäßiges Muster von 16 Feldern, welches in Abbildung 3.5c dargestellt ist. Im Gegensatz zu den vorher vorgestellten Deskriptoren, werden so, anstatt weniger genauer Merkmale und Assoziationen, sehr viele, aber weniger genaue Merkmalsassoziationen hergestellt. Durch Plausibilisierung des Flusses werden anschließend Ausreißer erkannt und entfernt. Daher ist dieser Deskriptor für Visuelle Odometrie äußerst gut geeignet, da viele Punktassoziationen in kurzer Zeit erstellt werden können. Auf Rotations- und Skaleninvarianz wird explizit verzichtet, da diese für Visuelle Odometrie nur geringen Vorteil bieten, jedoch verhältnismäßig hohen Aufwand mit sich bringen.

Kapitel 4

Frame-zu-Frame-Bewegungsschätzung

Für eine akkurate sowie schnelle Lösung des aufwendigen Bündelausgleichsproblems ist eine sehr gute Frame-zu-Frame-Bewegungsschätzung von großer Wichtigkeit. Insbesondere trägt die Rotationsschätzung einen äußerst wichtigen Anteil dazu bei (Carlone et al. [5]). Die hier verwendete Frame-zu-Frame-Bewegungsschätzung erfolgt zuerst monokular mithilfe der in Abschnitt 3.2 dargestellten Epipolargeometrie. Hierbei wird die Kostenfunktion

$$\mathcal{E}_{\times}(\mathcal{X},\mathbf{M}) = \sum_{i} \epsilon_{\times}((\mathbf{p}_{i,\tau}, \mathbf{p}_{i,\upsilon}), \mathbf{M})^2 \tag{4.1}$$

minimiert, um die Frame-zu-Frame-Bewegung $\mathbf{M}$ mithilfe der Merkmalskorrespondenzen des Kamerabilds $\mathcal{X} = \{(\mathbf{p}_{0,\tau}, \mathbf{p}_{0,\upsilon}),(\mathbf{p}_{1,\tau}, \mathbf{p}_{1,\upsilon}), \dots\}$ zu schätzen [31]. Der Term $\epsilon_{\times}$ bezeichnet hierbei die für die Schätzung verwendete Fehlermetrik.
Nach einer Motivation durch den Stand der Technik in Abschnitt 4.1 wird in Abschnitt 4.2 das Optimierungsproblem zur Bewegungsschätzung als M-Schätzer formuliert. Anschließend wird in Abschnitt 4.3 darauf eingegangen, welchen Nutzen Bewegungsmodelle für die Bewegungsschätzung haben und wie diese integriert werden können. In Abschnitt 4.4 wird erläutert, wie die Bewegungsschätzung auf eine beliebige Anzahl von Kameras erweitert werden kann. Abschließend werden in Abschnitt 4.5 drei Kandidaten für Fehlermetriken $\epsilon_{\times}$ untersucht.

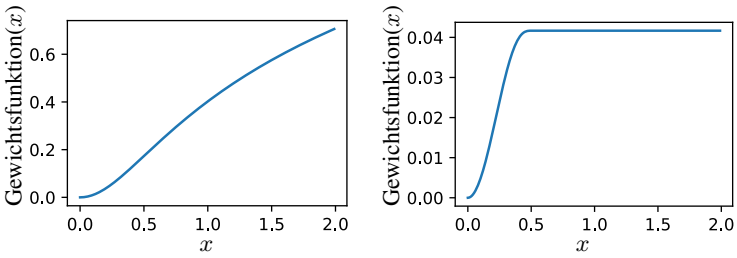
4.1. Stand der Technik

Die Standardmethode für epipolargeometriebasierte Bewegungsschätzung wurde von Hartley et al. [6] vorgestellt. In diesem sogenannten Acht-Punkt-Algorithmus wird die Bewegungsschätzung als linearisiertes Least-Squares-Problem aufgefasst. Hierbei wird der linearisierte Epipolarfehler $\mathcal{E}_{\mathrm{Linear}}$ verwendet. In einer perfekten Umgebung liefert diese Methode sehr gute Ergebnisse. Allerdings ist diese Formulierung sehr empfindlich gegenüber Ausreißern. Auf dem Stand der Technik wird dies durch stichprobenbasierte Ausreißerbehandlung mittels RANSAC [24] oder LMEDS [32] gelöst. Dafür müssen jedoch viele Bewegungshypothesen getestet werden, was diese Verfahren aufwendig macht.
Ein weiterer Nachteil des Acht-Punkt-Algorithmus sind seine acht Freiheitsgrade. Für intrinsisch kalibrierte Kameras mit voller dreidimensionaler Bewegung schlug Níster et al. den Fünf-Punkt-Algorithmus [33] vor, welcher mithilfe einer Eigenwertzerlegung die Freiheitsgrade auf fünf reduziert. Die weitere Reduktion der Freiheitsgrade, um zum Beispiel ein Fahrzeugmodell zu integrieren, ist jedoch nicht-trivial und auf sehr einfache Modelle beschränkt.
Diese Arbeit hat zum Ziel, die Integration von Bewegungsmodellen zu vereinfachen. Hierzu wird die Fehlerfunktion 4.1 so umformuliert, dass ein beliebiges Bewegungsmodell integriert werden kann. Die Ausreißerbehandlung wird zudem durch einen M-Schätzer umgesetzt, um Vorwissen über die Bewegung effektiv nutzen zu können.

4.2. Formulierung als nicht-lineares Optimierungsproblem

Die Umsetzung nicht-holonomer Bewegungsmodelle in der herkömmlichen Formulierung ist schwierig und die Ausreißerbehandlung aufwendig. Daher wurde in dieser Arbeit ein Verfahren entwickelt, welches auf die linearisierte Formulierung verzichtet und somit die folgenden wünschenswerten Eigenschaften aufweist:

1. Implizite Robustifizierung der Kostenfunktion durch einen M-Schätzer.
2. Modellierung der Bewegung auf einer Mannigfaltigkeit des $\mathbb{R}^6$.
3. Generalisierung auf kalibrierte Multikamerasysteme ohne überlappendes Sichtfeld.
4. Nutzung aller Kameramodelle, welche einen gemeinsamen Strahlenursprung aufweisen (Single-Viewpoint).



(a) Cauchy-Gewichtsfunktion (b) Tukey-Gewichtsfunktion

Abbildung 4.1.: Robuste Gewichtungsfunktionen für nichtlineare Optimierungsprobleme mit Kennzahl $a = 0{,}5$. Die Tukey-Loss-Gewichtsfunktion gewichtet Messungen deutlich stärker herunter als die Cauchy-Gewichtsfunktion. Daher wird die Tukey-Gewichtsfunktion nur angewendet, falls die Startlösung des Optimierungsproblems sehr nah an der Lösung liegt.

Wie vorher dargelegt, erkennen Methoden auf dem Stand der Technik Ausreißer mithilfe stichprobenbasierter Verfahren und können diese damit herausfiltern. Die in dieser Arbeit entwickelte robuste Kostenfunktion wird mit einer Gewichtungsfunktion $\rho(x)$ formuliert, welche große Residuen abschwächt. Prominente Kandidaten für $\rho(x)$ sind die Cauchy- oder Tukey-gewichtsfunktion. Die Cauchy-Gewichtsfunktion mit

$$\rho_\phi(x) = a^2 \cdot \log(1 + \frac{x^2}{a^2}) \tag{4.2}$$

ermöglicht moderate Gewichtsreduktion. Die Kennzahl a gibt hierbei an, ab welchem Fehler x die Messung als Ausreißer zu bewerten ist. Im Fall der Cauchy-Gewichtsfunktion entspricht das dem halben Quartilsabstand der Cauchy-Verteilung. Hat das Optimierungsproblem eine große Anzahl von starken Ausreißern, so ist die Tukey-Gewichtsfunktion mit

$$\rho_\theta(x) = \begin{cases} \frac{a^2}{6}(1 - (1 - \frac{x^2}{a^2})^3) & \text{für } x \leq a^2 \\ \frac{a^2}{6} & \text{sonst} \end{cases} \tag{4.3}$$

eine gute Wahl. Diese sind in Abbildung 4.1 dargestellt. Da diese Gewichte für jede Iteration des Optimierungsproblems dynamisch angepasst werden, ist diese Problemformulierung formal ein M-Schätzer. Die daraus folgende Kostenfunktion ist somit

$$\mathcal{E}_{\times,\text{Robust}}(\mathcal{X},\mathbf{M}) = \sum_i \rho(\epsilon_\times((\mathbf{p}_{i,\tau},\mathbf{p}_{i,\upsilon}),\mathbf{M})) \tag{4.4}$$

Eine Darstellung des Problems als lineares Least-Squares-Problem ist somit nicht mehr möglich und iterative Verfahren werden für die Optimierung benötigt. Daher können auch Fehlerfunktionen $\mathcal{E}_\times$ verwendet werden, welche nicht mehr linear sind. Die in dieser Arbeit untersuchten Kandidaten ζ, ϵ sind im Grundlagenkapitel 3.2 näher erläutert. Die Fehlerfunktion η ist die Sampsondistanz und ist bei Hartley und Zisserman [34] zu nachzulesen.

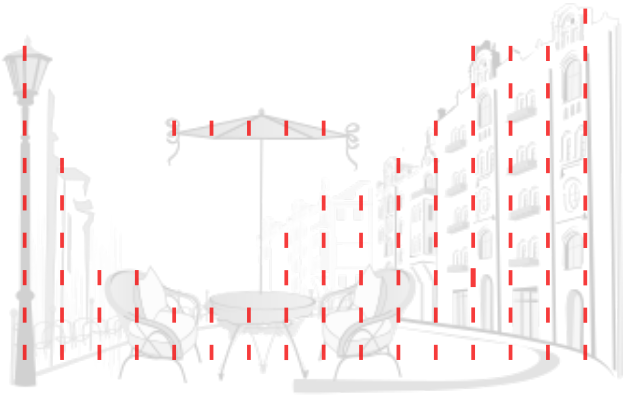
4.3. Nutzung von Bewegungsmodellen

Diese Arbeit zielt auf eine Anwendung der vorher beschriebenen Methoden in Fahrzeugen ab, welche nicht-holonome Bewegungen ausführen. Das System hat daher weniger als die vollen sechs Bewegungsfreiheitsgrade. Beschränkt man den Lösungsraum auf eine Mannigfaltigkeit des vollen sechsdimensionalen Raums, welche der Bewegung des Fahrzeugs besser entspricht, kann das Problem schneller gelöst werden und Ausreißer werden besser erkannt. Außerdem können somit uneindeutige Messungen besser interpretiert werden, wie in Abbildung 4.2 gezeigt ist.
Für Spezialfälle können solche Bewegungsmodelle umgesetzt werden [36]. Beliebige Modelle jedoch können mit der linearen Formulierung nicht verwendet werden.
Die in dieser Arbeit entwickelten Algorithmen sollen auf eine Vielzahl von Systemen flexibel anwendbar sein. Daher wird hierin eine Formulierung des Problems vorgeschlagen, welches für beliebige Bewegungsmodelle anwendbar ist.
In dieser Arbeit wird das kinematische Einspurmodell [37] verwendet, welches für die Modellierung der Bewegung üblicher, zweiachsiger Kraftfahrzeuge geeignet ist. Wird Schlupf vernachlässigt, kann das Modell zu einer ebenen Bewegung auf einer Kurve mit konstanter Krümmung (i. e. Kreis oder Gerade), wie in Abbildung 4.3 gezeigt, reduziert werden.

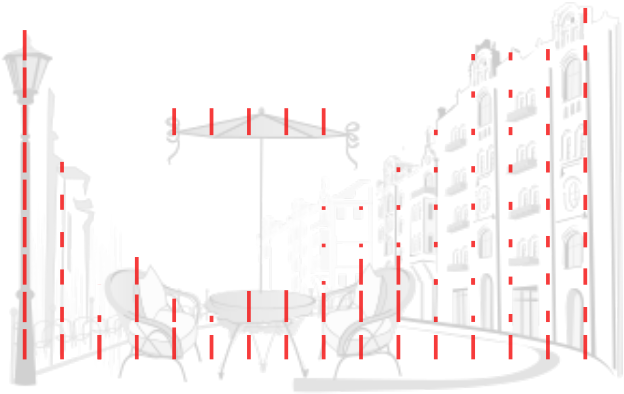
Mit dem Kreisradius $r = \frac{l}{\gamma}$ kann die Fahrzeugbewegung $\mathbf{M}(l,\gamma)$ wie folgt formuliert werden:

$$\mathbf{M}(l,\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & \sin(\gamma)r \\ \sin(\gamma) & \cos(\gamma) & 0 & (1-\cos(\gamma))r \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.5}$$

Um die Bewegung in drei Dimensionen zu beschreiben, wird dieses zweidimensionale Modell um Nicken und Wanken erweitert.



(a) Fluss für Nickbewegung.



(b) Fluss für Translation nach oben.

Abbildung 4.2.: Skizzierter Fluss (rot) für Nickbewegung und Translation nach oben, typischerweise z-Richtung. Vernachlässigt man die durch den Bildsensor erzeugten Quantisierungsfehler, so erzeugt die Nickbewegung Flussvektoren gleicher Länge. Die Translation nach oben erzeugt Fluss unterschiedlicher Länge, je näher der Punkt dem Horizont kommt, desto kürzer der Flussvektor. In einem Beispiel wie diesem, mit einer Szene mit variierender Tiefe, sind diese gut unterscheidbar. Ist die Szene jedoch vorwiegend in einer Ebene parallel zur Bildebene, so werden Nicken und Translation nach oben ununterscheidbar. In diesem Fall hilft das hier vorgestellte Bewegungsmodell den Freiheitsgrad nach oben einzuschränken, so dass das Nicken beobachtbar wird. Das Hintergrundbild von freedesignfile ist lizenziert unter Creative Commons (CC BY 2.0) [35].

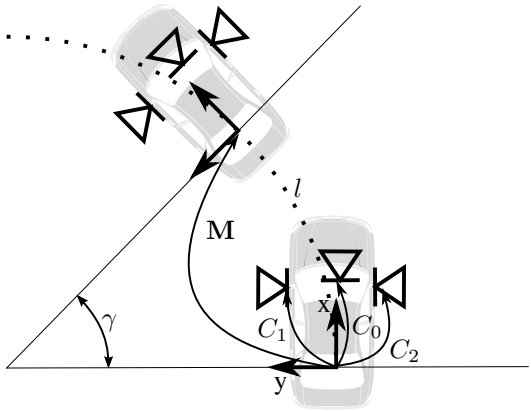


Abbildung 4.3.: Skizze der Bewegung zwischen zwei Zeitpunkten auf einer Kreisbahn. x und y sind globale Koordinaten, γ ist die Änderung des Gierwinkels, l ist die zurückgelegte Bogenlänge auf der Kreisbahn. Folgende Transformationen sind definiert: $\mathbf{M}$ als Transformation des Fahrzeugkoordinatensystems vom einen zum anderen Zeitpunkt; $C_{0\ldots2}$ als die extrinsischen Kalibrierungen der Kameras.

Das in Gleichung 4.5 gezeigte konkrete Bewegungsmodell für das Einspurmodell kann durch die nicht-lineare Formulierung des Problems durch ein beliebiges Modell ausgetauscht werden. Die einzige Bedingung an dieses ist die Existenz einer Abbildung von der Mannigfaltigkeit des Bewegungsmodells auf den sechsdimensionalen Bewegungsraum mit dreidimensionaler Translation sowie dreidimensionaler Rotation.
Es ist zu beachten, dass die hier verwendete Fehlermetrik auf der Epipolargeometrie beruht. Das bedeutet, dass a priori ein Parameter, nämlich die Länge des Translationsvektors (Skale), nicht beobachtet werden kann, wie in Abschnitt 3.2 erläutert wird. Diese Dimension muss also fix sein, was durch Nebenbedingungen gelöst wird. Hier ist das zum Beispiel die konstante Bogenlänge l für das Bewegungsmodell in Gleichung 4.5.
Um zuletzt das Bewegungsmodell in die Fehlerfunktion zu integrieren, wird die Fahrzeugbewegungshypothese mithilfe der extrinsischen Kamerakalibrierung $\mathbf{C}_0$ in das Kamerakoordinatensystem transformiert und angewendet, wie in Abbildung 4.3 dargestellt ist. Somit ist das Optimierungsproblem für eine einzelne Kamera durch

$$\underset{\gamma}{\operatorname{argmin}}\, \mathcal{E}_{\times}(\mathcal{X}_j, \mathbf{C}_0^{-1}\mathbf{M}(\gamma)\mathbf{C}_0) \tag{4.6}$$

formuliert. Hierbei ist zu beachten, dass das Fahrzeug als Starrkörper modelliert ist, sodass die extrinsische Kalibrierung als konstant angenommen wird.

4.4. Erweiterung auf Multikamerasysteme

In Abschnitt 4.3 wird beschrieben, wie eine einzelne Kamera mithilfe der extrinsischen Kalibrierung in die Kostenfunktion integriert werden kann. Damit kann das Problem vom Fahrzeugkoordinatensystem zu jedem Punkt im Raum ausgedrückt werden. Dies kann nun auch einfach auf Multikamerasysteme übertragen werden. Hierfür wird die Bewegungshypothese $\mathbf{M}$ mithilfe der extrinsischen Kalibrierung in jede Kamera überführt und dort evaluiert. Für Multikamerasysteme ist das Optimierungsproblem wie folgt darstellbar:

$$\underset{\gamma}{\operatorname{argmin}} \sum_{j=0}^{N-1} \mathcal{E}_{\times}(\mathcal{X}_j, \mathbf{C}_j^{-1}\mathbf{M}(\gamma)\mathbf{C}_j), \tag{4.7}$$

wobei N die Anzahl der Kameras ist, $\mathcal{X}_j$ die Punktkorrespondenzen und $\mathbf{C}_j$ die extrinsische Kamerakalibrierung beschreiben. Die Nutzung mehrerer Perspektiven hat folgende Vorteile:

- Da die Umgebung von vielen verschiedenen Sichtpunkten aus betrachtet wird, beruht die Schätzung auf aussagekräftigeren Messungen.
- Der Effekt von Blendungen, Verschmutzungen und Regentropfen auf den Linsen der Kameras sowie komplette Verdeckung einzelner Kameras, zum Beispiel durch andere Fahrzeuge, kann durch zeitgleiche Beobachtungen aus unterschiedlichen Perspektiven ausgeglichen werden.

4.5. Auswahl der Fehlermetrik durch simulierte Eingangsdaten

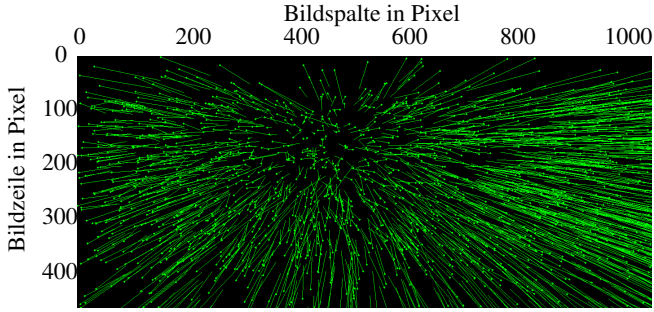
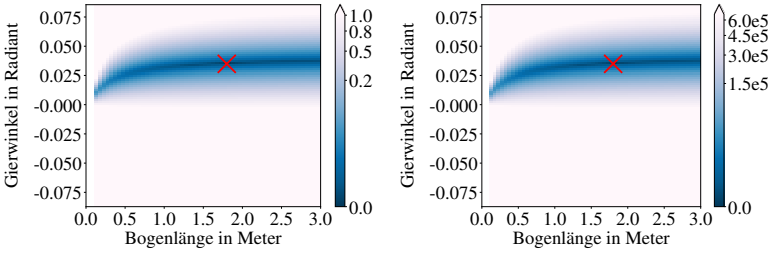


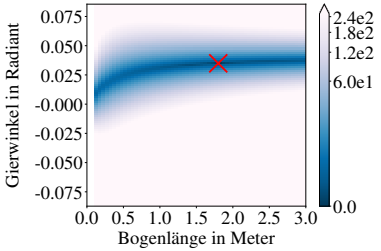
Abbildung 4.4.: Simulierter Fluss für eine Kamerabewegung auf einer Kreisbahn mit Kreiswinkel $\gamma = 2^\circ$ und Bogenlänge $l = 2\ \mathrm{m}$. Normalverteiltes Rauschen mit der Standardabweichung $\sigma = 5$ Pixel wird auf die Anfangs- und Endpunkte der Flussvektoren addiert.

In der Literatur sind verschiedene Fehlermetriken bekannt, welche für $\mathcal{E}_\times(\mathcal{X}_j, \mathbf{C}_j^{-1}\mathbf{M}\mathbf{C}_j)$ in Frage kommen. Zwei davon sind in Kapitel 3, Abschnitt 3.2 dargelegt. Eine weitere hier untersuchte Fehlermetrik ist die Sampson-Distanz, welche eine Approximation ersten Grades des in Gleichung 3.11 formulierten Fehlers darstellt. Für eine Herleitung sei auf Hartley und Zissermann [6] verwiesen. Um deren Vor- und Nachteile für das hier vorliegende Optimierungsproblem zu quantisieren, werden im Rahmen dieser Arbeit der optische Fluss simuliert und geeignete Fehlermetriken evaluiert. Für die Simulation werden eine Boxwelt erstellt und auf der Oberfläche jeder Box zufällig Punkte gezogen. Mit einer vorgegebenen Kamerabewegung und Rückprojektion ins Kamerabild kann so der optische Fluss extrahiert werden. Zudem wird normalverteiltes Rauschen auf die



(a) Fehler ζ

(b) Fehler ϵ



(c) Fehler η

Abbildung 4.5.: Fehlerlandschaft für den in Abbildung 4.4 gezeigten optischen Fluss. Gierwinkel und Bogenlänge beziehen sich auf die Eigenbewegung auf einer Kreisbahn. Die Fehlerlandschaften von ζ und ϵ sind quasi identisch. Für η hingegen ist das Tal bei kleinen Bogenlängen breiter. Dies deutet darauf hin, dass die Linearisierung der Sampson-Distanz bei kleinen Bogenlängen ungültig wird.

Anfangs- und Endpunkte der Flussvektoren addiert. Die Kamera wird als Lochkamera angenommen. Ein Beispiel für die so simulierten Eingangsdaten ist Abbildung 4.4 zu entnehmen. Zur Auswahl der am besten für das Problem geeigneten Fehlerfunktion wurde der so erzeugte optische Fluss auf die in Abschnitt 3.2 beschriebenen Fehlerfunktionen ϵ, ζ sowie auf die Sampson-Distanz η angewendet. Der Fehler wird für verschiedene Bewegungshypothesen, i. e. für verschiedene Bogenlängen und Gierwinkel, berechnet und aufgetragen, wie in Abbildung 4.5 dargestellt. Die so entstehende dreidimensionale Darstellung mit zwei Bewegungsparametern auf den Achsen und dem farbkodierten Fehler wird im Folgenden als Fehlerlandschaft bezeichnet.
In Abschnitt 3.2 wird der Zusammenhang zwischen ϵ und ζ beschrieben. Wie zu erwarten zeigen beide nahezu identische Fehlerlandschaften, mit einem gut erkennbaren Minimum und einer konvexen Fehlerlandschaft. Hier-

bei fällt auf, dass für η das Kostental bei kleinen Bogenlängen sehr viel breiter als bei ϵ und ζ ist, was auf die Ungültigkeit der bei dieser Fehlermetrik angewendeten Linearisierung nahe $l = 0\ \mathrm{m}$ hinweist. Der Vorteil von ϵ zeigt sich bei der Evaluation der Konvergenz-Geschwindigkeit. In Abbildung 4.6 ist zu sehen, dass die Bewegungsschätzung mit ϵ nach vier Iterationen deutlich näher am Minimum ist als die Bewegungsschätzung mit ζ. Ein weiterer großer Vorteil von ϵ ist, dass diese Metrik nicht auf den zweidimensionalen Messungen in der Bildebene beruht, sondern nur Sichtstrahlen im dreidimensionalen Raum verwendet. Somit können auch Kameramodelle verwendet werden, die nicht gut durch das Lochkameramodell abgebildet werden können, wie zum Beispiel Fischaugen-Kameras. Aufgrund dieser Eigenschaften wurde für Block C, siehe Kapitel 2, die Fehlermetrik ϵ gewählt, welcher den Cosinus des Sichtstrahls zur Epipolarebene beschreibt.

Hat die Kamera eine von dem Fahrzeugkoordinatenursprung abweichende Position, verändert die Fehlerlandschaft ihre Form. Somit wird auch über der Bogenlänge das Minimum bestimmbar und die Skale kann somit für ausreichend große Gierwinkel-Differenzen geschätzt werden. Eine genauere Analyse zu diesem Sachverhalt wird in Abschnitt 6.3 dargelegt.

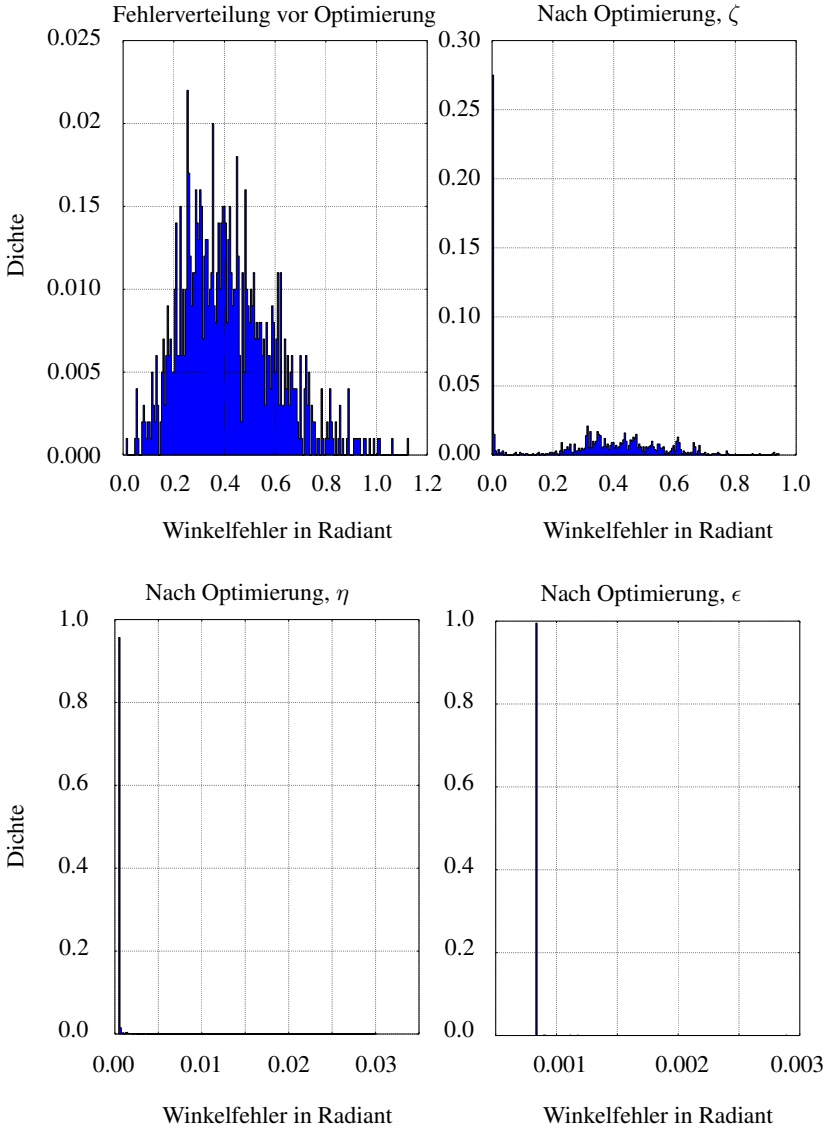


Abbildung 4.6.: Histogramm der Restfehler der Bewegungsschätzung zur Grundwahrheit nach vier Iterationen. Der Winkelfehler ist die Betragssummennorm der Abweichung von Roll-, Nick- und Wankwinkel. ϵ konvergiert am schnellsten, gefolgt von η und ζ.

Kapitel 5

Zeitliche Inferenz

Wird die Bewegungsschätzung, wie in Kapitel 4 beschrieben, zwischen zwei Zeitpunkten ausgeführt und zu einer Trajektorie kombiniert, so akkumulieren sich Fehler. Der so über die Zeit ansteigende Fehler wird als *Drift* bezeichnet. Damit die Trajektorienschätzung aber auch über lange Zeit genau bleibt, muss der Drift reduziert werden. Hierzu ist es sinnvoll, die Schätzung auf viele Zeitpunkte auszuweiten (Block D, Schaubild 2.1). Um die hier vorliegende Struktur des Schätzproblems besser zu veranschaulichen, ist die Darstellung in Form eines Graphen nützlich. Diese Repräsentation wird in Abschnitt 5.1 eingeführt und veranschaulichend auf die Frame-zu-Frame-Schätzung angewandt. Anschließend wird in Abschnitt 5.2 der Graph auf einen größeren Zeitraum erweitert. Dort wird das Optimierungsproblem formuliert, welches die im Graphen enthaltenen Informationen gemeinsam optimiert, um daraus eine Trajektorie mit minimalem Drift für den betrachteten Zeitraum abzuleiten. Werden die Messungen als Evidenz interpretiert, so entspricht diese Optimierung einer Inferenz des Graphen. Da diese Inferenz vor allem genutzt wird um den Graphen zeitlich konsistent zu machen, wird dies in dieser Arbeit als *zeitliche Inferenz* bezeichnet. Abschließend wird in den Abschnitten 5.3, 5.4 und 5.5 der Frage nachgegangen, wie die Komplexität dieses aufwendigen Optimierungsproblems reduziert und es robust gegen Ausreißer gemacht werden kann.

5.1. Bewegungsschätzungsproblem als Graph

Um Bewegungsschätzung graphisch darstellen zu können, ist die Darstellung als Graph, wie zum Beispiel von Triggs et al. [4] vorgestellt, nütz-

lich [38]. Hierzu werden die Kameraposen zu jedem Messzeitpunkt als Knoten eines Graphen aufgefasst. Diese werden im Folgenden als *Kameraknoten* bezeichnet. Zudem werden alle Strukturen in der Umgebung, welche Messungen erzeugen, auch als Knoten dargestellt, die hier mit *Landmarkenknoten* bezeichnet werden. Beobachtungen werden nun durch gewichtete Kanten zwischen Kamera- und Landmarkenknoten modelliert, wie in Abbildung 5.1 gezeigt — diese werden als *Messungskanten* bezeichnet. Fahrzeugknoten sind untereinander durch Kanten verbunden, welche die Posendifferenz beschreiben, die hier *Posenkanten* genannt werden. Aus Darstellungszwecken werden Kameraknoten, welche gleichen Zeitpunkten zugehören, zu *Fahrzeugknoten* zusammengefasst. Alle Kameraknoten sind mit dem zugehörigen Fahrzeugknoten durch Posenkanten verbunden, welche aus Darstellungsgründen gestrichelt dargestellt sind.

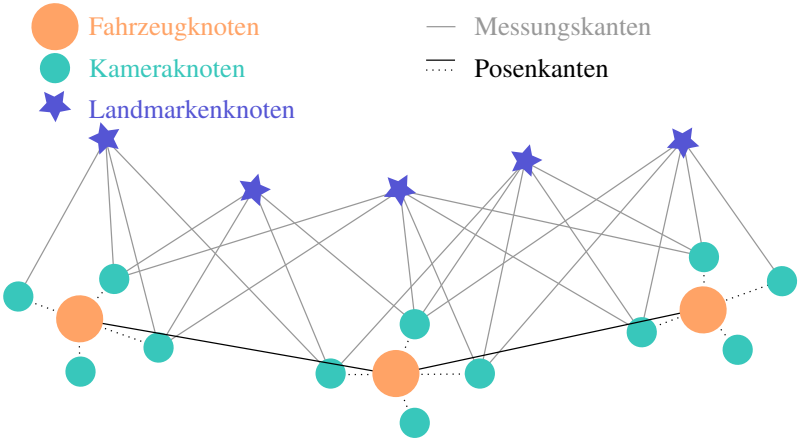


Abbildung 5.1.: Modellierung der gleichzeitigen Bewegungsschätzung und Umgebungsrekonstruktion (sogenannter *SLAM*) als Graph.

Die in Kapitel 4 beschriebene Frame-zu-Frame-Schätzung resultiert also in einem Graphen, welcher aus Teilgraphen G_p mit jeweils zwei Fahrzeugknoten besteht. Die Teilgraphen sind untereinander unverbunden, was in Abbildung 5.2 verdeutlicht ist. Bei Verwendung einer epipolargeometriebasierten Fehlerfunktion als Kantengewicht ist die Position der Landmarkenknoten für die Kameraknotenschätzung unbedeutend und wird daher nicht zum Optimierungsproblem hinzugefügt. Das in Abschnitt 4.2 formulierte Optimierungsproblem schätzt daher die optimale Relativpose des zweiten Fahrzeugknotens zum ersten Fahrzeugknoten im Teilgraphen.

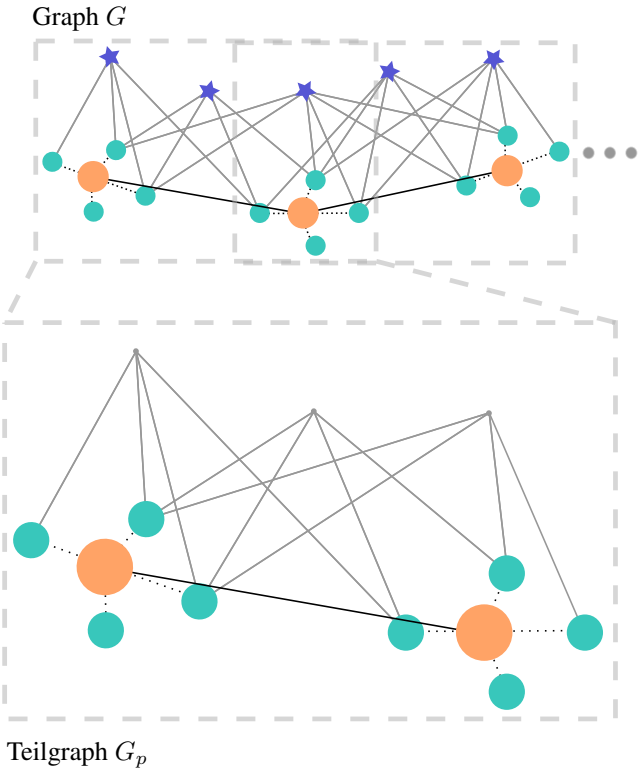


Abbildung 5.2.: Modellierung der Frame-zu-Frame-Bewegungsschätzung als Graph. Die Landmarkenknoten werden aus dem Problem entfernt, da eine epipolargeometriebasierte Fehlermetrik verwendet wird.

5.2. Erweiterung des Graphen auf mehrere Zeitpunkte

Wie in Kapitel 3.5 erläutert, sind jedoch Messungen in der Kamera nicht nur zwischen zwei, sondern einer Vielzahl von Bildern assoziierbar. Somit beinhaltet Gesamtgraph G Landmarkenknoten, welche mit einer Vielzahl von Kameraknoten verbunden sind, wie in Abbildung 5.1 dargestellt. Um die besten Knotenparameter für G zu finden, wird ein Optimierungspro-

blem in Abhängigkeit der Landmarkenknoten und Kameraknoten aufgestellt:

$$\underset{\mathbf{P}_j \in \mathcal{P}', \mathbf{l}_i \in \mathcal{L}'}{\operatorname{argmin}} \quad \sum_i \sum_j \|\phi(\mathbf{l}_i, \mathbf{P}_j)\|_2^2,$$
$$\phi(\mathbf{l}_i, \mathbf{P}_j) = \bar{\mathbf{l}}_{i,j} - \pi(\mathbf{l}_i, \mathbf{P}_j), \tag{5.1}$$

mit $\mathcal{P}'$ und $\mathcal{L}'$ als die Gesamtheit der Posen- und Landmarkenparameter. Hierbei sind $\mathbf{P}_j$ die drei translativen und die drei rotatorischen Parameter der Transformation vom Referenzsystem zum j-ten Fahrzeugknoten. Die Projektionsfunktion $\pi(\dots)$, siehe Abschnitt 3.1, transformiert die Landmarken von der dreidimensionalen Domäne in Bildkoordinaten. Dabei bezeichnet $\bar{\mathbf{l}}_{i,j}$ die Beobachtung der i-ten Landmarke im j-ten Bild.

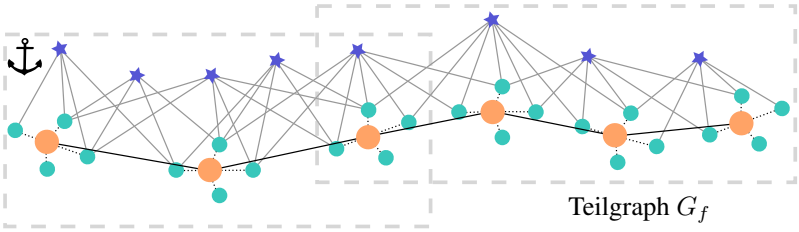


Abbildung 5.3.: Um die Komplexität des Optimierungsproblems zu verringern, wird der Graph G des Gesamtproblems in zwei Teile zerlegt. Der Teilgraph mit aktuellen Posen G_f wird optimiert, wohingegen der Rest von G fixiert ist.

Durch die hohe Anzahl an Messungen ist das Problem überbestimmt und die Landmarkenknoten können zugleich optimiert werden. Werden die Landmarkenknoten mit in das Optimierungsproblem aufgenommen, spricht man von *Simultaneous Localization and Mapping (SLAM)*. Der Startwert für Kamera- und Landmarkenknoten ist hierbei von großer Wichtigkeit für die Konvergenz des Problems. Falls keine Tiefeninformationen vorhanden sind, hat die Poseninitialisierung noch größere Bedeutung, da die Startwerte der Landmarken durch Triangulation[1] geschätzt werden.
Löst man diese Gleichung für alle bekannten Posen und Landmarken gleichzeitig, erhält man eine in sich konsistente und optimale Lösung. Diese Lösung hat also minimalen Drift, da alle Informationen für Landmarken und

[1] Bei der Triangulation wird die Landmarkenposition berechnet, welche den Abstand zu allen ihren Sichtstrahlen minimiert. Für eine formale Beschreibung sei hierfür auf Szeliski et al. verwiesen [8].

Posen genutzt werden können. Allerdings hat das Optimierungsproblem eine sehr große Anzahl an Parametern. Um den Rechenaufwand zu verringern, wird daher der gesamte Graph G, wie in Abbildung 5.3 gezeigt, in zwei Teile zerlegt — die n neuesten Fahrzeugknoten G_f und der Rest von G. Der hintere Teil von G wird fixiert und nur G_f wird optimiert. So können die bereits optimierten, älteren Knoten aus dem Optimierungsproblem entfernt werden. Wird eine neue Beobachtung gemacht, so wird G_f um einen Zeitschritt verschoben und erneut gelöst. Dieses Optimierungsschema wird auch als *gleitendes Fenster* bezeichnet. Durch die Fixierung von G wird jedoch der Drift vergrößert, da mit den Parametern auch Fehler fixiert werden und nicht mehr ausgeglichen werden können.
Durch diese Methodik wird ein Dilemma ersichtlich: Auf der einen Seite sollte G_f so groß wie möglich sein, um die Reduktion des Drifts zu maximieren, auf der anderen Seite steigt der Rechenaufwand mit wachsender Fenstergröße. Um das bestmögliche Ergebnis mit geringem Rechenaufwand zu bekommen, muss also die im Graphen vorhandene Information verdichtet werden — Beobachtungen, welche nur wenig zur Lösung des Optimierungsproblems beitragen, sollen zurückgewiesen werden, um die für eine genaue Posenschätzung relevanten Informationen zu konzentrieren. Konkret sind das fehlerhafte und unpräzise Messungen sowie redundante Messungen zum Beispiel von nah beieinander liegenden Landmarken. Dies wird hier durch Landmarkenauswahl und die Auswahl von sogenannten *Keyframes* umgesetzt und in Abschnitt 5.3 und Abschnitt 5.4 beschrieben.

5.3. Keyframe-Auswahl

5.3.1. Nomenklatur

Frames sind konsekutive Zeitpunkte, zu welchen Beobachtungen assoziiert werden können[2]. Dies entspricht in der Regel dem Triggerzeitpunkt der Kameras. Ein *Keyframe* ist ein Frame, welcher für die zeitliche Inferenz ausgewählt wurde. Somit ist die Menge aller Keyframes stets eine Teilmenge aller Frames. Frames werden wie folgt kategorisiert:

1. Benötigt. Frames, welche für die Stabilität des Optimierungsproblems unabdingbar sind.

2. Zurückgewiesen. Frames, welche vorwiegend Informationen enthalten, die das Ergebnis der Bewegungsschätzung verschlechtern.

[2] Würden alle jemals definierten Frames verwendet werden, so folgte daraus das Problem in Gleichung 5.1.

3. Ausgedünnt. Frames, welche redundante Information enthalten und somit entfernt werden können, ohne das Ergebnis der Bewegungsschätzung wesentlich zu verschlechtern.

Diese Kategorisierung erfolgt ausschließlich auf den Eigenschaften der Frames für das Optimierungsproblem. Würden benötigte Frames entfernt (Punkt 1), so resultierte das in einem instabilen Optimierungsproblem mit lokalen Minima, welche unter Umständen stark von einer guten Lösung abweichen werden. Dies ist zum Beispiel der Fall, wenn der Winkel, aus welchem die Landmarken beobachtet werden, starken Änderungen unterliegt, wie in Kurven mit dichter Bebauung neben der Straße. Da in solchen Fällen der optische Fluss im Bild sehr groß werden kann, können Merkmale nicht mehr assoziiert und daher nur über einen kurzen Zeitraum beobachtet werden. Somit muss in einem solchen Fall die Dichte der Keyframes erhöht werden. In dieser Arbeit ist der Indikator dafür die Differenz der Fahrzeugorientierung aus der Frame-zu-Frame-Schätzung (Abschnitt 4).
Das Zurückweisen von Frames (Punkt 2), und damit allen zu ihnen assoziierten Messungen, sollte mit äußerster Vorsicht gehandhabt werden. Meistens reicht der Ausschluss einzelner Messungen, wie in Abschnitt 5.4 beschrieben. Ein Fall, in dem jedoch ganze Frames zurückgewiesen werden müssen, ist, wenn sich das Fahrzeug nur langsam bewegt und der so gemessene optische Fluss klein wird. Ohne ausreichende Parallaxe kann keine Tiefeninformation über die Landmarkenknoten extrahiert werden, da die gemessenen Sichtstrahlen nahezu parallel sind. Somit wird die Translationsschätzung sehr empfindlich gegenüber fehlerhaften Messungen. Daher kann in diesem Fall schon das Sensorrauschen der Kamera zu einer falschen Bewegungsschätzung führen. Die Zurückweisung der Frames anhand des mittleren optischen Flusses im Bild stellt hierbei eine effektive Methode dar, diese Instabilität zu umgehen.
Alle restlichen Frames, welche weder benötigt noch zurückgewiesen sind, werden gesammelt (Punkt 3). Die Integration dieser Messungen in das Optimierungsproblem würde die Bewegungsschätzung ein wenig genauer machen, jedoch auch die Rechenzeit erhöhen. Daher werden aus dieser Menge Frames im Abstand von $0{,}3$ s als Keyframes gewählt.
Die aus diesen drei Schritten ausgewählten Frames werden anschließend zum Optimierungsproblem hinzugefügt. Die Menge von Keyframes ist damit klein, enthält aber eine große Menge an Informationen, um die Stabilität und Genauigkeit des Optimierungsproblem zu gewährleisten.

Schätzung von Posen zwischen Keyframes Werden Keyframes wie in Abschnitt 5.3 ausgewählt, so erhält man eine Posenschätzung für diese. Um ei-

ne Posenschätzung für die nicht als Keyframes ausgewählten Frames zu erhalten, wird eine schnelle Methode benötigt. Hierzu wird der mit zeitlicher Inferenz erstellte Teilgraph G_f fixiert. Anschließend wird die Pose des aktuellen Frames mit seinen Messungen zu G_f registriert, ohne dabei Schätzgrößen von G_f zu verändern. Dies erfolgt in der Regel sehr schnell, wie in Abschnitt 8.4 dargelegt. Diese Methode wird hier mit *Frame-Angleich* bezeichnet, wohingegen die Lösung des Graphen G_f für alle Landmarken und Posen *volle Lösung* genannt wird.

5.3.2. Fensterlänge des Optimierungsproblems

Ein wichtiger Schritt für ein System mit geringem Rechenaufwand ist die Wahl der Länge des gleitenden Fensters. Die Auswahl von Keyframes ist wichtig, da so das Optimierungsfenster trotz einer geringen Anzahl an Keyframes einen längeren Zeitraum umspannen kann und so der Drift reduziert wird. Aktuelle Systeme setzen oft eine fixe Länge des Optimierungsfensters ein. Sind jedoch Kameraknoten nur spärlich durch Messungen verbunden, beruht das Zusammenhängen von G nur auf wenigen Messungen. Dies kann dazu führen, dass Rechenzeit auf die Optimierung von alten Messungen verwendet wird, welche sich nur schwach auf die aktuellste Bewegungsschätzung auswirken. Daher werden solche spärlich verbundenen Knoten identifiziert, um dort das Optimierungsfenster abzuschneiden. Hierzu wird die Konnektivität des Graphen G_f evaluiert, indem die Anzahl der Verbindungen zwischen Landmarkenknoten und Kameraknoten festgestellt wird. Ist die Konnektivität eines Kameraknotens gering, so wird dort das Ende des Optimierungsfensters gesetzt. Um eine zu kurze oder zu lange Fensterlänge zu vermeiden, ist die Länge des Fensters durch harte Schranken an beiden Enden limitiert.

5.4. Landmarkenauswahl

Die Landmarkenauswahl ist eines der meist diskutierten Themen im Bereich der Visuellen Odometrie, da sie eine zentrale Komponente jedes kamerabasierten Algorithmus zur Bewegungsschätzung darstellt. Die optimale Landmarkenmenge sollte die folgenden Bedingungen erfüllen:

- Gut beobachtbar, sodass Landmarken exakt rekonstruiert werden können.
- Möglichst klein, um den Rechenaufwand zu minimieren.
- Frei von Ausreißern.

- Gleichverteilt im Bild und im euklidischen Raum.

In dieser Arbeit wird die Landmarkenauswahl in einem der letzten Schritte umgesetzt, wie in der Übersicht 2.1, Block D gezeigt. Diese Auswahlstrategie beruht darauf, so viele Merkmalsassoziationen wie möglich zu generieren und erst anschließend zu filtern, wenn mehr für die Auswahl nutzbare Informationen extrahiert wurden. Mithilfe der initialen Bewegungsschätzung (Abschnitt 4) können assoziierte Merkmale trianguliert werden, was in einer dreidimensionalen Punktmenge resultiert. Hieraus können die für die Optimierung verwendeten Landmarkenknoten ausgewählt werden.

5.4.1. Landmarkenkategorisierung

Der zentrale Schritt in der Landmarkenauswahl erfolgt durch die Einteilung der Landmarken gemäß ihrer Distanz zum Fahrzeugknoten. Diese werden in drei Gruppen kategorisiert: nah, mittel und fern. Messungen aus jeder dieser Kategorien tragen unterschiedlich zur Bewegungsschätzung bei:

- Nahe Landmarken sind wichtig für die Schätzung der Translation, aber sie sind schwierig zu messen, da ihr optischer Fluss groß ist.
- Mittlere Landmarken tragen sowohl zur Rotationsschätzung als auch zur Translationsschätzung bei. Hier werden diese vor allem verwendet, um aus lokalen Minima zu entkommen.
- Ferne Landmarken tragen vor allem zur Rotationsschätzung bei und sind hierfür besonders wichtig, da sie einfach und daher über einen großen Zeitraum zu verfolgen sind.

Diese Kategorisierung ist in Abbildung 5.4 dargestellt. Nah, mittel und fern ist dabei relativ zur Kamera und deren Parametern zu sehen. Ist die Brennweite klein, so müssen die Grenzen der Kategorien näher gewählt werden. In unseren Experimenten haben sich die Grenzen $0 - 20\ \mathrm{m}$ nah, $20 - 50\ \mathrm{m}$ mittel und $> 50\ \mathrm{m}$ für eine Lochkamera mit Brennweite $f = 718\ \mathrm{Pixel}$ bewährt.

Um möglichst gleichmäßig verteilte Landmarken auszuwählen, wird in einem ersten Schritt ein Filter verwendet, um lokale Häufungen von Landmarken auszudünnen. Hierzu wird der Raum in Voxel unterteilt und daraus jeweils die Landmarke extrahiert, welche am ehesten in der Mitte des Voxels ist. Somit werden zwar gut beobachtbare Strukturen für die Bewegungsschätzung verwendet, dominieren diese aber nicht. Für jede Kategorie wird im Anschluss eine fixe Anzahl an Merkmalen extrahiert, wofür eine spezifische Auswahlstrategie verfolgt wird.

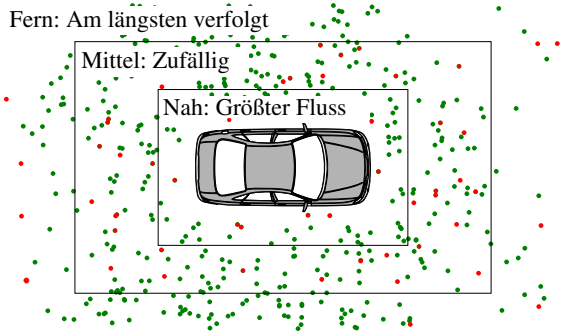


Abbildung 5.4.: Landmarkenkategorisierung in nah, mittel und fern spielt eine zentrale Rolle in der Landmarkenauswahl. Für jede Kategorie werden Landmarken ausgewählt (grün) oder abgelehnt (rot). Im Fernbereich werden Landmarken ausgewählt, welche am längsten verfolgt werden können und im Nahbereich solche, welche den größten Fluss besitzen. Im mittleren Bereich werden Landmarken zufällig gezogen, um aus lokalen Minima entkommen zu können. Der Maßstab der Grenzen ist in Abhängigkeit der Kamerabrennweite zu wählen.

Nahe Landmarken Nahe Landmarken sind in der Regel nur über wenige Frames beobachtbar. Daher werden in dieser Kategorie Merkmalsassoziationen mit möglichst großem optischen Fluss ausgewählt. Aufgrund des dadurch deutlich besseren Signal-Rausch-Verhältnis werden die Landmarken ausgewählt, welche am genauesten aus wenigen Frames rekonstruiert werden können.

Mittlere Landmarken Steckt das Optimierungsproblem in einem lokalen Minimum fest, so konvergieren die Kamera- und Landmarkenknoten zu einer Konfiguration, welche zwar lokal optimal ist, jedoch nicht die echte Umgebungsstruktur und Fahrzeugbewegung abbildet. Die Verwendung von Beobachtungen, welche zu vorangegangenen Zeitpunkten noch nicht Teil des Optimierungsproblems waren, schafft hierbei Abhilfe. Hierzu werden in dieser Kategorie die Hälfte der Landmarken aus schon optimierten Landmarken und die andere Hälfte aus vorher nicht verwendeten Landmarken zufällig ausgewählt.

Ferne Landmarken Landmarken in dieser Kategorie können nicht zur Translationsschätzung beitragen, sind allerdings von hoher Relevanz für die Rotationsschätzung. Um die Anzahl der Landmarkenparameter klein zu halten, aber trotzdem möglichst viele Beobachtungen im Optimierungspro-

blem zu verwenden, werden Landmarken mit maximaler Anzahl an Messungen verwendet.

5.4.2. Auswahl durch semantische Informationen

Diese Arbeit widmet sich auch der Frage, ob semantische Informationen über die Objekte, auf welchen Landmarken liegen, die Qualität der Bewegungsschätzung verbessern können. Hierzu wird untersucht, ob eine geringere Gewichtung von Landmarken auf Vegetation vorteilhaft ist. Intuitiv ist der Einfluss dieser zwiespältig. Einerseits haben Sträucher und Bäume eine reichhaltige photometrische Struktur und sind daher hervorragend zu assoziieren und zu verfolgen. Andererseits sind diese nicht starr — ihre Eigenbewegung stellt eine Verletzung der grundlegenden Annahme einer statischen Umgebung dar. Um diesen Einfluss zu quantifizieren, wird Vegetationslandmarken ein geringeres Gewicht zugewiesen und der Bewegungsfehler ausgewertet. Das Ergebnis ist in Abschnitt 8 kurz beschrieben.
Die semantischen Informationen werden als zusätzliche Ausreißerdetektion verwendet — nur Merkmale auf Objekten, welche zu statischen Klassen wie Infrastruktur, Gebäude, Vegetation, etc. gehören, werden als Messungen aufgenommen. Verworfen werden zum Beispiel Messungen aus den Klassen Fahrzeug, Fußgänger, Himmel, Motorrad. In Szenen mit vielen parkenden Autos gehen so zwar einige Messungen verloren, jedoch sind Messungen auf dahinter liegender Infrastruktur ausreichend für eine gute Schätzung. Im Gegensatz dazu können bewegte Fahrzeuge die Schätzung sehr stark beeinträchtigen, weswegen Messungen auf Fahrzeugen kategorisch verworfen werden.

Plausibilisierung mithilfe der Chiralitätsbedingung Durch Messrauschen und fehlerhafte Assoziation ist es möglich, dass triangulierte Landmarken hinter der Bildebene positioniert werden. Diese können direkt aussortiert werden. Man spricht hierbei von der Verletzung der sogenannten Chiralitätsbedingung, siehe dazu auch Hartley and Zissermann [6].

5.5. Robustifizierung und Problemformulierung

Ausreißer verhindern, dass ein Least-Squares-basiertes Optimierungsproblem zum korrekten Minimum konvergiert, wie von Torr et al. ([39], [7]) beschrieben. Obwohl, wie in Abschnitt 5.4 beschrieben, semantische Informationen und die Chiralitätsbedingung verwendet werden, um Ausreißer zu eliminieren, verbleiben Messungen im Optimierungsproblem, welche

nicht modelliert werden können (z. B. da sie zu einem unabhängig bewegten Objekt gehören, falsch assoziiert sind, ...). Daher wird auch hier eine robustifizierte Loss-Funktion verwendet. So verbleiben Ausreißer zwar im Optimierungsproblem, jedoch wird ihr Gewicht stark reduziert, sodass ihr Einfluss auf das Ergebnis sehr klein wird (siehe dazu Abschnitt 4.2). Die Cauchy-Funktion $\rho_\phi(x)$ wird als Loss-Funktion verwendet. Mit $\mathcal{P}$ und $\mathcal{L}$, den Mengen der gewählten Keyframes und Landmarken, ist das robustifizierte Optimierungsproblem für die Lösung des Graphen G durch

$$\underset{\mathbf{P}_j \in \mathcal{P}, \mathbf{l}_i \in \mathcal{L}, d_i \in \mathcal{D}}{\operatorname{argmin}} \quad \sum_i \sum_j \rho_\phi(\|\phi(\mathbf{l}_i, \mathbf{P}_i)\|_2) \tag{5.2}$$

mit dem Rückprojektionsfehler $\phi(\mathbf{l}_i, \mathbf{P}_j) = \bar{\mathbf{l}}_{i,j} - \pi(\mathbf{l}_i, \mathbf{P}_j)$ gegeben.
Zwar ist der Einfluss der Ausreißer gering, jedoch verbleiben diese im Optimierungsproblem und die korrespondierenden Residuen werden in jeder Iteration neu evaluiert. Besonders für rechenintensive Anwendungen, wie das hier vorliegende SLAM-Problem, wird daher Rechenzeit auf Messungen verwendet, welche nicht zur Lösung des Problems beitragen. Daher wird in dieser Arbeit eine Variante der Trimmed-Least-Squares-Methode verwendet, welche die zu Ausreißern gehörigen Residuen und Parameter iterativ entfernt, wie in Algorithmus 2 beschrieben. Dieses Vorgehen konnte bei gleicher Genauigkeit die Rechenzeit halbieren.

Data : Optimierungsproblem p;
Schrittanzahl n;
Schwellwert r;
Kostenterme $\mathcal{E}$;
Result : Optimierungsproblem;
foreach $s \in n$ **do**
    Mache s Iterationen von p;
    foreach $\epsilon \in \mathcal{E}$ **do**
        Entferne die $r\%$ größten Residuen nach $\epsilon(p)$ aus p;
    end
    Entferne alle Parameter ohne Residuen von p;
end
Optimiere p bis Konvergenz;

Algorithmus 2 : Variante der Trimmed-Least-Squares-Methode für SLAM. Anfangs werden einige wenige Iterationen des Optimierungsproblems ausgeführt und die größten Residuen aussortiert. Somit wird die Geschwindigkeit der finalen Optimierung bis zur Konvergenz stark erhöht. Hierbei wird das Trimmen für alle Kostentermarten separat durchgeführt. Eine Kostentermart ist hierbei zum Beispiel der Rückprojektionsfehler oder der Tiefenfehler, wie in Abschnitt 6.2.1 beschrieben.

Kapitel 6

Skalenschätzung

Die Verwendung eines monokularen Kamerasystems hat viele Vorteile: Es ist kostengünstig und einfach zu warten, flexibel montierbar und klein. Ein solches System hat jedoch einen Nachteil: Ein Parameter, die sogenannte *Skale*, ist nicht beobachtbar. Dies ist ersichtlich anhand der Projektionsgleichung 3.1 in Abschnitt 3.1. Da durch diese Projektion auf den Kamerasensor die Tiefenkomponente entfällt, kann die Skale nicht beobachtet werden. Hierdurch können zwar die Größenverhältnisse der Umgebungsstruktur wahrgenommen werden, um allerdings deren echte Größe und somit auch die zurückgelegte Strecke in der Welt zu kennen, muss die Skale bekannt sein. Hierzu wurden im Rahmen dieser Arbeit mehrere Methoden entwickelt, um die Skale aus unterschiedlichen Informationsquellen zu schätzen und in das Gesamtsystem einzugliedern (siehe Schaubild 2.1 in Kapitel 2, Block S).
Zuerst wird dafür in Abschnitt 6.1 die Skalenschätzung anhand geometrischer Informationen über die Umwelt untersucht. In Abschnitt 6.2 wird ein weiterer Sensor, ein LIDAR, genutzt, um Skaleninformation zu erhalten. Zusätzlich wird in Abschnitt 6.3 theoretisch gezeigt, wie die Skale monokular mithilfe eines Bewegungsmodells in Kurven geschätzt werden kann.

6.1. Skalenschätzung anhand geometrischer Informationen

Eine Informationsquelle, aus welcher die Skale geschätzt werden kann, ist das Vorwissen über die Geometrie der Umgebung. Hierbei können zum Beispiel Größen von Straßenschildern, Höhen von Autos und vieles mehr

genutzt werden. Ein besonders wertvolles Merkmal ist hierbei der Abstand zwischen Kamera und Bodenoberfläche. Im Gegensatz zu Objekten in der Umgebung kann diese nämlich durch den Systemaufbau kontrolliert werden. Zudem ist durch den Mindestabstand zwischen Fahrzeugen in den meisten Szenarien Bodenoberfläche zu sehen. Ist die Einbauhöhe über Grund bekannt, so kann die Bodenoberfläche rekonstruiert und somit die Skale für das Problem bestimmt werden. Im Rahmen dieser Arbeit wurden dafür drei Methoden entworfen und ausgewertet, welche im Folgenden dargestellt sind. Angefangen bei einer Frame-zu-Frame-Bodenoberflächenschätzung [40] wird übergeleitet zu einem A-Posteriori-Schätzer der Bodenoberfläche für Methoden mit zeitlicher Inferenz hin zu einer in das Optimierungsproblem integrierten Bodenoberflächenschätzung. Hierbei wird der Boden als Ebene durch die Gleichung

$$\mathbf{n}^T \mathbf{p} + d = 0 \tag{6.1}$$

modelliert, wobei $\mathbf{n} \in \mathbb{R}^3$ den Normalenvektor und $d \in \mathbb{R}$ den Abstand der Ebene zum Ursprung bezeichnen. Der dreidimensionale Punkt $\mathbf{p} \in \mathbb{R}^3$ liegt auf dieser Ebene. Die restriktive Ebenenannahme wird in der integrierten Bodenoberflächenschätzung abgeschwächt, indem der Boden durch mehrere lokale Ebenen modelliert wird und somit eine stückweise lineare Bodenoberflächenapproximation möglich wird.

6.1.1. Bodenebenenschätzung für Frame-zu-Frame Visuelle Odometrie mit Fluchtpunkten

Um die Skale für Frame-zu-Frame-basierte Schätzmethoden aus der Bodenebene bereitstellen zu können, muss diese aus nur zwei aufeinanderfolgenden Frames geschätzt werden. Die Rekonstruktion der Bodenpunkte ist daher schwierig, weil aus nur zwei Messungen deren dreidimensionale Position geschätzt werden muss. Da Punkte auf dem Boden schwierig zu verfolgen sind, ist deren Genauigkeit gering und die resultierende Punktmenge unterliegt starkem Rauschen. Aufgrund dessen muss die Bodenebene mithilfe zusätzlicher Information stabilisiert werden. Diesbezüglich wird in dieser Arbeit ein Verfahren vorgestellt, um mithilfe der im Kamerabild abgebildeten Fluchtpunkte die Richtung der Bodenebene separat zu schätzen und somit die Bodenebenenrekonstruktion zu stützen.
Der prinzipielle Ablauf dieser Methode ist in Abbildung 6.1 zu sehen. Die unskalierte Bewegung zwischen den Bildern wird geschätzt und durch Triangulation der Sichtstrahlen kann eine dreidimensionale Punktmenge aus den verfolgten Merkmalspunkten erstellt werden. Hierzu können die in Abschnitt 4 erarbeiteten Methoden verwendet werden. Da die triangulierte

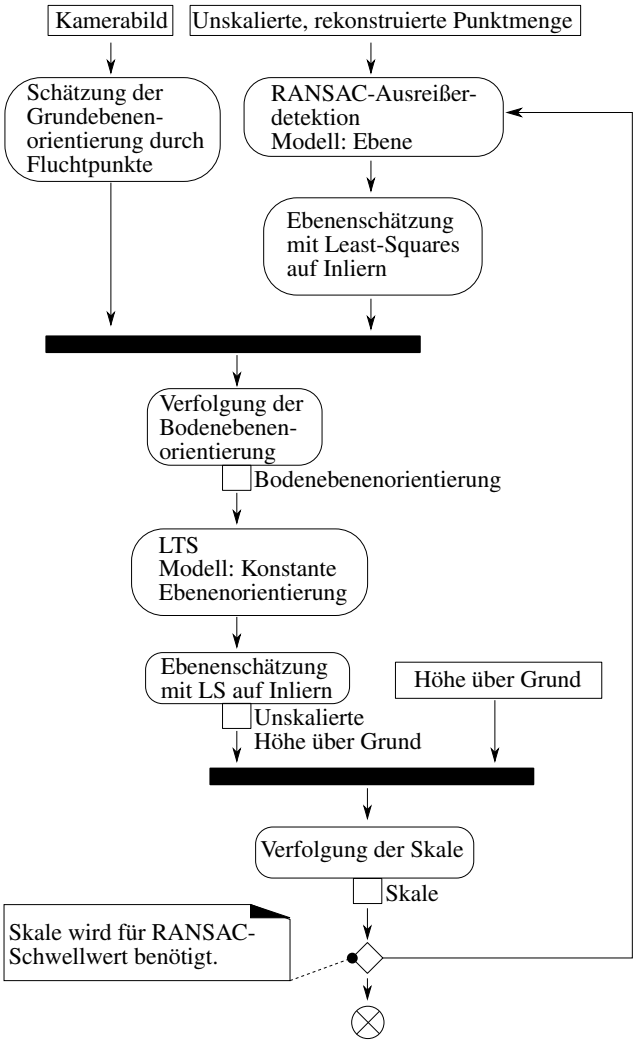


Abbildung 6.1.: Aktivitätsdiagramm der Bodenebenenschätzung für Frame-zu-Frame Visuelle Odometrie. Die ungenaue Bodenebenenrekonstruktion aus nur zwei Bildern wird mithilfe von Fluchtpunkten gestützt.

Punktmenge stark verrauscht ist und typischerweise viele Ausreißer beinhaltet, wird mit RANSAC (siehe Abschnitt 3.4) das plausibelste Grundebenenmodell gefunden. Anschließend werden die Inlier des besten Ebenenkandidats verwendet, um mit der Methode der kleinsten Fehlerquadrate die Grundebene zu schätzen. Hierzu wird das Optimierungsproblem

$$\underset{\mathbf{n},d}{\operatorname{argmin}} \sum_{i} \left(\mathbf{n}^T \mathbf{p}_i + d\right)^2 \tag{6.2}$$

gelöst. Dieses Problem ist linear und daher sehr effizient lösbar. Die damit geschätzte Distanz der Ebene wird zuerst missachtet und nur die geschätzte Orientierung weiter verwendet.
Parallel hierzu wird die Grundebenenorientierung mithilfe der Fluchtpunkte des Kamerabilds extrahiert. Diese werden durch den von Schwarze et al. [41] vorgestellten Algorithmus geschätzt. Unter der Annahme, dass die Strukturen, welche einen vertikalen Fluchtpunkt erzeugen, senkrecht auf dem Boden stehen, wie zum Beispiel Häuser, Pfähle, Leitplanken etc. entspricht die vertikale Fluchtpunktrichtung dem Normalenvektor der Bodenebene. Diese Annahme trifft für die meisten innerstädtischen Szenarien in ausreichender Näherung zu. In diesen Szenarien ist zudem die Wahrscheinlichkeit, dass die Bodenebene verdeckt ist, deutlich höher als außerhalb bebauter Gebiete. Die Fluchtpunktschätzung liefert hier also wertvolle komplementäre Informationen zur Bodenebenenrekonstruktion. In außerstädtischen Szenarien ist zwar der vertikale Fluchtpunkt schwieriger zu schätzen, jedoch ist hier wiederum die Bodenebenenrekonstruktion einfacher, da der Boden typischerweise besser beobachtbar ist. Um die Vorteile beider Seiten zu nutzen, werden die Bodenebenenorientierungen anschließend mit einem Kalman-Filter [42] fusioniert. Einzelheiten hierzu sind in Abschnitt 6.1.1.1 erläutert.
Die fusionierte Bodenebenenorientierung wird abschließend verwendet, um die Distanz der Bodenebene zur Kamera d zu schätzen. Hierzu wird die Orientierung der Ebene fixiert und mithilfe der Methode der kleinsten getrimmten Fehlerquadrate (LTS, siehe Abschnitt 3.3.1) wird d geschätzt. Durch das Vorwissen der Kamerahöhe über Grund h kann nun die Skale

$$s = \frac{h}{d} \tag{6.3}$$

berechnet werden. Zusätzlich wird s durch einen weiteren Kalman-Filter geglättet und plausibilisiert. Dies kann kurze Verdeckungen der Grundebene überbrücken. Die geschätzte Skale muss in die Ausreißerdetektion

durch RANSAC zurückgeführt werden, um den Inlier-Schwellwert für die nächste Schätzung festzulegen.

6.1.1.1. Orientierungsschätzung der Bodenebene

Um die Orientierungsschätzung der Bodenebene durchzuführen, werden zwei komplementäre Methoden fusioniert:

1. Ebenenschätzung mit rekonstruierten dreidimensionalen Merkmalspunkten.

2. Extraktion der Bodenebenennormale durch Fluchtpunkte.

Eine RANSAC-basierte Ausreißererkennung liefert gefolgt von einem LS-Schätzer die Parameter der Bodenebene. Der Inlier-Schwellwert für RANSAC liefert wiederum die vorangegangene Skalenschätzung. Damit wird angenommen, dass die Skale zwischen zwei Schritten nur leichten Veränderungen unterliegt. Für Fahrzeuge ist dies gegeben. Da der Translationsvektor auf eine konstante Länge normiert wird, hängt da die Skale nur von der Fahrzeuggeschwindigkeit ab. Die auf dem Fahrzeug mögliche Beschleunigung verursacht Skalenveränderung, welche zwischen zwei Zeitschritten als klein anzunehmen ist.
Um die Fluchtpunkte der Szene zu schätzen, werden Linien aus dem Kamerabild extrahiert und Fluchtpunkthypothesen zugeordnet. Anschließend wird die Fluchtpunktrichtung verbessert, indem der Winkelfehler zwischen den Linien und der Fluchtpunktrichtung minimiert wird. Aufgrund der neuen Fluchtpunkthypothese werden neue Linien aus dem Kamerabild zugeordnet und der Vorgang wiederholt, bis Konvergenz eintritt. Für Einzelheiten sei auf Schwarze et al. verwiesen [41].
Diese Methode ist abhängig von einer guten Fluchtpunktinitialisierung. Zu Anfang wird hierzu eine aus der Rekonstruktion geschätzte Ebene verwendet. Anschließend werden die geschätzten Fluchtpunktrichtungen mithilfe der Rotation der geschätzten Eigenbewegung in darauf folgende Bilder projiziert. Ein Beispiel mit drei geschätzten Fluchtpunktrichtungen und assoziierten Linien ist in Abbildung 6.2 dargestellt.
Für die Fusion der geschätzten Bodenebenenorientierung werden die beiden Winkel der Bodenebenen mithilfe von Kalman-Filtern unabhängig voneinander verfolgt. Da nur kleine Winkelveränderungen vorliegen, hat sich hierbei ein simples Systemübergangsmodell mit konstantem Winkel bewährt. Die Systemübergangs- und Messgleichungen formulieren sich wie folgt:

$$\mathbf{x}_{k+1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x}_k + \mathbf{w}_k, \tag{6.4}$$

Abbildung 6.2.: Kamerabild mit Linien aus Grauwertkanten, welche Fluchtpunkten zugeordnet sind. Grüne Linien sind dem in Grün gezeigten Fluchtpunkt zugeordnet, welcher auf dem Horizont liegt. Rote Linien sind dem vertikalen und blaue Linien dem horizontalen Fluchtpunkt zugeordnet, welche außerhalb des Bildes liegen. Der vertikale Fluchtpunkt beschreibt die Bodenebenennormale.

mit $\mathbf{x}_k$ als den zwei Winkeln der Ebene zum Zeitpunkt k mit normalverteiltem, mittelwertfreiem Rauschen $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ und

$$\mathbf{y}_k = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x}_k + \mathbf{v}_k, \tag{6.5}$$

mit $\mathbf{y}_k$ als den Schätzwerten der zwei Ebenenwinkel zum Zeitpunkt k und normalverteiltem, mittelwertfreiem Rauschen $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. Messwerte, welche von der Schätzung zu weit abweichen, werden zurückgewiesen. Im Sinne von normalverteilten Winkeln wird hierzu die Mahalanobis-Distanz [8] gewählt.

6.1.1.2. Verfeinerung der Ebenenschätzung

Die in Abschnitt 6.1.1.1 geschätzte Bodenebenenorientierung wird verwendet, um die Distanz d der Grundebene durch die rekonstruierten dreidimensionalen Merkmalspunkte zu schätzen. Eine auf RANSAC basierende Ausreißerbehandlung benötigt einen Schwellwert für die Identifikation von Inliern. Dieser ist jedoch abhängig von der Skale — der Raum der rekonstruierten Punkte wird so mit sich verändernder Fahrzeuggeschwindigkeit komprimiert oder auseinandergezogen. Um diese Rückkopplung zu verringern, wird die in Abschnitt 3.3.1 vorgestellte LTS-Methode herangezogen. Als untere Grenze wird das 40%-Perzentil und für die obere Grenze das 90%-Perzentil verwendet. Projizierte Bodenebenenpunkte nach der Ausreißerdetektion sind Abbildung 6.3 zu entnehmen.

Im letzten Schritt wird die Distanz zur Bodenebene und ihr Drift durch

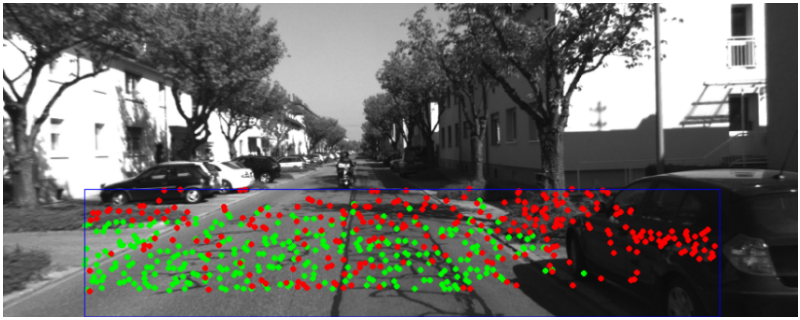


Abbildung 6.3.: Rekonstruierte Punkte aus zwei aufeinanderfolgenden Bildern nach Ausreißerdetektion. Zur Bodenebene gehörende Inlier sind grün, Ausreißer sind rot dargestellt.

einen Kalman-Filter geschätzt, welcher durch die Systemübergangsgleichung

$$\mathbf{s}_{k+1} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \mathbf{s}_k + \mathbf{w}_k \tag{6.6}$$

mit $\mathbf{s}_k = (s, \dot{s})^T$ als Skale und Skalendrift-Geschwindigkeit zum Zeitpunkt k gegeben ist. Das Rauschen ist als $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ bezeichnet. Der Zeitunterschied zwischen diesen Zuständen wird mit Δt bezeichnet. Es wird also konstanter Skalendrift angenommen, Veränderungen werden durch die Unsicherheit des Zustands modelliert. Die Integration der gemessenen Skale wird durch die Messgleichung

$$u_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{s}_k + \mathbf{v}_k \tag{6.7}$$

ausgedrückt, wobei u_k die Schätzung der Skale zum Zeitpunkt k und $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ das Rauschen beschreibt.

6.1.2. Bodenebenenschätzung für Visuelle Odometrie mit zeitlicher Inferenz

Wird die in Kapitel 5 dargestellte Methodik zur Bewegungsschätzung mit zeitlicher Inferenz genutzt, weist die Umgebungsrekonstruktion eine deutlich höhere Genauigkeit auf, als wenn nur aus zwei Zeitpunkten rekonstruiert wird. Dies kann insbesondere durch die höhere Anzahl an Messungen und das stärkere Signal des optischen Flusses erklärt werden. Die Stützung der Orientierung durch Fluchtpunkte ist hier im Allgemeinen nicht mehr

nötig. In dieser Arbeit werden daher zwei Methoden der Integration der Bodenebene zur Skalengewinnung untersucht:

1. A-Posteriori-Skalenschätzung mithilfe von LTS Ebenenschätzung.
2. In das Optimierungsproblem integrierte Skalenschätzung.

6.1.2.1. A-Posteriori-Ebenenschätzung

Möchte man ein möglichst modulares System konstruieren, so bietet es sich an, die aus der Bewegungsschätzung erhaltene Umgebungsrekonstruktion als Eingang für die Skalenschätzung zu verwenden. Somit haben die Annahmen, welche für die Skalenschätzung getroffen werden, keine Auswirkungen auf die Bewegungsschätzung und die Methodik kann als eigenständiges, abgeschlossenes Modul einfach ausgetauscht werden.
Um die Skale durch die Annahme konstanter Einbauhöhe schätzen zu können, muss die Bodenoberfläche klassifiziert und extrahiert werden. Hierzu wird ein LTS-Ansatz (siehe Abschnitt 3.3.1) verwendet. Die Bodenoberfläche wird als Ebene modelliert und mit Gleichung 6.2 berechnet. Da die Schätzung im Betrieb auf dem Versuchsfahrzeug angewandt werden soll, wird hier nicht die gesamte bekannte Rekonstruktion verwendet, sondern nur der Teil der Umgebung, welcher sich im aktuellen Optimierungsfenster befindet. Somit wird die Annäherung der Bodenoberfläche als Ebene nur in diesem Bereich angewandt, welcher je nach Fahrzeuggeschwindigkeit typischerweise $10\ \mathrm{m}$ bis $30\ \mathrm{m}$ umfasst. Durch die Einbauposition kann der LTS sehr gut initialisiert werden. Eine zusätzliche Verfolgung der Skale, wie in Abschnitt 6.1.1 vorgestellt, glättet das Ergebnis. Fehlschätzungen der Ebene, zum Beispiel durch Verdeckung, können dadurch überbrückt werden. Der größte Nachteil dieser Methode ist ihre Abhängigkeit von einer guten Rekonstruktion der Bodenebene. Ist diese nicht ausreichend beobachtbar, wie zum Beispiel im Stau, wenn Fahrzeuge die Bodenebene größtenteils verdecken, sind nicht genug Punkte vorhanden, um diese hinreichend genau zu schätzen. Dieser Nachteil kann durch die Integration der Bodenebenenannahme in das in Abschnitt 5 vorgestellte Optimierungsproblem besser gehandhabt werden.

6.1.2.2. Integration lokaler Bodenebenen in das Optimierungsproblem

Wird die Bodenoberfläche mit der in Abschnitt 6.1.2.1 beschriebenen A-Posteriori-Methode geschätzt, hat das zwei wesentliche Nachteile:

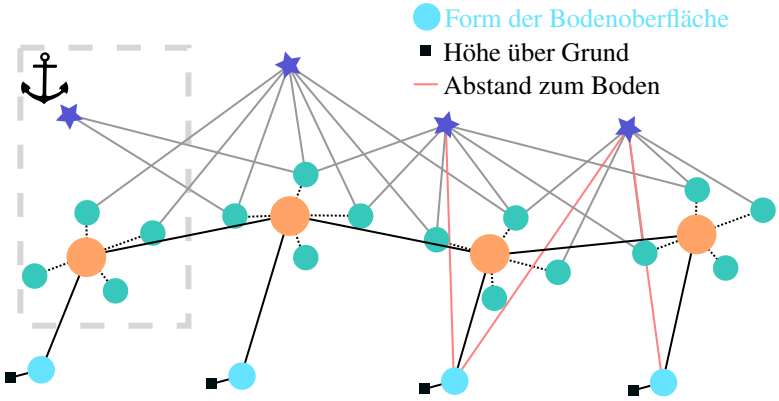


Abbildung 6.4.: Graph für Visuelle Odometrie mit integrierter Bodenebene. Für jeden Fahrzeugknoten existiert ein Knoten mit Formparametern der Bodenoberfläche (türkis). In diesem Fall sind die Formparameter die zwei Winkel der Normalen der lokalen Bodenebene, allerdings ist dieses Konzept auch auf komplexere Modelle mit mehr Freiheitsgraden anwendbar. Die vorher bekannte Höhe über Grund wird als Faktor dargestellt (schwarze Quadrate).

1. Es wird eine Ebene für das gesamte Fenster des Bündelausgleichs verwendet.

2. Die Schätzung der Bodenoberfläche enthält wertvolle Informationen, welche so nicht für die Posen- und Landmarkenschätzung verwendet werden können.

Um diese Nachteile zu vermeiden, wird in dieser Arbeit untersucht, wie die Integration der Bodenoberflächenschätzung in das Optimierungsproblem der Bewegungsschätzung (Gleichung 5.2) die Trajektorienqualität verändert. Hierzu werden die Fahrzeugknoten, wie in Abbildung 6.4 gezeigt, um zwei zusätzliche Parameter, die Richtungen der Bodenebenennormale, erweitert[1]. Diese werden als dreidimensionaler Vektor modelliert und mithilfe einer lokalen Parametrisierung auf die Einheitskugel abgebildet. Damit wird die Oberfläche nicht mehr als eine Ebene, sondern als viele stückweise definierte Ebenen approximiert. Somit wird der erste oben genannte Nachteil insofern abgeschwächt, als dass angenommen wird, dass nur in direkter Nähe um die Pose der Boden eben ist. Die Höhe über Grund muss wie zuvor als konstant angenommen werden und wird als Faktor dem Gra-

[1] Dieses Konzept kann mit der in Abschnitt 6.2 gezeigten Skalenschätzung mit LIDAR kombiniert werden. In diesem Fall hat jeder Bodenoberflächenknoten zusätzlich die Höhe über Grund als Parameter.

phen hinzugefügt. Um die Ebenenstücke zu schätzen und die Skale in das Problem zu integrieren, werden für jede Landmarke auf dem Boden zusätzliche Regularisierungsterme $\alpha_{k,j}$ zum Optimierungsproblem hinzugefügt. Die Regularisierungsterme werden durch

$$\alpha_{k,j} = \left(1 - \frac{\min\left(a, \|\mathbf{P}_k \mathbf{l}_j\|_2^2\right)}{a}\right) \left(\mathbf{n}_k^T (\mathbf{P}_k \mathbf{l}_j) + h\right) \tag{6.8}$$

ausgedrückt, wobei $\mathbf{l}_j$ die Landmarke benennt, welche der Pose des Fahrzeugknotens $\mathbf{P}_k$ am nächsten ist, und $\mathbf{n}_k$ die Normale der zu $\mathbf{P}_k$ gehörigen Ebene bezeichnet. Die Höhe h ist konstant und eingemessen. Somit wird der Abstand von $\mathbf{l}_j$ zur Ebene bestraft und die Skale in das Problem injiziert. Der Gewichtungsfaktor $1 - \frac{\min\left(a, \|\mathbf{P}_k \mathbf{l}_j\|_2^2\right)}{a}$ verringert den Einfluss von Landmarken in großer Distanz, welche die Ebenenannahme verletzen. Der Parameter a gibt hierbei an, ab welcher Distanz Punkte keine Auswirkung mehr auf die Bodenebenenschätzung haben.
Zusätzlich dazu können durch den zusätzlichen Kostenterm

$$\beta_k = \mathbf{n}_k^T \cdot \text{Translation}(\mathbf{P}_k^{-1} \mathbf{P}_{k+1}) \tag{6.9}$$

Abweichungen aufeinanderfolgender Posen von der Ebene bestraft werden, um die Translation des Fahrzeugs auf die lokalen Ebenen zu zwingen. Auch eine Bestrafung von Abweichungen der Normalen durch den Kostenterm

$$\gamma_k = 1 - \mathbf{n}_k^T \mathbf{n}_{k+1} \tag{6.10}$$

ist sinnvoll, um die Übergänge der lokalen Ebenen zu glätten, insbesondere falls nur wenige Bodenpunkte vorhanden sind. Die Klassifikation der Landmarken als Boden wird durch maschinell gelernte semantische Segmentierung realisiert, welche in Abschnitt A.3 umrissen wird.

6.2. Skalenschätzung mit Hilfe eines LIDAR

Die wohl kostspieligste, jedoch genaueste Variante der Skalenschätzung liegt in der Verwendung zusätzlicher Sensorik. Hierbei müssen nach der intrinsischen Kalibrierung der Kameras die zusätzlichen Sensoren extrinsisch zu den Kameras kalibriert werden, also die Transformation zwischen ihren Koordinatensystemen geschätzt werden. Da jedoch moderne autonome Systeme meist verschiedenste Sensorik aufweisen, ist die Nutzung komplementärer Messsysteme sehr attraktiv und birgt großes Potential.

6.2.1. Vorgehen

Der LIDAR-Sensor ist ein sehr gut geeignetes Komplement zur Kamera. Obwohl dieser eine deutlich geringere Auflösung aufweist als die Kamera, ist die Tiefenschätzung hochgenau und die Reichweite groß. Zudem ist die Genauigkeit der Tiefenschätzung auch in großer Entfernung noch sehr gut. Das bedeutendste Problem dieses Sensors ist seine geringe Auflösung. Um die Tiefeninformation für die Kamera nutzbar zu machen, müssen daher die Messungen zur Kamera assoziiert und gegebenenfalls interpoliert werden [43].
Im ersten Schritt werden zu jeder Kameramessung benachbarte LIDAR-Punkte extrahiert. Dafür wird die extrinsische Kalibrierung von LIDAR zu Kamera (siehe Abschnitt 7) und die intrinsische Kalibrierung der Kamera genutzt, um LIDAR-Messungen, mithilfe der Projektionsgleichung 3.1, auf die Kamera-Bildebene zu projizieren. Zudem werden, wie in Kapitel 3.5 beschrieben, Merkmale f aus den Kamerabildern extrahiert und über die Zeit verfolgt. Für jedes Merkmal werden die folgenden Schritte angewandt:

1. Wähle eine Menge $\mathcal{F}$ von projizierten LIDAR-Messungen in einer *Region Of Interest (ROI)* um jedes Merkmal f, wie in Abschnitt 6.2.2 beschrieben.

2. Segmentiere aus $\mathcal{F}$ die Messungen im Vordergrund $\mathcal{F}_{\text{seg}}$. Hierzu wird die in Abschnitt 6.2.3 beschriebene Methode verwendet.

3. Schätze eine Ebene p unter Verwendung der Elemente aus $\mathcal{F}_{\text{seg}}$, wie in Abschnitt 6.2.4 dargestellt. Falls f zur Bodenebene gehört, erhöht eine spezialisierte Parametrisierung und eine zusätzliche Regularisierung (Abschnitt 6.2.5) die Genauigkeit der Tiefenschätzung.

4. Berechne den Schnittpunkt von p mit dem zu f korrespondierenden Sichtstrahl, um die Tiefe von f zu erhalten.

5. Teste die Gültigkeit der Tiefenschätzung. Der Gültigkeitstest detektiert Merkmale mit schwierig zu schätzender Tiefe und weist diese zurück. Ein Kriterium hierfür ist der Winkel zwischen dem Sichtstrahl von f und der Normale von p. Ist dieser zu klein, können schon kleine Abweichungen zu großen Fehlern in der Tiefe führen. Aufgrund der projektiven Eigenschaften der Kamera wird außerdem die lokale Ebenenannahme stärker verletzt, je größer die Distanz ist. Daher werden Merkmale mit geschätzter Tiefe von mehr als 30 m pauschal zurückgewiesen.

Dieses Vorgehen ist in Abbildung 6.5 graphisch dargestellt.

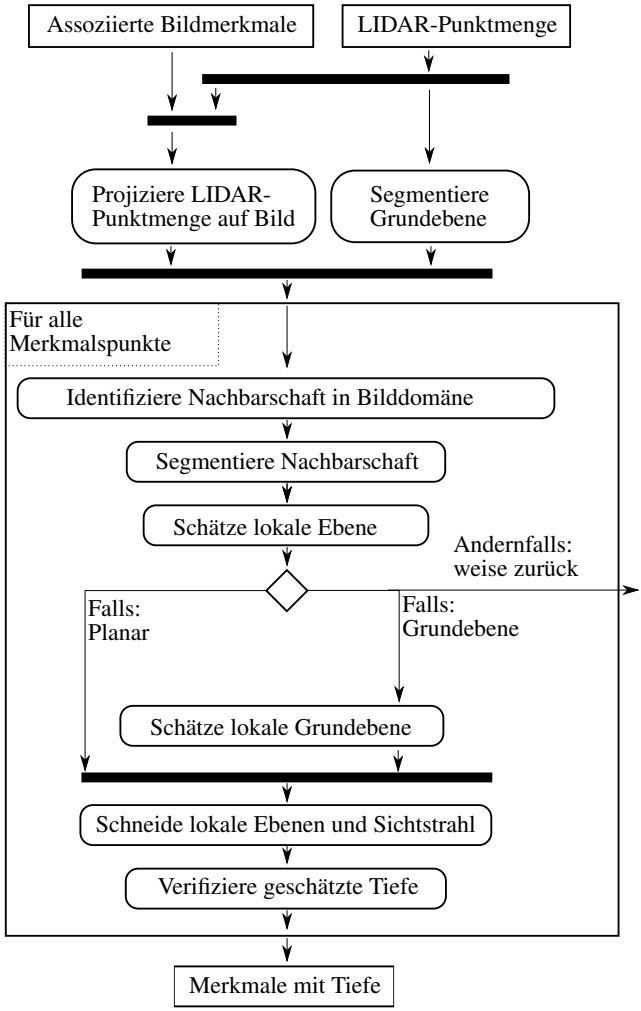


Abbildung 6.5.: Aktivitätsdiagramm zur Tiefenschätzung aus LIDAR für monokulare Merkmalsassoziationen. Es werden nur Tiefenschätzungen ausgegeben, wenn diese mit hoher Präzision berechnet werden können. Somit erzeugt die Methodik wenige, aber dafür sehr genaue Merkmale mit Tiefenschätzung. Durch die Nutzung von zeitlicher Inferenz reichen diese relativ wenigen Merkmale jedoch aus, um die Skale präzise zu schätzen. Die Tiefenschätzung auf der Bodenebene wird gesondert behandelt, da sie speziellen Bedingungen unterliegt, wie in Abschnitt 6.2.4 beschrieben.

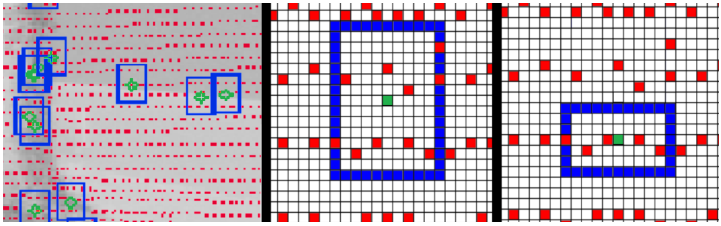


Abbildung 6.6.: Die Auswahl einer adäquaten Region-Of-Interest (ROI, blau) zur Bestimmung benachbarter, projizierter LIDAR-Punkte (rot) um ein detektiertes Merkmal (grün) ist wichtig für die präzise Tiefenschätzung. Damit die lokale Ebene stabil geschätzt werden kann, muss die ROI mehrere Zeilen des LIDAR erfassen (Mitte) und nicht nur Punkte auf einer einzelnen Zeile (rechts).

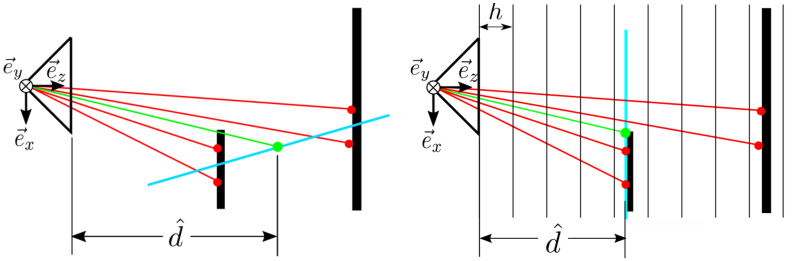
6.2.2. Auswahl benachbarter Messungen

Um eine lokale Ebene um f schätzen zu können, werden zuerst benachbarte Tiefenmessungen ausgewählt. Hierzu werden die LIDAR-Punkte auf die Bildebene der Kamera projiziert und anschließend in einem Rechteck auf der Bildebene ausgewählt. Die Rechteckgröße muss hierbei gut gewählt werden, um Singularitäten in der Ebenenschätzung zu vermeiden, wie in Abbildung 6.6 dargestellt. Dieser Schritt ist insbesondere wichtig, falls die Punktmenge ungeordnet ist.

6.2.3. Segmentierung des Vordergrunds

In dem Fall, dass der Merkmalspunkt f auf einer ebenen Oberfläche wie zum Beispiel einer Fassade liegt, ist die Modellierung der Oberfläche durch eine lokale Ebene sehr präzise. Somit kann die Tiefe von f sehr gut geschätzt werden. Im Regelfall sind Merkmalspunkte jedoch häufig an Kanten und Ecken zu finden, da diese oft hohe Grauwertgradienten erzeugen und somit sehr gut zu verfolgen sind (siehe hierzu Shi et al. [44]). Punkte der Nachbarschaft lägen somit auf mehreren Ebenen, wodurch die Schätzung der lokalen Ebenen fehlerhaft wäre, wie in Abbildung 6.7a dargestellt.

Um dem entgegenzuwirken, wird vor Ausführung der Ebenenschätzung der Vordergrund $\mathcal{F}_{\text{seg}}$ segmentiert, wie in Abbildung 6.7b gezeigt ist. Elemente aus $\mathcal{F}$ werden hierzu in Abhängigkeit ihrer Tiefe in ein Histogramm eingeordnet. Hierbei wird eine fixe Klassenweite von h genutzt. Eine Lücke im Histogramm deutet auf einen Tiefensprung hin. Somit kann effizient der Vordergrund segmentiert werden, indem die nächste signifikante Klassengruppe isoliert wird, wie in Abbildung 6.8 visualisiert.



(a) Punkte im Hintergrund verdrehen die geschätzte lokale Ebene und erzeugen somit eine falsche Tiefenschätzung.

(b) Für die Schätzung der lokalen Ebene werden ausschließlich Punkte im Vordergrund verwendet. Daraus folgt eine akkurate Tiefenschätzung.

Abbildung 6.7.: Tiefenschätzung $\hat{d}$ ohne (a) und mit (b) Vordergrund-Segmentierung mithilfe eines Tiefen-Histogramms mit Klassenweite h. Die geschätzte lokale Ebene ist hellblau dargestellt.

6.2.4. Lokale Ebenenschätzung

Aus der Menge $\mathcal{F}_{\text{seg}}$ werden drei Punkte gewählt, sodass das Dreieck Δ, welches durch sie aufgespannt wird, die größtmögliche Fläche hat. Die lokale Ebene p wird anschließend mithilfe dieser Punkte durch Gleichung 6.2 bestimmt. Damit wird die Schätzung robust gegenüber ungenauen Messungen. Ist die Fläche von Δ zu klein, wird keine Tiefe für f geschätzt. Auf eine abschließende Schätzung von p mit mehr Punkten aus $\mathcal{F}_{\text{seg}}$, die nahe zu p liegen, wird verzichtet, da dies in Experimenten die Rechenzeit erhöht, ohne die Genauigkeit merklich zu verbessern.

6.2.5. Spezialfall: Punkte auf der Bodenebene

Die geringe horizontale Auflösung des LIDAR ist unproblematisch, wenn seine Strahlen ungefähr parallel zur Normalen der zu schätzenden Ebene sind. Die Bodenebene und die Strahlen des LIDAR bilden jedoch in der Regel einen spitzen Winkel. Somit ist die Abweichung zwischen Histogramm-Klassen groß (Abbildung 6.9) und die in Abschnitt 6.2.3 vorgestellte Tiefenschätzung nicht mehr effektiv. Insbesondere auf Landstraßen und auf Autobahnen spielen Merkmale auf dem Boden eine große Rolle, da diese oft die einzigen sind, welche in hinreichender Distanz beobachtet werden können. Darum wird für Merkmale auf dem Boden die Methode anders konditioniert, um ihre Genauigkeit zu erhöhen.

Zuerst wird die Grundebene p_{Grund} in der gesamten LIDAR-Punktmenge

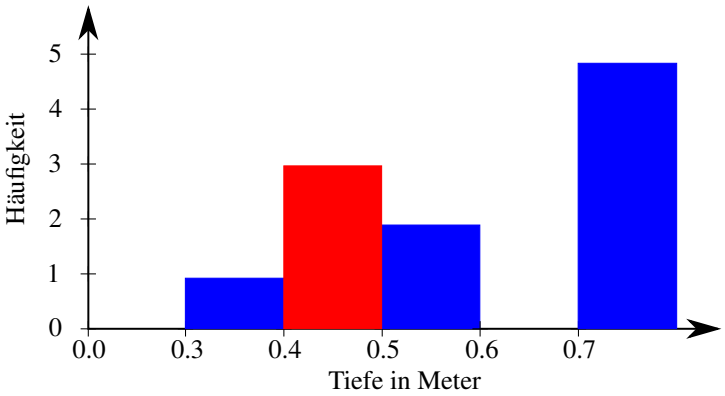


Abbildung 6.8.: Alle Punkte in der ROI werden in einem Tiefen-Histogramm akkumuliert. Durch die Gruppierung der Histogramm-Klassen kann der Vordergrund effizient segmentiert werden und eine Gruppe zur Tiefenberechnung gewählt werden (rot). In diesem Beispiel wird also nur mit Punkten in $0{,}4{-}0{,}5\mathrm{m}$ Tiefe die lokale Ebene geschätzt.

durch den RANSAC-Algorithmus geschätzt. Dann werden alle LIDAR-Messungen $\mathcal{G}$ nahe zur Grundebene ausgewählt. Nun werden, wie in Abschnitt 6.2.3 beschrieben, zu jeder Messung aus dem Kamerabild benachbarte LIDAR-Punkte aus $\mathcal{G}$ identifiziert und eine lokale Ebene p_{Lokal} geschätzt. Da p_{Lokal} in der Regel nur wenig von p_{Grund} abweicht, führt hier eine Regularisierung der Normalenrichtung von p_{Lokal} durch p_{Grund} zu einer deutlich zuverlässigeren Tiefenschätzung. Zusätzlich ermöglicht eine auf die Grundebene spezialisierte Parametrisierung des Algorithmus die genaue Schätzung von p_{Lokal}. Insbesondere wird eine größere ROI für die Nachbarschaftsauswahl sowie eine größere minimale Fläche von Δ angenommen.

6.2.6. Integration der Tiefenmessungen in den Graphen

Um die geschätzten Tiefen der Bildmerkmale für die Skalenschätzung zu verwenden, werden diese in das Optimierungsproblem der Bewegungsschätzung mit zeitlicher Inferenz (Gleichung 5.2) integriert. Dazu werden zusätzliche Residuen

$$\xi(\mathbf{l}_i, \mathbf{P}_j) = \begin{cases} 0, & \text{wenn } \mathbf{l}_i \text{ keine Tiefenschätzung hat} \\ \widehat{d}_{i,j} - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{P}_j \mathbf{l}_i & \text{andernfalls} \end{cases} \tag{6.11}$$

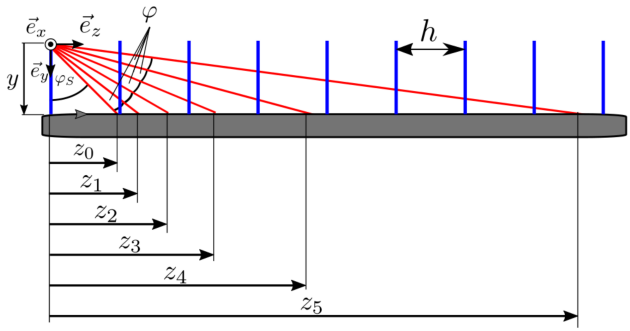


Abbildung 6.9.: Merkmalspunkte auf der Bodenebene sind von großer Relevanz für eine genaue Bewegungsschätzung, insbesondere auf der Autobahn oder auf Landstraßen. Die Tiefen der Sichtstrahlen $z_i = y \cdot \tan(\varphi_s + i \cdot \varphi)$ sind von Strahl zu Strahl stark unterschiedlich. Daher ist die Segmentierung aus Abschnitt 6.2.3 mit Histogrammen mit konstanter Klassenweite h nicht sinnvoll. Stattdessen werden Merkmale auf der Bodenebene zuerst detektiert und die lokale Ebenenschätzung durch die Bodenebene regularisiert.

für das Optimierungsproblem hinzugefügt, welche Abweichungen der geschätzten Landmarkentiefe vom gemessenen Wert aus dem LIDAR bestrafen. Hierbei bezeichnet $\mathbf{l}_i$ die i-te Landmarke, $\mathbf{P}_j$ die j-te Pose in homogenen Koordinaten und $\widehat{d}$ die geschätzte Merkmalstiefe. Die Indizes i,j bezeichnen Landmarken-Posen-Paare, für welche valide Tiefenschätzungen vorliegen.
Für urbane Szenarien mit einer großen Anzahl an Tiefenschätzungen ist der zusätzliche Fehlerterm ξ hinreichend. Auf der Autobahn und der Landstraße hingegen sind typischerweise nur wenige Merkmale vorhanden, für welche sich eine valide Tiefe schätzen lässt. Daher wird in diesem Fall eine aus der monokularen Visuellen Odometrie bekannte Regularisierung eingesetzt. Die ältesten Posen im Optimierungsfenster sind die genauesten — ausgehend vom Zeitpunkt ihrer Aufnahme haben sie die meiste Information über ihre Zukunft. Im Optimierungsproblem soll daher eine Abweichung der Skale von diesen ältesten Posen bestraft werden. Dazu wird ein zusätzlicher Kostenfunktor ν eingeführt, welcher Abweichungen des relativen Translationsvektors zwischen diesen Posen bestraft:

$$\nu(\mathbf{P}_1, \mathbf{P}_0) = \widehat{s}(\mathbf{P}_1,\mathbf{P}_0) - \bar{s} \tag{6.12}$$

mit $\mathbf{P}_0$, $\mathbf{P}_1$ als den beiden ältesten Posen im Optimierungsfenster und

$$\widehat{s}(\mathbf{P}_1,\mathbf{P}_0) = \|\text{Translation}(\mathbf{P}_0^{-1}\mathbf{P}_1)\|_2^2. \tag{6.13}$$

Die Konstante $\bar{s}$ hat hierbei den Wert von $\widehat{s}(\mathbf{P}_1,\mathbf{P}_0)$ vor der Optimierung. Diese Regularisierung der Skale führt zu einer glatteren Trajektorie und macht das Optimierungsproblem robuster gegenüber wenigen und fehlerhaften Tiefenschätzungen. Eine Darstellung des Problems als Graph ist in Abbildung 6.10 gegeben.

Das gesamte Optimierungsproblem setzt sich also aus Regularisierung

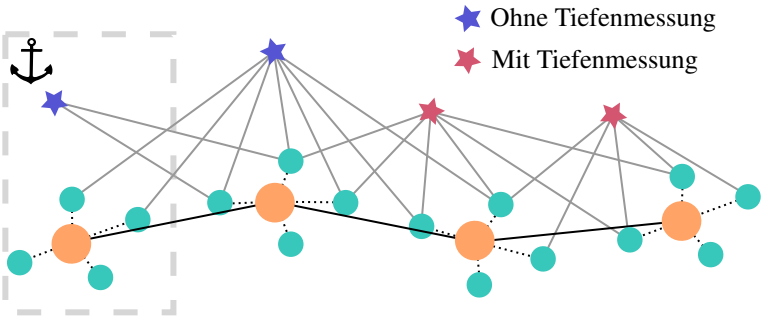


Abbildung 6.10.: Graph zur Schätzung von Visueller Odometrie mit Tiefenmessungen. Da teilweise nur wenige Tiefenschätzungen hoher Qualität gewonnen werden können, werden Messungen mit (rot) und ohne Tiefenmessungen (blau) in das Optimierungsproblem integriert. Dies bewirkt eine hohe Robustheit und Genauigkeit sowohl für die Translationsschätzung als auch für die Rotationsschätzung.

$\nu(\mathbf{P}_1,\mathbf{P}_0)$, Rückprojektionsfehler $\phi(\mathbf{l}_i,\mathbf{P}_i)$ und Tiefenfehler $\xi(\mathbf{l}_i,\mathbf{P}_j)$ zusammen. Zur Robustifizierung werden Cauchy-Gewichtsfunktionen $\rho_\phi(x)$, $\rho_\xi(x)$ eingesetzt, um Ausreißer in den Fehlerresiduen sowohl des Rückprojektionsfehlers als auch des Tiefenfehlers niedriger zu gewichten. Mit $\mathcal{P}$ und $\mathcal{L}$, den Mengen der gewählten Keyframes und Landmarken, wird das resultierende Optimierungsproblem demnach wie folgt formuliert:

$$\begin{aligned}&\underset{\mathbf{P}_j\in\mathcal{P},\mathbf{l}_i\in\mathcal{L},d_i\in\mathcal{D}}{\operatorname{argmin}} \quad \|\nu(\mathbf{P}_1,\mathbf{P}_0)\|_2^2 \\ &+\sum_i\sum_j w_0\rho_\phi(\|\phi(\mathbf{l}_i,\mathbf{P}_i)\|_2) + w_1\rho_\xi(\|\xi(\mathbf{l}_i,\mathbf{P}_j)\|_2),\end{aligned} \tag{6.14}$$

mit dem Rückprojektionsfehler $\phi(\mathbf{l}_i,\mathbf{P}_j) = \bar{\mathbf{l}}_{i,j} - \pi(\mathbf{l}_i, \mathbf{P}_j)$ und den Gewichten w_0 und w_1, welche verwendet werden, um die Skale der Kostenfunktionen anzupassen. Um Rechenaufwand zu sparen, wird zusätzliches Trimmen der Fehlerresiduen, wie in Abschnitt 5.5 beschrieben, angewandt.

6.3. Skalenschätzung aus Bewegungsrelation der Kamera zum Bewegungsmodell

Ist die Kamera nicht im Bewegungszentrum des Fahrzeugs montiert und bewegt sich das Fahrzeug gemäß des kinematischen Einspurmodells, so kann die Skale unter bestimmten Bedingungen berechnet werden. Die Annahmen hierfür lauten:

- Das Fahrzeug bewegt sich zwischen den ausgewerteten Frames auf einer Kreisbahn gemäß dem kinematischen Einspurmodell.
- Der Fahrzeugreferenzpunkt bewegt sich tangential zum Momentanpol der Bewegung. Im Fall des kinematischen Einspurmodells bedeutet dies, dass der Referenzpunkt auf der Hinterachsmitte liegen muss.
- Der Schräglaufwinkel muss null sein.

Insbesondere für Frame-zu-Frame-Bewegungsschätzungen ist das sehr nützlich, da so die Skale ohne Rekonstruktion der Umgebung geschätzt werden kann. Die Herleitung ist hier der Übersichtlichkeit wegen auf den zweidimensionalen Fall auf der Bodenebene beschränkt, jedoch durch zusätzliche Freiheitsgrade für Nicken und Wanken auch in drei Dimensionen übertragbar.
Die Skale ist über die Bogenlänge l und damit über den Kreisradius $r_i = \frac{l}{\gamma}$ mit Kreiswinkel γ der Fahrzeugbewegung beschrieben. Ist die Kamera in Fahrzeuglängsrichtung gegenüber der Fahrzeugreferenzposition verschoben, so wird die Skale in Kurvenfahrten beobachtbar. Dies ist in Abbildung 6.11 dargestellt und sei im Folgenden erläutert.
Das Fahrzeug bewegt sich auf dem inneren Kreis mit Radius $r_i = \frac{l}{\gamma}$. Die Kamera hingegen bewegt sich durch den Kameralängsversatz a zum Fahrzeugreferenzpunkt auf dem äußeren Kreis mit Radius r_a. Mit der monokularen Kamera können der Differenzwinkel γ sowie die Richtung der Translation $\mathbf{t} = (t_x, t_y)^T$ mit $\|\mathbf{t}\|_2 = 1$ zwischen P_0 und P_1 beobachtet werden. Der Kameralängsversatz a ist bekannt und gesucht ist die Länge s

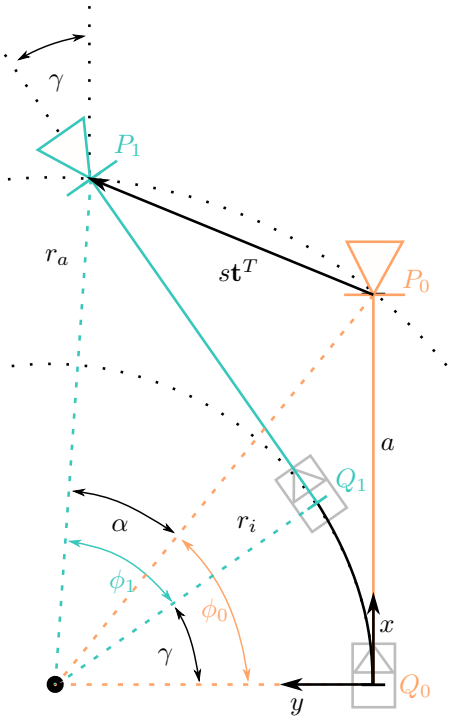


Abbildung 6.11.: Skizze zur Schätzung der Skale in Kurven. Durch den Längsversatz a der Kamera zum Fahrzeug wird die Skale in Kurven beobachtbar.

des Translationsvektors.
Es gilt

$$s \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \overrightarrow{P_0P_1} = \overrightarrow{Q_0P_1} - \overrightarrow{Q_0P_0}. \tag{6.15}$$

mit

$$\overrightarrow{Q_0P_0} = \begin{pmatrix} a \\ 0 \end{pmatrix} \tag{6.16}$$

und

$$\overrightarrow{Q_0P_1} = \overrightarrow{Q_0Q_1} + \overrightarrow{Q_1P_1} = r_i \begin{pmatrix} \sin(\gamma) \\ 1 - \cos(\gamma) \end{pmatrix} + a \begin{pmatrix} \cos(\gamma) \\ \sin(\gamma) \end{pmatrix}. \tag{6.17}$$

Daraus folgt

$$s \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \overrightarrow{Q_0 P_1} - \overrightarrow{Q_0 P_0} = r_i \begin{pmatrix} \sin(\gamma) \\ 1 - \cos(\gamma) \end{pmatrix} + a \begin{pmatrix} \cos(\gamma) - 1 \\ \sin(\gamma) \end{pmatrix}. \quad (6.18)$$

Zusätzlich existiert eine Beziehung zwischen s und r_i. Die Länge s des Translationsvektors ist die Kreissehne des Kreisstücks mit Kreiswinkel α und Radius r_a (in Abbildung 6.11). Somit gilt

$$s = 2\sqrt{r_i^2 + a^2} \sin\left(\frac{\alpha}{2}\right). \quad (6.19)$$

Durch die Bedingungen $\alpha = \phi_1 + \gamma - \phi_0$, $\tan(\phi_0) = \frac{a}{r_i}$ und $\tan(\phi_1) = \frac{a}{r_i}$ wird ersichtlich, dass

$$\alpha = \gamma \quad (6.20)$$

gilt. Durch Substitution von s in Gleichung 6.18 folgen die Bedingungen

$$t_x \cdot 2\sqrt{r_i^2 + a^2} \sin\left(\frac{\gamma}{2}\right) = r_i \sin(\gamma) + a(\cos(\gamma) - 1) \quad (6.21)$$

und

$$t_y \cdot 2\sqrt{r_i^2 + a^2} \sin\left(\frac{\gamma}{2}\right) = r_i(1 - \cos(\gamma)) + a \sin(\gamma). \quad (6.22)$$

Eine dieser zwei Gleichungen kann zur Ermittlung von r_i verwendet werden, womit die Skale s bzw. die Bogenlänge l bestimmbar werden. Falls γ, t_x oder t_y unbekannt sind, kann mit der übrigen Gleichung neben der Skale eine weitere Größe geschätzt werden.
Dieses Ergebnis ist für eine einzelne Kamera in Simulation bestätigt und in Abbildung 6.12 gezeigt. Hierbei führt jedoch schon die Zugabe von kleinem Rauschen dazu, dass das Minimum nicht mehr gut ausgeprägt ist. Für mehrere Kameras ist allerdings auch auf realen Daten das Minimum sichtbar, wie in Abbildung 6.13 und Abbildung 6.14 zu sehen ist. Die Skale ist in der Fehlerlandschaft jedoch lediglich zu drei Zeitpunkten der gesamten Kurvenfahrt deutlich erkennbar. Einerseits ist das auf die schlecht konditionierte Beobachtbarkeit des Problems zurückzuführen. Dies könnte in Zukunft durch genauere Merkmalsextraktion verbessert werden. Eine weitere Einschränkung dieser Methodik ist die grundlegende Bedingung der Kreisfahrt ohne Schräglauf. Für Kurvenfahrten mit erhöhter Geschwindigkeit ist dies im Allgemeinen nicht erfüllt. Für langsam durchfahrene Kurven, wie zum Beispiel in innerstädtischen Szenarien, ist diese Methodik jedoch eine wertvolle zusätzliche Informationsquelle für die Skalenschätzung.

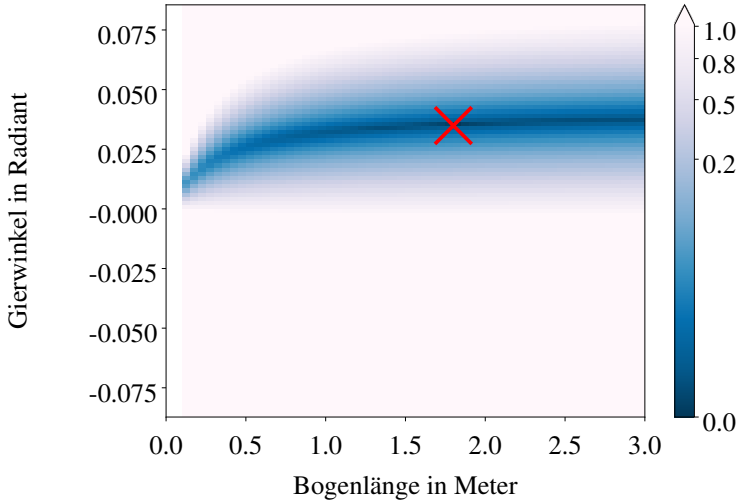


Abbildung 6.12.: Fehlerlandschaft des in Abbildung 4.4 gezeigten optischen Flusses ohne Rauschen. Hier ist eine einzelne Kamera mit den gleichen Parametern wie im KITTI-Datensatz simuliert. Das rote Kreuz markiert das globale Optimum. Das Minimum des Gierwinkels ist deutlich besser bestimmt als das der Skale. Ohne Rauschen kann die Bogenlänge jedoch korrekt als $l = 1{,}8\ \mathrm{m}$ geschätzt werden.

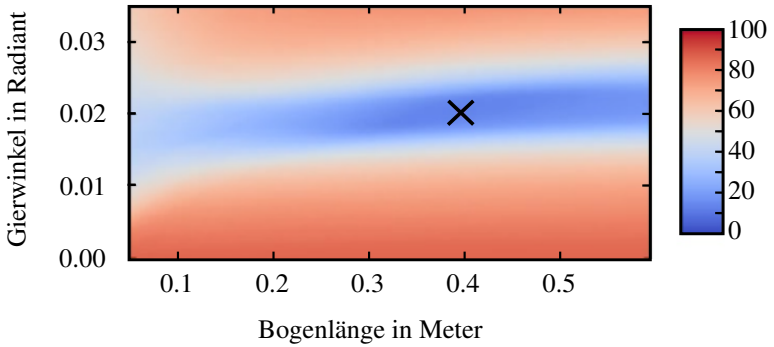


Abbildung 6.13.: Fehlerlandschaft zu Gleichung 4.7 mit mehreren Kameras auf dem Versuchsfahrzeug während einer Kurve. Der Fehler ist im Bezug auf den Maximalfehler in Prozent angegeben und das schwarze Kreuz zeigt das Minimum der Fehlerlandschaft. Wie in Abschnitt 6.3 erläutert, ist die Skale in Kurven beobachtbar. Durch das langsame Durchfahren der Kurve sind die Annahmen, insbesondere die von vernachlässigbarem Schräglaufwinkel, hier erfüllt.

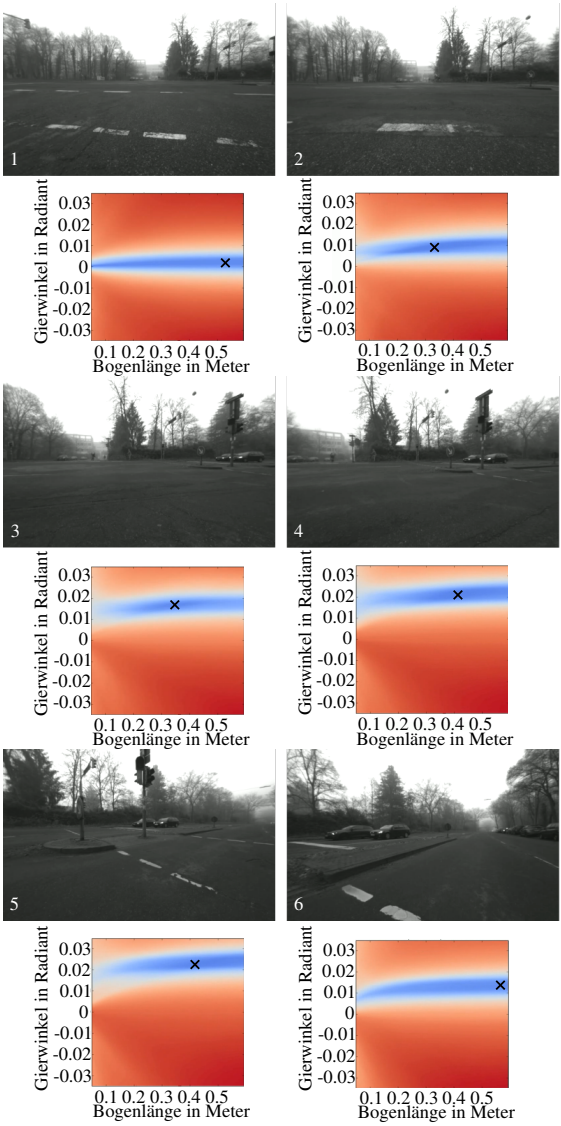


Abbildung 6.14.: Momentaufnahmen der Fehlerlandschaft des Optimierungsproblems während einer Kurvenfahrt. Rot bezeichnet große Fehler, dunkelblau kleine. Zu Beginn ist die Skale noch nicht bestimmbar (Bilder 1 und 2). Während der Kurvenfahrt wird im Kostental jedoch ein Minimum sichtbar (schwarze Kreuze, Bilder 3 bis 5), was die Beobachtbarkeit der Skale zeigt. Bei der Kurvenausfahrt (Bild 6) wird die Skale jedoch wieder unbestimmbar, da sich das Minimum von Frame zu Frame stark unterscheidet.

Kapitel 7

Extrinsische Kalibrierung von LIDAR und Kamera

Um die in Abschnitt 6.2.1 beschriebene Skalenschätzungsmethode mit Tiefeninformation aus einem LIDAR umsetzen zu können, muss die Projektion der LIDAR-Messungen in das Kamerabild möglich sein. Neben der intrinsischen Kalibrierung der Kamera muss dafür die Transformation des LIDAR- in das Kamerakoordinatensystem bekannt sein, welche auch als extrinsische Kalibrierung bezeichnet wird. Hierzu wird in dieser Arbeit eine Methode vorgestellt, welche diese Transformation genau schätzen kann, indem sie die Sichtbarkeit des vom LIDAR ausgesendeten Infrarotlichts auf dem Kamerasensor ausnutzt [45].

7.1. Stand der Technik

Die extrinsische Kalibrierung von LIDAR zu Kamera mit hoch genauen und dichten Punktmengen ist ein intensiv betrachteter Bereich aktueller Forschung ([46], [47], [48], [49], [50], [51]). Eine in der Wissenschaft häufig genutzte Sensorreihe ist die der Firma $Velodyne$, welche Sensoren mit 64, 32 und 16 Scanebenen und einer horizontalen Auflösung von $0{,}2°$ und Tiefenrauschen von ca. $1{,}5\ \mathrm{cm}$ Standardabweichung bereitstellen. Mit diesem Sensor kann eine detailreiche, dreidimensionale Punktmenge der Szene gewonnen werden. Die meisten Kalibrieralgorithmen dieser und ähnlicher Sensoren rekonstruieren dreidimensionale Punkte aus Kamerabildern und registrieren diese zur vom LIDAR gemessenen Punkt-

menge ([46], [47]). Dabei wird die Rekonstruktion der Punktmenge aus einer monokularen Kamera durch Schachbretter ermöglicht. Die Registrierung der Punktmenge wird mithilfe des bekannten ICP-Algorithmus [52] oder durch Abgleich der Oberflächennormalen durchgeführt ([47], [46]). Zhang et al. [48] stellten eine ähnliche Methode vor, bei welcher zuerst eine Kalibrierebene mithilfe von Schachbrettern aus der Kamera rekonstruiert und anschließend die Distanz der LIDAR-Punkte zu dieser minimiert wird. Hierzu können folglich auch LIDAR-Sensoren, welche nur in einer Ebene abtasten, verwendet werden.
Die Verwendung von Kalibrierkörpern erzeugt zusätzlichen Aufwand und Kosten. Darum verwenden Scaramuzza et al. [49] eine Umgebungsrekonstruktion aus einer monokularen Kamera, ähnlich zu der in Abschnitt 5 beschriebenen Methode. Die resultierende Punktmenge wird wiederum zur LIDAR-Punktmenge mithilfe von ICP registriert. Somit werden keine vorab bekannten Kalibrierkörper benötigt.
Kalibrierverfahren, welche auf Umgebungsrekonstruktion basieren und mit geometrischen Beziehungen die extrinsische Kalibrierung schätzen, bedürfen jedoch stets einer hohen Auflösung des LIDAR. Rekonstruktionsfreie Methoden hingegen sind besser generalisierbar und können auch auf Sensoren mit niedriger Auflösung angewendet werden. Ein populäres Beispiel hierfür wurde von Taylor et al. [51] vorgestellt, welche bewegungsgebende Sensoren registrieren, indem ihre Trajektorien angeglichen werden. Nachteilig ist hierbei, dass die Kalibrierung unter den Ungenauigkeiten der Bewegungsschätzung leidet.
Li et al. ([53], [50]) umgehen die Umgebungsrekonstruktion, indem Grauwertkanten im Kamerabild erkannt und zu Tiefensprüngen registriert werden. Die Minimierung des Rückprojektionsfehlers resultiert in der extrinsischen Sensorkalibrierung.
All diese Methoden gehen jedoch von einem LIDAR aus, welcher mit hoher horizontaler Auflösung die Umgebung wahrnimmt. Insbesondere für normalenbasierte Ansätze ([46], [47], [48]) ist die Auflösung kritisch für den Erfolg der Kalibrierung. Auch Tiefenkanten ([53]) können nur durch LIDAR-Sensoren mit hoher Winkelauflösung hinreichend genau detektiert werden. Diese feine horizontale Auflösung ist nur möglich, wenn der vom einzelnen Laserstrahl erzeugte Lichtfleck klein ist. Da augenverträgliche Laser in ihrer Energiedichte beschränkt sein müssen, folgt daraus eine Beschränkung der Reichweite des LIDAR. Werden also hohe Reichweiten benötigt oder müssen Produktionskosten gering gehalten werden (wie zum Beispiel bei aktuellen Solid-State-LIDAR-Sensoren [54]), ist die Auflösung typischerweise sehr viel geringer und sind aktuelle Methoden nicht anwendbar.

Darum wird in dieser Arbeit eine Methode vorgestellt, welche auch diese LIDAR-Varianten zu einer monokularen Kamera kalibrieren kann. Hierzu wird ausgenutzt, dass das vom LIDAR emittierte Infrarotlicht in handelsüblichen Kamerasensoren zu sehen ist. Die für diese Methode verwendete Fehlermetrik ist für eine Vielzahl von Kameras anwendbar, inklusive solcher mit großen Verzerrungen (Fischaugenkameras) und unterschiedlichem Ursprung der Sichtstrahlen (wie zum Beispiel Katadioptrische Kameras). Dies ermöglicht die Verwendung von Sensoren ab drei Punkten pro Scan und großem Tiefenrauschen.

7.2. Methode

7.2.1. Problemformulierung

Grundvoraussetzung für eine extrinsische Kalibrierung ist die individuelle intrinsische Kalibrierung aller Sensoren. Diese wird im Folgenden als gegeben vorausgesetzt und drückt sich in folgenden Abbildungen aus:

$$\pi_{LIDAR,i}(d_i) \rightarrow \mathbf{x}_i \quad , d_i \in \mathbb{R}, \mathbf{x}_i \in \mathbb{R}^3, i = 1 \dots n \tag{7.1}$$

von einer Tiefenmessung d_i eines Strahls i zu einem dreidimensionalen Punkt $\mathbf{x}_i$ im LIDAR-Koordinatensystem und

$$\begin{aligned} &\pi_{Kamera}(\mathbf{p}_i) \rightarrow \mathbf{v}_i \cdot s_i + \mathbf{r}_i, \\ &s_i \in \mathbb{R}, \mathbf{p}_i \in \mathbb{R}^2, \mathbf{v}_i, \mathbf{r}_i \in \mathbb{R}^3, \|\mathbf{v}_i\|_2 = 1 \end{aligned} \tag{7.2}$$

von einem Punkt auf der Bildebene der Kamera $\mathbf{p}_i$ zu einem Sichtstrahl mit Aufpunkt $\mathbf{r}_i$ und Richtung $\mathbf{v}_i$. Die Variable s bezeichnet den hier nicht beobachtbaren Abstand des zu $\mathbf{p}_i$ korrespondierenden dreidimensionalen Punktes. Um Messungen im LIDAR den Bildpunkten zuordnen zu können, muss die extrinsische Kalibrierung zwischen Kamera und LIDAR bestimmt werden. Diese wird im Folgenden mit $\Delta\mathbf{P}$ bezeichnet.
Ein entscheidender Punkt für ihre korrekte Schätzung ist die Wahl geeigneter Merkmale, um eine Kalibrierungshypothese zu bewerten. Klassischerweise werden hierfür geometrische Merkmale, wie zum Beispiel die Form eines Kalibrierkörpers, verwendet. Ist die Auflösung des LIDAR jedoch gering, so ist dies nicht akkurat. Darum wird in dieser Arbeit ausgenutzt, dass handelsübliche Kamerasensoren das von LIDAR typischerweise ausgesendete Infrarotlicht wahrnehmen können. Dadurch kann der Punkt, an welchem der Laserstrahl ein Objekt trifft, im Kamerabild wahrgenommen werden, wie in Abbildung 7.1 dargestellt ist.

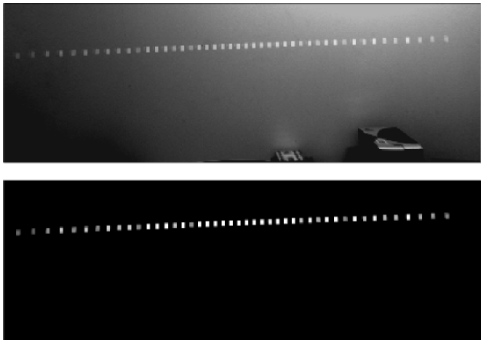


Abbildung 7.1.: Lichtflecken des LIDAR an einer Wand, welche von einer Kamera aufgenommen werden. Im oberen Bild ist die Szene mit Umgebungsbeleuchtung zu sehen, im unteren Teil herrscht Dunkelheit. Die Belichtungszeit der Kamera muss bei der Kalibrierung typischerweise deutlich länger gewählt werden als während des Gebrauchs bei Tageslicht, um das Infrarotlicht des LIDAR wahrnehmen zu können.

In Abbildung 7.2 wird Schritt für Schritt die Methode dargestellt. Diese wird um die Lösung des Optimierungsproblems

$$\Delta\hat{\mathbf{P}} = \underset{\Delta\mathbf{P}=f(\alpha,\beta,\gamma,t_x,t_y,t_z)}{\operatorname{argmin}} \quad \sum_i \lambda(\pi_{Kamera}(\mathbf{p}_i), \Delta\mathbf{P} \cdot \pi_{LIDAR,i}(d_i)) \tag{7.3}$$

herum aufgebaut. Hierbei bezeichnen α,β,γ die Rotationswinkel, t_x,t_y,t_z die Elemente des Translationsvektors und die Abbildung $f(\dots)$ transformiert diese in eine Darstellung als homogene Matrix des $\mathbb{R}^{4\times4}$. Der Auswahl der Fehlermetrik $\lambda(\dots,\dots)$ zwischen Sichtstrahl und LIDAR-Punkt wird hier das Unterkapitel 7.2.2 gewidmet. Die als Eingang für das Optimierungsproblem verwendeten Merkmale werden in den Abschnitten 7.2.3 und 7.2.4 erläutert.

7.2.2. Unterstützung verschiedener Kameramodelle

Die allgemeinste Beschreibung der Kamera-Kalibrierung ist eine Abbildung von einer zweidimensionalen Bildkoordinate auf einen dreidimensionalen Sichtstrahl, wie in Gleichung 7.2 formuliert. In diesem Fall wird die Fehlermetrik als die Summe der Abstandsquadrate zwischen Sichtstrahl und der Punktmessung des LIDAR gewählt und das zugehörige Optimierungsproblem kann wie folgt formuliert werden:

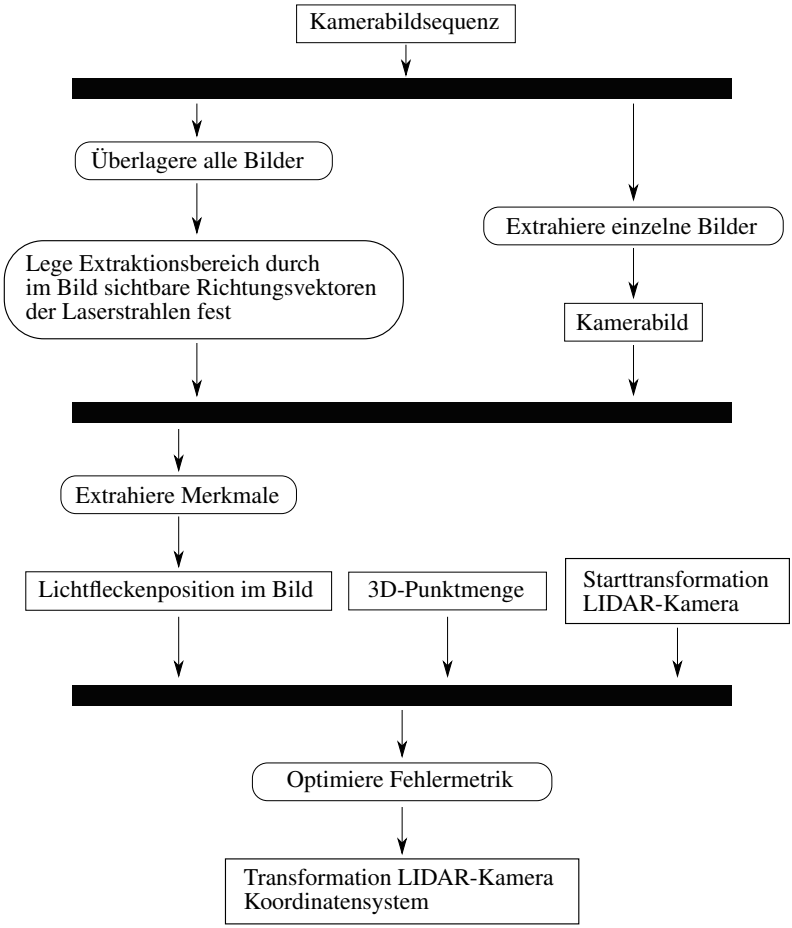
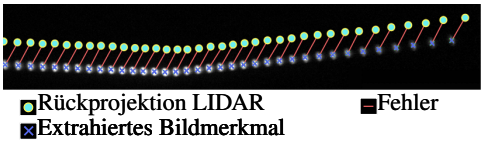
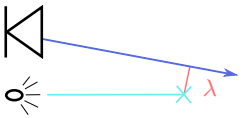


Abbildung 7.2.: Ablauf der extrinsischen LIDAR-zu-Kamera-Kalibrierung als Aktivitätsdiagramm.



(a) Fehlermetrik für eine als Lochkamera modellierte Kamera.



(b) Fehlermetrik für ein generalisiertes Kameramodell.

Abbildung 7.3.: Zwei Fehlermetriken für das Optimierungsproblem in Gleichung 7.3. Für die in Abbildung 7.3a dargestellte Fehlermetrik wird der in Abschnitt 3.1 formulierte Rückprojektionsfehler zwischen der LIDAR-Projektion (zyan) und dem extrahierten Merkmal (grün) verwendet. Die in Abbildung 7.3b gezeigte generalisierte Fehlermetrik ist die minimale Euklidische Distanz (rot) zwischen der Messung des LIDAR (blau) in $\mathbb{R}^3$ zum Sichtstrahl des in der Kamera beobachteten Merkmals (grün). Während die Fehlermetrik in Abbildung 7.3b auch auf komplexere Kameramodelle angewandt werden kann, zeigt die Fehlermetrik in Abbildung 7.3a bessere Konvergenzeigenschaften für das Lochkameramodell.

$$\Delta\hat{\mathbf{P}} = \underset{\Delta\mathbf{P}=f(\alpha,\beta,\gamma,t_x,t_y,t_z)}{\operatorname{argmin}} \sum_i \left\| \frac{(\Delta\mathbf{P}\mathbf{x}_i - \mathbf{r}_i)}{\|\Delta\mathbf{P}\mathbf{x}_i - \mathbf{r}_i\|_2} \times \mathbf{v} \right\|_2^2, \tag{7.4}$$

wobei der $\times$-Operator das Kreuzprodukt symbolisiert. Diese Fehlermetrik ist in Abbildung 7.3b dargestellt. Mit dieser können somit auch Kameras mit hochgradig nicht-linearen Kameramodellen für die Kalibrierung verwendet werden, wie zum Beispiel solche mit Fischaugenobjektiven oder Katadioptrische Kameras mit einem nicht einheitlichen Aufpunkt der Sichtstrahlen. Für Kameras, welche durch eine Lochkamera angenähert werden können, ist diese Metrik zwar auch gültig, jedoch zeigt der Rückprojektionsfehler bessere Konvergenzeigenschaften auf, wie durch Hartley und Zissermann [34] beschrieben. Im Falle eines Lochkameramodells existiert eine eindeutige Abbildung $\pi^{-1}_{Kamera}(\mathbf{x}_i)$ von $\mathbf{x}_i \in \mathbb{R}^3$ nach $\mathbf{p}_i \in \mathbb{R}^2$. Somit kann die geschätzte Position der LIDAR-Messungen abhängig von $\Delta\mathbf{P}$ in das Kamerabild projiziert und die quadratische Distanz zur Beobachtung $\mathbf{p}_i$ in der Kamera minimiert werden (siehe Abbildung 7.3a). Dieses Optimierungsproblem ist in Gleichung 7.5 formuliert.

$$\Delta\hat{\mathbf{P}} = \underset{\Delta\mathbf{P}=f(\alpha,\beta,\gamma,t_x,t_y,t_z)}{\operatorname{argmin}} \sum_i \left\| \pi^{-1}_{Kamera}(\Delta\mathbf{P}\mathbf{x}_i) - \mathbf{p}_i \right\|_2^2. \tag{7.5}$$

Um den Messbereich möglichst vollständig abzudecken, ist es wichtig, Messungen in variierender Distanz zu generieren. Dies wird durch ein Brett

erreicht, welches in unterschiedlichen Distanzen vor den Sensoren platziert wird.

7.2.3. Ausreißerbehandlung und Merkmalsassoziation

Dieser Abschnitt beschreibt die Extraktion der visuellen Merkmale aus dem Kamerabild und deren Assoziation mit dem zugehörigen Laserstrahl. Ist der Abstand zwischen LIDAR und Wand groß, so ist die in der Kamera wahrgenommene Lichtmenge klein, wohingegen bei geringem Abstand die Lichtflecken ineinander fließen (Abbildung 7.4). Die hier vorgestellte

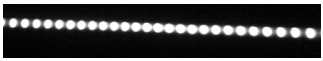


Abbildung 7.4.: Wird die Distanz zwischen LIDAR und Objekt gering, so fließen die Lichtflecken im Kamerabild ineinander über.

Ausreißerbehandlung und Assoziationsstrategie löst dieses Problem. Hierzu werden die Sensoren fixiert und der Abstand zwischen Sensoren und einem Brett variiert. Die so gemessene Sequenz von Intensitätsbildern wird akkumuliert und normiert, sodass eine in Abbildung 7.5 gezeigte fächerartige Struktur sichtbar wird. Jeder Strahl des Fächers korrespondiert hierbei zu einem Laserstrahl. Somit wird also die in Abschnitt 3.2 vorgestellte Epipolargeometrie zwischen LIDAR und Kamera physikalisch sichtbar, wobei jeder Strahl des Fächers die zugehörige Epipolarlinie darstellt. Anschließend werden Messungen in der Nähe der Epipolarlinien gesucht und Ausreißer entfernt. Kriterien hierfür sind die Entfernung der Messung zur Epipolarlinie und die in der Kamera wahrgenommene Helligkeit der Messung. Mithilfe einer grob eingemessenen Start-Transformation zwischen Kamera und LIDAR können so zusätzlich Kameramessungen zu LIDAR-Strahlen assoziiert werden.

7.2.4. Merkmalsextraktion

Die hohe Präzision der extrahierten Merkmale ist grundlegend für eine qualitativ hochwertige Kalibrierung. Dank der Abschottung von Umgebungslicht kann die Positionserkennung der Merkmale mit sehr hoher Genauigkeit ausgeführt werden. Die Merkmalsextraktion wird mithilfe der in Abschnitt 7.2.3 extrahierten Epipolarlinien durchgeführt, wie in Abbildung 7.6 schematisch dargestellt ist.
In einem ersten Vorverarbeitungsschritt wird eine Gauß-Glättung auf das Kamerabild angewandt, um Rauschen zu entfernen. Anschließend wird um

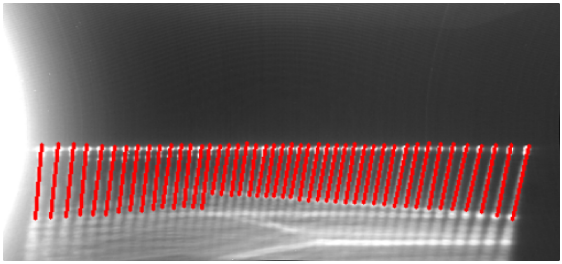


Abbildung 7.5.: Zeitlich akkumuliertes Kamerabild mit dem LIDAR-Sensor als Lichtquelle und variierender Distanz zur Ebene. Die Akkumulation der einzelnen Punkte jeder Laserdiode in unterschiedlicher Distanz bringt die Richtung der Laserstrahlen zum Vorschein (rot).

jede Epipolarlinie ein Suchbereich gelegt und der Bildpunkt mit maximaler Helligkeit ausgewählt. Somit können Merkmale in nur einem Schritt extrahiert und assoziiert werden.

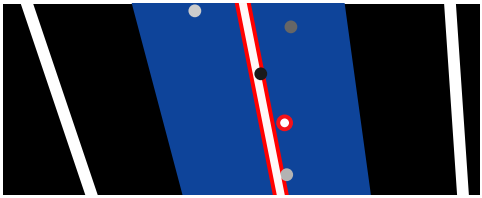


Abbildung 7.6.: Skizze der Merkmalsextraktion. Die Richtungen der Laserstrahlen sind weiß, der momentan betrachtete Laserstrahl ist rot unterlegt. Lichtflecken des Infrarotlichts sind als graue bis weiße Punkte skizziert, um unterschiedliche Reflektanz zu symbolisieren. Zuerst wird ein Suchbereich (blau) um den betrachteten Laserstrahl ausgewählt. Innerhalb des Suchbereichs wird der zum Strahl zugehörige visuelle Messpunkt (rot) als der Punkt mit maximaler Reflektanz ausgewählt. Punkte mit zu geringer Reflektanz (grau und schwarz) werden zurück gewiesen.

7.3. Ergebnisse

Um die Anwendbarkeit dieser Methode auf LIDAR-Sensoren mit niedriger und hoher Auflösung zu demonstrieren, werden die Experimente auf zwei Sensoraufbauten angewendet, welche mit den in Abbildung 7.7 gezeigten LIDAR-Sensoren ausgestattet sind.

(a) *Spies RMS4/90–106 B* und Kamera.

(b) *Pepperl & Fuchs R2000* und Kamera.

Abbildung 7.7.: Für die Experimente verwendete LIDAR-Sensoren.

- Ein Prototyp *Spies RMS4/90–106 B* [55] (Abbildung 7.7a) mit einer Tiefengenauigkeit von $8\ \mathrm{cm}$ Standardabweichung, einer Reichweite von $120\ \mathrm{m}$ und einer horizontalen Auflösung von $2°$.
- Ein *Pepperl & Fuchs R2000* [56] (Abbildung 7.7b) mit einer Tiefengenauigkeit von $0{,}2\ \mathrm{cm}$ Standardabweichung, einer Reichweite von $10\ \mathrm{m}$ und einer horizontalen Auflösung von $0{,}07°$. Für die Kalibrierung wird dessen Drehgeschwindigkeit so angepasst, dass seine horizontale Auflösung auch $2°$ beträgt, um die Unterscheidbarkeit der LIDAR-Strahlen zu gewährleisten.

Der *Spies RMS-4/90–106 B* fokussiert auf hohe Lichtstärke und somit auf hohe Reichweite (ca. $120\ \mathrm{m}$). Der *R2000* hingegen verfügt über eine sehr hohe Distanzpräzision mit einer Standardabweichung von $0{,}2\ \mathrm{cm}$, hat allerdings eine maximale Reichweite von ca. $10\ \mathrm{m}$. Damit ist seine Distanzpräzision zwar um eine Größenordnung höher als die des *RMS-4/90–106 B*, allerdings haben die für die Messungen verwendeten Lichtflecken auch eine kleinere Fläche. Aufgrund der geringen Lichtleistung können nur Distanzen bis ungefähr $3\ \mathrm{m}$ für die Kalibrierung verwendet werden. Dafür ist jedoch mit diesem Sensor eine sehr dichte Abtastung möglich.
Die verwendete Kamera ist vom Typ *PointGrey Flea2 (FL2–14S3M)* mit Global Shutter und $1{,}4\ \mathrm{Megapixel}$ bei $15\ \mathrm{Hz}$. Die Experimente wurden in einem dunklen Raum ohne Tageslicht durchgeführt. Die Datenqualität und die Konvergenzeigenschaften des Algorithmus sind in Abbildung 7.8 illus-

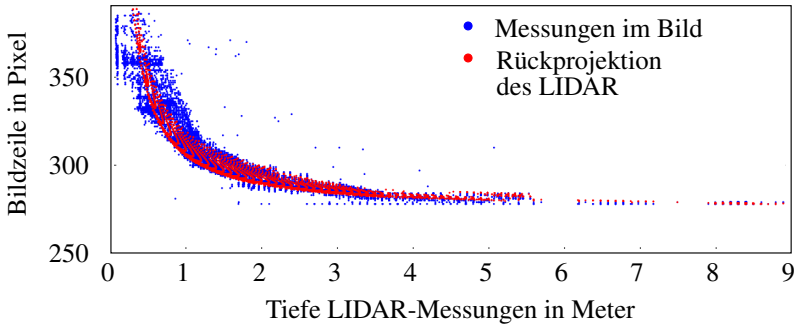


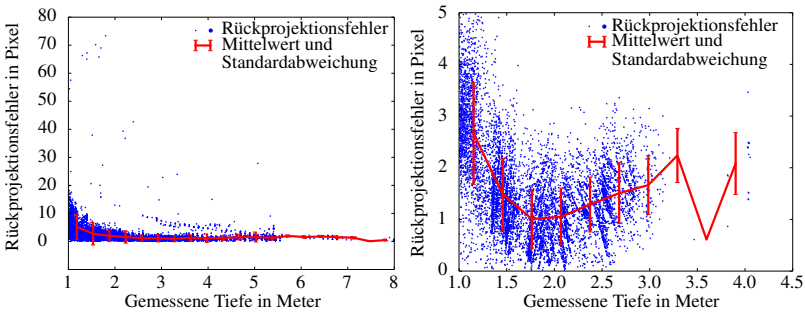
Abbildung 7.8.: Evaluation der Konvergenzeigenschaften des Algorithmus. Messpunkte im Kamerabild sind blau gezeigt und LIDAR-Punkte, welche mithilfe der geschätzten Sensortransformation projiziert wurden, sind rot dargestellt. Bis zu einem Sensorabstand von ca. 1 m weichen die Messungen im Kamerabild aufgrund des Sensorrauschens des LIDAR von der Rückprojektion ab. Ab einem Abstand von ca.1,5 m sind die Einhüllenden der roten und blauen Punkte quasi identisch, was auf die Konvergenz des Algorithmus hinweist.

triert.

Um die Genauigkeit der extrahierten Merkmale bewerten zu können, wird der in Gleichung 7.5 definierte Rückprojektionsfehler verwendet. In Abbildung 7.9 ist der Rückprojektionsfehler in Pixel über den gemessenen Sensorabstand dargestellt. Wie durch die Geometrie zu erwarten, wird der Rückprojektionsfehler mit steigender Tiefe kleiner. Die Konvergenz des Algorithmus ist dadurch zu sehen, dass der Rückprojektionsfehler gegen Null strebt und somit keine systematische Abweichung vorhanden ist. Der leichte Anstieg des Fehlers in Abbildung 7.9b ist auf einen Winkelrestfehler zurückzuführen, welcher durch die geringe Distanz der Messungen hervorgerufen wird. Im Allgemeinen ist der Restfehler mit knapp 1 Pixel sehr gut für den Gebrauch geeignet.

Qualitative Ergebnisse der Kalibrierung des *Spies RMS4/90–106 B* sind in Abbildung 7.10b und für den *Pepperl & Fuchs R2000* in Abbildung 7.10a gezeigt.

Der Rückprojektionsfehler zeigt zwar die Konvergenz des Algorithmus auf, kann jedoch keine quantitative Aussage über die Genauigkeit der Kalibrierung geben. Dazu wird ein Vergleichsverfahren mit deutlich höherer Genauigkeit benötigt, auf welches in Abschnitt 7.4 näher eingegangen wird.



(a) Rückprojektionsfehler für *RMS-4/90–106 B*

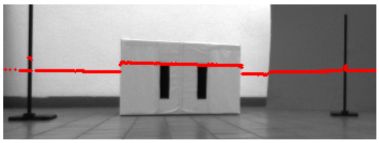
(b) Rückprojektionsfehler für *R2000*

Abbildung 7.9.: Rückprojektionsfehler für *Pepperl & Fuchs R2000* und den Prototyp *Spies RMS-4/90–106 B*. Diese Sensoren repräsentieren LIDAR-Sensoren, welche mit sehr unterschiedlichem Fokus gebaut wurden. Die hier vorgestellte Methode ist im Stande diese beiden sehr unterschiedlichen Sensoren zu kalibrieren, da sowohl präzise, nahe Merkmale als auch weniger genaue, entfernte Merkmale in die Fehlermetrik mit einfließen.

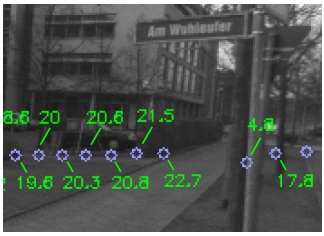
7.4. Evaluation

Die quantitative Evaluation des Kalibrierungsfehlers muss auf einem Verfahren beruhen, welches eine deutlich höhere Genauigkeit als das Schätzverfahren bietet. Darum wird für die Evaluation ausgenutzt, dass der *Pepperl & Fuchs R2000* über eine sehr viel höhere horizontale Auflösung verfügt als für die Kalibrierung nötig ist. Somit können hoch genaue geometrische Merkmale extrahiert werden, um die Genauigkeit der hier vorgestellten Methode für diesen Sensor zu quantifizieren.
Die Referenzmethode ist sehr ähnlich zu herkömmlichen Kalibrierungsverfahren, in welchen Bildkanten zu Tiefenkanten registriert werden, wie zum Beispiel von Li et al. [53] vorgestellt. Um eine hohe Genauigkeit dieses Merkmals zu erreichen, wird ein eigener Kalibrierkörper verwendet, welcher in Abbildung 7.11 zu sehen ist. Dieser besteht aus einer geschlitzten, weißen Box. Bis auf die Schlitze ist diese nach außen hin verschlossen und innen schwarz bemalt, um den Kontrast im Bild zu erhöhen und so die Kantendetektion zu vereinfachen.
Mit diesem Versuchsaufbau können Kanten im LIDAR und im Grauwertkantenbild wie folgt verglichen werden:

1. a) Gruppiere den Scan anhand von Tiefensprüngen.



(a) Projektion einer LIDAR-Messung (rot) eines Scans mit hoher Abtastrate in das Kamerabild nach der Kalibrierung für den *Pepperl & Fuchs R2000*.



(b) Projektion von LIDAR-Messungen (blau) eines niedrig aufgelösten Scans in das Kamerabild nach der Kalibrierung für den *Spies RMS-4/90–106 B*. Tiefenwerte sind grün dargestellt.

Abbildung 7.10.: Rückprojektion für LIDAR-Messungen der Sensoren mit hoher und niedriger Auflösung. Vertikale Sprünge der Scan-Linie deuten auf eine Veränderung der Tiefe hin. Die Korrespondenz zu Objekten im Vordergrund ist gut erkennbar und veranschaulicht die Genauigkeit der Methode.

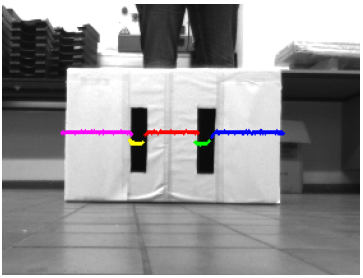


Abbildung 7.11.: Kalibrierkörper mit projizierten LIDAR-Messungen.

b) Finde die Gruppierung, welche am besten zur Form des Kalibrierkörpers passt (Abbildung 7.11). Hierzu werden Körperlänge, Körpertiefe und Schlitzbreite abgeglichen.

c) Extrahiere die Tiefenkanten in den LIDAR-Messungen. Die Randpunkte der zu den Schlitzen gehörenden Gruppen werden mithilfe der geschätzten LIDAR-Kamera-Transformation ins Kamerabild projiziert.

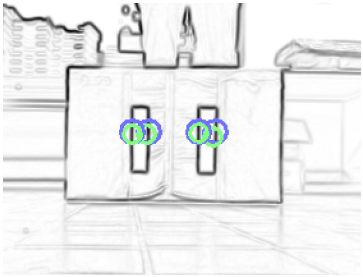


Abbildung 7.12.: Inverses Gradientenbild mit in der Kamera erkannten Kanten (blau) und den korrespondierenden, projizierten LIDAR-Messungen (grün). Die euklidische Distanz zwischen diesen quantifiziert die Genauigkeit der Kalibrierung an diesen Positionen.

2. a) Berechne den Intensitätsgradienten im Kamerabild

$$G = \sqrt{{G_x}^2 + {G_y}^2},$$

mit den Antworten von Sobelfiltern in x- und y-Richtung G_x und G_y.

b) Projiziere alle LIDAR-Messpunkte im Vordergrund auf den Kalibrierkörper und schätze dadurch eine Linie.

c) Extrahiere die Schlitzkanten im Kamerabild. Hierzu werden Maxima in G auf der geschätzten Linie verwendet.

3. Berechne den Rückprojektionsfehler der Kantenpunkte, wie in Abbildung 7.12 dargestellt.

Dieser Fehler wird wiederum in Funktion des gemessenen Abstands in Abbildung 7.13 aufgetragen. Bei geringer Distanz dominiert das Sensorrauschen den Kalibrierungsfehler. Es wird jedoch deutlich, dass ab einem Abstand von $2\ \mathrm{m}$ der Rückprojektionsfehler auf ca. $0{,}5\ \mathrm{px}$ und eine Standardabweichung von $0{,}25\ \mathrm{px}$ fällt. Diese Genauigkeit ist ähnlich zu der des Kantendetektors im Kamerabild, wodurch die hohe Genauigkeit dieser Methode gezeigt wird.

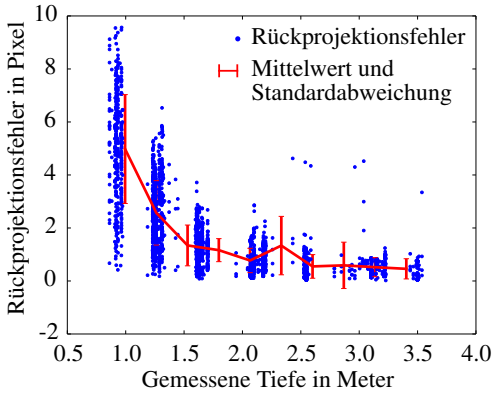


Abbildung 7.13.: Ergebnis der Kamera zu LIDAR-Kalibrierung des *Pepperl & Fuchs R2000*. Der Fehler ist blau gezeigt mit Mittelwert und Standardabweichung in rot. Bei kleinem Abstand zum Sensor überwiegt das Sensorrauschen. Dessen Einfluss klingt jedoch für größere Distanzen ab, sodass ab einer Distanz von $2\ \mathrm{m}$ der Kalibrierfehler von circa $0{,}5\ \mathrm{px}$ mit einer Standardabweichung von $0{,}25\ \mathrm{px}$ sichtbar wird.

Kapitel 8

Evaluation

Die in den Kapiteln 4, 5, 6 beschriebenen Methoden werden in diesem Kapitel evaluiert und untereinander sowie zu anderen Arbeiten verglichen. Hierbei sind die Evaluationen in der Reihenfolge vorzufinden, in welcher die Methodenansätze vorgestellt wurden:

1. Frame-zu-Frame-Bewegungsschätzung
    a) Bewegungsschätzung ohne Skale
    b) Frame-zu-Frame-Skalenschätzung mit Fluchtpunkten
2. Methoden mit zeitlicher Inferenz
    a) Skalenschätzung mit A-Posteriori-Bodenebenenschätzung
    b) Skalenschätzung mit integrierter Bodenoberflächenschätzung
    c) Skalenschätzung mit LIDAR
    d) Skalenschätzung mit LIDAR und integrierter Bodenoberflächenschätzung

Zusätzlich wird die Skalenschätzung durch LIDAR mit zeitlicher Inferenz zu einer Frame-zu-Frame-Methode mit LIDAR verglichen, um die Vor- und Nachteile der zeitlichen Inferenz zu diskutieren. Abschließend werden die besten der vorgestellten Methoden zu konkurrierenden Ansätzen auf dem KITTI-Benchmark [1] in Kontext gesetzt.

Vorgehen der Evaluation Um die Vor- und Nachteile der in dieser Arbeit vorgestellten Ansätze vergleichen zu können, ist ein gemeinsamer

Datensatz notwendig. Im Bereich der Visuellen Odometrie hat sich KITTI [1] als der Standard-Datensatz hervorgetan. Dieser umfasst 22 Sequenzen mit einem Farb- und einem Grauwert-Stereokameraaufbau sowie LIDAR-Messungen eines *Velodyne HDL64*. Zusätzlich verfügt der Datensatz über eine Grundwahrheit der Trajektorien, welche mittels eines differentiellen GPS kombiniert mit einer IMU bereit gestellt wurde. Unter guten Bedingungen, wie freie Sicht auf mehr als fünf Satelliten, erreicht dieser Sensor einen Positionsfehler von unter 5 cm [16]. Alle hier dargestellten Rotations- und Translationsfehler nutzen die für KITTI verwendete Metrik von Geiger et al. [1].
Dieser Datensatz ist als Benchmark konzipiert. Daher sind nur für 11 dieser Sequenzen die Referenztrajektorien veröffentlicht. Evaluiert wird auf den restlichen 11 Sequenzen, für welche die Grundwahrheit nicht öffentlich verfügbar ist. Die Posenschätzungen müssen online eingereicht werden, um zu verhindern, dass auf diesen Daten trainiert wird. Dank der großen Beliebtheit, mit mehreren Uploads pro Monat und knapp 100 veröffentlichten Algorithmen, ist mit diesem Benchmark ein internationaler und objektiver Vergleich der dargestellten Methoden möglich.
Insbesondere für die in Kapitel 4 gezeigte Methode der Rotationsschätzung aus einem Multikamerasystem ist dieser Datensatz jedoch nicht geeignet, da keine Rundumsicht der Kameras vorhanden ist. Um das Gesamtsystem zu evaluieren, werden daher zusätzlich Versuche auf dem am Institut vorhandenen Versuchsträger *BerthaOne* ausgewertet.

Hinweis Bei der Bewertung der Trajektorien für Methoden mit zeitlicher Inferenz ist zu beachten, dass die Latenz, mit der eine Schätzung verfügbar sein soll, Auswirkung auf die Genauigkeit der Schätzung hat. So ist die Schätzung einer Pose akkurater, je weiter hinten sie sich im Optimierungsfenster befindet, da so mehr Informationen über zukünftige Landmarken vorhanden sind. In dieser Arbeit wird stets die erste verfügbare Pose im Optimierungsfenster, also diejenige mit geringster Latenz, verwendet.

8.1. Frame-zu-Frame-Schätzung

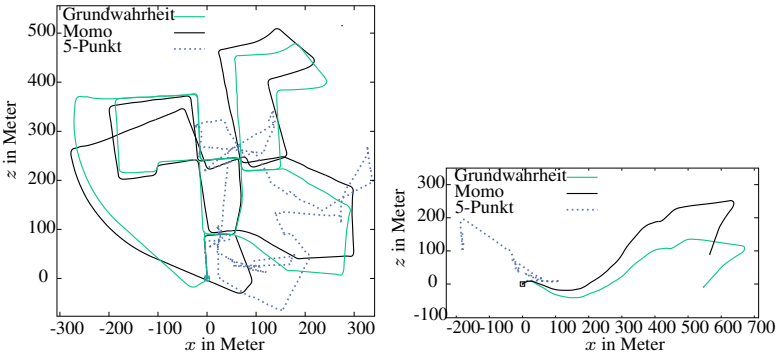
Im Folgenden wird die in Abschnitt 4 vorgestellte Methode zur Rotationsschätzung (hier *MOMO* genannt) auf einer Multikameraplattform evaluiert. Es wird auf dem KITTI-Datensatz gezeigt, wie durch den Einsatz des M-Schätzers die Rotationsschätzung verbessert werden kann. Versuche auf dem durch das Institut bereit gestellten Versuchsträger *BerthaOne* zeigen die akkurate Rotationsschätzung mit mehreren Kameras. Des Weiteren

wird die in Abschnitt 6.1.1 beschriebene Skalenschätzung mit Fluchtpunkten für Frame-zu-Frame-Methoden evaluiert.

8.1.1. Evaluation auf dem KITTI-Datensatz

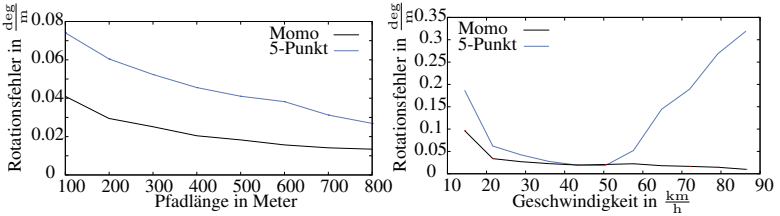
Die hier vorgestellte Methode zur Frame-zu-Frame-Bewegungsschätzung wurde auf dem KITTI-Datensatz evaluiert [16]. Meistens ist hier die Gierwinkeldifferenz zu klein, um die Skale, wie in Abschnitt 6.3 beschrieben, zu schätzen. Daher wurden die hier vorgestellten Algorithmen auf dem öffentlichen Teil des Datensatzes evaluiert, um die Skale aus der Grundwahrheit berechnen zu können. In der Vorverarbeitung wird zunächst eine Gammakorrektur auf die Bilder angewandt, um Helligkeitsunterschieden entgegenzuwirken. Anschließend werden Merkmalspunkte extrahiert sowie Punktkorrespondenzen mithilfe der Deskriptoren und des Matching-Verfahrens von Geiger et al. [17] hergestellt, wie in Abschnitt 3.5 beschrieben ist. Die im vorigen Zeitschritt geschätzte Pose wird als Startwert für die Optimierung verwendet. Der halbe Quartilsabstand der Cauchy-Verteilung, wie in Abschnitt 4.2, beschrieben wird auf 0.0065 eingestellt.
Um die Bewegungsschätzung in wenig texturierter Umgebung zu demonstrieren, wurden lediglich 100–300 Punktkorrespondenzen verwendet, indem ein großer Bereich von $10\ \mathrm{Pixel}$ für die Unterdrückung der Nebenmaxima der Keypoints gewählt wurde. Außerdem wurde bei dieser Evaluation bewusst keine Drift-Reduktion durch zeitliche Inferenz angewendet, um die Bewegungsschätzung zwischen zwei Zeitschritten zu zeigen. In Abbildung 8.1 sind zwei beispielhafte Trajektorien aus dem KITTI-Datensatz in Draufsicht gezeigt. Der mittlere Rotationsfehler über die ersten 11 Sequenzen des KITTI-Datensatzes ist in Abbildung 8.2 gezeigt. Die hier vorgestellte Methodik wird mit dem durch RANSAC robustifizierten Fünf-Punkt-Algorithmus von Níster et al. [33] verglichen.

Die wohl größte Stärke der hier vorgestellten Methode ist die Möglichkeit, ein Multikamerasystem verwenden zu können. So kann die Rundumsicht zur Bewegungsschätzung verwendet werden, was die Genauigkeit und Robustheit der Trajektorienschätzung erhöht, da so mehr Messungen aus verschiedenen Blickwinkeln verwendet werden können. Um dies zu zeigen, wurde die hier vorgestellte Methodik auf einem anspruchsvollen Datensatz evaluiert, welcher in Karlsruhe aufgenommen wurde. Dieser beinhaltet mehrspurige breite Straßen sowie enge Straßen und Innenstadtszenen. Es wurden vier Kameras mit einem Öffnungswinkel von $110°$ verwendet, welche gleichmäßig um das Versuchsfahrzeug *BerthaOne* montiert sind, wodurch eine Rundumsicht entsteht (Abbildung 8.5). Die Skale wurde aus dem Raddrehzahlmesser des Fahrzeugs ermittelt. Ein aufwendiger Visual-



(a) KITTI Sequenz 00 (b) KITTI Sequenz 10

Abbildung 8.1.: Zwei beispielhafte Trajektorien der Multikamerabewegungsschätzung von Frame zu Frame (*MOMO*) des KITTI-Datensatzes in der Draufsicht. Es wurden lediglich 100–300 Punktkorrespondenzen verwendet. Die Skale wurde von der Grundwahrheit übernommen. Während der Fünf-Punkt-Algorithmus (blau gepunktet) den zurückgelegten Pfad nicht korrekt schätzen kann, ist die hier vorgestellte Methodik in der Lage, eine sinnvolle Bewegung zu schätzen (schwarz).



(a) Rotationsfehler über zurückgelegtem Weg. (b) Rotationsfehler über Geschwindigkeit.

Abbildung 8.2.: Rotationsfehler der Multikamerabewegungsschätzung von Frame zu Frame (*MOMO*) und dem Fünf-Punkt-Algorithmus auf dem gesamten KITTI-Datensatz, aufgetragen über der zurückgelegten Distanz und der Geschwindigkeit. Obwohl 1800–2500 Punktkorrespondenzen für den Fünf-Punkt-Algorithmus verwendet wurden und für MOMO lediglich 100–300, ist das Ergebnis für MOMO erheblich besser. Insbesondere für hohe Geschwindigkeiten wird dies deutlich. Dies ist auf die Verwendung des Bewegungsmodells für MOMO zurückzuführen.

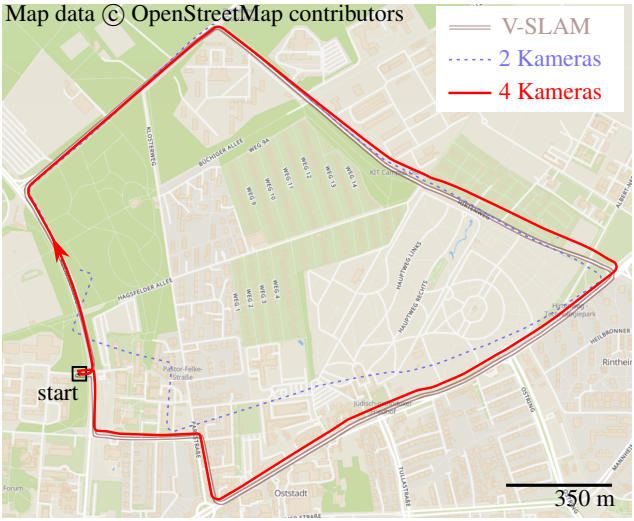
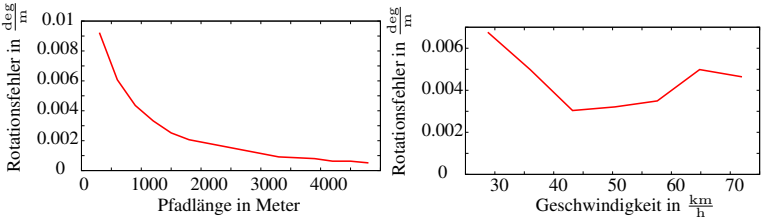


Abbildung 8.3.: Trajektorie, generiert durch Akkumulation der Multikamerabewegungsschätzung von Frame zu Frame. Die Sequenz umfasst eine Strecke von 5.1 km Länge und es wurden vier Kameras benutzt. Die Skale wurde aus dem Raddrehzahlmesser des Fahrzeugs extrahiert. Ein aufwendiger visueller SLAM mit Kreisschlussdetektion wird als Grundwahrheit verwendet (VSLAM, braune Doppellinie). Die geschätzte Trajektorie (durchgezogen rot) ist sehr nah zur Grundwahrheit, obwohl diese nur zwischen zwei aufeinander folgenden Zeitpunkten geschätzt ist und sich Fehler darum schnell akkumulieren. Im Vergleich zur Trajektorie, welche nur aus der hinteren Kamera und der linken Seitenkamera geschätzt ist (gepunktet blau), wird der Vorteil eines Aufbaus mit Rundumsicht deutlich.



(a) Rotationsfehler über Pfadlänge aufgetragen.

(b) Rotationsfehler über Geschwindigkeit aufgetragen.

Abbildung 8.4.: Fehlerdiagramme der Multikamerabewegungsschätzung mit vier Kameras von Frame zu Frame.

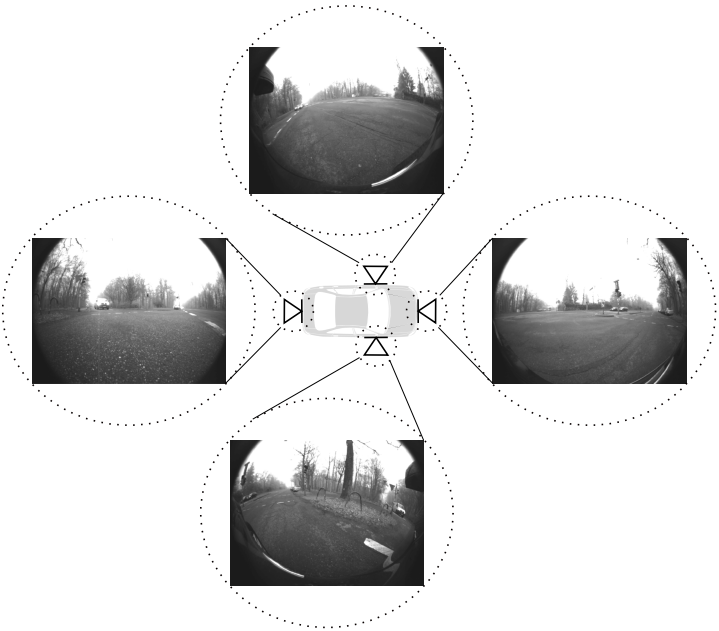


Abbildung 8.5.: Beispieldaten des Versuchsträgers *BerthaOne*. Vier Kameras mit Öffnungswinkel 110° sind um das Fahrzeug herum montiert.

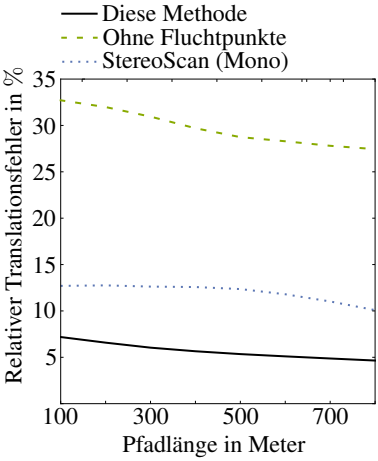
SLAM mit Kreisschlussdetektion von Sons et al. [18] wird als Grundwahrheit verwendet. Sie ist als Kartierungsalgorithmus konzipiert und nutzt daher zehn mal mehr Punktkorrespondenzen als MOMO und zeitliche Inferenz. Die Ergebnisse sind in Abbildung 8.3 und Abbildung 8.4 zu sehen. Obwohl dieser Datensatz mit Geschwindigkeiten von $0 - 72\ \frac{\mathrm{km}}{\mathrm{h}}$ und Blendung durch tiefstehende Sonne sehr herausfordernd ist, erreicht die Bewegungsschätzung auch ohne zeitliche Inferenz sehr hohe Genauigkeit.

8.1.2. Ergebnisse der Skalenschätzung mit Fluchtpunkten

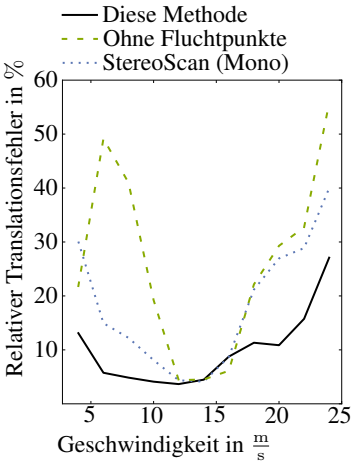
Die Evaluation der in Abschnitt 6.1.1 vorgestellten Methode wird auf dem KITTI-Datensatz durchgeführt. Um den Nutzen der Fluchtpunktschätzung zu bewerten, wird der Algorithmus mit und ohne Fluchtpunktschätzung evaluiert und verglichen. Der Algorithmus ohne Fluchtpunkte ist im Wesentlichen identisch zur monokularen Variante von StereoScan [17].

Die Ergebnisse können Abbildung 8.6 und Abbildung 8.7 entnommen werden. Für Geschwindigkeiten zwischen $6\frac{\mathrm{m}}{\mathrm{s}}$ und $14\frac{\mathrm{m}}{\mathrm{s}}$ reduziert die Nutzung der Fluchtpunkte den mittleren Translationsfehler stark auf 6% bis

4%. Im Vergleich zur Referenzmethode *StereoScan* kann der mittlere Translationsfehler somit um 30% reduziert werden. In Bereichen mit reichhaltiger Struktur für die Fluchtpunkte kann der Fehler um mehr als 50%, auf Werte zwischen 3.5% und 2%, reduziert werden (Abbildung 8.7). Folglich sind Fluchtpunkte sehr nützlich in urbaner Umgebung.
Wird die Geschwindigkeit größer als $14\frac{\mathrm{m}}{\mathrm{s}}$, divergiert die Skale. Der Grund hierfür liegt in der Konfiguration der Methode, welche für innerstädtische Szenarien entwickelt wurde. Bei hoher Geschwindigkeit unterliegt der Nahbereich des Kamerabilds Bewegungsunschärfe. Dadurch wird die in Abschnitt 3.5 beschriebene Merkmalsextraktion schwierig und es können nur wenige nahe Punkte mit ausreichender Genauigkeit rekonstruiert werden.

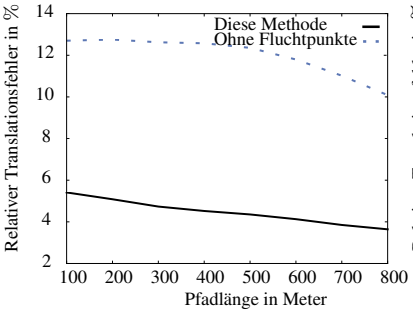


(a) Mittlerer Translationsfehler über Pfadlänge.

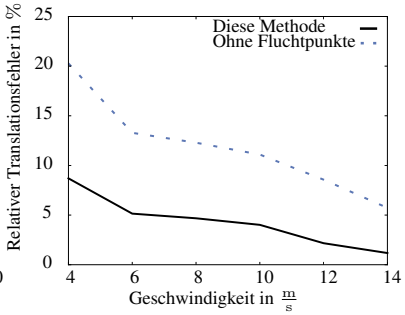


(b) Mittlerer Translationsfehler über Geschwindigkeit.

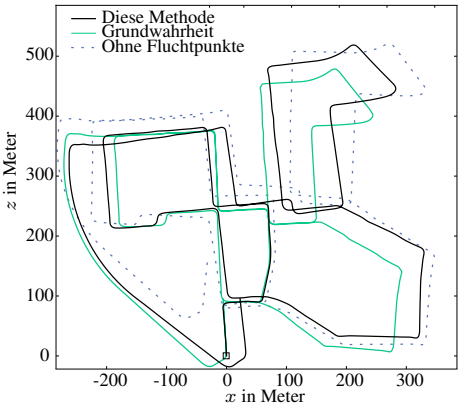
Abbildung 8.6.: Fehler der Frame-zu-Frame-Skalenschätzung mit Fluchtpunkten auf KITTI. Zwischen Geschwindigkeiten von $6\frac{\mathrm{m}}{\mathrm{s}}$ und $14\frac{\mathrm{m}}{\mathrm{s}}$ befindet sich der mittlere Translationsfehler bei ca. 5%. Bei kleineren Geschwindigkeiten ist ein Anstieg dieses Fehlers zu beobachten. Dies ist auf höhere Fehler bei der Rekonstruktion zurückzuführen, da der optische Fluss im Bild kleiner wird und somit die Rekonstruktion mit Triangulation einer Singularität zustrebt.



(a) Translationsfehler über Pfadlänge. Die Methode konvergiert zu einem Translationsfehler von 4%.



(b) Translationsfehler über Geschwindigkeit. Die Integration der Fluchtpunkte erhöht die Genauigkeit auf Fehler zwischen 1% und 5% bei Geschwindigkeiten zwischen $6\,\frac{\mathrm{m}}{\mathrm{s}}$ und $14\,\frac{\mathrm{m}}{\mathrm{s}}$.



(c) Trajektorie geschätzt von der hier vorgestellten Methode im Vergleich zur Grundwahrheit des KITTI-Datensatzes.

Abbildung 8.7.: Frame-zu-Frame-Skalenschätzung mit Fluchtpunkten: Translationsfehler über Pfadlänge und Geschwindigkeit sowie geschätzte Pfade für Sequenz 00 des KITTI-Datensatzes, in welcher urbane Szenerie vorherrscht.

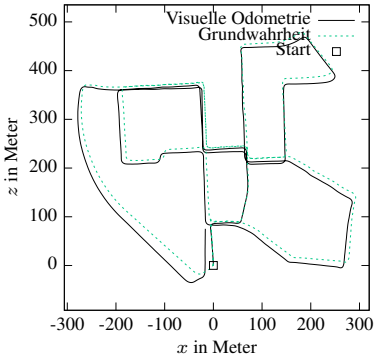
8.2. Methoden mit zeitlicher Inferenz

In diesem Abschnitt werden die Bewegungsschätzungsverfahren evaluiert, welche zeitliche Inferenz nutzen, um den in Abschnitt 5 gezeigten Graphen mithilfe eines Optimierungsproblems zu lösen. Hierbei werden zwei monokulare Methoden vorgestellt, welche die Skale aus dem Abstand zur Bodenoberfläche gewinnen, sowie zwei Verfahren, welche einen LIDAR-Sensor nutzen, um die Skaleninformation zu erhalten. Um die Notwendigkeit von zeitlicher Inferenz zu untersuchen, wird die Bewegungsschätzung mit LIDAR zudem zu einer Frame-zu-Frame-Methode mit Tiefe aus LIDAR verglichen.

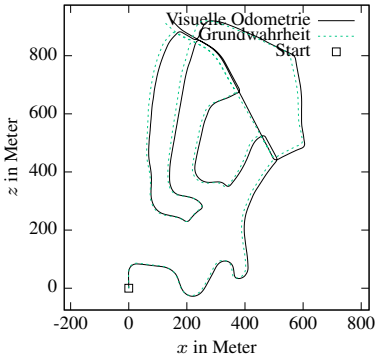
8.2.1. Monokulare Skalenschätzung durch A-Posteriori-Bodenebenenschätzung

Zuerst wird die Skalenschätzung aus der Bodenoberfläche isoliert betrachtet. Wie in Abschnitt 6.1.2.1 genauer beschrieben, wird zuerst mithilfe der zeitlichen Inferenz die Umgebung rekonstruiert und anschließend eine Ebene an die so entstehende Punktmenge der Landmarken angepasst. Daher wird diese Methode als A-Posteriori-Bodenebenenschätzung bezeichnet. Zuletzt verfolgt und glättet das Skalenschätzungsmodul die Ebenenschätzung. Die Evaluation auf dem KITTI-Datensatz zeigt, dass mit dieser einfachen Modellierung der Bodenoberfläche als Ebene schon gute Ergebnisse erzielt werden können, selbst wenn, wie in Sequenz 02, eine starke Krümmung der Bodenoberfläche vorhanden ist. Auf dem KITTI-Benchmark erreicht diese Methodik einen Translationsfehler von 2.95% und einen Rotationsfehler von $0.004\frac{\mathrm{deg}}{m}$. Beispiele sind in Abbildung 8.8 gezeigt. Für die kurzzeitige Berechnung der Skale, zum Beispiel für das in Abschnitt 9 vorgestellte Abbiegeassistenzsystem, ist diese Methode daher sehr gut geeignet.

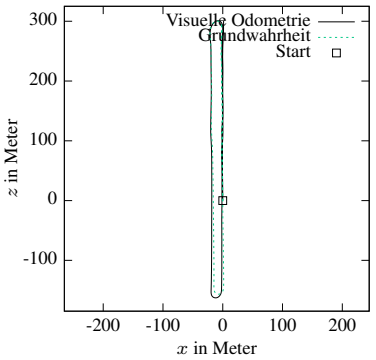
Obwohl die Umsetzung als eigenständiges Modul für die Programmstruktur von Vorteil ist, sind dafür algorithmische Limitierungen notwendig. So kann die Skaleninformation nicht mit der Rekonstruktion gekoppelt werden und kein konsistentes Umgebungsmodell aufgebaut werden. Des Weiteren sind eine Vielzahl von Messungen auf der Bodenebene notwendig, um eine stabile Skale zu schätzen. Gibt es zu wenige Messungen auf dem Boden oder sind diese zu ungenau, kann die Skale springen, wie in Abbildung 8.9 dargestellt. Zudem ist für eine dauerhafte Berechnung der Trajektorie die Annahme der Bodenoberfläche als Ebene über mehrere Keyframes hinweg sehr stark und führt zu Ungenauigkeit.



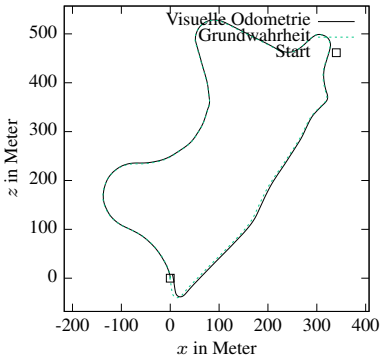
(a) Sequenz 00



(b) Sequenz 02



(c) Sequenz 06



(d) Sequenz 09

Abbildung 8.8.: Beispiele aus dem KITTI-Benchmark für die A-Posteriori-Bodenoberflächenschätzung. Dargestellt sind Fahrzeugtrajektorien über mehrere Kilometer hinweg in Draufsicht.

8.2.2. Monokulare Skalenschätzung durch integrierte Bodenebenenschätzung

Die in Abschnitt 8.2.1 dargestellten Nachteile der Bodenoberflächenschätzung als eigenständiges Modul können durch die in Abschnitt 6.1.2.2 beschriebene Integration der Bodenoberfläche in das Schätzproblem ausgeglichen werden. Hierbei wird die Bodenoberfläche nur lokal, um jeden Keyframe, als Ebene approximiert. Durch eine Bestrafung der Bodeno-

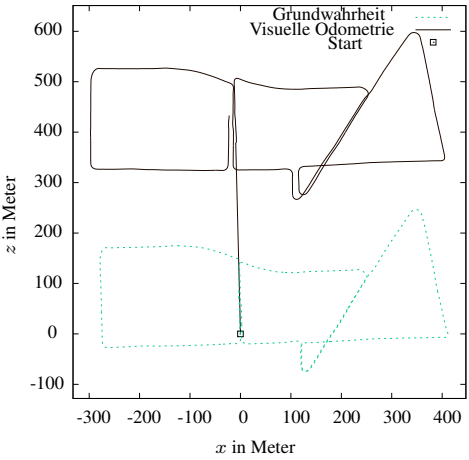
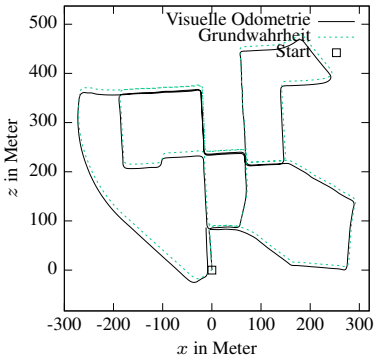
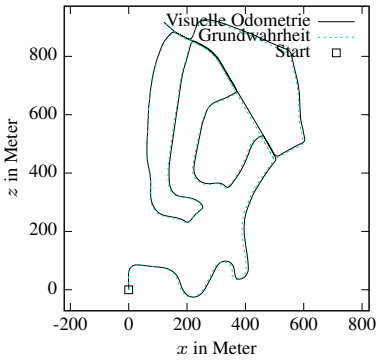


Abbildung 8.9.: Trajektorie aus dem KITTI-Datensatz Sequenz 13 mit A-Posteriori-Bodenoberflächenschätzung. Hier ist die Bodenebene nur durch einen kleinen Korridor auf der Straße gegeben. Zusätzlich verdecken entgegenkommende Fahrzeuge kurzzeitig den Boden. Somit reichen wenige falsche Bodenabstandsschätzungen aus, damit eine falsche Geschwindigkeit des Fahrzeugs prädiziert wird und so das Fahrzeug um ca. 350 m deplatziert ist.

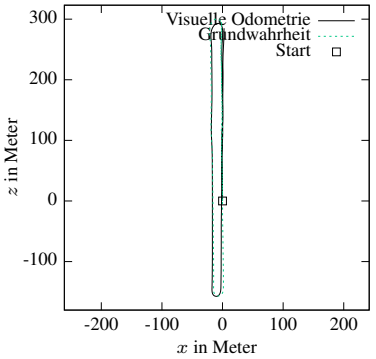
berflächenrichtungen und der Posenabweichung von der Ebene kann eine global konsistent skalierte Trajektorienschätzung erreicht werden. Dies erhöht die Genauigkeit und Robustheit der Schätzung. Auf dem KITTI-Datensatz konnte damit ein Translationsfehler von 1.11% sowie ein Rotationsfehler von $0.0023\frac{\deg}{m}$ erreicht werden und ist somit der A-Posteriori-Skalenschätzung deutlich überlegen. Beispiele sind in Abbildung 8.10 zu sehen.



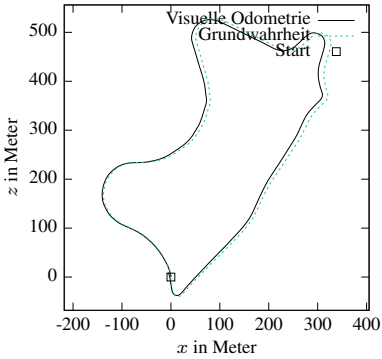
(a) Sequenz 00



(b) Sequenz 02



(c) Sequenz 06

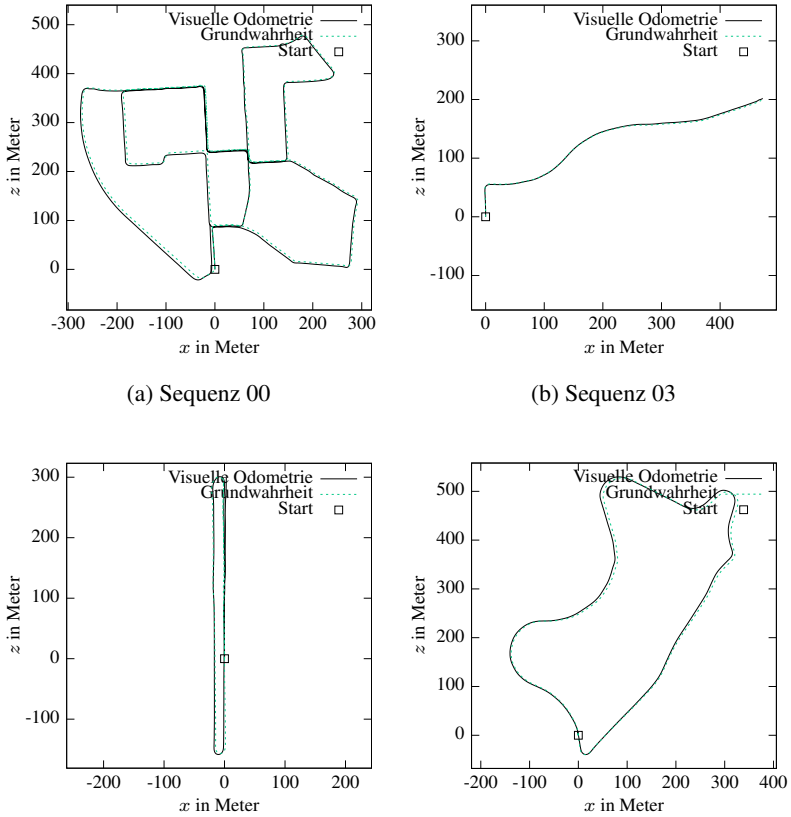


(d) Sequenz 09

Abbildung 8.10.: Beispiele aus dem KITTI-Benchmark für die integrierte Bodenoberflächenschätzung. Dargestellt sind Fahrzeugtrajektorien über mehrere Kilometer hinweg in Draufsicht.

8.2.3. Skalenschätzung durch LIDAR-Information

In Abbildung 8.11 und Abbildung 8.16 sind beispielhafte Trajektorien der Skalenschätzung durch LIDAR vorgestellt. Die durch diese Methode geschätzten Trajektorien sind von hoher Präzision mit sehr kleinem Drift über mehrere Kilometer, wobei keine Kreisschlussdetektion verwendet wurde. In Abbildung 8.12 ist eine weitere Beispiel-Trajektorie gezeigt, in welcher zusätzlich die berechneten Landmarken visualisiert sind.



(a) Sequenz 00

(b) Sequenz 03

(c) Sequenz 06

(d) Sequenz 09

Abbildung 8.11.: Beispielhafte Trajektorien aus dem Trainingsdatensatz des KITTI-Benchmark unter Verwendung der Tiefe aus LIDAR ohne Bodenoberflächenschätzung.

Vergleich mit Frame-zu-Frame-Bewegungsschätzung aus extrahierten LIDAR-Tiefenschätzungen. Die in Abschnitt 6.2.1 erläuterte Methodik wurde auf dem KITTI-Datensatz [16] evaluiert. Um die Notwendigkeit der rechenaufwändigen zeitlichen Inferenz zu prüfen, wurde eine Frame-zu-Frame-Schätzung anhand der extrahierten Tiefeninformationen umgesetzt und steht hier zum Vergleich. Die Frame-zu-Frame-Schätzung (genannt *LIVIODO*) verwendet ähnlich zu herkömmlichen Stereo-Visuelle-Odometrie-Algorithmen den Rückprojektionsfehler der Messungen mit Tiefe zwischen Bildern, um die Bewegung zu schätzen. Dabei werden jedoch keine Landmarken optimiert. Dieser Algorithmus wird in der Literatur Perspective-N-Point-Algorithmus [57] genannt.
Insbesondere für lange Trajektorien wird der durch die zeitliche Inferenz gewonnene Vorteil deutlich: der Drift wird im Vergleich zur Frame-zu-Frame-Schätzung stark reduziert.
Diese Ergebnisse für die Frame-zu-Frame-Schätzung und die gesamte Methode mit zeitlicher Inferenz sind auf dem KITTI-Benchmark veröffentlicht (genannt LIVIODO und LIMO, respektive). LIVIODO ist auf Rang 37[1] mit einem mittleren Translationsfehler von 1.22% und einem mittleren Rotationsfehler von $0.0042\frac{\text{deg}}{\text{m}}$ zu finden. LIMO hingegen belegt Rang 11[1] mit einem mittleren Translationsfehler von 0.86% und einem mittleren Rotationsfehler von $0.0022\frac{\text{deg}}{\text{m}}$. Während der Vorteil der zeitlichen Inferenz im Vergleich zu LIVIODO nur einen leichten Gewinn im Translationsfehler erzeugt, ist die Verbesserung im Rotationsfehler von fast 40% drastisch. Insbesondere für Anwendungen, in welchen ein kleiner End-Punkt-Fehler wichtig ist, ist der Nutzen der zeitlichen Inferenz somit eklatant.

[1] Stand 28.8.2018.

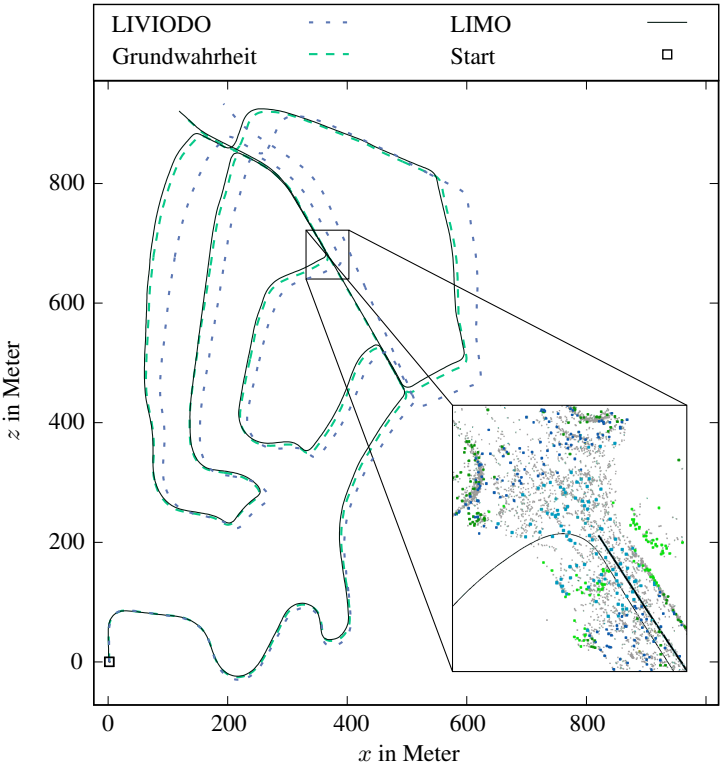
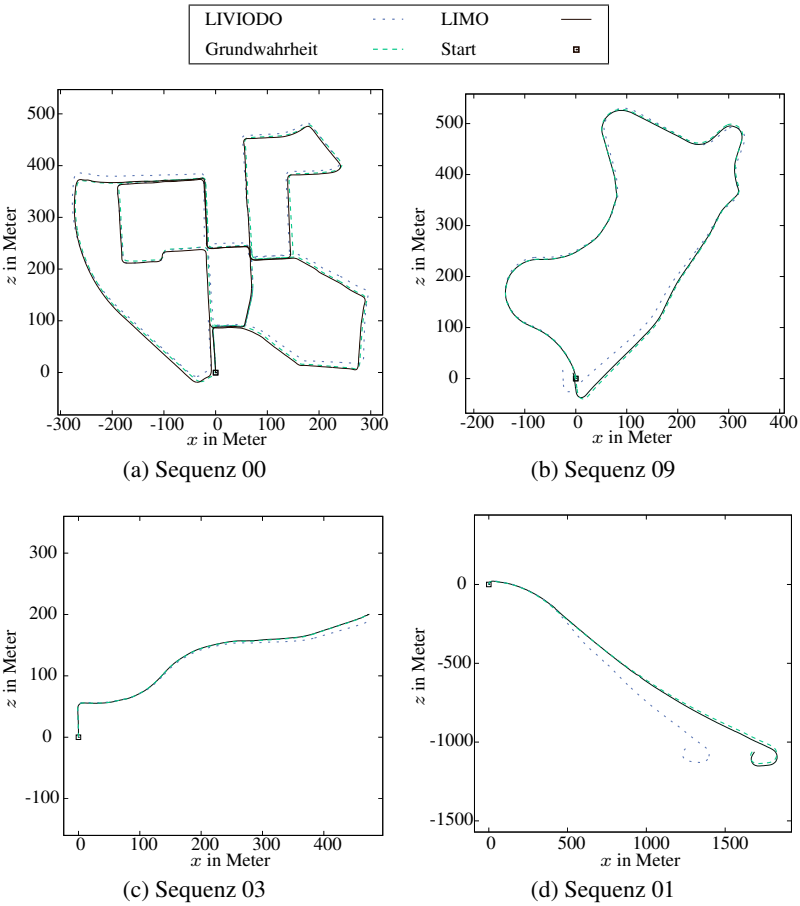


Abbildung 8.12.: Trajektorie mit Frame-zu-Frame-Schätzung mit dem Perspective-N-Point-Algorithmus (LIVIODO; blau, gestrichelt) und der Schätzung mit zeitlicher Inferenz (LIMO; schwarz, durchgezogen) der Sequenz 02 des KITTI-Datensatzes. Die geschätzte Trajektorie kann die Referenztrajektorie (zyan, gestrichelt) mit sehr hoher Genauigkeit nachfahren, ohne dabei Kreisschluss-Detektion zu verwenden. Dies ist ein Indikator für den geringen Drift dieser Methodik. In dem vergrößerten Fenster sind die geschätzten Landmarken zu sehen. Hierbei sind monokular beobachtete Landmarken (nah: zyan, mittelweit entfernt: blau) und Landmarken mit Tiefe aus dem LIDAR gezeigt (nah: hellgrün, mittelweit entfernt: dunkelgrün). Zurückgewiesene Landmarken sind grau gezeigt.



(a) Sequenz 00

(b) Sequenz 09

(c) Sequenz 03

(d) Sequenz 01

Abbildung 8.13.: Beispielhafte Trajektorien aus dem Trainingsdatensatz des KITTI-Benchmark für die Frame-zu-Frame-Schätzung mit dem Perspective-N-Point-Algorithmus (LIVIODO; blau, gestrichelt) und der Schätzung mit zeitlicher Inferenz (LIMO; schwarz, durchgezogen). Für Visuelle Odometrie auf kurzen Strecken und in urbaner Umgebung mit reichhaltiger Struktur wie zum Beispiel Sequenz 00 ist die Frame-zu-Frame-Schätzung gut geeignet. Für längere Strecken wird jedoch der Zugewinn in der Driftreduktion durch die zeitliche Inferenz sichtbar. Auch in Sequenzen mit weiter, offener Szenerie und weniger Struktur kann der Inferenzblock (siehe Aktivitätsdiagramm in Abbildung 2.1) ungenaue Tiefenschätzungen durch viele Messungen kompensieren und somit die Schätzung stark verbessern. Insbesondere auf der Autobahn (Sequenz 01), wo hohe Geschwindigkeiten hinzukommen, wird die zeitliche Inferenz wichtig für eine genaue Schätzung.

8.2.4. Kombinierte Schätzung der Skale durch LIDAR und Bodenebenenannahme

Durch die in Abschnitt 6.1.2.2 beschriebene Integration der Bodenebenenschätzung in das Optimierungsproblem kann die Skaleninformation einfach als zusätzlicher Kostenfunktor in das Optimierungsproblem eingefügt werden. Somit können sowohl Kostenfunktoren für Tiefenabweichungen aus dem LIDAR, siehe Gleichung 6.11, als auch solche für die Bodenebene verwendet werden, siehe Gleichung 6.8. Um die Annahme einer konstanten Höhe über Grund abzuschwächen, wird der Abstand des Fahrzeugknotens zur Bodenebene im Problem mitoptimiert. Beispiel-Trajektorien sind in Abbildung 8.13 dargestellt. Diese sind sehr ähnlich zur Trajektorienschätzung ohne Bodenoberfläche und sind daher im Appendix zu finden. Die Auswertung und der Vergleich mit den anderen vorgestellten Methoden ist Abschnitt 8.4 zu entnehmen.

8.3. Rechenzeiten

In diesem Abschnitt werden die in dieser Arbeit erarbeiteten Algorithmen im Hinblick auf die benötigte Rechenzeit bewertet. Hierzu werden die Methoden auf repräsentativen Datensätzen ausgeführt und ihre Rechenzeiten analysiert. Histogramme mit den einzelnen Zeiten der Hauptkomponenten sind in Abbildung 8.14 und Abbildung 8.15 zu finden. Mittlere Rechenzeiten sind Tabelle 8.1 und Tabelle 8.2 zu entnehmen. Alle Experimente sind auf einem *Intel Core i7 - 4771* mit vier Kernen mit jeweils $3{,}5\,\mathrm{Ghz}$ Taktfrequenz und vier virtuellen Kernen ausgeführt.
Zuerst ist zu bemerken, dass für die Bildfrequenz von $10\,\mathrm{Hz}$ die Algorithmen alle eingehenden Daten zur Laufzeit verarbeiten können. Der Unterschied der Methoden ist in der Latenz der Bewegungsschätzung unter der

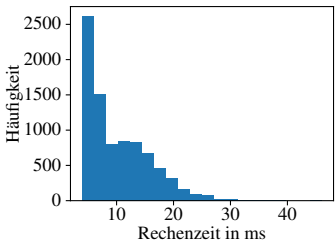
Methode	Median	Mittelwert	Standard-abweichung
Merkmalsextraktion	32	33	5
Tiefenextraktion	8	11	9
LIMO, volle Lösung	220	217	69
LIMO, Frame-Angleich	4	4	2
Mono Boden integriert, volle Lösung	163	171	75
Mono Boden integriert, Frame-Angleich	6	7	4

Tabelle 8.1.: Rechenzeiten in Millisekunden der Algorithmen mit zeitlicher Inferenz (Abschnitte 5 und 6.1).

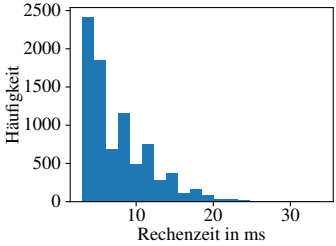
Methode	Median	Mittelwert	Standard-abweichung
Merkmalsextraktion Links/Rechts	9	11	5
Merkmalsextraktion Vorne/Hinten	6	8	4
MOMO 2D	16	20	13
MOMO 3D	44	48	19

Tabelle 8.2.: Rechenzeiten in Millisekunden der Frame-zu-Frame-Bewegungsschätzung (Abschnitt 4). Die Bewegungsschätzung in zwei Dimensionen (MOMO 2D) ist durch die geringere Anzahl an Parametern deutlich schneller als die drei dimensionale Variante (MOMO 3D).

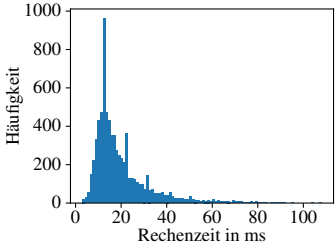
möglichen Framerate zu sehen.
Die Multikamerabewegungsschätzung hat eine mittlere Latenz von ca. 31 ms. Dies setzt sich aus 11 ms für die Merkmalsassoziation sowie 20 ms für die Bewegungsschätzung zusammen. Hierbei wird die Merkmalsassoziation der vier verwendeten Kameras parallelisiert und auch das Optimierungsproblem der Bewegungsschätzung nutzt die Parallelisierbarkeit der Kostenfunktionsevaluation aus. Somit ist eine maximale Bildfrequenz von 30 Hz erreichbar.
Dem gegenüber ist die Bewegungsschätzung mit zeitlicher Inferenz rechenaufwändiger. Für jeden Frame wird die Merkmalsassoziation einer Kamera sowie die Tiefenschätzung benötigt. Dies benötigt durchschnittlich 33 ms und 11 ms. Die höhere Rechenzeit für die Merkmalsextraktion von 33 ms ist hierbei auf eine höhere Anzahl extrahierter Merkmale zurückzuführen. Parallel dazu wird mithilfe der Grafikkarte das Bild semantisch segmentiert, was ca. 100 ms benötigt. Die Bewegungsschätzung selbst variiert in ihrer Geschwindigkeit abhängig davon, ob das volle Problem gelöst wird, mit allen Landmarken- und Posenparametern, oder nur der Frame-Angleich ohne Landmarkenoptimierung durchgeführt wird. Ohne Landmarkenoptimierung ist die Schätzung mit 4 ms schnell, werden Landmarken optimiert, ist die Optimierung langsam mit ca. 217 ms. Die volle Lösung wird hierbei nur berechnet, wenn neue Keyframes hinzugefügt werden, also maximal alle 300 ms, wie in Abschnitt 5.3 beschrieben. Der schnelle Frame-Angleich kann für die restlichen Frames diese hohe Rechenzeit kompensieren. Damit kann zwar verhindert werden, dass Daten unverarbeitet verworfen werden müssen, jedoch ist die Latenz mit minimal 104 ms für den Frame-Angleich und maximal 317 ms für die volle Lösung bedeutend. Die Latenz der monokularen Methode mit Skaleninformation aus der Höhe über Grund der Kamera ist mit 106 ms und maximal 271 ms ein wenig geringer.
Zusammenfassend ist die Frame-zu-Frame-Multikamerabewegungsschät-



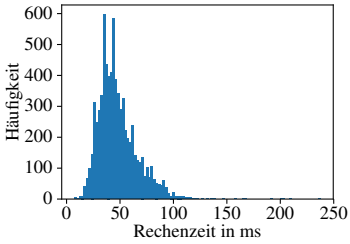
(a) Merkmalsextraktion links/rechts



(b) Merkmalsextraktion vorne/hinten



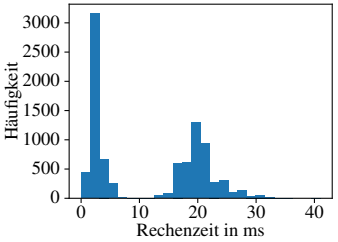
(c) MOMO zweidimensional



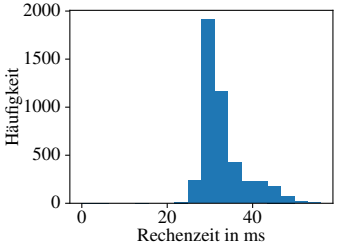
(d) MOMO dreidimensional

Abbildung 8.14.: Histogramme der Rechenzeiten in Millisekunden der Multikamerabewegungsschätzung. Die Merkmalsextraktion für die Kameras links und rechts sowie vorne und hinten sind quasi identisch.

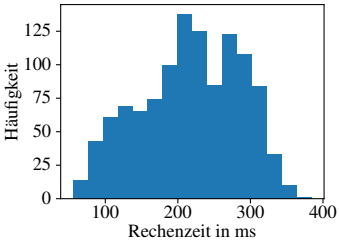
zung vorzuziehen, falls die Skale aus dem Odometer bekannt ist und geringe Latenz benötigt wird. Ist jedoch die Langzeitgenauigkeit der Bewegungsschätzung wichtig oder ist nur eine Kamera vorhanden, so sind die hier vorgestellten Methoden mit zeitlicher Inferenz mit und ohne LIDAR deutlich besser geeignet.



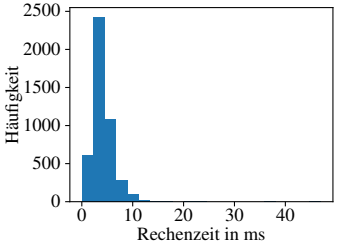
(a) Tiefenextraktion



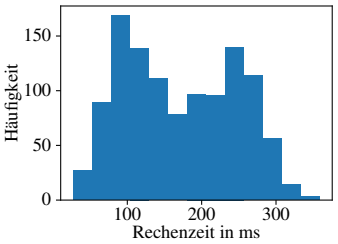
(b) Merkmalsextraktion



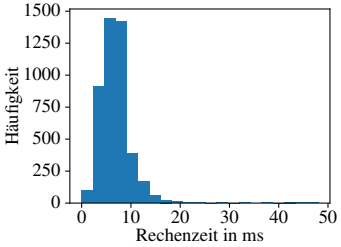
(c) LIMO, volle Lösung



(d) LIMO, Frame-Angleich



(e) Mono integriert, volle Lösung



(f) Mono integriert, Frame-Angleich

Abbildung 8.15.: Histogramme der Rechenzeiten in Millisekunden der Methoden mit zeitlicher Inferenz. In seltenen Fällen kann die Optimierung sehr lange dauern, da das Minimum nur schwach ausgeprägt ist. Dies kann für eine Bildfrequenz von 10 Hz jedoch problemlos wieder aufgeholt werden. Für eine hohe Framerate der Kameras kann die Optimierungszeit hart begrenzt werden und gegebenenfalls die Optimierung frühzeitig abgebrochen werden.

8.4. Fazit

In Abbildung 8.16 sind alle vier in dieser Arbeit vorgestellten Algorithmen zur Skalenschätzung gegenübergestellt. Deren Ergebnisse auf den Trainingssequenzen des KITTI-Benchmark sind in Tabelle 8.3 und in Tabelle 8.4 gezeigt.

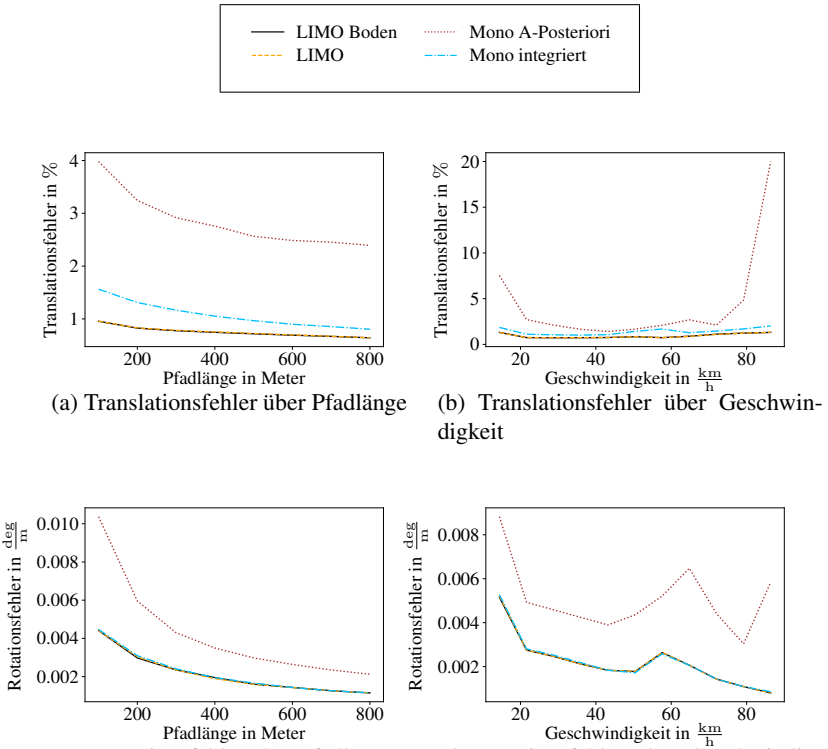
Methode	Trans.fehler in %	Rot.fehler in $\frac{\text{deg}}{\text{m}}$
LIMO mit Boden integriert	0.765	0.0022
LIMO	0.769	0.0022
Mono mit Boden integriert	1.11	0.0023
Mono A-Posteriori-Skalenschätzung	2.94	0.0046

Tabelle 8.3.: Mittlerer Translations- und Rotationsfehler aller vorgestellten Methoden auf Sequenz 00 bis 10 des KITTI-Datensatz.

Methode	Trans.fehler in %	Rot.fehler in $\frac{\text{deg}}{\text{m}}$
LIMO mit Boden integriert	0.84	0.0022
LIMO	0.86	0.0022
Mono mit Boden integriert	1.28	0.0022
Mono A-Posteriori-Skalenschätzung	2.94	0.0046

Tabelle 8.4.: Mittlerer Translations- und Rotationsfehler aller vorgestellten Methoden auf Sequenz 11 bis 21 des KITTI-Datensatzes.

Hieraus wird ersichtlich, dass die A-Posteriori-Bodenschätzung in jeder Hinsicht deutlich schlechter abschneidet als die Vergleichsalgorithmen. Das ist auf starke Modellannahmen sowie die Notwendigkeit, viele Messungen auf dem Boden extrahieren zu müssen, zurückzuführen. Am besten schneiden die Bewegungsschätzungen mit LIDAR ab. Das Hinzufügen der Bodenebenenannahme zum Optimierungsproblem ergibt im Mittel nur eine kleine Verbesserung. Hieraus wird deutlich, dass die Approximation der lokalen Umgebung als Ebene kaum einen Vorteil bietet, da die Qualität der Tiefenmessungen aus dem LIDAR sehr gut ist. Die Tiefenmessungen dominieren somit das Schätzproblem. Für Aufbauten, in welchen nur wenige Tiefenmessungen vorhanden sind oder die Qualität der Tiefenmessungen schlechter ist, bietet die Kombination aus Bodenebenenschätzung und LIDAR-Tiefenschätzung jedoch großes Potential. Bemerkenswert ist das sehr gute Abschneiden der monokularen Bewegungsschätzung mit integrierter



(a) Translationsfehler über Pfadlänge

(b) Translationsfehler über Geschwindigkeit

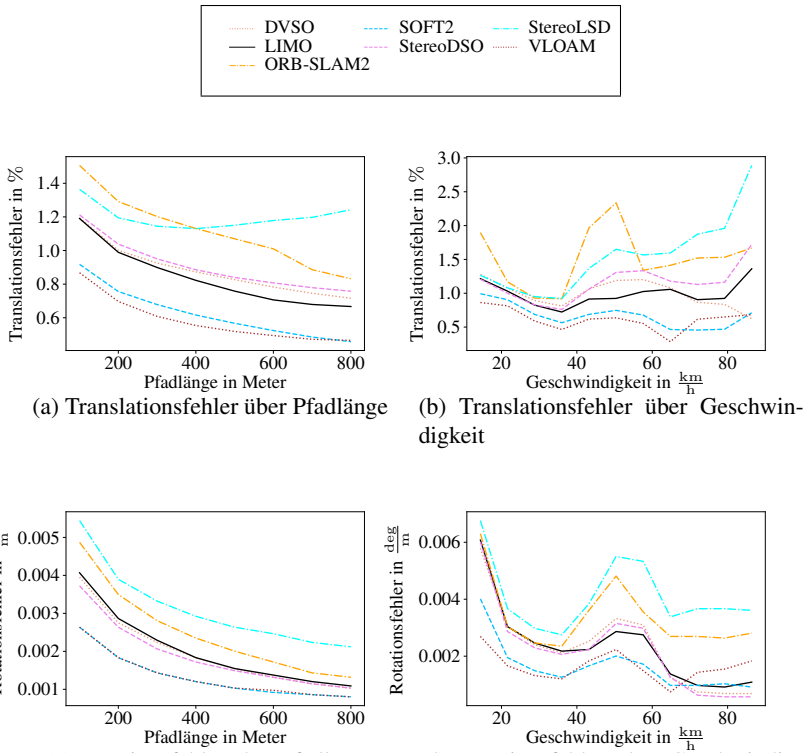
(c) Rotationsfehler über Pfadlänge

(d) Rotationsfehler über Geschwindigkeit

Abbildung 8.16.: Mittlere Fehler aller hier vorgestellten Methoden auf dem Trainingsdatensatz des KITTI-Datensatzes (Sequenz 00 bis 10). Es ist ersichtlich, dass die A-Posteriori-Bodenoberflächenschätzung deutlich schlechter abschneidet als die anderen Verfahren. Der Unterschied von LIDAR und Kamera ohne Bodenoberflächenschätzung (*LIMO*) zu LIDAR, Kamera und Bodenoberflächenschätzung (*LIMO Boden*) ist vernachlässigbar und im Schaubild nicht zu unterscheiden. In Abbildung A.1 sind darum die gleichen Ergebnisse vorgestellt, jedoch ohne die A-Posteriori-Bodenoberflächenschätzung. Auch ohne LIDAR kann mit der in das Optimierungsproblem integrierten Bodenoberflächenschätzung ein sehr gutes Ergebnis erzielt werden (*Mono integriert*).

Bodenebene. Ohne den LIDAR kann hiermit schon ein sehr guter mittlerer Translationsfehler von 1.11% und 1.28% erreicht werden.
Der KITTI-Benchmark ermöglicht einen direkten, objektiven und internationalen Vergleich von Odometrie-Schätzungsalgorithmen. Die hier vorgestellte Variante mit Tiefenschätzung aus LIDAR wird unter dem Namen *LIMO* veröffentlicht und mit anderen populären Algorithmen verglichen. LIMO ohne Bodenebene erzielt dabei Platz 11[2] und die Variante mit Bodenebenenschätzung erreicht Platz 9, bezüglich des Translationsfehlers. Im Bezug auf den Rotationsfehler teilen sich die Methoden den 8. Platz. Zieht man die sehr hohe Beliebtheit des KITTI-Benchmarks mit 85 Einreichungen in Betracht, ist das ein hervorragendes Ergebnis. Dies macht LIMO zur besten visuellen Odometrie mit veröffentlichter Codebasis und lässt populäre Algorithmen wie ORB-SLAM [21], LSD-SLAM [58] und DSO [20] hinter sich. Es ist die zweitbeste veröffentlichte LIDAR-Kamera-Methode und die beste, welche keine LIDAR-SLAM-basierten Verfeinerungsschritte verwendet. Unter den LIDAR-Methoden belegt sie Platz 6, wobei die ersten drei Methoden Kreisschlussdetektion nutzen. Ein quantitativer Vergleich mit anderen populären Algorithmen ist in Abbildung 8.17 gegeben.
Als qualitatives Ergebnis ist in Abbildung 8.18 eine Umgebungsrekonstruktion mithilfe der geschätzten Posen durch die Methode mit zeitlicher Inferenz und Skalenschätzung aus LIDAR- und Bodenoberfläche gezeigt.

[2] Stand 28.8.2018.



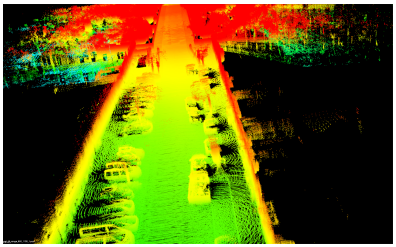
(a) Translationsfehler über Pfadlänge

(b) Translationsfehler über Geschwindigkeit

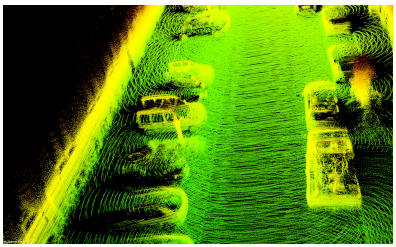
(c) Rotationsfehler über Pfadlänge

(d) Rotationsfehler über Geschwindigkeit

Abbildung 8.17.: Mittlere Fehler auf den öffentlichen Evaluationssequenzen des KITTI-Datensatzes im Vergleich zu anderen veröffentlichten Methoden. LIMO schneidet sehr gut ab und lässt populäre feature-basierte Stereo-Algorithmen wie ORB-SLAM2 hinter sich. Auch semi-direkte Stereo-Methoden wie StereoLSD, StereoDSO und DVSO (eine DSO-Variante mit maschinell gelernter Tiefe) sind hinter LIMO platziert. VLOAM nutzt Kameradaten, um einen LIDAR-ICP vorzukonditionieren, und ist seit vielen Jahren Spitzenreiter. Sowohl VLOAM als auch SOFT2 (ein Stereo-SLAM) nutzen Kreisschlussdetektion. Dies ist zwar sehr nützlich für Kartierung und den KITTI-Datensatz, hat jedoch nur wenig Nutzen für echte Visuelle Odometrie auf autonomen Fahrzeugen, da typischerweise Pfade nur einmal befahren werden.



(a) Weitläufige Szene



(b) Szenenauschnitt

(c) Szene mit überlagerter Farbe

Abbildung 8.18.: Um die Genauigkeit der Bewegungsschätzung mit Kamera und LIDAR zu demonstrieren, werden die Posen, welche durch die Methode mit zeitlicher Inferenz, LIDAR und Bodenoberfläche geschätzt wurden, verwendet, um die durch den LIDAR erhaltenen Punktmengen über die Zeit zu akkumulieren. Damit können die umgebende Infrastruktur, aber auch parkende Fahrzeuge mit sehr hoher Genauigkeit rekonstruiert werden, wie die scharfen Konturen der Fahrzeuge zeigen. Die Farbkodierung der Punktmengen in Abbildungen 8.18a und 8.18b verläuft entlang der Hochachse. In Abbildung 8.18c wird eine Szene mit überlagerter Farbinformation aus der Kamera gezeigt.

Kapitel 9

Anwendungsfall ABALID

In diesem Kapitel wird eine konkrete Anwendung der in dieser Arbeit beschriebenen Algorithmen zur Eigenbewegungsschätzung von Fahrzeugen gezeigt. Hierzu wurde im Rahmen eines Projektes des Bundesministeriums für Bildung und Forschung (BMBF) ein Fahrerassistenzsystem für LKW entwickelt, welches vor Unfällen mit Fahrradfahrern in Rechtsabbiegesituationen im innerstädtischen Verkehr warnen soll [59]. Für dessen Umsetzung müssen die Trajektorie des LKW sowie die Trajektorie des Fahrradfahrers bekannt sein, um Kollisionen vorhersagen zu können. Somit stellt dieses Projekt einen Anwendungsfall von großer Aktualität und öffentlichem Interesse der hier erarbeiteten Methoden dar. In diesem Kapitel wird die Problemstellung des Projekts sowie das Gesamtsystem des erarbeiteten Lösungsvorschlags beschrieben. Abschließend wird ein Einblick in die Probandenstudie gegeben, welche für den Projektabschluss durchgeführt wurde.

9.1. Einführung Projekt ABALID

Die Nutzung des Fahrrads als innerstädtisches Transportmittel ist attraktiv für Städte und deren Bewohner [60]. Fahrradfahrer setzen sich jedoch im gemischten Verkehr Gefahren aus, da sie in Unfällen häufig die Leidtragenden sind. Die kritischsten Szenarien sind dabei Unfälle mit Lastkraftwägen (LKW), da Radfahrer hierbei in vielen Fällen schwere Verletzungen davon tragen. Oft enden solche Unfälle tödlich [61]. Auffällig häufig geschehen diese Unfälle im Kreuzungsbereich bei Rechtsabbiegemanövern eines LKW. Dies ist direkt ersichtlich aus deren baubedingt großem toten Win-

kel. Radfahrer werden darum schlicht übersehen. Seit 2007 müssen LKW in der EU mit zusätzlichen Weitwinkel- und Nahbereichsspiegeln ausgerüstet sein, welche den toten Winkel verkleinern. Somit muss der Fahrer zusätzlich zur Fahrtätigkeit bis zu 6 Spiegel gleichzeitig überwachen. Diese hohen kognitiven Anforderungen sind schwierig zu erfüllen und Fahrradfahrer werden weiterhin übersehen. Um die Anzahl solcher Unfälle zu reduzieren, sollte im Rahmen des Projekts *ABALID: Abbiegeassistent mit 3D-LIDAR-Sensorik* ein Assistenzsystem für LKW entwickelt werden, das den LKW-Fahrer bei der Umgebungsüberwachung unterstützt und dieser so auf sich anbahnende Unfälle reagieren kann. Das Ziel des Projektes war es, einen Prototypen für dieses Assistenzsystem zu entwickeln. Hierbei bestand ein Schwerpunkt darauf, mit Hilfe eines LIDAR-Sensors und einer Kamera die Fahrradfahrer zu erfassen und Unfälle prädizieren zu können. Bestand Kollisionsgefahr, wurde der LKW-Fahrer visuell gewarnt. Abschließend wurde das System experimentell in Form einer Probandenstudie evaluiert.

Zusammenfassend sollte Folgendes umgesetzt werden:

- Die Bestimmung der Bewegung des LKW.
- Die Erkennung von Fahrradfahrern im Kamerabild.
- Die Schätzung der Bewegung der erkannten Fahrradfahrer.

Fehlerhafte Informationen wie Falscherkennungen würden das Fahrverhalten stören und auf Dauer zur Abschaltung des Systems führen. Dies wird von befragten LKW-Fahrern als einer der Hauptgründe genannt, weshalb sich existierende Totwinkelassistenten für LKW bislang nicht am Markt durchsetzen. Somit haben die hier erstellten Verfahren hohe Anforderungen an Robustheit, Recheneffizienz und Zuverlässigkeit zu erfüllen, um im Fahrzeug mit begrenzter Rechenleistung, in Echtzeit und in einer Vielzahl von Verkehrsszenarien zuverlässig anwendbar zu sein.

9.2. Systemstruktur und Modulübersicht

Das fertige und montierte System während der Versuche ist in Abbildung 9.2 zu sehen. Hierbei wurde starker Wert auf Seriennähe gelegt, weshalb der in Abschnitt 7 beschriebene LIDAR der Firma Spies verwendet wurde, mit vier Scanebenen und 45 Messungen pro Scan.
Im Folgenden wird ein Überblick über die einzelnen Systemmodule gegeben, welche in Abbildung 9.1 grafisch dargestellt sind. Zuerst wird die

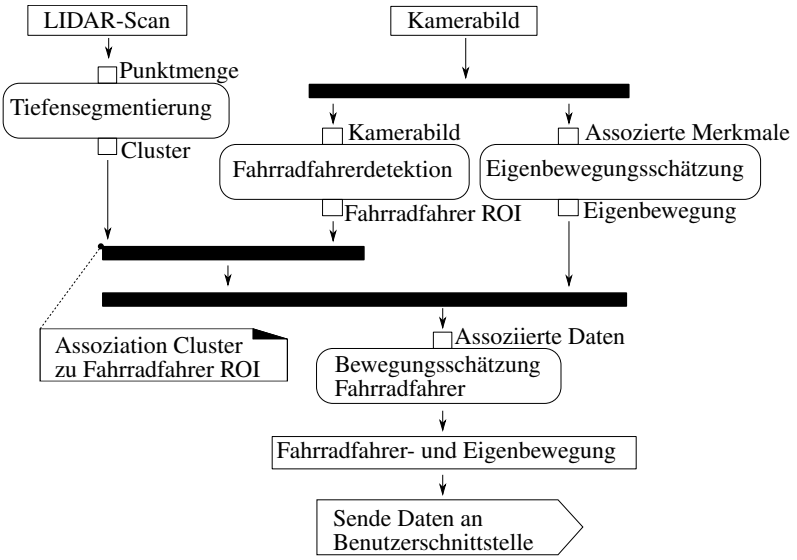


Abbildung 9.1.: Software-Framework als Aktivitätsdiagramm [62]. Das Kamerabild dient der Fahrradfahrerdetektion sowie der Eigenbewegungsschätzung des Fahrzeugs. Die Schätzung des Fahrradfahrerzustands basiert auf dem durch LIDAR gemessenen Abstand und der Fahrzeugpose. Die Fahrradfahrerposition wird an die Schnittstelle zur Warnstrategie weitergeleitet, wo eine Kritikalitätsbewertung der Situation stattfindet, um den LKW-Fahrer gegebenenfalls zu warnen.

Fahrzeug-Eigenbewegung mit den in Kapitel 5 erarbeiteten Methoden bestimmt. Um Gefahrensituationen vorhersagen zu können, sind hoch genaue Trajektorienschätzungen notwendig, weswegen hier auf die Methode mit zeitlicher Inferenz zurückgegriffen wird. Da der verwendete LIDAR nur über eine sehr geringe Auflösung verfügt, musste hier auf den in Abschnitt 6.1 vorgestellten Ansatz zur Skalenschätzung durch die Grundebene zurückgegriffen werden.
Parallel dazu wird der Fahrradfahrer im Bild detektiert und verfolgt. Dies ist durch die Algorithmen von Tian et al. [63] umgesetzt. Das Ergebnis der Detektion ist ein rechteckiger Bildausschnitt, in welchem sich der Fahrradfahrer befindet.
Um die Bewegung des Fahrradfahrers zu schätzen, ist zusätzlich Information über dessen dreidimensionale Position notwendig. Darum werden die Distanzmessungen aus dem in Abschnitt 7 vorgestellten LIDAR der Firma Spies in die Fahrradfahrerschätzung integriert. Die Methode zur extrinsischen Kalibrierung dieses LIDARs zur Kamera wurde in Abschnitt 7

Abbildung 9.2.: Hardware-Konfiguration für die abschließende Probandenstudie. Beide Sensoren werden durch Saugnäpfe an der rechten Tür des LKW angebracht. Die Datenverarbeitung geschieht durch einen handelsüblichen Laptop in der Fahrerkabine. Auf einem Bildschirm werden dem Fahrer optisch Informationen über mögliche und akute Kollisionsgefahren übermittelt.

vorgestellt. Der im Rahmen dieses Projektes entwickelte Algorithmus zur Bewegungsschätzung des Fahrradfahrers ist im Appendix, Abschnitt A.4 zu finden.
Das Ergebnis sind die geschätzten Trajektorien und Geschwindigkeiten aller erkannten Fahrradfahrer sowie des Ego-Fahrzeugs. Damit wurden Kollisionen prädiziert und diese Informationen mit Hilfe einer grafischen Oberfläche an den Fahrer weitergegeben. In Abbildung 9.3 ist ein Beispiel für die Eigenbewegungsschätzung und die Fahrradfahrerbewegungsschätzung gezeigt. Die Kollisionsprädiktion sowie die zugehörige Kollisionswarnung sind in Abbildung 9.4 dargestellt.

9.3. Ergebnisse aus der Probandenstudie

Zu Projektende wurde eine praktische Studie mit sechs Probanden und einem LKW auf einer Straße durchgeführt. Während dieses Tests konnte einerseits das generelle Funktionsprinzip des Systems im Hinblick auf die Warnsignale evaluiert werden, andererseits konnte die Robustheit der Fahrradfahrererkennung und Bewegungsschätzung in realitätsnahen Verkehrsszenarien getestet werden. Die Warnsignale wurden auf einem Bildschirm angezeigt, damit kritische Situation erkannt werden und der LKW-Fahrer

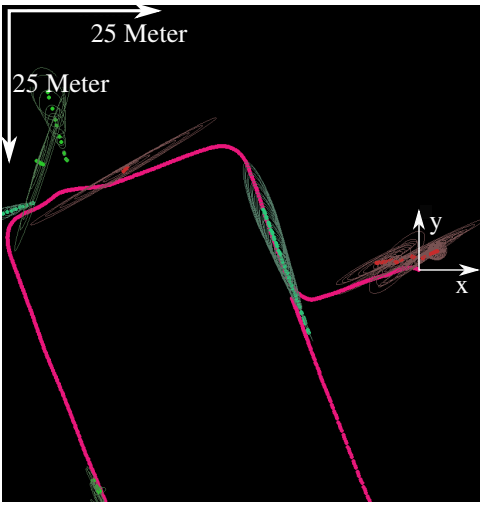


Abbildung 9.3.: Beispiel für die Eigenbewegungsschätzung und die Fahrradfahrerbewegungsschätzung nach der Umfahrung eines Häuserblocks mit erkannten Fahrradfahrern. Zusätzlich zur Eigenbewegung (pink) wird die Unsicherheit für die Fahrradfahrerbewegung geschätzt (grüne, rote, türkisfarbene Ellipsen), welche hier durch die Ein-Sigma-Ellipse der Positionsschätzung gezeigt ist. Die Unsicherheit longitudinal zum Fahrzeug ist höher, da nur zu ungefähr jedem dritten Kamerabild eine Tiefenschätzung aus dem LIDAR zugeordnet werden kann. Die hier entwickelte Methode kann nichtsdestotrotz die Position und Geschwindigkeit des Fahrradfahrers zuverlässig schätzen.

gewarnt werden kann. Eine solche Warnung ist in Abbildung 9.4 dargestellt.

Diese Studie fand im Innovationspark Wuhlheide in Berlin statt — einem realistischen innerstädtischen Szenario mit anspruchsvollen, engen Kurven und variablem Abstand der Fahrradwege zum Fahrzeug. Während der Evaluation fuhr der Fahrradfahrer mit unterschiedlichen Geschwindigkeiten und Abständen an der rechten Seite des LKWs, wie in Abbildung 9.5 zu sehen ist. Es wurden diverse Szenarien mit Überholen, Ein- und Ausscheren sowie Verdeckung des Fahrradfahrers behandelt und es konnten reale Verkehrsszenarien im Test simuliert und ausgewertet werden. Dabei wurde die Wahrnehmung des LKW-Fahrers mit einbezogen. Mithilfe der Rückmeldung des LKW-Fahrers zum Nutzen der Warnung konnten nicht nur die Algorithmen auf Genauigkeit und Robustheit getestet werden, sondern es konnte das Gesamtsystem mit Fokus auf den Anwender evaluiert werden. Während der gesamten Dauer der Testfahrten von ungefähr sechs Stunden

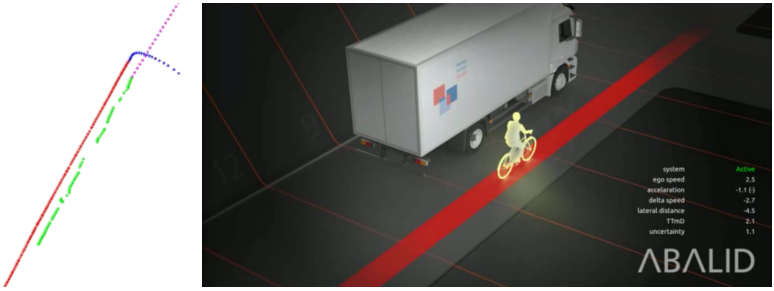


Abbildung 9.4.: Kollisionsprädiktion (links) und grafisch dargestellte Warnung (rechts) für den LKW-Fahrer. In der Anzeige sind verschiedene Informationen für den LKW-Fahrer kodiert. Über die Position der Fahrradfahrerposition in lateraler und longitudinaler Richtung des LKW kann der Fahrer erkennen, ob sich eine kritische Situation anbahnt und diese schon frühzeitig abzuwenden. Die metrische Darstellung des LKW und des Fahrradfahrers wurden dabei von Probanden als besonders hilfreich empfunden, da in Spiegeln Entfernungen oft schwierig abzuschätzen sind, in dieser Darstellung jedoch intuitiv begreifbar werden. Wird eine Kollision wie in diesem Fall prädiziert, wird der Fahrradfahrer gelb hervorgehoben, um den Fahrer zu warnen.

arbeiteten die Verfahren zur Erkennung von Fahrradfahrern und die Bestimmung von deren Bewegung sowie der Eigenbewegung des LKW zuverlässig und robust. Trotz der relativ geringen verfügbaren Rechenleistung ei-

(a) (b)

Abbildung 9.5.: Durchführung der Tests im realen Szenario: Der LKW versuchte, bei der Kreuzung rechts abzubiegen, wobei gleichzeitig ein Fahrradfahrer entlang der rechten Seite vorbeifuhr. (a) zeigt ein schwierigeres Szenario als (b), da der Fahrradfahrer durch die Bäume zwischen dem Radweg und der Fahrbahn teilweise verdeckt ist.

nes handelsüblichen Laptops aus dem Jahr 2013[1] liefen die Algorithmen in Echtzeit. Lediglich bei schwierigen Lichtverhältnissen sowie bei umfangreicher Verdeckung der Fahrradfahrer konnten diese im Kamerabild nicht detektiert werden. Dennoch wurde das System von den beteiligten LKW-Fahrern als ausgesprochen nützlich und hilfreich empfunden. Eine systematische Auswertung der Probandenstudie ist im zugehörigen Projektbericht zu finden [64].

[1] Dell Latitude E6430, Prozessor Intel Core i7 vPro, keine Nutzung der Grafikkarte für die entwickelten Algorithmen.

Kapitel 10

Fazit

Autonomes Fahren ist einer der nächsten großen Technologiesprünge und wird den Personen- und Güterverkehr revolutionieren. Voraussetzung dafür sind intelligente Systeme, welche die Umgebung erfassen und interpretieren können. Das Wissen über die zurückgelegte Strecke ist hierbei wichtig, um Umgebungsinformationen zeitlich integrieren zu können, zum Beispiel, um Objekte zuverlässig prädizieren zu können.
In dieser Arbeit wurde hierfür eine echtzeitfähige Methode zur langzeitgenauen Eigenbewegungsschätzung aus monokularen Videosequenzen entwickelt. Diese erreicht auch über lange Strecken eine sehr hohe Genauigkeit von unter $0{,}9\,\frac{\mathrm{cm}}{\mathrm{m}}$ Translationsfehler sowie $0{,}0022\,\frac{\mathrm{deg}}{\mathrm{m}}$ Rotationsfehler und fällt damit unter die zehn besten Algorithmen[1] auf dem KITTI-Datensatz. Hierbei wurden die Schlüsselkomponenten des Systems näher untersucht und evaluiert:

- Frame-zu-Frame-Bewegungsschätzung aus mehreren Kameras unter Nutzung von Bewegungsmodellen.
- Landmarken- und Posenauswahl für die zeitliche Inferenz des Graphen.
- Skalenschätzung aus verschiedenen Informationsquellen.

Die Frame-zu-Frame-Bewegungsschätzung aus mehreren Kameras zeigte hierbei eine deutliche Robustheitssteigerung gegenüber herkömmlichen Verfahren mit nur einer Kamera. So konnte gezeigt werden, dass durch die

[1] Stand 28.8.2018.

Formulierung als M-Schätzer und die Verwendung eines Bewegungsmodells das Verfahren auch mit nur wenigen Messungen zuverlässig angewandt werden kann.
Die zeitliche Inferenz von Informationen eines größeren Zeitraums ermöglicht eine genauere Trajektorienschätzung. Zwar konnten durch die in dieser Arbeit erarbeiteten Heuristiken Landmarken und Posen effektiv ausgewählt werden, um die Rechenzeit zu reduzieren, allerdings sind diese Methoden stets rechenaufwändiger als Frame-zu-Frame-Methoden, wenn die Skale außer Acht gelassen wird. Für die Skalenschätzung für Frame-zu-Frame-Methoden werden jedoch zusätzliche Schritte wie die Fluchtpunktschätzung benötigt, wohingegen die Skalenschätzung mit Methoden mit zeitlicher Inferenz weniger aufwendig ist. Wird also Wert auf kleinen Rechenaufwand und niedrige Latenz gelegt und ist die Skale zum Beispiel aus einem Raddrehzahlmesser bekannt, so ist die Multikamera-Frame-zu-Frame-Schätzung das Mittel der Wahl. Ist jedoch eine skalierte und langzeitgenaue Trajektorie benötigt, so sind Methoden mit zeitlicher Inferenz vorzuziehen.
Um die Skale für diese Methodenklasse zu schätzen, können effektiv zusätzliche Informationen genutzt werden. Die kostengünstigste Variante stellt hierbei die Nutzung des Wissens um die Einbauhöhe der Kamera dar. Von den zwei hier vorgestellten Methoden ist die in das Optimierungsproblem integrierte Bodenoberflächenschätzung der A-posteriori-Oberflächenschätzung vorzuziehen, da sie die Oberfläche genauer approximieren kann, jedoch kaum zusätzlichen Aufwand benötigt. Mit einem Translationsfehler von $1{,}2\ \frac{\mathrm{cm}}{\mathrm{m}}$ und einem Rotationsfehler von $0{,}0022\ \frac{\mathrm{deg}}{\mathrm{m}}$ ist diese Alternative für die meisten Anwendungen hervorragend geeignet.
Ist jedoch höchste Genauigkeit gefordert, so ist der Einsatz eines LIDAR zur Skalenschätzung von großem Vorteil. Die hier vorgestellte Methode kann die vom LIDAR gemessenen Punkte effektiv interpolieren, um die Tiefenschätzungen für Kamerabildmerkmale in das Optimierungsproblem zu integrieren. Der Translationsfehler von $0{,}9\ \frac{\mathrm{cm}}{\mathrm{m}}$ ist hierbei insbesondere für lange zurückgelegte Trajektorien entscheidend. Die zusätzliche Schätzung der in das Optimierungsproblem integrierten Bodenoberfläche verbessert die Genauigkeit geringfügig, was jedoch in der Praxis vernachlässigbar ist. Wird für weiterführende Anwendungen die Bodenoberfläche benötigt, so stellt dies allerdings eine effektive Möglichkeit dar, diese mitzuberechnen.
Durch die hohe Genauigkeit der vorgestellten Methoden rücken andere Anwendungen in greifbare Nähe. So könnte zum Beispiel durch Angleich der geschätzten Trajektorien an Straßenkarten oder sogar an GPS-Messungen die hochgenaue globale Lokalisierung möglich werden. Weiteres Entwick-

lungspotential besteht im Bereich der Landmarkenauswahl — der Ersatz der Heuristiken durch intelligente, selbstlernende Algorithmen bietet enorme Möglichkeiten, das Optimierungsproblem weiter zu verkleinern und somit effizienter zu machen.

Anhang A

A.1. Parameter für LIMO

non maximum suppression number	7
non maximum suppression response	40
match bin size	200
match radius	400
outlier flow tolerance	3
multi stage	1
half resolution	0
roi width	6
roi height	9
ransac plane max iterations	600
ransac plane probability	0.99
ransac plane distance threshold	0.2
maximum depth	30.0
max number landmarks near bin	400
max number landmarks middle bin	400
max number landmarks far bin	400
time between keyframes sec	0.3
critical rotation difference	0.03
max solver time	0.4
robust loss depth threshold	0.16
robust loss reprojection threshold	1.6
outlier rejection quantile	0.95
outlier rejection number iterations	1
vegetation weight	0.9

min number connecting landmarks	10

Tabelle A.1.: Parameter für Merkmalsextraktion, Tiefenschätzung und zeitliche Inferenz für LIMO.

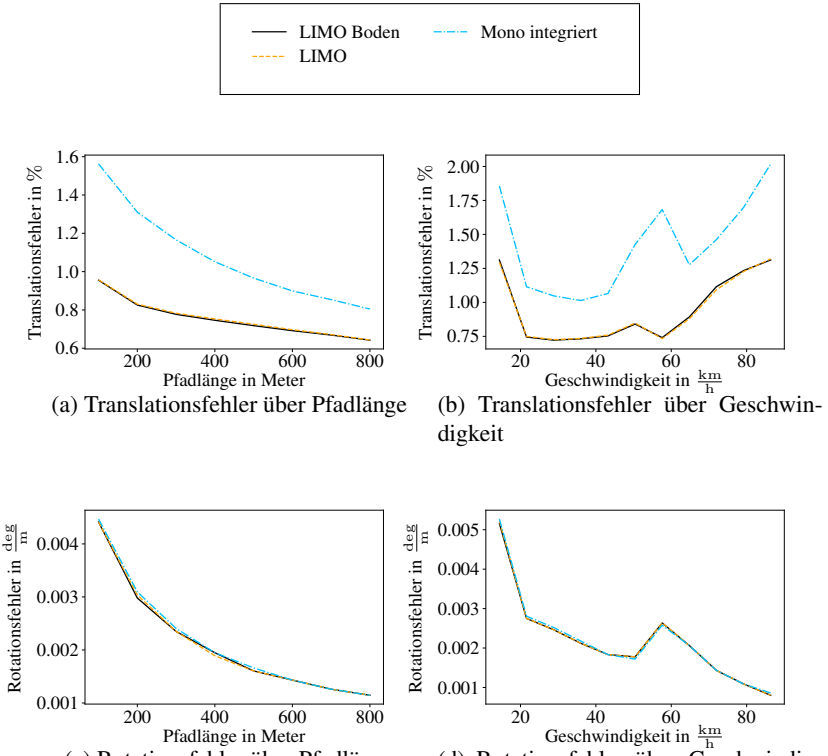
A.2. Zusätzliche Schaubilder für die Evaluation

In Abbildung A.1 und Abbildung A.2 sind Zusatzmaterialien zu Kapitel 8 zu sehen. Abbildung A.1 zeigt dieselben Daten für die Evaluation wie Abbildung 8.16, aber ohne A-Posteriori-Bodenebenenschätzung, um die Varianten von LIMO (nur LIDAR, LIDAR mit Bodenoberfläche, nur integrierte Bodenoberfläche) besser vergleichen zu können. In Abbildung A.2 sind zur Vollständigkeit die Trajektorien der Schätzung mit LIDAR und Bodenoberfläche gegeben. Diese sind visuell identisch zu den Ergebnissen nur mit LIDAR, welche in Abbildung 8.11 dargestellt sind.

A.3. Nutzung semantischer Information

Das Entfernen von Ausreißern, welche nicht der statischen Umgebung entsprechen, ist entscheidend für eine genaue Bewegungsschätzung. Zwar können diese durch robuste Schätzverfahren identifiziert und ausgeschlossen werden, jedoch ist es von großem Vorteil, möglichst viele dieser Messungen vorher auszuschließen. Des Weiteren ist semantische Information für vielerlei Anwendungen auf dem autonomen Fahrzeug notwendig. Daher werden in dieser Arbeit den Merkmalsassoziationen semantische Klassen wie Infrastruktur, Straße, Fußgänger, Auto, Vegetation etc. zugeordnet. Hierzu wird mit Hilfe eines neuronales Netzes eine semantische Klassifikation des Eingangsbildes berechnet. Möglicherweise bewegte Klassen wie Auto, Fahrrad und Fußgänger werden anschließend kategorisch von der Schätzung ausgenommen. Außerdem ermöglicht die Klassifizierung der Messungen eine Untersuchung, welche Klassen besonders wichtig sind und welche die Genauigkeit der Trajektorienschätzung verringern, wie zum Beispiel Merkmale auf Vegetation.

Zuweisung von Klassen zu Messungen Zur Schätzung der semantischen Klassifizierung wird ein Resnet38 [65] verwendet, welches die semantische Klassifikation mit 100 ms pro Bild auf einer Nvidia TitanX Pascal berechnen kann. Die daraus erzeugten semantischen Bilder werden nach Klassen binarisiert und mit einer Kernelgröße von 21 Pixel erodiert, sodass durch Fehlschätzung entstandene Lücken gefüllt werden. Anschließend wird die Klasse der Merkmalsassoziationskette über ein Mehrheitsvotum auf einer



(a) Translationsfehler über Pfadlänge

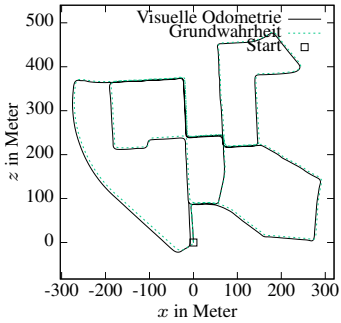
(b) Translationsfehler über Geschwindigkeit

(c) Rotationsfehler über Pfadlänge

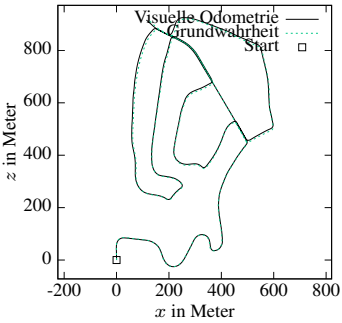
(d) Rotationsfehler über Geschwindigkeit

Abbildung A.1.: Mittlere Fehler auf dem Trainingsdatensatz des KITTI-Datensatzes (Sequenz 00 bis 10). Der Unterschied von LIDAR und Kamera ohne Bodenebenenschätzung (*LIMO*) zu LIDAR, Kamera und Bodenebenenschätzung (*LIMO Boden*) ist vernachlässigbar. Auch ohne LIDAR kann mit der in das Optimierungsproblem integrierten Bodenoberflächenschätzung ein sehr gutes Ergebnis erzielt werden (*Mono integriert*).

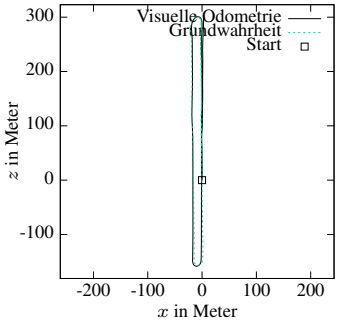
3×3 Nachbarschaft um die aktuellste Messung bestimmt. Jede Messung, welche der Vegetation zugeordnet wird, bekommt ein verringertes Gewicht in der Optimierung. Um das Gewicht zu bestimmen, wurden die Trajektorien mit verschiedenen Vegetationsgewichten auf dem Trainingsdatensatz des KITTI-Benchmark ausgewertet und verglichen. Hierbei zeigte sich ein kleiner Genauigkeitsgewinn für ein Gewicht von 0.9.



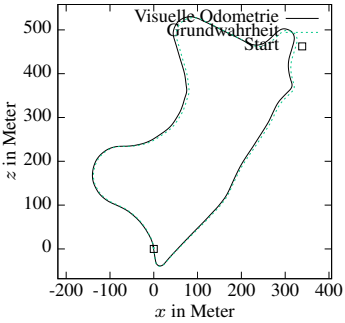
(a) Sequenz 00



(b) Sequenz 02



(c) Sequenz 06



(d) Sequenz 09

Abbildung A.2.: Beispielhafte Trajektorien aus dem Trainigsdatensatz des KITTI-Benchmark unter Verwendung der Tiefe aus LIDAR und der Bodenoberflächenschätzung.

A.4. Fahrradfahrer-Bewegungsschätzung

In diesem Kapitel wird die für das in Abschnitt 9 beschriebene Projekt Abalid entwickelte Eigenbewegungsschätzung für Fahrradfahrer beschrieben [66]. Eine Besonderheit dieses Sensoraufbaus ist, dass zwei verschiedene Arten von Messungen vorliegen, welche in einem regressionsbasierten Schätzproblem verwendet werden: die Sichtstrahlen aus der bildbasierten Detektion des Fahrradfahrers und die Distanzmessungen aus dem LIDAR. Aus der Kamera ist die Distanz des Fahrradfahrers nicht messbar, wohingegen der LIDAR nicht den genauen Sichtwinkel zum Fahrradfahrer messen kann. Um dieses Problem zu lösen, wird hier eine Regression mit latenten Variablen eingesetzt. Die latente Variable ist dabei die Distanz des Fahrradfahrers zur Kamera, welche weniger häufig verfügbar ist als die Messung seiner Richtung. Durch das hohe Rauschen in der LIDAR-Tiefenmessung müssen des Weiteren starke Modellannahmen getroffen werden. Darum wird der Fahrradfahrer durch eine Punktmasse approximiert, welche sich mit konstanter Geschwindigkeit entlang einer Linie im dreidimensionalen Raum bewegt. Diese Annahme trifft nicht auf die reale Fahrradfahrerbewegung zu. Daher wird diese entschärft, indem die Regression nicht über alle Messungen erfolgt, sondern nur in einem Fenster von n Bildern. So können auch Beschleunigung und Kurvenfahrten modelliert werden.

Algorithmen Im Folgenden werden die zur Bewegungsschätzung der Fahrradfahrer verwendeten Algorithmen näher beschrieben: Um die Position $\mathbf{x} = (x,y,z)^T$ und die Geschwindigkeit $\mathbf{v} = (v_x,v_y,v_z)^T$ zum Zeitpunkt t_i zu schätzen, wird der quadratische Fehler aus Gleichung A.1 minimiert.

$$\underset{\mathbf{x},\mathbf{v}}{\operatorname{argmin}} \quad \sum_{i=1}^{n} \|(\mathbf{x} + t_i\mathbf{v} - \mathbf{p}_i)\|_2^2, \tag{A.1}$$

wobei $\mathbf{p}_i \in \mathbb{R}^3$ die gemessene Fahrradfahrerposition in Weltkoordinaten zum Zeitpunkt t_i beschreibt. Durch Nullsetzen der ersten Ableitung von $\mathbf{x}$ und $\mathbf{v}$ erhält man das lineare Least-Squares-Problem aus Gleichung A.2.

$$\begin{bmatrix} \mathbf{I} \cdot n & \mathbf{I}\sum_i t_i \\ \mathbf{I}\sum_i t_i & \mathbf{I}\sum_i t_i^2 \end{bmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \sum_i \mathbf{p}_i \\ \sum_i t_i \mathbf{p}_i \end{pmatrix}, i = 1 \ldots n \tag{A.2}$$

$\mathbf{I}$ bezeichnet hierbei die Einheitsmatrix aus $\mathbb{R}^3$. Die Fahrradfahrerposition $\mathbf{p}_i$ wird vom Fahrzeug aus Position $\mathbf{p}_{i,\text{Fahrzeug}}$ gemessen, wobei durch den

LIDAR die Tiefe r_i und durch die Kamera die Richtung $\mathbf{w}_i$ bekannt ist. Daraus folgt:

$$\mathbf{p}_i = \mathbf{p}_{i,\text{Fahrzeug}} + r_i \mathbf{w}_i, \quad \|\mathbf{w}_i\|_2 = 1 \tag{A.3}$$

Dadurch, dass Kamera und LIDAR nicht synchronisiert sind, kann die Distanz r_i nicht zu jedem Zeitpunkt bestimmt werden. Um diese latente Variable nicht explizit schätzen zu müssen, wird ein impliziter Schätzwert für r_i genutzt. Dafür wird die lokale Fahrradfahrerposition $\mathbf{x}+t_i\mathbf{v}-\mathbf{p}_{i,\text{Fahrzeug}}$ zu jedem Zeitpunkt t_i auf die aus der Kamera bekannten Richtung $\mathbf{w}_i$ projiziert.

Der Tiefen-Schätzwert r_i' für nicht beobachtete r_i kann daher durch Gleichung A.4 ausgedrückt werden.

$$r_i' = \mathbf{w}_i^T \mathbf{x}_i \tag{A.4}$$

Es können also drei verschiedene Typen von Messungskonfigurationen beobachtet werden:

1. Eine LIDAR-Messung kann zu einer Kameramessung assoziiert werden. Damit ist $\mathbf{w}_i$ mithilfe der Kamera beobachtet und r_i durch den LIDAR bekannt.

2. Es kann keine LIDAR-Messung zur Kameramessung assoziiert werden. Hier wird Gleichung A.4 genutzt, um r_i mit r_i' abzuschätzen. Die Richtung $\mathbf{w}_i$ ist weiterhin durch die Kameramessung bekannt.

3. Ein Punktecluster nahe einer vorherigen Schätzung kann aus dem LIDAR extrahiert werden. In diesem Fall können sowohl r_i als auch $\mathbf{w}_i$ aus dem Mittelstrahl des Clusters extrahiert werden.

Diese können nun in Algorithmus 3 verwendet werden, um die Position und die Geschwindigkeit des Fahrradfahrers in allen drei Raumrichtungen zu erhalten.

Bemerkenswert ist hierbei, dass die Fahrradfahrerbewegung mit einer linearen Methode geschätzt werden kann. Die Rechenzeit ist daher unter $1\ \text{ms}$ mit einer Fenstergröße von $n = 12$ Bildern auf einem handelsüblichen Laptop des Typs *Dell Latitude E6430*. Im Allgemeinen hat sich $n = 12$ als ein guter Kompromiss zwischen Modellierung nicht-linearer Bewegung und Ausgleich des Messrauschens erwiesen. Bei der hier verwendeten Bildfrequenz von $10\ \text{Hz}$ entspricht das $1.2\ \text{s}$.

Clustering und Assoziation von LIDAR- und Kameramessungen Um die vom LIDAR gemessenen Punktmenge nutzen zu können, werden in ihr

Data : Zeitinstanzen (Größe n): t;
Richtungsvektoren von Kamera zu Fahrradfahrer (Größe n): $\mathbf{w}$;
Kamerapositionen in globalen Koordinaten (Größe n): $\mathbf{P}$;
Tiefenmessungen von Kamera zu Fahrradfahrer (Größe $< n$): r;
Result : Position und Geschwindigkeit des Fahrradfahrers: XV;
$\mathbf{M} \leftarrow$ Nullmatrix von $\mathbb{R}^{6\times 6}$;
$\mathbf{C} \leftarrow \begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T$;
$\mathbf{I} \leftarrow$ Identität von $\mathbb{R}^{3\times 3}$;
foreach $i \in n$ **do**
  $\mathbf{M} \leftarrow \mathbf{M} + \begin{pmatrix}\mathbf{I} & t(i)\mathbf{I} \\ t(i)\mathbf{I} & t(i)^2\mathbf{I}\end{pmatrix}$;
  $\mathbf{C} \leftarrow \mathbf{C} + \begin{pmatrix}\mathbf{P}(i) \\ t(i)\mathbf{P}(i)\end{pmatrix}$;
  if $Observed$ **then**
    $\mathbf{C} \leftarrow \mathbf{C} + \begin{pmatrix}r(i)\mathbf{w}(i) \\ t(i)r(i)\mathbf{w}(i)\end{pmatrix}$;
  else
    $\mathbf{WW} \leftarrow \mathbf{w}(i) \cdot \mathbf{w}(i)^T$;
    $\mathbf{M} \leftarrow \mathbf{M} - \begin{pmatrix}\mathbf{WW} & t(i)\mathbf{WW} \\ t(i)\mathbf{WW} & t(i)^T \cdot \mathbf{W}\end{pmatrix}$;
    $\mathbf{C} \leftarrow \mathbf{C} - \begin{pmatrix}\mathbf{WW} \cdot \mathbf{P}(i) \\ t(i) \cdot \mathbf{WWP}(i)\end{pmatrix}$;
  end
end
$\mathbf{XV} \leftarrow \mathbf{M}^{-1}\mathbf{C}$;

Algorithmus 3 : Algorithmus zur Schätzung der Fahrradfahrerposition und -geschwindigkeit mithilfe eines linearen Least-Squares-Problems mit latenten Variablen.

charakteristische Muster identifiziert, welche zum Fahrradfahrer gehören könnten. Zuerst wird die Punktewolke in Cluster segmentiert, welche sich durch unstetige Tiefe voneinander abheben. Implementiert ist dies durch die *Linkage*-Metrik von Moosmann et al. [67]. Da es sich hier um einzelne Linien handelt, werden über einen Schwellwert neue Cluster instanziiert. Zu jeder Detektion aus der Kamera soll das zugehörige Punktmengen-Cluster assoziiert werden. Die Kriterien hierfür sind die folgenden:

1. Der zu erwartende Abstand des Fahrradfahrers zur Kamera.

2. Der räumliche Abstand des Fahrradfahrer-Sichtstrahls zum Cluster.

Für das erste Kriterium werden vorherige Schätzungen genutzt, um die detektierte Position, Höhe und Breite des Fahrradfahrers im Bild vorherzusagen. Nimmt man eine durchschnittliche Höhe für den Fahrradfahrer als gegeben an, kann man mithilfe der Kameraparameter dessen zu erwartende Höhe im Bild schätzen. Die Tiefe z ergibt sich also zu $z = f\frac{\Delta h}{\Delta v}$ mit der Brennweite der Kamera f, der Höhe des Fahrradfahrers Δh in Metern und der im Bild gemessenen Höhe Δv in Pixel. Mithilfe einer Taylorreihenentwicklung erster Ordnung um Δh und Δv kann der Fehler in z fortgepflanzt werden:

$$\sigma_z = \frac{f}{\Delta v}\sqrt{\frac{\Delta h^2}{\Delta v^2}\cdot\sigma_{\Delta v}^2 + \sigma_{\Delta h}^2}. \tag{A.5}$$

Um diese Information zur Assoziation nutzen zu können, müssen sie auf das Intervall $[0, 1]$ skaliert werden.
Als zweites Kriterium zur Zuordnung der Cluster wird der räumliche Abstand des Cluster-Mittelpunkts $\mathbf{m}$ zum durch die Fahrradfahrerdetektion bekannten Sichtstrahl $\mathbf{r}$ genutzt. Der Sichtstrahl $\mathbf{r}$ folgt direkt aus der Anwendung des inversen Kameramodells auf den Mittelpunkt der Detektion aus der Bildebene.

Nun kann der minimale Abstand d von $\mathbf{r}$ zu $\mathbf{p}$ mithilfe von

$$d = (\mathbf{r}^T\mathbf{p})\mathbf{r} - \mathbf{p} \tag{A.6}$$

ermittelt werden. Um beide Kriterien vergleichen zu können, wird d wiederum auf das Intervall $[0,1]$ skaliert und werden beide Kriterien mithilfe einer *Fuzzy-Logik* kombiniert. Dies ist mithilfe einer parametrisierten Sigmoidfunktion $sigm(x,o,g) = \frac{1}{1+g\cdot\exp(x-o)}$ umgesetzt, wobei o den Offset in x darstellt und g den exponentiellen Verstärkungsfaktor bezeichnet. Die hier verwendete Sigmoidfunktion ist in Abbildung A.3 zu sehen. Somit werden Abstände bis zu ca. $2\,\mathrm{m}$ toleriert, bevor der Wert der Sigmoidfunktion stark absinkt. Für jedes potentielle Cluster-Bild-Paar wird anschließend

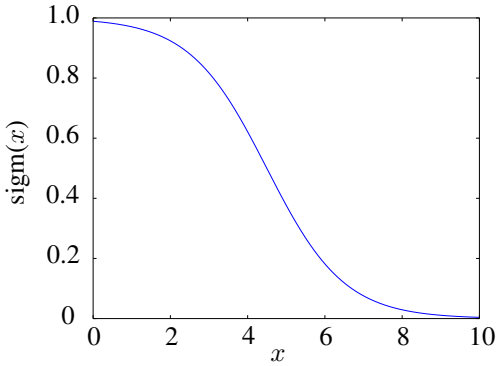


Abbildung A.3.: Sigmoidfunktion für die Tiefenassoziation mit Offset $o = 1.5$ und Verstärkungsfaktor $g = 0.05$.

der Mittelwert der beiden Kriterien gebildet und werden die am besten bewerteten Paarungen für die Bewegungsschätzung des Fahrradfahrers verwendet.

Literatur

[1] A. Geiger, P. Lenz und R. Urtasun, „Are we ready for autonomous driving? the kitti vision benchmark suite“, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2012, S. 3354–3361.

[2] Kraftfahrt-Bundesamt. (Juni 2018). Verkehr in Kilometern deutscher Kraftfahrzeuge 2016, Adresse: `https://www.kba.de/DE/Statistik/Kraftverkehr/VerkehrKilometer/verkehr_in_kilometern_node.html`.

[3] Statistisches Bundesamt. (Juni 2018). Polizeilich erfasste Unfälle, Adresse: `https://www.destatis.de/DE/ZahlenFakten/Wirtschaftsbereiche/TransportVerkehr/Verkehrsunfaelle/Tabellen_/Strassenverkehrsunfaelle.html`.

[4] B. Triggs, P. F. McLauchlan, R. I. Hartley und A. W. Fitzgibbon, „Bundle adjustment—a modern synthesis“, in *International workshop on vision algorithms*, Springer, 1999, S. 298–372.

[5] L. Carlone, R. Tron, K. Daniilidis und F. Dellaert, „Initialization techniques for 3d SLAM: A survey on rotation estimation and its use in pose graph optimization“, in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, S. 4597–4604.

[6] R. Hartley und A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[7] P. H. Torr und A. W. Fitzgibbon, „Invariant fitting of two view geometry“, *IEEE transactions on pattern analysis and machine intelligence*, Bd. 26, Nr. 5, S. 648–650, 2004, IEEE.

[8] R. Szeliski, *Computer vision: Algorithms and applications*. Springer Science & Business Media, 2010.

[9] M. Buczko und V. Willert, „How to distinguish inliers from outliers in visual odometry for high-speed automotive applications“, in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2016, S. 478–483.

[10] ——, „Flow-decoupled normalized reprojection error for visual odometry“, in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, S. 1161–1167.

[11] N. Fanani, A. Stürck, M. Barnada und R. Mester, „Multimodal scale estimation for monocular visual odometry“, in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, S. 1714–1721.

[12] B. Kitt, J. Rehder, A. Chambers, M. Schönbein, H. Lategahn und S. Singh, „Monocular visual odometry using a planar road model to solve scale ambiguity.“, in *IEEE European Conference on Mobile Robotics (ECMR)*, IEEE, 2011, S. 43–48.

[13] D. Zhou, Y. Dai und H. Li, „Reliable scale estimation and correction for monocular visual odometry“, in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2016, S. 490–495.

[14] T. Caselitz, B. Steder, M. Ruhnke und W. Burgard, „Monocular camera localization in 3d lidar maps“, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, S. 1926–1931.

[15] J. Zhang und S. Singh, „Visual-lidar odometry and mapping: Low-drift, robust, and fast“, in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, S. 2174–2181.

[16] A. Geiger, P. Lenz, C. Stiller und R. Urtasun, „Vision meets robotics: The KITTI dataset“, *The International Journal of Robotics Research*, Bd. 32, Nr. 11, S. 1231–1237, 2013, Sage Publications Sage UK: London, England.

[17] A. Geiger, J. Ziegler und C. Stiller, „Stereoscan: Dense 3d reconstruction in real-time“, in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, S. 963–968.

[18] M. Sons, H. Lategahn, C. G. Keller und C. Stiller, „Multi trajectory pose adjustment for life-long mapping“, in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2015, S. 901–906.

[19] H. Lategahn, A. Geiger und B. Kitt, „Visual SLAM for autonomous ground vehicles“, in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2011, S. 1732–1737.

[20] J. Engel, J. Stückler und D. Cremers, „Large-scale direct SLAM with stereo cameras“, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, S. 1935–1942.

[21] R. Mur-Artal und J. D. Tardós, „Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras“, *IEEE Transactions on Robotics*, Bd. 33, Nr. 5, S. 1255–1262, 2017, IEEE.

[22] D. Gálvez-López und J. D. Tardos, „Bags of binary words for fast place recognition in image sequences“, *IEEE Transactions on Robotics*, Bd. 28, Nr. 5, S. 1188–1197, 2012, IEEE.

[23] A. Nordmann. (Juni 2018). Eigenes Werk. CC BY-SA 3.0, Adresse: `https://commons.wikimedia.org/w/index.php?curid=3302426`.

[24] M. A. Fischler und R. C. Bolles, „Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography“, *Communications of the ACM*, Bd. 24, Nr. 6, S. 381–395, 1981, ACM.

[25] C. Harris und M. Stephens, „A combined corner and edge detector“, in *Alvey vision conference*, Citeseer, Bd. 15, 1988, S. 10–5244.

[26] D. G. Lowe, „Object recognition from local scale-invariant features“, in *IEEE international conference on Computer Vision (ICCV)*, IEEE, Bd. 2, 1999, S. 1150–1157.

[27] H. Bay, A. Ess, T. Tuytelaars und L. Van Gool, „Speeded-up robust features (SURF)“, *Computer vision and image understanding*, Bd. 110, Nr. 3, S. 346–359, 2008, Elsevier.

[28] E. Rublee, V. Rabaud, K. Konolige und G. Bradski, „Orb: An efficient alternative to SIFT or SURF“, in *IEEE international conference on Computer Vision (ICCV)*, IEEE, 2011, S. 2564–2571.

[29] M. Calonder, V. Lepetit, C. Strecha und P. Fua, „Brief: Binary robust independent elementary features“, in *European conference on computer vision (ECCV)*, Springer, 2010, S. 778–792.

[30] H. Lategahn, J. Beck und C. Stiller, „DIRD is an illumination robust descriptor“, in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2014, S. 756–761.

[31] J. Gräter, T. Strauss und M. Lauer, „Momo: Monocular motion estimation on manifolds“, in *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2017, S. 1–6.

[32] P. J. Rousseeuw und A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, Bd. 589.

[33] D. Nistér, „An efficient solution to the five-point relative pose problem“, *IEEE transactions on pattern analysis and machine intelligence*, Bd. 26, Nr. 6, S. 756–770, 2004, IEEE.

[34] R. Hartley und A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[35] freedesignfile. (Sep. 2018). Skizze einer Stadt, Adresse: `https://all-free-download.com/free-vector/download/hand-drawn-town-streets-design-vector-set_573423.html`.

[36] D. Scaramuzza, „1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints“, *International journal of computer vision*, Bd. 95, Nr. 1, S. 74–85, 2011, Springer.

[37] J. Ziegler, „Optimale Bahn- und Trajektorienplanung für Automobile“, Diss., Karlsruher Institut für Technologie, 2015.

[38] C. Wu, T. A. Huang, M. Muffert, T. Schwarz und J. Gräter, „Precise pose graph localization with sparse point and lane features“, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, S. 4077–4082.

[39] P. H. Torr und A. Zisserman, „Mlesac: A new robust estimator with application to estimating image geometry“, *Computer vision and image understanding*, Bd. 78, Nr. 1, S. 138–156, 2000, Elsevier.

[40] J. Gräter, T. Schwarze und M. Lauer, „Robust scale estimation for monocular visual odometry using structure from motion and vanishing points“, in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2015, S. 475–480.

[41] T. Schwarze und M. Lauer, „Minimizing odometry drift by vanishing direction references“, 2015, Citeseer.

[42] R. E. Kalman, „A new approach to linear filtering and prediction problems“, *Journal of basic Engineering*, Bd. 82, Nr. 1, S. 35–45, 1960, American Society of Mechanical Engineers.

[43] J. Gräter, A. Wilczynski und M. Lauer, „Limo: Lidar-monocular visual odometry“, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, S. 1–8.

[44] J. Shi u. a., „Good features to track“, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 1994, S. 593–600.

[45] J. Gräter, T. Strauss und M. Lauer, „Photometric laser scanner to camera calibration for low resolution sensors“, in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, S. 1552–1557.

[46] G. Pandey, J. McBride, S. Savarese und R. Eustice, „Extrinsic calibration of a 3d laser scanner and an omnidirectional camera“, in *IFAC symposium on intelligent autonomous vehicles*, IFAC, Bd. 7, 2010.

[47] A. Geiger, F. Moosmann, Ö. Car und B. Schuster, „Automatic camera and range sensor calibration using a single shot“, in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2012, S. 3936–3943.

[48] Q. Zhang und R. Pless, „Extrinsic calibration of a camera and laser range finder (improves camera calibration)“, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Bd. 3, 2004, S. 2301–2306.

[49] D. Scaramuzza, A. Harati und R. Siegwart, „Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes“, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2007, S. 4164–4169.

[50] K.-H. Lin, C.-H. Chang, A. Dopfer und C.-C. Wang, „Mapping and localization in 3d environments using a 2d laser scanner and a stereo camera.“, *Journal of information science and engineering*, Bd. 28, Nr. 1, S. 131–144, 2012, Institute of Information Science Academia Sinica.

[51] Z. Taylor und J. Nieto, „Motion-based calibration of multimodal sensor arrays“, in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, S. 4843–4850.

[52] S. Rusinkiewicz und M. Levoy, „Efficient variants of the ICP algorithm“, in *IEEE International Conference on 3-D Digital Imaging and Modeling*, IEEE, 2001, S. 145–152.

[53] G. Li, Y. Liu, L. Dong, X. Cai und D. Zhou, „An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features“, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2007, S. 3854–3859.

[54] Leddartech. (Mai 2018). Solid state lidars, Adresse: `https://leddartech.com/technology-fundamentals/`.

[55] Spies. (Mai 2016). Engineers Spies, Adresse: http://www.ib-spies.de/index.php.

[56] Pepperl&Fuchs. (Mai 2016). Pepperl&Fuchs, R2000, Adresse: http://files.pepperl-fuchs.com/selector%5C_files/navi/productInfo/doct/doct3469b.pdf.

[57] V. Lepetit, F. Moreno-Noguer und P. Fua, „Epnp: An accurate o (n) solution to the pnp problem“, *International journal of computer vision*, Bd. 81, Nr. 2, S. 155, 2009, Springer.

[58] J. Engel, T. Schöps und D. Cremers, „Lsd-slam: Large-scale direct monocular SLAM“, in *European Conference on Computer Vision (ECCV)*, Springer, 2014, S. 834–849.

[59] J. Gräter, T. Wei, M. Lauer und S. Christoph, „Abalid: Abbiegeassistenz mit 3D-LIDAR-Sensorik. Abschlussbericht des Teilvorhabens 3D-Objekterkennung und semantische Analyse.“, 2016, KIT.

[60] Wikipedia. (Sep. 2018). Fahrradfahren in Kopenhagen, Adresse: https://de.wikipedia.org/wiki/Radfahren_in_Kopenhagen.

[61] NDR. (Juni 2018). Artikel des NDR über einen Unfall, Adresse: https://www.ndr.de/nachrichten/niedersachsen/hannover_weser-leinegebiet/Toedliches-Lkw-Unglueck-Neues-Gesetz-gefordert,hannover13276.html.

[62] Microsoft. (Sep. 2018). Uml activity diagrams guidelines, Adresse: https://docs.microsoft.com/en-gb/visualstudio/modeling/uml-activity-diagrams-guidelines?view=vs-2015.

[63] W. Tian und M. Lauer, „Fast cyclist detection by cascaded detector and geometric constraint“, in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2015, S. 1286–1291.

[64] J. Thomas. (Juni 2018). Projekt ABALID - Abbiegeassistent mit 3D-LIDAR-Sensorik : Schlussbericht : Laufzeit des Vorhabens: 01.03.2013 bis 31.05.2016, Adresse: https://www.tib.eu/en/search/id/TIBKAT%3A875173470/Projekt-ABALID-Abbiegeassistent-mit-3D-LIDAR-Sensorik/.

[65] K. He, X. Zhang, S. Ren und J. Sun, „Deep residual learning for image recognition“, in *IEEE conference on computer vision and pattern recognition*, IEEE, 2016, S. 770–778.

[66] J. Gräter und M. Lauer, „Monoscopic automotive ego-motion estimation and cyclist tracking“, in *Doctoral Consortium - DCVISIGRAPP, (VISIGRAPP)*, INSTICC, SciTePress, 2015, S. 37–44.

[67] F. Moosmann, „Interlacing self-localization, moving object tracking and mapping for 3d range sensors“, Diss., Karlsruher Institut für Technologie, 2013.