



Universität der Bundeswehr München  
Fakultät für Luft- und Raumfahrttechnik  
Institut für Technik Autonomer Systeme

# Modellbasierte Fahrzeugerkennung eines Fahrerassistenzsystems zum autonomen Folgen im Konvoi

Carsten Fries, M.Sc.



Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. rer. nat. Matthias Gerds  
1. Berichterstatter: Prof. Dr.-Ing. Hans-Joachim Wünsche  
2. Berichterstatter: Prof. Dr.-Ing. habil. Dirk Wollherr

Diese Dissertation wurde am 16.01.2019 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 24.06.2019 angenommen. Die mündliche Prüfung fand am 01.07.2019 statt.



## Vorwort

An dieser Stelle möchte ich die Gelegenheit nutzen, einigen Unterstützern dieser wissenschaftlichen Arbeit meinen Dank auszusprechen.

Ein besonderer Dank geht an Herrn Prof. Dr.-Ing. Hans-Joachim Wünsche für die Promotionsmöglichkeit am Institut für Technik Autonomer Systeme an der Universität der Bundeswehr München. Ich bekam jederzeit wertvollen akademischen Rat und hatte meinen wissenschaftlichen Freiraum.

In der Wissenschaft wird das errungene Wissen miteinander geteilt, damit der wissenschaftliche Fortschritt effektiver vorangetrieben wird. Die vorliegende Arbeit basiert auf den Arbeiten und Erkenntnissen von Dr.-Ing. Michael Manz. Ich danke dir für deine Vorarbeiten.

Diese Arbeit befasst sich mit selbstständig fahrenden Folgefahrzeugen in Konvois. Der entwickelte Folgeassistent wurde durch die gute Zusammenarbeit aller Institutsmitarbeiter realisiert: Ein großer Dank ist Thorsten Lüttel auszusprechen, der stets mit Rat und Tat zur Verfügung stand. Meinen Bürokollegen Sebastian Schneider und Michael Kusenbach danke ich besonders für die angenehme Arbeitsatmosphäre und die konstruktiven akademischen Gespräche. Ein besonderes Dankeschön geht an Sebastian Bayerl, der die Spurgenerierung von Michael Manz weiterentwickelt hat. Nicht zu vergessen ist Dennis Faßbender, der mir hilfreiche Verbesserungsvorschläge bei meinen wissenschaftlichen Veröffentlichungen gab und in Industrieprojekten immer zur Stelle war. Des Weiteren sei mein Dank allen anderen Institutsmitarbeitern ausgesprochen: Jan Kallwies, Patrick Burger, Michael Himmelsbach, Benjamin Heinrich, Georg Müller, Hanno Jaspers, Torsten Engler, Benjamin Naujoks, Felix Ebert, Martin Michaelis, Philipp Berthold, André Müller, Felix von Hundelshausen, Alois Unterholzner und Falk Hecker. Ihr wart alle sehr hilfreich und habt das Arbeiten im Institut positiv bereichert. Ebenso danke ich Martina Falter, die immer ein offenes Ohr für mich hatte, und den technischen Angestellten des Lehrstuhls: Rudolf Waldherr, Anton Pröls und Günther Hofbauer. Darüber hinaus bedanke ich mich beim 2. Berichterstatter Herrn Prof. Dr.-Ing. habil. Dirk Wollherr und beim Vorsitzenden des Promotionsausschusses Herrn Prof. Dr. rer. nat. Matthias Gerdts.

Die Betreuung und Ergebnisse der ehemaligen Masterstudenten Arne Birnbaum und Torben Teepe brachten mir nützliches Wissen im Bereich der visuellen Modellgenerierung. Danke für eure Unterstützung.

Ein ganz besonderer Dank geht schließlich an meine Lebensgefährtin, Eltern und Geschwister, die mich in jeder Lebenslage unterstützen.

Feldkirchen, den 7. Juli 2019



## Kurzfassung

In dieser Arbeit wird eine modellbasierte Fahrzeugerkennung zum autonomen Fahren im Konvoi vorgestellt. Der Fokus liegt zum einen auf einer präzisen Erkennung eines Führungsfahrzeuges, sodass diesem spurtreu gefolgt werden kann. Zum anderen soll der Folgeassistent umgebungsunabhängig und bei Tag als auch bei Nacht einsatzbereit sein.

Für ein spurtreues Folgen ist eine exakte Schätzung der 6D-Pose (Position und Orientierung) des Führungsfahrzeuges notwendig. Das in dieser Arbeit vorgestellte Verfahren ermittelt die ego-relative 6D-Pose des Führungsfahrzeuges sowie dessen Geschwindigkeit und Lenkwinkel mit einer hohen Genauigkeit. Die Wahrnehmung von Fahrzeugen erfolgt dabei modellbasiert, weshalb für jeden Fahrzeugtyp einmalig ein sogenanntes 3D-Merkmalmodell erstellt werden muss. Diese Arbeit beschreibt sowohl eine manuelle als auch eine automatische Modellgenerierung. In der Generierung von 3D-Merkmalmodellen werden die markanten Textur-, Farb-, Wärme- und Hindernismerkmale für jeden Fahrzeugtyp ermittelt und gespeichert. Anschließend ist eine modellbasierte Fahrzeugerkennung mit solchen fahrzeugspezifischen Merkmalsmodellen möglich. Hinsichtlich der Fahrzeugerkennung und dessen zeitlicher Verfolgung werden drei verschiedene Algorithmen beschrieben und verglichen. Der erste Algorithmus arbeitet mit einem Unscented Kalman-Filter mit sequentieller Innovation. Anstelle eines Kalman-Filters verwendet der zweite Algorithmus einen Partikel-Filter und der dritte Algorithmus kombiniert beide Verfahren. Die Verwendung des jeweiligen Algorithmus hängt vom Einsatzzweck ab. Darf ein Fahrzeugerkennungssystem nur wenig Rechnerlast verursachen, wird die Verwendung des ersten Algorithmus empfohlen. Liegt die Priorität dagegen auf der Genauigkeit der Posenbestimmung, ist die zweite und dritte Variante zu bevorzugen.

Die Praxistests erfolgten primär in unstrukturiertem Terrain und mit einem Konvoi von bis zu drei Teilnehmern. Hierbei kamen sowohl unterschiedliche Pkw als auch Lkw zum Einsatz, die mehrere hundert Kilometer im autonomen Betrieb absolvierten. Die grundlegende Sensorausstattung der autonomen Fahrzeuge besteht aus einem Inertialen Navigationssystem, einem Laserscanner und mindestens einer Tageslichtkamera. Für die Fahrzeugerkennung bei Nacht wird diese Grundausstattung mit einer Wärmebildkamera ergänzt. Optional können zusätzlich Low-Light-Kameras und Radarsysteme verwendet werden.

Das echtzeitfähige Gesamtsystem zeigte sich beständig gegenüber schwierigen Witterungsbedingungen (Regen, Schnee) sowie teilweisen Verdeckungen des Führungsfahrzeuges (Scheibenwischer, Verkehrsteilnehmer) und überzeugte durch spurtreues Folgen bei Geschwindigkeiten von bis zu 100 km/h.



## Summary

### **Model-based Vehicle Tracking of an Advanced Driver Assistance System for Autonomous Following in a Convoy**

This thesis presents a model-based vehicle tracking system for autonomous convoy driving. On one hand, it focuses on precise vehicle detection for path-accurate following. On the other hand, the overall assistance system functionality should be robust against changing conditions of the environment and illumination.

The path-accurate following requires the exact estimation of the local convoy leader's 6D pose (position and orientation). The proposed path-accurate vehicle tracking system estimates the ego-relative 6D pose of the convoy leader as well as its velocity and steering angle. The vehicle recognition is based on vehicle-specific and predefined 3D feature models. This thesis presents a manual and an automatic approach to build 3D feature models. Both methods determine the prominent texture, color, thermal and occupancy features for each vehicle type and store them in reusable 3D feature models. Afterwards, vehicle detection and tracking based on the resulting feature models are possible. For this purpose, three different model-based algorithms will be proposed and compared. The first algorithm uses an unscented Kalman filter with sequential innovation. It detects and tracks the convoy leader with low computational costs and less precision. The second algorithm uses a particle filter instead of a Kalman filter whereas the third algorithm combines both methods. The last two algorithms are computationally expensive due to the particle filter but the precision of the vehicle detection is higher.

All algorithms have been tested in unstructured terrain and with convoys of up to three participants. In the practical evaluation, passenger cars and trucks were used to autonomously drive several hundred kilometers. Each autonomous vehicle consists of a sensor base and an extended equipment for driving by night. The sensor base has an inertial navigation system, one laser scanner and at least one day-light camera. The extended equipment includes a thermal camera in addition. Optionally, additional sensors are low-light cameras and radar sensors.

The tests of the proposed advanced driver assistance system shows robustness against rough weather conditions (rain, snow) and partial occlusion of the convoy leader (e. g. windshield wiper and traffic participants). It is able to follow path-accurately and has been tested with convoy velocities up to 100 km/h.



„As it turned out, it worked.“

---

*(Prof. Dr.-Ing. Ernst D. Dickmanns, Pionier des dynamischen  
maschinellen Sehens und der autonomen Fahrzeuge.)*

# Inhaltsverzeichnis

<b>Symbolverzeichnis</b>	<b>XI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Forschungsbeitrag . . . . .	3
1.2 Struktur der Arbeit . . . . .	4
<b>2 Stand der Forschung</b>	<b>5</b>
2.1 Autonomes Folgen im Konvoi . . . . .	6
2.1.1 Historische Platooning-Projekte . . . . .	6
2.1.2 Aktuelle Forschungsprojekte des BAAINBw . . . . .	9
2.1.3 Serienreife Assistenzsysteme . . . . .	10
2.2 Fahrzeugerkennung, -klassifizierung und -verfolgung . . . . .	11
2.2.1 Technische Komponenten . . . . .	11
2.2.2 Markerbasierte Verfahren . . . . .	12
2.2.3 Markerlose Verfahren . . . . .	13
<b>3 Grundlagen</b>	<b>17</b>
3.1 Sensorkalibrierung und -synchronisierung . . . . .	17
3.1.1 Sensorkalibrierung . . . . .	17
3.1.1.1 Kalibrierung extrinsischer Größen . . . . .	18
3.1.1.2 Kalibrierung intrinsischer Größen . . . . .	20
3.1.2 Sensorsynchronisierung . . . . .	27
3.2 Rekursive Zustandsschätzung dynamischer Systeme . . . . .	29
3.2.1 Kalman-Filter . . . . .	29
3.2.1.1 Prädiktion . . . . .	30
3.2.1.2 Innovation . . . . .	31
3.2.1.3 Sequentielle Innovation . . . . .	32
3.2.1.4 Scaled Unscented Transform . . . . .	34
3.2.2 Partikel-Filter . . . . .	36
3.2.2.1 Initialisierung . . . . .	37
3.2.2.2 Prädiktion . . . . .	37
3.2.2.3 Innovation . . . . .	38
3.3 Kaskadierte Klassifikatoren . . . . .	40
3.3.1 Trainingsphase . . . . .	41
3.3.2 Anwendung . . . . .	46
3.4 Region Growing . . . . .	46
3.4.1 Algorithmus und Varianten . . . . .	46
3.4.2 Multi-Dynamic Region Growing . . . . .	47
<b>4 Modellgenerierung</b>	<b>51</b>
4.1 Manuelle Modellgenerierung . . . . .	51

4.2	Automatische Modellgenerierung . . . . .	54
4.2.1	Identifizierung des OOI . . . . .	56
4.2.2	Sensordatenreduktion . . . . .	56
4.2.3	Merkmalsdetektion . . . . .	58
4.2.4	Aktualisierung des 3D-Merkmalmodells . . . . .	65
<b>5</b>	<b>Fahrzeugeterkennung und -verfolgung</b>	<b>71</b>
5.1	Objekterkennung durch rekursive Zustandsschätzung . . . . .	72
5.1.1	Filterinitialisierung . . . . .	73
5.1.2	Visuelle Objekterkennung mittels Kalman-Filter . . . . .	74
5.1.2.1	Prädiktion . . . . .	75
5.1.2.2	Innovation . . . . .	77
5.1.3	Objekterkennung mittels Partikel-Filter . . . . .	78
5.1.3.1	Prädiktion . . . . .	78
5.1.3.2	Innovation . . . . .	81
5.1.3.3	Filterergebnis zur Fahrzeugführung . . . . .	97
5.1.3.4	Kompatibilität mit mehreren Kameras . . . . .	97
5.1.4	Kombinierte Objekterkennung mit Partikel- und Kalman-Filter	98
5.2	Aufmerksamkeitssteuerung . . . . .	100
5.2.1	Aktive Blickrichtungssteuerung . . . . .	100
5.2.2	Hardwarebasierte AOI einer Kamera . . . . .	100
5.2.3	Softwarebasierte AOI einer Kamera . . . . .	102
5.3	Berechnung eines Qualitätsmaßes . . . . .	105
5.3.1	Auswertung der Filter-Kovarianzmatrix . . . . .	105
5.3.2	Merkmalsauswertung mit definierten Schwellwerten . . . . .	106
5.3.3	Trainieren eines Klassifikators . . . . .	107
5.4	Fahrzeugführung . . . . .	112
5.4.1	Spurgenerierung . . . . .	112
5.4.2	Quer- und Längsregelung . . . . .	114
<b>6</b>	<b>Ergebnisse</b>	<b>117</b>
6.1	Versuchsfahrzeuge . . . . .	117
6.1.1	Selbstfahrende Versuchsfahrzeuge . . . . .	117
6.1.2	Führungsfahrzeuge . . . . .	124
6.2	Modellgenerierung . . . . .	125
6.2.1	Manuelle Modellgenerierung . . . . .	125
6.2.2	Automatische Modellgenerierung . . . . .	130
6.2.3	Manuelle vs. automatische Modellgenerierung . . . . .	130
6.3	Fahrzeugeterkennung und -verfolgung . . . . .	133
6.3.1	Fahrzeugeterkennung und -verfolgung bei Tag . . . . .	133
6.3.2	Fahrzeugeterkennung und -verfolgung bei Nacht . . . . .	139
6.4	Beurteilung des Gesamtsystems . . . . .	146
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>151</b>
7.1	Stand der Arbeit . . . . .	151
7.2	Weiterentwicklungen . . . . .	153

<b>A Anhang</b>	<b>155</b>
A.1 Sensorik Datenblätter . . . . .	155
A.1.1 Abstandsgebende Sensoren . . . . .	156
A.1.2 Bildgebende Sensoren . . . . .	157
A.1.3 Inertiale Sensoren . . . . .	160
A.2 Kalibrierungsergebnisse der intrinsischen Kameraparameter . . . . .	161
A.3 Farbräume . . . . .	165
A.3.1 RGB-Farbraum . . . . .	165
A.3.2 HSV-Farbraum . . . . .	166
A.3.3 Umwandlung vom RGB- in den HSV-Farbraum . . . . .	167
A.4 Einspurmodell für vorderachsgelenkte Fahrzeuge . . . . .	168
A.5 Visuelle Markererkennung . . . . .	170
A.5.1 Detektionsalgorithmus . . . . .	170
A.5.2 Bewertung . . . . .	177
A.6 Inbetriebnahme der Software . . . . .	179
 <b>Index</b>	 <b>184</b>
 <b>Abbildungsverzeichnis</b>	 <b>185</b>
 <b>Tabellenverzeichnis</b>	 <b>190</b>
 <b>Literaturverzeichnis</b>	 <b>191</b>

# Symbolverzeichnis

## Abkürzungen

ACC	Abstandsregeltempomat, engl. <i>Adaptive Cruise Control</i>
AdaBoost	Adaptive Boosting
ADAS	Fahrerassistenzsystem, engl. <i>Advanced Driver Assistance System</i>
AMOR	Autonomous Mobile Outdoor Robot
AOI	Interessenbereich, engl. <i>Area of Interest</i>
BAAINBw	Bundesamt für Ausrüstung, Informationstechnik und Nutzung der Bundeswehr
BWB	Bundesamt für Wehrtechnik und Beschaffung
CAD	Computer Aided Design
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
DAB	Discrete AdaBoost
DARPA	Defense Advanced Research Projects Agency
DGPS	Globales Positionsbestimmungssystem mit Differentialsignal, engl. <i>Differential Global Positioning System</i>
EDA	Elektronische Deichsel für landwirtschaftliche Arbeitsmaschinen
EDAUG	Elektronische Deichsel für landwirtschaftliche Arbeitsmaschinen mit Umfeldsensorik und zusätzlichen Geoinformationen
ELROB	European Land Robot Trial
ERICard	Emergency Response Intervention Card
F&T	Forschung und Technologie
FAST	Features from Accelerated Segment Test
FOV	Sichtfeld, engl. <i>Field of View</i>
GAB	Gentle AdaBoost
GLONASS	Globales Satellitennavigationssystem, russ. <i>Globalnaja nawigazionnaja sputnikowaja sistema</i>
GNSS	Globales Navigationssatellitensystem
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HOG	Histograms of Oriented Gradients
IMU	Inertiale Messeinheit, engl. <i>Inertial Measurement Unit</i>
INS	Inertiales Navigationssystem, engl. <i>Inertial Navigation System</i>
KIT	Karlsruher Institut für Technologie
Ko-FAS	Kooperative Fahrzeugsicherheit
LB	Logit Boost
LBP	Local Binary Pattern
LCA	Spurwechselassistent, engl. <i>Lane Change Assistant</i>

LiDAR	Light Detection And Ranging
LKAS	Spurhalteassistenzsystem, engl. <i>Lane Keeping Assist System</i>
Lkw	Lastkraftwagen
MarVEye-8	Multifokales aktives / reaktives Fahrzeug-Auge der 8. Generation, engl. <i>Multifocal active / reactive Vehicle Eye 8<sup>th</sup> Generation</i>
MuCAR-3	Munich Cognitive Autonomous Robot Car 3 <sup>rd</sup> Generation
MuCAR-4	Munich Cognitive Autonomous Robot Car 4 <sup>th</sup> Generation
NAVSTAR	Navigational Satellite Timing and Ranging
OOI	Objekt von Interesse, engl. <i>Object of Interest</i>
Pkw	Personenkraftwagen
PROMETHEUS	PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety
RAB	Real AdaBoost
Radar	Radio Detection And Ranging
RMMV	Rheinmetall MAN Military Vehicles
RMSE	Root Mean Square Error
RVM	Relevance Vector Machine
SARTRE	Safe Road Trains for the Environment
SidCell	Scale Invariant Descriptive Cells
SIFT	Scale-Invariant Feature Transform
StrAsRob	Straßentransport mit Assistenzfunktionen von Robotern
StVZO	Straßenverkehrs-Zulassungs-Ordnung
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TAS	Institut für Technik Autonomer Systeme
TTL	Transistor-Transistor-Logik
TULF	Technologieträger Unbemanntes Landfahrzeug
UKF	Unscented Kalman-Filter
UMRR	Universal Medium Range Radar
UMTS	Universal Mobile Telecommunications System
UTM	Universal Transverse Mercator
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VaMoRs	Versuchsfahrzeug für autonome Mobilität und Rechnersehen
VaMP	VaMoRs-Pkw
VIAC	VisLab Intercontinental Autonomous Challenge
WDF	Wahrscheinlichkeitsdichtefunktion

## Konventionen

$(u, v)$	Bildkoordinate
$(\cdot)^*$	Prädizierte Werte
$\alpha$	Skalierungskoeffizient / Winkel
$\beta$	Skalierungskoeffizient / Winkel
$\dot{\psi}$	Gierrate / Winkelgeschwindigkeit ( $^\circ/s$ )
$\phi$	Rollwinkel ( $^\circ$ )
$\psi$	Gierwinkel / Bahnazimutwinkel ( $^\circ$ )
$\theta$	Nickwinkel ( $^\circ$ )
$h$	(Relative) Fahrzeughöhe
$\lambda$	Lenkwinkel ( $^\circ$ )
<b>c</b>	Bildhauptpunkt $\mathbf{c} = [c_u, c_v]$
<b>e</b>	Fehler
<b>F</b>	Farbbild
<b>f</b>	Bildweite
<b>H</b>	Homogene Transformationsmatrix
$I_B$	Binärbild
$I_\Delta$	Gradientenbild
$I$	Intensitätsbild / Grauwertbild
$I_L$	Linienintegralbild
<b>K</b>	Kalman-Verstärkungsmatrix
<b>k</b>	Verzeichnungskoeffizienten
$k$	Diskreter Zeitpunkt
$\kappa$	Skalierungskoeffizient
$l$	Radstand / Achsabstand (m)
<b>m</b>	Fahrzeugmerkmale
<b>MP</b>	Momentanpol
<b>P*</b>	Vorhergesagte Schätzfehlerkovarianzmatrix
<b>P</b>	(Schätzfehler-)Kovarianzmatrix
$\text{obj} \mathbf{P}_{\text{ego}}$	Ego-relative Fahrzeugpose
<b>p</b>	3D-Punkt $\mathbf{p} = [x, y, z]$
<b>p</b>	2D-Punkt $\mathbf{p} = [u, v]$
$p(\cdot)$	(Bedingte) Wahrscheinlichkeit / Wahrscheinlichkeitsdichtefunktion
<b>Q</b>	Systemfehlerkovarianzmatrix
<b>R</b>	Messfehlerkovarianzmatrix
$r$	Radius eines Kreises / Zylindrischer Abstand
<b>S</b>	Kartesisches Koordinatensystem
<b>s</b>	Störvektor
$s$	Scherungskoeffizient, welcher den Winkel zwischen den zwei Achsen eines Kamerasensorchips beschreibt. Bei rechtwinkligen Pixeln ist dieser null.
$t$	Zeit

$u$	Horizontale Bildkoordinate
$v$	Systemrauschvektor
$v$	Vertikale Bildkoordinate / Geschwindigkeit in (m/s)
$w$	Gewichtungsvektor
$\hat{x}$	Korrigierter Zustandsvektor
$x^*$	Vorhergesagter Zustandsvektor
$x$	Sigma punkt
$x$	Zustandsvektor
$Y$	Endliche Menge von Messvektoren
$y^*$	Vorhergesagter Messvektor
$y$	Messvektor

# 1 Einleitung

## Inhalt

---

1.1 Forschungsbeitrag . . . . .	3
1.2 Struktur der Arbeit . . . . .	4

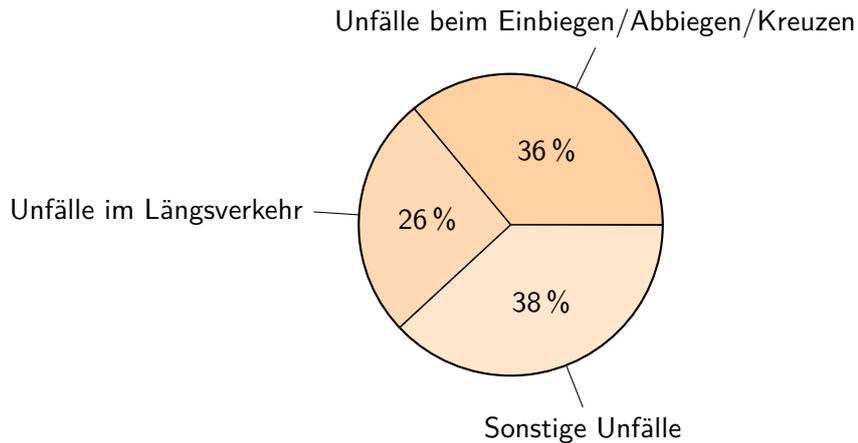
---

Heutzutage existiert kein Fahrzeug im öffentlichen Verkehr, das ohne die Hilfe und Überwachung eines Menschen von alleine fahren und alle gängigen Aufgaben eines menschlichen Fahrzeugführers selbstständig durchführen kann. Die Wahrnehmung, Interpretation und das situationsbedingte Agieren eines Menschen ist derart komplex, dass die industrielle und wissenschaftliche Forschung in kleinen Schritten versucht, bestimmte Teilfähigkeiten eines Menschen in Hardware- und Softwarelösungen umzusetzen.

In der Automobilindustrie wird von sogenannten Fahrerassistenzsystemen, engl. *Advanced Driver Assistance Systems* (ADAS), gesprochen, die derzeit zur Erhöhung des Fahrkomforts und der Sicherheit entwickelt werden. Langfristig gesehen wird ebenfalls eine ökonomische Verbesserung angestrebt [Lammert et al., 2014]. Bezüglich der Erhöhung der Verkehrssicherheit sind Unfallstatistiken die Informationsgrundlage für die Wahl der zu entwickelnden ADAS (vgl. Reif [2010]). Das Statistische Bundesamt erfasste 3459 Verkehrstote für das Jahr 2015 in Deutschland. Davon wurden 47 Prozent durch Personenkraftwagen (Pkw) getötet. Besonders hoch lag die Unfallrate beim Einbiegen, Abbiegen und Kreuzen. Des Weiteren zeigt das Diagramm der Abbildung 1.1 eine hohe Unfallrate durch Auffahrunfälle im Längsverkehr. Aufgrund dieser Informationsgrundlage des Statistischen Bundesamtes und zur Verbesserung des Fahrkomforts erfolgte unter anderem die Entwicklung des Abstandsregeltempomaten, engl. *Adaptive Cruise Control* (ACC), und des Spurhalteassistenzsystems, engl. *Lane Keeping Assist System* (LKAS). Hierbei regelt das ACC in Abhängigkeit des Abstands und der Geschwindigkeit des Vorderfahrzeuges eine vom Fahrer vorgegebene Maximalgeschwindigkeit ein. Der LKAS übernimmt dagegen die Querregelung innerhalb der eigenen Fahrspur und entlastet damit den Fahrer.

Die Norm SAE J3016 [2018] definiert den Autonomiegrad von Verkehrsfahrzeugen in fünf Stufen:

- Level 0 - Der Fahrer muss eigenständig fahren.
- Level 1 - Einzelne ADAS unterstützen bei der Längs- oder Querführung.



**Abbildung 1.1:**

Statistik über die am häufigsten vorkommenden Unfallarten auf deutschen Straßen. Die Prozentangaben sind bezogen auf 305 659 Straßenverkehrsunfälle von Fahrzeugen mit Personenschaden im Jahr 2015 [Statistisches Bundesamt Deutschland, 2015, S. 66].

- Level 2 - ADAS übernehmen gleichzeitig die Längs- und Querführung.
- Level 3 - Automatisiertes Fahren für definierte einfache Fahraufgaben mit menschlichem Fahrer als Rückfallebene.
- Level 4 - Hochautomatisiertes Fahren, bei dem ADAS dynamische Fahraufgaben übernehmen und das Fahrzeug in einen sicheren Zustand bringen, auch wenn der Fahrer auf die Aufforderung, die Fahrzeugführung zu übernehmen, nicht reagiert.
- Level 5 - Vollautomatisiertes Fahren, bei dem das System unabhängig vom Fahrweg sowie der Umgebungsbedingungen alle gängigen dynamischen Fahraufgaben wie ein Mensch durchführt.

Im Jahr 2017 entsprechen ADAS den Levels 2-3 und können dem Fahrer im Stau, auf Autobahnen und beim Einparken bei der Längs- und Querführung unterstützen, soweit bestimmte Umgebungsbedingungen vorliegen und der Fahrer dies erwünscht [Watzenig und Horn, 2017].

Sowohl ein ADAS als auch ein Mensch führen ihre Aktionen durch die Interpretation von Umgebungsinformationen durch. Ein Mensch benutzt für die allgemeine Umgebungswahrnehmung seine fünf Sinne: Sehen, Hören, Riechen, Schmecken und Tasten. Die gewonnenen Umgebungsinformationen werden mit dem gelernten Modellwissen interpretiert, um anschließend situationsbedingt zu agieren. Im Gegensatz zum Menschen verwenden ADAS Informationen von technischen Geräten bzw. Sensoren.

## 1.1 Forschungsbeitrag

Diese Arbeit befasst sich mit einem Folgeassistenten, der ein autonomes und spurtreues Folgen eines ausgewählten Führungsfahrzeuges ermöglicht. Der Schwerpunkt liegt auf der kontinuierlichen Wahrnehmung von Fahrzeugen mit den nachfolgenden Zielen:

- Funktionsfähigkeit unabhängig vom Fahrzeugtyp (Pkw, Lkw), Terrain (städtisch, unstrukturiert), Helligkeitsgrad und künstlichen Markern (Fahrbahnmarkierungen)
- Funktionsfähigkeit zu jeder Tageszeit (Tag, Dämmerung, Nacht)
- Funktionsfähigkeit bei teilweisen Verdeckungen, bspw. durch Regen, Nebel, Schnee, Scheibenwischer, andere Verkehrsteilnehmer oder Vegetationen
- Datenfusion auf Merkmalsebene mit beliebig vielen Kameras, unterschiedlichen Kamerateypen (Tageslicht, Wärme, Low-Light) und abstandsgebenden Sensoren (LiDAR und Radar)
- Schätzung der relativen 6D-Pose, der Geschwindigkeit, der Beschleunigung und des Lenkwinkels des Führungsfahrzeuges über die Zeit
- Aufmerksamkeitssteuerung zur Reduktion der zu verarbeitenden Sensordaten
- Automatische Generierung von fahrzeugspezifischen 3D-Merkmalmodellen in Echtzeit

Diese Arbeit beschreibt ein modellbasiertes Fahrzeugerkennungssystem, das die aufgezählten Ziele weitgehend erfüllt. Das Fahrzeugerkennungssystem verwendet rekursive Schätzalgorithmen sowie den sogenannten 4D-Ansatz von Dickmanns [2007] und basiert auf den Vorarbeiten von Manz [2013].

Das System wurde in vielseitigen Praxistests mit verschiedenen Fahrzeugtypen und bei unterschiedlichen Helligkeits-, Witterungs- und Geländebedingungen ausgiebig sowie erfolgreich getestet. Insgesamt ermöglichte die Fahrzeugerkennung das autonome Folgen über mehrere hundert Kilometer. Dabei fuhr das Folgefahrzeug bis zu 100 km/h autonom und regelte die Lenkung, das Gaspedal, die Bremse und die Schaltung selbstständig.

Einige Themengebiete dieser Arbeit wurden bereits auf internationalen Konferenzen veröffentlicht [Fries et al., 2013, Fries und Wuensche, 2014, 2015, 2016, Fries et al., 2017].

## 1.2 Struktur der Arbeit

Die vorliegende Arbeit ist in sieben Kapitel gegliedert. Diese Kapitel haben folgende Inhalte:

**Kapitel 1** beschreibt die Ziele und den Forschungsbeitrag dieser Arbeit.

**Kapitel 2** gibt einen Überblick über relevante Forschungsergebnisse und -projekte im Bereich der Objekterkennung und -verfolgung, der Modellgenerierung und von selbstfahrenden Fahrzeugen.

**Kapitel 3** enthält Grundlagen für das Verständnis dieser Arbeit.

**Kapitel 4** erläutert eine manuelle sowie eine automatische Vorgehensweise, mit der aus perspektivisch unterschiedlichen Sensordaten ein fahrzeugspezifisches 3D-Merkmalmodell erstellt werden kann.

**Kapitel 5** beschreibt die Erkennung und Verfolgung eines Fahrzeuges durch einen kontinuierlichen Abgleich der sensorischen Wahrnehmung mit einer bekannten Modellvorstellung. Hierbei entspricht das Modellwissen einem fahrzeugspezifischen 3D-Merkmalmodell. Des Weiteren erfolgt die Beschreibung, wie zeitdiskrete Schätzwerte über das Führungsfahrzeug zur Aufmerksamkeitssteuerung, Qualitätseinschätzung sowie Fahrzeugführung verwendet werden können.

**Kapitel 6** stellt die Versuchsfahrzeuge dieser Arbeit vor und bewertet die Modellgenerierung und Fahrzeugerkennung.

**Kapitel 7** fasst die Resultate dieser Arbeit zusammen und geht auf mögliche Weiterentwicklungen ein.

## 2 Stand der Forschung

### Inhalt

---

2.1	Autonomes Folgen im Konvoi . . . . .	6
2.1.1	Historische Platooning-Projekte . . . . .	6
2.1.2	Aktuelle Forschungsprojekte des BAAINBw . . . . .	9
2.1.3	Serienreife Assistenzsysteme . . . . .	10
2.2	Fahrzeugerkennung, -klassifizierung und -verfolgung . . . . .	11
2.2.1	Technische Komponenten . . . . .	11
2.2.2	Markerbasierte Verfahren . . . . .	12
2.2.3	Markerlose Verfahren . . . . .	13

---

Der Schwerpunkt dieser Arbeit liegt auf der modellbasierten Wahrnehmung von Fahrzeugen mit dem übergeordneten Ziel des autonomen spurtreuen Folgen hinter einem Führungsfahrzeug. Dieses Kapitel benennt und beschreibt forschungsrelevante Literatur. Zu Beginn wird auf die Forschung im Bereich des autonomen Nachfahrens eines manuell geführten Fahrzeuges eingegangen. Anschließend erfolgt die Betrachtung von wissenschaftlichen Veröffentlichungen, die sich mit der Wahrnehmung von Fahrzeugen beschäftigen.

Forschungsarbeiten über selbstfahrende Fahrzeuge, die selbstständig zu einem Zielpunkt navigieren, werden nicht berücksichtigt. Dazu zählen zum einen die Wettbewerbe der Defense Advanced Research Projects Agency (DARPA): Grand Challenge [Thrun et al., 2006] und Urban Challenge [Buehler et al., 2009, Urmsen et al., 2008, Montemerlo et al., 2008, Kammel et al., 2008]. Des Weiteren fallen darunter die Projekte: Bertha Benz Fahrt [Ziegler et al., 2014], Google Self-Driving Cars<sup>1</sup>, Kooperative Fahrzeugsicherheit (Ko-FAS) [Westhofen et al., 2012], und das mit über 749 Millionen Euro geförderte europäische Forschungs- und Entwicklungsprojekt EUREKA-PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety (PROMETHEUS) [Catling und McQueen, 1991, Dickmanns et al., 1994, Dickmanns, 2007].

---

<sup>1</sup>[www.google.com/selfdrivingcar](http://www.google.com/selfdrivingcar)

## 2.1 Autonomes Folgen im Konvoi

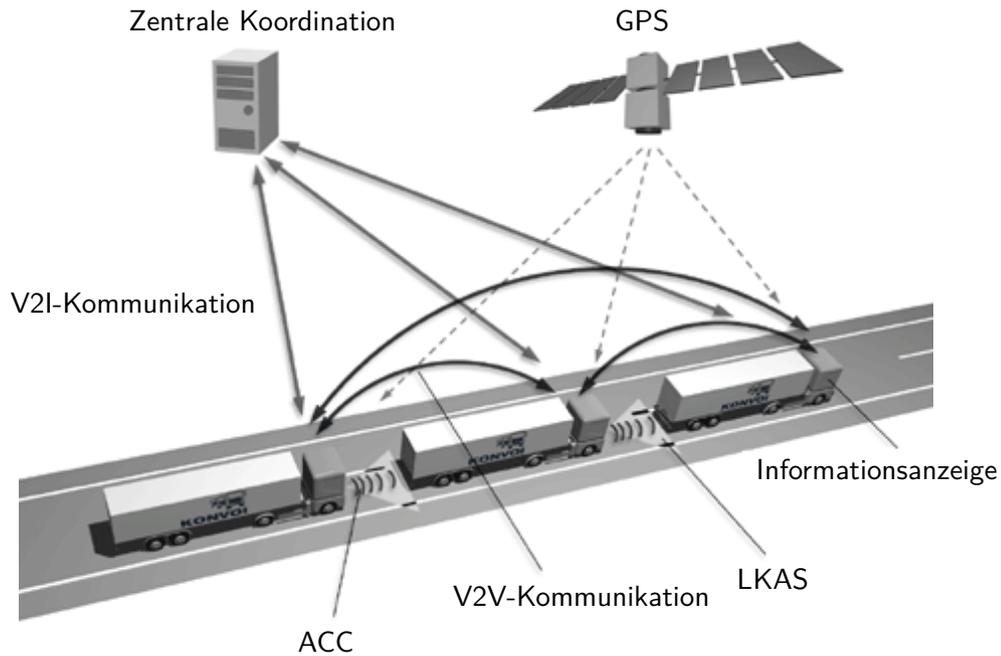
Die Forschung an (semi-)autonomen Konvoifahrzeugen begann in den achtziger Jahren. Im Nachfolgenden wird auf fünf historische Forschungsprojekte näher eingegangen.

### 2.1.1 Historische Platooning-Projekte

Im Jahre 1996 startete das europäische Forschungsprojekt PROMOTE CHAUFFEUR mit dem wissenschaftlichen Fokus auf der elektronischen Kopplung von schweren Lastkraftwagen (Lkw). Das Projekt konzentrierte sich hierbei auf den Einsatz auf Autobahnen. Das Ziel lag in der Reduzierung der Fahrzeugabstände bzw. der sicherheitsbedingten Minimalabstände. Der Sicherheitsabstand zwischen den 18 bis 41 Tonnen schweren Mercedes-Benz Lkw vom Typ Actros konnte auf bis zu 12 Meter reduziert werden. Das Gesamtsystem basiert auf einer Funk- und Infrarotverbindung der Konvoiteilnehmer untereinander und wurde 1999 erfolgreich auf der österreichischen Autobahn präsentiert. Diverse Weiterentwicklungen fanden zwischen 2000 und 2002 im Nachfolgeprojekt „Chauffeur Assistant“ statt. In diesem wurde das bestehende Konvoisystem flexibler gemacht, sodass auf Autobahnen ein beliebiges Führungsfahrzeug ausgewählt und mit einem kleineren Fahrzeugabstand gefahren werden konnte. [Bargende und Wiedemann, 2001, S. 714 f.]

Das Forschungsprojekt KONVOI beschäftigte sich von 2005 bis 2009 mit der Entwicklung und Untersuchung des Einsatzes von bis zu vier elektronisch gekoppelten Lkw. Die Projektteilnehmer stammten größtenteils von der RWTH Aachen, der MAN Nutzfahrzeug AG und der Wabco Development GmbH. Das interdisziplinär entwickelte ADAS konnte die Längs- und Querführung von bis zu vier Lkw steuern. Mit abstandsgebender Sensorik wurde der Abstand zum Vorderfahrzeug eingeregelt und LKAS regelten die Querführung innerhalb einer Fahrspur. Das Global Positioning System (GPS) wurde zur globalen Positionsbestimmung verwendet. Alle Fahrzeuge verfügten über eine 2,4 GHz Vehicle-to-Vehicle (V2V)-Breitbandkommunikation. Zusätzlich existierte ein zentraler Server, der den Konvoiteilnehmern über eine Vehicle-to-Infrastructure (V2I)-Mobilfunkschnittstelle (UMTS, GPRS, GSM) Koordinationsbefehle mitteilte und Statusinformationen empfing. Abbildung 2.1 veranschaulicht das Projekt, welches primär für den Einsatz auf Autobahnen entworfen wurde. Hierbei sind die angestrebten Ziele eine Steigerung des Verkehrsflusses und der Verkehrssicherheit sowie Einsparungen beim Kraftstoffverbrauch und der Ausstoßung von CO<sub>2</sub>-Emissionen. [Deuschle et al., 2010, Lank et al., 2011a,b]

Im Jahre 2009 forschte das europäische Forschungsprojekt „Safe Road Trains for the Environment (SARTRE)“ drei Jahre lang an einem Platooning-System speziell für Autobahnen. Das mit 3,8 Millionen Euro geförderte Projekt hatte zum einen das Zukunftsziel im Straßenverkehr weniger Kraftstoff zu verbrauchen, was wiederum



**Abbildung 2.1:**

Forschungsprojekt KONVOI: Einsatz von elektronisch gekoppelten Lkw-Konvois auf Autobahnen. Bilddarstellung nach Lank et al. [2011a].

eine Verringerung der CO<sub>2</sub>-Emissionen impliziert. Zum anderen wurde eine Erhöhung der Verkehrssicherheit angestrebt. Die Forschungstests wurden mit unterschiedlichen Fahrzeugen der Marke Volvo durchgeführt. Die selbstfahrenden Folgefahrzeuge waren Pkw und sollten einem manuell gefahrenen Lkw kooperativ sowie in kleinen Konvoiabständen nachfahren, sodass aufgrund der damit verbesserten Aerodynamik der Kraftstoffverbrauch sinkt. Für die Realisierung eines kooperativen autonomen Konvois erfolgte in allen Fahrzeugen die Integration eines Computers und einer V2V-Kommunikation. Zusätzlich wurden die Folgefahrzeuge mit Kameras sowie Radar- (Radio Detection And Ranging) und LiDAR- (Light Detection And Ranging) Sensoren ausgestattet. Mit diesen Sensoren und über Funkkommunikation übertragenen Fahrzeugdaten konnte jedes Folgefahrzeug eine Längs- und Querregelung zu seinem entsprechenden lokalen Führungsfahrzeug durchführen und dabei Aufschwingeffekte vermeiden. Bei den Praxistests konnte ein Sollabstand von vier Metern erfolgreich eingeregelt werden. Dabei wurden Fahrzeuggeschwindigkeiten bis zu 90 km/h erreicht. Für eine detaillierte Beschreibung des europäischen Forschungsprojektes SARTRE wird auf die Publikationen von Chan et al. [2012] und Bergenhem et al. [2012] verwiesen.

Das Karlsruher Institut für Technologie (KIT) arbeitete zwischen 2007 und 2010 an dem Projekt „Elektronische Deichsel für landwirtschaftliche Arbeitsmaschinen (EDA)“. Das Ziel war die Realisierung einer unbemannten landwirtschaftlichen Arbeitsmaschine, welche einem manuell gefahrenen Führungsfahrzeug spurtreu nachfahren kann. Ein Globales Positionsbestimmungssystem mit Differentialsignal, engl.

*Differential Global Positioning System* (DGPS), sowie eine Datenfunkverbindung zwischen zwei Traktoren bildete die Sensorbasis und ermöglichte die elektronische Kopplung im Praxisversuch. Dabei wurde ein Längs- und Querabstand vorgegeben und anhand der globalen Positionsinformationen und der daraus berechneten Fahrzeuggeschwindigkeiten eingeregelt [Zhang et al., 2010]. Eine Erweiterung des Systems fand im Forschungsprojekt „Elektronische Deichsel für landwirtschaftliche Arbeitsmaschinen mit Umfeldsensorik und zusätzlichen Geoinformationen (EDAUG)“ von 2011 bis 2014 statt. Dabei stand eine gesetzeskonforme Risikoanalyse des bestehenden Konvoi-Systems, die Erfüllung von rechtlichen Sicherheitsanforderungen und die Steigerung des Automatisierungsgrades im Vordergrund. Einerseits wurde die bestehende Sensorausstattung mit einem 2D-Laserscanner zur Umfeldüberwachung ergänzt, andererseits wurden zusätzlich ortsbezogene Geoinformationen zur proaktiven Hindernisvermeidung verwendet. [Jahnke et al., 2012]

Die European Land Robot Trial (ELROB) existiert seit 2006 und ist eine mehrtägige Veranstaltung, bei der europäische Teams versuchen, vorgegebene Aufgaben im Bereich der Robotik zu bewältigen. Eine gängige Aufgabe ist das autonome Folgen im Konvoi. Der Konvoi besteht dabei aus mindestens zwei Fahrzeugen, von denen das Führungsfahrzeug manuell gefahren werden darf, das Folgefahrzeug allerdings selbstständig fahren bzw. folgen muss. Der Fahrer des Führungsfahrzeuges erhält eine digitale Karte bestehend aus UTM-Wegpunkten (Universal Transverse Mercator), welche anschließend abzufahren sind. Das Folgefahrzeug folgt dabei dem Führungsfahrzeug eigenständig. Das zu befahrende Gelände ist überwiegend unstrukturiertes Terrain. Insgesamt wurde das Konvoiszenario 2008, 2010, 2012, 2016 und 2018 angeboten. Zu den besten Teams zählten 2008 die Leibniz Universität Hannover mit dem Fahrzeug RTS-HANNA [Langerwisch et al., 2010], 2010 die Universität Siegen mit dem Fahrzeug Autonomous Mobile Outdoor Robot (AMOR) [Schneider et al., 2010a] und in allen Jahren die Universität der Bundeswehr München mit Munich Cognitive Autonomous Robot Car 3<sup>rd</sup> Generation (MuCAR-3).

Der Vollständigkeit halber werden abschließend für dieses Unterkapitel weitere nennenswerte Platooning-Projekte referenziert: VisLab Intercontinental Autonomous Challenge (VIAC) [Broggi et al., 2012], PATH-Platooning [Shladover, 2007], SCANIA-Platooning [Alam, 2011], Peloton Technology Inc.-Platooning [Gaudet, 2014, S. 33], Energy ITS [Tugawa et al., 2011], European Truck Platooning Challenge<sup>2</sup> und die Grand Cooperative Driving Challenge (GCDC)<sup>3</sup>. Eine übersichtliche Beschreibung dieser Platooning-Projekte geben Tugawa et al. [2016] und Bergenhem et al. [2012].

---

<sup>2</sup>[www.eutruckplatooning.com](http://www.eutruckplatooning.com)

<sup>3</sup>[www.gcdc.net](http://www.gcdc.net)

### 2.1.2 Aktuelle Forschungsprojekte des BAAINBw

Das Bundesamt für Ausrüstung, Informationstechnik und Nutzung der Bundeswehr (BAAINBw), ehemals Bundesamt für Wehrtechnik und Beschaffung (BWB), fördert seit vielen Jahren die Forschung und Entwicklung im Bereich der Automatisierung von Transportfahrzeugen für militärische Konvois. Aus militärischer Sicht soll die kognitive Belastung und Gefährdung von Soldaten im Einsatz verringert werden. Ein weiteres Ziel ist die durch eine reduzierte Fahrzeugpanzerung mögliche Erhöhung der maximalen Nutzlast bei unbemannten Transportfahrzeugen. Aufgrund dieser und weiterer Gründe veranlasste das BAAINBw größere Investitionen in die Forschung und Entwicklung von (teil-)autonomen Robotern. Für nähere Informationen wird auf Petermann und Grünwald [2011] verwiesen.

**Technologieträger Unbemanntes Landfahrzeug (TULF)** ist eine Forschung und Technologie (F&T)-Studie, die vom BAAINBw gefördert wird. Das Ziel der Studie ist die Portierung von bestehenden Autonomiefunktionen auf einen Militär-Lkw vom Typ Rheinmetall MAN Military Vehicles (RMMV) HX58. Eine Autonomiefunktion ist das spurtreue Folgen eines manuell gefahrenen Führungsfahrzeuges. Hierbei ist die Folgefunktion bzw. die visuelle Fahrzeugerkennung markerbasiert oder markerlos verwendbar. Die primären Projektteilnehmer sind Diehl BGT Defence, Rheinmetall Landsysteme und die Universität der Bundeswehr München. Das Projekt wurde im Jahr 2013 erfolgreich auf einem Standortübungsplatz in unstrukturiertem Terrain demonstriert (vgl. Fassbender et al. [2014]).

Zwei Jahre später erfolgte eine zusätzliche Weiterentwicklung des TULF-Projektes, indem der bestehende Folgeassistent um eine sensorbasierte Wegerkennung ergänzt wurde. Die Erkennung des Weges ermöglicht bei Sichtverlust des Führungsfahrzeuges einen vorübergehenden Übergang zur selbstständigen Navigation.

**Straßentransport mit Assistenzfunktionen von Robotern (StrAsRob)** ist eine im Jahr 2012 gestartete F&T-Studie und baut auf den Erfahrungen des TULF-Projektes auf. Das übergeordnete Ziel blieb die Entlastung des Militärkraftfahrers durch Assistenzfunktionen. Allerdings wurde im Vergleich zu TULF seriennähere Sensorik verwendet. Anstelle zweier Velodyne Laserscanner – mit jeweils 32 Laserdioden – kam nun ein 8-Zeilen Laserscanner vom Typ Ibeo LUX 8L und ein Radarsensor aus dem Automotivebereich zum Einsatz. Die Verwendung von diversen Kameras blieb bestehen. Das Projekt konnte Ende 2015 erfolgreich abgeschlossen werden.

Weiterführend wurde nach Projektende beschlossen, dass die Versuchsträger aus den Projekten TULF und StrAsRob zusammen an der militärischen ELROB 2016 teilnehmen. Dort präsentierten beide Lkw einen autonomen Konvoi und erzielten nach MuCAR-3 die zweitbeste Platzierung im Wettbewerb.

### 2.1.3 Serienreife Assistenzsysteme

Dieses Kapitel beschreibt eine kleine Auswahl von serienreifen Assistenzsystemen, die besonders für (semi-)autonomes Fahren geeignet sind. Der Einsatzbereich der genannten ADAS ist auf gut ausgebaute Straßen (Autobahnen, Schnellstraßen) mit unbeschädigten Straßenmarkierungen begrenzt.

Das ACC ist ein serienreifes ADAS, das selbstständig die Längsführung übernimmt und damit aktiv in die Fahrdynamik eingreift. Die gewünschte maximale Fahrzeuggeschwindigkeit wird durch den Fahrer vorgegeben. Zur Vermeidung von Auffahrunfällen werden Radar- und LiDAR-Sensoren verwendet. Die Sensoren liefern Abstands- und auch Geschwindigkeitsinformationen zu vorausfahrenden Fahrzeugen. Mit den Informationen wird ein geschwindigkeitsabhängiger Sicherheitsabstand zum vorausfahrenden Fahrzeug bestimmt und eingeregelt. Die serienmäßige Einführung in den Automobilmarkt begann schon vor der Jahrtausendwende. Anfangs durften ACC-Assistenzsysteme nur bei einer Mindestgeschwindigkeit von 30 km/h aktiviert werden (vgl. ISO 15622 [2018]). Heutzutage dürfen sowie existieren ACC-Systeme, welche sogar bei Stillstand des vorausfahrenden Fahrzeugs hinter diesem selbstständig anhalten („Follow-To-Stop“) sowie bei dessen Weiterfahrt wieder losfahren („Stop&Go“). Detaillierte Informationen finden sich in Winner et al. [2015] und Eigel [2010].

Ein weiteres serienreifes ADAS ist das LKAS. Dieses verwendet optische Sensoren zur Erkennung der Fahrbahnmarkierungen und ermöglicht dadurch die Querführung innerhalb der eigenen Fahrspur. Die Kombination vom ACC mit dem LKAS erlaubt das automatisierte Folgen eines Führungsfahrzeuges, ohne dass der Mensch das Lenkrad, die Bremse oder das Gaspedal bedienen muss. Eine übersichtliche Beschreibung über die angebotenen LKAS der Automobilhersteller Honda, Lexus und VW findet sich in Harrer und Pfeffer [2016] auf Seite 537.

Der Spurwechselassistent, engl. *Lane Change Assistant* (LCA), überwacht den benachbarten und rückwärtigen Straßenverkehr und warnt den Fahrer bei Überholmanövern vor einer Kollision. Es gibt Erweiterungen, in der das System einen Fahrstreifenwechsel und sogar einen vollständigen Überholvorgang durchführen kann. Dies beschreibt Rauch et al. [2012] mit der Verwendung von digitalen Karten, Laserscannern, Radarsensoren, Kameras, Inertialsensorik und DGPS. Hierbei erfolgt eine aktive Umfelderkennung inkl. Fahrzeugerkennung und Ego-Lokalisierung innerhalb der Fahrspur.

Zur Vermeidung von Auffahrunfällen bieten zahlreiche Automobilhersteller sogenannte Notbremsassistenten an. Diese erkennen Verkehrsteilnehmer wie Fahrzeuge, Radfahrer und Fußgänger, berechnen eine Kollisionswahrscheinlichkeit mit dem eigenen Fahrzeug und leiten bei Bedarf eine Notbremsung ein (vgl. Kämpchen et al. [2010]). Einige Systeme kontaktieren sogar nach einer Notbremsung bzw. einem Verkehrsunfall den Notruf von selbst. Darunter zählen unter anderem die Systeme

me „BMW Assist Advanced eCall“, „Ford SYNC Emergency Assistance“, „Citroen Localized Emergency Call“ und „Peugeot Connect SOS“.

Im Jahr 2017 bieten Automobilhersteller bereits ADAS mit einem Autonomiegrad der Stufe 2-3 an [Watzenig und Horn, 2017]. Hierbei bezieht sich die Stufenangabe auf die Norm SAE J3016 [2018]. Zu den serienreifen Systemen gehören unter anderem der „Autopilot“ von Tesla, der „Driving Assistant Plus“ von BMW, der „Drive Pilot“ von Mercedes-Benz, der „Traffic Jam Assist“ von Audi, „Honda Sensing“ sowie der „ProPILOT“ von Nissan. Primär übernehmen die genannten Systeme das Einparken eines Fahrzeuges und führen in wenigen ausgewählten Verkehrssituationen (Stau, Autobahn) sowohl die Längs- als auch die Querführung durch. Weitere Informationen finden sich in Watzenig und Horn [2017], Neckermann [2017] und Dudenhöffer [2016] ab Seite 169.

## 2.2 Fahrzeugerkennung, -klassifizierung und -verfolgung

Nachfolgend werden die eng verzahnten Themen der Erkennung eines Fahrzeuges sowie dessen Klassifizierung und Verfolgung beschrieben. Zu Beginn wird auf die unterschiedliche Verwendung der technischen Komponenten eingegangen. Im Anschluss daran werden markerbasierte und markerlose Verfahren vorgestellt.

### 2.2.1 Technische Komponenten

Die meisten wissenschaftlichen Veröffentlichungen verwenden nur einen Sensor zur Objektwahrnehmung [Viola und Jones, 2001b, Sivaraman und Trivedi, 2012, 2011, Pantilie und Nedeveschi, 2010, Barth und Franke, 2010, Robert, 2009, Chen et al., 2011, Alcantarilla et al., 2008, Dominguez et al., 2011, Himmelsbach et al., 2009a]. Seit einigen Jahren ist die gleichzeitige Verwendung unterschiedlicher Sensoren typisch. Denn zum einen führt die Fusion von Sensordaten zu einem Informationsgewinn in der Umgebungswahrnehmung, und zum anderen werden die Algorithmen robuster gegen Umwelteinflüsse wie wechselhafte Wetter- und Lichtbedingungen [Schweiger et al., 2005, Kim et al., 2005, Manz et al., 2011]. Zu den gängigen Sensoren für die Umgebungswahrnehmung gehören Tageslicht- und Infrarotkameras, Ultraschallsensoren sowie Radar- und LiDAR-Systeme. Hierbei variiert der Kaufpreis der unterschiedlichen Sensortypen stark. LiDAR-Sensoren und Wärmebildkameras sind im Anschaffungspreis noch relativ teuer, während Tageslichtkameras und Ultraschallsensoren vergleichsweise günstig sind. Die häufig verwendeten Tageslichtkameras haben zum anderen den Vorteil eines niedrigeren Stromverbrauches und einer höheren Winkelauflösung im Messbereich. Ein Nachteil ist die starke Abhängigkeit von Licht- und Witterungsbedingungen. Sind die Bedingungen nicht optimal, sinkt die Bildqualität drastisch. Dagegen sind LiDAR- und Radarsensoren deutlich robuster gegen

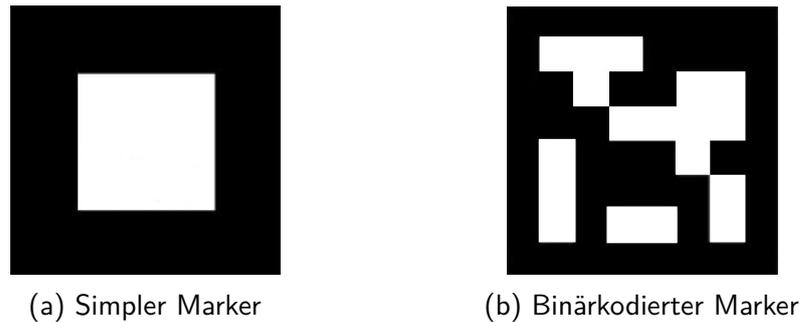
wechselhafte Licht- und Wetterbedingungen. Einen weiteren Vorteil verschaffen sie sich durch die hohe Reichweite von bis zu 200 Metern (Automotivebereich).

Abgesehen von der Umgebungswahrnehmung wird üblicherweise eine Inertiale Messeinheit, engl. *Inertial Measurement Unit* (IMU), zur Eigenbewegungsschätzung und ein Globales Navigationssatellitensystem (GNSS) zur globalen Positionsschätzung verwendet. Zu den bekanntesten GNSS gehören das US-amerikanische NAVSTAR GPS, GLONASS- (Globales Satellitennavigationssystem, russ. *Globalnaja nawigazionnaja sputnikowaja sistema*) der Russischen Föderation, Galileo der Europäischen Union und Beidou der Chinesischen Volksrepublik. Die Kombination von IMU und GNSS wird als Inertiales Navigationssystem, engl. *Inertial Navigation System* (INS), bezeichnet. Des Weiteren kann ein GNSS mit einer eingemessenen geostationären Referenzstation gekoppelt werden. Ein solches System wird als „Differential GNSS“ (z.B. DGPS) bezeichnet und erzielt Messgenauigkeiten im Zentimeterbereich.

Die Positionierung der Sensoren erfolgt in Abhängigkeit vom Anwendungsbereich geografisch stationär oder an bewegten Fahrzeugen. Im Bereich der Verkehrsüberwachung sind die Sensorsysteme geostationär in die Infrastruktur integriert. Im Automotivebereich ist eine Sensorintegration in ein Fahrzeug gängig. Beispielsweise erfolgt durch die sensorielle Wahrnehmung der Verkehrsteilnehmer eine Steuerung des eigenen Fernlichts zur Vermeidung von Blendungen des vorausfahrenden Verkehrs sowie des Gegenverkehrs. Andere Anwendungsbeispiele sind die Kollisionsvermeidung und die aktive Spurführung [Chen und Chiang, 2010, O'Malley et al., 2010]. Die geostationären Systeme haben allerdings einen großen Vorteil bei der Detektion von Objekten. Das Wissen über das Aussehen der Szene (Hintergrund) kann verwendet werden, um die Verkehrsteilnehmer (Vordergrund) mit dem sogenannten „Frame Differencing“- bzw. „Background Subtraction“-Verfahren zu trennen [Guo und Shen, 2013]. Diese Methode funktioniert allerdings nur bei einer konstanten Belichtung [Chen et al., 2011]. Insbesondere bei Dämmerung und in der Nacht verursachen die Fahrzeuglichter unterschiedliche Lichteffekte im Kamerabild, was diese Methode bei völlig fehlender Umgebungsbelichtung im Straßenverkehr unbrauchbar macht.

### 2.2.2 Markerbasierte Verfahren

In diesem Abschnitt werden markerbasierte Verfahren zur Erkennung und Klassifizierung von Fahrzeugen beschrieben. Die Verfahren benötigen Wissen über das Aussehen der künstlichen Marker, wie z. B. die Geometrie und Textur eines Schachbrettmusters oder die Anordnung von (Infrarot-)Leuchtdioden. Im Bereich der Fahrzeugerkennung kann ein gut sichtbarer Marker an der Rückseite eines Fahrzeuges angebracht werden. Die Abbildung 2.2 veranschaulicht zwei verschiedene Markertypen. Grafik 2.2a zeigt einen simplen Marker, der aus einer weißen Grundfläche und einem schwarzem Rahmen besteht. Für die Markererkennung ist hierbei das Wissen über die Geometrie – die Anordnung der vier inneren Eckpunkte – notwendig. Besteht zusätzlich das Wissen über die Entfernung der Eckpunkte zueinander, kann die Position und



**Abbildung 2.2:** Künstliche Marker.

Orientierung zwischen der Kamera und dem Marker bestimmt werden [Kato und Billingham, 1999]. Für die Erkennung von Eckpunkten existieren diverse Methoden. Ein gängiger Ansatz ist die Extraktion von Linien, deren Sortierung in horizontale und vertikale Linien und der anschließenden Suche nach Linien, die sich in einem orthogonalen Winkel schneiden [Fries und Schuhfuß, 2010]. Der Marker der Grafik 2.2b illustriert einen binärkodierten Marker, welcher eindeutig identifiziert werden kann. Dies ermöglicht die eindeutige Erkennung mehrerer Fahrzeuge. Des Weiteren eignen sich aufgeheizte Drähte für eine Fahrzeugerkennung unabhängig von der Umgebungsbeleuchtung. Für die Realisierung eines automatisierten Folgeassistenten montierten Schaub et al. [2004] einen aufheizbaren Draht am Heck eines militärischen Kettenfahrzeuges. Anschließend konnte der aufgeheizte Draht mit einer Infrarotkamera visuell im Nahbereich wiedererkannt werden. Der entwickelte Folgemodus wurde zwischen 2002 und 2003 mit einem Konvoi bestehend aus zwei Kettenfahrzeugen erfolgreich erprobt [Schaub et al., 2004, S. 163].

Für weitere Markervarianten sei auf Wagner et al. [2008], Maiero [2009] und Fiala [2005] verwiesen.

### 2.2.3 Markerlose Verfahren

Dieses Kapitel beschreibt die Wahrnehmung von Fahrzeugen ohne Verwendung von künstlichen Markern. Folglich müssen die Algorithmen die natürlichen Merkmale eines Fahrzeuges erkennen. Zu den natürlichen Fahrzeugmerkmalen gehören zum einen die Front- und Rücklichter, welche mit einer Tageslichtkamera sowohl bei Tag als auch bei Nacht gut zu erkennen sind. Chen et al. [2011] entwickelte eine Fahrzeug- und Motorraderkennung speziell für Nachtfahrten. Der Algorithmus konvertiert ein Grauwertbild mit einem Schwellenwertverfahren namens *Multilevel Histogram Thresholding* in ein Binärbild. Anschließend erfolgt die Bildsegmentierung nach potentiellen Fahrzeug- sowie Motorradlichtern, durch die Suche nach hellen sowie kreisähnlichen Flächen. Die segmentierten Lichter führen zur Fahrzeugdetektion, indem Paare von Fahrzeuglichtern auf Ähnlichkeit untersucht werden. Alcantarilla et al. [2008] verwendeten ein ähnliches Vorgehen für die Fahrzeugerkennung bei

Nacht. Allerdings mit zwei Unterschieden: Zum einen befindet sich die Schwarzweiß-Kamera am Fahrzeug hinter der Windschutzscheibe und nicht geostationär in der Infrastruktur. Zum anderen werden die detektierten Fahrzeuge mit einem Kalman-Filter über die Zeit verfolgt.

An der Universität der Bundeswehr München forschten Thomanek und Dickmanns schon frühzeitig an der Wahrnehmung von Fahrzeugen [Thomanek und Dickmanns, 1992, Thomanek et al., 1994, Thomanek, 1996]. Die Priorität lag auf der visuellen Fahrzeugerkennung, zur Entwicklung einer automatisierten Längsregelung bzw. eines ACC, für Schnellstraßen und Autobahnen. Fahrzeuge besitzen typischerweise eine markante horizontale Kante unterhalb und jeweils eine vertikale Kante links und rechts der Fahrzeugkarosserie. Ein Algorithmus zur Fahrzeugerkennung basiert auf der Suche nach diesen drei Kanten und wurde auf Autobahnen in ausgiebigen ACC-Praxisversuchen erprobt. Weiterführend wurde die visuelle Fahrzeugerkennung mit Informationen eines frontal am Fahrzeug montierten Radarsensors erweitert. Hierbei konnte ein robusteres Gesamtsystem realisiert werden (vgl. Hofmann et al. [2000]).

Neben den „einfachen“ Merkmalen wie farbige Flächen, Kanten und Ecken wurden diverse Verfahren zur Erkennung „komplexer“ texturbasierter Merkmale veröffentlicht. Die bekanntesten sind Haar [Viola und Jones, 2001b], Scale-Invariant Feature Transform (SIFT) [Lowe, 1999], Speeded Up Robust Features (SURF) [Bay et al., 2006], Histograms of Oriented Gradients (HOG) [Dalal und Triggs, 2005] und Scale Invariant Descriptive Cells (SidCell) [Schweitzer und Wuensche, 2009]. Das gängige Vorgehen ist das Trainieren einer Support Vector Machine (SVM) oder eines kaskadierten Klassifikators (engl. *Cascaded Classifier*). Die Grundlage sind sogenannte „positive“ und „negative“ Trainingsdaten. Die positiven Trainingsdaten beinhalten Sensordaten, in denen das Zielobjekt manuell markiert bzw. gelabelt wurde. Die negativen Trainingsdaten bestehen aus Sensordaten, welche das Zielobjekt nicht enthalten. Mit diesen Daten und geeigneten maschinellen Trainingsalgorithmen kann anschließend eine SVM oder ein kaskadierter Klassifikator trainiert werden. Diese Klassifikatoren ermöglichen im Praxisbetrieb die Detektion eines bestimmten Objektes (z. B. weißer VW Tiguan) oder einer Objektkategorie (Pkw, Lkw) [Hota et al., 2010, Rezaei et al., 2015]. Neben einer SVM oder einem kaskadierten Klassifikator kann auch ein neuronales Netz trainiert werden. Hierbei basiert der Trainingsvorgang ebenfalls auf gelabelten positiven Trainingsdaten und einem geeigneten Trainingsalgorithmus. Besonders gängig ist hierbei die Verwendung von Kamerabildern. Ein Anwendungsbeispiel zeigten Goerick et al. [1996], welche mit einem neuronalen Netz Fahrzeuge auf Autobahnen visuell erkennen konnten. Sie modellierten ein aufgeschaltetes neuronales Netzwerk mit etwa 400 Eingangsneuronen, einer aus 25 Neuronen bestehenden verdeckten Schicht und zwei Ausgangsneuronen als Klassifikationsergebnis (Fahrzeug, kein Fahrzeug). Für die verdeckte Schicht wurde eine geeignete Neuronenanzahl experimentell ermittelt. Als Trainingsalgorithmus fand der populäre *Backpropagation*-Algorithmus [Rumelhart et al., 1986] Verwendung, welcher zu den überwachten Lernverfahren gehört. Weiterführend entwickelten sich im Laufe vieler Jahre unterschiedliche Netzarchitekturen. Hierzu zählen rekurrente (rückgekoppelte) neuronale Netze sowie sogenannte *Convolutional Neural Networks*. Des Weiteren

werden derzeit viele Publikationen im Bereich der visuellen Objekterkennung mit *Deep Learning* veröffentlicht [Szegedy et al., 2013, Lange et al., 2016, Krishnan und Larsson, 2016]. Hierbei bezieht sich der Begriff *Deep Learning* auf das Trainieren eines vielschichtigen neuronalen Netzes. Aus diesem Grund wird diese vielschichtige Netzarchitektur als *Deep Neural Network* bezeichnet. Das Trainieren eines solchen Netzes benötigt viele Trainingsdaten und ist aufgrund der hohen Netzkomplexität sehr rechenaufwendig und damit zeitintensiv. Im Vergleich zu traditionellen Lernverfahren kann mit *Deep Learning* ein deutlich verbessertes Klassifikationsergebnis erzielt werden. An dieser Stelle sei auf Krizhevsky et al. [2012] verwiesen, welche ein sehr komplexes *Deep Convolutional Neural Network* mit insgesamt 650.000 Neuronen, 60 Millionen Parametern und 630 Millionen Verbindungen trainierten und mit diesem im Jahre 2012 auf der ImageNet Large Scale Visual Recognition Challenge (ILSVRC) das beste Klassifikationsergebnis erzielten. Das Trainieren der besten Netzparametrierung dauerte grob eine Woche. Die auf der ILSVRC erzielte Fehlerrate betrug 15 Prozent. Abschließend sei auf ausgewählte Publikationen im Bereich der visuellen Fahrzeugerkennung mit *Deep Neural Networks* verwiesen: Huval et al. [2015], Szegedy et al. [2013], Lange et al. [2016] sowie Krishnan und Larsson [2016].

Des Weiteren soll ein einfacher, jedoch effizienter Algorithmus von Dominguez et al. [2011] erwähnt werden. Hierbei erfolgt die Wahrnehmung von Fahrzeugen anhand der Rohdaten eines Ein-Ebenen LiDAR-Sensors. Der Algorithmus segmentiert im ersten Schritt die LiDAR-Punktwolken in einzelne Fragmente. Dies geschieht durch die Betrachtung der Punktabstände. Ist der Punktabstand zu groß, gehören die Punkte nicht zum selben Fragment. Nach der Segmentierung erfolgt die Fahrzeugklassifizierung durch eine Filterung nach Fahrzeugspezifischen Konturen. Eine L-förmige Kontur korreliert z.B. mit einer größeren Wahrscheinlichkeit zu einem Fahrzeug.

Eine weitere Vorgehensweise ist die Wahrnehmung von statischen und dynamischen Objekten in einer Ego-bezogenen Umgebungskarte. Die Veröffentlichungen von Tanzmeister [2016], Tanzmeister und Wollherr [2017], Steyer et al. [2018] beschreiben die Möglichkeit der Erkennung und Verfolgung von statischen und dynamischen Objekten auf Basis einer gitterbasierten Umgebungskarte und einem Partikel-Filter.

Je nach Anwendungsbereich ist eine hohe Messgenauigkeit und -dimension vonnöten. Die meisten Veröffentlichungen fokussieren sich auf die Schätzung der ego-relativen 2D-Position des Verkehrsteilnehmers sowie dessen Geschwindigkeit. Ein Folgefahrzeug im Konvoi muss dagegen spurtreu folgen, andernfalls können Kollisionen mit anderen Verkehrsteilnehmern und der Umwelt auftreten. Negativbeispiele sind das Schneiden von scharfen Kurven oder aufschaukelnde Kettenreaktionen beim Mehrfachkonvoi. Im Nachfolgenden werden einige Veröffentlichungen aufgelistet, bei denen die Fahrzeugerkennung ein spurtreues Folgen zulässt.

Die Veröffentlichung von Himmelsbach et al. [2009a] erlaubt die Bestimmung der ego-relativen 3D-Position und -Orientierung eines Verkehrsteilnehmers und dessen Geschwindigkeit. Dafür werden die Entfernungsdaten eines Velodyne-Laserscanners

verwendet, der 64 Laserdioden besitzt und die 360°-Fahrzeugumgebung vermisst. Die Detektion und Klassifikation von Verkehrsteilnehmern erfolgt in zwei Schritten. Zu Beginn wird in der gesamten Punktwolke die Bodenebene ermittelt, um diese aus der Punktwolke zu entfernen (Datenreduktion). Anschließend wird die Punktwolke wie bei Dominguez et al. [2011] in einzelne Cluster segmentiert, indem Punktabstände betrachtet werden. Im letzten Schritt wird jeweils ein Histogramm für jedes Cluster berechnet, damit die Klassifizierung mit einer vorab manuell trainierten SVM erfolgen kann.

Im Vergleich zu Himmelsbach et al. [2009a] verwenden Manz et al. [2011] zusätzlich zu einem LiDAR-Sensor die Farbinformationen einer Tageslichtkamera. Das System von Manz et al. [2011] ist auf die Verfolgung eines bekannten Führungsfahrzeuges ausgelegt und nicht wie bei Himmelsbach et al. [2009a] zur Detektion und Klassifikation von unbekannten Verkehrsteilnehmern. Aus diesem Grund wird keine SVM trainiert, sondern für jedes Führungsfahrzeug ein fahrzeugspezifisches 3D-Merkmalmodell manuell erstellt. Dabei werden die signifikanten Merkmale wie die roten Fahrzeugrücklichter sowie dunkle Flächen an Reifen und innenliegenden Fensterkanten modelliert. Die Schätzung der ego-relativen 3D-Pose, der Geschwindigkeit und des Lenkwinkels des Führungsfahrzeuges erfolgt mit einem Partikel-Filter. Das Filter generiert u. a. mögliche 3D-Posen des Führungsfahrzeuges und projiziert die Merkmale des 3D-Merkmalmodells in das aktuelle Kamerabild. Anschließend erfolgt ein Merkmalsvergleich zwischen den projizierten und den im Kamerabild vorhandenen Merkmalen. Je besser die projizierten Merkmale mit der Wahrnehmung übereinstimmen, desto höher werden die Partikel gewichtet. Abschließend wird das Schätzergebnis des gesamten Partikel-Filters berechnet. Dafür liefert der arithmetische Mittelwert der am höchsten gewichteten Partikel ein Filterergebnis, welches von einem Zeitschritt zum anderen nicht zu stark schwankt. Für die Mittelwertberechnung beziehen Manz et al. [2011] 50 Prozent der Partikelmenge ein und erzielen dadurch ein sprunghaftes bzw. ruhiges Fahrverhalten. Insgesamt überzeugte das System in zahlreichen Praxisversuchen. Hierbei erfolgten die Praxistests überwiegend in unstrukturiertem Terrain, zum Teil bei Regen und teilweisen Verdeckungen des Führungsfahrzeuges. Die Testfahrten fanden ausschließlich bei Tag und im Konvoi bestehend aus zwei Teilnehmern statt, unter anderem auf der militärischen ELROB im Jahr 2010.

In Robert [2009] und Lou et al. [2005] kommen wie in Manz et al. [2011] 3D-Merkmalmodelle für die Fahrzeugerkennung zum Einsatz. Beide Veröffentlichungen erstellen ebenfalls in einem Offline-Schritt ein Merkmalsmodell pro Fahrzeugtyp. Die Anzahl an unterschiedlichen Merkmalen unterscheidet sich allerdings gravierend. Robert [2009] modelliert lediglich die Kanten der Scheiben und Lou et al. [2005] erstellen ein kantenbasiertes Drahtgittermodell für jedes Fahrzeug.

## 3 Grundlagen

### Inhalt

---

3.1	Sensorkalibrierung und -synchronisierung . . . . .	17
3.1.1	Sensorkalibrierung . . . . .	17
3.1.2	Sensorsynchronisierung . . . . .	27
3.2	Rekursive Zustandsschätzung dynamischer Systeme . . . . .	29
3.2.1	Kalman-Filter . . . . .	29
3.2.2	Partikel-Filter . . . . .	36
3.3	Kaskadierte Klassifikatoren . . . . .	40
3.3.1	Trainingsphase . . . . .	41
3.3.2	Anwendung . . . . .	46
3.4	Region Growing . . . . .	46
3.4.1	Algorithmus und Varianten . . . . .	46
3.4.2	Multi-Dynamic Region Growing . . . . .	47

---

Dieses Kapitel beschreibt die zum Verständnis dieser Arbeit notwendigen Grundlagen und besteht aus vier Unterkapiteln: Sensorkalibrierung und -synchronisierung, Rekursive Zustandsschätzung dynamischer Systeme, Kaskadierte Klassifikatoren und Region Growing.

### **3.1 Sensorkalibrierung und -synchronisierung**

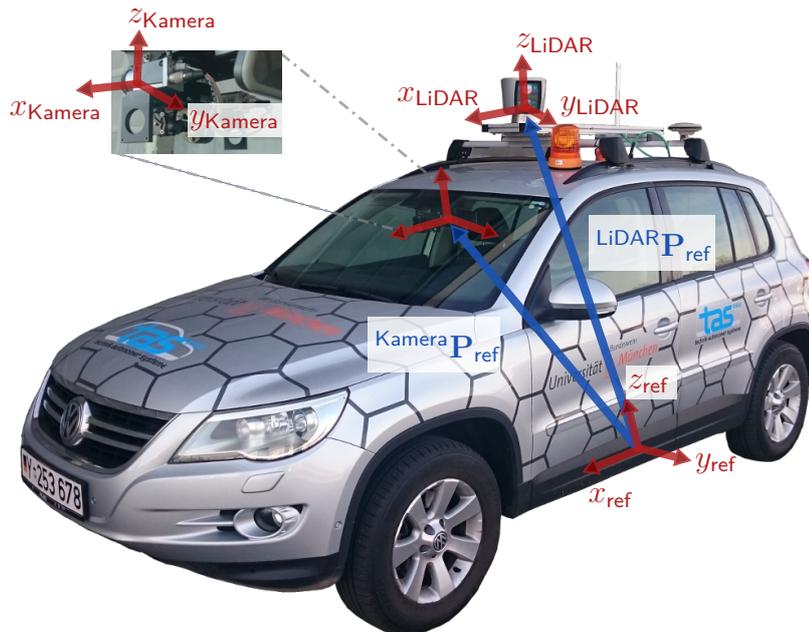
Die nachfolgenden zwei Abschnitte gehen auf die Kalibrierung und Synchronisierung von Sensoren ein.

#### **3.1.1 Sensorkalibrierung**

Die Kalibrierung von abstands- und bildgebenden Sensoren gliedert sich in die Bestimmung der extrinsischen und intrinsischen Sensorgrößen.

### 3.1.1.1 Kalibrierung extrinsischer Größen

Die extrinsischen Größen  ${}^{\text{Sensor}}\mathbf{P}_{\text{ref}} = \{x, y, z, \phi, \theta, \psi\}$  beschreiben die räumliche Position und Orientierung eines Sensors relativ zu einem Referenzpunkt (vgl. Abbildung 3.1). Hierbei sind  $\{x, y, z\}$  kartesische Positionskoordinaten und  $\{\phi, \theta, \psi\}$  Roll-, Nick- und Gierwinkel. Die initiale Bestimmung dieser sechs Größen wird typischerweise per Hand durchgeführt und anschließend mit Softwarealgorithmen, die Sensordaten analysieren, korrigiert.



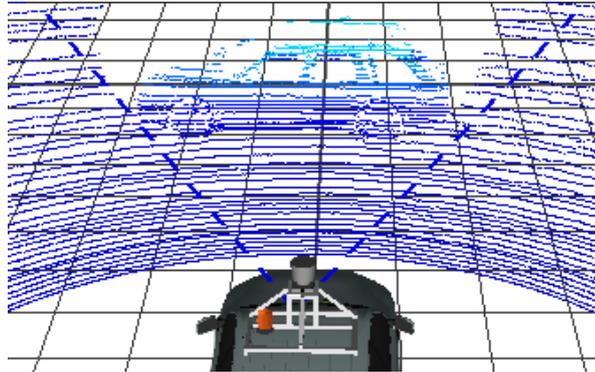
**Abbildung 3.1:**

Räumliche Beschreibung der Lage und Orientierung von Sensoren relativ zu einem Referenzpunkt.

Für die automatische Berechnung sowie Korrektur von Kalibrierwerten existieren unterschiedliche Herangehensweisen. Hierfür wird auf die Publikationen Himmelsbach et al. [2011], Schneider et al. [2013] sowie Mueller und Wuensche [2016] verwiesen. Im Nachfolgenden wird auf eine manuelle Kalibrieremethode eingegangen, welche in dieser Arbeit nach jeder erstmaligen Sensormontage zum Einsatz kam.

Der erste Schritt einer manuellen Vorkalibrierung ist das grobe Abschätzen der Position und Orientierung eines Sensors relativ zu einem fahrzeugfesten Referenzpunkt. Damit kann eine visuelle Darstellung aller Sensordaten relativ zu dem festgelegten Referenzpunkt erfolgen. Hierbei ist die Verwendung von CAD-Zeichnungen empfehlenswert. Anschließend ist die Vorkalibrierung zu verbessern, indem die Sensordaten der einzelnen Sensoren miteinander verglichen werden. Zunächst wird ein einzelner Sensor als extrinsisch kalibriert definiert, z. B. der auf dem Fahrzeugdach montierte LiDAR-Sensor aus Abbildung 3.1. Danach erfolgt eine manuelle Nachjustierung der

Kalibrierwerte der anderen Sensoren, bis alle Sensordaten zu dem ausgewählten Referenzsensor passen. Demzufolge sind die Laserdaten des LiDAR-Referenzsensors in die Kamerabilder der sich im Fahrzeug befindenden Kameras zu projizieren. Die Abbildung 3.2 veranschaulicht die visuelle Darstellung von Kamera- und LiDAR-Daten sowie die manuelle Nachkalibrierung der grob abgeschätzten extrinsischen Kalibrierung.



(a) 3D-Punktwolke



(b) Stark fehlerbehaftete Kalibrierung



(c) Korrekte Kalibrierung

**Abbildung 3.2:**

Manuelle extrinsische Kalibrierung einer Kamera in Bezug zu einem LiDAR-Sensor: (a) 3D-Punktwolke. (b) Die in das Kamerabild projizierten Laserpunkte stimmen nicht mit der visuellen Wahrnehmung überein. (c) Korrekte Kalibrierung, veranschaulicht durch die visuelle Übereinstimmung der Sensordaten.

### 3.1.1.2 Kalibrierung intrinsischer Größen

Die intrinsischen Größen eines Sensors beschreiben dessen interne Geometrie. Die Sensorhersteller veröffentlichen diese Größen pro Sensortyp bzw. Baureihe. Bei der Fertigung von Sensoren desselben Typs treten allerdings minimale Abweichungen auf, sodass die Herstellerangaben fehlerbehaftet sind und damit korrigiert werden sollten. Die nachfolgenden zwei Abschnitte erläutern die intrinsischen Größen von LiDAR- und Kamerasensoren sowie die Berechnung der intrinsischen Kameraparameter.

#### Intrinsische Größen von LiDAR-Sensoren

Die Positionierung sowie Orientierung der Laserdiode(n) sind die primären intrinsischen Größen eines LiDAR-Sensors. Die dreidimensionale Position einer Laserdiode ist relativ zu einem definierten Referenzpunkt im LiDAR-Sensor anzugeben. Die Orientierung besteht aus einem Nick- und Gierwinkel und ist folglich zweidimensional. Es existieren unterschiedliche Verfahren für die Korrektur der vom Hersteller angegebenen intrinsischen Größen. Für Informationen über die Kalibrierung dieser fünf Größen sei auf Chen et al. [2012] und Glennie und Lichti [2010] verwiesen.

#### Intrinsische Größen von Kamerasensoren

Die sogenannten intrinsischen Kameraparameter beschreiben die physikalischen Eigenschaften einer Kamera und damit die Abbildung zwischen dem 3D-Kamera- und 2D-Bildkoordinatensystem. Sie bestehen aus den folgenden intrinsischen Größen:

- Bildweite  $\mathbf{f} = [f_u, f_v]$
- Verschiebung des Bildhauptpunktes  $\mathbf{c} = [c_u, c_v]$
- Scherungskoeffizient  $s$
- Linsenverzeichnung  $\mathbf{k}_{1,\dots,5} = [k_1, k_2, k_3, k_4, k_5]$

Die Bildweite  $\mathbf{f}$  (engl. *Focal Length*) beschreibt den Abstand der Sensorfläche zur Linsenmitte.<sup>1</sup> Die Hersteller geben die Bildweite in Millimeter oder Mikrometer an, dagegen ist die Einheit „Pixel“ für mathematische Berechnungen gängig. Eine Umrechnung zwischen Millimeter- und Pixelangaben ist nach Gleichung 3.1 möglich.

$$\mathbf{f}_{[\text{px}]} = \frac{\text{Bildma\ss e}_{[\text{px}]}}{\text{Sensorma\ss e}_{[\text{mm}]}} \cdot \mathbf{f}_{[\text{mm}]} \quad (3.1)$$

Demzufolge setzt die Gleichung 3.1 bzw. die Umrechnung der Bildweite von Millimeter nach Pixel ( $\mathbf{f}_{[\text{mm}]} \rightarrow \mathbf{f}_{[\text{px}]}$ ) eine bekannte Bildweite in Millimeter sowie bekannte Bild- und Sensormaße voraus. In dem Fall, dass die horizontale und vertikale Bildweite nicht identisch sind, wird diese durch zwei Größen  $f_u$  und  $f_v$  definiert.

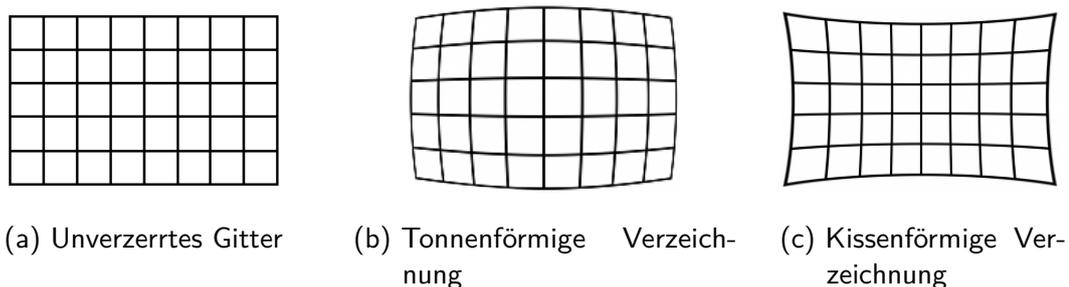
<sup>1</sup>Die Distanz zwischen Linsenmitte und Brennpunkt wird dagegen als Brennweite bezeichnet.

Der nächste intrinsische Kameraparameter ist die Verschiebung des Bildhauptpunktes (engl. *Principal Point*). Der Bildhauptpunkt befindet sich auf der Bildebene am Durchstoßpunkt der optischen Achse und liegt typischerweise in der Nähe des Bildmittelpunktes. Beispielsweise müsste sich der Bildhauptpunkt bei einer Kameraauflösung von  $1280 \times 1024$  Pixel in etwa an der Bildposition  $[u, v] = [640, 512]$  befinden.

Der Scherungskoeffizient  $s$  (engl. *Skew Coefficient*) kennzeichnet die Scherung zwischen der  $x$ - und  $y$ -Sensorachse und ist ein weiterer intrinsischer Kameraparameter. Im gängigen Fall von rechtwinkligen Bildpixeln ist der Winkel zwischen der  $x$ - und  $y$ -Sensorachse 90 Grad groß, was einen Scherungskoeffizienten von null ergibt.

Des Weiteren existieren fünf intrinsische Kameraparameter für die Beschreibung der radialsymmetrischen und tangentialen Linsenverzeichnung  $k_{1,\dots,5}$ . Die tangentielle Linsenverzeichnung besteht aus zwei Koeffizienten  $k_3$  und  $k_4$ , welche die vertikale und horizontale Verzeichnung beschreiben. Diese Linsenverzeichnung bewirkt eine Scherung des Bildes und entsteht, wenn der Bildsensor nicht exakt orthogonal zur optischen Achse ausgerichtet ist.

Die radialsymmetrische Linsenverzeichnung wird durch drei Koeffizienten  $k_1$ ,  $k_2$  und  $k_5$  beschrieben und entsteht durch die Lichtstrahlenablenkung und -brechung an der Linse des Objektivs. Die Stärke der radialsymmetrischen Verzeichnung hängt von der Qualität der Linse ab, steigt je größer der Öffnungswinkel eines Objektivs ist und nimmt ausgehend vom Bildhauptpunkt zum Bildrand zu. Für das Verzeichnungsmuster sind zwei Bezeichnungen gebräuchlich. Bei der sogenannten tonnenförmigen Linsenverzeichnung verläuft die Krümmung von der Bildmitte weg und wird zum Bildrand stärker. Dagegen existiert die kissenförmige Linsenverzeichnung, bei der die Krümmung zur Bildmitte hin verläuft. Die beiden Verzeichnungsarten sind in der Abbildung 3.3 veranschaulicht.



**Abbildung 3.3:** Zwei Verzeichnungsarten einer Kamera.

Die beschriebenen intrinsischen Kameraparameter werden folgendermaßen verwendet:

- Für die Abbildung von 3D-Kamera- auf 2D-Bildkoordinaten.
- Zur Ver- und Entzerrung von Bildpunkten bzw. Bildern.

Im Nachfolgenden wird auf diese zwei Anwendungsfälle näher eingegangen.

Die Zentralprojektion beschreibt die Abbildung eines 3D-Raumpunktes  $\mathbf{p} = [x, y, z]^T$  im Kamerakoordinatensystem auf einen 2D-Bildpunkt  $\mathbf{p}' = [u, v]^T$  und kann mit der Verwendung der Strahlensätze aus Gleichung 3.2 und 3.3 beschrieben werden (vgl. Abbildung 3.4).

$$\frac{y}{x} = \frac{u}{f_u} \quad (3.2)$$

$$\frac{z}{x} = \frac{v}{f_v} \quad (3.3)$$

Zu Beginn sind die Gleichungen 3.2 und 3.3 nach  $u$  und  $v$  umzustellen:

$$\mathbf{p}' = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u \cdot \frac{y}{x} \\ f_v \cdot \frac{z}{x} \end{bmatrix}. \quad (3.4)$$

Danach wird die Verschiebung des Bildhauptpunktes  $[c_u, c_v]^T$  durch Subtraktion des jeweiligen  $u$ -/ $v$ -Versatzes berücksichtigt:

$$\mathbf{p}' = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u \cdot \frac{y}{x} - c_u \\ f_v \cdot \frac{z}{x} - c_v \end{bmatrix}. \quad (3.5)$$

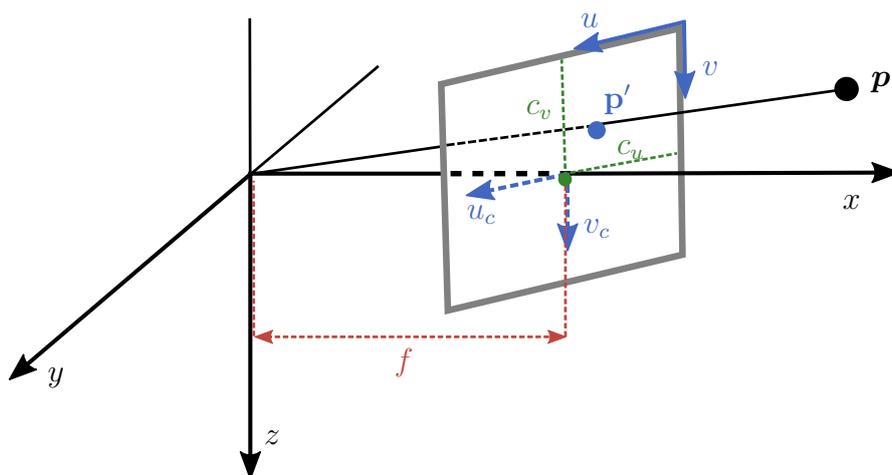


Abbildung 3.4: Zentralprojektion.

Abschließend ist der Scherungsfaktor  $s$  mit in die Abbildung einzubeziehen. Zusammenfassend ergibt sich eine  $3 \times 3$  Kamera- bzw. Kalibriermatrix  $\mathbf{K}$ :

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_u \\ \mathbf{K}_v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & f_u \cdot s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.6)$$

Mit dieser Matrix können Punktkoordinaten im Kamerakoordinatensystem auf die Bildebene abgebildet werden:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \cdot \begin{bmatrix} y \\ z \\ x \end{bmatrix}. \quad (3.7)$$

Die Ver- und Entzerrung von Bildpunkten ist ein wesentlicher Bestandteil in der Bildverarbeitung und wird aus diesem Grund im Nachfolgenden aufgeführt. Grundsätzlich ist die Ver- sowie Entzerrung relativ zum Bildhauptpunkt durchzuführen:

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} u - c_u \\ v - c_v \end{bmatrix}. \quad (3.8)$$

Die Gleichung 3.9 beschreibt die Verzerrung von Bildpunkten. In dieser entspricht  $r^2 = u_c^2 + v_c^2$  der Koordinatenentfernung zum Bildhauptpunkt und  $[\Delta u_c, \Delta v_c]^T$  der tangentialen Linsenverzeichnung, die nach Gleichung 3.10 berechnet werden kann (vgl. Brown [1966, 1971]).

$$\begin{bmatrix} u_{c,\text{verzerrt}} \\ v_{c,\text{verzerrt}} \end{bmatrix} = [1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_5 \cdot r^6] \cdot \begin{bmatrix} u_c \\ v_c \end{bmatrix} + \begin{bmatrix} \Delta u_c \\ \Delta v_c \end{bmatrix} \quad (3.9)$$

$$\begin{bmatrix} \Delta u_c \\ \Delta v_c \end{bmatrix} = \begin{bmatrix} 2 \cdot k_3 \cdot u_c \cdot v_c + k_4 \cdot [r^2 + 2 \cdot u_c^2] \\ 2 \cdot k_4 \cdot u_c \cdot v_c + k_3 \cdot [r^2 + 2 \cdot v_c^2] \end{bmatrix} \quad (3.10)$$

Anschließend wird die Entzerrung eines Bildpunktes beschrieben. Zu Beginn muss ein verzerrter Bildpunkt  $[u_{c,\text{verzerrt}}, v_{c,\text{verzerrt}}]^T$  in das 3D-Kamerakoordinatensystem abgebildet werden. Hierfür wird der Bildpunkt durch die Bildweite  $\mathbf{f} = [f_u, f_v]^T$  dividiert und die Scherung  $s$  der Sensorachse berücksichtigt, welche sich ausschließlich

auf die  $u$ -Koordinate auswirkt. Es ergeben sich die Gleichungen 3.11 und 3.12 sowie zusammenfassend 3.13.

$$z = \frac{v_{c,\text{verzerrt}}}{f_v} \quad (3.11)$$

$$y = \frac{u_{c,\text{verzerrt}}}{f_u} + s \cdot z \quad (3.12)$$

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} \frac{u_{c,\text{verzerrt}}}{f_u} + s \cdot \frac{v_{c,\text{verzerrt}}}{f_v} \\ \frac{v_{c,\text{verzerrt}}}{f_v} \end{bmatrix} \quad (3.13)$$

Der nächste Schritt ist die Kompensation der nichtlinearen Linsenverzeichnung. Eine Möglichkeit ist eine iterative Vorgehensweise, in der  $[y, z]^T$  als Startwert zu nehmen ist [Bouguet, 2015].

Die Iteration wird typischerweise nach einer festgelegten Anzahl an Durchläufen (z. B. 20) abgebrochen und besteht aus den folgenden fünf Einzelschritten:

1. Anlegen einer Hilfsvariable

$$r^2 = y^2 + z^2$$

2. Berechnung der radialen Linsenverzeichnung

$$\mathbf{k}_{\text{radial}} = [1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_5 \cdot r^6]$$

3. Berechnung der tangentialen Linsenverzeichnung

$$\begin{bmatrix} \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 2 \cdot k_3 \cdot y \cdot z + k_4 \cdot [r^2 + 2 \cdot y^2] \\ 2 \cdot k_4 \cdot y \cdot z + k_3 \cdot [r^2 + 2 \cdot z^2] \end{bmatrix}$$

4. Subtraktion der tangentialen Linsenverzeichnung

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} y \\ z \end{bmatrix} - \begin{bmatrix} \Delta y \\ \Delta z \end{bmatrix}$$

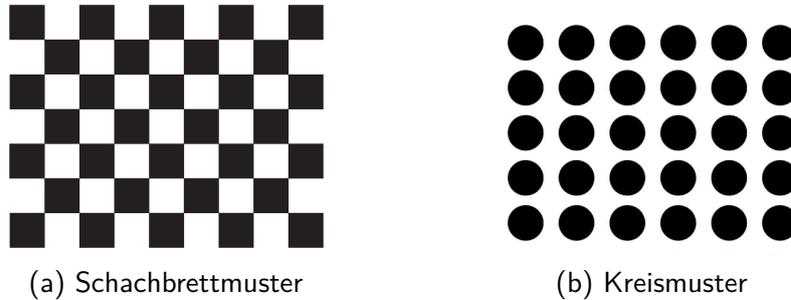
5. Division durch die radiale Linsenverzeichnung

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} \frac{y}{k_{\text{radial}}} \\ \frac{z}{k_{\text{radial}}} \end{bmatrix}$$

Dieses Verfahren ergibt die entzerrten Koordinaten  $[y, z, 1]^T$  im Kamerakoordinatensystem. Im letzten Schritt müssen diese nun zurück auf die Bildebene projiziert werden. Dafür wird die Kamerakalibriermatrix  $\mathbf{K}$  verwendet (vgl. Gleichung 3.6).

### Bestimmung der intrinsischen Kameraparameter

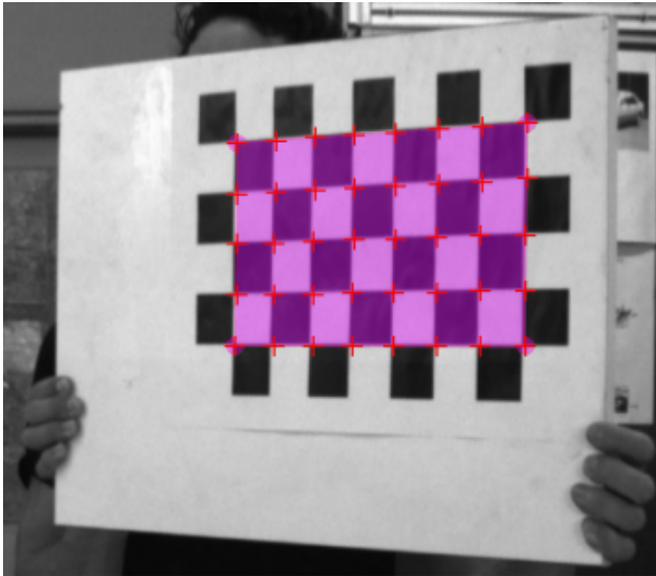
In diesem Abschnitt wird ein Verfahren zur automatisierten Bestimmung der intrinsischen Kameraparameter beschrieben. Die Methode benötigt von einem bekannten Kalibriermuster perspektivisch unterschiedliche Kameraaufnahmen. Gängig sind ebene Schachbrett- oder Kreismuster, wie in der Abbildung 3.5 dargestellt.



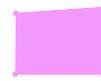
**Abbildung 3.5:**

Ebene Kalibriermuster zur Ermittlung der intrinsischen Kameraparameter.

Die Bestimmung der intrinsischen Kameraparameter erfolgt mit dem Wissen über die Geometrie des Kalibrierusters. Für ein Schachbrettmuster muss die Anzahl horizontaler und vertikaler Vierecke sowie deren Seitenlängen bekannt sein. Bei einem Kreismuster wird die Anzahl der horizontalen und vertikalen Kreise sowie die Abstandsmaße zwischen den Kreismittelpunkten benötigt. Die Erkennung der Muster erfolgt durch Bildverarbeitungsalgorithmen. Beispielsweise können Kanten mit dem sogenannten *Canny Edge Detector* [Canny, 1986] erkannt werden. Anschließend führen, wie in Abbildung 3.6 gezeigt, Schnittpunkte von Kanten zu den gesuchten Eckpunkten. Abschließend sind die intrinsischen Kameraparameter durch Vergleich zwischen dem visuell wahrgenommenen und dem bekannten Muster (Vorwissen) zu berechnen. Für eine detaillierte Beschreibung wird auf Bouguet [2015] und Bouguet [1999] verwiesen.

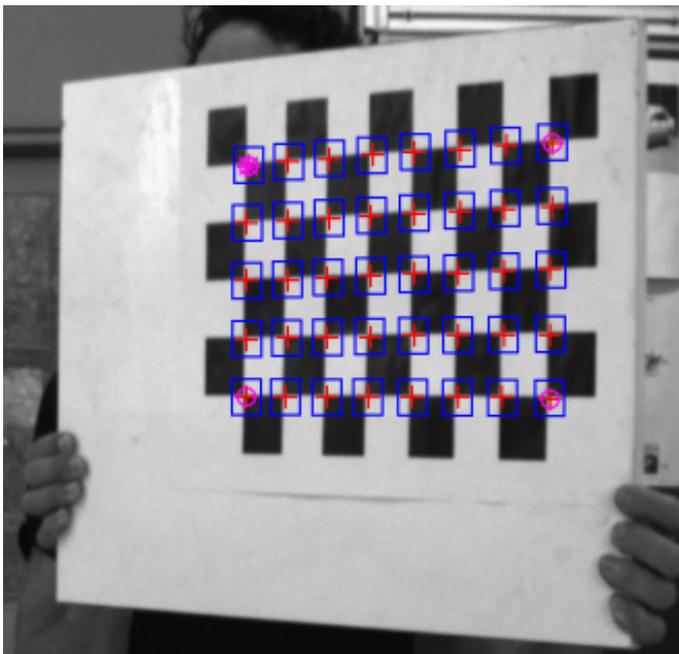


+ Eckpunktdetektion



Innerer Markerbereich

(a) Initiale Eckpunktdetektion



□ Suchfenster

+ Eckpunktdetektion

○ Äußere Ecken

\* Referenzpunkt

(b) Subpixelgenaue Suche

**Abbildung 3.6:**

Visuelle Detektion eines bekannten Schachbrettmusters: (a) Initiale Suche nach Eckpunkten. (b) Anschließende subpixelgenaue Suche innerhalb der Suchfenster.

### 3.1.2 Sensorsynchronisierung

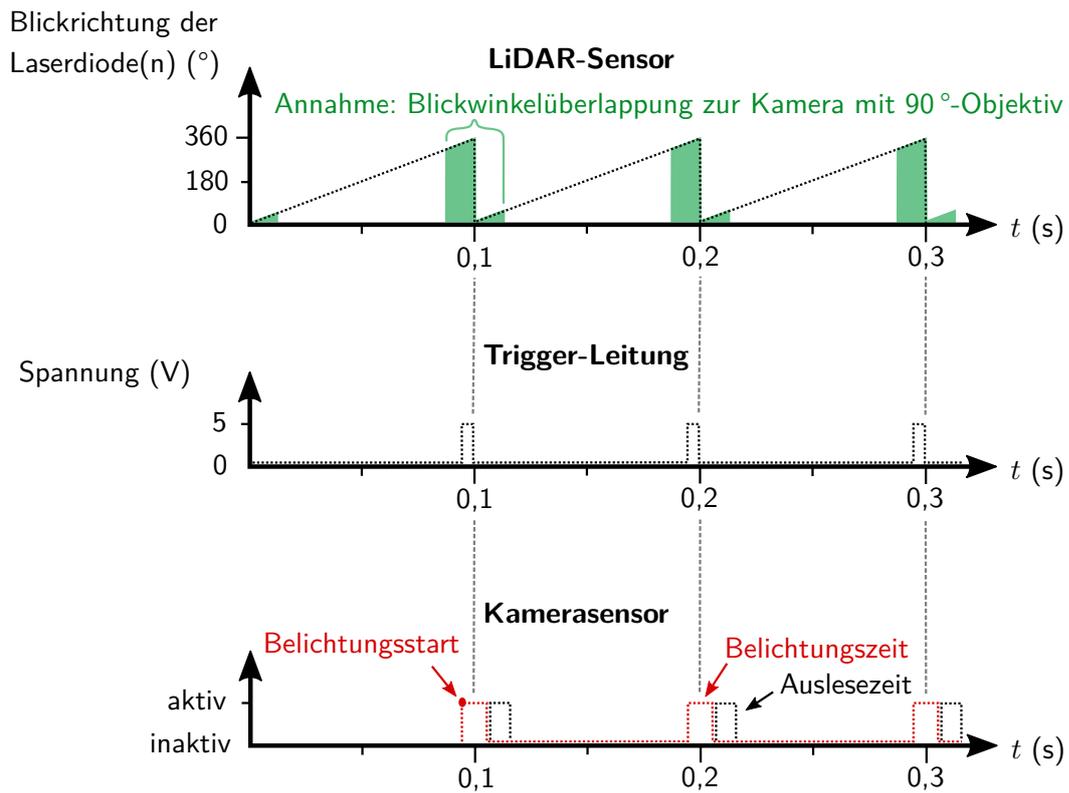
Bei der Verwendung von mehreren Sensoren ist die Synchronisierung von Sensordaten eine wichtige Aufgabe. Dieses Kapitel beschreibt den Unterschied zwischen einer hardware- und softwarebasierten Sensorsynchronisierung, den optimalen Triggerzeitpunkt und den kombinierten Einsatz von LiDAR- und Kamerasensoren.

Eine Sensorsynchronisierung kann hardwarebasiert oder softwarebasiert erfolgen. Bei der hardwarebasierten Triggerung von Sensoren wird über eine zusätzliche elektrische Leitung ein Spannungswert bzw. TTL-Spannungspegel angelegt. Hierbei ist ein gängiger Spannungswert fünf Volt. Getriggert wird entweder auf die steigende ( $0\text{V} \rightarrow 5\text{V}$ ) oder fallende ( $5\text{V} \rightarrow 0\text{V}$ ) Flanke des Spannungswertes.

Als Alternative zur hardwarebasierten Triggerung kann auch eine Softwarelösung verwendet werden. Dort erfolgt die Triggermitteilung an einen Sensor über ein Bussystem. Zu den gängigen Bussystemen gehören Ethernet, FireWire, USB und CameraLink. Das Versenden einer Triggernachricht über diese Bussysteme dauert zum einen aufgrund der geringeren Übertragungsgeschwindigkeit deutlich länger als bei der genannten hardwarebasierten Vorgehensweise. Zum anderen ist die Paketlaufzeit nicht deterministisch, wenn weitere Pakete von anderen Applikationen über denselben Bus gesendet werden. Aus diesen Gründen ist eine hardwarebasierte Triggerung von Sensoren vorzuziehen.

Bei der Kombination eines rotierenden  $360^\circ$ -LiDAR-Sensors mit einem oder mehreren Kamerasensoren ist der Triggerzeitpunkt in Abhängigkeit zum Laserdiodenblickwinkel und zur Kamerabelichtungszeit zu berücksichtigen [Schneider et al., 2010c]. Zum Beispiel befindet sich ein rotierender  $360^\circ$ -LiDAR-Sensor auf dem Dach eines Fahrzeuges sowie eine in Fahrtrichtung ausgerichtete Kamera mit einem  $90^\circ$ -Objektiv. In diesem Fall sollte die Kamera in dem Moment die Belichtung durchführen, wenn der LiDAR-Sensor mit seinen Laserdioden gerade den Kamerasichtbereich abtastet. Dies ermöglicht aufgrund der zeitlich synchronen Sensorrohdaten eine bessere Datenfusion bzw. Sensordatenverarbeitung.

Des Weiteren kann beim Triggerzeitpunkt die Belichtungszeit einer Kamera zusätzlich berücksichtigt werden. Hierbei empfiehlt sich die Triggerung einer auf dem Fahrzeugdach in Fahrtrichtung montierten Kamera, kurz bevor die rotierenden Laserdioden ebenfalls in Fahrtrichtung positioniert sind. Genau genommen sollte die Triggerung um die Hälfte der durchschnittlichen Belichtungszeit früher erfolgen. Die Abbildung 3.7 veranschaulicht diese sensorspezifische Triggerungsweise.



**Abbildung 3.7:** Blickrichtungsabhängige Synchronisierung von Sensordaten zwischen einem 360°-LiDAR- und einem Kamerasensor.

## 3.2 Rekursive Zustandsschätzung dynamischer Systeme

Dieses Kapitel behandelt die rekursive Zustandsschätzung von dynamischen Systemen mit zeitlich veränderlichen Zustandsgrößen.

Ein Zustand eines dynamischen Systems beinhaltet im sogenannten Zustandsvektor  $x_k$  die zu schätzenden Größen für einen bestimmten Zeitpunkt  $k$ . Die gesuchten Größen sind in der Regel nicht direkt messbar, können jedoch über die Zeit geschätzt werden.

Eine Zustandsschätzung ist aufgrund von Rauschfaktoren und Berechnungsvereinfachungen fehlerbehaftet. Aus diesem Grund wird von einem Schätzwert und nicht von der Bestimmung des wahren Zustandswertes gesprochen. Dabei müssen die Zustandsgrößen über die Zeit beobachtbar sein, andernfalls ist eine Schätzung nicht möglich. Hierbei wird im Allgemeinen angenommen, dass die Beobachtungen mittelwertfreiem und normalverteiltem Messrauschen unterliegen. Die Unsicherheit des dynamischen Systems wird als System- oder Prozessrauschen bezeichnet. Das Rauschen wird durch Wahrscheinlichkeitsdichtefunktionen wie die Gauß- oder Cauchy-Verteilung modelliert.

Ein dynamisches System besitzt zeitlich veränderliche Zustände. Durch Zustandsübergangsfunktionen (Transitionen) wird der Übergang zwischen zwei Zuständen beschrieben. Aus der Beobachtung des Systems ergeben sich Messwerte, welche die rekursive Korrektur des Schätzwertes bzw. Zustandswechsel ermöglichen. Ein zeitkontinuierlicher Vergleich zwischen Vorhersage und Messung bildet das Grundgerüst einer rekursiven Zustandsschätzung. Dabei besteht die Rekursion aus einem Prädiktions- und einem Innovations-schritt (vgl. Simon [2006]).

Der sogenannte 4D-Ansatz beschreibt eine Art der rekursiven Zustandsschätzung eines dynamischen Systems. Der Fokus liegt in der Modellierung und Schätzung der Form und Dynamik eines bewegten Objektes (z. B. ein Verkehrsteilnehmer) innerhalb der dreidimensionalen Welt und nicht in z. B. einem 2D-Kamerabild. Dieser Ansatz wurde von Dickmanns [2007] durch die zeitliche Beobachtung (4. Dimension) in der dreidimensionalen Welt als „4D-Ansatz“ bezeichnet. Im Vergleich zur Schätzung im Kamerabild ist diese Herangehensweise deutlich recheneffizienter, da kein hochgradig nichtlinearer Übergang vom 2D-Kamerabild in die 3D-Weltvorstellung erfolgt.

In dieser Arbeit kommen zwei rekursive Bayessche Schätzverfahren zum Einsatz. Im Nachfolgenden werden diese mit den zugehörigen Filtergleichungen beschrieben.

### 3.2.1 Kalman-Filter

Das Kalman-Filter [Kalman, 1960] ist ein rückgekoppeltes Regelungssystem und wird zur Zustandsschätzung eines zeitdiskreten linearen dynamischen Systems verwendet.

Die rekursive Schätzung erfolgt durch einen kontinuierlichen Vergleich zwischen prädierten und tatsächlichen Messwerten und der Minimierung der mittleren Abweichung mit einem quadratischen Gütekriterium.

Der Kalman-Filter ermittelt Mittelwert und Kovarianz der Zustandsgrößen und setzt bekanntes, normalverteiltes, weißes System- und Messrauschen voraus. Diese Rauschfaktoren sowie der Zustandsvektor sind initial zu definieren. Im Anschluss erfolgt der zweiphasige Filterzyklus, welcher nachfolgend näher betrachtet wird.

### 3.2.1.1 Prädiktion

Der erste Schritt ist die Vorhersage für die Zustandswerte  $\mathbf{x}$  zu einem diskreten Zeitpunkt  $k + 1$  und wird durch die Rekursionsgleichung 3.14 beschrieben.

$$\mathbf{x}^*_{k+1} = \Phi_k \cdot \hat{\mathbf{x}}_k + \mathbf{B}_k \cdot \mathbf{u}_k \quad (3.14)$$

Diese Vorhersage wird als A-priori-Schätzung bezeichnet und verwendet den deterministischen Teil eines vorgegebenen dynamischen Modells. Die Prädiktion erfolgt ausgehend vom bisherigen Schätzergebnis  $\hat{\mathbf{x}}_k$  (Gedächtnis; Markow-Kette erster Ordnung). Die Modellierung des dynamischen Modells findet im sogenannten Prozessmodell statt. Hierbei kommen Bewegungsmodelle zum Einsatz und die Eigenbewegung kann berücksichtigt werden.

Die Prädiktion setzt sich aus zwei Termen zusammen. Zum einen beschreibt die Transitionsmatrix  $\Phi_k$  die Überführung ( $\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}$ ) zwischen zwei zeitlich aufeinanderfolgenden Zuständen. Zum anderen müssen die Steuergrößen  $\mathbf{B}_k \cdot \mathbf{u}_k$  berücksichtigt werden. Die Transitionsmatrix  $\Phi_k$  und Eingangsmatrix  $\mathbf{B}_k$  werden als bekannt vorausgesetzt.

Eine Prädiktion ist im Normalfall fehlerbehaftet. Die Unsicherheit einer Prädiktion beschreibt die Kovarianzmatrix  $\mathbf{P}^*_{k+1}$ , deren Berechnung in der nachfolgenden Gleichung 3.15 aufgeführt ist.

$$\mathbf{P}^*_{k+1} = \Phi_k \cdot \mathbf{P}_k \cdot \Phi_k^T + \mathbf{Q}_k \quad (3.15)$$

Hierbei entspricht  $\mathbf{Q}_k$  der Kovarianzmatrix eines normalverteilten weißen Rauschprozesses der Systemstörung. Dieser ist mittelwertfrei und unkorreliert und spiegelt Störeinflüsse wie z. B. Modellierungsfehler wider.

Die Prädiktion der Messgrößen erfolgt nach Gleichung 3.16. Die Messmatrix  $\mathbf{C}_{k+1}$  beschreibt die Abbildung zwischen den Schätz- und Messgrößen.

$$\mathbf{y}^*_{k+1} = \mathbf{C}_{k+1} \cdot \mathbf{x}^*_{k+1} \quad (3.16)$$

Abschließend wird am Ende des Prädiktionsschrittes oder alternativ am Anfang des Innovationsschrittes die Kalman-Filtermatrix berechnet:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^* \cdot \mathbf{C}_{k+1}^T \cdot \left( \mathbf{C}_{k+1} \cdot \mathbf{P}_{k+1}^* \cdot \mathbf{C}_{k+1}^T + \mathbf{R}_{k+1} \right)^{-1} \quad (3.17)$$

Diese Filtermatrix dient als Verstärkungsfaktor innerhalb der Innovation und ist ein Maß, wie stark die nächste Messung gewichtet wird. Die Messunsicherheit wird durch einen normalverteilten weißen Rauschprozess beschrieben. Die Kovarianzmatrix dieses Prozesses wird mit  $\mathbf{R}_{k+1}$  bezeichnet.  $\mathbf{C}_{k+1}^T$  entspricht der transponierten Messmatrix.

### 3.2.1.2 Innovation

Jede neue Messung  $\mathbf{y}$  ist eine (verrauschte) Beobachtung und kann in einer Filterinnovation genutzt werden. In dieser wird der prädizierte Zustand  $\mathbf{x}_{k+1}^*$  anhand der aktuellen Messung  $\mathbf{y}_{k+1}$ , deren Prädiktion  $\mathbf{y}_{k+1}^*$  und der Kalman-Filtermatrix  $\mathbf{K}$  korrigiert. Je niedriger die Differenz zwischen der prädizierten und der gemessenen Messung im Mittel ausfällt, desto genauer ist die Vorhersage und damit die Zustandsschätzung.

In der nachfolgenden Gleichung 3.18 ist die Mittelwertschätzung  $\hat{\mathbf{x}}_{k+1}$  der Zustandsgrößen niedergeschrieben.

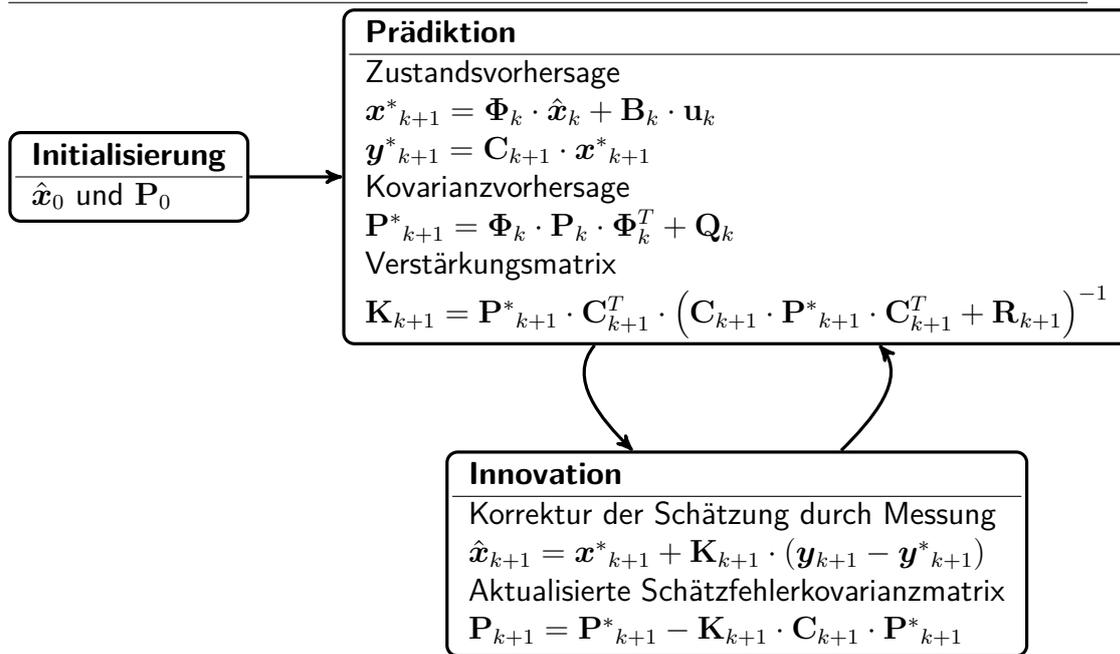
$$\hat{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1}^* + \mathbf{K}_{k+1} \cdot \overbrace{(\mathbf{y}_{k+1} - \mathbf{y}_{k+1}^*)}^{\text{Residuum}} \quad (3.18)$$

Die Kalman-Filtermatrix  $\mathbf{K}_{k+1}$  wird auch als Kalmanverstärkung bezeichnet. Im Prädiktionsschritt wird  $\mathbf{K}_{k+1}$  durch stochastische Betrachtungen berechnet. Gleichmaßen beeinflusst diese Matrix auch die A-posteriori-Schätzfehlerkovarianz  $\mathbf{P}_{k+1}$  bzw. verringert diese:

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^* - \mathbf{K}_{k+1} \cdot \mathbf{C}_{k+1} \cdot \mathbf{P}_{k+1}^* \quad (3.19)$$

Nach dem Innovationsschritt liegt ein korrigierter Schätzwert  $\hat{\mathbf{x}}_{k+1}$  vor, welcher als sogenannter A-posteriori-Schätzwert bezeichnet wird. Abschließend startet der Algorithmus mit dem Prädiktionsschritt für den nächsten Zeitpunkt  $k + 2$ .

Zusammenfassend sind im Algorithmus (1) die grundlegenden Filtergleichungen des diskreten linearen Kalman-Filters aufgelistet. Die Notation der Prädiktions- und Innovationsgleichungen ist an Wuensche [1987] angelehnt. Für eine ausführliche Herleitung der Filtergleichungen sei auf Kalman [1960] verwiesen.

**Algorithmus 1** Rekursive Zustandsschätzung mit einem diskreten Kalman-Filter.**3.2.1.3 Sequentielle Innovation**

Die Innovationsgleichungen des linearen diskreten Kalman-Filters (vgl. Abschnitt 3.2.1.2) können für eine sequentielle Einfilterung einer variablen Anzahl von Messwerten  $i = 1, \dots, j$  - die in einem diskreten Zeitpunkt  $k$  eintreffen - abgeändert werden. Die hier vorgestellte recheneffiziente Abwandlung gliedert sich in vier Rekursionsschritte, welche eine rechenaufwendige Inversion der  $j \times j$  Matrix  $(\mathbf{C}_{k+1} \cdot \mathbf{P}^*_{k+1} \cdot \mathbf{C}_{k+1}^T + \mathbf{R}_{k+1})^{-1}$  aus Gleichung 3.17 auf eine Rekursion mit skalarer Division zurückführt. Als Voraussetzung dürfen die Messwerte nicht korreliert und müssen damit unabhängig sein. Dies ist der Fall, wenn die Kovarianzmatrix der Messunsicherheit  $\mathbf{R}_k$  einer Diagonalmatrix entspricht.

## 1. Prädiktion

Die Vorhersage der Zustandsgrößen durch Mittelwert  $\mathbf{x}^*_k$  und Kovarianz  $\mathbf{P}^*_k$  erfolgt gleichermaßen wie im vorherigem Abschnitt 3.2.1.1:

$$\mathbf{x}^*_{0,k} = \mathbf{x}^*_k = \Phi_{k-1} \cdot \hat{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1} \cdot \mathbf{u}_{k-1} \quad (3.20)$$

$$\mathbf{P}_0 = \mathbf{P}^*_k = \Phi_{k-1} \cdot \mathbf{P}_{k-1} \cdot \Phi_{k-1}^T + \mathbf{Q}_{k-1} \quad (3.21)$$

## 2. Initialisierung der Innovationsrekursion

Die Innovation erfolgt aufgrund der sequentiellen Einfilterung in einer zusätzlichen Rekursion. Vorab ist der korrigierte Zustand  $\delta \hat{\mathbf{x}}_0$  mit null zu initialisieren und die  $i$  prädizierten Messgrößen  $\mathbf{y}^*_{i,0,k}$  (Index 0 entspricht null einbezogene Innovationen) des Messvektors  $\mathbf{y}_k$  für den Zeitpunkt  $k$  zu berechnen.

## 3. Durchführung von Teilinnovationen (Rekursion)

Nach jeder Rekursion erfolgt die erneute Prädiktion der Zustandsgrößen

$$\mathbf{x}^*_{i,k} = \hat{\mathbf{x}}_{i-1,k} = \mathbf{x}^*_{0,k} + \delta \hat{\mathbf{x}}_{i-1} \quad (3.22)$$

und der Messgrößen

$$\mathbf{y}^*_{i,k} = \mathbf{y}^*_{i,0,k} + \mathbf{c}_i \cdot \delta \hat{\mathbf{x}}_{i-1}. \quad (3.23)$$

Dabei erfolgt die Verarbeitung der Messmatrix  $\mathbf{C}_k$  zeilenweise. In der Gleichung (3.23) entspricht der Vektor  $\mathbf{c}_i$  der zu  $\mathbf{y}_i$  zugehörigen  $i$ -ten Zeile der Messmatrix  $\mathbf{C}_k$ . Die Filterverstärkung berechnet sich damit durch

$$\mathbf{K}_i = \mathbf{P}_{i-1} \cdot \mathbf{c}_i^T \cdot \left( \mathbf{c}_i \cdot \mathbf{P}_{i-1} \cdot \mathbf{c}_i^T + r_i \right)^{-1}, \quad (3.24)$$

in der  $r_i$  ein Skalar ist, der das Messrauschen bezüglich der  $i$ -ten Messkomponente beschreibt. Die Berechnung der Gleichung 3.24 erfordert lediglich eine skalare Division. Anschließend kann durch Verwendung des  $i$ -ten Messwertes  $\mathbf{y}_i$  eine Teilinnovation durchgeführt werden. Die nachfolgende Gleichung 3.25 beschreibt die Aktualisierung des aktuellen Schätzwertes für den Mittelwert

$$\delta \hat{\mathbf{x}}_i = \delta \hat{\mathbf{x}}_{i-1} + \mathbf{K}_i \cdot (\mathbf{y}_i - \mathbf{y}^*_i), \quad (3.25)$$

welcher die übergeordnete Zustandsschätzung

$$\hat{\mathbf{x}}_{i,k} = \hat{\mathbf{x}}_{0,k} + \delta \hat{\mathbf{x}}_i \quad (3.26)$$

korrigiert. Die Schätzfehlerkovarianz muss nach jedem Eintreffen einer Messung berechnet werden und ergibt sich durch die folgende Gleichung:

$$\mathbf{P}_i = \mathbf{P}_{i-1} - \mathbf{K}_i \cdot \mathbf{c}_i \cdot \mathbf{P}_{i-1}. \quad (3.27)$$

## 4. Zusammenführung der Teilinnovationen

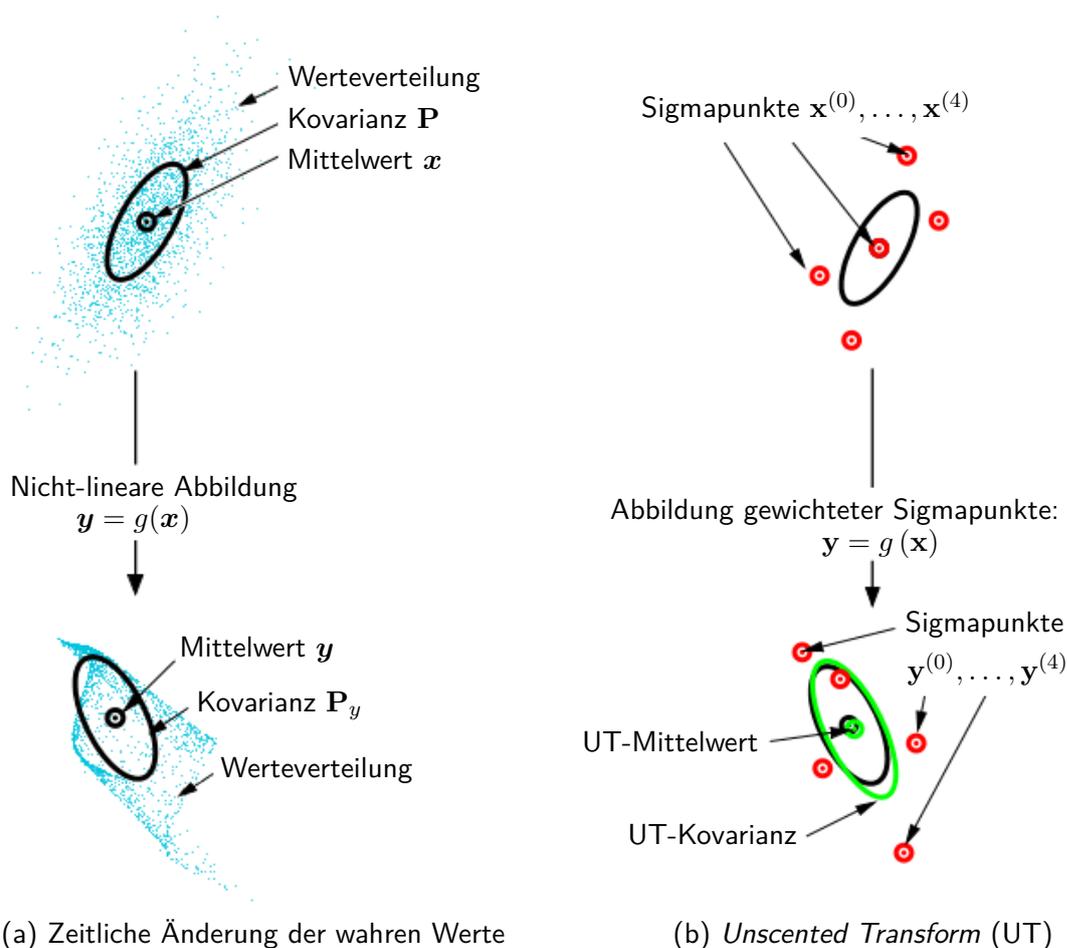
Im Anschluss an die innere Rekursion „Durchführung von Teilinnovationen“ ist eine abschließende Korrektur der Zustandsschätzung durchzuführen:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{j,k} = \mathbf{x}^*_{0,k} + \delta \hat{\mathbf{x}}_{j,k} \quad (3.28)$$

$$\mathbf{P}_k = \mathbf{P}_{j,k}. \quad (3.29)$$

### 3.2.1.4 Scaled Unscented Transform

Dieser Abschnitt beschreibt eine Abwandlungsform des vorgestellten linearen Kalman-Filters auf ein zeitdiskretes nichtlineares dynamisches System. Julier et al. [1995] entwickelten die sogenannte *Unscented Transform*, die keine nichtlineare Abbildungsgleichung annähert, sondern eine Gauß-Verteilung durch sogenannte „Sigmapunkte“ approximiert. Eine beispielhafte Visualisierung von Sigmapunkten zur Schätzung der Gaußverteilung und damit eines Mittelwertes mit Varianz visualisiert die Abbildung 3.8.



**Abbildung 3.8:**

*Unscented Transform*: (a) Tatsächliche Verteilung mit Mittelwert und Varianz zum Zeitpunkt  $k$  und  $k + 1$ . (b) Approximation einer (Gauß-)Verteilung durch Sigmapunkte sowie deren zeitlich-prädizierte Abbildungen. Bildarstellung nach Wan und Merwe [2000].

Die Verwendung der *Unscented Transform* im Kalman-Filter wird als sogenannter Unscented Kalman-Filter (UKF) bezeichnet. Weiterführend veröffentlichte Julier [2002] mit dem Titel „*The Scaled Unscented Transformation*“ eine generalisierte Abbildungsform, welche eine beliebige Streuung der Sigmapunkte zulässt. Für die

Skalierung der Sigmakpunkte zueinander existiert ein einstellbarer Parameter  $\kappa$ , welcher in der *Scaled Unscented Transform* um zwei weitere Skalierungskoeffizienten  $\alpha$  und  $\beta$  erweitert wird. Eine hohe Skalierung der Sigmakpunkte führt zu einem großen Abdeckungsbereich der Gauß-Verteilung. Je höher allerdings die Skalierung gewählt wird, desto kleiner ist der Diskretisierungsbereich und damit die Abbildungsgenauigkeit. Diesbezüglich empfiehlt Julier [2002] zur optimalen Approximation einer Gauß-Verteilung die Parametrierung  $\kappa \geq 0$ ,  $\alpha \in (0,1]$  und  $\beta = 2$  zu wählen.

Die Anzahl der benötigten Sigmakpunkte ist abhängig von der Dimension des Zustandsraumes ( $n$ ) und wird durch  $(2 \cdot n + 1)$  definiert. Jeder Sigmakpunkt  $\mathbf{x}^{(i)}$  hat ein individuelles Gewicht  $\mathbf{w}$ :

$$\lambda = \alpha^2 \cdot (n + \kappa) - n \quad (3.30)$$

$$\mathbf{w}_0 = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) \quad (3.31)$$

$$\mathbf{w}_i = \frac{1}{2 \cdot (n + \lambda)} \quad \text{für } i = 1, \dots, (2 \cdot n). \quad (3.32)$$

Die Sigmakpunkte werden durch Wurzelziehung der gewichteten prädierten Kovarianzmatrix  $\mathbf{P}^*$  bestimmt. Die Berechnung ist in den Gleichungen 3.33 bis 3.35 niedergeschrieben.  $\mathbf{x}^*$  entspricht dem vorhergesagten Mittelwert. In den Gleichungen 3.34 und 3.35 steht der Term  $\left(\sqrt{(n + \kappa) \cdot \mathbf{P}^*}\right)_i$  für die  $i$ -te Spalte des Ergebnisses der Wurzelziehung von  $(n + \kappa) \cdot \mathbf{P}^*$ . Für die Berechnung der Wurzel muss die Kovarianzmatrix  $\mathbf{P}^*$  - z. B. mittels Cholesky-Zerlegung - in eine untere Dreiecksmatrix zerlegt werden.

$$\mathbf{x}^{(0)} = \mathbf{x}^* \quad (3.33)$$

$$\mathbf{x}^{(i)} = \mathbf{x}^* + \left(\sqrt{(n + \kappa) \cdot \mathbf{P}^*}\right)_i \quad \text{für } i = 1, \dots, n \quad (3.34)$$

$$\mathbf{x}^{(i)} = \mathbf{x}^* - \left(\sqrt{(n + \kappa) \cdot \mathbf{P}^*}\right)_{i-n} \quad \text{für } i = (n + 1), \dots, (2 \cdot n) \quad (3.35)$$

Die Vorhersage des Mittelwertes, der Varianz und der Messwerte ist in den Gleichungen 3.36 bis 3.38 beschrieben.

$$\mathbf{x}_{k+1}^* = \Phi_k \cdot \hat{\mathbf{x}}_k + \mathbf{B}_k \cdot \mathbf{u}_k \quad (3.36)$$

$$\mathbf{P}_{k+1}^* = \Phi_k \cdot \mathbf{P}_k \cdot \Phi_k^T + \mathbf{D}_k \cdot \mathbf{Q}_k \cdot \mathbf{D}_k^T \quad (3.37)$$

$$\mathbf{y}_{k+1}^* = \sum_i \{\mathbf{w}_i \cdot \mathbf{y}_i\} = \sum_i \{\mathbf{w}_i \cdot g(\mathbf{x}^{(i)})\} \quad (3.38)$$

Mit der Kovarianzmatrix der Innovation

$$\mathbf{P}_{yy} = \sum_{i=0}^{2 \cdot n} \mathbf{w}_i \cdot \{[\mathbf{y}_i - \mathbf{y}_{k+1}^*] \cdot [\mathbf{y}_i - \mathbf{y}_{k+1}^*]^T\} + \mathbf{R}_{k+1} \quad (3.39)$$

und der Kreuzkovarianzmatrix

$$\mathbf{P}_{xy} = \sum_{i=0}^{2 \cdot n} \mathbf{w}_i \cdot \left( [\mathbf{x}^{(i)} - \mathbf{x}^*] \cdot [\mathbf{y}_i - \mathbf{y}^*]^T \right) \quad (3.40)$$

zwischen Messung und Schätzung lässt sich die Kalman-Verstärkungsmatrix

$$\mathbf{K}_{k+1} = \mathbf{P}_{xy} \cdot \mathbf{P}_{yy}^{-1} \quad (3.41)$$

berechnen. Mit der Kalman-Verstärkungsmatrix kann anschließend der Messwert eingefiltert werden:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1}^* + \mathbf{K}_{k+1} \cdot (\mathbf{y}_{k+1} - \mathbf{y}_{k+1}^*) \quad (3.42)$$

Zusätzlich ist die Unsicherheit der Zustandsschätzung mit

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^* - \mathbf{K}_{k+1} \cdot \mathbf{P}_{yy} \cdot \mathbf{K}_{k+1}^T \quad (3.43)$$

zu aktualisieren.

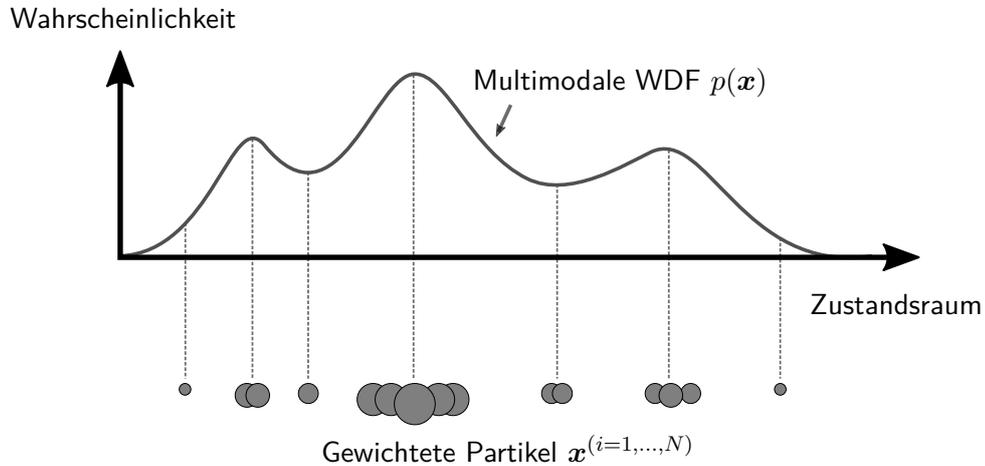
Für eine detaillierte Herleitung der in diesem Abschnitt verwendeten Gleichungen sei auf Julier et al. [1995] und Julier [2002] verwiesen.

Das in diesem Kapitel vorgestellte Kalman-Filter setzt normalverteiltes weißes System- sowie Messrauschen voraus und dient zur rekursiven Zustandsschätzung. Im nächsten Kapitel wird eine Möglichkeit vorgestellt, nicht-gaußsche multimodale Verteilungen zu schätzen.

### 3.2.2 Partikel-Filter

Dieses Kapitel beschreibt einen rekursiven Algorithmus zur Schätzung einer unbekannt multimodalen Wahrscheinlichkeitsdichte der Zustände eines dynamischen Systems. Es wird der zur Gruppe der Partikel-Filter zugehörige *Bootstrap Filter* [Gordon et al., 1993] vorgestellt. Die Grundidee liegt in der Approximation einer multimodalen Wahrscheinlichkeitsverteilung durch eine Vielzahl sogenannter Partikel (vgl. Abbildung 3.9). Dabei besteht jedes einzelne Partikel aus einem Gewicht und einer Zustandsrepräsentation bzw. Stützstelle der Wahrscheinlichkeitsverteilung, welche auch als Wahrscheinlichkeitsdichtefunktion (WDF) bezeichnet wird. Das probabilistische Partikel-Filter arbeitet in zwei Schritten. Im ersten Schritt erfolgt eine Prädiktion der WDF. Hierbei können stochastische Modelle der Systemdynamik berücksichtigt werden. In einem zweiten Schritt findet die Innovation durch Gewichtung und Neuverteilung der Partikel statt. Der rekursive Aufruf dieser beiden Schritte als Ganzes führt zu einem verbesserten Schätzergebnis der unbekannt Zustände des dynamischen Systems.

Es existiert eine Vielzahl von unterschiedlichen Implementierungen für die Neuverteilung der Partikel. Im Nachfolgenden wird auf die in dieser Arbeit verwendete Implementierung des Partikel-Filters sowie dessen Strategie zur Neuverteilung eingegangen.



**Abbildung 3.9:**

Approximation einer multimodalen Wahrscheinlichkeitsverteilung durch gewichtete Partikel (Kreise). Die Kreisgröße korreliert zum Partikelgewicht.

### 3.2.2.1 Initialisierung

Zu Beginn sollte ein Anfangsschätzwert  $\hat{\mathbf{x}}_{k-1} = \hat{\mathbf{x}}_0$  und die Anzahl an Partikel  $\mathbf{x}^{(i=1,\dots,N)}$  definiert werden. Anschließend sind mit der Vorgabe eines Streuungsbereiches die Zustandswerte  $\hat{\mathbf{x}}_0$  der Partikel zu berechnen. Im Folgenden approximieren die Partikel eine WDF  $p(\mathbf{x}_0)$ , die ein Maximum beim Anfangsschätzwert und Stützstellen im definierten Streuungsbereich besitzt.

### 3.2.2.2 Prädiktion

Als erstes wird die durch  $N$  Partikel  $\mathbf{x}^{(i=1,\dots,N)}$  approximierte WDF  $p(\mathbf{x})$  der Zustandsgrößen  $\mathbf{x}$  vom Zeitpunkt  $k - 1$  auf  $k$  prädiziert:

$$\mathbf{x}_k^{*(i)} = f_{k-1} \left\{ \hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}, \mathbf{s}_{k-1}^{(i)} \right\} \quad \text{für } i = 1, \dots, N. \quad (3.44)$$

Hierbei entspricht  $\hat{\mathbf{x}}_{k-1}^{(i)}$  dem vorherigen Schätzwert vom Partikel  $i$ ,  $\mathbf{u}_{k-1}$  optionalen Steuergrößen und  $\mathbf{s}_{k-1}^{(i)}$  den Störgrößen. Für eine realitätsnahe Vorhersage erfolgt die zeitliche Abbildung im Allgemeinen auf Basis eines Bewegungsmodells der Systemdynamik (Näheres siehe Anhang A.4).

Die zukünftige Messung  $\mathbf{y}_k^{*(i)}$  für jedes Partikel wird über eine zu definierende Abbildungsgleichung  $g\{\mathbf{x}_k^{*(i)}\}$  vorhergesagt:

$$\mathbf{y}_k^{*(i)} = g\{\mathbf{x}_k^{*(i)}\} + \mathbf{e}_k^{(i)} \quad \text{für } i = 1, \dots, N. \quad (3.45)$$

### 3.2.2.3 Innovation

Im zweiten Rekursionsschritt, der Innovation, erfolgt die Durchführung einer Messung  $\mathbf{y}_k^{(i)}$  zur Bewertung  $\mathbf{w}_k^{(i)}$  der Partikel bzw. der Zustandsinformationen, wodurch eine Neuverteilung ermöglicht wird. Eine Neuverteilung wird auch als *Resampling* bezeichnet und für die Durchführung existieren unterschiedliche Verfahren. Das sogenannte *Sequential Importance Resampling (SIR)* ist eine gängige Strategie zur Neuverteilung der Partikel und kommt in dieser Arbeit zum Einsatz. Im Nachfolgenden wird die Messung, Gewichtung sowie die Neuverteilung näher betrachtet.

**Messung und Gewichtung:** Der Vergleich einer Messung  $\mathbf{y}_k^{(i)}$  mit ihrer Vorhersage beschreibt die Gleichung 3.46.

$$\mathbf{y}_k^{(i)} - \mathbf{y}_k^{*(i)} = \mathbf{y}_k^{(i)} - \left(g\{\mathbf{x}_k^{*(i)}\} + \mathbf{e}_k^{(i)}\right) \quad (3.46)$$

Der daraus resultierende absolute Unterschied  $|\mathbf{y}_k^{(i)} - \mathbf{y}_k^{*(i)}|$  zwischen der Messwertvorhersage und der Wahrnehmung führt zu der Gewichtung  $\mathbf{w}_k^{(i)}$  des Partikels  $i$ . Je größer die Abweichung, desto niedriger fällt das Partikelgewicht aus. Abschließend erfolgt eine Normalisierung der Partikelgewichte, sodass die Summe der Gewichte eins ergibt:

$$\mathbf{w}_k^{(i)} = \frac{\mathbf{w}_k^{(i)}}{\sum_{j=1}^N \mathbf{w}_k^{(j)}} \quad \text{für } i = 1, \dots, N. \quad (3.47)$$

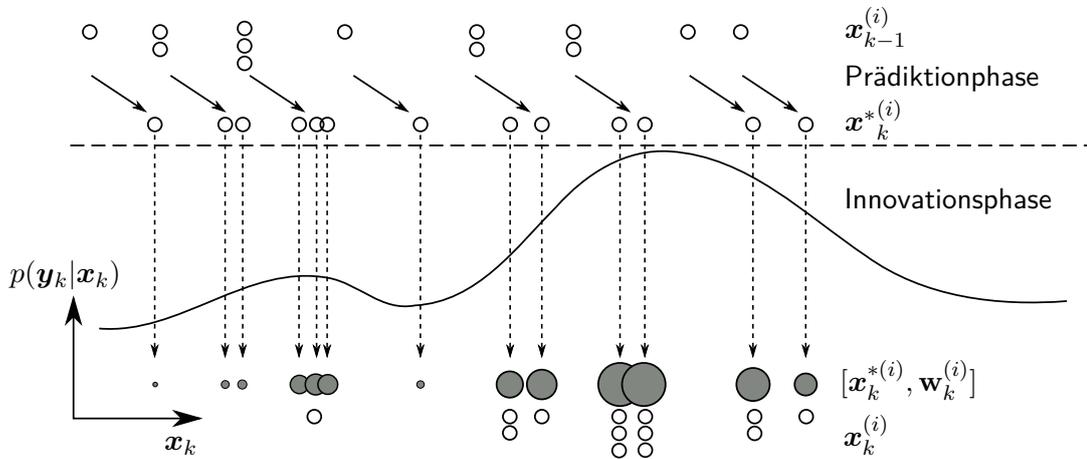
**Neuverteilung:** Die Neuverteilung der Partikel wird als *Resampling* bezeichnet und erfolgt auf Basis der normalisierten Partikelgewichte. Das bedeutet, dass Partikel mit niedrigem Gewicht tendenziell gelöscht und Partikel mit hohem Gewicht tendenziell dupliziert werden, sodass eine Vielzahl der Partikel an den Maxima der WDF liegen. Die Partikelanzahl bleibt allerdings immer konstant. Eine gewichts-abhängige Neuverteilung kann dazu führen, dass sich über die Zeit viele bis sogar sämtliche Partikel an einer Position im Zustandsraum häufen. Dies wird als Stichproben-Verarmung (engl. *Sample Impoverishment*) bezeichnet und kann mit gesonderten Methoden vermieden werden (siehe Simon [2006]). Eine Möglichkeit einer Partikelverarmung entgegen zu wirken ist das Hinzufügen von mittelwert-freien zufälligen Rauschen.

Eine Neuverteilung muss nicht zwangsläufig in jeder Rekursion erfolgen. Es kann auch die effektive Anzahl der Partikel

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N \left( \mathbf{w}_k^{(i)} \right)^2} \quad (3.48)$$

mit dem Wertebereich  $1 \leq N_{\text{eff}} \leq N$  berechnet werden. Besitzen alle Partikel das gleiche Gewicht, folgt dass alle  $N_{\text{eff}} = N$  Partikel sinnvoll genutzt werden. Besitzt nur ein Partikel ein echt positives Gewicht, folgt  $N_{\text{eff}} = 1$ , was eine Neuverteilung erzwingen sollte. Die Festlegung eines Schwellwertes  $N_{\text{eff,min}}$  ist somit ein experimentell zu ermittelnder Parameter und legt fest, wann eine Neuverteilung stattfindet.

Zusammenfassend veranschaulicht die Abbildung 3.10 den rekursiven Prädiktions- und Innovationschritt des Partikel-Filters.



**Abbildung 3.10:** Partikel-Filter: Prädiktions- und Innovationphase. Bilddarstellung nach Manz [2013].

Als übergeordneter Schätzwert kann bei einer unimodalen WDF der arithmetische Mittelwert der  $N$  höchstgewichteten Partikel mit

$$\hat{\mathbf{x}}_k = \frac{1}{N} \cdot \sum_{i=1}^N \hat{\mathbf{x}}_k^{(i)} = \sum_{i=1, \dots, g; g \in [1, \dots, N]} \left( \mathbf{x}_k^{*(i)} \cdot \mathbf{w}_k^{(i)} \right) \quad (3.49)$$

und die dazugehörige Kovarianz berechnet werden. In der Gleichung 3.49 entspricht  $\mathbf{w}_k^{(i)}$  dem normalisierten Gewicht des  $i$ -ten Partikels. Bei WDF mit mehreren Maxima sollte dagegen das Partikel mit dem höchsten Gewicht hergenommen werden:

$$\hat{\mathbf{x}}_k = \underset{\mathbf{x}}{\operatorname{argmax}} \left( p(\mathbf{x}_k | \mathbf{y}_k) \right). \quad (3.50)$$

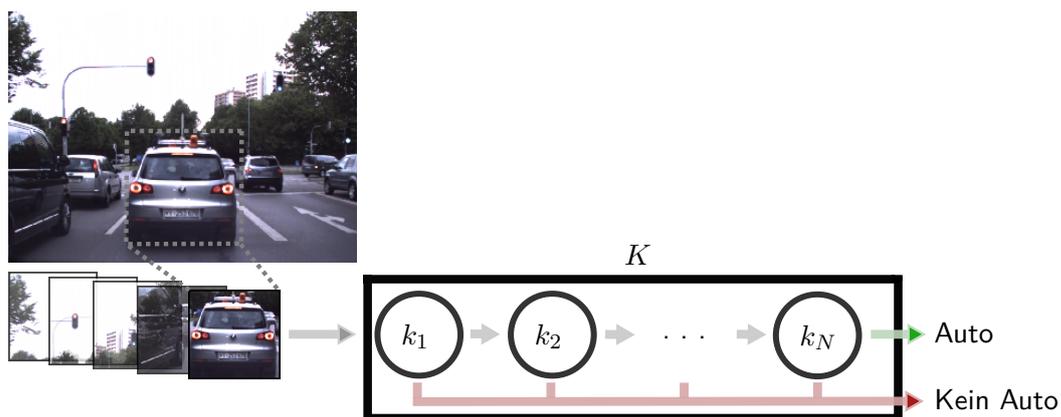
Mit dem in diesem Kapitel vorgestellten Partikel-Filter kann eine unbekannte multimodale WDF mit einer Vielzahl von Partikeln über die Zeit geschätzt werden. Der

Nachteil des Partikel-Filters liegt in der hohen Anzahl an Partikeln, welche für ein gutes Schätzergebnis erforderlich sind. Dies impliziert einen sehr hohen Rechenaufwand, da jedes Partikel propagiert und ausgewertet werden muss. Aus diesem Grund sollte die Implementierung derart erfolgen, dass die Partikel parallelisiert verarbeitet werden können.

### 3.3 Kaskadierte Klassifikatoren

Dieses Kapitel beschreibt verkettete Klassifikatoren für den Anwendungsbereich der visuellen Objekterkennung. Auf diesem Gebiet ist deren Aufgabe die korrekte Klassifizierung eines Bildausschnittes bezüglich der Präsenz eines Objekttyps bzw. einer Objektklasse.

Eine gängige Methode zur Objektklassifizierung ist die Verkettung sogenannter „schwacher“ Klassifikatoren  $k_i$ , welche im Einzelnen viele richtige und sehr viele fehlerhafte Klassifikationsentscheidungen treffen. Die Verkettung mehrerer schwacher Klassifikatoren kann zu einem „starken“ Klassifikator  $K$  führen, der überwiegend richtige Entscheidungen trifft. Die Abbildung 3.11 veranschaulicht eine gängige Verkettungsart in Reihe. Dort untersucht der erste schwache Klassifikator  $k_1$  einen Bildausschnitt auf ein sehr markantes Merkmal und dient damit zur groben Vorfiltrierung. Bei einer erfolgreichen Merkmalsdetektion sucht der zweite Klassifikator nach z. B. zwei anderen dominanten Merkmalen. Werden auch diese im Bildausschnitt erkannt, führt der nächste Klassifikator eine strengere bzw. komplexere Klassifizierung durch. Dieses Vorgehen ist aufgrund der frühzeitigen Aussonderung von Bildausschnitten sehr performant und erzielt als Ganzes eine hohe Erkennungsrate bei einer niedrigen Anzahl an Fehldetektionen.



**Abbildung 3.11:**

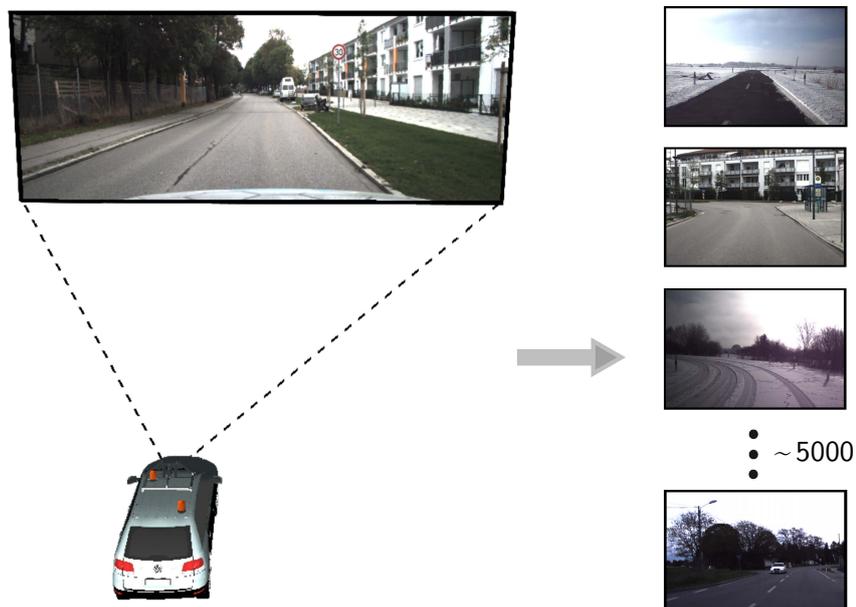
Kaskadierung von mehreren schwachen Klassifikatoren  $k_{1,\dots,N}$  zu einem starken Klassifikator  $K$ .

Das Trainieren eines Klassifikators basiert auf sogenannten „positiven“ und „negativen“ Datensätzen. Hinsichtlich der visuellen Objekterkennung sind die Datensätze Kamerabilder. Positive Datensätze beinhalten das zu klassifizierende Objekt sowie die Information über die Datenbereiche, die zum Objekt bzw. zum Hintergrund gehören. Dagegen bestehen die negativen Datensätze nur aus Hintergrundinformationen und zeigen das zu klassifizierende Objekt nicht. Mit den negativen und den gelabelten positiven Daten sowie einer vorgegebenen Merkmalsauswahl können maschinelle Lernverfahren einen starken Klassifikator ermitteln, der das gelabelte Objekt zuverlässig klassifizieren kann. Im Nachfolgenden wird auf die Trainingsphase näher eingegangen.

### 3.3.1 Trainingsphase

Zwei vorbereitende Schritte sind für die Trainingsphase vorab durchzuführen. Der erste Schritt ist die Aufzeichnung von Sensordaten und die Erstellung von positiven und negativen Datensätzen. Dabei ist besonders darauf zu achten, dass die Datensätze möglichst praxisbezogen, umfangreich und unterschiedlich sind. Beispielsweise soll ein Klassifikator für die Erkennung einer spezifischen Fahrzeugrückseite trainiert werden. In diesem Fall sollten sowohl die positiven als auch die negativen Datensätze viele Verkehrs- und Umgebungssituationen zeigen, die unter verschiedenen Licht- und Wetterbedingungen aufgenommen wurden. Dabei ist die Menge an Datensätzen für die in der Trainingsphase anzuwendenden maschinellen Lernalgorithmen sehr von Bedeutung. Je mehr unterschiedliche Datensätze in den Trainingsvorgang einfließen können, desto größer ist die Wahrscheinlichkeit für einen resultierenden Klassifikator, welcher zum einen eine hohe Erkennungsrate aufweist und zum anderen wenig Fehldetektionen liefert. Die meisten Publikationen begrenzen jedoch den zeitlichen Aufwand für das manuelle Labeln auf einige hundert Datensätze (siehe Hota et al. [2010] und Rezaei et al. [2015]). Dagegen erfordern die negativen Datensätze keinen manuellen Eingriff, wodurch schnell mehrere tausend Datensätze aufgenommen werden können. Zur Veranschaulichung zeigt die nachfolgende Abbildung 3.12 einige negative Kameraaufnahmen, die während einer Autofahrt aufgenommen wurden.

Für die Erstellung der positiven Datensätze existieren zwei Möglichkeiten. Das verbreitetste Vorgehen ist das manuelle Labeln in aufgezeichneten Sensordaten. Hierzu müssen Kamerabilder aufgezeichnet werden, die das Objekt beinhalten, welches der resultierende Klassifikator wiedererkennen soll. Anschließend erfolgt das manuelle Labeln pixelweise oder konturbasiert. Das pixelweise Labeln ist extrem zeitaufwendig. Der Cityscapes-Datensatz [Cordts et al., 2016] beinhaltet zum Beispiel mehrere tausend annotierte Kamerabilder. Der Labelvorgang dauerte mehr als 1,5 Stunden pro Bild. Das konturbasierte Labeln kann schneller durchgeführt werden, jedoch ist dieses Vorgehen ebenfalls sehr zeitaufwendig, wenn mehrere tausend Kamerabilder annotiert werden müssen. Zur Veranschaulichung illustriert die Abbildung 3.13 das konturbasierte Labeln, durch die Selektion von vier Bildpunkten, welche jeweils einen Objektumriss kennzeichnen.

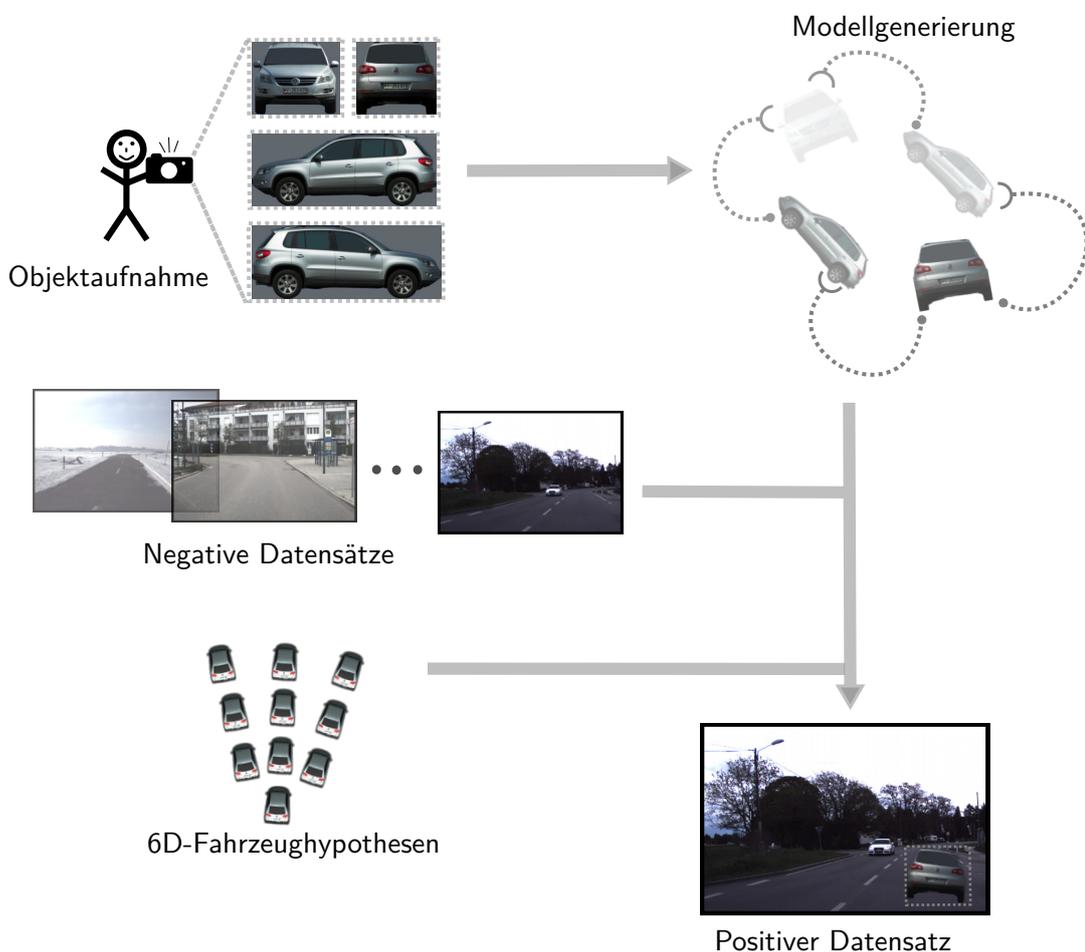


**Abbildung 3.12:**  
Aufnahme von negativen Kamerabildern, welche das zu klassifizierende Objekt nicht darstellen.



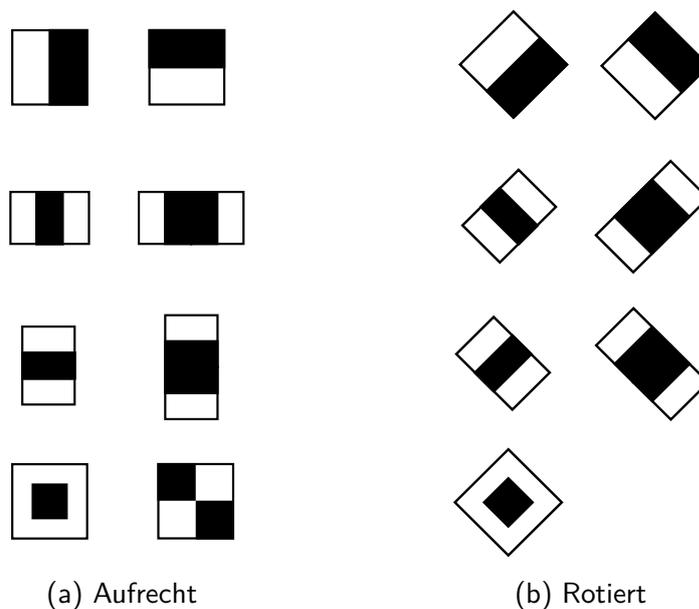
**Abbildung 3.13:**  
Zeitaufwendiges manuelles Labeln von mehreren tausend Kamerabildern.

Eine alternative Möglichkeit für die Erstellung der positiven Datensätze wurde in Fries und Wuensche [2014] vorgestellt. Dort ist das manuelle Labeln nicht erforderlich, sondern der Benutzer muss lediglich die vier orthogonalen Objektseiten einmalig aufnehmen. Anschließend kann automatisiert anhand der vier orthogonalen Aufnahmen ein 3D-Modell erstellt werden. Mit der Kombination aus einem 3D-Modell und den negativen Datensätzen sind die positiven Datensätze automatisch erzeugbar. Dafür wird das 3D-Modell in den negativen Datensätzen mit unterschiedlichen Positionen und Orientierungen integriert. Eine Möglichkeit ist die zufällige Erstellung von ego-relativen 6D-Objektposen. Anschließend werden die Objektposen mit dem vorab-erstellten 3D-Modell in das Kamerabild projiziert und die vier Bildpunkte berechnet, welche die Objekthülle (engl. *Bounding Box*) beschreiben. Dabei sollten die projizierten objekt-zugehörigen Pixel zufällig belichtet bzw. abgedunkelt werden, sodass ein praxisnäherer bzw. generalisierter Klassifikator trainiert werden kann. Eine Illustration der halb-automatischen Objektmarkierung zeigt die Abbildung 3.14.



**Abbildung 3.14:** Halb-automatisches Labeln von mehreren tausend Kamerabildern.

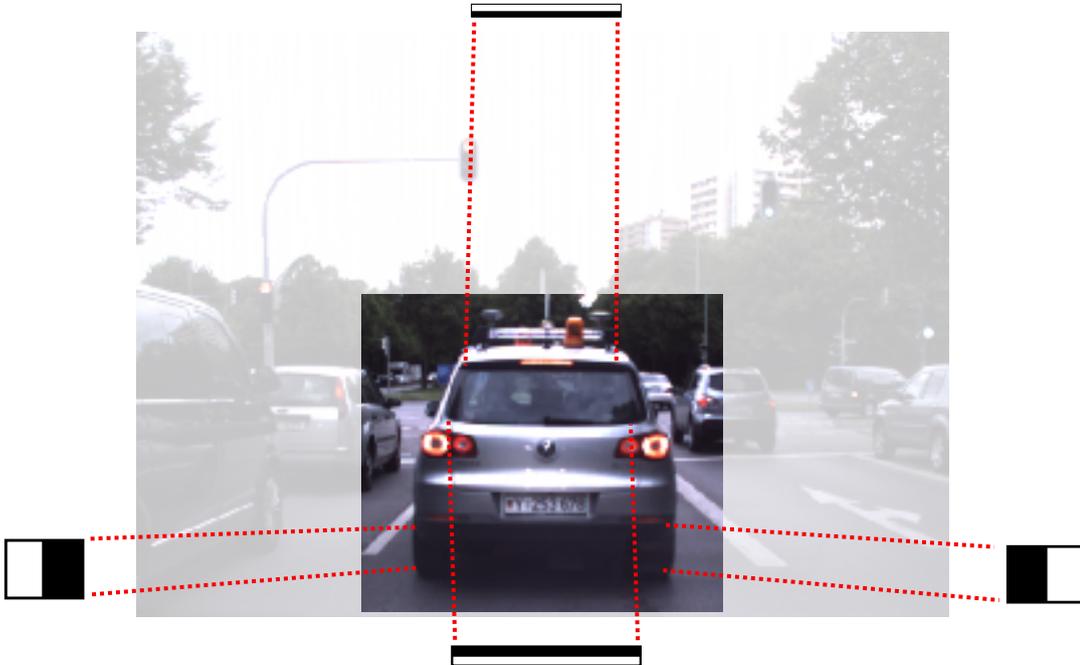
Die Festlegung der Merkmalstypen ist der zweite und letzte vorzubereitende Schritt, bevor die Trainingsphase eines Klassifikators begonnen werden kann. Zu den gängigen Merkmalstypen zählen die gradienten-basierten HOG- [Dalal und Triggs, 2005], pixel-basierte Local Binary Pattern (LBP)- [Ojala et al., 1994] und Haar-like- [Viola und Jones, 2001b] Merkmale. Dabei sind Haar-Merkmale aufgrund der schnellen Verarbeitung am verbreitetsten. Acht aufrechte und sieben rotierte Haar-Merkmale zeigt die Abbildung 3.15. Dort besteht jedes Merkmal aus weißen und schwarzen Flächen. Jede weiße Fläche besitzt ein negatives und jede schwarze ein positives Gewicht. Insgesamt ist ein solches Merkmal eine Bildmaske, bei der sich das Maskenergebnis aus der Summe der einzelnen Gewichte ergibt. Die Merkmalstypen sind von Viola und Jones [2001b] für unterschiedliche Bildmuster konstruiert worden. Die Merkmale sind mit Absicht recht einfach gehalten und besitzen keine komplexen Strukturen. Der Fokus liegt hierbei in einem Merkmalsset, welches aus ausreichend einfachen Merkmalen besteht, die in zahlreichen Kombinationen sämtliche Kanten- und Eckregionen eines Objektes eindeutig beschreiben können. Die Verarbeitung solcher einfachen Flächenmerkmale ist in Echtzeit durchführbar. Dafür erfolgte in Viola und Jones [2001b] und Lienhart und Maydt [2002] die Veröffentlichung von aufrechten und um  $45^\circ$ -rotierten Integralbildern, in denen sich Flächen durch jeweils eine Addition und zwei Subtraktionen ermitteln lassen.



**Abbildung 3.15:** Flächenbasierte Haar-Merkmale.

Nach der Festlegung der zu verwendenden Merkmalstypen und der Erstellung von positiven und negativen Datensätzen kann die Trainingsphase erfolgen. In dieser sucht ein maschineller Lernprozess aus dem vorgegebenen Pool von unterschiedlichen Merkmalen diejenigen Merkmale, die das gelabelte Objekt zuverlässig beschreiben. Dabei ist das Suchkriterium, diejenigen Objektmerkmale zu finden, welche in den meisten positiven Datensätzen wiedererkennbar sind, die jedoch nicht im Objekt-

hintergrund bzw. in den negativen Datensätzen auftauchen. Dadurch erfolgt nicht nur die Integration von gelabeltem Objektwissen, sondern die Information über die Objektumgebung wird zusätzlich berücksichtigt. Eine Visualisierung von markanten Objektmerkmalen - die mit einem maschinellen Lernverfahren ermittelt wurden - illustriert die Abbildung 3.16.



**Abbildung 3.16:**

Illustration von Haar-Merkmalen, die ein maschineller Lernprozess an vier Bildpositionen als Alleinstellungsmerkmal identifizierte.

Das bekannteste maschinelle Lernverfahren für das Trainieren eines Klassifikators ist der *Boosting*-Ansatz namens Adaptive Boosting (AdaBoost) [Freund und Schapire, 1995]. Der Ansatz führt die Selektierung von Merkmalen aus einem vorgegebenen Merkmalspool durch, evaluiert diese auf den Datensätzen und verändert dabei die Gewichte der schwachen Klassifikatoren sowie der einzelnen Merkmale adaptiv über die Zeit. Im Laufe der Zeit haben sich weitere *Boosting*-Trainingsstrategien entwickelt. Zu den bekannten Strategien zählen Discrete AdaBoost (DAB), Real AdaBoost (RAB), Gentle AdaBoost (GAB) und Logit Boost (LB) [Friedman et al., 2000, Viola und Jones, 2001b, Hastie et al., 2008]. Dabei stellt sich für jeden Anwendungsfall die Frage, welche Trainingsstrategie und welche Merkmalstypen für das Trainieren eines Klassifikators am besten geeignet ist. Je nach Anwendungsfall, Merkmalsauswahl und in Abhängigkeit von der Parameteranzahl führen die Trainingsalgorithmen zu unterschiedlichen Ergebnissen bzgl. Performance und Klassifikationsgenauigkeit.

### 3.3.2 Anwendung

Die Anwendung eines trainierten Klassifikators erfolgt auf dem sogenannten *Sliding Window*-Prinzip. Ein Suchfenster wird in unterschiedlichen Skalierungen über eine Kameraaufnahme geschoben und mit dem kaskadierten Klassifikator auf Merkmale untersucht. Das Vorgehen ist in der bereits vorgestellten Abbildung 3.11 veranschaulicht. Hierbei klassifizieren schwache Klassifikatoren sequenziell Bildausschnitte auf die Präsenz markanter Fahrzeugmerkmale. Werden die in der Trainingsphase definierten markanten Merkmale erfolgreich erkannt, dann erfolgt eine positive Rückmeldung bzgl. der Bildauswertung hinsichtlich eines Fahrzeuges.

## 3.4 Region Growing

In der Bildverarbeitung ist die Segmentierung von Pixelbereichen ein gängiger Vorverarbeitungsschritt. Der Bereich der visuellen Objekterkennung nutzt die Segmentierung zur Datenreduktion. Dieses Kapitel beschreibt einen Algorithmus zur Erkennung von Bildarealen mit ähnlichen Pixelwerten und stellt einen für den Bereich der Objekterkennung angepassten Segmentierungsalgorithmus vor.

### 3.4.1 Algorithmus und Varianten

Der sogenannte *Region Growing*-Algorithmus vergleicht ausgehend von einem Saatpixel (engl. *Seed Point*) die benachbarten Pixelwerte und pflanzt sich von dort aus kontinuierlich fort. Für die Art und Weise der Fortpflanzung sind unterschiedliche Vorgehensweisen gängig. Ein üblicher Nachbarschaftsvergleich ist die Von-Neumann-Nachbarschaft, bei der die jeweils vier orthogonalen Pixelwerte - oben, unten, links, rechts - mit dem Saatpixel verglichen werden. Korrespondiert der Pixelwert  $I(u, v)$  mit dem Wert des Saatpixels  $I(u_s, v_s)$ , pflanzt sich der Algorithmus über den Nachbarpixel fort. Dabei ist für den Farb- oder Intensitätsvergleich ein gewisser Spielraum  $\Delta$  empfehlenswert:

$$I(u, v) = \begin{cases} I(u_s, v_s), & \text{falls } |I(u, v) - I(u_s, v_s)| < \Delta \\ I(u, v), & \text{sonst} \end{cases} \quad (3.51)$$

Als Alternative zur Von-Neumann-Nachbarschaft können auch alle acht Nachbarn mit dem Saatpixel verglichen werden. Diese Betrachtung wird als Moore-Nachbarschaft bezeichnet und ist deutlich zeitaufwendiger, jedoch werden dadurch nicht nur Kanten sondern auch Ecken berücksichtigt. Des Weiteren muss das Fortpflanzungskriterium nicht fest gesetzt sein. Zum Beispiel kann ausgehend von einem Saatpixel nach jeder Fortpflanzung ein neuer Vergleichswert bestimmt werden. Die Differenz zum

aktuellen Pixel oder die Berechnung des arithmetischen Mittelwertes der vier oder acht Nachbarn wäre als neuer Vergleichswert möglich.

Der Anwendungsbereich des *Region Growing*-Algorithmus ist in den meisten Fällen die Segmentierung eines gesamten Bildes. In diesem Fall reicht ein einzelner Saatpunkt typischerweise nicht aus. Stattdessen erfolgt eine (gleichverteilte) Streuung von Saatpunkten über das gesamte Bild. Von diesen Saatpunkten erfolgt eine parallele Fortpflanzung und es entstehen Segmentierungsbereiche. Anschließend werden so lange neue Saatpunkte erzeugt, bis jeder Pixel einer Region zugeordnet wurde.

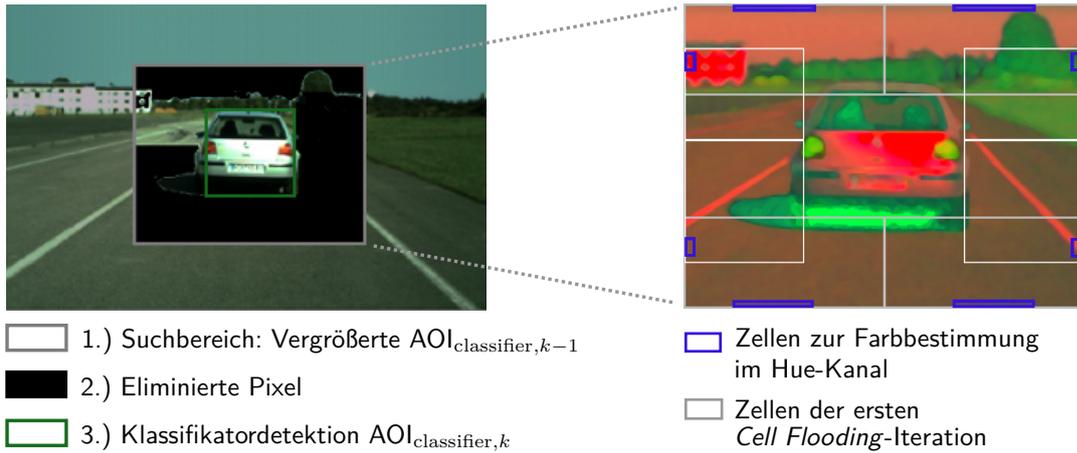
Des Weiteren hat sich der Grundalgorithmus weiterentwickelt und diverse anwendungsspezifische Varianten wurden im Laufe der Zeit publiziert. Darunter fallen der *Fuzzy*- und der *Seeded-Region Growing*-Algorithmus, welche in Kang und Wang [2009] und Adams und Bischof [1994] beschrieben sind. Im nachfolgenden Abschnitt wird eine Variante vorgestellt, welche speziell für den Bereich der Objekterkennung entwickelt wurde.

### 3.4.2 Multi-Dynamic Region Growing

Das Besondere am *Region Growing*-Algorithmus ist das gute Segmentierungsergebnis unter Berücksichtigung von dominanten Kanten. Diese Eigenschaft wird in diesem Unterkapitel für eine Objekterkennung bzw. zur Eliminierung von Pixeln, die nicht zu einem Objekt gehören, verwendet. Dabei erfolgt die Annahme, dass die Umrisse jedes Objektes visuell erkennbar bleiben, auch wenn der Hintergrund eine ähnliche Farbe aufweist. In Extremsituationen, bei denen Objektbereiche mit dem Hintergrund farblich übereinstimmen, kann eine Person Kantenregionen, die visuell nicht vorhanden sind, voraussagen. Dies ist aufgrund der benachbarten Kantenregionen möglich, die auf den nicht eindeutig erkennbaren Bildbereich prädiert werden.

In diesem Unterkapitel wird ein *Multi-Dynamic Region Growing*-Algorithmus namens *Cell Flooding* beschrieben, welcher in Abhängigkeit von benachbarten Bildbereichen ein *Region Growing* durchführt und bereits in Fries et al. [2013] und Fries und Wuensche [2014] veröffentlicht wurde. Dabei ist das übergeordnete Ziel die Eliminierung (Schwärzung) des Objekthintergrundes, sodass eine anschließende Objektklassifizierung auf einem reduzierten Pixelbereich durchgeführt werden kann. Der Algorithmus setzt grobes Positionswissen über das Objekt voraus und nutzt dieses zur Steuerung der Pixeleliminierung. Dafür wird die Annahme getroffen, dass sich die Objektkontur zu den benachbarten Hintergrundpixeln größtenteils unterscheidet.

Im Nachfolgenden wird der Algorithmus am Beispiel der visuellen Fahrzeugerkennung erläutert. Die Eliminierung von Bildpixeln erfolgt in einem vorgegebenen Interessensbereich, engl. *Area of Interest* (AOI), wie in der Abbildung 3.17 dargestellt. Hierbei wurde ein Fahrzeug mit einem Klassifikator zum Zeitpunkt  $k$  grob detektiert. Folglich kann in Abhängigkeit von der Detektionsgenauigkeit und der vergangenen Zeit

**Abbildung 3.17:**

Objektbezogene Pixeleliminierung mit dem *Multi-Dynamic Region Growing*-Verfahren.

zwischen  $k - 1$  und  $k$  der Suchbereich  $AOI_{\text{classifier}}$  bestimmt werden. Anschließend ist in diesem Bereich der *Multi-Dynamic Region Growing*-Algorithmus anzuwenden. Die AOI wird dabei in Zellen eingeteilt, in denen ausgehend vom AOI-Rand in Richtung zum AOI-Zentrum das *Cell Flooding* (Pixeleliminierung) stattfindet. Hierbei steigt zum AOI-Mittelpunkt hin die Wahrscheinlichkeit, dass es sich um einen fahrzeugzugehörigen Pixel handelt.

Die Bestimmung, ob ein Pixel zum Hintergrund oder zum Fahrzeug gehört, erfolgt im HSV-Farbraum, welcher im Anhang A.3.2 beschrieben ist. Dieser ist im Gegenteil zum RGB-Farbraum für einen Farbvergleich besser geeignet. Jedes Pixel  $(u, v)$  besitzt damit einen Farbwert  $h$ , einen Sättigungswert  $s$  und eine Helligkeitsintensität  $v$ . Jedes Pixel, das im Vergleich zum Saatpixel  $(u_s, v_s)$  den gleichen  $(h, s, v)$ -Wert besitzt, wird dabei eliminiert bzw. geschwärzt. Die Farbwertvergleiche finden vom äußeren AOI-Rand bis zum -Zentrum statt, so lange bis die Farbwerte sich zu stark verändern. Dabei wird das Abbruchkriterium mit dem sinkenden Abstand zum AOI-Zentrum immer stärker verschärft. Der Pixelvergleich mit einer zell- und positionsabhängigen Gewichtung  $w_1$  und  $w_2$  ist durch die Gleichung 3.52 beschrieben.

$$I(u, v) = \begin{cases} 0, & \text{falls } (|I(u, v)_h - I(u_s, v_s)_h| < (\Delta h \cdot w_1 \cdot w_2)) \wedge \\ & (|I(u, v)_s - I(u_s, v_s)_s| < (\Delta s \cdot w_1 \cdot w_2)) \wedge \\ & (|I(u, v)_v - I(u_s, v_s)_v| < (\Delta v \cdot w_1 \cdot w_2)) \\ I(u, v), & \text{sonst} \end{cases} \quad (3.52)$$

Zusätzlich beeinflussen benachbarte Zellen das Abbruchkriterium pro Zelle ebenfalls, was ein Überschreiten der Fahrzeugkontur vermeidet.  $\Delta h$ ,  $\Delta s$  und  $\Delta v$  sind festgelegte Schwellwerte, die bestimmen, wie „gierig“ der *Region Growing*-Algorithmus in jeder Zelle fortschreitet. Nach Fries et al. [2013] sollte für den Farbton  $\Delta h$  ein niedriger Wert - wie bspw. 1 - und für die Sättigung  $\Delta s$  und die Helligkeitsintensität  $\Delta v$  ein höherer Wert - z. B. 5 - gewählt werden.

Zusammenfassend zeigt die Abbildung 3.17 ein Ergebnis im Bereich der visuellen Fahrzeugerkennung. Im linken Teil der Abbildung ist das Suchfenster durch ein graues Rechteck visualisiert. Dieses RGB-Bild wurde in den HSV-Raum transformiert, was in der rechten Grafik zu sehen ist. Dort zeigen die hellgrauen Rechtecke die einzelnen Zellen, in denen jeweils *Region Growing* angewendet wurde. Dabei liegt der zellabhängige Saatpunkt in dem Mittelpunkt eines festgelegten Bereiches, welcher durch ein blaues Rechteck gekennzeichnet ist. Jeder Saatpunkt erhält den arithmetischen Mittelwert der Pixelwerte des zugehörigen blauen Rechteckes zugewiesen. Anschließend kann mit diesem Saatpunkt das *Cell Flooding* erfolgen, wobei das Vergleichskriterium zum Bildmittelpunkt und in Abhängigkeit von den Nachbarzellen verschärft wird. Abschließend ist die Fahrzeugerkennung auf den Bildbereich  $AOI_{\text{classifier},k-1}$  anzuwenden. Aufgrund der großflächigen Pixelschwärzung kann das Fahrzeug zuverlässig und schnell wiedererkannt werden.



## 4 Modellgenerierung

### Inhalt

---

4.1	Manuelle Modellgenerierung . . . . .	51
4.2	Automatische Modellgenerierung . . . . .	54
4.2.1	Identifizierung des OOI . . . . .	56
4.2.2	Sensordatenreduktion . . . . .	56
4.2.3	Merkmalsdetektion . . . . .	58
4.2.4	Aktualisierung des 3D-Merkmalsmodells . . . . .	65

---

Die in dieser Arbeit vorgestellten Algorithmen zur Fahrzeugdetektion basieren auf 3D-Modellen, welche die fahrzeugspezifischen Merkmale beschreiben. In diesem Kapitel wird auf die manuelle und automatische Erstellung von 3D-Merkmalsmodellen eingegangen.

### 4.1 Manuelle Modellgenerierung

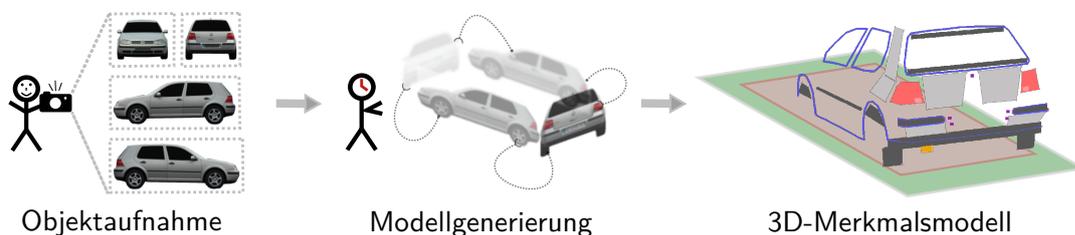
Die nachfolgend beschriebene manuelle Modellgenerierung stammt von Manz [2013] und wurde in dieser Arbeit unter anderem mit Temperaturmerkmalen erweitert. Die manuelle Modellgenerierung besteht aus fünf grundlegenden Schritten:

- Aufnahme der Fahrzeugseiten
- Bildskalierung
- Selektion signifikanter Bildmerkmale
- Positionierung der Bildmerkmale im 3D-Raum
- Selektion hindernisbehafter und hindernisfreier Untergrundzellen

Zu Beginn müssen mit einer Kamera die vier Fahrzeugseiten fotografiert werden. Dabei erleichtern orthogonal aufgenommene Kamerabilder die darauf aufbauende Modellgenerierung. Zusätzlich wird die Positionierung eines Meterstabs am Fahrzeugrand empfohlen, da die Kameraaufnahmen typischerweise nicht aus der exakt gleichen Entfernung zum Fahrzeug aufgenommen werden und dadurch eine divergente Pixeldarstellung zwischen den Aufnahmen entsteht. Ein am Fahrzeug positionierter Meterstab kann somit zur Korrektur der Bildskalierung genutzt werden und dient

ebenfalls zur maßstabgetreuen Modellgenerierung. Als Alternative kann auch bekanntes Modellwissen, wie unter anderem die standardisierte Geometrieinformation über deutsche Pkw-Kennzeichen, herangezogen werden.

Im nächsten Schritt erfolgt die Selektion fahrzeugspezifischer Bildmerkmale. Die Auswahl der Merkmale hängt von der möglichen Verwendung von unterschiedlichen Sensoren ab. Steht für die spätere Fahrzeugerkennung eine Farb- und Wärmekamera zur Verfügung, empfiehlt es sich, Eckpunkte, Kanten, farbige und warme Regionen als fahrzeugspezifische Merkmale zu selektieren. Markante Eckpunkte finden sich unter anderem am Kennzeichenrand. Innenliegende Fahrzeugkonturen, die einen starken Intensitätsübergang besitzen, können ebenfalls zuverlässig wiedererkannt werden. Als farbige Flächen eignen sich zum Beispiel zwei rote Rücklichter, zwei helle Fahrlichter und dunkle Bereiche am Reifen und unter dem Fahrzeug. Als Wärmeflächen kommen zum einen der heiße Auspuff und zum anderen die – sich durch Reibung aufheizenden – Reifen in Frage. Im Anschluss an die Selektierung der Bildmerkmale erfolgt die manuelle Positionierung im 3D-Raum. In dem Fall, dass für die spätere Fahrzeugerkennung ein LiDAR-Sensor zur Verfügung steht, werden abschließend hindernisbehaftete und hindernisfreie Untergrundzellen definiert. In der Regel ist die unmittelbare Fläche direkt um ein Fahrzeug herum hindernisfrei. Im Gegensatz dazu ist der Bereich des Fahrzeuges hindernisbehaftet. Die Abbildung 4.1 veranschaulicht die einzelnen Schritte einer manuellen Modellgenerierung und visualisiert ausgewählte Merkmale eines VW Golf. In der Abbildung 4.2 sind drei weitere 3D-Merkmalmodelle eines VW Tiguan dargestellt.



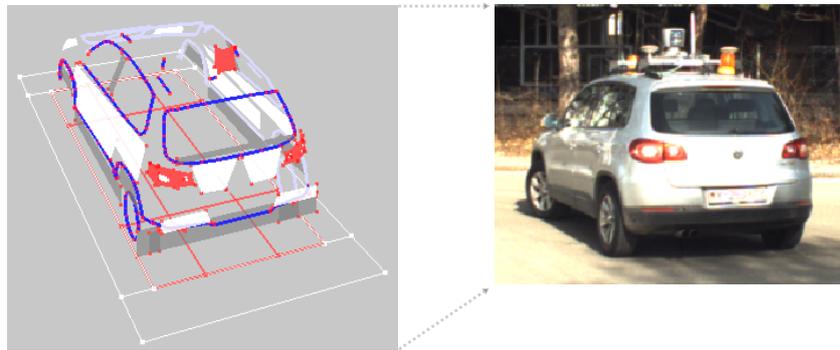
**Abbildung 4.1:**

Erstellung eines 3D-Merkmalmodells, bestehend aus Eckpunkten (lila Punkte), Kanten (blaue Linien), farbige / warme Regionen (Vierecke) sowie hindernisbehaftete und hindernisfreie Untergrundzellen. Grün dargestellte Untergrundzellen kennzeichnen Bereiche, die gängigerweise hindernisfrei sind. Im Gegensatz dazu sind die Zellen unterhalb des Fahrzeuges durch das Fahrzeug belegt.

Für die manuelle Modellgenerierung empfiehlt sich die Verwendung einer 3D-Grafiksoftware wie beispielsweise Blender<sup>1</sup>. Der zeitliche Aufwand einer manuellen Modellgenerierung hängt vom Detaillierungsgrad ab. Ein erfahrener Modellierer kann ein einfaches Modell in ein paar Stunden erstellen. Im Gegenteil dazu dauert die Verbesserung bzw. das Feintuning mehrere Tage.

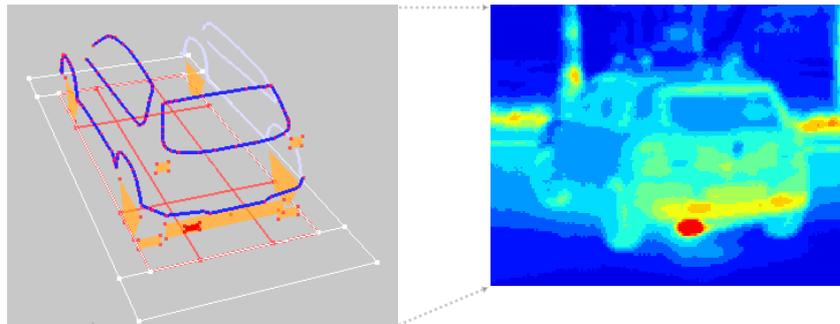
<sup>1</sup>[www.blender.org](http://www.blender.org)

Um diesen hohen zeitlichen Aufwand zu reduzieren, stellt das nächste Kapitel eine automatisierte Vorgehensweise zur Erstellung von 3D-Merkmalmodellen vor.



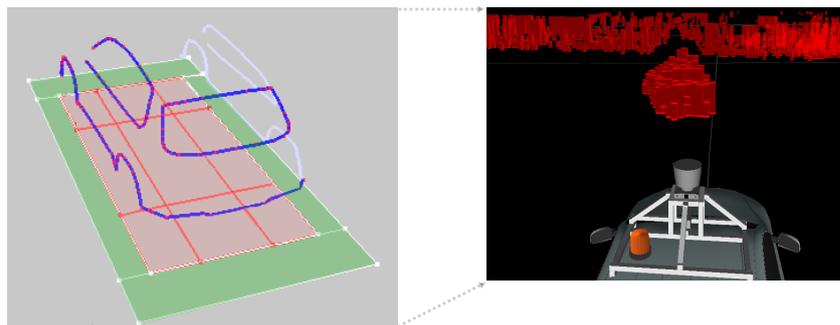
- Rote Flächen (Rücklichter, Warnlicht)
- Dunkle Flächen (Reifen, Unterboden)
- Helle Flächen (Scheinwerfer, Karosserie)
- Kanten

(a) Merkmalsmodell für Farbinformationen



- Heiße Flächen (Auspuff)
- Warme Flächen (Auspuffbereich, Lichter, Reifen)
- Kanten im Wärmebild

(b) Merkmalsmodell für Wärmedaten



- Hinderniszellen (variieren je nach perspektivischer Betrachtung)
- Hindernisfreie Zellen

(c) Merkmalsmodell für Tiefendaten (LiDAR, Stereo-Kamerasystem)

**Abbildung 4.2:** Sensorabhängige 3D-Merkmalmodelle eines VW Tiguan.

## 4.2 Automatische Modellgenerierung

In diesem Kapitel wird eine automatische Modellgenerierung vorgestellt, die ein maßstabgetreues 3D-Merkmalmodell von einem sogenannten Objekt von Interesse, engl. *Object of Interest* (OOI), erzeugt. Als Grundlage dienen Sensordaten, die das OOI aus unterschiedlichen Perspektiven und Entfernungen darstellen. Im Nachfolgenden erfolgt die Vorstellung eines echtzeitfähigen Verfahrens, sodass bereits direkt im Anschluss an die Datenaufnahme ein vollständiges 3D-Modell vorliegt und für eine Fahrzeugerkennung im autonomen Konvoi verwendet werden kann.

Der Sichtbereich der zu verwendenden Sensoren legt die Art der Datenaufzeichnung fest. In dieser Arbeit erfolgt die Aufzeichnung mit dem mobilen Fahrzeug MuCAR-3, das mit unterschiedlichen Sensoren ausgestattet ist. Es gibt viele mögliche Arten von Trajektorien zur Datenaufzeichnung. An dieser Stelle sollen zwei sehr unterschiedliche Trajektorien vorgestellt und bewertet werden. Die einfachste und schnellste Trajektorie ist das kreisförmige Umfahren des OOI. Bei einem 360°-LiDAR-Sensor und bewegbaren Kameras ist die sensorielle Fokussierung auf das OOI durchführbar. Dagegen ist das kreisförmige Umfahren nicht zur Datenaufzeichnung geeignet, wenn die Sensoren das OOI während der Fahrt nicht im Sichtbereich haben. Dies ist zum Beispiel bei nicht bewegbaren Sensoren mit einem kleinen Sichtbereich der Fall. Hierbei empfiehlt sich das Anfahren perspektivisch unterschiedlicher Positionen, bei denen das OOI in allen Sensordaten sichtbar ist. Diesbezüglich veranschaulicht die Abbildung 4.3 eine Datenaufzeichnungsfahrt, bei der vier Fahrzeugpositionen mit jeweils orthogonaler Ausrichtung angefahren wurden.

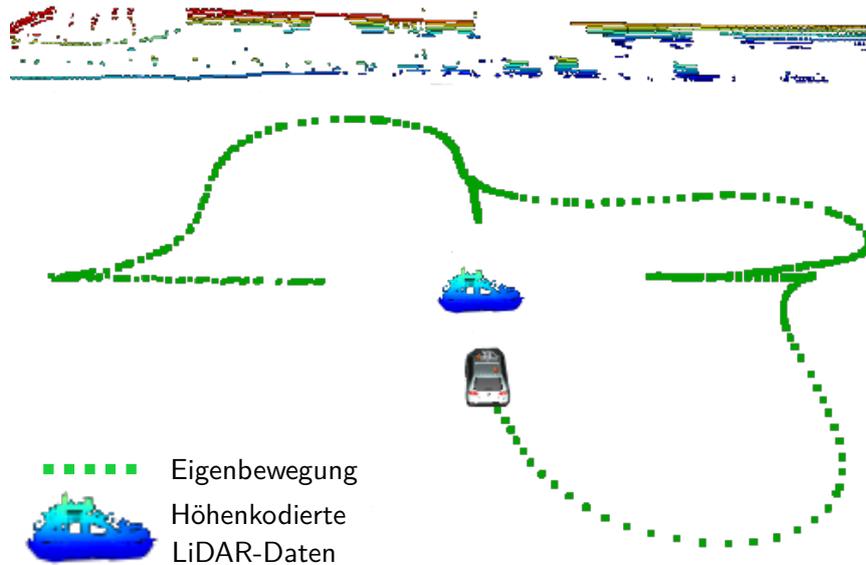
Während der Datenaufzeichnung ist eine automatisierte Modellerzeugung in Echtzeit möglich. Dabei ist die Erzeugung eines detaillierten bzw. umfangreichen 3D-Merkmalmodells nicht das Ziel dieser Arbeit. Je mehr Merkmale in einem Modell hinterlegt werden, desto größer wird der Verarbeitungsaufwand im Anwendungsfall der modellbasierten Fahrzeugerkennung (vgl. Kapitel 5). Demzufolge sollte das Modell nur die besonders markanten Merkmale beinhalten, sodass die spätere Verwendung zur modellbasierten Fahrzeugerkennung in Echtzeit erfolgen kann.

Die Modellgenerierung besteht aus den folgenden vier Phasen, die iterativ durchzuführen sind:

1. Identifizierung des OOI
2. Sensordatenreduktion
3. Merkmalsdetektion
4. Aktualisierung des 3D-Merkmalmodells



(a)



(b)

**Abbildung 4.3:**

Exemplarische Trajektorie zur Datenaufzeichnung: a) Aktuelle Aufnahme der Fahrzeugkamera. b) Anfahren von vier orthogonalen Perspektiven, in denen die sensorielle Fokussierung auf das OOI gerichtet ist.

Diese Phasen werden solange rekursiv durchgeführt, bis ein Aufzeichnungsfahrzeug das OOI aus allen Ansichten sensorbasiert erfasst hat. Zur Vereinfachung empfiehlt sich das automatische Starten der Modellgenerierung, wenn das mit Sensoren ausgestattete Aufzeichnungsfahrzeug still steht. Dadurch kann der Bediener das Aufzeichnungsfahrzeug in eine sensoroptimale Position fahren, bei der das OOI in allen Sensoren erfasst wird. Beispielsweise könnte das Aufzeichnungsfahrzeug mit Kameras, die lediglich über einen starren Blickwinkel in Fahrtrichtung verfügen, direkt in Längsrichtung zum OOI positioniert werden. Anschließend ist der Fahrzeugstillstand anhand der Odometrie- und INS-Daten feststellbar und die Modellgenerierung startet automatisch.

### 4.2.1 Identifizierung des OOI

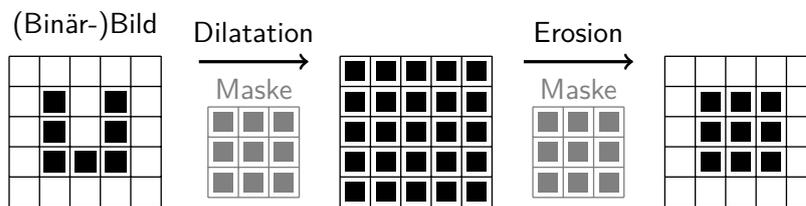
In der ersten Phase der Modellgenerierung findet die Identifizierung des OOI statt. Diesbezüglich erfolgt die Annahme, dass der Betrachter – in diesem Fall das mit Sensoren ausgestattete Fahrzeug – den sensorischen Fokus auf das OOI legt. Folglich müsste ein in Fahrtrichtung ausgesandter Sichtstrahl auf das unbekannte OOI und damit auf ein Hindernis treffen. Dies trifft allerdings nur zu, wenn die Sensoren auch in Fahrtrichtung orientiert sind. Aus diesem Grund ist die Konstruktion eines Sichtstrahls auf Basis der intrinsischen und extrinsischen Kameraparameter die richtige Vorgehensweise. Ausgehend von der bekannten Kameraposition, ihrer Orientierung und dem Kamerazentrum kann die Projektion eines Sichtstrahls über den Bildmittelpunkt hinaus in die 3D-Welt erfolgen. Dieser Vorgang ist als sogenannte Rückprojektion (engl. *Back Projection*) bekannt und erfolgt bis zu einer festgelegten Strahlentfernung bzw. Tiefe. Nach der Berechnung des Sichtstrahls ist dieser zu diskretisieren, sodass die erste Hindernisberührung bestimmt werden kann. Die Hindernisinformationen stammen dabei von einem LiDAR-Sensor oder Stereosystem. Die Verarbeitung der Rohdaten (3D-Laserpunkte, Disparitäten) wird an dieser Stelle aus Gründen der Robustheit und Echtzeitfähigkeit nicht empfohlen. Beispielsweise trifft in den seltensten Fällen ein Sichtstrahl auf einen 3D-Laserpunkt. Ebenfalls ist die allgemeine Verarbeitung von hunderttausenden von 3D-Laserpunkten sehr zeitintensiv. Aus diesen Gründen wird die Erzeugung eines zeitlich akkumulierten 2,5D-Hindernisgitters aus den Rohdaten des LiDAR-Sensors bzw. Stereosystems empfohlen. Ein solches Gitter beschreibt die ego-relativen Hinderniszellen mit jeweils einer Hinderniswahrscheinlichkeit sowie einer Höheninformation, falls mehr als zwei 3D-Punkte auf eine Zelle treffen. Dabei sollte die Zellgröße abhängig vom LiDAR-Sensor bzw. Stereosystem gewählt werden, sodass das gesamte OOI erfasst wird und somit keine Hindernislücken entstehen. Für einen hochauflösenden Velodyne HDL-32 LiDAR-Sensor mit 32 Laserdioden eignet sich z. B. eine Zellgröße von  $10\text{ cm} \times 10\text{ cm}$ .

### 4.2.2 Sensordatenreduktion

Nachdem mittels Sichtstrahltechnik eine OOI-zugehörige Hinderniszelle im 2,5D-Hindernisgitter ermittelt wurde, erfolgt die Bestimmung der acht 3D-Punkte, welche die konvexe Objekthülle beschreiben. Diese Objekthülle ermöglicht das Differenzieren zwischen OOI- und „Hintergrund“-zugehörigen Sensordaten. Demzufolge sind in dieser Arbeit Verfahren wie die *Background Subtraction* oder das *Frame Differencing* nicht erforderlich [Piccardi, 2004].

Die Objekthülle entspricht in dieser Arbeit einer Fahrzeughülle und kann folgendermaßen erzeugt werden: Ausgehend von der vom Sichtstrahl getroffenen Hinderniszelle ist die zusammenhängende Hindernisregion im 2,5D-Hindernisgitter zu erfassen. Dafür eignen sich Kontursuchverfahren wie die sogenannte *Connected-Component-Analyse* [Suzuki und Abe, 1985]. Das Verfahren verfolgt die hindernisbehafteten

Zellen, identifiziert dabei verbundene Zellen, nummeriert Konturbereiche und gliedert Konturen hierarchisch in eine Baumstruktur ein. Dabei erfolgt die Verfolgung von Hinderniszellen durch die Betrachtung der horizontalen, vertikalen und diagonalen Nachbarzellen, sodass auch komplexe Konturen erkannt werden. Das Verfahren liefert bei geschlossenen Hindernisregionen zuverlässige Ergebnisse, jedoch treten bei der Erstellung einer Hinderniskarte mit den Sensordaten eines LiDAR-Sensors oder Stereosystems viele Lücken auf. Folgende Gründe sind dafür maßgebend: Zum einen erfassen die genannten Sensoren aufgrund einer beschränkten horizontalen und vertikalen Sensorabdeckung nicht das gesamte Fahrzeug. Zum anderen verdeckt das Fahrzeug selbst Hindernisregionen. Diesbezüglich sollten diese Lücken im 2,5D-Hindernisgitter vorab geschlossen werden, sodass die gesamte Fahrzeugkontur bzw. ein größerer Konturbereich mit dem Kontursuchverfahren erfassbar wird. Zum Schließen von kleinen Hindernislöchern ist der Bildverarbeitungsoperator namens *Closing* geeignet, welcher morphologische Basisoperatoren kombiniert. Dieser besteht aus der sequenziellen Ausführung einer expandierend wirkenden Dilatation sowie einer schrumpfend wirkenden Erosion. Die Dilatation füllt durch Anwendung eines 2D-Strukturelementes alle kleinen Löcher im (binärbild-reduzierten) Hindernisgitter. Da das Strukturelement bereits bestehende Hindernisregionen verdickt bzw. ausweitet, erfolgt eine Erosion, welche das Hindernisgitter – mit Ausnahme der geschlossenen Lücken – auf die ursprüngliche Zellbelegung wiederherstellt. Diese Konkatination des morphologischen Dilatations- und Erosionsoperators ist in der Abbildung 4.4 grafisch veranschaulicht.



**Abbildung 4.4:**

Der morphologische Bildverarbeitungsoperator *Closing* besteht aus der Ausführung einer Dilatation, gefolgt von einer Erosion und dient zum Schließen von Löchern.

Dort wird ein Strukturelement in der Form einer  $3 \times 3$  Maske sukzessiv über ein Bild  $I_B$  bzw. Hindernisgitter geschoben. Beim Dilatationsoperator ist das Maskenergebnis der größte Pixel- bzw. Zellenwert der  $3 \times 3$  Maske. Dabei sind ausschließlich die Maskenwerte ungleich Null auszuwerten. In der genannten Abbildung sind alle neun Maskenwerte auf 1 gesetzt worden. Folglich werden mit dem Dilatationsoperator alle neun Pixelwerte an einer Ausgangsposition  $(u, v)$  ausgewertet:

$$I_B(u, v) = \max \left\{ \begin{array}{ccc} I_B(u-1, v-1), & I_B(u, v-1), & I_B(u+1, v-1), \\ I_B(u-1, v), & I_B(u, v), & I_B(u+1, v), \\ I_B(u-1, v+1), & I_B(u, v+1), & I_B(u+1, v+1) \end{array} \right\} \quad (4.1)$$

Im Gegensatz dazu liefert die Erosion für einen Maskenbereich den kleinsten Wert als Ergebnis. Zusammenfassend ermöglicht die sequentielle Ausführung des Dilatations-

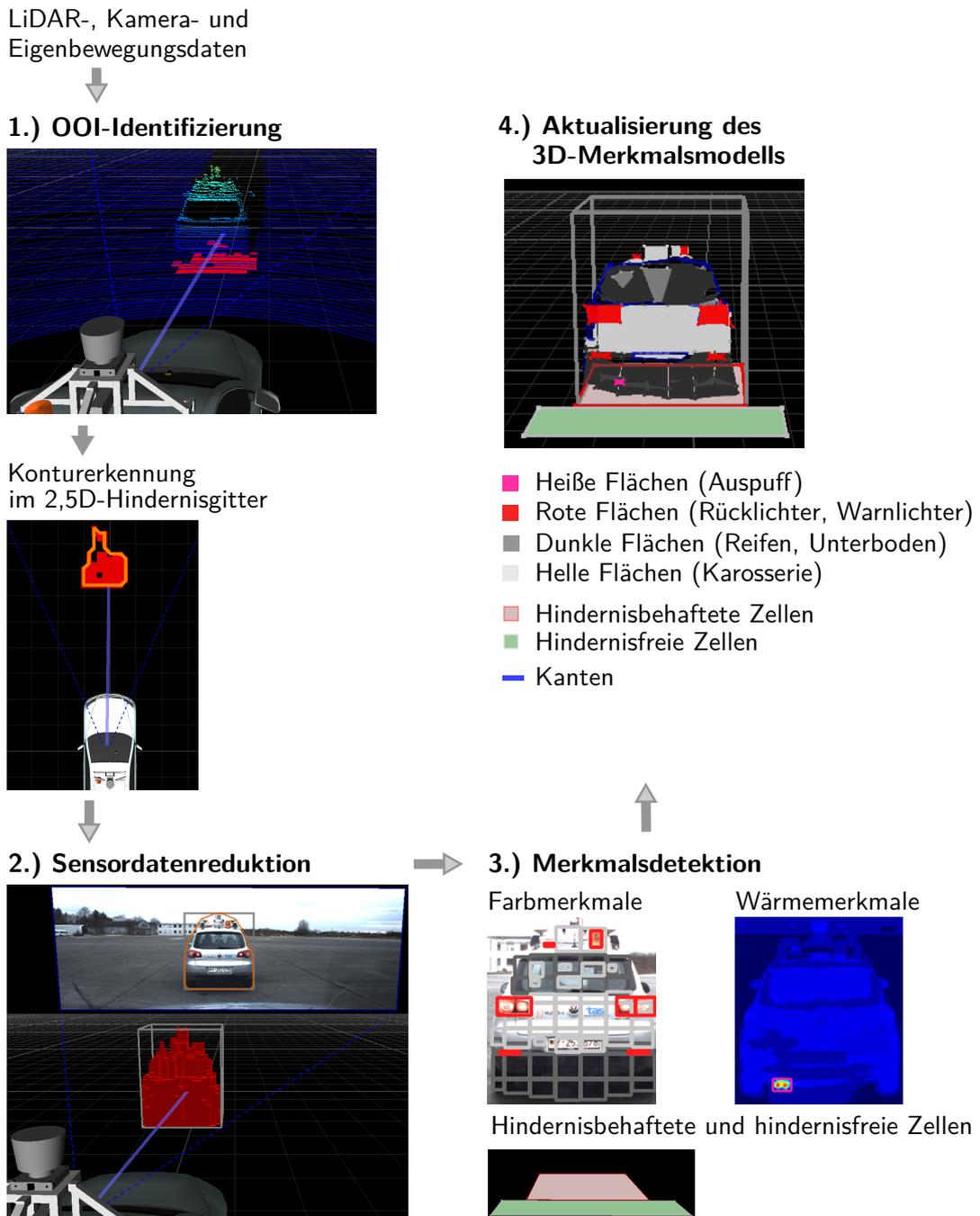
und Erosionsoperators das Füllen von Lücken im Hindernisgitter, sodass der reale Fahrzeugumriss mittels Konturerkennung besser approximiert werden kann. Diesbezüglich veranschaulicht die Abbildung 4.5 eine orange eingefärbte Konturerkennung im 2,5D-Hindernisgitter sowie deren Projektion auf die Bildebene einer Farbkamera. Bei einem 2D-Hindernisgitter kann lediglich eine 2D-Kontur der Fahrzeugbodenfläche berechnet werden. Dies ist für die spätere Suche nach visuellen Merkmalen nicht ausreichend, da in den Kameradaten eine exakte Bestimmung zwischen den Hintergrund- und Fahrzeugpixeln notwendig ist. Aus diesem Grund ist die 3D-Fahrzeugkontur auf Basis der Höheninformationen zu bestimmen.

### 4.2.3 Merkmalsdetektion

Nach der Bestimmung der OOI-zugehörigen Sensordaten und der damit durchführbaren Sensordatenreduktion folgt die Detektion von markanten Fahrzeugmerkmalen. Dazu zählen die bereits im vorherigen Kapitel vorgestellten visuellen Bildmerkmale und Hindernismerkmale: Zu den visuellen Bildmerkmalen gehören Eckpunkte, Kanten sowie Farb- und Wärmeregionen. Die Hindernismerkmale unterteilen sich in hindernisbehaftete und hindernisfreie Untergrundzellen.

Im Vergleich zum vorherigen Kapitel wird bei der automatischen Modellgenerierung zusätzlich das Kennzeichen erkannt. Das Kennzeichen ist zur Sicherstellung der Fahrzeugidentifizierung verwendbar, jedoch viel bedeutender ist die damit verbundene Möglichkeit, dem resultierenden 3D-Merkmalmodell eine eindeutige Bezeichnung zu geben. Im Nachfolgenden wird die automatische Extraktion der genannten Merkmale im Detail vorgestellt.

**Hindernisbehaftete und hindernisfreie Zellen:** Die Suche nach den belegten und freien Zellen kann sehr performant durchgeführt werden, da das grundsätzliche Wissen darüber bereits an dieser Stelle bekannt ist. In den ersten beiden Phasen der Modellgenerierung wurde das OOI identifiziert und dessen 3D-Fahrzeugkontur im 2,5D-Hindernisgitter ermittelt. Diese 3D-Fahrzeugkontur kann auf eine viereckige 2D-Untergrundkontur reduziert werden, was letztendlich den hier gesuchten hindernisbehafteten Zellen entspricht. Im Gegensatz dazu sind alle anderen Zellen des 2,5D-Hindernisgitters potentiell hindernisfrei. Dies trifft in der Praxis allerdings nicht zu, da z. B. andere Verkehrsteilnehmer sich dem OOI nähern können. Aus diesem Grund wird lediglich ein schmaler Fahrzeugrand am OOI als hindernisfrei modelliert. In der Abbildung 4.5 sind die hindernisfreien Zellen grün visualisiert, welche anhand der Sensordaten aus der fahrzeug-rückseitigen Perspektive berechnet wurden. Würde das Sensorfahrzeug neben dem OOI stehen und damit Sensordaten von einer Fahrzeugseite erfassen, so würde demzufolge ein schmaler Rand an der Fahrzeugseite als hindernisfrei modelliert werden.



**Abbildung 4.5:**

Ablaufdiagramm über die vier Phasen der automatischen Modellerzeugung: 1.) Identifizierung des OOI durch Verfolgung eines Sichtstrahls, welcher auf Hinderniszellen trifft. Die exakte Identifikation erfolgt mittels Konturanalyse im 2,5D-Hindernisgitter. 2.) Sensordatenreduktion anhand der ermittelten 3D-Fahrzeugkontur. 3.) Suche nach dominanten Fahrzeugmerkmalen durch die Auswertung der fahrzeugzugehörigen Sensordaten. 4.) Aktualisierung eines 3D-Modells mit den besonders markanten fahrzeugspezifischen Merkmalen.

**Kanten:** Für die Kantenextraktion aus einem Farbbild kann der häufig verwendete Algorithmus namens *Canny Edge Detector* [Canny, 1986] verwendet werden. Der Algorithmus führt die Suche nach Kanten in einem Grauwertbild  $I$  durch, daher muss ein Farbbild  $F$  erst in ein Grauwertbild umgewandelt werden. Die Umwandlung eines RGB-Farbbildes erfolgt gängigerweise nach Gleichung 4.2 und damit durch eine gewichtete Mittelwertbildung der einzelnen Farbkanäle.

$$I(u, v) = w_R \cdot F(u, v)_R + w_G \cdot F(u, v)_G + w_B \cdot F(u, v)_B \quad (4.2)$$

Dabei werden die einzelnen Farbkanäle unterschiedlich gewichtet, denn die wahrnehmbare Farbhelligkeit von Grün und Rot ist deutlich höher als die von Blau. Typische Gewichtungen für die roten, grünen und blauen Pixelwerte sind  $w_R = 0,299$ ,  $w_G = 0,587$  und  $w_B = 0,114$  [Burger und Burge, 2006]. Im nächsten Schritt erfolgt eine Bildglättung zur Unterdrückung von Rauschen. Dafür kann die nachfolgende Gauß-Faltungsmaske sukzessiv über das Grauwertbild geschoben werden:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Je größer die Faltungsmaske gewählt wird, desto stärker ist der Glättungseffekt. Anschließend dienen zwei weitere Faltungsmasken  $G_u$  und  $G_v$  zur separaten Extraktion der vertikalen und horizontalen Kanten:

$$G_u = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (4.3)$$

$$G_v = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (4.4)$$

Diese beiden Faltungsmasken entsprechen dem sogenannten *Sobel-Operator* [Danielsson und Seger, 1990]. Das Zusammenfügen der beiden Faltungsergebnisse erfolgt durch  $G = \sqrt{G_u^2 + G_v^2}$  oder approximiert durch  $G = |G_u| + |G_v|$  zur Steigerung der Performance. Des Weiteren ist es möglich ein Richtungsbild  $\phi = \arctan(G_v/G_u)$  zu erstellen, was in diesem Abschnitt jedoch nicht notwendig ist. Das zusammenfassende Ergebnis ist ein Kantenbild mit Informationen über die Kantenintensität.

Weiterführend wird in der Praxis häufig das Intensitätsbild auf ein Binärbild reduziert. Dieser Schritt beinhaltet eine Schwellwertbildung (engl. *Thresholding*) und ermöglicht den nachfolgenden Bildverarbeitungsalgorithmen, performanter zu arbeiten. Für die automatische Modellgenerierung wird dieser Schritt allerdings nicht empfohlen, da mit der Schwellwertbildung die Intensitätsinformationen verloren gehen, welche modellierte Kantenmerkmale eindeutiger kennzeichnen.

Der nächste Schritt ist die Extraktion aller Konturen, die sich im Intensitätsbild befinden. Dafür eignet sich die bereits auf S. 56 erwähnte *Connected-Component-*

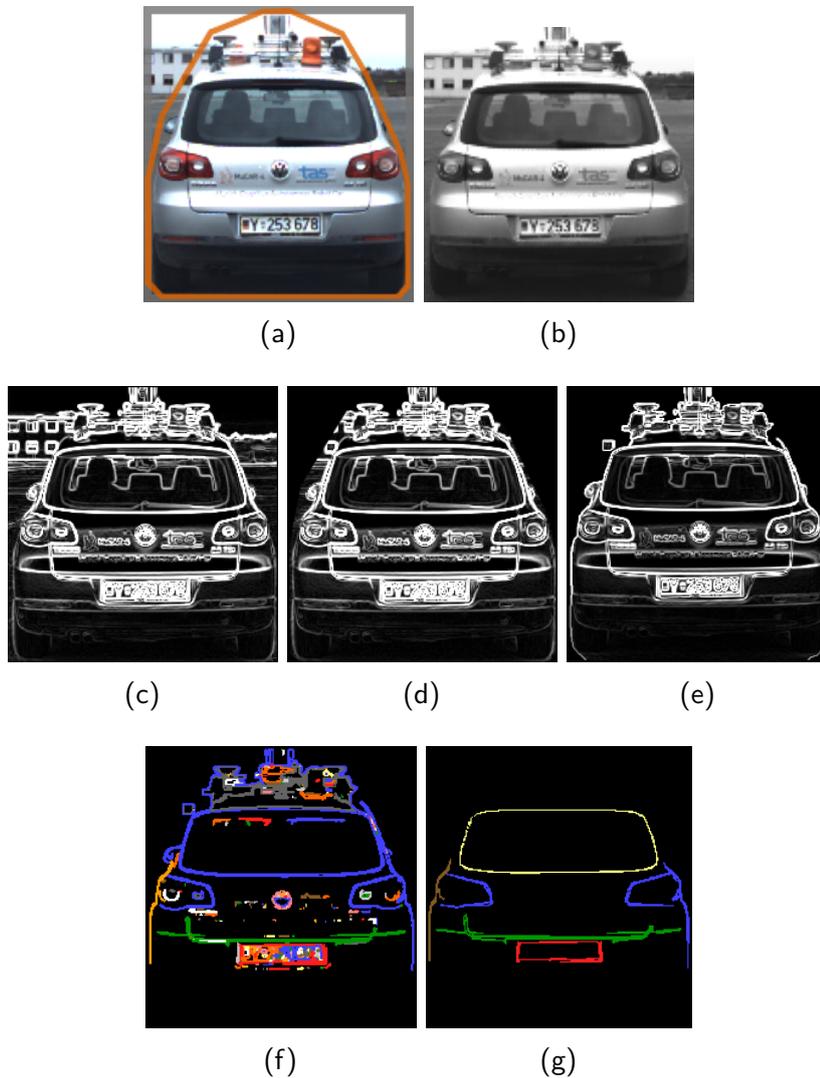
Analyse. Anschließend sind die besonders prägnanten Konturen zu suchen. Dazu zählen Konturen, die einen hohen Intensitätswert und eine ausreichende Länge aufweisen. Abbildung 4.6 veranschaulicht das bis hier beschriebene Vorgehen der Kantenextraktion.

**Eckpunkte:** Ein weiteres Bildmerkmal sind markante Eckpunkte, welche typischerweise an den sich kreuzenden Fahrzeugkanten, am Nummernschild oder am Herstellerlogo zu finden sind. Für die Extraktion von Eckpunkten gibt es unterschiedliche Vorgehensweisen. Im vorherigen Abschnitt zur Kantenextraktion wurden bereits zwei Faltungsmasken verwendet, um die horizontalen und vertikalen Kanten separat zu extrahieren. Unter anderem können diese Kanten zur Lokalisierung von Eckpunkten verwendet werden. Der Schnittpunkt von mindestens zwei Kanten führt zu potentiellen Eckpunkten. Eine weitere Möglichkeit ist die Betrachtung der Eigenwerte der Kovarianzmatrix der horizontalen und vertikalen Bildgradienten. Dafür sind zunächst die partiellen Ableitungen  $I_{\Delta u}$  sowie  $I_{\Delta v}$  und anschließend die  $2 \times 2$  Kovarianzmatrix

$$\mathbf{P} = \sum_{u,v} w(u,v) \begin{bmatrix} I_{\Delta u}^2 & I_{\Delta u}I_{\Delta v} \\ I_{\Delta u}I_{\Delta v} & I_{\Delta v}^2 \end{bmatrix} \quad (4.5)$$

für die unmittelbare Nachbarschaft  $w(\cdot)$  jedes Bildpunktes  $(u, v)$  zu berechnen. Je größer die zwei Eigenwerte der Kovarianzmatrix  $\mathbf{P}$  sind, desto höher ist die Wahrscheinlichkeit, dass an der Bildposition ein Eckpunkt vorliegt [Harris und Stephens, 1988, Bradski und Kaehler, 2008].

Der Vollständigkeit halber wird an dieser Stelle auf weitere Verfahren zur Findung von Eckpunkten hingewiesen. Vier bekannte Verfahren sind der *Wang and Brady Corner Detector* [Wang und Brady, 1995], der *Plessey-Punkt-Operator* [Harris und Stephens, 1988], *SidCell* [Schweitzer und Wuensche, 2009] und der von Rosten und Drummond [2006] entwickelte Operator namens *Features from Accelerated Segment Test (FAST)*.



**Abbildung 4.6:**

Suche nach dominanten Kanten: (a) Farbbild mit orange eingezeichneter OOI-Identifizierung, welche in den Modellgenerierungsphasen 1 und 2 durchgeführt wird. (b) Konvertierung in ein Grauwertbild. (c) Intensitätsbild mittels Canny-Algorithmus. (d) Intensitätsbild der OOI-identifizierten Pixel. (e) Optional: Schrittweise Verfeinerung der OOI-identifizierten Pixel durch *Multi-Dynamic Region Growing* (vgl. Kapitel 3.4). (f) Konturextraktion durch *Connected-Component-Analyse* mit anschließender Filterung nach Konturen mit einem hohem Intensitätswert. Die Konturseparierung ist farbig dargestellt. (g) Weitere Kantenfilterung bzgl. Kantenlänge und Linearität.

**Farb- und Wärmeflächen:** Der nächste Merkmalstyp sind Farb- und Wärmeflächen. Zu den dominanten (fahrzeugspezifischen) Farbmerkmalen gehören helle, dunkle und rote Flächen. Typische Wärmemerkmale sind kalte, warme und heiße Regionen. Die Festlegung, wie eine Fläche aussieht bzw. aus wie vielen Eckpunkten eine Fläche besteht, kann beliebig gewählt werden. Für eine schnelle Bildverarbeitung wird an dieser Stelle die Verarbeitung von viereckigen Flächen empfohlen.

Für die automatische Suche nach Farb- und Wärmeflächen evaluierte diese Arbeit zwei Vorgehensweisen. Zum einen können sämtliche Farb- und Temperaturwerte bzw. -bereiche über die Zeit ausgewertet werden. Anschließend kann mittels statistischer Auswertung bestimmt werden, welche Werte bzw. Bereiche kontinuierlich wiedererkennbar sind. Dieses Vorgehen ermöglicht ein detailliertes 3D-Merkmalmodell, benötigt allerdings viel Verarbeitungszeit, da jeder Farb- und Temperaturwert betrachtet werden muss.

Eine andere Möglichkeit ist die Suche nach einer begrenzten bzw. festgelegten Merkmalsauswahl, was die Verarbeitungszeit stark reduziert und damit eine echtzeitfähige Modellerzeugung begünstigt.

In dieser Arbeit haben sich diverse Farb- und Wärmebereiche herausgestellt, welche für die Fahrzeugerkennung als „dominant“ gelten. Auf die Detektion dieser Merkmale wird im Nachfolgenden eingegangen.

Als erstes erfolgt die Extraktion von bestimmten Farbflächen. Hierbei eignet sich der HSV- besser als der RGB-Farbraum. Die Farbräume und ihre Konvertierung werden im Anhang A.3 erläutert.

Ein weiterer Vorverarbeitungsschritt ist die Glättung des HSV-Bildes  $F_{hsv}$  zur Rauschunterdrückung. Hierbei hat sich aus Performancegründen ein Gauß-Filter bewährt. Im Anschluss der Glättung folgt die Merkmalsextraktion von Farbflächen. Da jedes zweispurige Fahrzeug nach §53 StVZO aus Gründen der Verkehrssicherheit zwei rote Rücklichter besitzen muss, empfiehlt sich die Suche nach roten Pixeln. Hierfür erfolgt die Schwellwertbildung aus Gleichung 4.6, deren Schwellwerte experimentell ermittelt wurden.

$$I_{B,rot}(u, v) = \begin{cases} 1, & \text{wenn } (F(u, v)_h \leq 40 \vee F(u, v)_h \geq 320) \wedge \\ & F(u, v)_s \geq 0.36 \wedge \\ & F(u, v)_v \geq 0.36 \\ 0, & \text{sonst} \end{cases} \quad (4.6)$$

Erfüllen die drei Pixelwerte  $[(u, v)_h, (u, v)_s, (u, v)_v]$  eines Pixels  $(u, v)$  die Bedingungen der Schwellwertbildung aus Gleichung 4.6, dann handelt es sich um einen roten Pixel, und im resultierenden Binärbild  $I_{B,rot}$  ist eine 1 einzutragen, andernfalls eine 0. Die Wertebereiche im HSV-Farbraum sind dabei folgendermaßen definiert: Die Farbtonskala  $h$  hat einen Wertebereich von 0 bis 359 und der Sättigungswert  $s$  sowie Helligkeitswert  $v$  besitzt jeweils einen Wertebereich von 0 bis 1. Die nächsten

Schritte sind die Extraktion der roten Pixelbereiche und deren Approximation durch Linien oder Vierecke. Bei der Extraktion von dominanten Kanten aus einem Kantenintensitätsbild wurde die *Connected-Component-Analyse* erwähnt, mit der sich Konturen aus einem Bild extrahieren lassen. Dieses Verfahren ist an dieser Stelle ebenfalls anzuwenden. Da ausschließlich die signifikanten Farbmerkmale relevant sind, werden zu kleine Konturen entfernt. Abschließend können die übrig gebliebenen Konturen zur Datenreduktion durch Linien oder Vierecke approximiert werden.

Ein weiteres Merkmal sind helle und dunkle Bildflächen. Deren Extraktion kann ähnlich dem Vorgehen bzgl. der roten Farbflächen erfolgen. Ein Grauwertbild  $I$  bzw. ein Interessenbereich (AOI) wird zu einem Binärbild  $I_B$  umgewandelt. In diesem kennzeichnet ein Pixelwert von 1, dass es sich hierbei um einen hellen bzw. dunklen Pixel handelt. Die gängige Berechnung eines Binärbildes erfolgt mittels Schwellwertbildung und einem Schwellwert (engl. *Threshold*). Beispielhaft zeigt die Gleichung 4.7 eine Schwellwertbildung, bei der der Pixelwert  $I(u, v)$  größer als der Schwellwert  $g_{\text{hell}}$  sein muss, damit dieser als „hell“ klassifiziert wird.

$$I_{B_{\text{hell}}}(u, v) = \begin{cases} 1, & \text{wenn } I(u, v) > g_{\text{hell}} \\ 0, & \text{sonst} \end{cases} \quad (4.7)$$

$$I_{B_{\text{dunkel}}}(u, v) = \begin{cases} 1, & \text{wenn } I(u, v) < g_{\text{dunkel}} \\ 0, & \text{sonst} \end{cases} \quad (4.8)$$

Dabei ist die Wahl des Schwellwertes für eine „korrekte“ Klassifikation entscheidend. Ein gängiges Vorgehen ist das einmalige Festlegen eines empirisch ermittelten Schwellwertes. Allerdings führt dieses Vorgehen bei unterschiedlichen Wetter- und Belichtungsbedingungen zu einer fehlerhaften Klassifikation der Merkmale. Aus diesem Grund sollte der Schwellwert adaptiv ermittelt werden. Eine Möglichkeit ist die Verwendung des arithmetischen Mittelwertes  $\mu$  und der Standardabweichung  $\sigma$  (unkorrigierte Stichprobenvarianz) der  $n$  Pixelwerte eines Grauwertbildes.  $\mu$  und  $\sigma$  entsprechen dem mittleren Intensitätswert und deren Streubereich und verändern sich damit bei wechselhaften Wetter- und Belichtungsbedingungen. Die Berechnung erfolgt nach den Gleichungen 4.9 und 4.10:

$$\mu = \frac{1}{n} \cdot \sum_{i=1}^n I(u, v)_i \quad (4.9)$$

$$\sigma = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (I(u, v)_i - \mu)^2} \quad (4.10)$$

Im Nachfolgenden beschreiben die Gleichungen 4.11 und 4.12 deren Verwendung zur Bestimmung eines adaptiven Schwellwertes für die Filterung nach hellen und dunklen Pixelwerten. Dabei definiert  $\alpha$  einen Gewichtungsfaktor der Standardabweichung, welcher je nach Anwendungsbereich zu definieren ist:

$$g_{\text{hell}} = \mu + \alpha \cdot \sigma \quad (4.11)$$

$$g_{\text{dunkel}} = \mu - \alpha \cdot \sigma \quad (4.12)$$

Temperaturmerkmale bzw. Wärmeflächen werden analog zu den hellen und dunklen Flächen verarbeitet. Zusätzlich kann an dieser Stelle Modellwissen verwendet werden. Beispielsweise heizt sich der Auspuff eines fahrenden Fahrzeuges sehr stark auf und am Auspuffende entstehen hohe Oberflächentemperaturen. In den Praxisversuchen dieser Arbeit lag die Oberflächentemperatur des Auspuffendes im Bereich von 25 bis 60 °C. Mit diesem Wissen konnten die adaptiv berechneten Schwellwerte auf Gültigkeit validiert werden.

**Kennzeichen:** Ein Kfz-Kennzeichen ist ein Texturmerkmal und befindet sich typischerweise an der Fahrzeugfront zwischen den beiden Frontscheinwerfern sowie am Heck zwischen den zwei roten Rücklichtern. Es ist ein markantes Merkmal und kennzeichnet ein Kraftfahrzeug eindeutig. Damit nicht ein komplettes Kamerabild auf ein Kennzeichen untersucht werden muss, ist der visuelle Suchraum auf den Bereich zwischen den Rück- und Fahrtlichtern zu begrenzen. Anschließend kann mit einer adaptiven Schwellwertbildung der Bildausschnitt eines Grauwertbildes in ein Binärbild konvertiert und mit der *Connected-Component*-Analyse nach kleinen, gleich großen sowie nebeneinanderstehenden Konturen gesucht werden. In dem Fall, dass keine oder nur wenige Konturen gefunden werden, erfolgt die Schwellwertbildung mit einem anderen Schwellwert erneut. Hierbei wird der Gewichtungsfaktor  $\alpha$  der Standardabweichung sowohl größer als auch kleiner gemacht.

Nachdem ausreichend viele gültige Konturen erkannt wurden, sind die Konturen mit der Zeichenerkennungssoftware *Tesseract* [Smith, 2007] analysierbar. Bei einer erfolgreichen Kennzeichenerkennung sollte die Modellgenerierung die Kennzeichen-grundfarbe für die fortlaufenden Bildverarbeitungsalgorithmen zwischenspeichern. Bei deutschen Verkehrsfahrzeugen ist dieser Farbwert als Referenzwert für die Farbe „weiß“ verwendbar, sodass Bildverarbeitungsalgorithmen auch bei wechselhaften Lichtverhältnissen einwandfrei funktionieren können. Zusätzlich wird aus Performancegründen empfohlen, die Kennzeichenerkennung für alle weiteren Kamerabilder zu deaktivieren, nachdem ausreichend viele Zeichen erkannt wurden.

#### 4.2.4 Aktualisierung des 3D-Merkmalmodells

Die vierte und letzte Phase ist die Integration der ermittelten Merkmale in ein gemeinsames 3D-Merkmalmodell. Dafür müssen für jedes Merkmal die 3D-Koordinaten in einem objektfesten Koordinatensystem bestimmt werden. Dieses ermöglicht eine Korrespondenzsuche zwischen bereits integrierten und zu integrierenden Modellmerkmalen. Dadurch kann das mehrfache Hinzufügen von gleichen Merkmalen vermieden und eine Wiedererkennungsrates berechnet werden. Weiterführend ermöglicht die zeitliche Beobachtung das Aussortieren von Merkmalen, welche nicht in mehreren

Perspektiven zuverlässig detektierbar sind. Nachstehend wird die Aktualisierung eines fahrzeugspezifischen 3D-Merkmalmodells im Detail beschrieben.

Der erste Schritt ist die einmalige Festlegung eines Referenzkoordinatensystems  $S_{\text{ref}}$ , auf das sich die 3D-Koordinaten des 3D-Merkmalmodells beziehen sollen. Erst dann sind die aus unterschiedlichen Perspektiven stammenden Merkmale in ein gemeinsames Modell integrierbar und miteinander vergleichbar. Als Referenzkoordinatensystem eignet sich beispielsweise die globale Pose des Egofahrzeuges für den Zeitpunkt der ersten OOI-Identifizierung. Für die Schätzung der globalen Fahrzeugpose wird ein hochwertiges INS-Gerät sowie ein erweiterter Kalman-Filter empfohlen. Ebenfalls sollte bei der Positionsschätzung berücksichtigt werden, dass jede Fahrzeugbewegung sprungfrei erfolgt.

Nach der Festlegung eines Referenzkoordinatensystems sind die ego-relativen 3D-Koordinaten für jedes Bildmerkmal zu berechnen. Da ein visuelles Merkmal durch eine Anzahl von 2D-Bildkoordinaten beschrieben ist, empfiehlt sich die Berechnung der 3D-Weltkoordinate jeder 2D-Bildkoordinate. Dazu ist die in der Modellgenerierungsphase 1 beschriebene Sichtstrahlverfolgung geeignet (siehe Abschnitt 4.2.1). Dort wird ein 2D-Bildpunkt anhand der intrinsischen und extrinsischen Kamerakalibrierung in die 3D-Welt hinausprojiziert und dessen Auftreffen auf ein Hindernis im 2,5D-Hindernisgitter erkannt. Die ermittelte Hinderniszelle – inkl. dem Wissen über die Höhe des Sichtstrahls an dieser Zellposition – führt zu der ego-relativen homogenen 3D-Koordinate  $\mathbf{p}_{\text{ego}} = [x, y, z, 1]^T$ . Diese beschreibt die Translation im Koordinatensystem des Sensorfahrzeuges  $S_{\text{ego}}$  vom Koordinatenursprung zum Merkmal.

Der nächste Schritt ist die Berechnung der 3D-Koordinate im Referenzkoordinatensystem  $S_{\text{ref}}$  des 3D-Merkmalmodells. Hierfür kann eine homogene  $4 \times 4$  Transformationsmatrix

$$\text{nach } \mathbf{H}_{\text{von}} = \left[ \begin{array}{c|c} \text{Rotationsmatrix} & \text{Translationsvektor} \\ \hline \mathbf{R}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ \hline \mathbf{0}_{1 \times 3} & 1 \end{array} \right] = \left[ \begin{array}{ccc|c} R_{11} & \dots & R_{13} & -x \\ \vdots & \ddots & \vdots & -y \\ R_{31} & \dots & R_{33} & -z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (4.13)$$

verwendet werden, welche die Abbildung zwischen zwei Koordinatensystemen beschreibt. Im Bezug auf die räumliche Bewegung des Sensorfahrzeuges definiert  ${}^{\text{ref}}\mathbf{H}_{\text{ego}_k}$  die Abbildung vom Ego- zum Referenzkoordinatensystem für den diskreten Zeitpunkt  $k$ . Weiterführend kann jede zum Sensorfahrzeug relative 3D-Koordinate in das Referenzkoordinatensystem abgebildet werden:

$$\mathbf{p}_{\text{ref}} = {}^{\text{ref}}\mathbf{H}_{\text{ego}_k} \cdot \mathbf{p}_{\text{ego}_k} \quad (4.14)$$

Der Vollständigkeit halber zeigt Gleichung 4.15 die Verwendung der bekannten homogenen Transformationsmatrix  ${}^{\text{ego}}\mathbf{H}_{\text{Kamera}}$  zur Abbildung von kamera-relativen 3D-Koordinaten in das Fahrzeugkoordinatensystem  $\mathbf{S}_{\text{ego}}$ .

$$\mathbf{p}_{\text{ego}} = {}^{\text{ego}}\mathbf{H}_{\text{Kamera}} \cdot \mathbf{p}_{\text{Kamera}} \quad (4.15)$$

Nachdem die 3D-Position jedes Merkmals im ortsfesten Referenzkoordinatensystem  $\mathbf{S}_{\text{ref}}$  bestimmt wurde, ist ein Vergleich von Merkmalen (engl. *Feature Matching*) durchführbar. Ein Merkmalsvergleich ermöglicht, dass ein Fahrzeugmerkmal nicht mehrfach zum Modell hinzugefügt wird, und dient der Statistikerstellung bzgl. der Wiedererkennbarkeit eines Merkmals. Ein recheneffizienter und damit echtzeitfähiger Vergleich von Merkmalen kann anhand des Typs (Kante, Farbfläche etc.) und der auf einen Referenzpunkt bezogenen 3D-Positionsdaten erfolgen. Hierfür wird abhängig vom Merkmalstyp überprüft, ob 3D-Merkmalskoordinaten der aktuellen Sensoraufnahme (Zeitpunkt  $k$ ) mit 3D-Merkmalskoordinaten der vorherigen Sensoraufnahme (Zeitpunkt  $k - 1$ ) korrespondieren.

Unter anderem eignet sich der euklidische Abstand  $d$  beim Vergleich zweier 3D-Koordinaten  $(x, y, z)_i$  und  $(x, y, z)_j$ :

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (4.16)$$

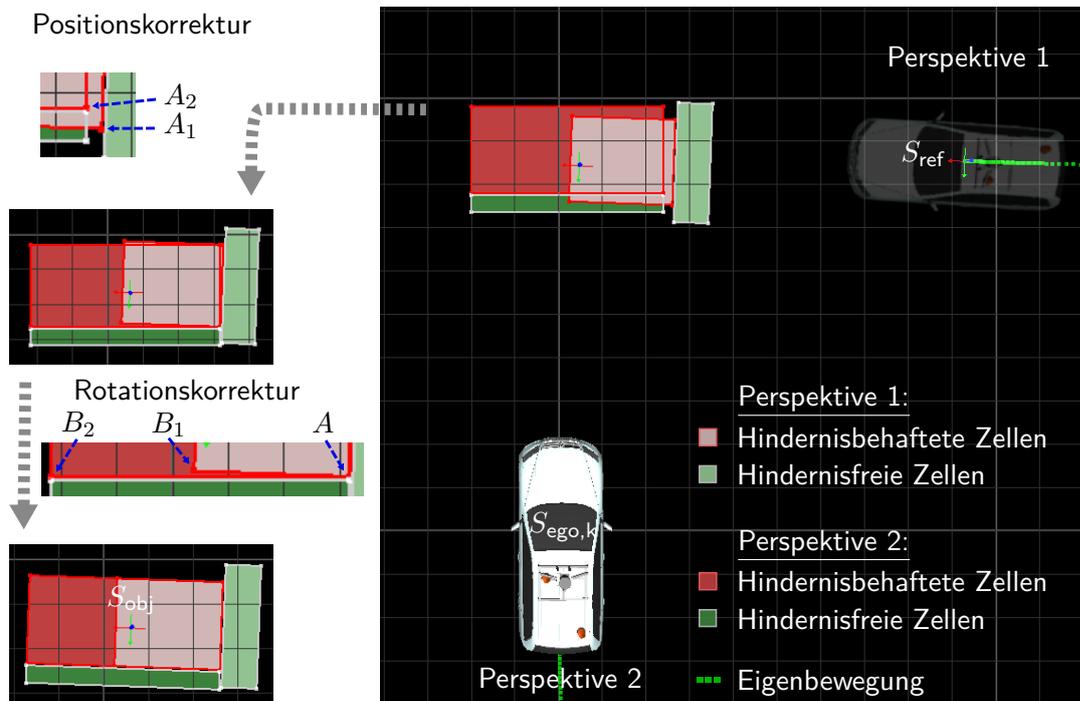
Unterschreitet dieses Abstandsmaß einen festgelegten Schwellwert, dann handelt es sich um ein wiedererkanntes Merkmal. Dabei hängt die Festlegung des Schwellwertes zum einen von der Genauigkeit der Eigenbewegungsschätzung und zum anderen von der Genauigkeit der berechneten 3D-Merkmalskoordinaten ab. In dieser Arbeit wird ein Schwellwert von sechs Zentimetern verwendet und entspricht damit dem dreifachen Fehler der verwendeten Eigenbewegungssensorik (siehe Anhang A.1.3).

Im Bezug auf die viereckigen Farb- und Wärmeflächen sind die jeweiligen 3D-Koordinaten der vier Eckpunkte für den Merkmalsvergleich zu verwenden. Bei der Korrespondenzsuche von Kanten können die Kantenstützstellen verglichen werden, und der Mittelwert gibt anschließend über die Korrespondenzwahrscheinlichkeit Auskunft. Allerdings ist die Anzahl der Kantenpunkte je nach perspektivischer Betrachtung unterschiedlich. Zum einen erhöht Bildrauschen die Anzahl der Richtungsänderungen einer Kante, und zum anderen sinkt die Anzahl der Kantenpunkte mit der Entfernung. Aus diesem Grund, und aus Gründen der späteren Verarbeitungseffizienz bei der modellbasierten Fahrzeugerkennung, sollte jede Kante normalisiert werden. Für die Punktreduzierung bzw. Linienglättung eignet sich der iterative *Ramer–Douglas–Peucker* (RDP)-Algorithmus [Ramer, 1972, Douglas und Peucker, 1973], mit dem sich die Anzahl der Kantenpunkte in Abhängigkeit von der Kantenlänge verringern bzw. vergrößern lässt.

Zusammenfassend ist die 3D-Merkmalposition zu ermitteln, ein Merkmalsabgleich mit dem derzeitigen 3D-Merkmalmodell durchzuführen, neue Merkmale sind zum

Modell hinzuzufügen und die Wiedererkennungsrate von wiedererkannten Merkmalen ist zu aktualisieren. Falls die Schätzung der Eigenbewegung zu ungenau ist, muss diese merkmalsbasiert korrigiert werden, ansonsten ist eine zuverlässige Korrespondenzfindung nicht gewährleistet. Hierfür empfiehlt sich die in jeder Perspektive erkannte viereckige sowie hindernisbehaftete Untergrundfläche des OOI: Die Eigenbewegungsschätzung des Sensorfahrzeuges liefert ein grobes Wissen über den Betrachtungswinkel der hindernisbehafteten Untergrundflächen relativ zu verschiedenen Aufzeichnungsperspektiven. Demzufolge kann das Sensorfahrzeug hinter dem OOI stehen, die hindernisbehafteten Untergrundflächen ermitteln, anschließend zur linken Fahrzeugseite fahren, erneut die hindernisbehafteten Untergrundflächen ermitteln und abschließend beide Flächen miteinander abgleichen und damit den Eigenbewegungsfehler korrigieren.

Abbildung 4.7 zeigt ein Praxisbeispiel, bei dem die fehlerbehaftete inverse Eigenbewegung für eine Merkmalstransformation zwischen zwei Perspektiven verwendet wird und anschließend ein großer Abbildungsfehler vorliegt.



**Abbildung 4.7:**

Merkmalstransformation in ein globales Referenzkoordinatensystem  $S_{ref}$ : Transformation der Merkmale aus der zweiten Perspektive in ein globales Referenzkoordinatensystem der ersten Perspektive. Die inverse Eigenbewegung bildet Merkmalskoordinaten der zweiten Perspektive auf das Referenzkoordinatensystem ab. Eine fehlerbehaftete Eigenbewegungsschätzung impliziert einen Abbildungsfehler, welcher mittels Merkmalskorrespondenzsuche zwischen den partiell-überlappenden hindernisbehafteten Flächen behoben werden kann.

Dieser Abbildungsfehler kann mit Merkmalskorrespondenzen korrigiert werden. Die aus jeweils einer Perspektive stammenden vier Untergrundeckpunkte  $\{A, B, C, D\}$  der hindernisbehafteten Zellen sind hierfür besonders gut geeignet. Diese bilden ein Viereck, sind eben und überlappen sich partiell. Demzufolge ist der Abbildungsfehler performant berechenbar: Als erstes erfolgt die positionelle Korrektur durch die Berechnung der 3D-Verschiebung  $\Delta(x, y, z)$  zweier 3D-Punkte. Dafür empfiehlt sich ein Eckpunkt einer hindernisbehafteten Zelle, der aus beiden Perspektiven erfasst wurde. In diesem Fall ist die Ecke  $A_1$  der Perspektive 1 und  $A_2$  der Perspektive 2 für die Positionskorrektur optimal:

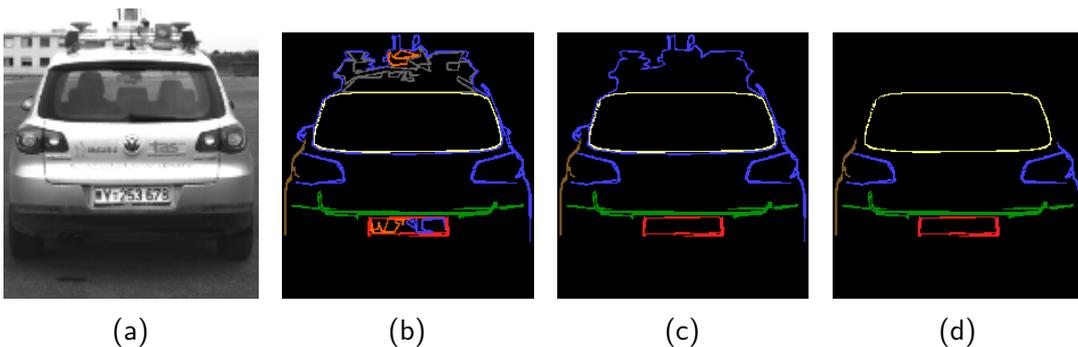
$$\Delta(x, y, z) = A_1 - A_2 \quad (4.17)$$

Anschließend ist der Drehfehler zu korrigieren, indem z. B. der Winkel zwischen den Eckpunkten  $A, B_1$  und  $B_2$  bzw. den zwei Vektoren  $(B_1 - A)$  und  $(B_2 - A)$  berechnet wird:

$$\angle(A, B_1, B_2) = \arccos\left(\frac{(B_1 - A) \cdot (B_2 - A)}{\sqrt{|(B_1 - A)| \cdot |(B_2 - A)|}}\right) \quad (4.18)$$

Im weiteren Ablauf fährt das Sensorfahrzeug diverse Perspektiven ab, sodass das 3D-Merkmalmodell schrittweise erweitert werden kann. Nachdem das OOI einmal von allen vier Seiten erfasst wurde, erfolgt eine abschließende Merkmalsfilterung anhand der jeweiligen Wiedererkennungsrate. Dabei werden alle visuellen Merkmale aus dem 3D-Merkmalmodell entfernt, bei denen die Wiedererkennungsrate einen Schwellwert unterschreitet.

Abbildung 4.8 veranschaulicht diese Filterung bezüglich der zeitlichen Beobachtbarkeit von Kantenmerkmalen.



**Abbildung 4.8:**

Suche nach dominanten Kanten mittels statistischer Auswertung einer zeitlichen Merkmalsbeobachtung: (a) Grauwertbild. (b) Extraktion der Konturen aus dem Grauwertbild. Die Kontursegmentierung ist farbig dargestellt. (c-d) Zeitliche Merkmalsbeobachtung: Markante Kanten, die über einen kurzen Zeitraum (c) und weiterführend aus unterschiedlichen Perspektiven (d) wiedergefunden wurden.

Im letzten Schritt der automatischen Modellgenerierung erfolgt das lokale Abspeichern des 3D-Merkmalmodells, sodass die modellbasierte Fahrzeugerkennung aus dem Kapitel 5 diese verwenden kann.

Derzeit befindet sich der Ursprung des Koordinatensystems beim Referenzkoordinatensystem  $S_{\text{ref}}$  der ersten Perspektive. Dadurch beziehen sich die 3D-Positionsangaben der Merkmale relativ zu diesem Koordinatensystem  $S_{\text{ref}}$ . Allerdings ist als Koordinatenbezugspunkt der Mittelpunkt des 3D-Merkmalmodells besser geeignet. Für die Transformation des Koordinatenursprunges in den Mittelpunkt des 3D-Merkmalmodells muss der Mittelpunkt  $S_{\text{obj}}$  bestimmt werden. Hierfür sind die minimalen und maximalen Positionswerte aller hindernisbehafteten 2,5D-Zellen zu berechnen, sodass zuerst die  $S_{\text{ref}}$ -relative konvexe Fahrzeughülle und deren Mittelpunkt sowie anschließend die homogene Abbildungsmatrix  ${}^{\text{obj}}\mathbf{H}_{\text{ref}}$  bestimmbar ist. Anschließend können sämtliche Positionswerte im 3D-Merkmalmodell mit der berechneten Transformationsmatrix  ${}^{\text{obj}}\mathbf{H}_{\text{ref}}$  in das neue Koordinatensystem  $S_{\text{obj}}$  abgebildet werden:

$$\mathbf{p}_{\text{obj}} = {}^{\text{obj}}\mathbf{H}_{\text{ref}} \cdot \mathbf{p}_{\text{ref}} \quad (4.19)$$

## 5 Fahrzeugerkennung und -verfolgung

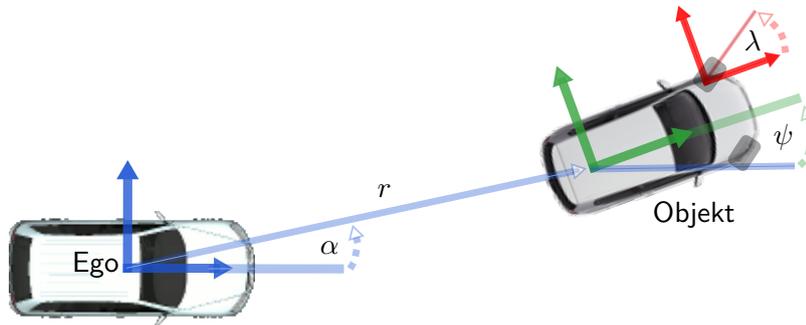
### Inhalt

---

5.1	Objekterkennung durch rekursive Zustandsschätzung . . . . .	72
5.1.1	Filterinitialisierung . . . . .	73
5.1.2	Visuelle Objekterkennung mittels Kalman-Filter . . . . .	74
5.1.3	Objekterkennung mittels Partikel-Filter . . . . .	78
5.1.4	Kombinierte Objekterkennung mit Partikel- und Kalman-Filter . . . . .	98
5.2	Aufmerksamkeitssteuerung . . . . .	100
5.2.1	Aktive Blickrichtungssteuerung . . . . .	100
5.2.2	Hardwarebasierte AOI einer Kamera . . . . .	100
5.2.3	Softwarebasierte AOI einer Kamera . . . . .	102
5.3	Berechnung eines Qualitätsmaßes . . . . .	105
5.3.1	Auswertung der Filter-Kovarianzmatrix . . . . .	105
5.3.2	Merkmalsauswertung mit definierten Schwellwerten . . . . .	106
5.3.3	Trainieren eines Klassifikators . . . . .	107
5.4	Fahrzeugführung . . . . .	112
5.4.1	Spurgenerierung . . . . .	112
5.4.2	Quer- und Längsregelung . . . . .	114

---

Der Schwerpunkt dieser Arbeit liegt in der Erkennung und zeitlichen Verfolgung eines ausgewählten Fahrzeuges im Konvoi. In diesem Kapitel werden drei Verfahren vorgestellt, die allesamt Modellwissen über jedes wahrzunehmende Fahrzeug voraussetzen und eine rekursive Zustandsschätzung durchführen. Das Modellwissen beinhaltet zum einen das Aussehen des Fahrzeuges inkl. der Angabe der besonders markanten Fahrzeugmerkmale, als auch das Wissen über die Fahrzeuggeometrie. Bei allen drei Verfahren liegt der Fokus in einer robusten und kontinuierlichen Fahrzeugerkennung mit einer möglichst genauen ego-relativen Positionsschätzung (siehe Abbildung 5.1), sodass einem Führungsfahrzeug spurtreu gefolgt werden kann. Das erste Verfahren setzt weniger Modellwissen voraus und verwendet kaskadierte Klassifikatoren zur rein visuellen Fahrzeugerkennung in Kamerabildern. Für die Positionsschätzung des Führungsfahrzeuges kommt ein Kalman-Filter zum Einsatz, der mit dem Wissen über die Fahrzeuggeometrie und den zu detektierenden Fahrzeugseiten eine Zustandsschätzung über die Zeit durchführt. Anstatt wie beim ersten Verfahren im



**Abbildung 5.1:**

Relative Fahrzeugpose in zylindrischen Koordinaten.  ${}^{\text{obj}}\mathbf{P}_{\text{ego}} = \{r, \alpha, h, \phi, \theta, \psi\}$  kennzeichnet die relative Lage [Abstand  $r$ , Winkel  $\alpha$ , Höhe  $h$ ] und Orientierung [Rollwinkel  $\phi$ , Nickwinkel  $\theta$ , Gierwinkel  $\psi$ ] und  $\lambda$  veranschaulicht den Lenkwinkel des Führungsfahrzeuges.

2D-Raum zu messen und im 3D-Raum zu schätzen, erfolgt im zweiten Verfahren mittels detailliertem 3D-Merkmalmodell und Partikel-Filter die direkte Messung im 3D-Raum. Das dritte Verfahren ist ein kombinierter Ansatz mit einem Partikel- und einem Kalman-Filter.

Im Nachfolgenden wird das Grundprinzip von rekursiven Zustandsschätzern kurz erläutert, auf deren Initialisierung eingegangen und anschließend die drei Verfahren zur Fahrzeugerkennung und -verfolgung vorgestellt. Für eine detaillierte Beschreibung über die rekursive Zustandsschätzung mit einem Kalman- oder Partikel-Filter sei auf das Grundlagenkapitel 3.2 *Rekursive Zustandsschätzung dynamischer Systeme* verwiesen.

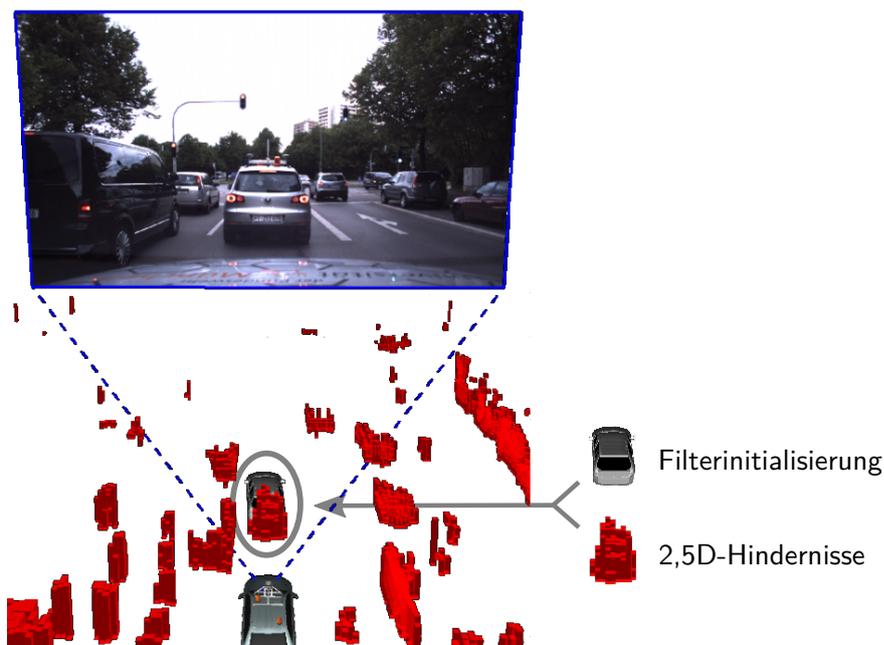
## 5.1 Objekterkennung durch rekursive Zustandsschätzung

Die Schätzung  $\hat{x}$  von beobachtbaren Zuständen  $x$  eines dynamischen Systems ist durch einen kontinuierlichen Vergleich zwischen einer Messung  $y$  und deren Vorhersage  $y^*$  möglich. Sie erfolgt unter Berücksichtigung sämtlicher Messungen  $\mathbf{Y}_k = \{y_0, \dots, y_k\}$  für einen diskreten Zeitpunkt  $k$ .

Das Kalman- und Partikel-Filter sind rekursive Bayessche Zustandsschätzer, welche die Schätzung durch eine Wahrscheinlichkeitsdichtefunktion  $p(\cdot)$  beschreiben. Die Filter besitzen zwei Phasen, die Prädiktion und Update genannt werden. Im Prädiktionsschritt werden die Zustandswerte vorhergesagt und im Update-Schritt erfolgt die Auswertung von Messergebnissen sowie die Korrektur der prädizierten Schätzung. Bevor das erste Mal eine Prädiktion erfolgt, sollten die Zustandswerte initialisiert werden. Nachfolgend werden drei Möglichkeiten vorgestellt.

### 5.1.1 Filterinitialisierung

Für die Initialisierung der Zustandswerte gibt es verschiedene Möglichkeiten. Im Idealfall besteht eine Funkverbindung im Konvoi, über die globale Fahrzeugpositionen sowie Geschwindigkeiten übermittelt werden können. Allerdings sind GNSS-Informationen immer mit Vorsicht zu verwenden, da diese in vielen Bereichen – wie bspw. im Wald oder bei hohen und nahestehenden Gebäuden – sehr ungenau sein können. Eine Alternative ist die Verwendung von bereits vorliegenden Schätzergebnissen von parallel laufenden Erkennungsalgorithmen [Fries et al., 2013, Fries und Wuensche, 2014]. Hierbei können redundante bzw. gekoppelte Systeme zum einen die Robustheit und zum anderen die Ausfallsicherheit steigern. In dem Fall, dass kein kommuniziertes oder berechnetes Vorwissen über die Fahrzeugposition erfolgt, sollten die Zustände mit möglichst praxisrelevanten Werten belegt werden. Zu Beginn des Konvoi-Szenarios befindet sich typischerweise das lokale Führungsfahrzeug direkt vor dem eigenem Fahrzeug und ist mit einer ähnlichen Fahrzeugorientierung ausgerichtet. Aus diesem Grund werden in dieser Arbeit die Zustände folgendermaßen initialisiert. Der zylindrische Abstand  $r$  der ego-relativen Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}} = \{r, \alpha, h, \phi, \theta, \psi\}$  wird bei 10 Metern vermutet, wodurch  ${}^{\text{obj}}\mathbf{P}_{\text{ego}}$  mit  $\{10, 0, 0, 0, 0, 0\}$  zu initialisieren ist. Des Weiteren wird die Geschwindigkeit  $v$  mit der eigenen Geschwindigkeit gleichgesetzt und der Lenkwinkel  $\lambda$  des Führungsfahrzeuges sollte auf 0 gesetzt werden. Eine solche Initialisierung zeigt die Abbildung 5.2 beim Einschalten des autonomen Folgemodus an einer Kreuzung im Straßenverkehr.



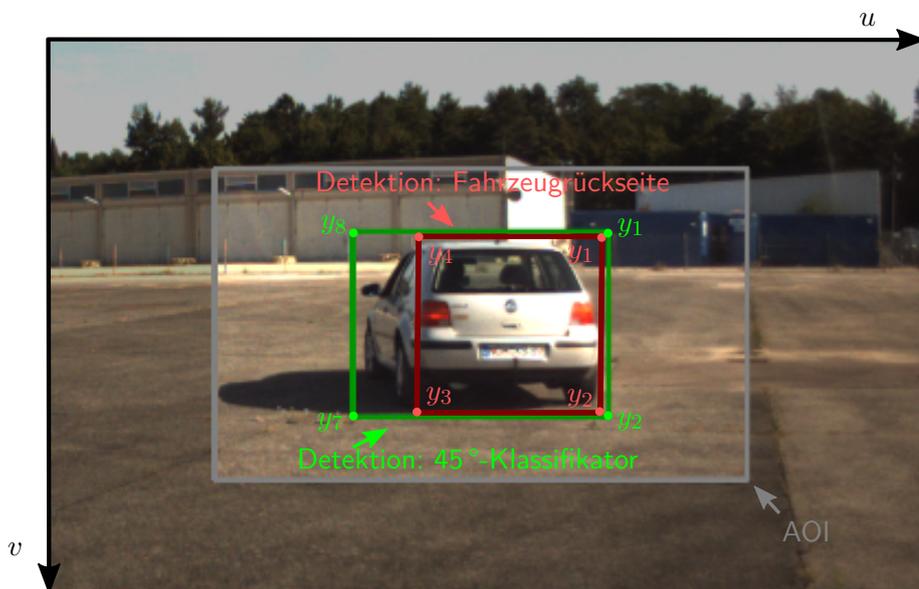
**Abbildung 5.2:**

Filterinitialisierung: Das lokale Führungsfahrzeug wird vor dem eigenen Fahrzeug in circa zehn Metern Entfernung erwartet.

Beim Partikel-Filter müssen die einzelnen Partikel unterschiedlich initialisiert werden. Hier empfiehlt sich das Verrauschen der Zustände durch Hinzufügen von Gaußschem Rauschen in einem vom Benutzer definierten Rauschbereich.

### 5.1.2 Visuelle Objekterkennung mittels Kalman-Filter

Dieses Kapitel stellt ein echtzeitfähiges Verfahren zur rein visuellen Erkennung und Verfolgung eines ausgewählten Führungsfahrzeuges sowie zur Positions- und Geschwindigkeitsschätzung vor. Der Algorithmus verwendet bekanntes Modellwissen, jedoch kein detailliertes 3D-Merkmalsmodell, detektiert Fahrzeugseiten mit kaskadierten Klassifikatoren und schätzt über die Zeit mit einem Kalman-Filter die Fahrzeugposition und dessen Geschwindigkeit. Im Gegensatz zu vielen Publikationen wird für die Positionsschätzung kein hochgradig nichtlineares und zeitaufwendiges *Perspective-n-Point*-Minimierungsproblem gelöst [Hartley und Zisserman, 2004, Lepetit et al., 2009]. Stattdessen schätzt ein Kalman-Filter die ego-relative Fahrzeugpose über die Zeit. Dafür müssen in jedem Kamerabild die sichtbaren Fahrzeugseiten erkannt und dessen Eckpunkte ermittelt werden. Die Detektion der Fahrzeugseiten erfolgt auf Basis kaskadierter Klassifikatoren (engl. *Cascaded Classifiers*), auf die ausführlich im Grundlagenkapitel 3.3 eingegangen wurde. Die Abbildung 5.3 veranschaulicht das Ergebnis von zwei unterschiedlich trainierten Klassifikatoren. Hierbei wurde ein Klassifikator für die Detektion von Fahrzeugrückseiten trainiert und ein weiterer Klassifikator dient der Erkennung von Fahrzeugen mit einem Gierwinkel von circa 45 Grad.



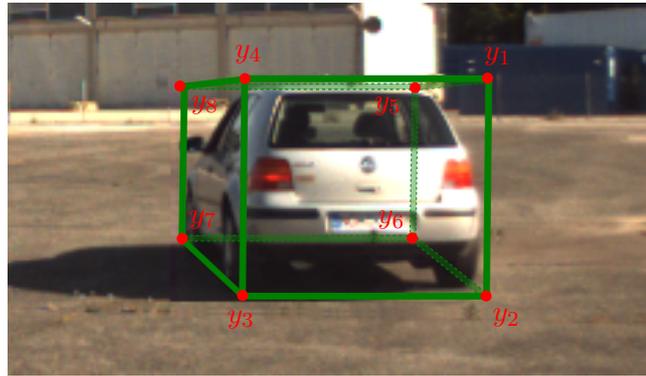
**Abbildung 5.3:**

Klassifikatoren zur Detektion von Fahrzeugseiten. Das graue, rote und grüne Rechteck visualisieren einen reduzierten Suchbereich (AOI), die Detektion der Fahrzeugrückseite sowie das Ergebnis eines 45°-Klassifikators.

Im Nachfolgenden wird der Prädiktions- und Innovationsschritt eines Scaled Unscented Kalman-Filters erläutert, welcher die Positions- und Geschwindigkeitsschätzung des Führungsfahrzeuges über die Zeit durchführt. Die Grundlagen über diesen Filter wurde im Kapitel 3.2.1.4 beschrieben.

### 5.1.2.1 Prädiktion

Im Prädiktionschritt erfolgt die Vorhersage der Zustandswerte  $\boldsymbol{x}^*$  und Messwerte  $\boldsymbol{y}^*$  für einen diskreten Zeitpunkt  $k$ . Die Zustandswerte  $\boldsymbol{x}$  bestehen aus der ego-relativen Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}} = \{r, \alpha, h, \phi, \theta, \psi\}$  in Zylinderkoordinaten sowie der Geschwindigkeit  $v$  in Längsrichtung und dem Lenkwinkel  $\lambda$  vom Führungsfahrzeug. Die Messwerte  $\boldsymbol{y}$  setzen sich aus den acht Eckpunkten zusammen, welche die Fahrzeugseiten bzw. die Fahrzeughülle in der Form eines Quaders im Kamerabild beschreiben (vgl. Abbildung 5.4).



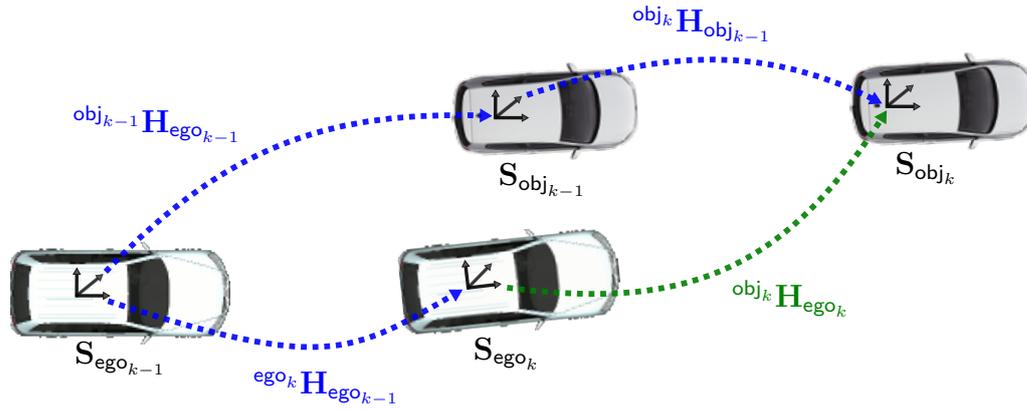
**Abbildung 5.4:**

Fahrzeug umschließendes Quader beschrieben durch 8 Eckpunkte  $\boldsymbol{y}_{1,\dots,8}$ .

Für eine möglichst realistische Vorhersage der Zustandswerte  $\boldsymbol{x}$  wird das aktuelle Schätzergebnis  ${}^{\text{obj}}_{k-1}\mathbf{P}_{\text{ego}_{k-1}}$ , die berechnete Eigenbewegung  ${}^{\text{ego}}_k\mathbf{H}_{\text{ego}_{k-1}}$  und die prädizierte Bewegung des Führungsfahrzeuges  ${}^{\text{obj}}_k\mathbf{H}_{\text{obj}_{k-1}}$  berücksichtigt. Die Abbildung 5.5 veranschaulicht die involvierten Transformationsmatrizen.

Als erstes muss das aktuelle Schätzergebnis  ${}^{\text{obj}}_{k-1}\mathbf{P}_{\text{ego}_{k-1}}$  in eine homogene  $4 \times 4$  Transformationsmatrix

$${}^{\text{obj}}_{k-1}\mathbf{H}_{\text{ego}_{k-1}} = \left[ \begin{array}{c|c} \text{Rotationsmatrix} & \text{Translationsvektor} \\ \hline \mathbf{R}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{array} \right] \quad (5.1)$$



**Abbildung 5.5:**

Prädiktion der ego-relativen Fahrzeugpose  ${}^{obj_k}P_{ego_k}$  bzw. der homogenen Transformationsmatrix  ${}^{obj_k}H_{ego_k}$ .

umgerechnet werden. Im nächsten Schritt wird das Schätzergebnis vom Zeitpunkt  $k - 1$  bzgl. der inversen Eigenbewegung  ${}^{ego_k}H_{ego_{k-1}}^{-1}$  berücksichtigt:

$${}^{obj_{k-1}}H_{ego_k} = {}^{obj_{k-1}}H_{ego_{k-1}} \cdot {}^{ego_k}H_{ego_{k-1}}^{-1} \quad (5.2)$$

In der Gleichung 5.2 kennzeichnet  $H^{-1}$  eine Matrixinversion. Abschließend ist für eine möglichst realistische Vorhersage die Bewegung des Führungsfahrzeuges  ${}^{obj_k}H_{obj_{k-1}}$  einzubeziehen:

$${}^{obj_k}H_{ego_k} = {}^{obj_k}H_{obj_{k-1}} \cdot {}^{obj_{k-1}}H_{ego_k} \quad (5.3)$$

In der Gleichung 5.3 beschreibt die homogene Transformationsmatrix  ${}^{obj_k}H_{obj_{k-1}}$  die prädizierte Bewegung des Führungsfahrzeuges vom Zeitpunkt  $k - 1$  bis zum Zeitpunkt  $k$ . Als Bewegungsmodell kann das Ackermann-Einspurmodell für vorderachsgelenkte Fahrzeuge verwendet werden:

$$\dot{\psi} = \frac{v \cdot \tan(\lambda)}{l} \quad (5.4)$$

Hierbei kennzeichnet  $\dot{\psi}$  die Gierrate,  $v$  die Fahrzeuggeschwindigkeit,  $\lambda$  den Lenkwinkel und  $l$  den Radstand. Details zum Ackermann-Einspurmodell findet sich im Anhang A.4.

Eine Vorhersage über die acht 2D-Eckpunkte  $\mathbf{y}_{1,\dots,8} = \{(u_1, v_1), \dots, (u_8, v_8)\}$  der Fahrzeugseiten ist durch Projektion der prädizierten Transformationsmatrix  ${}^{obj_k}H_{ego_k}$  auf die Bildebene möglich.

### 5.1.2.2 Innovation

Die Innovation besteht aus der Durchführung einer oder mehrerer Messungen und deren Verwendung zur Korrektur des bestehenden Schätzergebnisses über die Filterzustände  $x$ . Eine Messung besteht aus vier von insgesamt acht Eckpunkten  $y_{1,\dots,8}$ , welche die Fahrzeughülle bzw. ein Fahrzeug umschließendes Quader partiell beschreiben. Das Ergebnis eines einzelnen Klassifikators liefert keinen Eckpunkt (Fahrzeugansicht ist nicht sichtbar oder wurde nicht erkannt) oder vier Eckpunkte (Fahrzeugansicht wurde erkannt) die sequentiell eingefiltert werden. Die sequentielle Einfilterung von (Teil-)Messungen wurde im Kapitel 3.2.1.3 erläutert. Zum einen ist dies notwendig, wenn Klassifikatoren für unterschiedliche Fahrzeugansichten – Rotationen von  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $\dots$ ,  $315^\circ$  – trainiert werden. Des Weiteren soll die Verwendung von mehreren Kameras simultan möglich sein, da dies eine deutliche Robustheitssteigerung mit sich führt. Beispielsweise sind damit Kameras mit Tele- und Weitwinkelobjektiv kombinierbar.

In dieser Arbeit werden zur Erhöhung der Genauigkeit und Robustheit jeweils acht Klassifikatoren pro Fahrzeug trainiert. Bei der Verwendung von acht Klassifikatoren decken diese jeweils die  $360^\circ / 8 = 45^\circ$ -Bereiche vom Fahrzeug ab. Im Praxiseinsatz dienen sie zur Erkennung der Fahrzeugseiten bzw. dessen Eckpunkte, welche der Kalman-Filter wiederum zur Positionsschätzung verwendet. Zusätzlich ermöglicht die zeitliche Mitnahme von Wissen die Suche nach dem Führungsfahrzeug in einem reduzierten Bildbereich. Die Abbildung 5.3 veranschaulicht diese effiziente Objektverfolgung. Ein  $45^\circ$ -Klassifikator detektiert die Fahrzeugrückseite zusammen mit der sichtbaren linken Fahrzeugseite. Das Klassifikatorergebnis ist ein aufrechtes Rechteck bestehend aus vier 2D-Bildkoordinaten und wird in der Abbildung durch ein grünes Rechteck visualisiert. Diese Detektion bzw. die vier Eckpunkte werden in den Kalman-Filter eingefiltert. Beim Eintreffen des nächsten Kamerabildes wird eine AOI anhand der prädizierten Filterzustände berechnet, sodass der Klassifikator in einem kleinen Bildbereich angewendet werden kann.

Die in diesem Kapitel vorgestellte visuelle Fahrzeugerkennung und -verfolgung wurde in Fries et al. [2013] und Fries und Wuensche [2014] veröffentlicht. Zusammenfassend prädiziert ein Unscented Kalman-Filter die Zustands- und Messwerte für einen diskreten Zeitpunkt. Anschließend erfolgt die Detektion eines bestimmten Führungsfahrzeuges in allen vorhanden Kamerabildern. Das neu gewonnene Wissen über die Fahrzeugseiten bzw. dessen Eckpunkte entspricht der Messung, welche mit der prädizierten Messung sequentiell verglichen wird, damit eine Korrektur der Zustandsschätzung erfolgen kann. Auf handelsüblichen Computern benötigt dieser Vorgang wenig Rechenlast, was eine seriennahe Verwendung in der Automobilindustrie begünstigt. Allerdings existieren zwei Nachteile bei der Verwendung von trainierten Klassifikatoren. Zum einen ist der Trainingsaufwand groß und erfolgt gängigerweise überwacht. Zum anderen ist die Genauigkeit der Positionsschätzung nur teilweise für spurtreues Folgen geeignet. Das nächste Kapitel geht auf ein Verfahren ein,

welches viel Rechenleistung benötigt, allerdings eine sehr genaue Positions- und Geschwindigkeitsschätzung aufweist.

### 5.1.3 Objekterkennung mittels Partikel-Filter

In diesem Kapitel wird die Erkennung und Verfolgung eines bekannten Führungsfahrzeuges durch einen Partikel-Filter beschrieben. Auf den Aufbau und die Funktionsweise von diesem Filter wurde im Grundlagenkapitel 3.2.2 detailliert eingegangen, sodass nachfolgend dessen Verwendung beschrieben wird.

Ein Partikel-Filter ist ein geschlossener Regelkreis mit einer rekursiven Schätzung einer unbekannt multimodalen Wahrscheinlichkeitsdichtefunktion (WDF) der Zustände eines dynamischen Systems. Die Beschreibung einer WDF erfolgt durch eine Vielzahl von Partikeln, die aus einem Gewicht und einer Stützstelle der Wahrscheinlichkeitsverteilung bestehen. In dieser Arbeit werden je nach Rechenkapazität  $N = 500 \dots 2000$  Partikel und damit Fahrzeughypothesen verarbeitet. Die zu schätzenden Zustände sind einerseits die in Zylinderkoordinaten angegebene ego-relative Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}} = \{r, \alpha, h, \phi, \theta, \psi\}$  zum Führungsfahrzeug und des Weiteren dessen Längsbeschleunigung  $a$ , der Lenkwinkel  $\lambda$  und die Längs- und Quergeschwindigkeit  $\mathbf{v}_x$  und  $\mathbf{v}_y$ . Zusammenfassend ergibt sich der Zustandsvektor  $\mathbf{x} = [r, \alpha, h, \phi, \theta, \psi, a, \lambda, \mathbf{v}]$ . Nachfolgend wird auf den rekursiven Schätzzyklus, bestehend aus der Prädikation der WDF mit anschließender Innovation durch eine Gewichtung und Neuverteilung der Partikel, eingegangen.

#### 5.1.3.1 Prädiktion

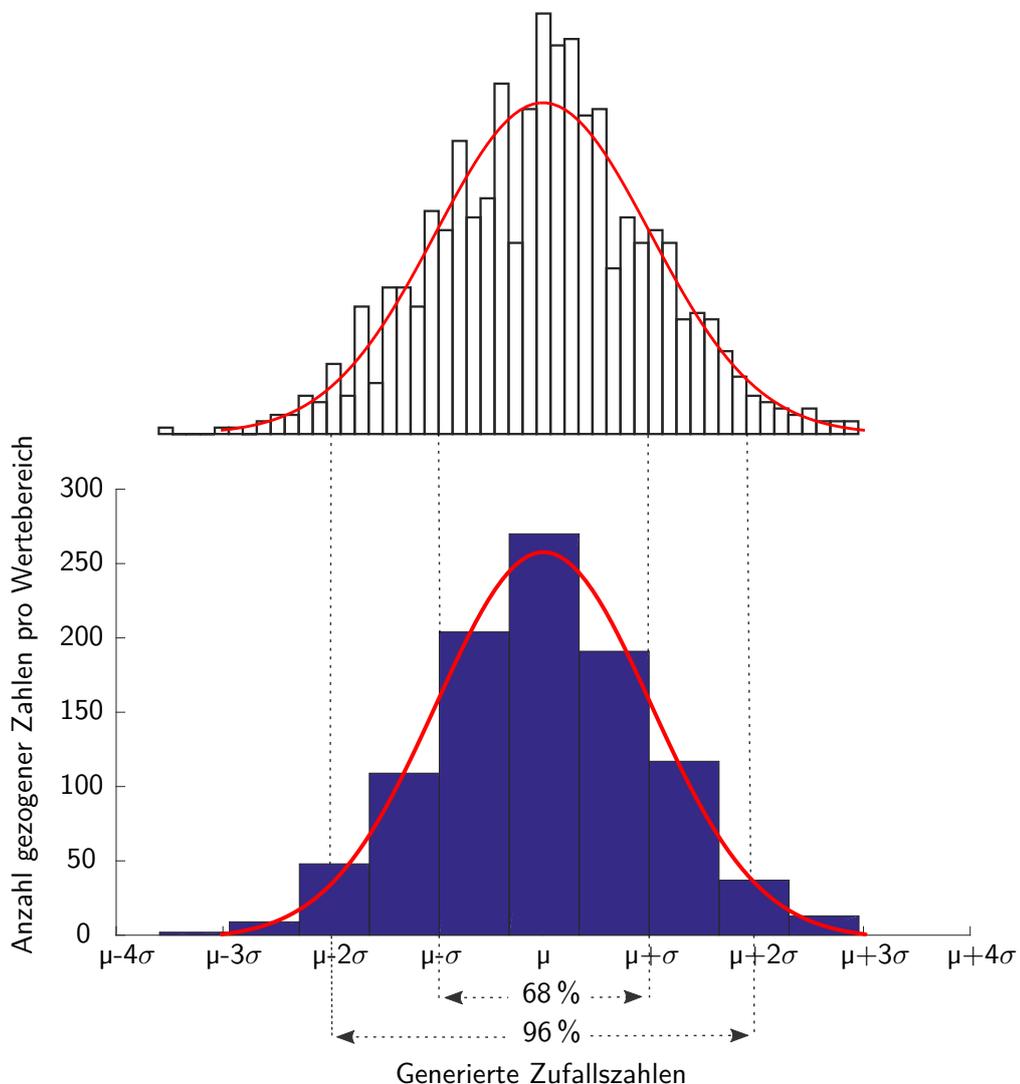
Im ersten Rekursionsschritt erfolgt die Vorhersage  $(\cdot)^*$  der genannten Zustandsgrößen  $\mathbf{x}$  für einen diskreten Zeitpunkt  $k$ . Für eine Vorhersage ist zum einen die Eigenbewegung vom vorherigen Zeitpunkt  $k-1$  auf den aktuellen Zeitpunkt notwendig. Zum anderen muss die Objektbewegung prädiziert werden. Für eine realistische Prädiktion der Fahrzeugbewegung und zur Bestimmung des Fahrzeuglenkwinkels  $\lambda$  ist ein Bewegungsmodell erforderlich. In dieser Arbeit wird das Ackermann-Einspurmodell verwendet, welches für vorderachsgelenkte Fahrzeuge geeignet und im Anhang A.4 näher beschrieben ist. Somit kann eine zeitliche Abbildung der  $N$  Partikel  $\mathbf{x}^{(i=1, \dots, N)}$  unter Berücksichtigung der Systemdynamik durchgeführt werden:

$$\mathbf{x}_k^{*(i)} = f_{k-1} \left\{ \hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}, \mathbf{s}_{k-1}^{(i)} \right\} \quad \text{für } i = 1, \dots, N. \quad (5.5)$$

Hierbei entspricht  $\mathbf{s}_{k-1}^{(i)}$  geringfügig zufälliges Rauschen, sodass fehlerhafte Modellierungen des Bewegungsmodells sowie anderweitige Störgrößen im Prädiktionsschritt berücksichtigt werden. Die Stärke der Verrauschung ist vom Anwendungsbereich abhängig. Steht beispielsweise das Ego-Fahrzeug still und das Führungsfahrzeug fährt ohne Lenkwinkeländerung mit 100 km/h davon, dann hat sich das Führungsfahrzeug

nach einem typischen Datenzyklus von 50 ms in etwa  $\frac{100 \text{ m}}{3,6 \text{ s}} \cdot 50 \text{ ms} \approx 1,4 \text{ m}$  bewegt. Würde in dieser Situation die Fahrzeugbewegung des Führungsfahrzeuges nicht berücksichtigt werden, dann müsste eine zusätzliche Verrauschung bzw. Streuung der Positionswerte der Partikel im Bereich von -1,4 bis 1,4 Metern erfolgen. Ansonsten kann es vorkommen, dass sich keine Fahrzeughypothese in der Nähe des Führungsfahrzeuges aufhält und damit eine Fahrzeugerkennung unmöglich ist. Aus diesem Grund wird das Hinzufügen von zufälligem Rauschen empfohlen. In dieser Arbeit erfolgt die zufällige Verrauschung der prädizierten Zustandswerte normalverteilt. Dafür wird einmalig für jede Zustandsgröße der Streubereich  $\sigma$  definiert. Anschließend kann jeder prädizierte Zustandswert in diesem Bereich verrauscht werden.

Die Abbildung 5.6 veranschaulicht die normalverteilte Ziehung von 1000 Zufallszahlen um einen Erwartungswert  $\mu$  herum.



**Abbildung 5.6:**

Zufallszahlengenerator für eine Normalverteilung mit dem Erwartungswert  $\mu$  und der Standardabweichung  $\sigma$ .

Des Weiteren zeigt die Abbildung 5.6, dass bei einer Normalverteilung statistisch gesehen 68,27 Prozent der Zufallszahlen im Bereich von  $\pm\sigma$  und 95,45 Prozent im Intervall von  $\pm 2\sigma$  liegen. Folglich decken die verrauschten prädizierten Zustandswerte den Streuungsbereich  $\pm 3\sigma$  aus statistischer Sicht immer ab, auch beim Sonderfall  $\hat{\mathbf{x}}^{(1)} = \hat{\mathbf{x}}^{(2)} = \hat{\mathbf{x}}^{(3)} = \dots = \hat{\mathbf{x}}^{(N)}$ .

In dieser Arbeit wurden diverse Streuungsbereiche evaluiert. Dabei haben sich die in der Tabelle 5.1 aufgelisteten Rausch- bzw. Streuungsbereiche bewährt. Diese sind aufgrund der Berücksichtigung einer zentimeter-genauen Eigenbewegungsschätzung sehr gering und für einen zeitlichen Prädiktionshorizont bis zu 50 ms gut geeignet. Sie ermöglichen das Verfolgen eines hochdynamischen Führungsfahrzeuges. Entsprechend kann bei langsamen geringdynamischen Bewegungen des Führungsfahrzeuges der Streuungsbereich reduziert werden. Für den Fall, dass neue Sensordaten erst nach 50 ms eintreffen, sollte der Streuungsbereich vergrößert werden. Andernfalls besitzen bei einer dynamischen Fahrzeugbewegung keine Partikel die näherungsweise wahren Fahrzeugzustände und die Fahrzeugdetektion schlägt fehl. In dieser Arbeit erfolgt aus diesem Grund die Verrauschung in Abhängigkeit des Sensordatenzyklus. Dadurch führen u. a. Netzwerküberbelastungen und deren Folgen (Bspw.: Ein Kamerabild trifft erst nach 0,5 Sekunden zur Verarbeitung ein) nicht zum Fahrzeugverlust, wenn ausreichend breitflächig gestreut wird. In einem solchen Fall sollte der Streuungsbereich zehn mal so groß wie der aufgelistete 50 ms-Streuungsbereich sein.

Zustandsgrößen		Formelzeichen	Einheit	$\sigma$	
obj $\mathbf{P}_{ego}$	Position	Abstand	$r$	(m)	$2 \cdot 10^{-1}$
		Winkel	$\alpha$	(°)	$1 \cdot 10^{-2}$
		Höhe	$h$	(m)	$2 \cdot 10^{-2}$
	Orientierung	Rollwinkel	$\phi$	(°)	$2 \cdot 10^{-2}$
		Nickwinkel	$\theta$	(°)	$5 \cdot 10^{-3}$
		Gierwinkel	$\psi$	(°)	$1 \cdot 10^{-1}$
Geschwindigkeit	Längsrichtung	$\mathbf{v}_x$	(m/s)	$2 \cdot 10^{-1}$	
	Querrichtung	$\mathbf{v}_y$	(m/s)	$2 \cdot 10^{-4}$	
Längsbeschleunigung		$a$	(m/s <sup>2</sup> )	$2 \cdot 10^{-2}$	
Lenkwinkel		$\lambda$	(°)	$3 \cdot 10^{-1}$	

**Tabelle 5.1:**

Hinzufügen von zufälligem Rauschen auf prädizierte Zustandswerte. Das Ziel ist eine ausreichende Zustandsraumabdeckung und -dichte mit Partikeln bzw. Fahrzeughypothesen. Dabei beziehen sich die Rauschbereiche auf einen 50 ms Daten- und Verarbeitungszyklus, eine bekannte Eigenbewegung und eine Fahrzeuggeschwindigkeit bis 100 km/h.

An dieser Stelle sei allerdings noch darauf hingewiesen, dass bei einer starken Streuung der Partikel die Dichte des Zustandsraumes geringer wird und somit eine Fahrzeugdetektion fehlerhaft bis gar unmöglich werden kann. Aus diesem Grund sollte die Partikelanzahl höher gewählt werden, wenn der Sensordatenzyklus langsam ist bzw. Netzwerkstörungen auftreten können.

Zusammenfassend wurde in diesem Abschnitt der Prädikationsschritt des Partikel-Filters beschrieben, welcher unter Berücksichtigung der Eigenbewegung, eines Bewegungsmodells und des Sensordatenzykluses die zu schätzenden Zustände – wie u. a. die ego-relative Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}}$  – voraussagt. Diese prädizierte Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}}^*$  wird zeitabhängig verrauscht und im nächsten Schritt - der Innovation - ausgewertet.

### 5.1.3.2 Innovation

Der zweite Rekursionsschritt wird als Innovation bezeichnet und besteht aus zwei grundlegenden Phasen:

- Messung und Gewichtung der Fahrzeughypothesen
- Neuverteilung der Partikel

Zuerst erfolgt die Durchführung der Messung

$$\mathbf{y}_k^{(i)} = \left[ \mathbf{y}_{\text{Merkmal 1}}^{(i)}, \mathbf{y}_{\text{Merkmal 2}}^{(i)}, \dots, \mathbf{y}_{\text{Merkmal } n}^{(i)} \right]_k^T \quad (5.6)$$

für den aktuellen Zeitpunkt  $k$ . Hierfür sind jeweils die sechs prädizierten zylindrischen Koordinaten einer Fahrzeugpose  ${}^{\text{obj}_k}\mathbf{P}_{\text{ego}_k}^{*(i)}$  und ein fahrzeugspezifisches 3D-Merkmalmodell erforderlich. Das Partikel-Filter arbeitet mit  $N$  Partikeln, wodurch eine Messauswertung für jedes Partikel bzw. jede vorhergesagte Fahrzeughypothese

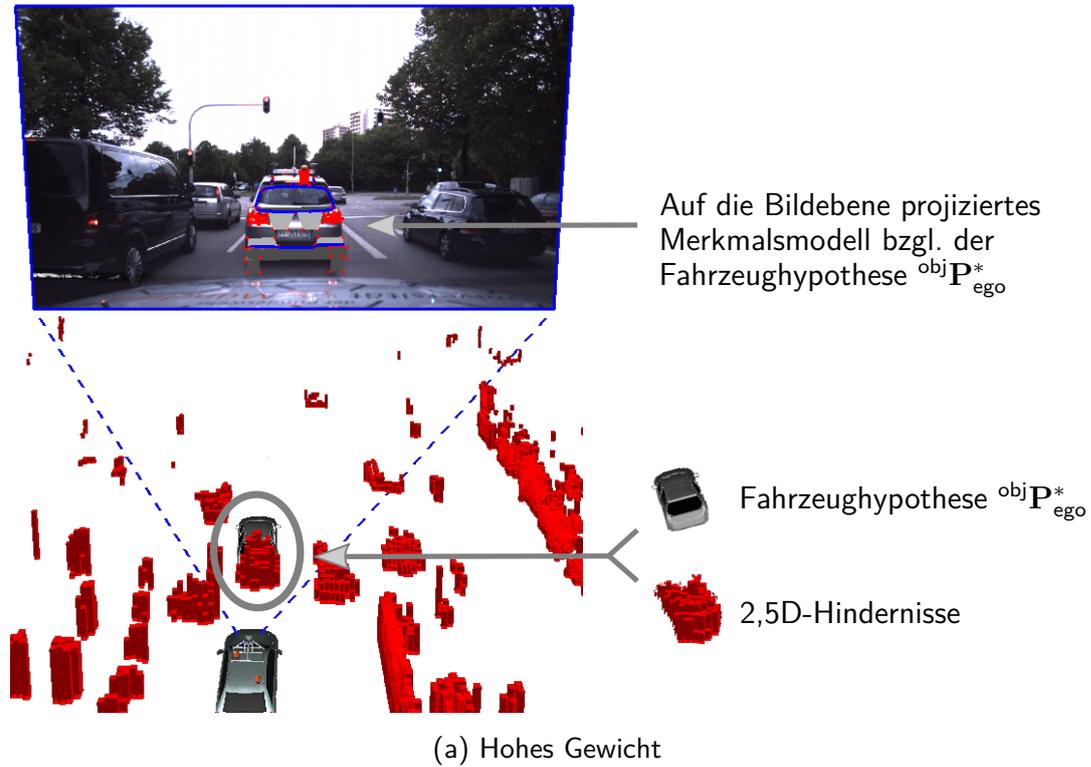
$${}^{\text{obj}_k}\mathbf{P}_{\text{ego}_k}^{*(i)} \in \left\{ {}^{\text{obj}_k}\mathbf{P}_{\text{ego}_k}^{*(1)}, {}^{\text{obj}_k}\mathbf{P}_{\text{ego}_k}^{*(2)}, \dots, {}^{\text{obj}_k}\mathbf{P}_{\text{ego}_k}^{*(N)} \right\} \quad (5.7)$$

durchgeführt werden muss. Das weitere Vorgehen ist damit die Bewertung

$$\mathbf{w}_k^{(i)} = \mathbf{w}_{k,\text{Merkmal 1}}^{(i)} \cdot \mathbf{w}_{k,\text{Merkmal 2}}^{(i)} \cdot \dots \cdot \mathbf{w}_{k,\text{Merkmal } n}^{(i)} \quad (5.8)$$

für jede prädizierte Fahrzeughypothese. Die Gewichtung von Fahrzeughypothesen erfolgt auf Grundlage eines 3D-Merkmalmodells, durch einen Abgleich zwischen vorhergesagten und wahrgenommenen fahrzeugspezifische Merkmalen.

Die Abbildung 5.7 veranschaulicht die Gewichtung von drei Partikeln exemplarisch. In der Abbildung 5.7a ist zusätzlich zum Kamerabild das 3D-Umgebungsmodell dargestellt. In diesem ist zum einen das autonome Ego-Fahrzeug und eine Fahrzeughypothese  ${}^{\text{obj}}\mathbf{P}_{\text{ego}_k}^*$  (i) zu sehen. Zum anderen sind die 2,5D-Hinderniszellen durch rote Balken dargestellt.



(b) Niedriges Gewicht



(c) Niedriges Gewicht

**Abbildung 5.7:**

Partikelgewichtung durch Abgleich zwischen vorhergesagten und wahrgenommenen fahrzeugspezifischen Merkmalen.

Wie bereits erwähnt, erfolgt die Bewertung der Fahrzeughypothese mit einem Abgleich zwischen gemessenen und vorhergesagten fahrzeugspezifischen Merkmalen. In diesem Fall werden Bildmerkmale und Hindernismerkmale gemessen. Die Auswertung der Hindernismerkmale kann direkt erfolgen. Bewertet wird ob die Zellen auf der Fahrzeughypothese belegt sind, ob die Höhe des Hindernisses stimmt und ob die benachbarten Zellen um das Fahrzeug herum frei sind. Die Auswertung von Bildmerkmalen kann nicht direkt erfolgen. Aus diesem Grund müssen die Bildmerkmale des 3D-Merkmalmodells ausgehend von der prädizierten Fahrzeugpose auf die Bildebene projiziert werden. Danach kann der Merkmalsabgleich zur Gewichtung der Fahrzeughypothese erfolgen. Bei der Verwendung von mehreren Kameras erfolgt entsprechend die Auswertung in allen Kamerabildern.

Für die Bewertung der vorhergesagten fahrzeugspezifischen Merkmale müssen einmalig pro diskretem Zeitpunkt  $k$  die Merkmale aus den aktuellen Sensordaten extrahiert werden. Dabei sollte die Merkmalsextraktion nicht – wie in Manz [2013] – in den gesamten Sensordaten erfolgen, stattdessen ist die Extraktion auf einen kleinen Teilbereich reduzierbar. Diese AOI berechnet sich aus dem aktuellen Streuungsbereich der Partikel und wird zusammen mit weiteren Strategien zur Aufmerksamkeitssteuerung im Kapitel 5.2 beschrieben.

Nachfolgend wird auf die **Extraktion** und den **Merkmalsabgleich** der Hindernis- und Bildmerkmale und der daraus resultierenden Partikelbewertung eingegangen.

- **Extraktion von Kanten mit Intensität und Richtung**

Zu den Bildmerkmalen gehören zum einen prägnante Kanten, welche sich typischerweise an den innenliegenden Fahrzeugkonturen, an Fenstern und am Nummernschild befinden. Für die Kantenextraktion aus einem Farbbild kann der häufig verwendete Canny-Algorithmus [Canny, 1986] verwendet werden. Der Algorithmus führt die Suche nach Kanten in einem Grauwertbild  $I$  durch, wodurch ein Farbbild  $F$  erst in ein Grauwertbild umgewandelt werden muss. Anschließend erfolgt im Grauwertbild eine Bildglättung zur Unterdrückung von Rauschen. Eine dafür zu verwendende Faltungsmaske wurde bereits im Abschnitt 4.2.3 auf S. 60 vorgestellt. Je größer die Glättungs- bzw. Faltungsmaske gewählt wird, desto stärker ist der Glättungseffekt. Anschließend sind die zwei Sobel-Faltungsmasken  $G_u$  und  $G_v$  für die getrennte Extraktion der vertikalen und horizontalen Kanten auf das geglättete Bild anzuwenden. Das Zusammenfügen der beiden Faltungsergebnisse folgt durch  $G = \sqrt{G_u^2 + G_v^2}$  und ergibt ein Kantenintensitätsbild. Alternativ kann das Faltungsergebnis durch  $G = |G_u| + |G_v|$  approximiert werden, sodass die Rechenzeit verringert wird. Weiterführend wird in der Praxis das Intensitätsbild in ein Binärbild reduziert, da die Bildverarbeitung auf Binärdaten deutlich performanter ist. Eine solche Binärkonvertierung erfolgt durch eine Schwellwertbildung. Die Abbildung 5.8 veranschaulicht eine solche Unterdrückung von Nicht-Maxima. Des Weiteren ist es möglich ein Richtungsbild  $I_\phi = \arctan\left(\frac{G_v}{G_u}\right)$  aus den Faltungsergebnissen des Sobel-Operators zu erstellen.



(a) Eingangsbild



(b) Intensitätsbild



(c) Binärbild

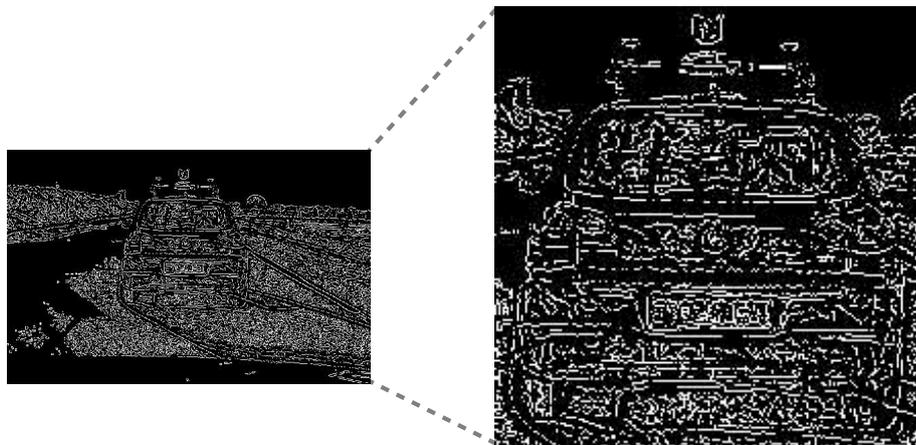
**Abbildung 5.8:**

Kantenextraktion mittels Canny-Algorithmus, durchgeführt innerhalb einer AOI: (a) Vollständiges Grauwertbild. (b) Resultierendes Intensitätsbild. (c) Binäres Kantenbild nach Unterdrückung von Nicht-Maxima.

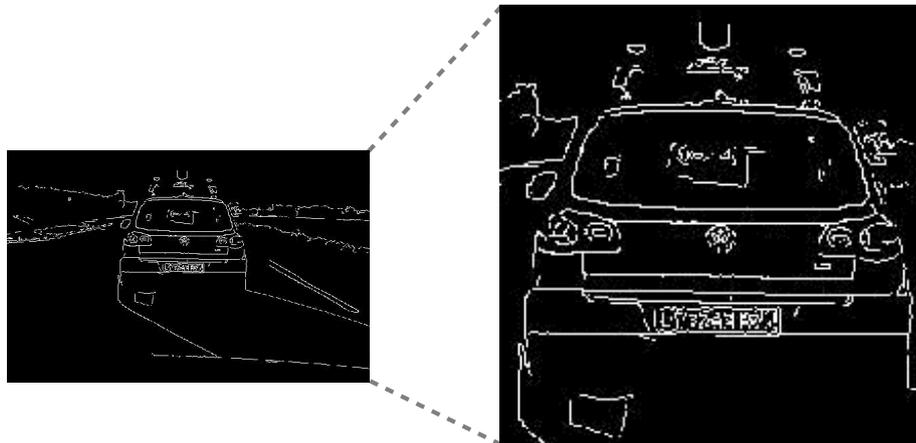
An dieser Stelle wird noch explizit auf die Größe ( $3 \times 3$  ;  $5 \times 5$  ;  $7 \times 7$  ; ...) einer Faltungsmaske eingegangen, denn diese ist je nach Anwendungsfall unterschiedlich einzustellen, was die Abbildung 5.9 deutlich veranschaulicht. In dem Anwendungsgebiet der Fahrzeugerkennung im Nahbereich sollte die Maskengröße nicht zu klein gewählt werden, ansonsten sind – wie in der Abbildung 5.9b gezeigt – die dominanten Fahrzeugkonturen nicht zuverlässig detektierbar. Dagegen sollte für eine Kantendetektion eines weit entfernten Fahrzeuges keine zu große Maske gewählt werden. Zusammenfassend wird eine variable entfernungsabhängige Maskengröße empfohlen.



(a) Eingangsbild



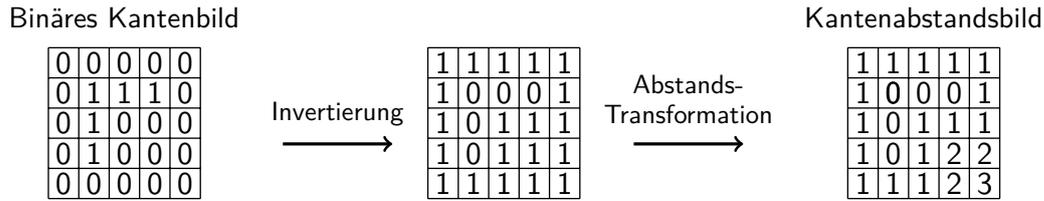
(b) Faltungsergebnis der suboptimal gewählten Maskengröße



(c) Faltungsergebnis einer besser gewählten Maskengröße

**Abbildung 5.9:**

Die Größe der Faltungsmaske ist ein wichtiger Parameter für die Suche nach bestimmten Kanten. Ein Anwendungsbeispiel ist die Detektion der Fahrzeugkonturen im Nahbereich.

**Abbildung 5.10:**

Berechnung der Kantenabstände mittels definierter Abstandstransformation.

- **Ableich zwischen wahrgenommenen und vorausgesagten Kanten**

Die Wahrnehmung der Kanten, sowie deren Intensität und Richtung aus einem aktuell eingetroffenen Kamerabild erlaubt die Bewertung der Partikel. Jedes Partikel entspricht einer ego-relativen Fahrzeughypothese, welche für eine Bewertung (Partikelgewichtung) mit den bekannten Modellmerkmalen in das aktuelle Kamerabild projiziert wird. Für die Bewertung ist der Kantenabstand, deren Orientierung und die Intensität zu betrachten.

Der Kantenabstand zwischen einer prädierten und einer gemessenen Kante  $K_{\text{Prädiziert}}$  und  $K_{\text{Messung}}$  kann durch die Berechnung eines Kantenabstandsbildes  $I_{\text{DT}}$  bestimmt werden. Ein Kantenabstandsbild kann durch eine Abstandstransformation (engl. *Distance Transform*) berechnet werden. Die Abbildung 5.10 veranschaulicht die Berechnung der Kantenabstände grafisch. Für detaillierte Informationen über eine geeignete Abstandstransformation wird auf Rosenfeld und Pfaltz [1966] verwiesen.

Der nächste Schritt ist die Diskretisierung der  $N$  prädierten Kanten  $K_{1\dots N}$  in  $n$  einzelne 2D-Punkte  $\mathbf{p}_j = [u, v]_j$ . Anschließend können die Kantenabstände direkt mit dem Kantenabstandsbild ermittelt, aufsummiert und gemittelt werden:

$$\mathbf{y}_{\text{Kantenabstand}} = \frac{1}{n} \cdot \sum_{j=1}^n I_{\text{DT}}(\mathbf{p}_j) \quad (5.9)$$

Der Ableich der prädierten und gemessenen Kantenintensität kann direkt anhand des Intensitätsbildes ermittelt werden, welches bereits mit dem Canny-Algorithmus berechnet wurde (vgl. Abbildung 5.8b). Hierbei werden die Pixelwerte bzw. Intensitäten der diskretisierten prädierten Kanten direkt aus dem Intensitätsbild ermittelt, aufsummiert und gemittelt. Es resultiert der Intensitätswert  $\mathbf{y}_{\text{Kantenstärke}}$ .

Der Winkelunterschied  $\mathbf{y}_{\text{Kantenrichtung}}$  zwischen einer prädierten und gemessenen Kante  $K_{\text{Prädiziert}}$  und  $K_{\text{Messung}}$  erfolgt in dieser Arbeit anhand des Skalarproduktes:

$$\angle(K_{\text{Prädiziert}}, K_{\text{Messung}}) = \arccos\left(\frac{K_{\text{Prädiziert}} \cdot K_{\text{Messung}}}{|K_{\text{Prädiziert}}| \cdot |K_{\text{Messung}}|}\right) \quad (5.10)$$

Abschließend erfolgt die Merkmalsbezogene Partikelgewichtung hinsichtlich der drei Kantenmerkmale:

$$\mathbf{w}_{k,\text{Kanten}}^{(i)} = \mathbf{w}_{k,\text{Kantenabstand}}^{(i)} \cdot \mathbf{w}_{k,\text{Kantenstärke}}^{(i)} \cdot \mathbf{w}_{k,\text{Kantenrichtung}}^{(i)} \quad (5.11)$$

Die Gleichung 5.12 beschreibt eine Gewichtungsvariante bezüglich der Kantenabstände.

$$\mathbf{w}_{k,\text{Kantenabstand}}^{(i)} = e^{-\left(\frac{y_{\text{Kantenabstand}}}{\sigma_{\text{Kantenabstand}}}\right)^2} \quad (5.12)$$

Die Exponentialfunktion  $e^{-\left(\frac{y_{\text{Kantenabstand}}}{\sigma_{\text{Kantenabstand}}}\right)^2}$  aus der Gleichung 5.12 liefert den Wert eins, falls der Kantenabstand  $y_{\text{Kantenabstand}}$  dem Wert null entspricht. Ist der Kantenabstand größer als null, sinkt der resultierende Gewichtungswert  $\mathbf{w}_{k,\text{Kantenabstand}}^{(i)}$ . Mit der Konstanten  $\sigma_{\text{Kantenabstand}}$  ist es möglich die Exponentialfunktion zu „dehnen“, sodass geringe Kantenabstände zu einem hohen Gewichtungswert führen. In dieser Arbeit wurde  $\sigma_{\text{Kantenabstand}}$  auf 15 Pixel gesetzt.

Der Kantenwinkelunterschied  $y_{\text{Kantenrichtung}}$  kann ebenfalls mit einer Exponentialfunktion bewertet werden:

$$\mathbf{w}_{k,\text{Kantenrichtung}}^{(i)} = e^{-\left(\frac{y_{\text{Kantenrichtung}}}{\sigma_{\text{Kantenrichtung}}}\right)^2} \quad (5.13)$$

Dagegen wird die Kantenintensität  $y_{\text{Kantenstärke}}$  nicht mit einer Exponentialfunktion bewertet. Hier ist  $y_{\text{Kantenstärke}}$  direkt für die Gewichtung verwendbar. Je größer die Kantenintensität  $y_{\text{Kantenstärke}}$ , desto höher ist der Gewichtungswert  $\mathbf{w}_{k,\text{Kantenstärke}}^{(i)}$ .

#### ▪ Extraktion und Vergleich von Eckpunkten

Prägnante Eckpunkte verbessern als weiteres Bildmerkmal ebenfalls die visuelle Objekterkennung. Typischerweise sind diese an sich kreuzenden Fahrzeugkanten, dem Nummernschild oder am Herstellerlogo zu finden. Für die Extraktion von Eckpunkten gibt es unterschiedliche Verfahren, auf die im Kapitel 4.2.3 eingegangen wurde. Zum einen können die Schnittpunkte von horizontalen und vertikalen Kanten zur Lokalisierung der Eckpunkte verwendet werden. Zum anderen ist die Betrachtung der Eigenwerte der Kovarianzmatrix der horizontalen und vertikalen Bildgradienten typisch. Je größer die Eigenwerte der Kovarianzmatrix sind, desto höher ist die Wahrscheinlichkeit, dass an einer Bildposition ein Eckpunkt vorliegt. Bezüglich der Partikelgewichtung steigt die Gewichtung pro Übereinstimmung zwischen prädiiziertem und wahrgenommenem Eckpunkt.

#### ▪ Extraktion von Flächenmerkmalen

Zu den Flächenmerkmalen gehören einerseits Farbflächen – wie bspw. hellgraue, dunkelgraue, rote oder grüne Flächen – und andererseits Temperaturbereiche.

Die Beschreibung der Flächengeometrie erfolgt durch eine festgelegte Anzahl an Eckpunkten. Für eine schnelle Bildverarbeitung wird die Modellierung von Flächen durch Vierecke empfohlen. Hierfür wird im Nachfolgenden eine effiziente Verarbeitung der Flächenmerkmale beschrieben. Begonnen wird mit den Farbflächen, gefolgt von den Temperaturflächen.

Zu Beginn müssen die Farbmerkmale einer Fahrzeughypothese über die ego-relative Fahrzeugpose und anhand der bekannten extrinsischen Lage und Orientierung der Kamera in das Kamerabild projiziert werden. Dort findet anschließend die Auswertung der Merkmale statt, indem überprüft wird, wie viele Pixel vom projizierten Merkmal tatsächlich mit der Wahrnehmung übereinstimmen. Damit nicht für jedes Partikel die Merkmalsfarbe zeitaufwendig aus dem RGB-Farbbild extrahiert werden muss, sollten einmalig pro Sensordatenzyklus und pro modellierter Farbe ein Binär- und Linienintegralbild [Viola und Jones, 2001a,b] erzeugt werden. Anschließend sind diese erzeugten Bilder für eine performante und parallele Merkmalsauswertung aller Fahrzeughypothesen nutzbar. Der erste Schritt ist daher die Erzeugung eines Binärbildes pro modellierter Farbe. Ein Binärbild wird durch eine Schwellwertbildung erzeugt und besteht aus Nullen und Einsen. Alle Pixel, die eine Eins als Pixelwert zugewiesen bekommen haben, besitzen den Farbwert nach dem gefiltert wurde. Da der RGB-Farbraum für eine Filterung nach einer bestimmten Farbe ungeeignet ist, erfolgt eine Farbsuche typischerweise im sogenannten HSV-Farbraum. Beide Farbräume und die Konvertierung vom RGB- in den HSV-Farbraum sind im Anhang A.3 ausführlich beschrieben. Im Nachfolgenden wird auf die Erzeugung eines Linienintegralbildes im Detail eingegangen. Dabei wird als Anwendungsbeispiel auf die Filterung nach der Farbe „rot“ eingegangen.

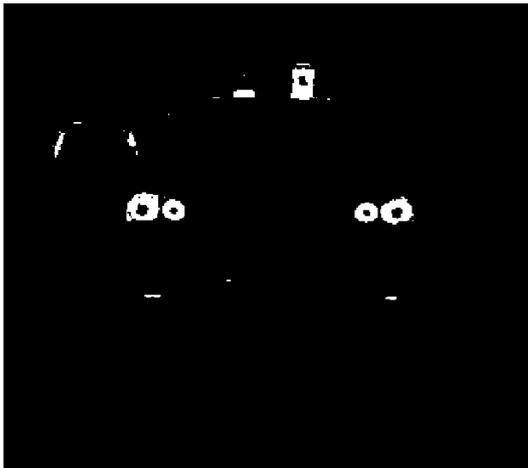
Zu Beginn erfolgt die Konvertierung eines RGB-Farbbildes in ein HSV-Bild nach Anhang A.3.3. Anschließend sollte das HSV-Bild zur Rauschunterdrückung geglättet werden. Hierbei kann eine Gauß-Filtermaske zum Einsatz kommen. Anschließend findet die Schwellwertbildung nach roten Pixeln statt, welche bereits im Kapitel 4.6 auf der S. 63 erläutert wurde. Die Schwellwertbildung erzeugt das Binärbild, welches an sich schon für die Auswertung aller roten Farbflächen nutzbar wäre, die Erzeugung und Verwendung eines Linienintegralbildes lässt allerdings eine deutlich schnellere Auswertung von Flächenmerkmalen zu. Ein Linienintegralbild  $I_L$  wird linienweise und parallelisiert aus einem Binärbild  $I$  erzeugt. Jeder Pixel im Linienintegralbild bekommt die kumulative Pixelsumme ausgehend vom ersten Pixel  $(1, v)$  seiner Zeile  $v$  bis zur eigenen Pixelposition  $(u, v)$  zugewiesen:

$$\forall v : I_L(u, v) = \sum_{u=1} I(u, v) \quad (5.14)$$

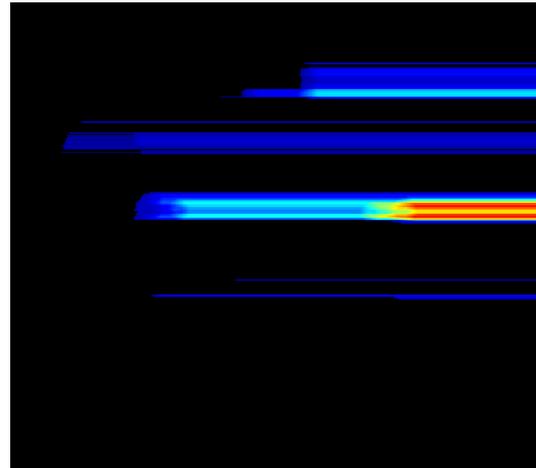
Die Abbildung 5.11 visualisiert das Ergebnis einer Farbfilterung und der damit verbundenen Berechnung eines Binär- und Linienintegralbildes. Dabei ist das erzeugte Linienintegralbild als Falschfarbenbild dargestellt, was die Aufsummierung von Pixelwerten sichtbar macht.



(a) Farbbild



(b) Binärbild bzgl. rote Pixel



(c) Linienintegralbild bzgl. rote Pixel

**Abbildung 5.11:**

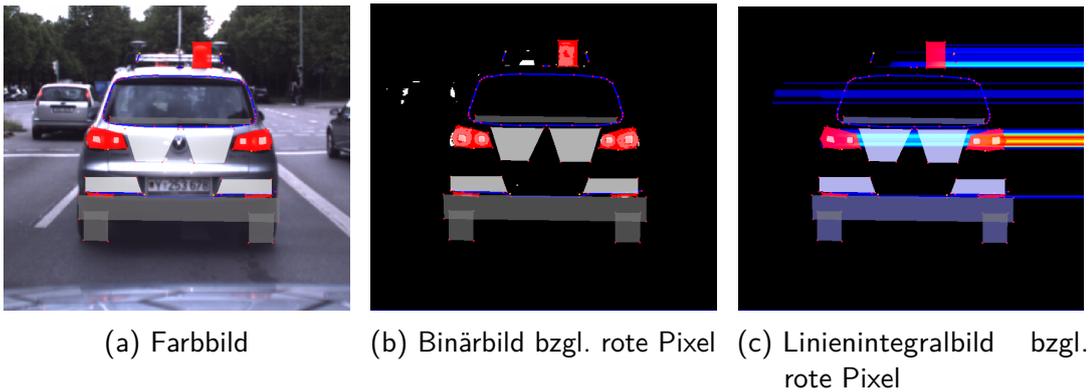
Filterung nach roten Bildpixeln: (a) RGB-Farbbild (b) Berechnetes Binärbild zeigt das Ergebnis der Filterung nach roten Bildpixeln. (c) Aus dem Binärbild erzeugtes Linienintegralbild, nutzbar für eine schnelle Abfrage der Anzahl von roten Pixeln in einem Bildbereich. Die Grafik ist als Falschfarbenbild dargestellt, sodass die Bilderzeugung mittels Pixelsummierung deutlich wird.

Die Generierung eines Binär- und Linienintegralbildes für ein Temperaturmerkmal erfolgt auf die gleiche Weise. Anstatt nach einer bestimmten Farbe im Farbbild zu filtern, wird nach einem Temperaturbereich im Wärmebild gefiltert. Dafür sollte die verwendete Wärmebildkamera kalibriert sein, sodass diese absolute realitätsgetreue Temperaturwerte misst. Ein Temperaturmerkmal ist zum Beispiel der heiße Auspuff eines Fahrzeuges. Dieser heizt sich unmittelbar nach dem Starten des Motors auf und es entstehen am Auspuffendstück Oberflächentemperaturen zwischen 25 und 60°C. Ein weiteres dominantes Temperaturmerkmal sind die Bereiche am Reifen. Die Reifen bestehen aus Gummi, welcher sich durch den Rollwiderstand bei einer Fahrzeugbewegung erwärmt. Die Suche nach durch Reibung aufgeheizte Reifen sollte durch eine adaptive Schwellwertbildung (engl. *Adaptive Thresholding*) [Fries und Wuensche, 2015] erfolgen. Der für die Schwellwertbildung notwendige Schwellwert ergibt sich durch die Detektion der Karosserietemperatur. Die Fahrzeugkarosserie nimmt die größte Fläche eines Fahrzeuges ein, was die Karosserie und deren Durchschnittstemperatur detektierbar macht. Da sich die Karosserie im Vergleich zu den Reifen nicht durch Bewegungswärme erwärmt, weisen die Reifen eine wärmere Temperatur auf. Der Temperaturwert für die Schwellwertbildung von Reifenmerkmalen muss aus diesem Grund höher sein als die Karosserietemperatur. In diesem Fall sollte allerdings der Temperaturwert für die Schwellwertbildung nicht dauerhaft festgelegt werden, da sich unterschiedliche Wetterbedingungen sehr stark auf die Materialtemperatur auswirken. Da zahlreiche Praxisversuch dieser Arbeit zeigten, dass sich die Reifen mindestens auf 18°C aufwärmen, kann höchstens eine Vorfilterung erfolgen. Der genaue Schwellwert wird folglich adaptiv bestimmt.

Die Auswertung von Farb- und Temperaturflächen kann mit Binärbildern erfolgen, wie bereits erwähnt wurde ist eine Auswertung mit Linienintegralbildern schneller, wenn viele Flächen für ein Bild ausgewertet werden müssen. Da in dieser Arbeit 500 bis 2000 Partikel zum Einsatz kommen und ein Fahrzeugmodell aus ca. 20 Flächenmerkmalen besteht, ist die Auswertung mit Linienintegralbildern zu empfehlen. Für die Auswertung eines in ein Kamerabild projizierten Flächenmerkmals (Viereck), muss dieses in  $l$  Linien  $\mathbf{l}_{1,\dots,l}$  diskretisiert werden (vgl. Diskretisierungsmethode von Bresenham [1965]). Anschließend ist die Merkmalsübereinstimmung durch die Berechnung der Pixelsumme der Merkmalsfläche mit lediglich  $l$  Subtraktionen und Additionen berechenbar:

$$\sum_{i=1}^l I_L(\mathbf{l}_{i,(u,v)_{\text{ende}}}) - I_L(\mathbf{l}_{i,(u,v)_{\text{start}}})$$

Hierbei kennzeichnen  $\mathbf{l}_{(u,v)_{\text{start}}}$  und  $\mathbf{l}_{(u,v)_{\text{ende}}}$  die jeweils erste und letzte Pixelkoordinate einer Linie  $\mathbf{l}$ .

**Abbildung 5.12:**

Hohe Partikelgewichtung aufgrund der Übereinstimmung zwischen den prädizierten und wahrgenommenen roten Flächenmerkmalen: (a) RGB-Farbbild inkl. projizierter Fahrzeughypothese. (b) Nach roten Bildpixeln gefiltertes Binärbild. (c) Aus dem Binärbild erzeugtes Linienintegralbild.

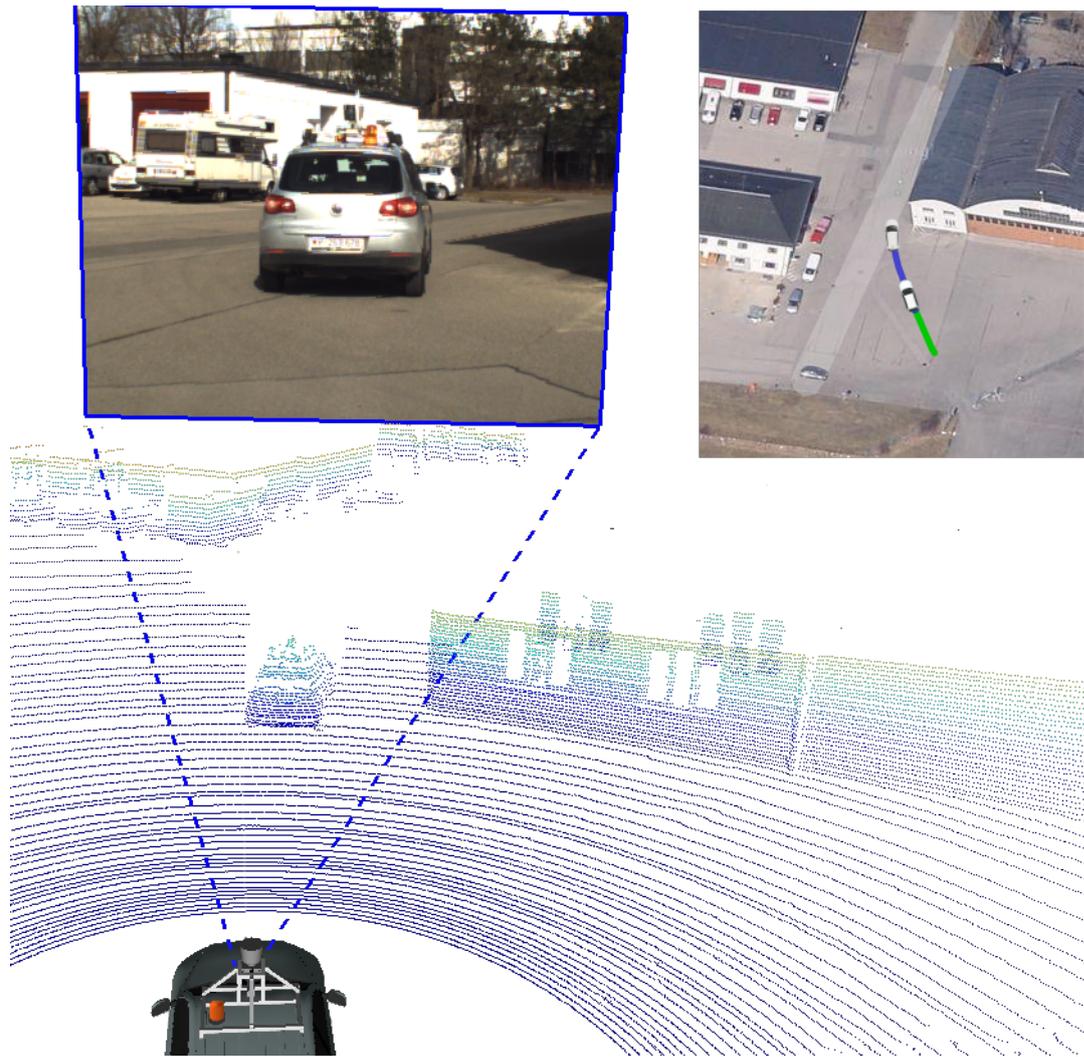
- **Gewichtung von Flächenmerkmalen**

Die Partikelgewichtung hängt von der Anzahl an prädizierten Pixeln ab, die mit den wahrgenommenen Farb- bzw. Temperaturwerten übereinstimmen. Je größer die Pixelsumme bzw. Merkmalsübereinstimmung ist, desto höher wird ein Partikel gewichtet. Die Abbildung 5.12 visualisiert eine gute Übereinstimmung zwischen den prädizierten und wahrgenommenen roten Flächenmerkmalen.

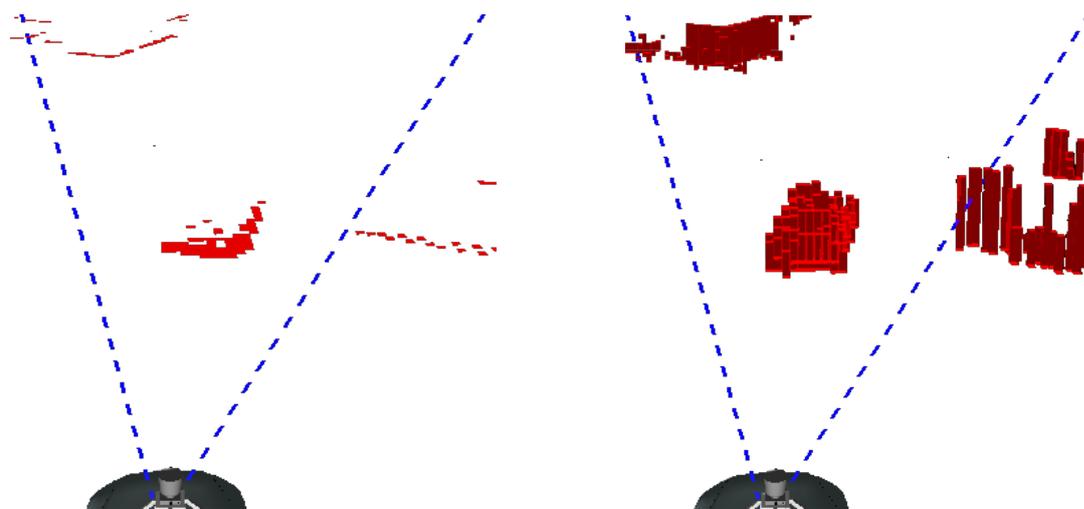
- **Hindernisbehaftete und hindernisfreie Zellen**

Mit dem Einsatz eines abstandsgebenden Sensors können die Hindernisse der Fahrzeugumgebung erfasst werden. Diese Arbeit verwendet einen oder mehrere LiDAR-Sensoren, die im Anhang A.1.1 beschrieben sind. Die Rohdaten eines rotierenden LiDAR-Sensors sind Abstandsinformationen, welche anhand der geschätzten Eigenbewegung [Luettel et al., 2009] zu einer zeitdiskreten 3D-Punktwolke zusammengefasst werden können [Schneider et al., 2010b]. Alternativ sind Tiefeninformationen mit einem kalibrierten Stereo-Kamerasystem ermittelbar. Die Verarbeitung von 3D-Punktwolken ist sehr zeitaufwendig. Stattdessen kann jede Punktwolke in ein 2D- oder 2,5D-Hindernisgitter (engl. *Occupancy Grid*) umgewandelt werden. Dafür muss einmalig die Größe des gesamten Hindernisgitters und der einzelnen Zellen definiert werden. In dieser Arbeit besitzt das zur Partikelgewichtung zum Einsatz kommende Hindernisgitter eine Größe von 50 - 100 m<sup>2</sup>. Das Hindernisgitter besteht wiederum aus symmetrischen Zellen, die 10 - 20 cm<sup>2</sup> groß sind. Die Erstellung eines Hindernisgitters aus einer 3D-Punktwolke ist folgendermaßen möglich. Treffen mindestens zwei Laserstrahlen auf eine Zelle, wird diese als belegt deklariert und die Hindernishöhe ist in diese Zelle einzutragen. Dabei berechnet sich die Hindernishöhe aus dem maximalen und minimalen Hinderniswert (Z-Koordinate) einer Zelle.

Die nachfolgende Abbildung 5.13 zeigt eine erstellte 2D- und 2,5D-Hinderniskarte.



(a) Punktwolke eines LiDAR-Sensors [Kartenmaterial: Google Maps]



(b) 2D-Hinderniskarte

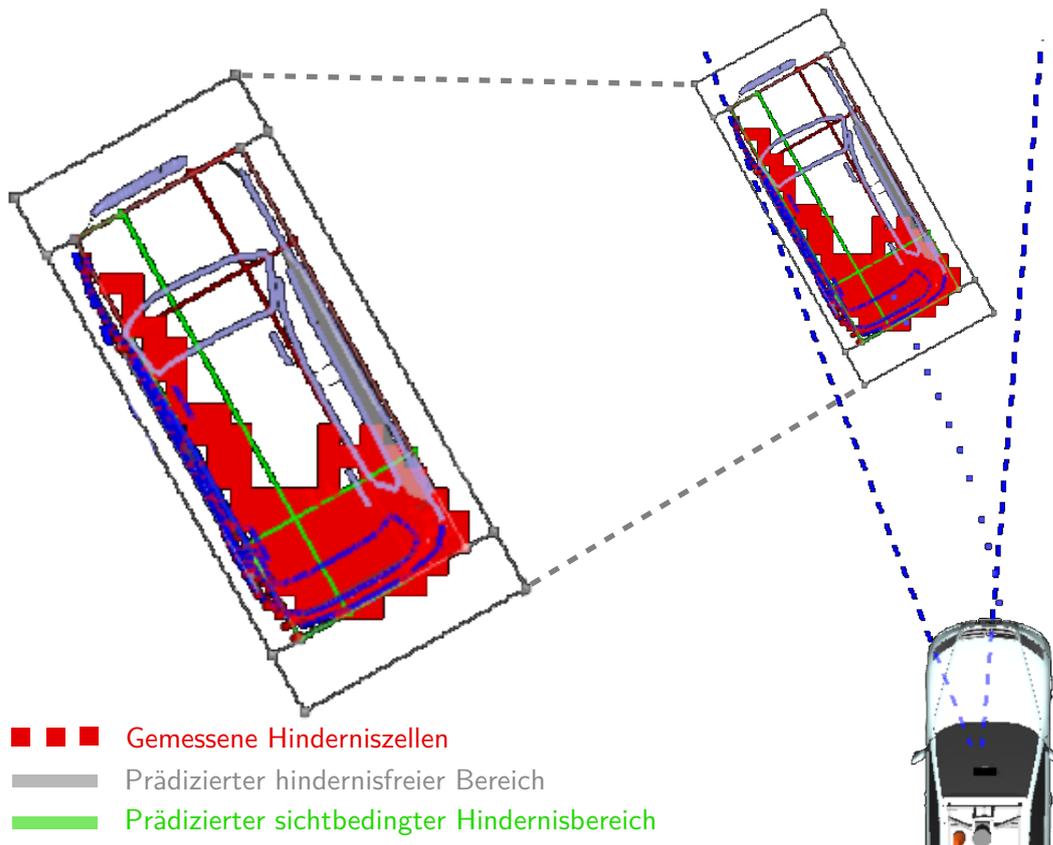
(c) 2,5D-Hinderniskarte

**Abbildung 5.13:** Erstellung einer Hinderniskarte aus Laserdaten.

Nachfolgend wird auf die modellierten tiefendatenbezogenen Fahrzeugmerkmale und deren Auswertung mit der Wahrnehmung eingegangen. Für den Fall der Verwendung eines abstandsgebenden Sensors können zwei Hindernismerkmale für die Gewichtung von Partikeln definiert werden. Das eine Merkmal kennzeichnet die Umgebungszellen, die typischerweise hindernisfrei sind. Dazu zählen z. B. die Randbereiche eines Fahrzeuges. Die Abbildung 5.14 veranschaulicht die durch rote und weiße Linien gekennzeichneten erwarteten hindernisfreien Bereiche. Das zweite Merkmal definiert die belegten Zellen, welche ebenfalls vom Sichtbereich des Sensors abhängig sind. Steht beispielsweise das Führungsfahrzeug unmittelbar und ohne Rotation vor dem autonomen Fahrzeug, erfasst ein LiDAR-Sensor aufgrund des Sichtbereiches lediglich das Heck des Führungsfahrzeuges und nicht die Fahrzeugseiten. In diesem Fall sollte daher der prädizierte Hindernisbereich nicht auf der gesamten Fläche des Führungsfahrzeuges, sondern lediglich am Heck sein. In der Abbildung 5.14 ist der prädizierte sensorbedingt-sichtbare Hindernisbereich eines Partikels durch grüne Linien dargestellt. Das Partikel besitzt eine Fahrzeughypothese mit einem deutlichen Gierwinkel, sodass in dieser Situation eine sichtbare Seitenansicht vorausgesagt wird. Die gemessenen Hinderniszellen sind durch rot-ausgefüllte Quadrate dargestellt und zeigen, dass der prädizierte Hindernisbereich mit dem wahrgenommenen Hindernisbereich gut übereinstimmt.

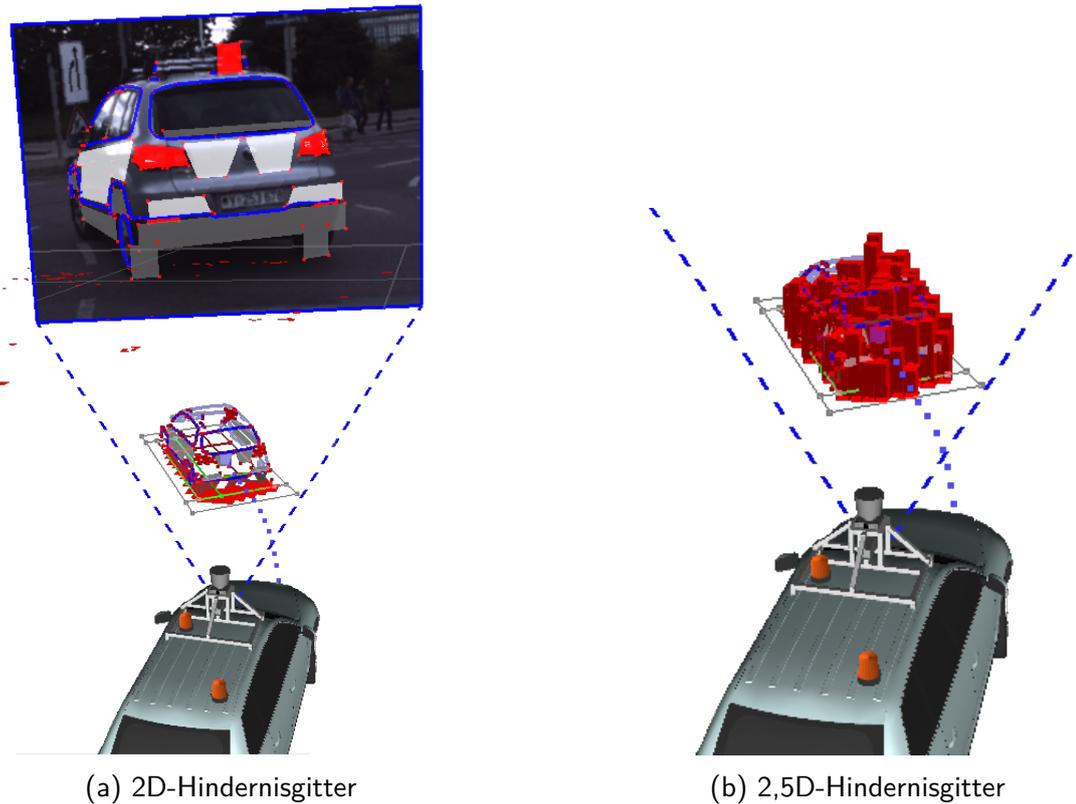
Die Auswertung zwischen den prädizierten und wahrgenommenen Hinderniszellen kann sowohl im 2D- als auch im 2,5D-Hindernisgitter durchgeführt werden. In Manz [2013] erfolgt dieser Merkmalsvergleich im 2D-Raum, indem bei der Erzeugung eines Hindernisgitters eine Zelle als belegt klassifiziert wird, wenn eine Mindesthöhe von 20 Zentimeter vorliegt. In dieser Arbeit werden die Hindernisse zusätzlich mit der zugehörigen Fahrzeughöhe prädiziert und ausgewertet. Dies ist geringfügig zeitaufwendiger, jedoch erhöht sich die Merkmalsdimension und damit die Genauigkeit der Fahrzeugerkennung. Die Abbildung 5.15 stellt die Merkmalsauswertung sowohl im 2D- als auch im 2,5D-Hindernisgitter grafisch dar.

Zusammenfassend erfolgt die Gewichtung der Partikel unter Berücksichtigung von drei Hinderniskriterien. Die Untergrundzellen einer Fahrzeughypothese müssen je nach sensorielle Betrachtung belegt sein. Dabei wird die Hindernishöhe ebenfalls berücksichtigt und ist damit ein weiteres Bewertungskriterium. Das dritte Kriterium bewertet die um das Führungsfahrzeug liegenden Zellen, welche hindernisfrei sein sollten.



**Abbildung 5.14:**

Auswertung von hindernisbehafteten und hindernisfreien Zellen im 2D-Hindernisgitter inkl. Sichtbereichsanalyse des LiDAR-Sensors: Grüne Linien kennzeichnen den Bereich, der hinsichtlich des Sensorsichtbereiches hindernisbehaftet sein sollte. Rote Linien visualisieren den für den Sensor nicht sichtbaren Fahrzeugbereich. Weiße Linien kennzeichnen den nahen Umgebungsbereich, der während einer Fahrt frei sein sollte.

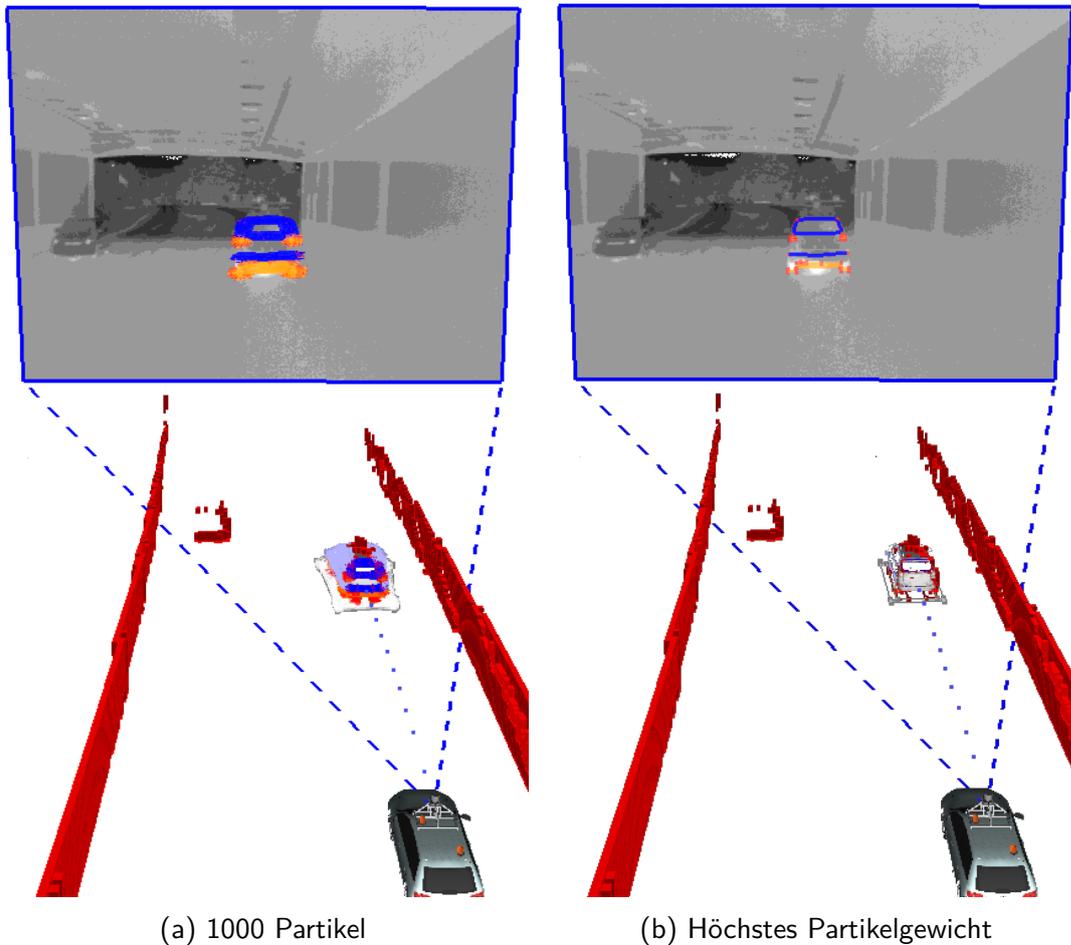
**Abbildung 5.15:**

Merkmalsauswertung im 2D- und 2,5D-Hindernisgitter inkl. Sichtbereichsanalyse des LiDAR-Sensors. (a) Visualisierung der Merkmalsauswertung eines Partikels im 2D-Hindernisgitter. (b) Visualisierung der Merkmalsauswertung eines Partikels im 2,5D-Hindernisgitter mit zusätzlicher Berücksichtigung der Fahrzeughöhe.

Bis zu dieser Stelle wurden die Extraktion, der Vergleich und die Gewichtung von Merkmalen einer partikelzugehörigen ego-relativen Fahrzeughypothesen beschrieben. Das Gesamtgewicht einer Fahrzeughypothese bzw. eines Partikels ergibt sich aus den einzelnen Gewichtungen von jedem Merkmal. Dabei werden die Einzelgewichte miteinander multipliziert, sodass eine Abhängigkeit zwischen den Merkmalen entsteht. Damit ein einzelnes Gewicht das Gesamtgewicht nicht zu Null beeinflussen kann wird jede Merkmalsgewichtung unter  $10^{-10}$  auf  $10^{-10}$  vergrößert.

Der abschließende Schritt der Filterinnovation ist die Neuverteilung der Partikel anhand der berechneten Gewichte. In diesem Vorgang ist das Ziel, die Partikel mit einem hohen Gewicht zu vermehren und die Partikel mit einem niedrigen Gewicht zu verwerfen. Dies hat zur Folge, dass sich die Fahrzeughypothesen an die wahren beobachteten Zustände des Führungsfahrzeuges annähern. Für die Neuverteilung existiert eine Vielzahl an Strategien. In dieser Arbeit bleibt die Anzahl der Partikel konstant und die Neuverteilung erfolgt nach der Strategie „Sequential Importance Resampling (SIR)“ (Näheres siehe Grundlagenkapitel 3.2.2.3).

Die Abbildung 5.16 veranschaulicht den Prädiktionsschritt nach einigen Filterzyklen. Es ist gut zu erkennen, dass sich der rekursive Schätzalgorithmus, bestehend aus der Prädikation der WDF mit anschließender Innovation durch eine Gewichtung und Neuverteilung der Partikel, an den wahren Fahrzeugzuständen annähert. Zur Veranschaulichung zeigt die Abbildung 5.16b noch separat das Partikel mit dem höchsten Gewicht.



**Abbildung 5.16:**

Prädiktionsschritt des Partikel-Filters inkl. der Projektion sämtlicher Fahrzeughypothesen auf die Wärmebildebene: (a) Alle Partikel bzw. ego-relativen Fahrzeughypothesen. (b) Partikel mit höchstem Partikelgewicht. Der obere Teil der Grafik veranschaulicht die Wärmebildebene. Der untere Grafikbereich zeigt die 3D-Welt inkl. eines 2,5D Hindernisgitters (Rote Balken).

### 5.1.3.3 Filterergebnis zur Fahrzeugführung

Das spurtreue Folgen eines Führungsfahrzeuges in einem geschwindigkeitsabhängigen Abstand erfordert ein genaues Schätzergebnis der ego-relativen Fahrzeugpose und der Objektgeschwindigkeit. In diesem Abschnitt wird das Filterergebnis der Objekterkennung erläutert, welches an die Fahrzeugführung aus Kapitel 5.4 kommuniziert wird.

Bei einer Objekterkennung auf Basis eines Partikel-Filters sollte nicht das beste Partikel und damit das Partikel mit dem höchsten Gewicht zum spurtreuen Folgen verwendet werden. Zahlreiche Praxisversuche dieser Arbeit zeigten, dass das beste Partikel von einem Moment auf den Anderen *springen* kann. In solchen Fällen führen die schlagartig wechselnden Zustandswerte zum unsaubereren Fahren des autonomen Folgefahrzeuges. Infolgedessen muss diese Messunsicherheit in den übergeordneten Schichten (Spurgenerierung, Längs- und Querregelung) geglättet werden. Eine bessere Herangehensweise evaluierte Manz [2013], welcher ebenfalls – wie in dieser Arbeit – mit einem Partikel-Filter die ego-relative Fahrzeugpose misst und diese zusammen mit der Längsgeschwindigkeit des Führungsfahrzeuges über die Zeit schätzt. Ein verbessertes übergeordnetes Filterergebnis ergab sich durch eine Mittelwertberechnung der 50% am höchsten gewichteten Partikel. Diese Mittelwertbildung führte zu stark geglätteten Eingangswerten für die Spurgenerierung, ermöglichte gerade deshalb eine ruckfreiere Quer- und Längsregelung und damit ein ruhigeres Folgeverhalten.

Untersuchungen dieser Arbeit ergaben, dass eine Mittelwertbildung über ca. 10% der am höchsten gewichteten Partikel zu einer Verbesserung der Schätzgenauigkeit führt [Fries et al., 2013]. Allerdings wurden noch bessere Ergebnisse in Fries und Wuensche [2015] erzielt, indem das Partikel mit dem höchsten Gewicht als Messwert für einen Scaled Unscented Kalman-Filter (vgl. Kapitel 3.2.1.4) verwendet wurde. Das Vorgehen ein Filterergebnis als Messwert in einen Kalman-Filter zu verwenden widerspricht zwar der Annahme vom Kalman-Filter über ein mittelwertfreies weißes Messrauschen, jedoch besagt dies lediglich, dass der Filter nicht „optimal“ filtert. In der Folge zeigten hunderte von Testkilometern ein spurtreueres Schätzergebnis als beim Folgen mit einem mittelwert-geglätteten Schätzergebnis. Eine ausführliche Evaluierung bezüglich dem Gesamtsystem sowie dessen Schätzgenauigkeit folgt im Kapitel 6.

### 5.1.3.4 Kompatibilität mit mehreren Kameras

In dieser Arbeit wurden zwei Vorgehen zur parallelen Nutzung von mehreren Kameras im Partikel-Filter erprobt. Die zu empfehlende Methode ist die Verwendung eines einzelnen Partikel-Filters für alle Kameras. In diesem Fall sind alle Fahrzeughypothesen in allen Kamerabildern parallel auszuwerten. Dieses Vorgehen ist allerdings nur bei einer sehr genauen intrinsischen und extrinsischen Kamerakalibrierung durchführbar. In den meisten Fällen ist jedoch die extrinsische Kalibrierung zu ungenau, sodass ein

in die Kamerabilder projiziertes Merkmal in der einen Kamera mit der Wahrnehmung übereinstimmt und in der anderen Kamera dies nicht der Fall ist. Beispielsweise wirkt sich schon ein Winkelfehler von einem halben Grad dermaßen stark auf die Merkmalsauswertung aus, dass dieser Ansatz unbrauchbar wird. Bei einer exakten Kamerakalibrierung führt diese Methode zu einer sehr robusten Objekterkennung. Denn es kann zum einen der visuelle Sichtbereich durch mehrere unterschiedlich positionierte Kameras erweitert werden. Zum anderen können unterschiedliche Kamertypen, wie beispielsweise Wärme-, Low-Light- und Tageslichtkameras, oder Farbkameras mit jeweils einem Weitwinkel- und einem Teleobjektiv kombiniert werden. Diese Kombination erweitert die Merkmalsauswertung deutlich, wodurch die Fahrzeugerkennung robuster und genauer wird.

Eine alternative Methode bei der Verwendung von eher ungenau zueinander kalibrierten Kameras ist die Nutzung von jeweils einem Partikel-Filter pro Kamera. In diesem Fall sind die jeweils zu einer Kamera zugehörigen Fahrzeugmerkmale auf die Bildebene zu projizieren und auszuwerten. Anschließend schätzt der Partikel-Filter anhand der Merkmalsübereinstimmung die ego-relative Pose des Führungsfahrzeuges und die Objektgeschwindigkeit. Das übergeordnete Filterergebnis von jedem Partikel-Filter ist dann z. B. das Partikel mit dem höchsten Gewicht, welches anschließend als Messwert für eine sequentielle Kalman-Filter-Innovation verwendet werden kann. Hierfür wurde im Grundlagenkapitel 3.2.1.3 die sequentielle Kalman-Filter-Innovation beschrieben. Des Weiteren sollte bei dieser Methode der Fehler der extrinsischen Kamerakalibrierung über die Zeit als langsam-veränderliche Zustandsvariable mit geschätzt werden. Der große Nachteil dieser alternativen Methode ist der ansteigende Rechenbedarf durch die vermehrte Anzahl an Partikel-Filtern.

### 5.1.4 Kombinierte Objekterkennung mit Partikel- und Kalman-Filter

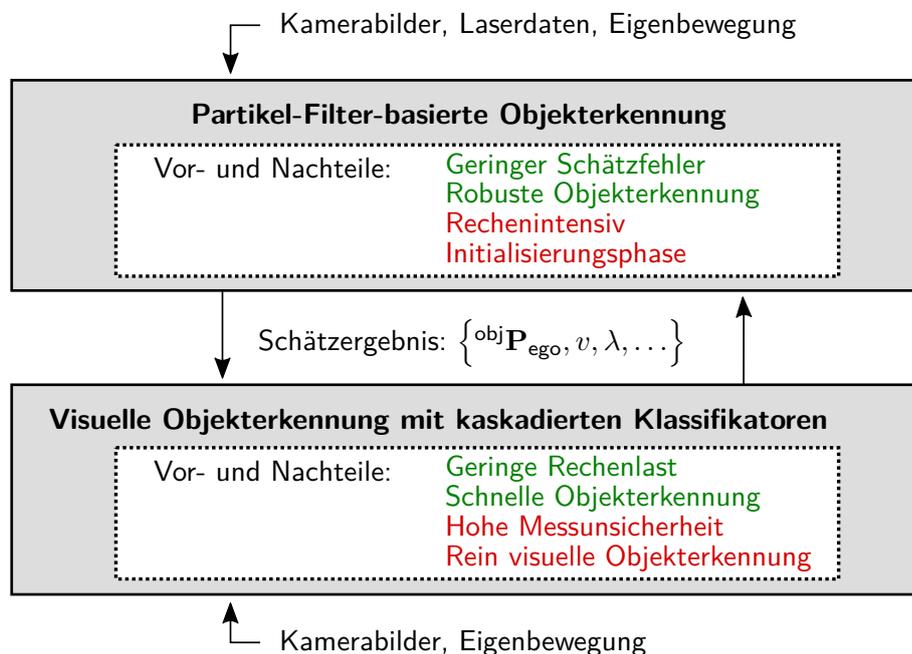
Dieser Abschnitt stellt eine Kombination von zwei unterschiedlichen Verfahren zur Objekterkennung vor, sodass sich diese gegenseitig stützen können.

Die in Kapitel 5.1.2 vorgestellte visuelle Objekterkennung basiert auf kaskadierten Klassifikatoren, welche die einzelnen Fahrzeugseiten schnell und mit wenig Rechenleistung detektieren. Jede erkannte Fahrzeugseite wird zur Bestimmung der ego-relativen Fahrzeugpose und -geschwindigkeit in einen Kalman-Filter sequentiell eingefiltert. Da die Fahrzeugseiten nur grob detektiert werden, ist demzufolge das resultierende Schätzergebnis stark fehlerbehaftet.

Dagegen liefert die im Kapitel 5.1.3 vorgestellte Objekterkennung mit einem rechenaufwendigen Partikel-Filter ein Schätzergebnis mit einer sehr hohen Genauigkeit. Dieser Filter benötigt allerdings bei der Initialisierung beim Konvoistart und bei einer Reinitialisierung (Verlust der Objekterkennung) eine gewisse Zeit zum einschwingen. Hierbei vergehen schnell einige Sekunden bis die Filterkovarianzen klein genug geworden sind, sodass ein zuverlässiges Schätzergebnis vorliegt und dieses zum autonomen

Nachfahren verwendet werden kann. Bei diesem Ansatz kommt es sogar vor, dass der Partikel-Filter gar nicht einschwingt und somit die (Re-)Initialisierung komplett fehlschlägt. Befindet sich das Führungsfahrzeug beispielsweise zu weit weg, ist aufgrund eines zu gering eingestellten Streuungsradius und einer zu geringen Partikelanzahl eine (Re-)Initialisierung selten möglich. In solchen Fällen ist eine grobe Angabe über die Position und Orientierung des Führungsfahrzeug von großem Nutzen, sodass die Partikelmenge in einem kleinen Teilbereich des hochdimensionalen Zustandsraumes gestreut werden kann. Aus diesem Grund wird in diesem Abschnitt die Kombination der zwei erwähnten Verfahren zur Objekterkennung beschrieben und empfohlen. Das Schätzergebnis vom Partikel-Filter wird aufgrund der hohen Schätzgenauigkeit als primäre Objekterkennung betrachtet und wird damit zum autonomen Nachfahren verwendet. Die visuelle Objekterkennung auf Basis kaskadierter Klassifikatoren detektiert das Führungsfahrzeug grob und schnell und ermöglicht dem Partikel-Filter eine schnelle (Re-)Initialisierungsphase. Beide Objekterkennungsverfahren stützen sich sogar beidseitig und fortlaufend, indem die Schätzergebnisse kontinuierlich zueinander kommuniziert werden. Des Weiteren steigert die Verwendung von zwei Objekterkennungsverfahren die Ausfallsicherheit. Im Idealfall laufen die Algorithmen sogar auf unterschiedlichen Rechnern, sodass beim Ausfall eines Rechners die autonome Fahrt nicht unterbrochen werden muss.

Zusammenfassend veranschaulicht die Abbildung 5.17 den in diesem Abschnitt kombinierten Ansatz zur Objekterkennung und listet die Vor- und Nachteile der jeweiligen Verfahren auf.



**Abbildung 5.17:**

Robustheitssteigerung des Gesamtsystems durch die Kopplung von zwei unterschiedlichen Verfahren zur Fahrzeugerkennung.

## 5.2 Aufmerksamkeitssteuerung

Dieser Abschnitt geht auf drei Verfahren zur Aufmerksamkeitssteuerung ein. Das erste Verfahren ermöglicht durch die räumliche Bewegung von Sensoren einen variablen Sichtbereich. Zwei weitere Verfahren beschreiben eine rein softwarebasierte variable Fokussierung auf die Sensordaten und ermöglichen damit eine Verringerung der Verarbeitungszeit von Sensordaten.

### 5.2.1 Aktive Blickrichtungssteuerung

Eine Methode zur Fokussierung auf einen bestimmten Sichtbereich ist die sogenannte Blickrichtungssteuerung. Dabei erfolgt die aktive Blickrichtungssteuerung durch eine mechanische Sensorbewegung mit einem rotatorischen oder translatorischen Aktor. Ein Beispiel ist die bewegbare Kameraplattform namens MarVEye-8, die am Institut für Technik Autonomer Systeme an der Bundeswehr Universität entwickelt wurde und in Unterholzner [2016] ausführlich beschrieben ist. Diese erlaubt einerseits eine Gierwinkelveränderung der gesamten Plattform bis zu  $\pm 120$  Grad (softwareseitige Begrenzung). Des Weiteren ist der Nickwinkel einer Kamera über einen rotierbaren Spiegel veränderbar. Die nachfolgende Abbildung 5.18 veranschaulicht die bewegbare Kameraplattform in zwei Kamerakonstellationen. Abbildung 5.18a zeigt die erste Konstellation bestehend aus 3 Farbkameras. Jeweils zwei Kameras mit einem Weitwinkelobjektiv befinden sich in einer Stereoanordnung. Eine Kamera mit einem Teleobjektiv befindet sich in einer Hohlwelle mit einer Blickrichtung auf einen rotierbaren Spiegel. Die zweite Kamerakonstellation veranschaulicht Abbildung 5.18b, welche um zwei Infrarotkameras in Stereoanordnung ergänzt wurde.

### 5.2.2 Hardwarebasierte AOI einer Kamera

Dieser Abschnitt beschreibt die Fokussierung auf einen Teilbereich eines Kamerachips. Eine Kamera besitzt einen Sensorchip auf dem sich licht-sensitive Zellen befinden. Dabei resultieren die Pixelwerte eines Kamerabildes aus der Lichtmenge, die jeweils auf die einzelnen Zellen trifft. Im Anwendungsfall der visuellen Objekterkennung ist die Auswertung des gesamten Sensorchips bis auf wenige Ausnahmen nicht notwendig. Besteht beispielsweise eine Vermutung, wo sich das zu erkennende Objekt befindet, müssen lediglich die Zellen vom Sensorchip ausgewertet werden, auf denen das Fahrzeug aktuell vermutet wird. Dieser sogenannte hardwarebasierte Interessenbereich, engl. *Area of Interest* (AOI), sollte ein wenig größer gewählt werden, damit dynamische Objekte fortlaufend visuell sichtbar und damit detektierbar bleiben.

Im Nachfolgenden wird die in dieser Arbeit zum Einsatz kommende hardwarebasierte AOI näher betrachtet. Jede Applikation, die eine Wahrnehmung durch-

führt, wie beispielsweise die Fahrzeug-, Weg- oder Kreuzungserkennung, besitzt die Möglichkeit, dem Sensordateneinzug Interessenbereiche mitzuteilen. Hierbei ist ein Interessenbereich genau einer Kamera zugeordnet und wird durch jeweils vier 2D-Bildkoordinaten beschrieben. Anschließend berechnet der Sensordateneinzug die übergeordnete AOI der jeweiligen Sensorchips. Diese ergibt sich aus der konvexen Hülle aller 2D-Bildkoordinaten. Der Vorteil einer hardwarebasierten AOI liegt zum einem in der reduzierten Datenmenge, die über den Netzwerkbus transportiert werden muss. Zum anderen kann der Sensorchip schneller ausgewertet werden, wodurch die Anzahl aufnehmbarer Kamerabilder pro Sekunde ansteigt. Des Weiteren erfolgt die Objekterkennung in kleineren Bildern, sodass die Verarbeitungszeit sinkt und fehlerhafte Detektionen reduziert werden. Da nicht jeder Sensor eine hardwarebasierte AOI unterstützt, wird im nächsten Kapitel auf eine softwarebasierte Lösung zur Datenfokussierung eingegangen.



(a) Kamerakonstellation von MuCAR-3 im Jahre 2013



(b) Kamerakonstellation von MuCAR-3 im Jahre 2016

**Abbildung 5.18:**

Aktive Blickrichtungssteuerung mit der bewegbaren Kameraplattform MarVEye-8.

### 5.2.3 Softwarebasierte AOI einer Kamera

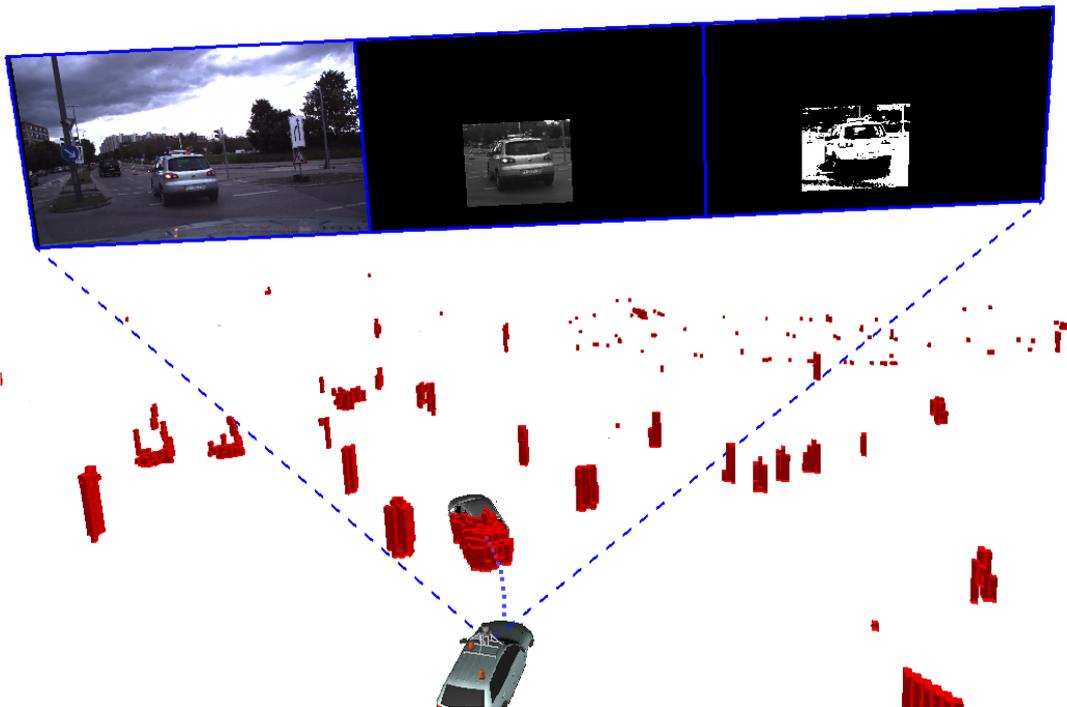
Bei einer softwarebasierten AOI fokussiert die Anwendungssoftware selbst und unabhängig von anderweitigen Applikationen den Datenbereich, der für die Anwendung von Interesse ist. Die Vorteile sind, wie im vorherigen Kapitel genannt, ein verkleinerter Auswertungsbereich, wodurch die Verarbeitungszeit signifikant sinkt und fehlerhafte Detektionen reduziert werden können.

Ein Beispiel für die Aufmerksamkeitssteuerung zeigt die Abbildung 5.19. Dort detektiert die Kalman-Filter-basierte Fahrzeugerkennung aus dem Abschnitt 5.1.2 ein ausgewähltes Führungsfahrzeug mit einer Farbkamera. Der Algorithmus sucht in einem Kamerabild ohne Vorwissen, wo sich das Führungsfahrzeug befindet. Nachdem dieses einmal detektiert wurde, muss fortlaufend nur noch ein Teilbereich des Kamerabildes nach Fahrzeugmerkmalen untersucht werden. Die Abbildung 5.19 visualisiert eine Detektion des Fahrzeughecks durch ein grünes Rechteck und die daraus ermittelte AOI für den nächsten Datenzyklus.



**Abbildung 5.19:** Visuelle Aufmerksamkeitssteuerung im Bereich der Objekterkennung.

Ein weiteres Anwendungsbeispiel ist in der Abbildung 5.20 zu sehen. Dort erfolgt eine LiDAR- und Kamera-basierte Fahrzeugerkennung mit einem Partikel-Filter. Im Kapitel 5.1.3 ist diese Objekterkennung beschrieben, welche u. a. helle und dunkle Flächen aus einem Farbbild extrahieren muss, damit ein Merkmalsabgleich erfolgen kann. Die Abbildung veranschaulicht den reduzierten Bildbereich, der zur Extraktion der hellen und dunklen Flächen untersucht werden muss. In diesem Anwendungsfall wird die AOI durch die konvexe Hülle aller in das Kamerabild projizierten Fahrzeughypothesen (Partikel) berechnet. Denn nur in diesem Bereich muss ein Merkmalsvergleich zwischen der aktuellen Wahrnehmung und den prädizierten Fahrzeugmerkmalen erfolgen.



**Abbildung 5.20:**

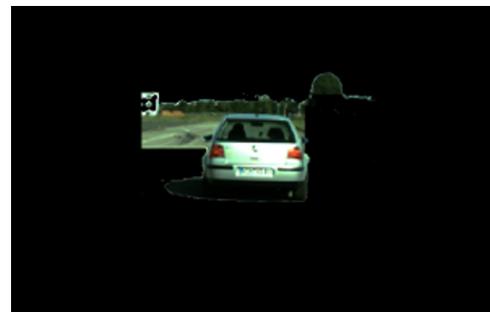
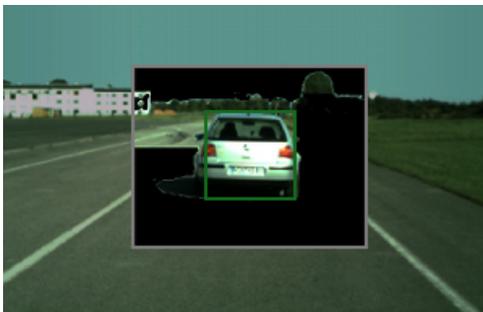
Softwarebasierte AOI: Fokussierung auf das Führungsfahrzeug und damit auf die bildrelevanten Pixel.

Eine weitere Methode im Bereich der softwarebasierten Aufmerksamkeitssteuerung wird als *Cell Flooding* oder *Multi-Dynamic Region Growing* bezeichnet und wurde in Fries et al. [2013] veröffentlicht und in Fries und Wuensche [2014] weiterentwickelt. Der Algorithmus sucht anhand einer groben Fahrzeughypothese die Bildpixel, die nicht zu dem Fahrzeug gehören. Dafür wird die Annahme getroffen, dass die Pixel der Fahrzeugkontur zu den benachbarten Hintergrundpixeln divergent sind.

Die Abbildung 5.21 zeigt ein Ergebnis vom *Cell Flooding*-Algorithmus. Der Algorithmus eliminiert Bildpixel – dargestellt durch geschwärtzte Bildpixel – innerhalb einer vorgegebenen AOI (Graues Rechteck). Die AOI wird dabei in Zellen eingeteilt, in

denen ausgehend vom AOI-Rand in Richtung zum AOI-Zentrum das *Cell Flooding* stattfindet. Dabei werden die Pixelfarbwerte im HSV-Farbraum (siehe Anhang A.3.2) betrachtet und mit den benachbarten Pixelwerten verglichen. Jeder Pixel, der den gleichen Pixelwert besitzt, wird dabei eliminiert bzw. geschwärzt.

Mit der Ausnahme, dass sich die Farbwerte zu stark verändern, finden die Farbwertvergleiche vom äußeren AOI-Rand bis zum -Zentrum statt. Dabei wird das Abbruchkriterium mit dem sinkenden Abstand zum AOI-Zentrum immer stärker verschärft. Des Weiteren beeinflussen benachbarte Zellen das Abbruchkriterium pro Zelle ebenfalls, was ein Überschreiten der Fahrzeugkontur vermeidet. Weiterführend sei auf das Grundlagenkapitel 3.4 verwiesen, in dem der *Region Growing*-Algorithmus sowie dessen Erweiterung zum *Multi-Dynamic Region Growing* beschrieben ist.



- 1.) Fahrzeugdetektion
  - 2.) Eliminierungsfenster  $AOI_{CellFlooding}$
  - 3.) Eliminierte Pixel
- (a) AOI-Pixeeliminierung

(b) Gesamtergebnis

**Abbildung 5.21:**

*Multi-Dynamic Region Growing* zur Aufmerksamkeitssteuerung: (a) Ausgehend von einem Suchfenster erfolgt die Schwärzung von Bildpixeln, die nicht zum Fahrzeug zugehörig sind. (b) Gesamtergebnis.

### 5.3 Berechnung eines Qualitätsmaßes

In diesem Kapitel wird die Berechnung einer Qualitätsgüte für die modellbasierte Fahrzeugerkennung vorgestellt. Eine Qualitätsaussage ist besonders wichtig, wenn weitere Anwendungsmodule, wie z. B. die Trajektorienplanung oder die objektbasierte Sensordatenfusion die Schätzergebnisse der Fahrzeugerkennung verwenden. Diese Module können in Abhängigkeit von der Schätzqualität die jeweiligen Objektinformationen unterschiedlich stark gewichten bzw. in die eigene Schätzung einfließen lassen. Aus Sicherheitsgründen sollte bereits die Fahrzeugerkennung keine Schätzergebnisse mehr an andere Module kommunizieren, wenn die Qualitätsgüte zu gering ist. Andernfalls sind Kollisionen, ein unsauberes Fahrverhalten oder das Folgen eines falschen (Führungs-)Fahrzeuges möglich. Zusätzlich kann der Fahrer sogar gewarnt oder aufgefordert werden, die Fahrzeugführung zu übernehmen.

Dieses Kapitel beschreibt drei Möglichkeiten zur Bestimmung einer Qualitätsgüte:

- Auswertung der Filter-Kovarianzmatrix
- Merkmalsauswertung mit definierten Schwellwerten
- Trainieren eines Klassifikators

#### 5.3.1 Auswertung der Filter-Kovarianzmatrix

Die Schätzfehler(ko)varianzen von einem Partikel- oder Kalman-Filter beschreiben die Unsicherheit einer Zustandsschätzung und können als Qualitätsgüte herangezogen werden. Bei einem Filter mit zehn Zustandsgrößen  $\mathbf{x} = [r, \alpha, h, \phi, \theta, \psi, a, \lambda, v_x, v_y]$  ist die Kovarianzmatrix folgendermaßen aufgebaut:

$$\mathbf{P} = \begin{bmatrix} \sigma_r^2 & \sigma_{r\alpha} & \sigma_{rh} & \sigma_{r\phi} & \sigma_{r\theta} & \sigma_{r\psi} & \sigma_{ra} & \sigma_{r\lambda} & \sigma_{rv_x} & \sigma_{rv_y} \\ \sigma_{\alpha r} & \sigma_\alpha^2 & \sigma_{\alpha h} & \sigma_{\alpha\phi} & \sigma_{\alpha\theta} & \sigma_{\alpha\psi} & \sigma_{\alpha a} & \sigma_{\alpha\lambda} & \sigma_{\alpha v_x} & \sigma_{\alpha v_y} \\ \sigma_{hr} & \sigma_{h\alpha} & \sigma_h^2 & \sigma_{h\phi} & \sigma_{h\theta} & \sigma_{h\psi} & \sigma_{ha} & \sigma_{h\lambda} & \sigma_{hv_x} & \sigma_{hv_y} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots \\ \sigma_{v_y r} & \sigma_{v_y \alpha} & \sigma_{v_y h} & \sigma_{v_y \phi} & \sigma_{v_y \theta} & \sigma_{v_y \psi} & \sigma_{v_y a} & \sigma_{v_y \lambda} & \sigma_{v_y v_x} & \sigma_{v_y}^2 \end{bmatrix} \quad (5.15)$$

Die Qualitätsgüte der Fahrzeugerkennung ist durch die manuelle Festlegung von Schwellwerten bezüglich der Varianzen  $[\sigma_r^2, \sigma_\alpha^2, \sigma_h^2, \sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2, \sigma_a^2, \sigma_\lambda^2, \sigma_{v_x}^2, \sigma_{v_y}^2]$  einschätzbar.

### 5.3.2 Merkmalsauswertung mit definierten Schwellwerten

Eine weitere Möglichkeit ist die Qualitätsgüte anhand der aktuellen Sensordaten (Kamerabilder, Laserdaten) und der geschätzten ego-relativen Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}}$  zu ermitteln. Hierfür werden die Fahrzeugmerkmale

$$\mathbf{m} = [\mathbf{m}_1, \dots, \mathbf{m}_n] = \left[ \begin{array}{c} \overbrace{\mathbf{m}_{\text{rot}}, \mathbf{m}_{\text{dunkel}}, \mathbf{m}_{\text{hell}}}^{\text{Farbflächen}}, \\ \overbrace{\mathbf{m}_{\text{warm}}, \mathbf{m}_{\text{heiß}}}^{\text{Temperaturflächen}}, \\ \overbrace{\mathbf{m}_{\text{Kantenstärke}}, \mathbf{m}_{\text{Kantenrichtung}}, \mathbf{m}_{\text{Kantenabstand}}}^{\text{Kanten}}, \\ \mathbf{m}_{\text{Ecke}}, \\ \mathbf{m}_{\text{Hindernis}}, \mathbf{m}_{\text{hindernisfrei}} \end{array} \right] \quad (5.16)$$

von der geschätzten Fahrzeugpose ausgehend in die Sensordaten projiziert und mit der Wahrnehmung verglichen. Damit ergibt sich ein Qualitätsmaß durch die Anzahl an Merkmalen, die mit der aktuellen Wahrnehmung übereinstimmen. Nachfolgend wird darauf eingegangen, wie festgestellt werden kann, ob ein in die Sensordaten projiziertes Merkmal „gut“ mit dessen Wahrnehmung (den Sensordaten) übereinstimmt.

Der erste Schritt ist die manuelle Festlegung, wie stark jedes Merkmal in der Wahrnehmung vorhanden sein muss. Hierbei ist manuell und einmalig für jedes Merkmal ein Übereinstimmungsschwellwert anhand von aufgezeichneten Kamera- und Laserdaten zu definieren. Begonnen wird z. B. mit dem Kantenmerkmal, bei dem der Abstand und der Richtungsunterschied zwischen Vorhersage und Wahrnehmung sowie dessen Stärke bewertet wird. Der Kantenabstand zwischen Vorhersage und Wahrnehmung in einem Kamerabild sollte so klein wie möglich sein. Der zu definierende Schwellwert ist abhängig von der Bildauflösung bzw. des zum Einsatz kommenden Kamerachips. Die in dieser Arbeit verwendeten Kameras sind im Anhang auf den Seiten 157 - 159 aufgelistet und beschrieben. Die Gleichung 5.17 beschreibt die Bewertung des Kantenabstands anhand eines festgelegten Schwellwertes  $\Delta\mathbf{m}_{\text{Kantenabstand}}$ .

$$p(\mathbf{m}_{\text{Kantenabstand}} | C = \text{Richtig}) = \begin{cases} 1, & \text{wenn } \mathbf{m}_{\text{Kantenabstand}} < \Delta\mathbf{m}_{\text{Kantenabstand}} \\ 0, & \text{sonst} \end{cases} \quad (5.17)$$

Die Qualitätseinschätzung bezüglich der modellbasierten Fahrzeugerkennung und dem Merkmal  $\mathbf{m}_{\text{Kantenabstand}}$  wird als korrekt erkannte Kante ( $C = \text{Richtig}$ ) eingestuft ( $p(\mathbf{m}_{\text{Kantenabstand}}) = 1$ ), unter der Bedingung das der Kantenabstand kleiner als der definierte Schwellwert  $\Delta\mathbf{m}_{\text{Kantenabstand}}$  ist. Die Bewertung der anderen Merkmale erfolgt gleichermaßen mit manuell-festgelegten Schwellwerten. Anschließend sind die als „gut“ klassifizierten Merkmale zu zählen, sodass die Qualität der Fahrzeugerkennung beim Erreichen einer benutzerdefinierten Anzahl an guten Merkmalen als „ausreichend“ eingestuft werden kann. In dieser Arbeit hat sich die Hälfte der Merkmalsanzahl  $n$  als guter Schwellwert herausgestellt:

$$p(\mathbf{m}|C = \text{Richtig}) = \begin{cases} 1, & \text{wenn } \left( \sum_{i=0}^n p(\mathbf{m}_i|C = \text{Richtig}) \right) \geq \frac{n}{2} \\ 0, & \text{sonst} \end{cases} \quad (5.18)$$

### 5.3.3 Trainieren eines Klassifikators

Bei den zwei Methoden aus den vorherigen Abschnitten erfolgt die Qualitätseinstufung der einzelnen Zustandsgrößen bzw. Merkmale binär. Aus diesem Grund klassifizierte Manz [2013] die Qualität der Merkmale mit einer Wahrscheinlichkeit  $p(\cdot)$  in Abhängigkeit zur geschätzten Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}}$ . Dabei werden die Wahrscheinlichkeitswerte anhand eines naiven Bayes-Klassifikators ermittelt.

Ein Bayes-Klassifikator basiert auf der Formel

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)} \quad (5.19)$$

von Bayes und führt damit eine Klassifikation auf Basis der bedingten Wahrscheinlichkeit durch. Hierbei tritt das Ereignis A unter der Voraussetzung, dass das Ereignis B bereits eingetreten ist, ein. Im Bezug auf das Trainieren eines Bayes-Klassifikators steigt die Anzahl der notwendigen Trainingsdaten mit der Dimension des Merkmalsvektor, ansonsten ist die Schätzung der bedingten Wahrscheinlichkeiten  $p(\mathbf{m}|C = \{\text{Richtig, Falsch}\})$  nicht gegeben. Der naive Bayes-Klassifikator nimmt dagegen an, dass die einzelnen Ereignisse (hier Merkmale) statistisch voneinander unabhängig sind und reduziert damit die Komplexität erheblich. Dadurch sind deutlich weniger negative und positive (manuell gelabelte) Sensordaten zum Trainieren eines Klassifikators notwendig.

Für den Trainingsvorgang verwendet der Benutzer – wie im Abschnitt 5.3.2 – aufgezeichnete Sensordatensätze. In diesen Datensätzen speichert der Benutzer die Merkmalsübereinstimmungen für ein paar unterschiedliche Szenen ab, in denen das Führungsfahrzeug visuell gut detektiert wird. Es ergeben sich positive Trainingsdatensätze, in denen z. B. das rote Flächenmerkmal zwei mal eine 60 %-ige und drei mal eine 70 %-ige Übereinstimmung mit der Wahrnehmung aufweist. Die negativen Trainingsdatensätze sind analog zu erstellen. Anschließend kann der naive Bayes-

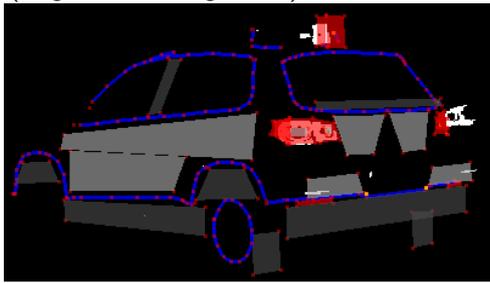
Klassifikator trainiert werden, welcher die optimale Trennung zwischen den positiven und negativen Datensätzen berechnet.

Die Abbildung 5.22a zeigt einen trainierten Klassifikator für das Merkmal „Rote Farbflächen“. Die rote Linie veranschaulicht die selektierten negativen Datensätze und die grüne Linie kennzeichnet die ausgewählten positiven Datensätze. Die orangefarbene Linie entspricht dem resultierenden Klassifikator, der in Abhängigkeit von einem gemessenen prozentualen Anteil an roten Pixeln die zugehörige Merkmalsqualität voraussagt. Mit diesem Vorgehen gelang es Manz [2013], die Qualität der Merkmale zu ermitteln, sodass ein Verlust innerhalb der Fahrzeugerkennung anhand der Merkmale erkannt und demzufolge keine Schätzwerte mehr an die Pfadplanung kommuniziert wurden. Daraufhin kann die Pfadplanung keinen Pfad zum Führungsfahrzeug berechnen und das autonome Folgefahrzeug wurde zu der zuletzt gesehenen Position des Führungsfahrzeuges gefahren. Anschließend stoppte die Konvoifahrt.

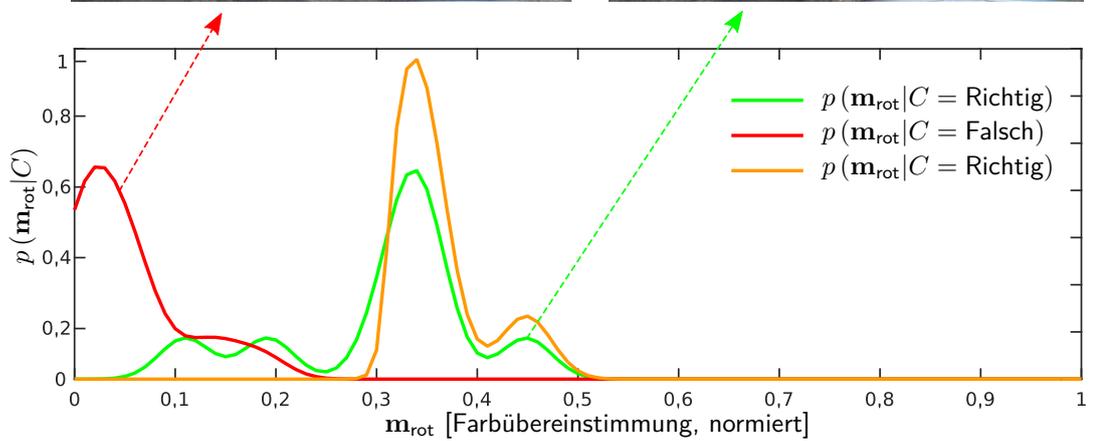
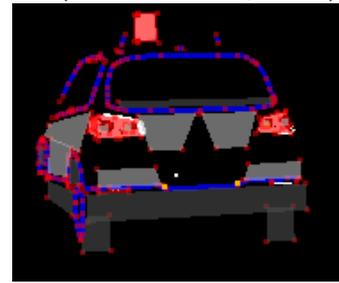
In dieser Arbeit führten diese naiven Bayes-Klassifikatoren in diversen Situationen zu einer fehlerhaften Einschätzung der Merkmalsqualität. Die Abbildung 5.22a zeigt den Grund. Das Maximum der Klassifikatorfunktion befindet sich an der Stelle 0,34. Damit liefert der Klassifikator bei einer Merkmalsübereinstimmung von 34 % die höchste Qualitätseinschätzung von 100 %. Ist die Merkmalsübereinstimmung kleiner als 34 %, dann wird die Qualität kleiner eingeschätzt, was in Ordnung ist. Allerdings sinkt die Qualitätseinschätzung auch bei einer höheren Übereinstimmung des Merkmales mit der Wahrnehmung. Diese fehlerhafte Klassifizierung liegt an den Trainingsdaten. Denn innerhalb der positiven Datensätze existieren keine Messwerte für den Bereich zwischen 0,53 und 1. In dieser Arbeit wird aus diesem Grund das Maximum automatisch detektiert und ausgehend von diesem die Qualitätseinstufung inkrementell erhöht. In der Abbildung 5.22b ist der modifizierte Klassifikator dargestellt.

Eine weiteres Beispiel zeigt die Abbildung 5.23. In dieser sind die positiven und negativen Daten zur Berechnung eines Klassifikators hinsichtlich des Kantenmerkmals  $m_{\text{Kantenabstand}}$  visualisiert. Es ist gut zu erkennen, dass der Klassifikator aus der Abbildung 5.23a eine fehlerhafte Qualitätseinschätzung bei sehr kleinen Kantenabständen besitzt. Der Grund liegt in den positiven Datensätzen, die für das Trainieren des Klassifikators verwendet wurden. In diesen befinden sich keine Datensätze mit sehr geringen Kantenabständen. Einen modifizierten Klassifikator veranschaulicht die Abbildung 5.23b. Dieser besitzt eine inkrementelle Erhöhung der Qualitätsgüte, welche ausgehend vom detektierten Maximum in die Richtung kleiner werdender Kantenabstände erfolgt.

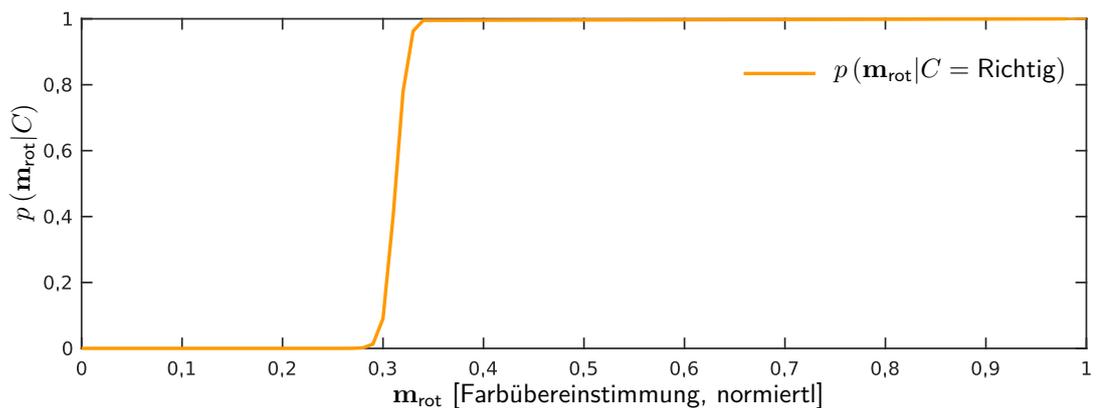
Szene mit falscher Fahrzeugerkennung  
(Negative Trainingsdaten)



Szene mit richtiger Fahrzeugerkennung  
(Positive Trainingsdaten)



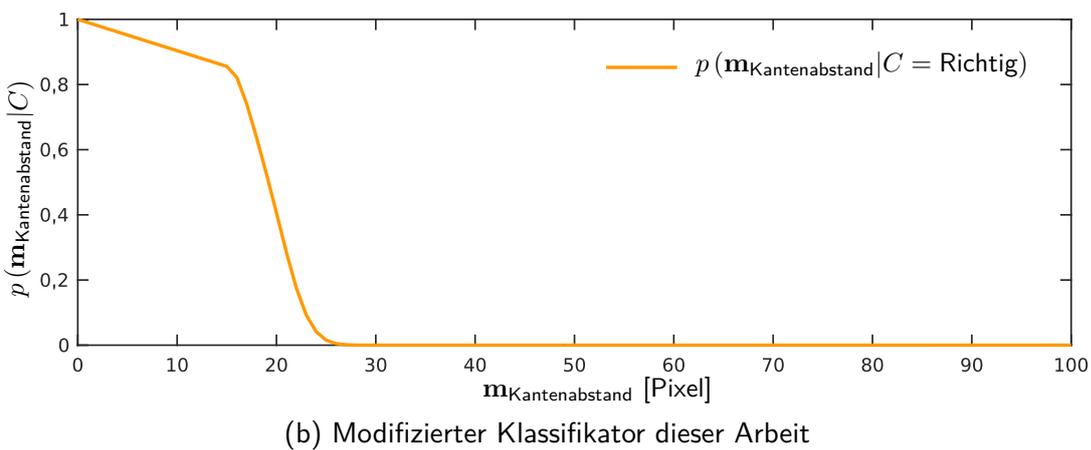
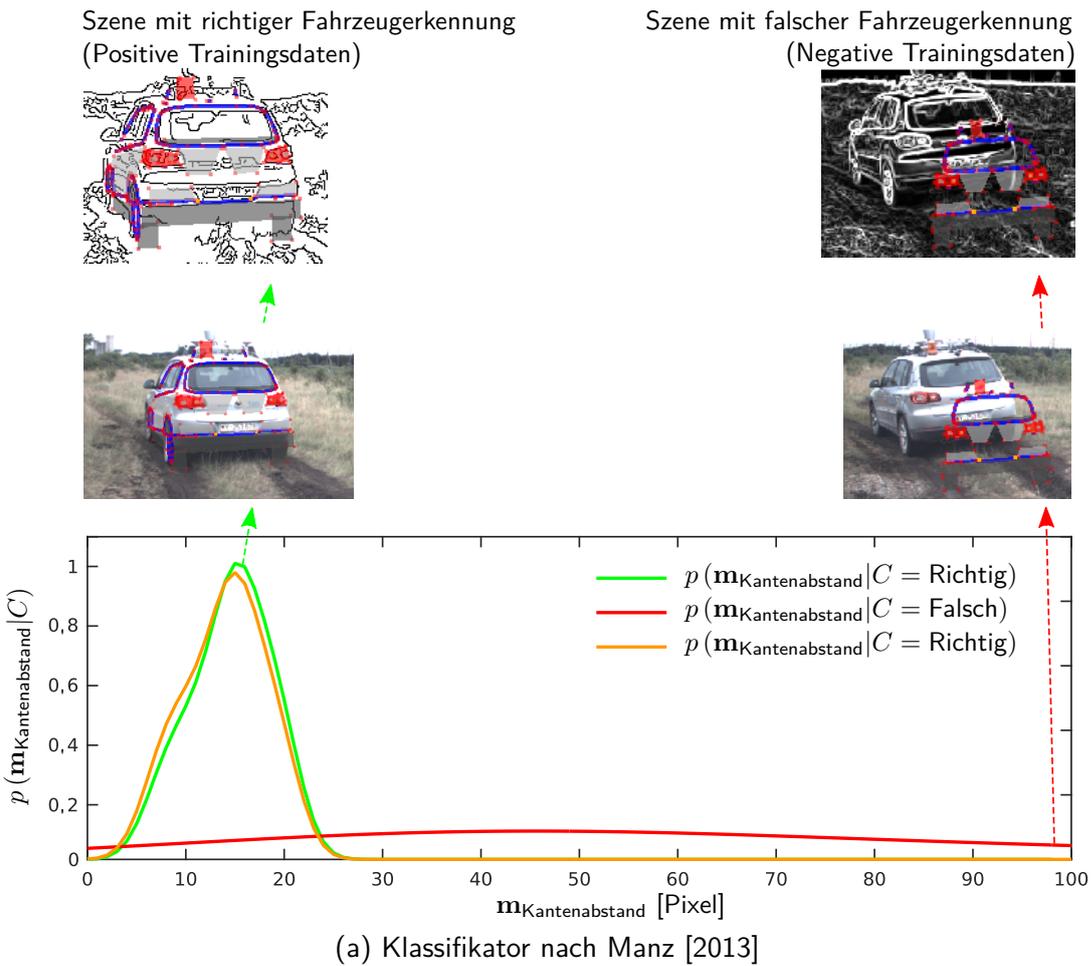
(a) Klassifikatorerzeugung anhand negativer und positiver Trainingsdaten



(b) Resultierender Klassifikator für rote Bildpixel

**Abbildung 5.22:**

Naiver Bayes-Klassifikator für die Qualitätseinschätzung bezüglich der Erkennung von roten Flächenmerkmalen: (a) Auf  $[0 \dots 1]$  normierter, gemessener prozentualer Anteil an Pixeln die Rot sein sollten. Die rote Linie kennzeichnet den Rotanteil bei schlechter Qualität (manuell definiert). Die grüne Linie entspricht dem gemessenen Rotanteil bei guter Qualität. Die orangefarbene Linie ist der daraus resultierende Klassifikator nach [Manz, 2013]. (b) Modifizierter Klassifikator dieser Arbeit.



**Abbildung 5.23:** Naiver Bayes-Klassifikator für die Qualitätseinschätzung über die Wahrnehmung von Kantenabständen. Die rote und grüne Linie kennzeichnen die negativen und positiven Datensätze des Merkmales. Aus diesen wurde der Klassifikator (orangefarbene Linie) zur Qualitätseinschätzung berechnet.

Die Qualitätseinschätzung mit einem modifizierten naiven Bayes-Klassifikator bewährte sich in zahlreichen Praxisversuchen dieser Arbeit. Bei einer fehlerhaften Objekterkennung muss ein Sicherheitsfahrer nicht mehr in das autonom fahrende Fahrzeug eingreifen. Stattdessen wird die fehlerhafte Objekterkennung mit Klassifikatoren erkannt und aktiv darauf reagiert. Das ermöglicht zum einen das automatische auditive Informieren der Fahrzeuginsassen, dass das Führungsfahrzeug nicht mehr zuverlässig detektiert wird. Zum anderen fährt das autonome Fahrzeug selbstständig zur letzten bekannten Position vom Führungsfahrzeug und führt dort eine sogenannte Reinitialisierung durch. Die Reinitialisierung deaktiviert einerseits sämtliche softwarebasierten und hardwarebasierten AOI, sodass die Software das Führungsfahrzeug im gesamten Sensorbereich suchen kann. Andererseits wird durch die Bewegung von Sensoren der Abdeckungsbereich der Sensoren vergrößert. Hierfür empfiehlt sich der Einsatz einer bewegbaren Kameraplattform.

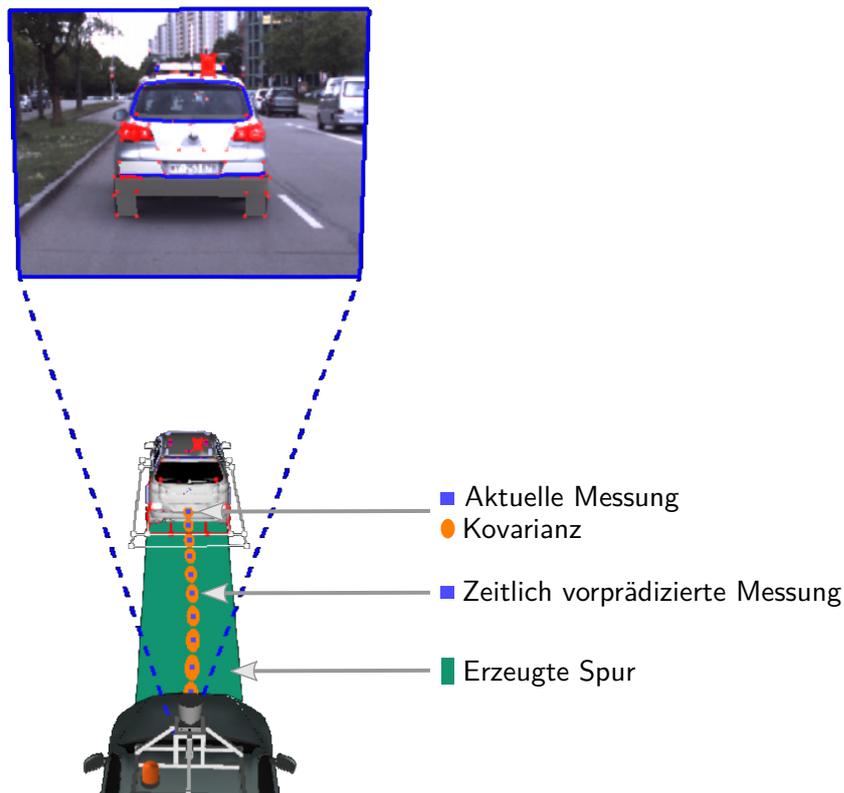
Anmerkend sei erwähnt, dass beim vorgestellten naiven Bayes-Klassifikator die Merkmale unabhängig voneinander bewertet werden. Die übergeordnete Qualitätseinschätzung über die Fahrzeugerkennung ergibt sich durch die Anzahl der Merkmale, die als „gut“ klassifiziert wurden. Die sogenannte Relevance Vector Machine (RVM) [Tipping, 2001] setzt die Auswertung der Merkmale in Relation zueinander, was zu einer besseren Qualitätseinschätzung führt. Als Grundlage dienen, wie beim naiven Bayes-Klassifikator, negative und positive Datensätze. Allerdings ist das Ergebnis nicht jeweils ein Klassifikator pro Merkmal, sondern ein einzelner Klassifikator für alle Merkmale. Der RVM-Klassifikator führt damit die übergeordnete Qualitätseinschätzung über die aktuelle Fahrzeugerkennung direkt durch. Im Vergleich zum naiven Bayes-Klassifikator liefert der RVM-Klassifikator eine zuverlässigere Qualitätseinschätzung. Besonders schwierig sind jedoch Situationen, in denen der RVM-Klassifikator eine fehlerhafte Qualitätsaussage voraussagt. Eine fehlerhafte Qualitätseinschätzung tritt während eines Praxisversuches auf, wenn die Klassifikatoreingabedaten nicht von den Trainingsdatensätzen abgedeckt wurden. In der allgemeinen Vorgehensweise bei einer falschen Klassifikation würde ein Benutzer die fehlerhafte Klassifikation überprüfen und den Klassifikator mit angepassten Trainingsdaten erneut trainieren. Allerdings kann der Anwender bei einer RVM nur schwer nachvollziehen, welche Merkmale das Ergebnis vom Klassifikator beeinflusst haben. Dagegen erkennt der Anwender bei vorherigen Methode, dass beispielsweise das „Rote Flächen“-Merkmal weniger rote Pixel aufweist als mindestens gefordert sind. Aus diesem Grund wurde der RVM-Klassifikator nicht eingesetzt.

## 5.4 Fahrzeugführung

Die Fahrzeugführung beschäftigt sich mit dem spurtreuen Nachfahren eines ausgewählten Führungsfahrzeuges und ist damit ein wesentlicher Bestandteil des autonom-fahrenden Folgefahrzeuges. Sie gliedert sich in die Berechnung einer fahrbaren Spur zum spurtreuen Folgen und der darauf aufbauenden Quer- und Längsregelung.

### 5.4.1 Spurgenerierung

Die in diesem Kapitel vorgestellte Spurgenerierung basiert auf den Arbeiten von Mueller et al. [2009] und wurde geringfügig abgeändert. Die Grundlage für die Berechnung einer Spur zum Führungsfahrzeug sind die Schätzergebnisse der Fahrzeugerkennung. Die vom Schätzergebnis berücksichtigten Daten sind die kartesische ego-relative 2D-Hinterachspolition des Führungsfahrzeuges, dessen Gierwinkel und Längsgeschwindigkeit sowie die dazugehörigen Kovarianzen. Die Abbildung 5.24 veranschaulicht die zeitdiskreten Positionsschätzungen und deren Unsicherheiten.



**Abbildung 5.24:**

Spurgenerierung zum Folgen eines Führungsfahrzeuges. Die grün dargestellte Spur wird anhand geschätzter ego-relativer Positionswerte, der geschätzten Geschwindigkeit des Führungsfahrzeuges und der Schätzunsicherheiten berechnet.

Die Positionsdaten sind durch blaue Quadrate gekennzeichnet und werden in Mueller et al. [2009] in jedem Zeitschritt in einem erweiterten Kalman-Filter mit der planaren inversen Eigenbewegung zeitlich vorprädiziert. Die Gleichung 5.20 beschreibt das Vorprädizieren von einem einzelnen ego-relativen Positionswert  $\mathbf{p}_{\text{ego}_k} = [x, y, 1]^T$ , welcher dem diskreten Zeitpunkt  $k$  zugehörig ist.

$$\mathbf{p}_{\text{ego}_{k+1}} = {}^{\text{ego}_{k+1}}\mathbf{H}_{\text{ego}_k} \cdot \mathbf{p}_{\text{ego}_k} \quad (5.20)$$

Hierbei beschreibt  ${}^{\text{ego}_{k+1}}\mathbf{H}_{\text{ego}_k}$  die räumliche Fahrzeugbewegung von dem Zeitpunkt  $k$  zu  $k + 1$  und entspricht einer homogenen Transformationsmatrix:

$$\text{nach } \mathbf{H}_{\text{von}} = \left[ \begin{array}{ccc|c} \mathbf{R}_{3 \times 3} & & & \begin{matrix} -x \\ -y \\ -z \end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (5.21)$$

Des Weiteren sind in der Abbildung 5.24 orangefarbene Ellipsen dargestellt, welche die Schätzkovarianz der Positionsdaten veranschaulichen. Die Berechnung der zu fahrenden Spur erfolgt anhand der zeitlich vorprädizierten Positionsdaten und deren Kovarianzen, indem zwei fahrbare Klothoiden geschätzt oder alternativ direkt in diese Punktsequenz „gefittet“ werden. Dabei dient eine nach vorne gerichtete Klothoide zum Vorwärtsfahren und eine nach hinten verlaufende Klothoide zum Rückwärtsfahren. Hierbei erfolgt das autonome Rückwärtsfahren des Folgefahrzeuges auf der bereits abgefahrenen Punktsequenz, wenn der Stoßstangenabstand zwischen Folge- und Führungsfahrzeug unter ein Meter sinkt und die geschätzte Längsgeschwindigkeit vom Führungsfahrzeug negativ ist. Bei der Berechnung der zwei Klothoiden werden die Positionsdaten je nach deren Kovarianz mehr oder weniger stark einbezogen. Des Weiteren ist die Anzahl der berücksichtigten Positionsdaten ein geschwindigkeitsabhängiger Optimierungsparameter. Je mehr Positionswerte für die Berechnung einer Klothoide verwendet werden, desto stärker entsteht ein Glättungseffekt, da ältere Positionswerte berücksichtigt werden. Nachfolgend wird die in dieser Arbeit verwendete Klothoidengleichung  $c(l)$  aufgeführt, die sich aus einer Anfangs- und Endkrümmung  $c_0$  und  $c_1$  sowie einer Bogenlänge  $l$  zusammensetzt:

$$c(l) = c_0 + c_1 \cdot l \quad (5.22)$$

Weiterführend hat sich herausgestellt, dass das zeitaufwendige Vorprädizieren von  $n$  ego-relativen Positionsdaten  $\left\{ (\mathbf{p}_{\text{ego}})_i = [x_i, y_i, 1]^T \right\}_{i=1, \dots, n}$  nicht in jedem Zeitschritt erfolgen muss. Stattdessen wird jeder neue Positionswert einmalig in das globale UTM-Koordinatensystem  $\mathbf{S}_{\text{utm}}$  abgebildet. Dieses Vorgehen ist effizienter und anhand der bekannten globalen Fahrzeugpose durch eine einzige Matrix-Vektor-Multiplikation durchführbar:

$$\mathbf{p}_{\text{utm}_k} = {}^{\text{utm}}\mathbf{H}_{\text{ego}_k} \cdot \mathbf{p}_{\text{ego}_k} \quad (5.23)$$

**Berücksichtigung von Hindernissen:** Bei der Berechnung einer Fahrspur sollte diese auf Hindernisse überprüft werden. Denn zeitlich vorprädizierte Fahrzeugpositionen können aufgrund von sich bewegenden Verkehrsteilnehmern nicht mehr befahrbar sein. Aus diesem Grund sollte unter Berücksichtigung eines Sicherheitsabstandes jede Fahrzeugposition auf Hindernisfreiheit geprüft werden. Befindet sich ein stationäres oder sich bewegendes Hindernis auf oder in der unmittelbaren Nähe der berechneten Fahrspur, kann entweder vor dem Hindernis mit einem vorgegebenen Sicherheitsabstand angehalten oder ein Pfad zum Umfahren des Hindernisses geplant werden [Mueller und Wuensche, 2012, Hecker et al., 2010].

#### 5.4.2 Quer- und Längsregelung

Der zweite Schritt der Fahrzeugführung ist die Quer- und Längsregelung des autonomen Fahrzeuges auf der erzeugten Spur. Dabei entspricht die Spur einer im vorherigen Abschnitt 5.4.1 beschriebenen Klothoidengleichung, die sich aus einer Bogenlänge, Anfangs- und Endkrümmung zusammensetzt. Zu Beginn ist die Ablage des Ego-Fahrzeuges zu dieser Klothoide zu bestimmen. Die Ablage definiert die Regelgröße für die Querregelung des Ego-Fahrzeuges. Die Längsregelung erfolgt durch die Veränderung des Gas- und Bremspedals und hängt von der eigenen Fahrzeuggeschwindigkeit  $v_{ego}$  und einem benutzerdefinierten Mindestabstand  $d_{min}$  ab. Der Mindestabstand von beispielsweise 10 Metern definiert den einzuregelnden Abstand zwischen der vorderen Stoßstange des Ego-Fahrzeuges und der hinteren Stoßstange des Führungsfahrzeuges bei Stillstand. Aus Sicherheitsgründen ist dieser geschwindigkeitsabhängig und vergrößert sich je schneller das autonome Fahrzeug fährt:

$$d_{soll} = k_1 \cdot |v_{ego}| + d_{min} \quad (5.24)$$

Der Gewichtungsfaktor  $k_1$  aus der Gleichung 5.24 sollte zur Vermeidung von starken Beschleunigungen und Abbremsvorgängen niedrig gewählt werden. Die von Mueller et al. [2009] empfohlene Gewichtung mit  $k_1 = 0,6s$  führte in dieser Arbeit zu kleinen Konvoiabständen und gewährleistete bei starken Bremsmanövern keinen ausreichenden Kollisionsschutz. Ein erhöhter Gewichtungsfaktor von  $0,9s$  lässt den einzuregelnden Abstand hinreichend groß werden und hält bei Geschwindigkeiten im Bereich von 0 bis 40 km/h ebenfalls den allgemein empfohlenen Mindestabstand („Halber Tachometerwert in Meter“) ein:

$$d_{soll} = 0,9 \cdot \overbrace{11,11}^{40 \text{ km/h}} \frac{\text{m}}{\text{s}} \cdot \frac{\text{s}}{1} + 10 \text{ m} = 20 \text{ m} \quad (5.25)$$

Bei Konvoifahrten über 40 km/h wird der allgemein empfohlene Mindestabstand nicht mehr eingehalten. Die resultierenden Abstände müssen klein genug sein, sodass das Führungsfahrzeug nicht aus dem Erfassungsbereich der – im Anhang A.1 beschriebenen – Sensoren herausfährt. Des Weiteren zeigten Praxistests, dass Verkehrsteilnehmer nicht „ermutigt“ werden zwischen zwei Konvoiteilnehmer einzuscheren.

Die Berechnung der Geschwindigkeitsstellgröße beschreibt die Gleichung 5.26, in der  $k_2$  und  $k_3$  zwei vom Benutzer einzustellende Verstärkungsfaktoren sind. Je größer  $k$  gewählt wird, desto stärker wird auf die Sollgröße geregelt. In Mueller et al. [2009] haben sich die Werte  $k_2 = 0,4$  und  $k_3 = 0,3 \frac{1}{s}$  für eine „weiche“ Regelung bewährt. Diese entsprechen überwiegend den ermittelten Verstärkungsfaktoren  $k_2 = 0,3$  und  $k_3 = 0,3 \frac{1}{s}$  dieser Arbeit, welche sich in diversen Praxisversuchen als optimal herausgestellt haben.

$$v_{\text{steuergröße}} = v_{\text{obj}} + k_2 \cdot \overbrace{(v_{\text{obj}} - v_{\text{ego}})}^{\Delta v} + k_3 \cdot \overbrace{(d_{\text{ist}} - d_{\text{soll}})}^{\Delta d} \quad (5.26)$$

Als abschließende Anmerkung zu diesem Kapitel sei erwähnt, dass die Spurgenerierung sowie die Längs- und Querregelung nicht Kernbestand dieser Arbeit sind. Aus diesem Grund wird nicht näher auf diese Themenbereiche eingegangen. Für eine detaillierte Beschreibung sei auf Hecker et al. [2010], Mueller et al. [2009] und Manz [2013] verwiesen.



## 6 Ergebnisse

### Inhalt

---

6.1	Versuchsfahrzeuge . . . . .	117
6.1.1	Selbstfahrende Versuchsfahrzeuge . . . . .	117
6.1.2	Führungsfahrzeuge . . . . .	124
6.2	Modellgenerierung . . . . .	125
6.2.1	Manuelle Modellgenerierung . . . . .	125
6.2.2	Automatische Modellgenerierung . . . . .	130
6.2.3	Manuelle vs. automatische Modellgenerierung . . . . .	130
6.3	Fahrzeugerkennung und -verfolgung . . . . .	133
6.3.1	Fahrzeugerkennung und -verfolgung bei Tag . . . . .	133
6.3.2	Fahrzeugerkennung und -verfolgung bei Nacht . . . . .	139
6.4	Beurteilung des Gesamtsystems . . . . .	146

---

In diesem Kapitel werden in vier Abschnitten die erzielten Ergebnisse der modellbasierten Fahrzeugerkennung zusammengefasst. Der erste Abschnitt stellt die in der Praxis eingesetzten Versuchsfahrzeuge im Detail vor. Im zweiten Abschnitt erfolgt ein Vergleich zwischen einer manuellen und automatischen Erzeugung von fahrzeugspezifischen Merkmalsmodellen. Anschließend folgen die Resultate der modellbasierten Fahrzeugerkennung und dessen Verfolgung sowohl bei Tag als auch bei Nacht. Der letzte Abschnitt beurteilt das Gesamtsystem.

### 6.1 Versuchsfahrzeuge

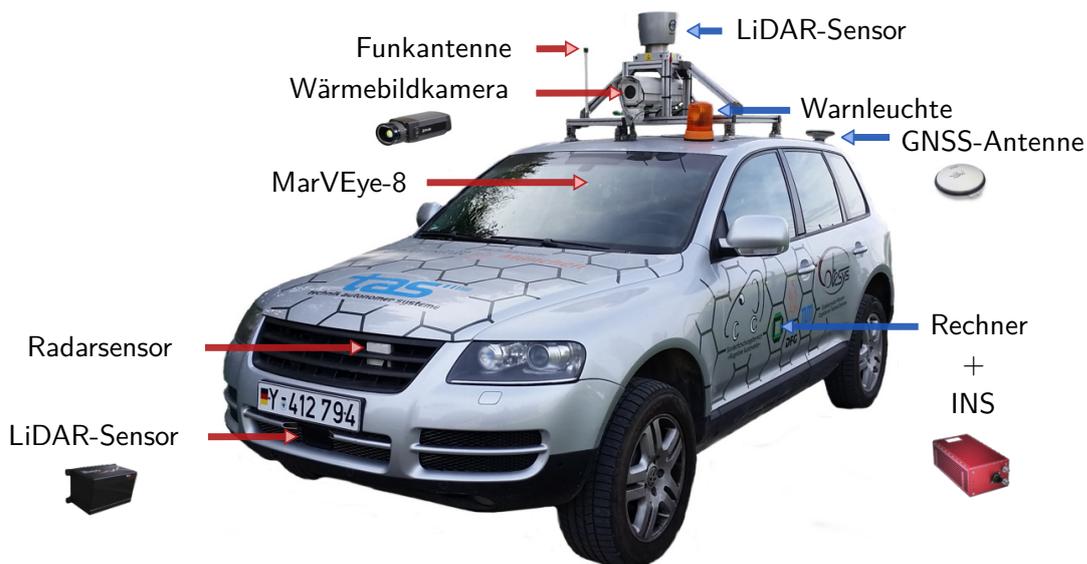
In dieser Arbeit kamen zahlreiche Versuchsfahrzeuge zum Einsatz, die im Nachfolgenden vorgestellt werden. Sie unterscheiden sich in selbstfahrende und manuell-geführte Fahrzeuge.

#### 6.1.1 Selbstfahrende Versuchsfahrzeuge

Damit ein Fahrzeug selbstständig fahren kann, muss dieses entsprechend umgebaut werden. Während dieser Arbeit kamen insgesamt vier Folgefahrzeuge zum Einsatz.

Die Umrüstung für einen autonomen Betrieb und damit mit mechanischen Bauteilen, Elektronik und Sensorik erfolgte – je nach Fahrzeug – durch das TAS-Institut, Diehl BGT Defence und Rheinmetall Landsysteme. Ergänzend wurde ein Fahrzeug zum autonomen Nachtbetrieb mit Sensorik erweitert. Im Nachfolgenden werden diese selbstfahrenden Versuchsfahrzeuge vorgestellt und die mechanischen Bauteile, die Elektronik und die Sensorik betrachtet. Zu beachten sei, dass im Anhang A.1 die verwendeten abstandsgebenden, bildgebenden und inertialen Sensoren gesondert und im Detail beschrieben sind.

**MuCAR-3** steht für Munich Cognitive Autonomous Robot Car 3<sup>rd</sup> Generation und ist nach VaMoRs (Versuchsfahrzeug für autonome Mobilität und Rechnersehen) [Dickmanns, 1989] und VaMP (VaMoRs-Pkw) [Dickmanns et al., 1994, Maurer, 2000] das 3. Versuchsfahrzeug, das an der Universität der Bundeswehr für die Forschung am autonomen Fahren aufgebaut wurde. Es handelt sich hierbei um einen VW Touareg, der seit 2006 die Fähigkeit zum autonomen Folgen im Zweierkonvoi [Himmelsbach et al., 2009b] besitzt. Fortlaufend bis heute wurde dieser Pkw mit Sensorik ergänzt, sodass mit diesem sogar autonome Fahrten bei Nacht durchgeführt werden können [Fries und Wuensche, 2015]. Die Abbildung 6.1 zeigt die am Fahrzeug montierten Sensoren.



**Abbildung 6.1:** MuCAR-3: Umgebauter VW Touareg und dessen Sensorausstattung.

An der Dachträgerkonstruktion ist zum einen ein Velodyne HDL-64E LiDAR-Sensor angebracht. Dieser rotiert mit 10 Hz um die eigene Achse und besitzt damit einen 360 Grad-Sichtbereich. Der vertikale Sichtbereich beträgt mit 64 Laserdioden 26,8 Grad. Zum anderen befindet sich unter dem LiDAR-Sensor eine FLIR A615 Wärmebildkamera, die kalibrierte Temperaturinformationen im Spektralbereich zwischen 7,5 und 13  $\mu\text{m}$  misst. Die Auflösung der Kamera beträgt  $640 \times 480$  Pixel und der eingestellte

Messbereich geht von  $-20$  bis  $150\text{ }^{\circ}\text{C}$  mit einer absoluten Messgenauigkeit von  $\pm 2\text{ }^{\circ}\text{C}$ . Da die Wärmestrahlung nicht durch die Windschutzscheibe messbar ist, muss die Kamera außerhalb des Fahrzeuges montiert werden. Zum Schutz vor äußerlichen Einflüssen wie Regen, Staub und Hitze befindet sich die Kamera in einem Schutzgehäuse mit einem speziellen hitzeabweisenden Sonnendach. An der Gehäusefront wurde seitens des Herstellers ein spezielles Schutzfenster aus Germanium eingesetzt, welches mehr als 90 Prozent der Wärmestrahlung durchlässt. Der horizontale Sichtbereich ist 45 Grad groß. Damit der Sichtbereich nicht durch tier- und wetterbedingte Einflüsse (Fliegen, Regen, Staub) eingeschränkt wird, wurden ein elektronisch ansteuerbarer Scheibenwischer und eine Sprühvorrichtung am Schutzfenster montiert.

Auf dem Fahrzeugdach befinden sich weitere Sensoren und Elektronik. Eine Funkantenne ermöglicht die Kommunikation zwischen Fahrzeugen. Zwei magnetisch befestigte Warnleuchten visualisieren nach außen hin, ob sich das Fahrzeug gerade im autonomen Betrieb befindet. Des Weiteren sind zwei GNSS-Antennen am Fahrzeugdach befestigt. Sie ermöglichen zum einen den Empfang von geostationären Positionsdaten und zum anderen die geostationäre Orientierung bei Stillstand des Fahrzeuges. An der Fahrzeugfront befindet sich unter dem Kennzeichen ein kleiner 8-Zeilen Ibeo Lux LiDAR-Sensor und auf Höhe des Herstellerlogos ein Radarsensor von Continental. Beide Sensoren wurden speziell für den Automobilbereich hergestellt. Im Fahrzeug befindet sich die schwenkbare Kameraplattform MarVEye-8 [Unterholzner und Wuensche, 2010], die sich direkt zwischen der Windschutzscheibe und dem Fahrerrückspiegel befindet. An dieser können bis zu fünf Kameras montiert werden. Während der letzten fünf Jahre wurden diverse Kamerakonstellationen erprobt. Eine aus drei Farbkameras bestehende Konstellation zeigt die Abbildung 6.2.



**Abbildung 6.2:** Fahrer- und Beifahrerraum von MuCAR-3.

Eine CCD Basler ace acA1300-30gc Farbkamera befindet sich in der Hohlwelle mit einer Orientierung zum Fahrzeugboden. Dabei ist die Kamera auf einen ansteuerbaren Spiegel und damit in Fahrtrichtung ausgerichtet. Diese Konstruktion dient zur Stabilisierung des Nickwinkels, sodass bspw. ein Führungsfahrzeug auch visuell im Blick gehalten werden kann, falls das autonome Fahrzeug in ein Schlagloch fahren sollte. Zusätzlich ermöglicht die Kameraplattform einen horizontal einstellbaren Schwenkbereich von  $\pm 120$  Grad (softwareseitige Begrenzung), was den visuellen Sichtbereich deutlich erweitert. Folglich kann der Gier- und Nickwinkel der auf den Spiegel ausgerichteten Kamera variabel verändert werden. Aus diesem Grund wurde diese Kamera mit einem Teleobjektiv bestückt. Das Teleobjektiv hat in horizontaler Richtung einen Sichtbereich von 26 Grad, wodurch Fahrzeuge in  $>100$  m Entfernung noch erkannt werden können. Die zweite Farbkamera ist eine CMOS Basler ace acA2040-25gc und wurde mit einem Weitwinkelobjektiv bestückt. Die Kamera verfügt über  $2046 \times 2046$  Bildpixel und kommt zur Umgebungswahrnehmung zum Einsatz. Die dritte Kamera, die sich an der Kameraplattform befindet, ist eine CMOS Basler ace acA2000-50gc, welche neben dem sichtbaren Spektralbereich auch den infraroten Spektralbereich abdeckt. In diesem Fall kommt ein sogenannter Tageslichtsperrfilter für die Blockierung von Lichtstrahlen aus dem sichtbaren Wellenlängenbereich zum Einsatz. Dies erlaubt eine Bildverarbeitung ausschließlich im infraroten Spektralbereich. Eine vierte Kamera befindet sich neben der Kameraplattform direkt hinter der Windschutzscheibe. Es handelt sich hierbei um eine TAU CNV Low-Light-Farbkamera des Herstellers FLIR, die über einen Dynamikbereich von 85 dB verfügt. Die Kamera wird speziell für den Einsatz bei Dämmerung und Nacht verwendet, da sie nur sehr wenig Licht (Mondschein- oder Laternenlicht) benötigt um ein (verraushtes) Farbbild zu rekonstruieren.

Neben Kamerasensoren ist ein Eigenbewegungssensor für ein autonomes spurtreues Folgen eines Führungsfahrzeuges notwendig. MuCAR-3 besitzt dafür Inkrementalgeber sowie eine INS vom Typ OxtS RT3003. Die INS besteht aus einer IMU, die mit DGPS inkl. Echtzeitkinematik (engl. *Real Time Kinematic* (RTK)) gekoppelt ist. Die GNSS-Daten beziehen sich hierbei auf eine bekannte Referenzstation und besitzen in der Nähe der Referenzstation Positionsgenauigkeiten von circa zwei Zentimetern. Die Abbildung 6.2 zeigt die INS-Einbaulage zwischen den beiden Fahrersitzen und damit in Schwerpunktnähe des Fahrzeuges. Neben der INS befindet sich eine Schalttafel mit der diverse Sensoriken (LiDAR, Radar, Kameras), der Funk, die Waschanlage vom Schutzgehäuse der Wärmebildkamera und der Hauptfahrzeugrechner (Intel Xeon L5640 Dual CPU Hexa Core) ein- und ausgeschaltet werden können. Des Weiteren ist in unmittelbarer Nähe des Fahrer- und Beifahrersitzes ein Notausknopf integriert, mit dem die Autonomiefunktion ausgeschaltet werden kann. Ebenfalls ist durch die Betätigung des Bremspedals die Autonomiefunktion deaktivierbar, da dort ein Drucksensor montiert wurde. In die Lenkung kann ein Sicherheitsfahrer jederzeit eingreifen und den vom Fahrzeugrechner kommandierten Lenkwinkel mit einem leichten Kraftaufwand übersteuern.

**Munich Cognitive Autonomous Robot Car 4<sup>th</sup> Generation (MuCAR-4)** ist ein VW Tiguan, der zusammen mit MuCAR-3 vom Institut für Technik Autonomer Systeme (TAS) der Universität der Bundeswehr betrieben wird. Seit Mitte 2012 besitzt es die Fähigkeit, bei guten Lichtbedingungen einem anderen Fahrzeug selbstständig zu folgen. Diese Funktionalität wurde Ende 2014 für autonome Dämmerungsfahrten erweitert. Das Fahrzeug besitzt einen Velodyne HDL-64E LiDAR-Sensor, eine Funkantenne, zwei GNSS-Antennen, eine Warnleuchte, einen leistungsstarken Fahrzeugrechner, Inkrementalgeber, eine OxTS RT3003 INS und die Kameraplattform MarVEye-8. Die Abbildung 6.3 veranschaulicht diese Komponenten. An der Kameraplattform befinden sich derzeit zwei Basler ace acA1300-30gc Farbkameras in Stereoanordnung. Die Kameras besitzen eine Auflösung von  $1296 \times 964$  Pixeln und liefern 20 Bilder pro Sekunde. Diese Bildzyklusrate ist benutzerdefiniert und einheitlich für die meisten Kameras dieser Arbeit eingestellt. An dieser Stelle sei zusätzlich erwähnt, dass zur Datensynchronisierung die meisten Kameras hardwareseitig getriggert werden.

Die dritte Farbkamera befindet sich wie beim MuCAR-3 in der Hohlwelle der Kameraplattform und ist auf einen elektronisch ansteuerbaren Spiegel ausgerichtet.



**Abbildung 6.3:** MuCAR-4: Umgebauter VW Tiguan und dessen Sensorausstattung.

**TULF** steht für „Technologieträger Unbemanntes Landfahrzeug“ und ist ein Lkw vom Typ RMMV HX58, welcher im Rahmen der gleichnamigen F&T-Studie (vgl. Abschnitt 2.1.2, S. 9) zum autonomen Fahren umgerüstet wurde. Bei den Umrüstungsarbeiten diente MuCAR-3 als Vorbild. So wurden zum einen diverse Basler Tageslichtkameras und eine unkalibrierte Xenics Gobi-640 Wärmebildkamera in das Fahrzeug integriert. Des Weiteren wurden zwei Velodyne HDL-32E LiDAR-Sensoren, die jeweils über 32 Laserdioden verfügen, sowie ein Continental Radarsensor an der Fahrzeugfront montiert. Auf dem Dach befindet sich eine Funkantenne, die nicht zur Fahrzeug-Fahrzeug-Kommunikation verwendet werden soll, sondern zur Kommunikation mit einem Leitstand (Abgebildet in der Grafik 6.4). Im Fahrzeug befinden sich diverse Rechner zur Sensorverarbeitung und Datenanalyse sowie eine Xsens MTi-G-700 INS zur Eigenbewegungsschätzung. Zusätzlich wurde ein Radarbasierter Geschwindigkeitssensor MSO SpeedWedge im Bereich der Fahrzeughinterachse montiert. Dieser ermittelt mit zwei Radar-Sensoren die Übergrundgeschwindigkeit des Fahrzeuges.



**Abbildung 6.4:**

TULF: Sensorausstattung eines umgebauten Lkw vom Typ RMMV HX58. Der Leitstand der sich auf der Ladefläche befindet wird im Einsatz abgeladen und kommuniziert über Funk mit dem Lkw.

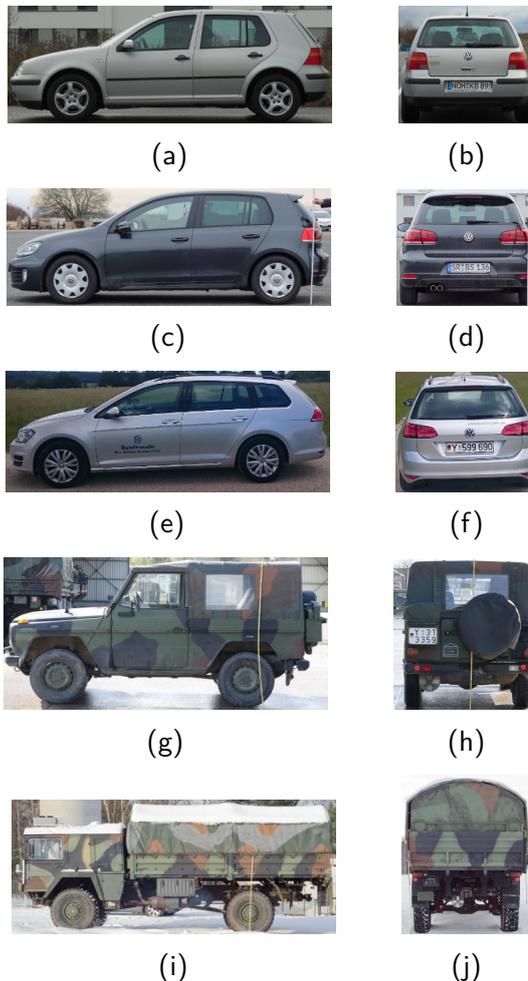
**StrAsRob** steht für „Straßentransport mit Assistenzfunktionen von Robotern“ und ist eine auf TULF aufbauende F&T-Studie (vgl. Abschnitt 2.1.2, S. 9). Der Fokus der Studie liegt in der autonomen Navigation und dem Folgen eines ausgewählten Führungsfahrzeuges mit seriennäherer Sensorik. Aus diesem Grund wurde anstelle der zwei kostspieligen 360 Grad Velodyne Laserscanner ein einzelner 8-Zeilen Ibeo LUX LiDAR-Sensor in das Fahrzeug integriert. Dieser stammt aus dem Automobilbereich und ist preiswerter, besitzt allerdings nur eine Laserdiode, die durch einen Spiegel 8-Ebenen abtastet. Eine weitere Änderung erfolgte bei der Integration eines Radarbasierten Übergrund-Geschwindigkeitssensors. Hier wurde der DICKEY-john Radar III im Bereich der Fahrzeughinterachse montiert, welcher im Vergleich zum MSO SpeedWedge eine höhere Messgenauigkeit aufweist. Zur Verbesserung der Eigenbewegungsschätzung wurde zusätzlich zur Xsens MTi-G-700 INS ein einachsiger faseroptischer Kreisel vom Typ KVH DSP-3000 integriert. Ansonsten sind die Kameras, der Funk, die INS, der an der Fahrzeugvorderseite verbaute Radarsensor und die Rechner die gleichen Elektronikkomponenten wie beim TULF. Im Nachfolgenden zeigt die Abbildung 6.5 die in den Lkw integrierten Komponenten.



**Abbildung 6.5:**  
StrAsRob: Sensorausstattung eines umgebauten Lkw vom Typ RMMV HX58.

### 6.1.2 Führungsfahrzeuge

Als Führungsfahrzeuge kamen eine Vielzahl von unterschiedlichen Fahrzeugen zum Einsatz. Dazu zählen zum einen die im vorherigen Abschnitt vorgestellten selbstfahrenden Versuchsfahrzeuge. Zum anderen dienten weitere zivile und militärische manuell gefahrene Fahrzeuge als Führungsfahrzeug. Dazu zählen ein grauer VW Golf 4, ein schwarzer VW Golf 6, ein silberner VW Golf Variant, ein militärischer Wolf gl (Mercedes-Benz G-Klasse) mit Tarnmuster und ein MAN mil gl Lkw - ebenfalls mit Tarnmuster. Die Abkürzung „mil gl“ steht für „militarisiert geländegängig“. Der MAN mil gl ist ein zweiachsiges Allradfahrzeug mit einer maximalen Nutzlast von fünf Tonnen und einer Bodenfreiheit von 40 Zentimetern. Das bis zu 80 km/h schnelle Transportfahrzeug ist 8 Meter lang und 2,5 Meter breit. Die einstellbare Höhe betrug in den Praxistests überwiegend 3,5 Meter. Die Abbildung 6.6 zeigt fünf Führungsfahrzeuge, denen auf Basis von Modellwissen autonom nachgefahren werden kann.



**Abbildung 6.6:**

Einige Führungsfahrzeuge die in dieser Arbeit zum Einsatz kamen: (a-b) VW Golf 4. (c-d) VW Golf 6. (e-f) VW Golf Variant. (g-h) Wolf gl mit Tarnmuster. (i-j) MAN mil gl 5t.

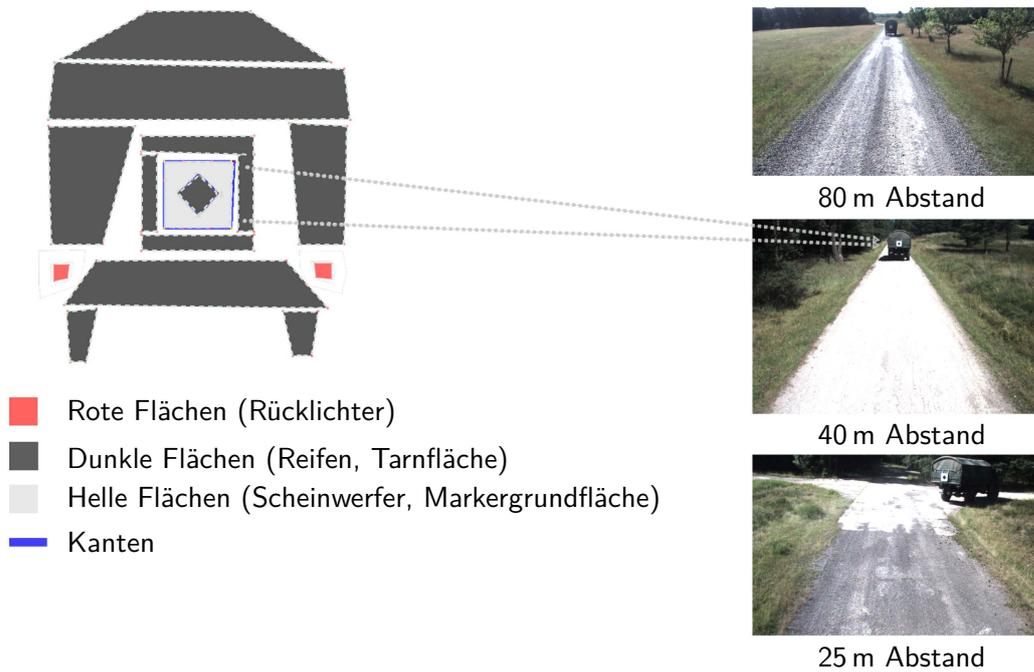
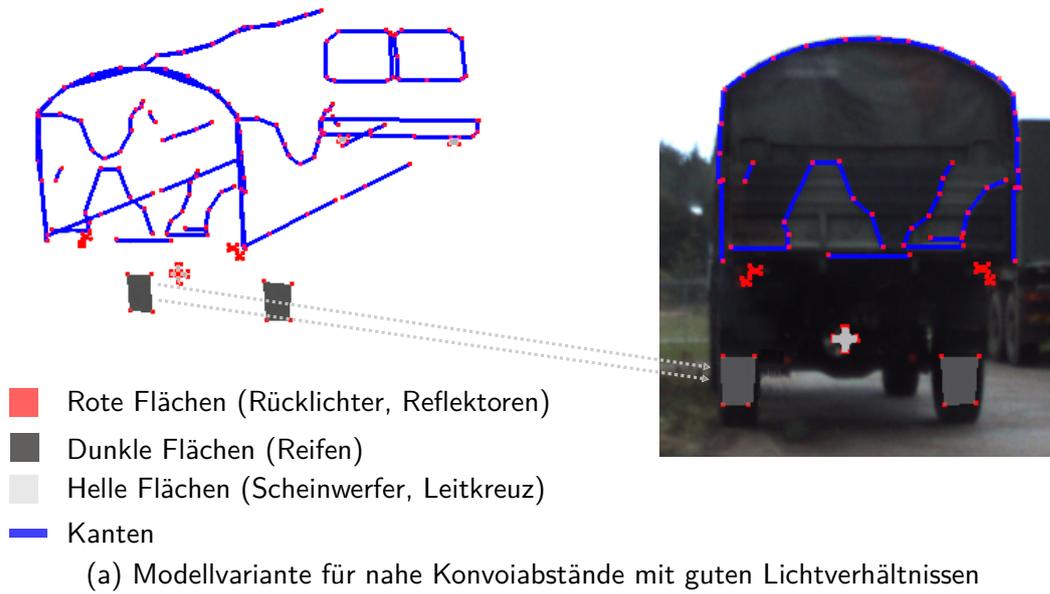
## 6.2 Modellgenerierung

Im Kapitel 4 wurden Algorithmen zur manuellen sowie automatischen Erstellung von fahrzeugspezifischen Merkmalsmodellen beschrieben. Dieses Kapitel stellt die Ergebnisse der manuellen und automatischen Modellgenerierung vor und vergleicht diese miteinander.

### 6.2.1 Manuelle Modellgenerierung

In dieser Arbeit sind zahlreiche Merkmalsmodelle manuell erstellt und fortlaufend erweitert worden. Im Kapitel 4 wurde bereits ein Modell eines VW Golf 4 und eines VW Tiguan vorgestellt. In diesem Abschnitt werden zwei weitere Fahrzeugtypen im Detail vorgestellt und auf die gesammelte Praxiserfahrung während der Modellerzeugung sowie -erweiterung eingegangen.

Angefangen wird mit den sensorabhängigen Merkmalsmodellen für einen militärischen Lkw. Es handelt sich hierbei um ein MAN mil gl Transportfahrzeug mit einem Flecktarnmuster. Die ersten zwei Modelle die vorgestellt werden sind mit einer Farbkamera nutzbar. Die Abbildung 6.7 zeigt diese mitsamt der modellierten Merkmale. Im Nachfolgenden wird das in der Abbildung 6.7a dargestellte Modell näher betrachtet. In diesem sind die an Intensitätsübergängen zu findenden markanten Kanten modelliert. Praxistests zeigten, dass die Kanten im Nahbereich und bei guten Lichtverhältnissen gut erkennbar bleiben. Erst bei Konvoiabständen über 30 Metern und bei schlechten Lichtverhältnissen (bspw.: Walddurchfahrt bei Sonnenschein) war eine zuverlässige Detektion der Kanten nicht mehr möglich. Aus diesem Grund musste das Modell um andere fahrzeugspezifische Merkmale erweitert werden. Dazu zählen zum einen zwei kleine rote Rücklichter, die im ausgeschalteten Zustand mit Bildverarbeitungsalgorithmen zu erkennen sind. Dabei ist der ausgeschaltete Zustand eine militärische Vorgabe zur Realisierung eines einsatznahen Systems. Praxistests zeigten, dass die ausgeschalteten Rücklichter mit den Farbkameras vom TULF sowie vom StrAsRob bis zu einem Konvoiabstand von 30 Metern erkannt werden können. Bei höheren Abständen sinkt die rot-segmentierte Fläche in den Kamerabildern auf unter einen Quadratpixel. Neben roten Flächen sind helle und dunkle Flächen weitere modellierbare Merkmale. Der MAN mil gl besitzt ein kleines weißes Leitkreuz, welches ebenfalls, wie die roten Scheinwerfer, nicht bei großen Konvoiabständen detektierbar ist. Allerdings hat sich das Merkmal für den Nahbereich als stützend herausgestellt, da ansonsten keine weitere Fläche am Heck weiß ist. Die meisten Bereiche am gesamten Fahrzeug sind eher dunkel. In der Evaluierung von dunklen Flächen zeigt sich, dass der befahrbare Untergrund im Vergleich zum Fahrzeug eher heller ist, sodass einzig die Reifen des Fahrzeuges als dunkle Flächen modelliert wurden. Insgesamt überzeugte das aus Kanten sowie roten, hellen und dunklen Flächen bestehende Modell mit einer zuverlässigen visuellen Fahrzeugerkennung im Nahbereich und bei guten Lichtverhältnissen.



**Abbildung 6.7:** 3D-Merkmalmodelle eines Lkw vom Typ MAN mil gl.

Aufbauend auf den Erkenntnissen vom ersten Merkmalsmodell wurde ein zweites Modell für große Konvoiabstände und wechselhafte Lichtverhältnissen entworfen. Bei diesem wurde auf die Modellierung der Fahrzeugaußenkanten und der Kanten vom Flecktarn verzichtet. Der Grund liegt zum einen darin, dass ein Flecktarnmuster bei großen Entfernungen komplett dunkel erscheint. Zum anderen ist in vielen Situationen die Farbe des Hintergrundes ähnlich bzw. identisch mit der des Fahrzeuges, was eine Erkennung der Fahrzeugaußenkanten unmöglich macht. Die beiden roten Rücklichter und die dunklen Flächen an den Reifen wurden ebenfalls wie beim ersten Modell modelliert. Zusätzlich kamen weitere dunkle Flächen für die restliche Fahrzeugfläche dazu, da diese bei hohen Abständen eher dunkel erscheint. Die bisher genannten Merkmale reichen nicht für eine zuverlässige Fahrzeugerkennung bei großen Konvoiabständen. Aus diesem Grund erfolgt bei Konvoifahrten mit Abständen größer als 30 Meter eine Montage eines markanten Markers. Die Abbildung 6.7b zeigt ein Merkmalsmodell, welches ein Markermodell beinhaltet. Dabei ist das Markermodell aufgrund der hierarchischen Modellstruktur<sup>1</sup> beliebig auswechselbar. In diesem Fall handelt es sich um einen Marker mit einer  $0,5 \times 0,5$  Quadratmeter großen weißen Grundfläche und einer aufgeklebten schwarzen Raute. Die Merkmale – weiße und dunkle Flächen sowie die acht Kanten des Markers – sind besonders signifikant und folglich zuverlässig detektierbar. Als Alternative wurden zwei weitere Marker modelliert, wobei die Markergröße beliebig skaliert werden kann. Ein Markertyp besteht aus einer weißen Grundfläche und einem schwarzem Kreis zur Markierung des Mittelpunktes. Ein weiterer modellierter Markertyp ist eine orangefarbene militärische Warntafel in der Form eines kleinen Rechteckes.

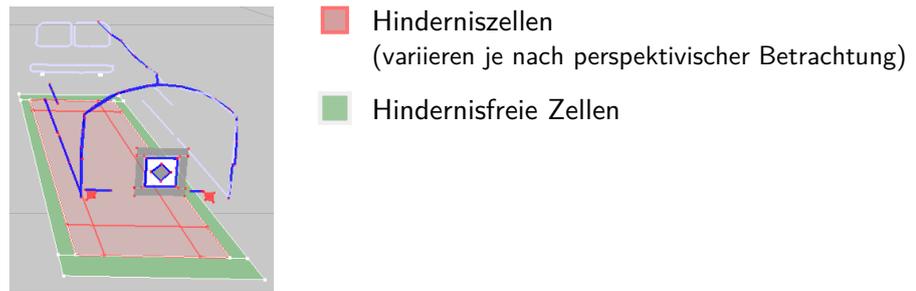
Mit der Integration eines Markermodells war eine Fahrzeugerkennung auch bei großen Konvoiabständen und schwierigen Lichtverhältnissen durchführbar. Eine fehlerhafte Detektion ist allerdings nicht auszuschließen. Zum einen kann eine tiefstehende Sonne bei einem ungünstigen Winkel die Kamera blenden, was eine visuelle Fahrzeugerkennung stark erschwert. Aus diesem Grund sollten unterschiedliche Sensortypen zum Einsatz kommen. Besitzt das autonome Fahrzeug z. B. einen Tiefensensor (LiDAR oder Stereo-Kamerasystem), dann sollte das Modell mit weiteren Merkmalen zur Robustheitssteigerung ergänzt werden. Mit einem Tiefensensor können u. a. Bereiche definiert werden, die hindernisbehaftet oder hindernisfrei sind. Die Abbildung 6.8a zeigt ein Merkmalsmodell mit solchen Merkmalen für den MAN mil gl. In den Konvoifahrten dieser Arbeit stellte sich heraus, dass der Fahrzeugrand in fast allen Fällen frei von Hindernissen bleibt. Lediglich bei Fahrten über hohem Gras war die komplette Fahrzeugumgebung hindernisbehaftet. Hinsichtlich der hindernisbehafteten Zellen vom Fahrzeug muss die perspektivische Betrachtung des Sensors berücksichtigt werden. Falls zum Beispiel das Führungsfahrzeug unmittelbar und mit gleicher Orientierung vor dem Egofahrzeug steht, ist lediglich das Heck des Führungsfahrzeuges sichtbar bzw. die Laserstrahlen treffen ausschließlich darauf. Dagegen sind bei einer 45 Grad-Rotation des Führungsfahrzeuges das Heck und eine Fahrzeugseite im Sensor sichtbar. Folglich sind die im Merkmalsmodell definierten

<sup>1</sup>Ein Modell kann wiederum aus Modellen bestehen.

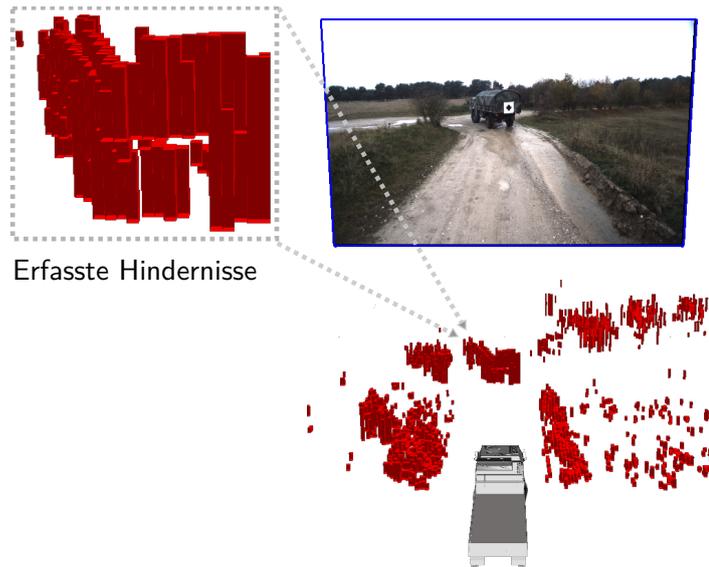
Zellen unter dem Fahrzeug je nach perspektivischer Betrachtung hindernisfrei oder hindernisbehaftet.

Ein weiterer wichtiger Aspekt, der bei der Modellierung von Merkmalen beachtet werden muss, ist der Erfassungsbereich und die Reichweite eines Sensors. Der TULF besitzt zwei 32-Zeilen LiDAR-Sensoren der Firma Velodyne, die auch Hindernisse in großen Entfernungen wahrnehmen können (vgl. Abbildung 6.8b). Dagegen erfasst ein kleiner 8-Zeilen Ibeo LiDAR-Sensor nur wenige Bereiche eines Führungsfahrzeuges. Die Abbildung 6.8c veranschaulicht eine problematische Situation. In dieser kann der StrAsRob mit seinem 8-Zeilen Ibeo LiDAR-Sensor für einen Moment noch wenige Hindernisinformationen vom Führungsfahrzeug wahrnehmen. Danach vergrößert sich der Abstand zum Führungsfahrzeug, wodurch nur noch einzelne bis keine Laserstrahlen auf das Fahrzeug treffen. Dies führt zu einem leeren 2,5D-Hindernisgitter, da kein Höhenunterschied mehr pro Zelle ermittelbar ist. Somit müsste die Fahrzeugerkennung rein auf Basis der visuellen Merkmale erfolgen. Aus diesem Grund wird ein LiDAR-Sensor mit vielen Laserdioden empfohlen.

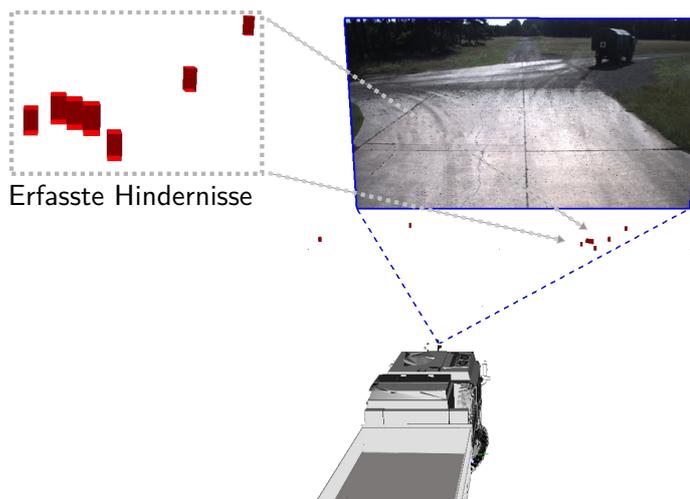
Abschließend wird auf den hohen zeitlichen Aufwand für die manuelle Erstellung eines 3D-Merkmalmodelles hingewiesen. Angefangen von der Erstellung eines groben Modellentwurfs ist eine stufenweise Modellanpassung anhand von aufgezeichneten Sensordaten erforderlich. Die benötigte Bearbeitungsdauer erstreckt sich von mehreren Stunden bis hin zu Tagen.



(a) Merkmalsmodell für Tiefendaten



(b) TULF: Erzeugtes 2,5D-Hindernisgitter aus den Tiefendaten der 32-Zeilen Velodyne LiDAR-Sensoren



(c) StrAsRob: Generiertes 2,5D-Hindernisgitter aus den Tiefendaten des 8-Zeilen Ibeo LiDAR-Sensors

**Abbildung 6.8:**

Sensorabhängiges Merkmalsmodell eines Lkw und dessen Verwendung mit zwei unterschiedlichen LiDAR-Sensortypen.

## 6.2.2 Automatische Modellgenerierung

Die automatische Modellgenerierung erzeugt – ohne softwareseitigen Eingriff – ein maßstabgetreues 3D-Merkmalmodell in Sekunden. Hierfür sind Sensordaten von dem gewünschten Führungsfahrzeug notwendig. Diese wurden in dieser Arbeit mit MuCAR-3 oder MuCAR-4 aufgenommen. Während der Aufnahme der Sensordaten erfolgt die Modellgenerierung parallel und in Echtzeit. Dabei können die Zwischenergebnisse, wie z. B. ein Modell ausschließlich von der Fahrzeugrückseite, unmittelbar zur modellbasierten Fahrzeugerkennung und folglich zur autonomen Fahrzeugführung verwendet werden.

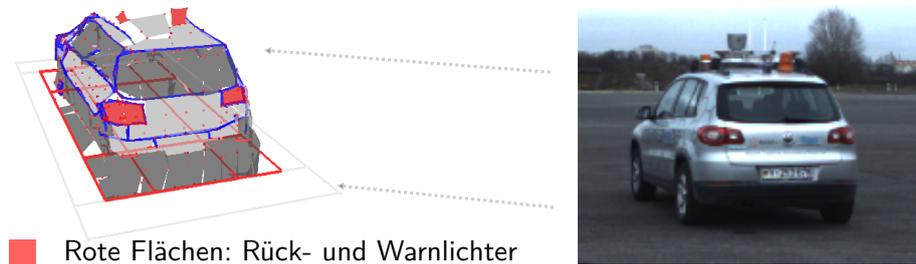
Ein resultierendes 3D-Merkmalmodell von einem zivilen Pkw stellt Abbildung 6.9 grafisch dar. Das Modell beinhaltet zum einen viele dominante Farb-, Wärme- und Hindernismerkmale und zum anderen das Kennzeichen, hinterlegt als Texturmerkmal im Farbmodell. Während der Modellerzeugung fuhr MuCAR-3 komplett um den Pkw herum, sodass Sensordaten aus unterschiedlichen Perspektiven verarbeitet wurden. Die Verarbeitungszeit dieser Sensordaten dauerte etwa 0,23 Sekunden. Die gesamte Fahrzeit betrug weniger als zwei Minuten.

Anschließend wurde das autonome Folgen getestet. Die modellbasierte Fahrzeugerkennung konnte das Führungsfahrzeug kontinuierlich detektieren und lieferte der Pfadplanung u. a. Positionsangaben zum autonomen Folgen. Das Folgen erfolgte ohne Unterbrechungen und für den Benutzer aus visueller Betrachtung spurtreu, was wiederum auf eine erfolgreiche Modellerzeugung hinweist. Zur Überprüfung der Modellgenauigkeit wurden die Höhe, Breite und Länge des Führungsfahrzeuges manuell vermessen. Anschließend konnte die automatisch ermittelte Fahrzeugdimension – bestimmt durch die konvexe Hülle der Merkmale – evaluiert werden. Diese weist einen Root Mean Square Error (RMSE) von 0,08 Metern auf.

Visuelle Eindrücke über die automatische Erzeugung von 3D-Merkmalmodellen und deren Verwendung finden sich auf: [www.mucar3.de/itsc2016](http://www.mucar3.de/itsc2016)  
Eine zugehörige Veröffentlichung findet sich in Fries und Wuensche [2016].

## 6.2.3 Manuelle vs. automatische Modellgenerierung

Abschließend erfolgt in diesem Abschnitt ein Vergleich der manuellen und automatischen Modellgenerierung von einem VW Tiguan. Die jeweiligen Modellergebnisse sind auf der S. 53 in der Abbildung 4.2 und auf der S. 131 in der Abbildung 6.9 visualisiert. Beide Modellergebnisse zeigen umfangreiche Farb-, Wärme- und Hindernismodelle. Die Hindernismodelle stimmen fast überein, dagegen sind die Wärme- und Farbmodelle sehr unterschiedlich. Hierbei wurden in der manuellen Modellgenerierung ausschließlich die vom Menschen als besonders markant wahrgenommene Merkmale modelliert. Im Gegensatz dazu modellierte die automatische Modellgenerierung deutlich diskreter und zusätzlich das Kennzeichen. Allerdings beeinflusst



- Rote Flächen: Rück- und Warnlichter
- Dunkle Flächen: Reifen, Fenster, Unterboden
- Helle Flächen: Frontscheinwerfer, Karosserie
- Kanten

(a) Modell mit Farbmerkmalen



- Heiße Flächen (Auspuff, Motor)
- Warme Flächen (Auspuffnähe, Reifen, Karosserie)

(b) Modell mit Wärmemerkmalen



- Hinderniszellen (Sichtabhängig)
- Hindernisfreie Zellen

(c) Modell mit Hindernismerkmalen

**Abbildung 6.9:**

Sensorabhängige 3D-Merkmalmodelle zur Auswertung von Farb-, Wärme- und Hindernismerkmalen.

die Anzahl der modellierten Merkmale den Rechenaufwand für die modellbasierte Fahrzeugerkennung mit einem Partikel-Filter. Infolgedessen sind die ermittelten warmen Flächen der automatischen Modellgenerierung zwar richtig, jedoch rechenaufwendiger. Des Weiteren ist die automatische Modellgenerierung ungenauer bzgl. der 3D-Positionen der Modellmerkmale. Dies liegt an der eingeschränkten horizontalen und vertikalen Auflösung eines LiDAR-Sensors.

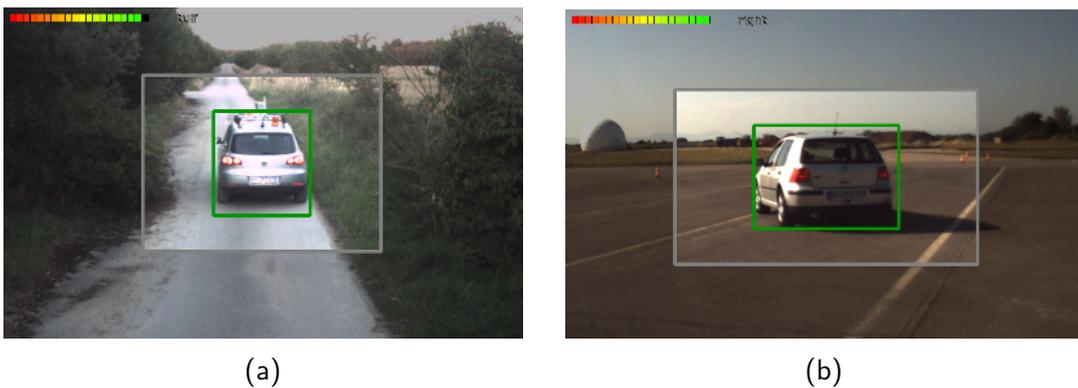
Ein großer Vorteil der automatischen Modellgenerierung ist der zeitlich geringe Aufwand zur Erstellung eines Modells. Eine manuelle Modellerstellung dauert circa vier bis acht Stunden bis zum ersten Entwurf. Anschließend erfolgt anhand von unterschiedlichen Sensordatensätzen eine zeitlich aufwändige sowie schrittweise Verbesserung des Modells. Die automatische Modellgenerierung verarbeitet dagegen die Sensordaten in Echtzeit, sodass unmittelbar einem unbekanntem Fahrzeug gefolgt werden kann.

## 6.3 Fahrzeugerkennung und -verfolgung

Dieses Kapitel geht auf die Praxisergebnisse der entwickelten Erkennung eines Führungsfahrzeuges und dessen zeitlicher Verfolgung (engl. *Tracking*) ein. In den nächsten zwei Abschnitten erfolgt einerseits eine Beschreibung über die Praxisversuche bei Tag sowie bei Nacht und andererseits wird das autonome Folgen anhand der Fahrzeugerkennung veranschaulicht. Abschließend erfolgt im dritten Abschnitt die Bewertung des Gesamtsystems hinsichtlich der Robustheit und der Schätzgenauigkeit.

### 6.3.1 Fahrzeugerkennung und -verfolgung bei Tag

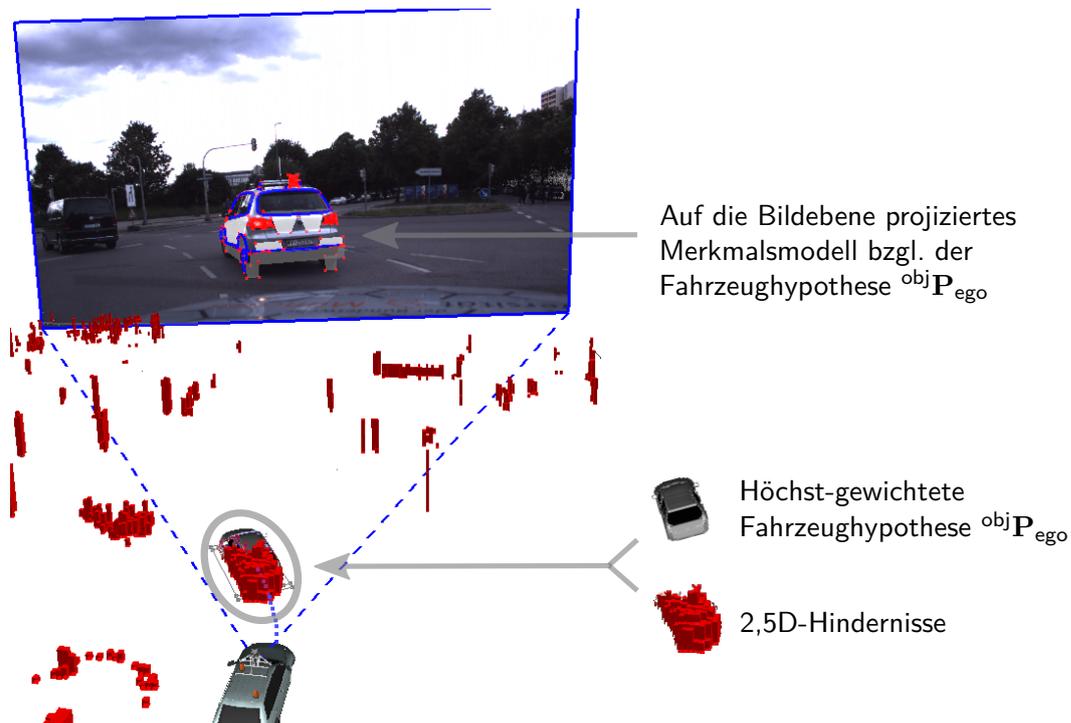
Für die Fahrzeugerkennung bei guten Lichtverhältnissen existieren zwei Verfahren. Das rein visuelle Verfahren aus dem Abschnitt 5.1.2 basiert auf kaskadierten Klassifikatoren (vgl. Grundlagenkapitel 3.3) und detektiert Fahrzeugseiten mit trainierten Klassifikatoren. Die detektierten Fahrzeugseiten bzw. die daraus abgeleiteten Fahrzeugaußenkanten werden anschließend in einem Kalman-Filter sequentiell eingefiltert. Dieser Filter schätzt auf Basis der Außenkanten die ego-relative Fahrzeugpose vom Führungsfahrzeug sowie dessen Geschwindigkeit über die Zeit. Die Abbildung 6.10 zeigt den Praxiseinsatz des echtzeitfähigen Verfahrens. Insgesamt wurden acht perspektivenabhängige Klassifikatoren trainiert, die zusammen ein Fahrzeug rundum detektieren können. Folglich detektiert jeder Klassifikator jeweils einen  $360^\circ / 8 = 45^\circ$ -Bereich eines Fahrzeuges. Das Verfahren und die Messgenauigkeit wurden in Fries et al. [2013] und Fries und Wuensche [2014] publiziert und ermöglichen das autonome Folgen eines bekannten Führungsfahrzeuges.



**Abbildung 6.10:**

Visuelle Fahrzeugerkennung mit kaskadierten Klassifikatoren. Das Detektionsergebnis der Fahrzeugseite(n) ist durch ein grünes Rechteck dargestellt. Ein graues Rechteck kennzeichnet einen reduzierten Suchbereich, welcher anhand der Kovarianzen eines Kalman-Filters berechnet wird.

Das zweite Verfahren verwendet für die Fahrzeugerkennung einen Partikel-Filter und basiert auf detailliertem Modellwissen. Die Methode wurde ausführlich im Kapitel 5.1.3 beschrieben und benötigt deutlich mehr Rechenleistung als das erste Verfahren. Allerdings ist die Zustandsschätzung genauer und robuster. Die Abbildung 6.11 zeigt die modellbasierte Fahrzeugerkennung in einer urbanen Umgebung. In der abgebildeten Verkehrssituation fährt ein VW Touareg einem VW Tiguan mit der Berücksichtigung eines geschwindigkeitsabhängigen Abstandes spurtreu nach. Für die Schätzung der Fahrzeugzustände werden ego-relative Fahrzeughypothesen (Partikel) durch LiDAR- und Bildmerkmale ausgewertet. Zu den geschätzten Fahrzeugzuständen gehören die ego-relative Pose des Führungsfahrzeuges sowie dessen Geschwindigkeit. Hinsichtlich der Bewertung der Fahrzeughypothesen werden zum einen die in der Abbildung 6.11 durch rote Balken visualisierten Hindernisinformationen berücksichtigt. Zum anderen werden visuelle Merkmale für eine robuste Fahrzeugerkennung und für eine hohe Schätzgenauigkeit ausgewertet. Zusätzlich zeigt die Abbildung die visuellen Merkmale des VW Tiguan: Helle, dunkle und rote Flächen, dominante Eckpunkte und innenliegende Kanten.



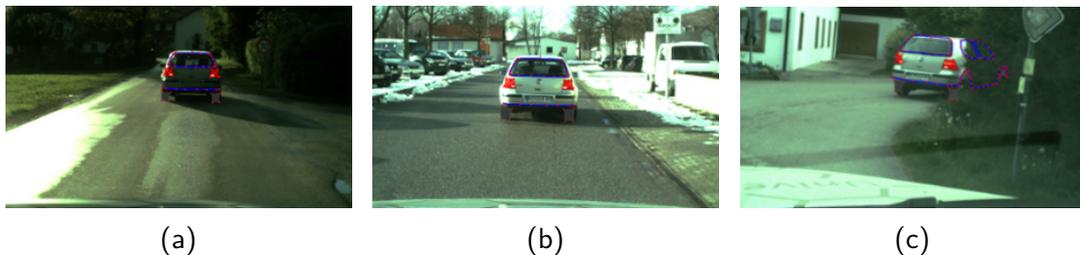
**Abbildung 6.11:**

Modellbasierte Fahrzeugerkennung mit einem Partikel-Filter. Die Aufnahme stammt aus dem Jahre 2015 und zeigt das aktuellste Merkmalsmodell eines VW Tiguan.

Im urbanen Umfeld erfolgten die autonomen Fahrten überwiegend im Geschwindigkeitsbereich zwischen 20 und 60 km/h. Der Abstand zum Führungsfahrzeug betrug beim Anhalten an einer Ampel fünf Meter und bei Fahrten mit hohen Geschwindigkeiten 30 Meter. Problematische Situationen ergaben sich beim Einscheren von

anderen Verkehrsfahrzeugen. Denn oftmals verdeckt das eingescherte Fahrzeug das Führungsfahrzeug so stark, dass nur noch wenig bis gar keine Sensordaten mehr vom Führungsfahrzeuge erfasst werden. Da in dieser Arbeit – aus Gründen des wissenschaftlichen Anspruchs – die Funkkommunikation zwischen den Fahrzeugen nicht verwendet wird, muss aus Sicherheitsgründen der Sicherheitsfahrer in solchen Situationen eingreifen und die Kontrolle über das Fahrzeug übernehmen. Dazu existiert einerseits ein Notausschalter in der Mittelkonsole, der den autonomen Modus unterbricht. Alternativ besitzt ein Sicherheitsfahrer auch die Möglichkeit des Übersteuerns.

Aufgrund der erzielten hohen Robustheit der Fahrzeugerkennung konnten zahlreiche autonome Fahrten, sowohl bei bewölkten und sonnigem Wetter als auch bei Schnee und Regen, erfolgreich durchgeführt werden. Die Abbildung 6.12 zeigt drei Situationen aus dem Jahre 2013, in denen trotz schlechter Sichtbedingungen das Führungsfahrzeug erkannt werden konnte.



**Abbildung 6.12:**

Modellbasierte Fahrzeugerkennung eines VW Golf trotz schlechter Sichtbedingungen: (a-b) Sonniges Wetter verursacht u. a. Schlagschatten. (c) Sichteinschränkung des Fahrzeuges durch Vegetationen.

Ergänzend zur urbanen Umgebung wurde die Fahrzeugerkennung auch in unstrukturierten Gebieten eingesetzt. Darunter zählt eine Vielzahl von Feldwegen, die an Waldgebiete angrenzten und durch Waldgebiete führten. Diese Wege sind meistens schmal und besitzen keine Straßenmarkierungen, sind eher uneben und oft ragen Sträucher und Äste seitlich in den Weg hinein. Auch auf diesen Wegen folgten die selbstfahrenden Fahrzeuge dieser Arbeit spurtreu. Insgesamt wurden mehrere hundert autonom gefahrenen Kilometer in unstrukturiertem Gelände absolviert. Die hohe Schätzgenauigkeit über die ego-relative Pose des Führungsfahrzeuges ermöglichte ein spurtreues Folgen, sodass schmale Wege für das Gesamtsystem kein Problem darstellten. Die Unebenheit der Wege und das Befahren von Hügeln war aufgrund der geschätzten 3D-Position und 3D-Orientierung (6D-Fahrzeugpose) möglich. Vegetation, die in LiDAR-Daten als Hindernisse erscheinen, führten durch die Kombination mit Kamerasensoren zu keinen Schwierigkeiten.

Die Abbildung 6.13 veranschaulicht drei autonome Fahrten in einer unstrukturierten Umgebung. Dabei stammen die drei Bilder (a-c) aus den Jahren 2014, 2015 und 2016.



(a) TULF folgt einem Lkw



(b) StrAsRob folgt einem Lkw



(c) TULF folgt einem StrAsRob

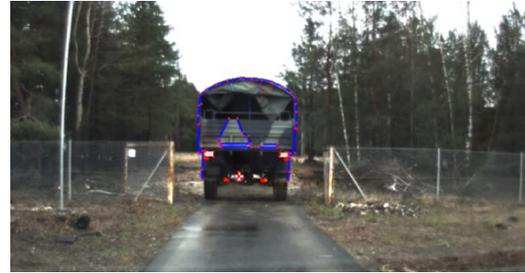
**Abbildung 6.13:**

Autonomes Folgen in einer unstrukturierten Umgebung: (a) TULF folgt einem MAN mil gl 5t im Jahre 2014. (b) StrAsRob folgt einem anderem MAN mil gl 5t im Jahre 2015. (c) TULF folgt dem StrAsRob im Jahre 2016.

Abschließend zeigen die Abbildungen 6.14 und 6.15 weitere visuelle Eindrücke von autonomen Tagesfahrten. Dabei wurden die einzelnen Bilder zwischen den Jahren 2013 und 2016 aufgenommen. Aufgrund der fortlaufenden Weiterentwicklung sind die fahrzeugspezifischen Merkmalsmodelle teilweise unterschiedlich dargestellt.



(a)



(b)



(c)



(d)



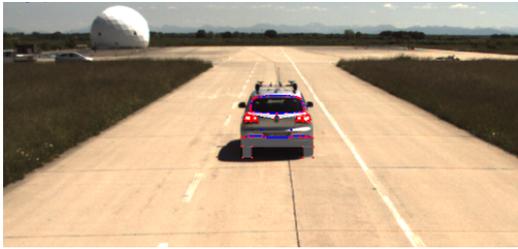
(e)



(f)

**Abbildung 6.14:**

Visuelle Eindrücke der modellbasierten Fahrzeugerkennung eines Lkw vom Typ MAN mit gl.



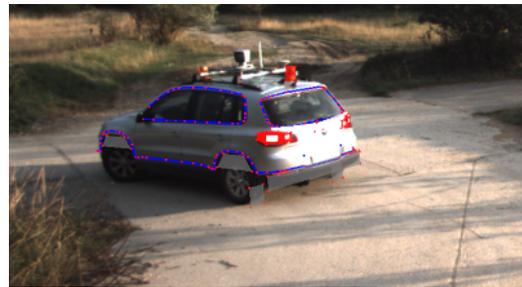
(a)



(b)



(c)



(d)



(e)



(f)

**Abbildung 6.15:**

Visuelle Eindrücke der modellbasierten Fahrzeugerkennung eines VW Tiguan: (a-b) Gute und schlechte Lichtbedingung. (c-d) Scharfe Kurvenfahrt. (e) Hoher Konvoiabstand von über 40 Metern. (f) Fahrzeugerkennung trotz teilweiser Verdeckungen.

### 6.3.2 Fahrzeugerkennung und -verfolgung bei Nacht

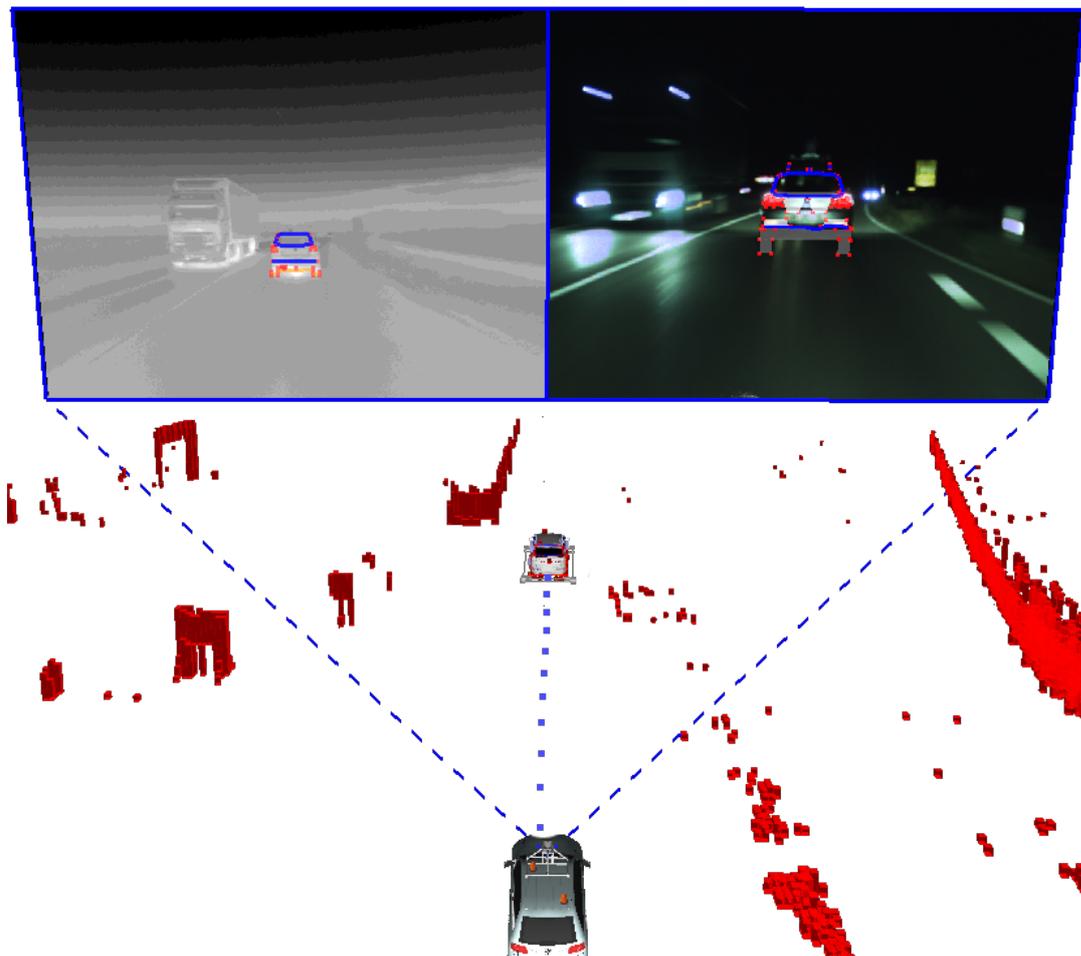
Die Fahrzeugerkennung konnte aufbauend auf den bisherigen Entwicklungsarbeiten sowohl für den Einsatz bei Dämmerung als auch für den Nachteinsatz erweitert werden. Die erste erfolgreiche autonome Fahrt bei Nacht erfolgte im September 2014. Dabei bestand der Konvoi aus zwei Fahrzeugen. MuCAR-3 folgte erstmals MuCAR-4 bei kompletter Dunkelheit und auf Basis von Bildmerkmalen aus einer Wärmebildkamera und anhand von LiDAR- und INS-Daten. Nach einigen erfolgreichen Testkilometern wurde der entwickelte Folgeassistent auf seine Grenzen getestet. Hierbei absolvierte das System einerseits scharfe Richtungswechsel bis zu 100 Grad erfolgreich. Andererseits ermöglichte die heterogene Sensorkonstellation starke Beschleunigungen aus dem Stand heraus sowie Konvoiabstände von 40 Metern. Anschließend erfolgte eine autonome Fahrt mit einer Fahrzeugerkennung bei der einmal nur Wärmedaten und ein anderes Mal nur Laserdaten verwendet wurden. Die Tests verliefen zwar überwiegend erfolgreich, zeigten allerdings schnell die Grenzen des Systems: In zwei Situationen war eine Fahrzeugerkennung nicht möglich und in einer weiteren Situation wurde die Fahrzeugpose zu ungenau geschätzt, sodass das Folgefahrzeug leicht ins Schwingen kam.

Als lediglich die Wärmebildinformationen verwendet wurden, konnte das Führungsfahrzeug bei scharfen Kurvenfahrten, aufgrund eines starren stark eingeschränkten Sichtbereiches, nicht detektiert werden. Die Wärmebildkamera befindet sich fest positioniert auf dem Dach des VW Touareg mit einem festen horizontalen Sichtbereich von 45 Grad. Dadurch stehen der Fahrzeugerkennung beim Verlassen des visuellen Sichtbereiches keine messbaren Informationen vom Führungsfahrzeug zur Verfügung. Diese Problematik könnte allerdings behoben werden. Dafür müsste entweder eine Optik mit einem größeren horizontalen Sichtbereich oder eine Aktorik zum Gieren der Kamera integriert werden.

Beim durchgeführten Testszenario, in dem ausschließlich Laserdaten verwendet wurden, konnte die Fahrzeugerkennung das Führungsfahrzeug in unstrukturierten Gelände nicht zuverlässig erkennen: Zum einen sind die Hinderniszellen von zu hohem Gras gleichermaßen wie die Hinderniszellen unter dem Führungsfahrzeug belegt. Andererseits ragen des Öfteren ungefährliche Äste und Büsche in Waldwege hinein, welche im 2,5D-Hindernissgitter als Hindernisse vorliegen.

Aus diesen genannten Gründen wird eine Kombination von verschiedenen Sensoren dringend empfohlen. Die Abbildung 6.16 visualisiert einen autonomen Konvoi bei Nacht in der unterschiedliche Sensoren zum Einsatz kamen. Die Mitternachtsfahrt fand auf einer Landstraße in Unterhaching, Bayern statt. Dabei fuhr MuCAR-3 einem VW Tiguan mit Geschwindigkeiten bis 60 km/h nach. In der linken oberen Ecke ist das verwendete Wärmebild zu sehen. Je heller der Pixelwert, desto größer ist der Temperaturwert. Die Wärmebildkamera benötigt eine sehr geringe Belichtungs- bzw. Integrationszeit von maximal 1 ms. Dagegen braucht eine Tageslichtkamera bei dunklen Lichtbedingungen viel Zeit zum Belichten der Kamerazellen. Dabei führt

eine zu lange Belichtungszeit, bei bewegten Objekten sowie bei einer bewegten Kamera, zu unscharfen Bildern. In der rechten oberen Ecke der Abbildung 6.16 ist diese Unschärfe zu erkennen. Die Tageslichtkamera belichtet 100 ms bei einer Fahrzeuggeschwindigkeit von 60 km/h. Das bedeutet, dass ein Kamerabild während einer  $\frac{60}{3,6} \frac{m}{s} \cdot 0,1s = 1,67m$  langen Fahrt aufgenommen wird. Dadurch resultiert ein Farbbild mit deutlichen Verschmierungen, was folglich die Fahrzeugerkennung erschwert und zu einem ungenaueren Schätzergebnis führt. Demzufolge werden bei der nächtlichen Fahrzeugerkennung keine Farbmerkmale bei hohen Geschwindigkeiten bzw. großen Belichtungszeiten verwendet. Zur Verbesserung der Fahrzeugerkennung kam ein LiDAR-Sensor bei der autonomen Nachtfahrt zum Einsatz. Dieser liefert wichtige Hindernisinformationen, die in der Abbildung durch rote Balken visualisiert sind.



**Abbildung 6.16:**

Modellbasierte Erkennung eines ausgewählten Führungsfahrzeuges bei Nacht: Rote Balken entsprechen 2,5D-Hindernisdaten. Die linke obere Ecke der Grafik zeigt das Wärmebild. Je heller der Pixel dargestellt ist, desto wärmer ist die Temperatur. In der oberen rechten Ecke ist ein Farbbild einer Tageslichtkamera abgebildet. Zusätzlich ist in allen Sensordaten die Fahrzeughypothese inkl. 3D-Merkmalmodell dargestellt, die das höchste Partikelgewicht besitzt.

Eine weitere autonome Nachtfahrt zeigt die Abbildung 6.17 im Überblick. Die abgebildete Route von insgesamt zwölf Kilometern wurde mit MuCAR-3 und MuCAR-4 durchgeführt. Der Start war auf dem Universitätsgelände der Bundeswehr in München. Auf diesem Gelände fuhr der VW Touareg dem VW Tiguan in einer Tempo-30-Zone nach. Der eingestellte geschwindigkeitsabhängige Sollabstand von zehn Metern in Kombination mit einer geringen Fahrtgeschwindigkeit vom Führungsfahrzeug führten zu einem sehr spurtreuen Folgen. Die anspruchsvollste Situation war eine 180 Grad-Wende auf einer zweispurigen Fahrbahn. Dort fuhr das Führungsfahrzeug aus dem visuellen Sichtbereich der Wärmebildkamera heraus, sodass lediglich die Laserdaten zur Fahrzeugerkennung zur Verfügung standen. Zusätzlich beschleunigte das Führungsfahrzeug aus der Kurve heraus, was die Fahrzeugerkennung ebenfalls erschwerte. Diese anspruchsvolle Verkehrssituation konnte allerdings von MuCAR-3 erfolgreich bewältigt werden. Anschließend wurde das Universitätsgelände der Bundeswehr verlassen, indem zwei Schranken passiert wurden. Dabei sind in den Schranken Bewegungssensoren sowie unterirdisch Induktionsschleifen integriert, die ein Fahrzeug passieren lassen, sobald es wenige Sekunden vor einer Schranke anhält. Die Situation an den Schranken stellte für das Gesamtsystem keine Herausforderung dar: MuCAR-4 fuhr als erster durch die Schranke und wartete kooperativ ein paar Meter hinter der Schranke, die in der Zwischenzeit wieder runterging. MuCAR-3 folgte MuCAR-4 mit einem geschwindigkeitsabhängigen Sollabstand bis zur Schranke, hielt an und fuhr beim Öffnen der Schranke selbstständig weiter. Anschließend verließ der Zweierkonvoi das Universitätsgelände und führte die Konvoifahrt auf einer Landstraße fort. Hierbei wurde der Ort Unterhaching mit einer Durchschnittsgeschwindigkeit von 42 km/h durchfahren. Dabei passierte MuCAR-3 einen verkehrsreichen Kreislauf autonom (vgl. Abbildung 6.18a). Anschließend wurde eine Autobahnauf- und -abfahrt mit einer Maximalgeschwindigkeit von 65 km/h erfolgreich passiert. Eine weitere Verkehrssituation war das Durchfahren eines langen Tunnels (siehe Abbildung 6.18b). Hierbei demonstrierte das in dieser Arbeit entwickelte Gesamtsystem erfolgreich, dass kein GNSS-Empfang für einen spurtreuen autonomen Konvoi notwendig ist.

Insgesamt konnte die zwölf Kilometer lange Strecke autonom und damit ohne Eingriff des Sicherheitsfahrers absolviert werden. Abschließend sei auf das Youtube-Video [http://youtu.be/Q8wtJ\\_YOg64](http://youtu.be/Q8wtJ_YOg64) verwiesen, das einen Ausschnitt der Konvoifahrt zeigt.



↑  
Abbildung 6.18a

**Abbildung 6.17:**

Überblick einer autonomen zwölf Kilometer langen Fahrt bei Nacht. Die Positions- bzw. Eigenbewegungsdaten des autonomen Folgefahrzeuges sind mit einer grünen Linie gekennzeichnet. Die ego-relative Positionsschätzung vom Führungsfahrzeug ist durch eine blaue Linie visualisiert. [Kartenmaterial: Google Maps]



(a) Verkehrsreicher Kreisel



(b) Langer Tunnel

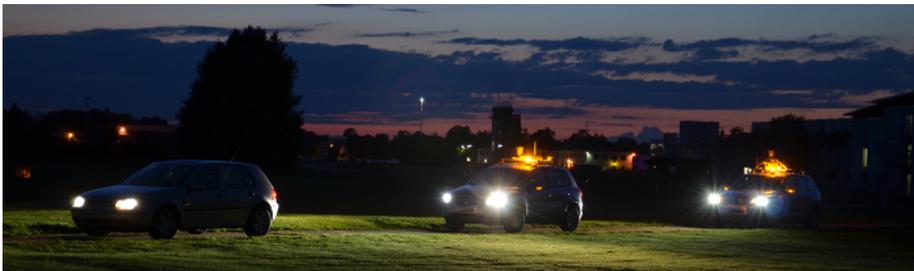
**Abbildung 6.18:**

Anspruchsvolle Verkehrssituationen während einer autonomen Nachtfahrt. Die grüne und blaue Linie veranschaulichen die Eigenbewegungsdaten vom Folgefahrzeug sowie die ego-relative Positionsschätzung des Führungsfahrzeuges. [Kartenmaterial: Google Maps]

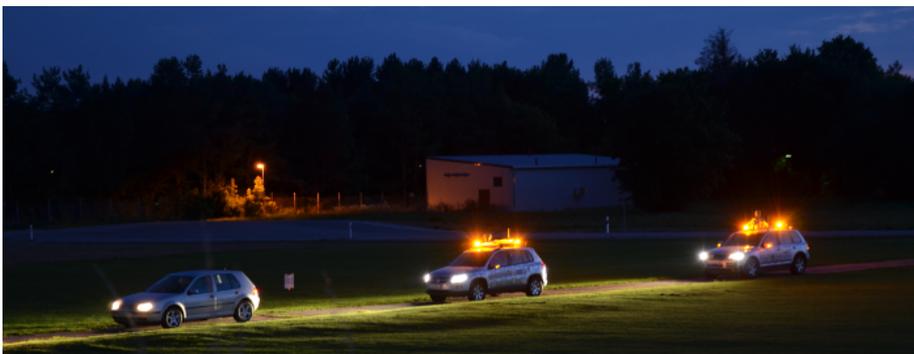
Die meisten Praxistests erfolgten in einem Konvoi bestehend aus zwei Fahrzeugen. Ergänzend wurde ein Mehrfachkonvoi mit drei Pkw erprobt. Dieser Praxistest erfolgte mit einem manuell gefahrenen VW Golf 4 als Führungsfahrzeug und MuCAR-3 sowie MuCAR-4 als selbstfahrende Folgefahrzeuge. Hierbei konnten die Versuchsfahrten sowohl bei Tag als auch bei Nacht erfolgreich durchgeführt werden. Das jeweils lokale Führungsfahrzeug wurde kontinuierlich und ohne Kommunikation zwischen den Fahrzeugen identifiziert. Das Gesamtsystem bewies, dass eine Wahrnehmung mit bild- und abstandsgebenden Sensoren für eine zuverlässige Fahrzeugerkennung ausreicht. Dabei ermöglicht die Positions-, Orientierungs- und Geschwindigkeitsschätzung ein überwiegend spurtreues Folgeverhalten, sogar bei hohen Geschwindigkeiten und Fahrzeugabständen. Die nachfolgende Abbildung 6.19 zeigt drei visuelle Eindrücke eines Mehrfachkonvois.



(a) Mehrfachkonvoi bei Dämmerung



(b) Mehrfachkonvoi bei Nacht



(c) Mehrfachkonvoi bei Nacht

**Abbildung 6.19:** Autonomes Fahren im Dreierkonvoi.

An dieser Stelle sei erwähnt, dass die Kommunikation zwischen den Fahrzeugen möglich und sinnvoll ist. In dieser Arbeit wurden über 95 Prozent der autonomen Fahrten ohne eine Kommunikationsverbindung durchgeführt. Der Grund dafür ist das Aufzeigen, wie erfolgreich die Wahrnehmung mit Kameras und Laserscannern sein kann. Die hohe Genauigkeit von der Fahrzeugerkennung führt zu sehr kleinen Ablagen in der Sollspur. Allerdings würden sich diese kleinen Ablagen bei einem großen Konvoi von z. B. fünf Teilnehmern akkumulieren und könnten zu kritischen Verkehrssituationen führen. Damit eine auftretende Ablage eines Konvoiteilnehmers keine Auswirkung auf die Folgefahrzeuge hat, kann optional die Ablage über eine Funkkommunikation übermittelt werden. Mit der Information über die Ablage des jeweils lokalen Führungsfahrzeuges, kann die Sollspurgenerierung derartig angepasst werden, dass das Akkumulieren von Schätzfehlern sowie Ablagen reduziert werden. Details über einen kooperativen Regler, zum spurtreuen Nachfahren im Mehrfachkonvoi, finden sich in Heinrich und Wuensche [2017].

Im Abschließenden wird auf die benötigte Rechenzeit der modellbasierten Fahrzeugerkennung eingegangen. Die Fahrzeugerkennung benötigt auf den Fahrzeugrechnern des MuCAR-3 und MuCAR-4 zwischen 20 und 80 ms. Die Rechenzeit ist sehr stark von der festgelegten Partikelanzahl abhängig, welche in den meisten Fällen zwischen 800 und 2000 Partikeln lag. Ebenfalls ist die Anzahl verwendeter Sensoren und die softwarebasierte Aufmerksamkeitssteuerung ein wichtiger Einflussfaktor. Je mehr Sensoren verwendet werden, desto mehr Daten müssen analysiert werden. Aufgrund der im Abschnitt 5.2 vorgestellten Aufmerksamkeitssteuerung kann die Datenmenge jedoch stark reduziert werden.

## 6.4 Beurteilung des Gesamtsystems

An dieser Stelle wird das autonome Folgen als Ganzes betrachtet und bewertet. Das Gesamtsystem kann mit mehreren Konvoiteilnehmern betrieben werden und funktioniert sowohl bei hellen als auch dunklen Lichtbedingungen. Dadurch kann sowohl bei Tag als auch bei Dämmerung und in der Nacht autonom gefahren werden. Für eine zuverlässige Erkennung eines Fahrzeuges werden die folgenden heterogenen Sensorkonfigurationen empfohlen:

- **Tag:** 360°-LiDAR-Sensor + Wärmebild- und Tageslichtfarbkamera(s)
- **Dämmerung:** 360°-LiDAR-Sensor + Wärmebild- und Farbkamera(s)
- **Nacht:** 360°-LiDAR-Sensor + Wärmebildkamera

Mit diesen Sensorkonstellationen war der Folgeassistent, sowohl bei Regen und Schnee, als auch bei bewölktem und sonnigem Wetter, spurtreu und zuverlässig einsatzbereit. Scharfe Kurven und teilweise Verdeckungen des lokalen Führungsfahrzeuges stellten keine Probleme dar. Diese Robustheit bzgl. teilweiser Verdeckungen ermöglichte zum einen das Fahren durch hohes Gras im Gelände als auch das Folgen bei Regen mit eingeschaltetem Scheibenwischer auf der höchsten Wischerstufe.

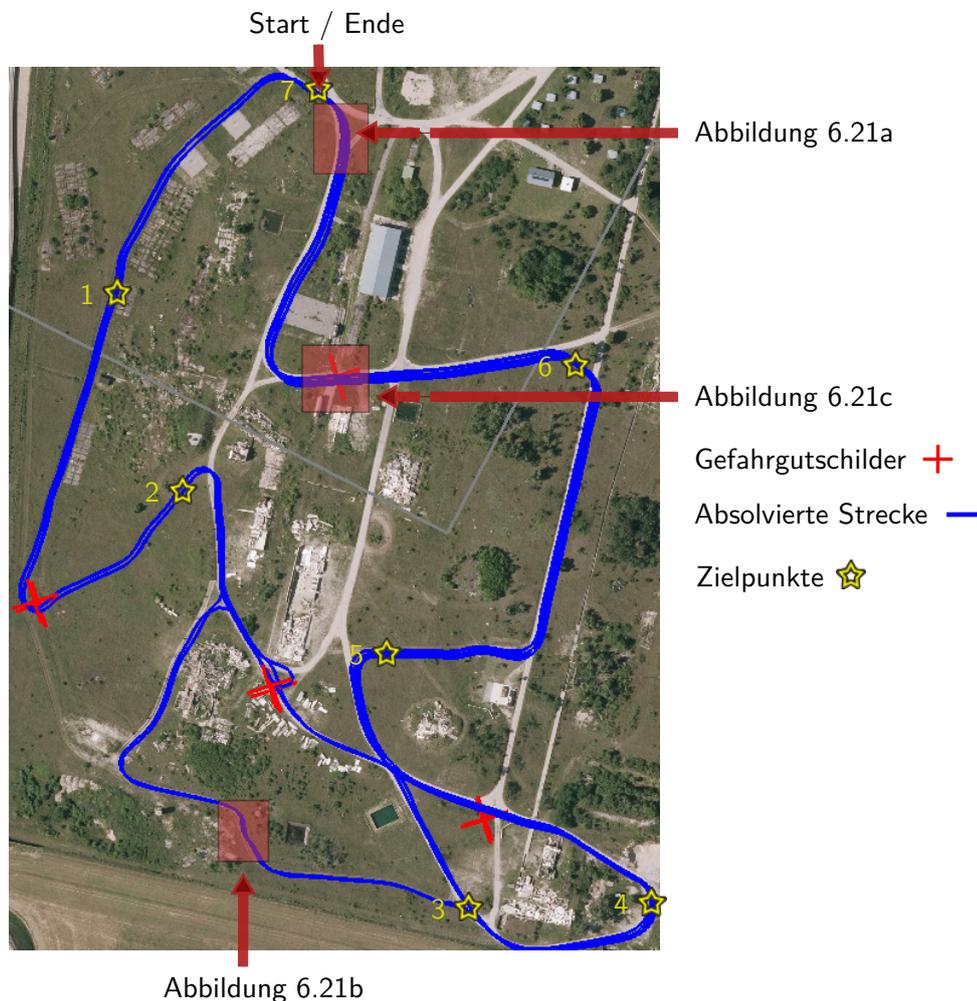
Insgesamt wurden mit dem System – welches sich kontinuierlich über die letzten Jahre weiterentwickelte – hunderte von Kilometern autonom gefahren. Dabei zählen circa 90 Prozent der gefahrenen Kilometer zu den Konvoifahrten bei Tag.

Die Teilsysteme des Gesamtsystems laufen alle in Echtzeit, sodass hohe Geschwindigkeiten gefahren werden können. Die maximale Geschwindigkeit bei autonomen Fahrttests bei Tag lag bei 100 km/h. Diese wurde auf einem abgesperrten Bereich des Universitätsgeländes der Bundeswehr erreicht. Die maximale Geschwindigkeit bei Nacht lag bei 75 km/h.

Aus Sicherheitsgründen überprüft der Folgeassistent die Zuverlässigkeit der Fahrzeugerkennung kontinuierlich und warnt gegebenenfalls den Fahrer und fordert die Übernahme der Fahrzeugführung. Reagiert der Fahrer nicht auf eine Aufforderung, parkt das System das Fahrzeug sicher in den Stillstand (Vgl. Abschnitt 5.3). Falls das Gesamtsystem fehlerhaft reagiert bzw. ausfällt, muss ein Sicherheitsfahrer einschreiten.

Mehrere Praxisvergleiche von dem in dieser Arbeit vorgestellten Folgeassistenten mit anderen Folgeassistenten erfolgte auf den Veranstaltungen der ELROB. Dazu zählen unter anderem die ELROB'12 in der Schweiz als auch die ELROB'16 in Österreich. Auf der ELROB'12 trat MuCAR-3 mit dem gekoppelten Objekterkennungsansatz aus Abschnitt 5.1.4 an. Der modellbasierte Ansatz mit der rechenintensiven Verarbeitung von 1000 Fahrzeughypothesen eines Partikel-Filters wurde mit einer schnellen, unge-

nauen sowie rein visuellen Fahrzeugerkennung basierend auf kaskadierten Klassifikatoren gekoppelt. Dies ermöglichte eine schnelle (Re-)Initialisierung des Partikel-Filters, wodurch MuCAR-3 letztendlich ohne manuelle Eingriffe die Wettkampfstrecke der ELROB meistern konnte. Weiterführend kam auf der ELROB'16 der Algorithmus zur modellbasierten Fahrzeugerkennung zum Einsatz. Dort wurde das Ergebnis des modellbasierten Ansatzes zusammen mit anderen Objekterkennungsverfahren auf Objektebene fusioniert (siehe Fries et al. [2017] und Luettel et al. [2017]). Die Spurtreue war aufgrund der schlechteren Detektionsgenauigkeit der anderen Objekterkennungsalgorithmen zwar nicht so genau wie auf der ELROB'12, jedoch überzeugte das Gesamtsystem mit einer hohen Robustheit. Konkret absolvierte MuCAR-3 die autonome Fahrt durch schwieriges unstrukturiertes Gelände, das bei leichtem Regen, sowie unter Berücksichtigung von dynamischen Hindernissen, durchfahren werden musste. Die Abbildungen 6.20 und 6.21 veranschaulichen zu passierende Wegpunkte, die absolvierte Strecke und erfolgreich detektierte ERICard-Gefahrgutschilder.



**Abbildung 6.20:**

Ergebniskarte vom Team MuCAR-3: Abgebildet sind die zu erreichenden sieben Zielpunkte, die absolvierte Wegstrecke und die vier erfolgreich detektierten Gefahrgutschilder. [Kartenmaterial: Google Maps]

## 6 Ergebnisse



(a) Hohe Konvoigeschwindigkeit



(b) Blockade



(c) Überqueren eines Gleisübergangs sowie erfolgreich detektiertes Gefahrgutschild

### Abbildung 6.21:

Anspruchsvolle Situationen während des Konvoiszenarios der ELROB'16.

Die Tabelle 6.1 listet die Jurybewertung der drei besten ELROB-Teilnehmer auf.

Team	MuCAR-3	Smart Military Vehicles	Austrian Technology
Punkte für absolvierte Wegstrecke	5600	4200	1400
Punkte für erkannte Gefahrgutschilder	1000	900 (manuell)	0
Strafe bzgl. manueller Eingriffe	0	-72	-722
Live-Übertragung, etc.	466	400	400
-----	-----	-----	-----
Erreichte Punkte	7066	5508	1078
Platzierung	1	2	3

### Tabelle 6.1:

Jurybewertung über die erfolgreichen Teilnehmer des Konvoiszenarios der ELROB'16.

Insgesamt konnte MuCAR-3 mit dem Folgeassistenten dieser Arbeit sowohl auf der ELROB'12 als auch auf der ELROB'16 als leistungsstärkstes Gesamtsystem überzeugen. Zusätzlich war MuCAR-3 das einzige Team, welches gut sichtbare sowie versteckte ERICard-Gefahrgutschilder mit Sensordaten erkennen konnte. Hinsichtlich der visuellen Erkennung von Gefahrgutschildern kam eine Abwandlung von dem im Anhang A.5 vorgestellten Verfahren zum Einsatz, welches in Fries et al. [2017] detailliert beschrieben ist.

Im Vergleich zu den Arbeiten von Manz [2013] konnte die modellbasierte Fahrzeugerkennung deutlich verbessert werden. Im Nachfolgenden werden die wesentlichen Verbesserungen aufgelistet:

- Die Fahrzeugerkennung ist unabhängig von den Lichtverhältnissen funktionsfähig und erlaubt damit Fahrten bei Tag, Dämmerung und Nacht.
- Das Gesamtsystem benötigt bei einem Programmstart, sowie bei einer notwendigen Reinitialisierung, keine Benutzereingabe über die ego-relative Position vom Führungsfahrzeug. Dieser manuelle Vorgang war notwendig, wenn sich das Führungsfahrzeug nicht direkt oder rotiert vor dem Fahrzeug befand. In dieser Arbeit kann das Führungsfahrzeug rotiert sein und sich außerhalb des Nahbereiches ( $> 20$  Meter) aufhalten.
- Die benötigte Rechenzeit wurde durch die Integration einer hardware- und softwareseitigen AOI stark reduziert.
- Die Möglichkeit der Sensordatenverarbeitung von über fünf Kameras wurde realisiert.
- Diverse Filteranpassungen und vor allem die Verbesserung der fahrzeugspezifischen Merkmalsmodelle führten zu einer Robustheitssteigerung und verbesserten die Schätzgenauigkeit (siehe Tabelle 6.2).

Im Nachfolgenden wird auf die Schätzgenauigkeit des Gesamtsystems eingegangen. Dabei wird zum einen die zum Führungsfahrzeug relative Pose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}} = \{r, \alpha, h, \phi, \theta, \psi\}$  des Folgefahrzeuges (Ego) als auch dessen Längsgeschwindigkeit  $v$  betrachtet.  $\mathbf{P}$  besteht aus einer in Zylinderkoordinaten angegebenen Positionsangabe  $\{r, \alpha, h\}$  und einer Fahrzeugorientierung  $\{\phi, \theta, \psi\}$ . Für die Untersuchung der Schätzgenauigkeit werden Referenz- bzw. Wahrheitswerte benötigt. In dieser Arbeit wurde für die Berechnung der Wahrheitswerte die globalen Positions- und Geschwindigkeitsinformationen der fahrzeugeigenen INS verwendet. Als Testfahrzeuge kamen die Institutsfahrzeuge MuCAR-3 und MuCAR-4 zum Einsatz, da beide über eine sehr hochwertige INS verfügen und zwischen den Fahrzeugen eine Funkkommunikation möglich ist. Für die Evaluierung der Schätzgenauigkeit diente das Testgelände der Universität der Bundeswehr, auf dem beide Fahrzeuge einen DGPS-Empfang besitzen. Es wurde eine fünf Kilometer lange Strecke mit scharfen Kurven und unterschiedlichem Terrain autonom abgefahren. Dabei folgte das Fahrzeug MuCAR-3 dem Führungsfahrzeug MuCAR-4 mit einem geschwindigkeitsabhängigen Sollabstand. Für die Evaluierung der Schätzgenauigkeit wurden zum einen die Schätzergebnisse gespeichert und zum anderen erfolgte die Berechnung und Speicherung der Wahrheitswerte aus den kommunizierten Fahrzeugdaten. Die kommunizierten Daten waren Positions-, Orientierungs- und Geschwindigkeitsinformationen. Dabei konnte die kommunizierte- mit der geschätzten Geschwindigkeit direkt verglichen werden. Für die Berechnung der wahren ego-relativen Fahrzeugpose  ${}^{\text{obj}}\mathbf{P}_{\text{ego}}$  erfolgte die Transformation zwischen der globalen Ego-pose des Folgefahrzeuges und der kommunizierten

globalen Pose vom Führungsfahrzeug. Die resultierenden Schätzfehler sind in der Tabelle 6.2 aufgelistet. Dieser ist angegeben als RMSE. Je größer dieser Wert ist, desto schlechter passt der Schätzwert zu dem Wahrheitswert. Leider konnte die modellbasierte Fahrzeugerkennung aus Manz [2013] das Führungsfahrzeug nicht auf dem gesamten Datensatz detektieren. Aus diesem Grund können lediglich die in Manz [2013] publizierten Schätzunsicherheiten als Vergleichsmaß angegeben werden.

	Ego-relative Fahrzeugpose						$v \left(\frac{\text{m}}{\text{s}}\right)$
	Rel. Position			Orientierung			
	$r \text{ (m)}$	$\alpha \text{ (}^\circ\text{)}$	$h \text{ (m)}$	$\phi \text{ (}^\circ\text{)}$	$\theta \text{ (}^\circ\text{)}$	$\psi \text{ (}^\circ\text{)}$	
$\max_{\text{Nacht}}$	32,7	37,9	0,76	0,94	2,41	84,6	20,7
$\epsilon_{\text{Nacht}}$	0,34	0,87	0,08	0,67	1,47	2,19	0,32
$\epsilon_{\text{Dammerung}}$	0,31	0,56	0,06	0,50	1,21	1,82	0,29
$\max_{\text{Tag}}$	36,1	32,7	1,34	5,68	7,44	62,8	28,1
$\epsilon_{\text{Tag}}$	0,31	0,32	0,06	0,45	1,16	1,50	0,27
$\epsilon_{\text{Tag}} _{\text{Manz, 2013, S. 189}}$	0,36	0,50	0,04	0,40	1,62	1,81	0,30

**Tabelle 6.2:**

Evaluierung der Schatzgenauigkeit: Die betrachteten Zustandswerte sind die in Zylinderkoordinaten angegebene ego-relative Fahrzeugposition  $\{r, \alpha, h\}$  zum Führungsfahrzeug sowie dessen Orientierung  $\{\phi, \theta, \psi\}$  und Geschwindigkeit  $v$ . „max“ und  $\epsilon$  kennzeichnen die maximalen Zustandswerte und den RMSE.

Abschlieend sei auf den Anhang A.6 verwiesen, in dem die vier notwendigen Arbeitsschritte zur Inbetriebnahme der Software beschrieben sind. Dort veranschaulichen diverse Grafiken dem Leser die einfache Bedienbarkeit von dem in dieser Arbeit realisierten Folgeassistenten.

## 7 Zusammenfassung und Ausblick

### Inhalt

---

7.1 Stand der Arbeit . . . . .	151
7.2 Weiterentwicklungen . . . . .	153

---

In diesem Kapitel werden die wesentlichen Ergebnisse der vorliegenden Arbeit zusammengefasst und Möglichkeiten zur Weiterentwicklung vorgeschlagen.

#### 7.1 Stand der Arbeit

Diese Arbeit stellte eine modellbasierte Fahrzeugerkennung für autonomes Fahren im Konvoi vor. Das erste Kapitel gab eine Einleitung in das Themengebiet und erläuterte die Ziele dieser Forschungsarbeit. Zum einen liegt der Fokus auf einer präzisen Erkennung eines Führungsfahrzeuges, sodass diesem spurtreu gefolgt werden kann. Zum anderen sollte das Gesamtsystem unabhängig von den Umgebungs- und Lichtverhältnissen einsatzfähig sein. Anschließend widmete sich das zweite Kapitel dem Überblick über die forschungsrelevanten Veröffentlichungen. Das dritte Kapitel ging auf die zum Verständnis dieser Arbeit notwendigen Grundlagen ein. Dazu zählen die Sensorkalibrierung und -synchronisierung, rekursive Zustandsschätzung dynamischer Systeme, kaskadierte Klassifikatoren zur Fahrzeugerkennung und *Region Growing*-Algorithmen zur Datenreduktion.

In dieser Arbeit erfolgt die Wahrnehmung von Fahrzeugen primär modellbasiert, wodurch einmalig für jedes Führungsfahrzeug ein sogenanntes 3D-Merkmalmodell erstellt werden muss. Das Kapitel 4 beschrieb eine manuelle und eine automatische Modellgenerierung. In dieser werden die markanten Textur-, Farb-, Wärme- und Hindernismerkmale für jedes Führungsfahrzeug ermittelt und in einem sogenannten 3D-Merkmalmodell zur weiteren Verwendung abgelegt. Im Anschluss ist auf Basis dieser fahrzeugspezifischen Merkmalsmodelle eine zuverlässige Fahrzeugerkennung möglich.

Eine zuverlässige Umgebungswahrnehmung sowie das autonome Folgen eines Fahrzeuges erfordert die Verwendung unterschiedlicher Sensoren. Die vier in dieser Arbeit vorgestellten Fahrzeuge besitzen eine grundlegende Sensorausstattung, die aus einem Inertialen Navigationssystem, einem Laserscanner und einer oder mehrerer Tages-

lichtkamas besteht. Diese Grundausstattung wurde für eine Fahrzeugerkennung bei Nacht mit einer Wärmebildkamera ergänzt. Optional sind Low-Light-Kameras und Radarsensoren in das System integrierbar.

Für das spurtreue Folgen eines anderen Fahrzeuges ist eine exakte Positions- und Orientierungsschätzung notwendig. Kapitel 5 beschrieb drei verschiedene Algorithmen für eine präzise Fahrzeugerkennung und dessen zeitliche Verfolgung. Der erste Algorithmus erkennt, mit den in Kapitel 3.3 vorgestellten kaskadierten Klassifikatoren, Fahrzeugseiten performant sowie rein visuell. Diese Fahrzeugdetektionen werden für die Schätzung der ego-relativen 6D-Pose zum Führungsfahrzeug in einem Unscented Kalman-Filter sequentiell „eingefiltert“. Zusätzlich ermöglicht die rekursive Zustandsschätzung die Bestimmung der Fahrzeuggeschwindigkeit und des -lenkwinkels über die Zeit. Der zweite Algorithmus verwendet für die rekursive Zustandsschätzung einen Partikel-Filter anstelle eines Kalman-Filters. Mit diesem erfolgt die rekursive Schätzung einer multimodalen Wahrscheinlichkeitsdichtefunktion (WDF). Die zu schätzenden Zustände sind die ego-relative 6D-Pose des Führungsfahrzeuges, dessen Längs- und Quergeschwindigkeit, dessen Längsbeschleunigung sowie dessen Lenkwinkel. Die Beschreibung der WDF erfolgt durch eine Vielzahl von Partikeln, die aus einem Gewicht und einer Stützstelle der WDF bestehen. Dabei entspricht ein Partikel einer Fahrzeughypothese und kann anhand der zugehörigen ego-relativen Fahrzeugpose und einem bekannten fahrzeugspezifischen 3D-Merkmalmodell durch einen Abgleich mit der Wahrnehmung gewichtet werden (siehe Abschnitt 5.1.3). Die dritte Methode ist die Kombination der beiden genannten Algorithmen und vereint deren individuelle Stärken. Dadurch kann bei einer (Re-)Initialisierung der erste Algorithmus eine grobe Fahrzeugdetektion zeiteffizient berechnen und diese dem zweiten Partikel-Filter-basierenden Algorithmus für eine schnelle (Re-)Initialisierungsphase mitteilen. Schlussendlich hängt die Verwendung des jeweiligen Algorithmus von dem Einsatzzweck ab. Darf ein Fahrzeugerkennungssystem nur wenig Rechenlast produzieren, wird der erste Algorithmus mit den kaskadierten Klassifikatoren empfohlen. Liegt dagegen die Priorität auf der Genauigkeit der Posenbestimmung, ist die zweite und dritte Variante die bessere Wahl.

Die präzisen Schätzergebnisse der vorgestellten Fahrzeugerkennungsalgorithmen ermöglichen ein spurtreues Folgen eines Führungsfahrzeuges. Kapitel 5.4 ging zur Vervollständigung dieser Arbeit kurz auf die Berechnung einer fahrbaren und hinderisfreien Folgespur und die darauffolgende Quer- und Längsregelung ein. Anschließend folgte das Ergebniskapitel, in dem die Beschreibung der Versuchsfahrzeuge, die erzielten Resultate der modellbasierten Fahrzeugerkennung und die Beurteilung über das Gesamtsystem niedergeschrieben sind.

Das Gesamtsystem wurde vielfältig und intensiv mit einem Konvoi von bis zu drei Teilnehmern evaluiert. Dabei kamen unterschiedliche Pkw und Lkw zum Einsatz, die mehrere hundert Kilometer autonom fuhren. Das Terrain war dabei primär unstrukturiert und das ADAS führte dynamische Fahraufgaben selbstständig durch.

Das echtzeitfähige Gesamtsystem zeigt sich beständig gegenüber schwierigen Witterungsbedingungen wie Regen oder Schnee und kann mit teilweisen Verdeckungen des Führungsfahrzeuges umgehen. Reagiert der Fahrer nicht auf die Aufforderung, die Fahrzeugführung zu übernehmen, parkt das in dieser Arbeit realisierte System das Fahrzeug sicher in den Stillstand. Ebenfalls nennenswert ist das spurtreue Folgen, das in dieser Arbeit bis zu einer Geschwindigkeit von 100 km/h erprobt wurde.

## 7.2 Weiterentwicklungen

An dieser Stelle werden aufbauend auf den gewonnenen Erkenntnissen Weiterentwicklungen vorgeschlagen. Zunächst wird auf die automatische Erstellung von 3D-Merkmalmodellen eingegangen. Zum einen sollte bei der Verwendung von Laserdaten deren Intensitätsinformation mit berücksichtigt werden. Hierbei zeigten Praxistests deutliche Intensitätsunterschiede zwischen dem Kennzeichen und der Fahrzeugkarosserie. Aus diesem Grund empfiehlt sich die Suche und Integration markanter Intensitätsmerkmale. Des Weiteren ist die rein visuelle Erzeugung von 3D-Merkmalmodellen mittels Stereovision zu verbessern. Derzeit ist die visuelle Tiefenbestimmung im Vergleich zu einem hochauflösenden Velodyne HDL-64E LiDAR-Sensor deutlich ungenauer. Das liegt vor allem an dem kleinen Basisabstand der Stereokameras, der derzeit 0,2 Meter beträgt. Die Messgenauigkeit verringert sich, je kleiner der Basisabstand ist. Allerdings steigt der Überlappungsbereich der zwei Kamerabilder mit einem kleineren Basisabstand. Dies wiederum ermöglicht die Messung von vielen Disparitäten. Folglich ist ein veränderbarer Basisabstand [Fanto, 2012] oder die Kombination unterschiedlicher Basisabstände [Okutomi und Kanade, 1993] empfehlenswert. Beispielsweise könnten zwei Kameras mit einem Basisabstand von 1,5 Metern für eine präzise Tiefenbestimmung zum Einsatz kommen. Ebenfalls würde eine zusätzliche zentral positionierte Kamera eine Stereobildverarbeitung mit der jeweils links- und rechtsseitigen Kamera ermöglichen.

Weiterführend könnten die Stereoalgorithmen sowohl Sensordaten von Wärmebildkameras als auch von Tageslichtkameras verwenden [Hajebi, 2007]. Die potenziellen Vorteile sind ein dichteres Disparitätenbild und mehr Robustheit gegenüber wechselhaften Lichtverhältnissen.

In der automatischen Generierung von 3D-Merkmalmodellen sind die Merkmale derzeit überwiegend vorgegeben. Das bedeutet, dass nicht alle Farben bzw. Farbbereiche evaluiert werden. Die Farbe Grün wird unter anderem nicht berücksichtigt, da sich im unstrukturierten Gelände viel Vegetation befindet. Maschinelle Lernalgorithmen könnten unabhängig von Benutzervorgaben Merkmale in Sensordaten identifizieren, welche für den Benutzer nicht erkennbar sind. Dabei sollten nicht Hunderte bis Tausende von gelabelten Referenzdatensätzen zeitaufwendig erstellt und mit Lernalgorithmen analysiert werden. Stattdessen verwendet ein maschineller Lernalgorithmus Sensordaten, in denen das Führungsfahrzeug überwiegend dargestellt ist,

und muss durch die zeitliche Betrachtung der Sensordaten das Führungsfahrzeug bzw. dessen dominante Merkmale ermitteln.

Des Weiteren sollte verfügbares Kartenwissen berücksichtigt und in das Gesamtsystem integriert werden. Eine Möglichkeit wäre die Verwendung von OpenStreet-Map<sup>1</sup> [Landsiedel und Wollherr, 2017]. Hierbei könnte die in dieser Arbeit erzeugte 2,5D-Hinderniskarte mit Kartenwissen über die Fahrzeug-umgebende Infrastruktur angereichert sowie abgeglichen werden.

---

<sup>1</sup>[www.openstreetmap.org](http://www.openstreetmap.org)

# A Anhang

## Inhalt

---

A.1	Sensorik Datenblätter . . . . .	155
A.1.1	Abstandsgebende Sensoren . . . . .	156
A.1.2	Bildgebende Sensoren . . . . .	157
A.1.3	Inertiale Sensoren . . . . .	160
A.2	Kalibrierungsergebnisse der intrinsischen Kameraparameter . . . . .	161
A.3	Farbräume . . . . .	165
A.3.1	RGB-Farbraum . . . . .	165
A.3.2	HSV-Farbraum . . . . .	166
A.3.3	Umwandlung vom RGB- in den HSV-Farbraum . . . . .	167
A.4	Einspurmodell für vorderachsgelenkte Fahrzeuge . . . . .	168
A.5	Visuelle Markererkennung . . . . .	170
A.5.1	Detektionsalgorithmus . . . . .	170
A.5.2	Bewertung . . . . .	177
A.6	Inbetriebnahme der Software . . . . .	179

---

Der Anhang dieser Arbeit geht auf die Sensorkomponenten der autonomen Fahrzeuge, Kalibrierungsergebnisse, Farbräume, ein Einspurmodell, die visuelle Markererkennung und die Softwareinbetriebnahme ein.

### A.1 Sensorik Datenblätter

In der Zusammenarbeit mit Projektpartnern und den Mitarbeitern vom Institut für Technik Autonomer Systeme wurden die vier Fahrzeuge MuCAR-3, MuCAR-4, TULF und StrAsRob zum autonomen Fahren mit diverser Sensorik bestückt [Manz, 2013, Unterholzner, 2016]. Hinsichtlich der Integration von Sensorkomponenten stellte das Kapitel 6.1 sämtliche Fahrzeuge vor und ging auf die jeweils montierte Sensorik ein. Im Nachfolgenden werden die einzelnen Sensorkomponenten im Detail beschrieben.

### A.1.1 Abstandsgebende Sensoren

In dieser Arbeit kamen LiDAR- und Radarsensoren zum Einsatz. Beide Sensortypen messen die vergangene Zeit  $\Delta t$  zwischen einem ausgesandten und einem empfangenen Signal und bestimmen aus der bekannten Lichtgeschwindigkeit  $c$  die Entfernung  $s$  zum Objekt, an dem das Signal reflektiert wurde. Die Basisformel lautet  $s = c \cdot \frac{\Delta t}{2}$ .

Die im MuCAR-3 (Smartmicro UMRR), TULF und StrAsRob (Continental ARS 300) verwendeten Radarsensoren sind frequenzmodulierte Dauerstrichradargeräte, welche die Frequenz der ausgesendeten elektromagnetischen Wellen kontinuierlich und rampenförmig ändern. Anhand der messbaren Frequenzänderung zwischen dem ausgesendeten und empfangenen Signal kann der Abstand und die Relativgeschwindigkeit zwischen dem Sensor und dem Objekt ermittelt werden (Doppler-Effekt).

LiDAR-Sensoren sind in allen vier autonomen Fahrzeugen verbaut. In MuCAR-3 und MuCAR-4 befindet sich auf dem Dachträger ein einzelner Velodyne HDL-64E S2. Dieser besitzt 64 Laserdioden, welche für eine erweiterte Umgebungsabtastung um 360 Grad rotiert werden. Dagegen wurden zwei kleinere Velodyne HDL-32E am TULF montiert, welche über jeweils 32 Laserdioden verfügen, die ebenfalls um 360 Grad rotiert werden. Die Montage der HDL-32E erfolgte unterhalb der Lkw-Spiegel, was die sensorielle Beobachtung der vorderen und seitlichen Lkw-Umgebung ermöglicht. Noch eine Stufe kleiner ist der Ibeo LUX 8L, der sich an der Fahrzeugfront des StrAsRob-Lkws und zu Testzwecken an MuCAR-3 befindet. Dieser besitzt lediglich eine Laserdiode, die mit einem Spiegel acht Umgebungsebenen abtastet. Weitere Informationen über die Laserscanner können aus der Tabelle A.1 entnommen werden.

	Velodyne HDL-64E S2 	Velodyne HDL-32E 	Ibeo LUX 8L 
Laserdioden / Empfänger	64 / 64	32 / 32	1 / 1
Wellenlänge	905 nm		
Hori. / Vert. Sichtbereich	360° / 26,8°	360° / 41,3°	110° / 6,4°
Hori. / Vert. Auflösung	0,17° / 0,3–0,5° bei 10 Hz	0,1–0,4° / 1,33°	0,125° / 0,8°
Reichweite	120 m	100 m	200 m
Remission	-0,1 bei 50 m   -0,8 bei 120 m	-	
Messgenauigkeit	≤ 2 cm (1σ)	± 2 cm (1σ bei 25 m)	10 cm
Update rate	5-20 Hz	5-20 Hz	6,25 / 12,5 / 25 Hz
Messwerte pro Sekunde	≤ 1,33 · 10 <sup>6</sup>	≤ 7 · 10 <sup>5</sup>	-
Dimensionierung (B × H)	22 × 26 cm <sup>2</sup>	9 × 15 cm <sup>2</sup>	17 × 9 cm <sup>2</sup>
Anschaffungspreis	~80 000 €	~35 000 €	~16 000 €
Fahrzeug	MuCAR-3 + MuCAR-4	TULF	StrAsRob + MuCAR-3

**Tabelle A.1:**

Sensoreigenschaften der in dieser Arbeit zum Einsatz kommenden LiDAR-Sensoren.

### A.1.2 Bildgebende Sensoren

Alle vier selbstfahrenden Fahrzeuge wurden mit Kameras zur visuellen Umgebungserfassung bestückt. Da der Anschaffungspreis von Tageslichtkameras im Vergleich zu LiDAR-Sensoren und Wärmebildkameras deutlich geringer ist, erfolgte die Integration von mehreren Kameras mit Weitwinkel- und Teleobjektiven. Hierbei lag der Anschaffungspreis einer Tageslichtkamera zwischen 500 € bis 2000 €. Aufgrund der industriellen Weiterentwicklung der Kamerasensoren erfolgte im Jahr 2013 die Beschaffung von neuen Gigabit-Ethernet Farbkameras, welche die bestehenden FireWire-Kameras der bewegbaren Kameraplattform MarVEye-8 von MuCAR-3 ersetzen. Darunter zählen zwei hochauflösende Basler acA2040-25gc in Stereoanordnung und eine Basler acA1300-30gc. Die zwei Kameras in Stereoorientierung haben eine sehr große horizontale und vertikale Auflösung von  $2048 \times 2048$  Pixel und sind zusammen mit einem Weitwinkelobjektiv zur nahen Umgebungserfassung gedacht. Dagegen ist der Anwendungsbereich der Basler acA1300-30gc die Objekterkennung in der Ferne. Dafür kommt ein Teleobjektiv zum Einsatz. Damit sind ausreichend Pixelinformationen für eine zuverlässige Objekterkennung auch bei Entfernungen von über 70 m im Kamerabild vorhanden. Der Nachteil eines solchen Teleobjektives liegt in dem kleinen Sichtbereich - in diesem Fall  $26 \times 21$  Grad. Allerdings ist der Sichtbereich in der Gierachse durch die Verwendung einer Kameraplattform ausreichend veränderbar, was die visuelle Verfolgung eines Führungsfahrzeuges auch bei engen Kurven ermöglicht. Des Weiteren schaut die Kamera auf einen Spiegel, wodurch der Nickwinkel geändert werden kann. Dies ist vor allem bei Fahrten im unstrukturiertem Gelände nützlich, da ansonsten die Kamera beim Durchfahren von tiefen Schlaglöchern zum Teil in den Boden hinein schaut und damit das Führungsfahrzeug nicht im Sichtbereich hat. Hier ist eine inertielle Stabilisierung der Nickachse durch Kreisel empfehlenswert.

Für eine zuverlässige Umgebungswahrnehmung bei Dämmerungs- und Nachtfahrten wurden einerseits Wärmebild- und andererseits Low-Light-Kameras erprobt und bewiesen sich als nützlich. In MuCAR-3 wurde die hochwertige Wärmebildkamera FLIR A615 in ein wetterbeständiges Schutzgehäuse auf dem Dachträger installiert. Die Kamera besitzt einen ungekühlten Mikrobolometer und liefert absolute Temperaturinformationen. Die Auflösung beträgt  $640 \times 480$  Pixel mit einer hohen Pixelwerttiefe von 16 Bit. Im TULF und StrAsRob wurde eine andere Wärmebildkamera vom Hersteller Xenics integriert. Die Temperaturmessung der Kamera ist im Vergleich zur FLIR A615 nicht kalibriert, wodurch dem Benutzer lediglich relative Messwerte für die Bildverarbeitung zur Verfügung stehen. Ein weiterer Kamertyp ist die sogenannte Low-Light-Farbkamera, die in MuCAR-3 hinter die Frontscheibe montiert wurde. Die Kamera liefert auch bei Dämmerung und zum Teil bei Nachtfahrten Farbinformationen von der Fahrzeugumgebung. Die Kamera verstärkt das auf den Sensorchip eintreffende Licht und ermöglicht auch bei dunklen Lichtverhältnissen die Aufnahme eines leicht bis stark verrauschten Farbbildes.

## A Anhang

Abschließend gibt die Tabelle A.2 einen Auszug aus den Datenblättern der genannten Wärme-, Low-Light- und Tageslichtfarbkameras. Auf früher benutzte Tageslichtkameras wird an dieser Stelle nicht eingegangen. Des Weiteren ist die Aufzählung der aktuell verwendeten Tageslichtkameras nicht vollständig. Grundsätzlich kann allerdings ausgesagt werden, dass die Kameratypen aus den vier selbstfahrenden Fahrzeugen den Basler acA1300-30gc und acA2040-25g ähnlich sind.

	Sensortyp					
	Wärmebildkameras		Low-Light-Farbkamera	Tageslichtfarbkameras		
	FLIR A615	Xenics Gobi-640	FLIR TAU CNV	Basler acA1300 30/60gc	Basler acA2040-25gc	
						
Auflösung (B × H)	640 × 480		1280 × 960	1296 × 964	1280 × 1024	2048 × 2048
Chiptyp	Ungekühlter Mikrobolometer (a-Si)		CMOS	CCD Sony ICX445	CMOS e2v EV76C560	CMOS CMOSIS CMV4000
Chipgröße	1"		2/3"	1/3"	1/1,8"	1"
Pixelgröße	17 μm <sup>2</sup>		6,5 μm <sup>2</sup>	3,75 μm <sup>2</sup>	5,3 μm <sup>2</sup>	5,5 μm <sup>2</sup>
Pixelwerttiefe	16 Bit			12 Bit		
Bilder pro Sekunde	50		30		60	25
Shutter	Rolling			Global		
Dimensionierung (B × H × L)	7 × 8 × 23 cm <sup>3</sup>	5 × 5 × 8 cm <sup>3</sup>	5 × 5 × 7 cm <sup>3</sup>	4 × 3 × 3 cm <sup>3</sup>		
Schnittstelle	GigE					
Spektralbereich	7,5-14 μm	8-14 μm	-			
Messgenauigkeit	± 2 °C		-			
Anschaffungspreis	~15 000 €	~9000 €	~8000 €	~500 €		~2000 €
Fahrzeug	MuCAR-3	TULF + StrAsRob	MuCAR-3	MuCAR-3 + MuCAR-4		MuCAR-3 + TULF
Objektiv, FOV	45 ° × 34 °	56 ° × 44 °	Weitwinkel 74 ° × 58 °	57 ° × 44 °	Tele 26 ° × 21 °	Weitwinkel 68 ° × 68 °

**Tabelle A.2:** Auflistung der in dieser Arbeit verwendeten Kameras.

### A.1.3 Inertiale Sensoren

Die Messung der Fahrzeugbeschleunigung wird für die Eigenbewegungsschätzung verwendet. In dieser Arbeit kamen zwei unterschiedliche inertielle Sensoren zum Einsatz. In den Institutsfahrzeugen MuCAR-3 und MuCAR-4 wurde die hochwertige GNSS-gekoppelte Oxford OxTs RT3003 eingebaut. Die IMU besteht aus einem triaxialen Beschleunigungs- und Drehratensensor und misst damit sechs Größen direkt. Diese und weitere Größen, wie die Position, Orientierung und Geschwindigkeit des Fahrzeuges, werden dabei hochpräzise über die Zeit ermittelt. Einen Auszug aus dem Datenblatt listet die Tabelle A.3 auf.

Im TULF und StrAsRob befindet sich ein kostengünstiger Sensor der Firma Xsens, welcher ebenfalls in der Tabelle aufgelistet ist. Dieser misst die Beschleunigungswerte und Rotationsgeschwindigkeiten mit einer deutlich schlechteren Präzision, der Unterschied im Anschaffungspreis ist allerdings erheblich. Ergänzend wurde im TULF der Radarbasierte Geschwindigkeitssensor MSO SpeedWedge im Bereich der Fahrzeughinterachse montiert. Beim StrAsRob erfolgte die Integration des Radarbasierten Übergrund-Geschwindigkeitssensors DICKEY-john Radar III, welcher eine höhere Messgenauigkeit aufweist. Zur Verbesserung der Eigenbewegungsschätzung beim StrAsRob wurde zusätzlich zur Xsens MTi-G-700 INS ein einachsiger faseroptischer Kreisel vom Typ KVH DSP-3000 integriert.

	Sensortyp	
	Oxford OxTs RT3003	Xsens MTi-G-700
		
Typ	IMU + GNSS	IMU + GPS
Positionsmessgenauigkeit	0,02 m ( $1\sigma$ DGPS)	-
Roll-/Nickwinkelgenauigkeit	$0,03^\circ$ ( $1\sigma$ )	$\leq 0,5^\circ$
Gierwinkelgenauigkeit	$0,03^\circ$ ( $1\sigma$ )	$\leq 1^\circ$
Beschleunigungsmessgenauigkeit	$10 \text{ mm/s}^2$ ( $1\sigma$ )	$0,02 \text{ m/s}^2$
Messabweichung der Rotationsgeschwindigkeit	$2^\circ/\text{h}$	$20^\circ/\text{h}$
Geschwindigkeitsschätzungsgenauigkeit	$0,05 \text{ km/h RMSE}$	-
Latenzzeit	0,5 ms	$\leq 2 \text{ ms}$
Frequenz	100 Hz	256 Hz
Anschaffungspreis	~50 000 €	~3500 €
Fahrzeug	MuCAR-3 + MuCAR-4	TULF + StrAsRob

**Tabelle A.3:**

Inertialsensoren: (a) Hochwertige Oxford OxTs RT3003. (b) Preiswertere Xsens MTi-G-700.

## A.2 Kalibrierungsergebnisse der intrinsischen Kameraparameter

In diesem Kapitel werden die ermittelten intrinsischen Kameraparameter von zwei Wärmebild-, einer Low-Light- und zwei Tageslichtfarbkameras aufgelistet. Die intrinsischen Kameraparameter wurden im Grundlagenkapitel 3.1.1.2 zusammen mit der Ent- und Verzerrung von Bildpunkten beschrieben.

Zu den intrinsischen Kameraparametern gehören zum einen die in Pixel angegebene horizontale und vertikale Bildweite  $\mathbf{f} = [f_u, f_v]$  und der Bildhauptpunkt  $\mathbf{c} = [c_u, c_v]$ . Zum anderen existieren fünf Koeffizienten  $\mathbf{k}_{1,\dots,5}$  zur Beschreibung der Linsenverzerrung sowie ein Scherungsfaktor  $s$ , der bei quadratischen Pixeln eins ist.

Die intrinsischen Kameraparameter werden vom Hersteller auf eine Baureihe bezogen angegeben und sind damit für die einzelnen Kamerasensoren fehlerbehaftet. Da die Verwendung von fehlerhaften Kameraparametern zu einer fehlerhaften Abbildung zwischen dem Kamera- und Bildkoordinatensystem führen, wirkt sich dies implizit auf die in dieser Arbeit zum Einsatz kommende visuelle Fahrzeugerkennung aus. Aus diesem Grund werden die intrinsischen Kameraparameter typischerweise durch die Aufnahme eines bekannten Kalibrierungsmusters fehlerfrei ermittelt. Für eine detaillierte Beschreibung über die Kalibrierungsmethode mit einem bekannten Muster wird auf die S. 25 im Grundlagenkapitel 3.1.1.2 verwiesen.

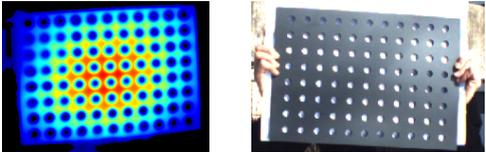
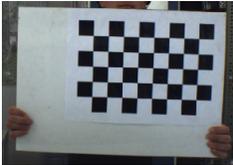
Für die Kalibrierung der Tageslicht- und Low-Light-Kamera wurde ein Schachbrettmuster mit  $9 \cdot 7 = 63$  Quadraten verwendet. Dagegen konnte das Schachbrettmuster nicht für die Kalibrierung von Wärmebildkameras verwendet werden. Die weißen und schwarzen Quadrate besaßen keinen messbaren Temperaturunterschied, sodass die Detektion der Eckpunkte nicht möglich war. Aus diesem Grund wurde eine Platte aus Hartplastik zugeschnitten und ein Kreismuster hinein gebohrt. Dabei erfolgte die Fertigung automatisiert und mit einer hohen Präzision durch eine CNC-Maschine. Anschließend konnte die Hartplastikplatte für die Kalibrierung einer Wärmebildkamera verwendet werden. Dafür wurde die Platte gleichmäßig erhitzt, sodass die Platte im Wärmebild signifikant wärmer ist als die Löcher. Nach dem Erhitzen erfolgten circa 30 Wärmebilddaufnahmen von der Hartplastikplatte. Dabei wurde darauf geachtet, dass die Aufnahmen aus unterschiedlichen Perspektiven im Raum aufgenommen werden und dass das Kreismuster in jedem Teilbereich des Kamerabildes mindestens einmal zu sehen ist. Dieses Vorgehen erhöhte die Kalibrierungsgenauigkeit signifikant. Im Vergleich zum Schachbrettmuster sind beim Kreismuster die einzelnen Kreismittelpunkte mit Bildverarbeitungsalgorithmen zu detektieren. Anschließend sind mit der bekannten Kreismustergeometrie die intrinsischen Kameraparameter bestimmbar.

Die Kalibrierungsergebnisse sind in der Tabelle A.4 aufgelistet. Die Herstellerangabe, dass der Bildhauptpunkt im Bildmittelpunkt liegt ist falsch. Wird zum Beispiel die Wärmebildkamera FLIR A615 betrachtet, dann müsste sich der Bildhauptpunkt folglich bei  $[320; 240]$  befinden. Dagegen ergab die genauere Kalibrierung einen Bildhauptpunkt bei  $[323; 255]$ , was eine Abweichung von  $[3; 15]$  Pixeln beträgt. Aus diesem

Grund sind die Herstellerangaben lediglich als Startwert für einen Kalibriervorgang zu verwenden und Bildverarbeitungsalgorithmen bzw. Abbildungsmodellierungen sollten mit eigens kalibrierten Werte Berechnungen durchführen.

Neben der Bestimmung des Bildhauptpunktes  $c$  und der Bildweite  $f$  muss auch die Art und Stärke der Linsenverzeichnung bestimmt werden. Insbesondere bei Weitwinkelobjektiven ist die Bildverzerrung sehr stark und die Verzerrungskoeffizienten  $k$  sind für eine exakte Abbildungsbeschreibung zwingend notwendig. Im TULF und StrAsRob wurde z. B. eine zentral nach vorne ausgerichtete Tageslichtkamera mit einem Weitwinkelobjektiv verwendet. Die Bilder weisen eine tonnenförmige Verzeichnung auf, welche in den Bildverarbeitungsalgorithmen bzw. Abbildungsgleichungen berücksichtigt werden muss.

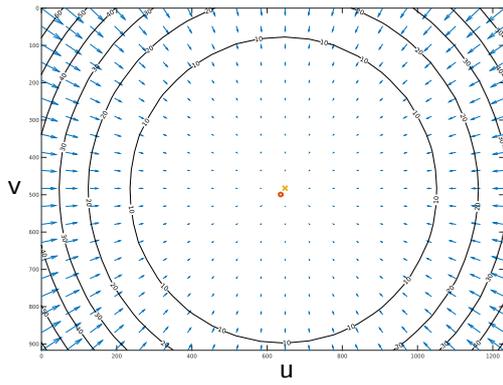
Abschließend wird in der Abbildung A.1 der ermittelte Bildhauptpunkt  $c$  und die tangentielle und radialsymmetrische Verzeichnung eines Weitwinkelobjektives veranschaulicht. In der Grafik ist die Abweichung zwischen dem ermittelten Bildhauptpunkt und dem Bildmittelpunkt deutlich erkennbar. Dabei ist der Bildmittelpunkt mit einem gelben Kreuz und der Bildhauptpunkt durch einen roten Kreis gekennzeichnet. Die Intensität der tangentialen und radialen Verzerrung des Weitwinkelobjektives ist durch blaue Pfeile gekennzeichnet. Die Pfeile veranschaulichen die Stärke und Richtung der jeweiligen Verzerrung in einer Bildregion. Dabei ist die zum Bildrand stärker werdende Verzerrung deutlich erkennbar.

	Wärmebildkamera		Low-Light-Farbkamera	Tageslichtfarbkamera	
	FLIR A615	Xenics Gobi-640	FLIR TAU CNV	Basler acA1300-60gc + Teleobjektiv	Basler acA2040-25gc
Kalibrierungsmuster	Erhitztes Kreismuster 			Schachbrett 	
Auflösung	640 × 480		1280 × 960	1280 × 1024	2048 × 2048
Bildhauptpunktannahme $\mathbf{c} = [c_u; c_v]$	[320 ; 240]		[640 ; 480]	[640 ; 512]	[1024 ; 1024]
Messung des Bildhauptpunktes $\mathbf{c} = [c_u; c_v]$	[323 ; 255]	[319 ; 247]	[674 ; 465]	[574 ; 545]	[1021 ; 1020]
Messung der Bildweite $[\mathbf{f}_u; \mathbf{f}_v]$	[800 ; 801]	[597 ; 593]	[753 ; 760]	[4909 ; 4883]	[1502 ; 1509]
Messung der Verzeichnung $\mathbf{k}_{1,\dots,5}$	[-0,19 ; 0,27 ; 0 ; 0 ; 0,43]	[-0,49 ; 0,32 ; 0 ; 0 ; -0,12]	[-0,26 ; 0,13 ; 0 ; 0 ; -0,03]	[-0,41 ; 0 ; 0 ; 0 ; 0]	[-0,19 ; 0,13 ; 0 ; 0 ; 0,03]
Scherungskoeffizient $s$			1		

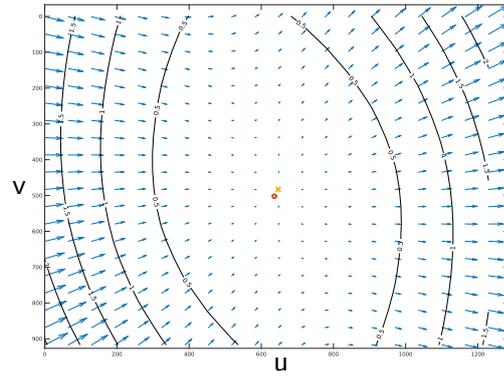
**Tabelle A.4:**

Ermittelte intrinsische Kameraparameter: Bildhauptpunkt  $\mathbf{c}$ , horizontale sowie vertikale Bildweite  $\mathbf{f}$  und radialsymmetrische sowie tangentielle Verzerrungskoeffizienten  $\mathbf{k}$ .

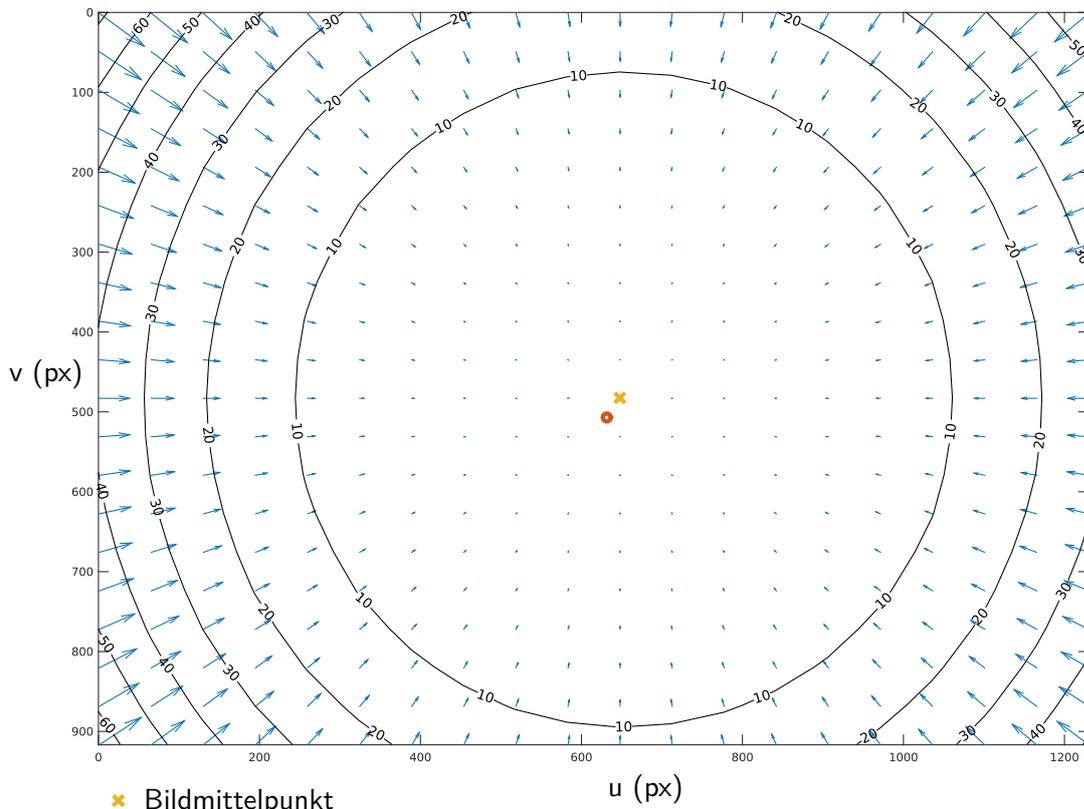
## A Anhang



(a) Tangentiale Verzerrung



(b) Radiale Verzerrung



- ✕ Bildmittelpunkt
- Gemessener Bildhauptpunkt
- ⇨ Verzerrungsstärke und -richtung

(c) Tangentiale und Radiale Linsenverzerrung

### Abbildung A.1:

Intrinsische Kameraparameter: Bildhauptpunkt, tangentielle und radiale Verzeichnung.

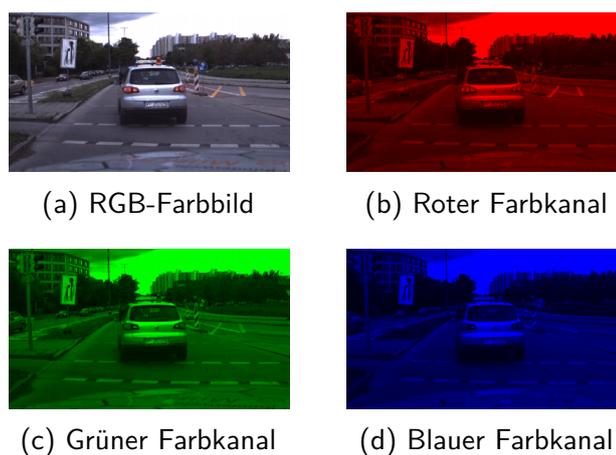
## A.3 Farbräume

Dieses Kapitel beschreibt zwei gängige Farbräume im Bereich der Bildverarbeitung. Zum einen wird der RGB-Farbraum vorgestellt, da die Bilddaten meistens in diesem Format vorliegen. Zum anderen geht dieses Kapitel auf den sogenannten HSV-Farbraum ein, welcher zur Filterung nach einer bestimmten Farbe gut geeignet ist.

### A.3.1 RGB-Farbraum

Die meisten Kameraaufnahmen werden im **RGB**-Format dargestellt, was den RGB-Farbraum zum bekanntesten Farbraum macht. In diesem erfolgt die Farbdarstellung durch die Kombination der drei Grundfarben Rot, Grün und Blau. In der Bildverarbeitung besitzt ein Pixelwert somit einen **R**ot-, **G**rün- und **B**lauanteil. Der Wertebereich liegt dabei zwischen 0 und 255.

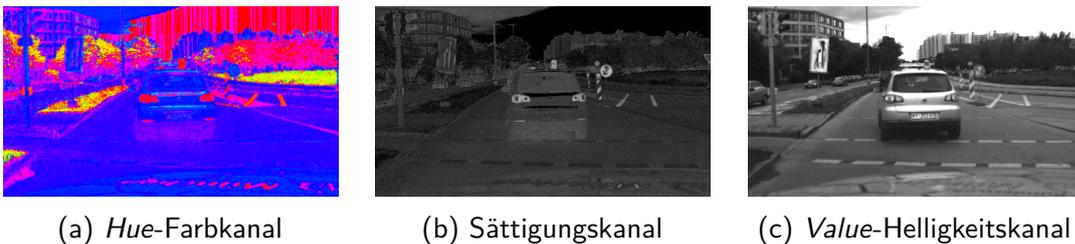
Für den Menschen bzw. technische Darstellungen sind Kameraaufnahmen im RGB-Farbraum für visuelle Zwecke gut geeignet, allerdings ist dieser Farbraum zur Farbfilterung nicht optimal. Dies veranschaulicht die nachfolgende Abbildung A.2, welche ein RGB-Farbbild einer Verkehrsszene und dessen einzelne Farbkanäle darstellt. Es ist gut erkennbar, dass die Suche nach einer bestimmten Farbe nicht direkt aus den einzelnen Farbkanälen abgelesen werden kann. Ein Beispiel ist die Suche nach den roten Rücklichtern eines Verkehrsfahrzeuges. Hierfür ist der rote Farbkanal eines RGB-Bildes ungeeignet (siehe Abbildung A.2b). Aus diesem Grund wird im nächsten Abschnitt ein Farbraum erläutert, welcher für eine Farbfilterung besser geeignet ist.



**Abbildung A.2:** Visualisierung des RGB-Farbraumes und dessen einzelner Farbkanäle.

### A.3.2 HSV-Farbraum

Der **HSV**-Farbraum besteht aus drei Komponenten, dem Farbton (engl. *Hue*), der Farbsättigung (engl. *Saturation*) und dem Helligkeitswert (*Value*). Die Wertebereiche der einzelnen Komponenten können folgendermaßen definiert werden. Der Farbton wird durch einen Farbkreis abgebildet und besitzt damit einen Wertebereich von 0 bis 359 ( $^{\circ}$ ). Dagegen wird die Sättigung und die Helligkeit in Prozent oder normiert auf den Bereich zwischen 0 und 1 angegeben. In dieser Arbeit wird mit dem normierten Wertebereich gearbeitet. Die Abbildung A.3 visualisiert die jeweiligen drei Komponenten des HSV-Farbraumes einer Verkehrsszene.

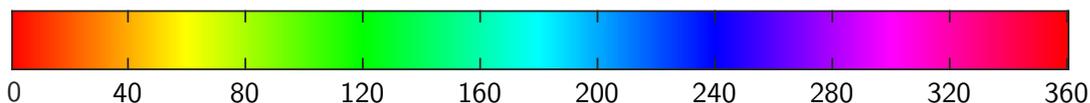


**Abbildung A.3:** Visualisierung des HSV-Farbraumes und dessen einzelne Kanäle.

Der HSV-Farbraum eignet sich aufgrund der isolierten Farbtonkomponente gut für die Filterung nach einer bestimmten Farbe. Dies verdeutlicht die Abbildung A.4, in der deutlich zu erkennen ist, dass lediglich zwei Abfragen zur Filterung nach einem Farbton erforderlich sind. Zusätzlich erfolgt jeweils eine Abfrage, ob der Farbwert eine ausreichende Sättigung und Helligkeit besitzt. Somit könnte die Suche nach roten Rücklichtern durch die nachfolgenden Abfragen realisiert werden:

- Farbton  $\leq 40$  und  $\geq 320$  bei einem Wertebereich von  $[0 \dots 360)(^{\circ})$
- Farbsättigung  $\geq 0,36$  bei einem Wertebereich  $[0 \dots 1]$
- Farbhelligkeit  $\geq 0,36$  bei einem Wertebereich  $[0 \dots 1]$

Weiterführend sei auf Gonzalez und Woods [2008] hingewiesen, die einen allgemeinen Überblick über vorhandene Farbräume geben und auf diese ausführlich eingehen.



**Abbildung A.4:** Farbtonskala vom HSV-Farbraum.

### A.3.3 Umwandlung vom RGB- in den HSV-Farbraum

Dieser Abschnitt beschreibt eine Methode zur Konvertierung eines RGB-Farbpixels  $(u, v) = [R, G, B]$  in den HSV-Raum  $(u, v) = [H, S, V]$ . Die Konvertierungsart zwischen den Farbräumen ist an Gonzalez und Woods [2008] angelehnt. Als Vorbedingung gilt, dass die R, G und B Farbwerte im normierten Intervall zwischen 0 und 1 liegen.

Zu Beginn ist jeweils der kleinste und größte Farbwert bzw. Farbkanal für jeden Pixel zu bestimmen:

$$MIN = \min(R, G, B) \quad (A.1)$$

$$MAX = \max(R, G, B) \quad (A.2)$$

Der resultierende Maximalwert  $MAX$  entspricht dem Intensitätswert  $I$  und die Sättigung  $S$  lässt sich mit den einzelnen RGB-Farbkomponenten und deren Minimalwert  $MIN$  berechnen:

$$S = 1 - \frac{3}{R + G + B} \cdot MIN \quad (A.3)$$

Die Berechnung des Farbtons  $H$  ist durch die Gleichungen A.4 und A.5 möglich und deutlich rechenaufwendiger als die Berechnung der Sättigung und Farbintensität.

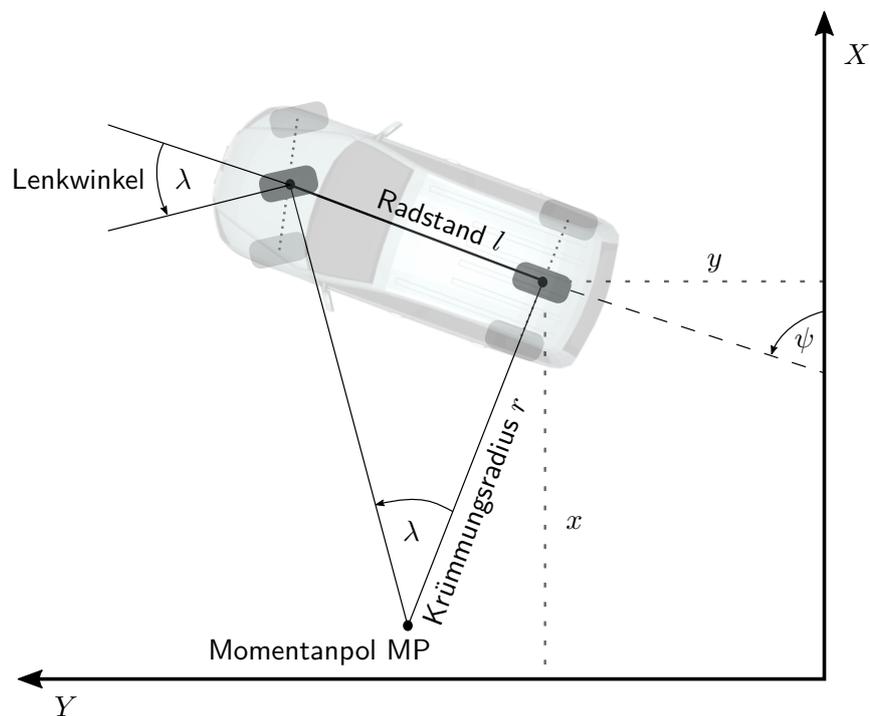
$$\theta = \cos^{-1} \left( \frac{\frac{1}{2} \cdot ((R - G) + (R - B))}{((R - G)^2 + (R - B) \cdot (G - B))^{1/2}} \right) \quad (A.4)$$

$$H = \begin{cases} \theta, & \text{wenn } B \leq G \\ 360 - \theta, & \text{wenn } B > G \end{cases} \quad (A.5)$$

#### A.4 Einspurmodell für vorderachsgelenkte Fahrzeuge

Die Dynamik eines Fahrzeuges wird durch Bewegungsmodelle beschrieben. Dabei unterscheiden sich die Modelle in der Modellierung von Fahrzeuggrößen wie bspw. der Lenk- und Schwimmwinkel. In den meisten Fällen wird die Modellierung der Fahrzeugbewegung aus Performance- und Beobachtbarkeitsgründen stark vereinfacht. In dieser Arbeit wird ein kinematisches Einspurmodell für vorderachsgelenkte Fahrzeuge verwendet, das sogenannte *Ackermann Steering Model*. Es beschreibt die zeitlich bedingte Bewegung eines Fahrzeuges, fasst Räder einer Fahrzeugachse zu einem gemeinsamen Rad zusammen und berücksichtigt unter anderem nicht den Seitenlaufwinkel oder den Schwimmwinkel. Im Nachfolgenden wird der mathematische Zusammenhang der Fahrzeuggrößen im Ackermann-Einspurmodell betrachtet und die Gleichungen zur Schätzung des Fahrzeuglenkwinkels sowie dessen Gierrate erläutert. Dabei wird der Fahrzeugschwerpunkt illustrativ im Mittelpunkt der Hinterachse modelliert und eine bekannte Fahrzeuggeschwindigkeit  $v$  angenommen.

Die Abbildung A.5 zeigt das Ackermann-Einspurmodell und veranschaulicht die Drehbewegung des Fahrzeugschwerpunktes um den sogenannten Momentanpol MP.



**Abbildung A.5:**

Einspurmodell eines vorderachsgelenkten Fahrzeuges mit einem Schwerpunkt auf der Hinterachse. Dargestellte Größen sind der Radstand  $l$ , der Lenkwinkel  $\lambda$ , der Krümmungsradius der Hinterachse  $r$ , der Momentanpol MP, der Fahrzeuggierwinkel  $\psi$  und die mittlere Position der Fahrzeughinterachse  $[x, y]$ .

Der Abstand zwischen den zwei Fahrzeugachsen wird als Radstand  $l$  bezeichnet und als bekannt vorausgesetzt. Dagegen ist der sogenannte Krümmungsradius  $r$  unbekannt. Er steht allerdings mit dem bekannten Achsabstand und dem über die Zeit schätzbaren Lenkwinkel  $\lambda$  in Beziehung. Die Gleichung A.6 zeigt diese trigonometrische Beziehung, die für rechtwinklige Dreiecke gilt.

$$r = \frac{l}{\tan(\lambda)} \quad (\text{A.6})$$

Die Gierrate  $\dot{\psi}$  eines Fahrzeuges lässt sich anhand der Fahrzeuggeschwindigkeit  $v$  und des Krümmungsradius der Hinterachse  $r$  bestimmen:

$$\dot{\psi} = \frac{v}{r} \quad (\text{A.7})$$

Da der Krümmungsradius nicht direkt gemessen werden kann, ist die Gleichung A.6 in A.7 einzusetzen:

$$\dot{\psi} = \frac{v \cdot \tan(\lambda)}{l} \quad (\text{A.8})$$

Für eine ausführliche Beschreibung über die Modellierung der Fahrzeugdynamik sei auf Mitschke und Wallentowitz [2004] verwiesen. Weiterführend sind dort auch der Schwimm- und Schräglaufwinkel beschrieben, welche in dieser Arbeit nicht berücksichtigt werden.

Der Vollständigkeit halber wird abschließend darauf eingegangen, wie eine Fahrzeugbewegung mit dem vorgestellten Einspurmodell und einer Transformationsmatrix  ${}^{\text{obj}_k}\mathbf{H}_{\text{obj}_{k-1}}$  beschrieben werden kann. Hierbei erfolgt die Annahme, dass sich ein Fahrzeug mit konstantem Lenkwinkel  $\lambda$  sowie konstanter Geschwindigkeit  $v$  auf einer stationären Kreisfahrt bewegt. Gesucht ist die homogene Transformationsmatrix  ${}^{\text{obj}_k}\mathbf{H}_{\text{obj}_{k-1}}$ , welche die Positions- und Orientierungsänderung vom diskreten Zeitpunkt  $k-1$  zu  $k$  beschreibt. Die Gleichung A.9 zeigt die Berechnung von  ${}^{\text{obj}_k}\mathbf{H}_{\text{obj}_{k-1}}$ , in der eine stationäre Kreisfahrt durch eine Bewegung auf der Kreissekanten angenähert wird.

$${}^{\text{obj}_k}\mathbf{H}_{\text{obj}_{k-1}} = \overbrace{\mathbf{R}_{\psi}(0,5 \cdot \dot{\psi} \cdot \Delta t) \cdot \mathbf{t}(v_x \cdot \Delta t, v_y \cdot \Delta t, 0)}^{\text{Bewegung auf Kreissekante}} \cdot \overbrace{\mathbf{R}_{\psi}(0,5 \cdot \dot{\psi} \cdot \Delta t)}^{\text{Vollendung der Drehung}} \quad (\text{A.9})$$

Hierbei erfolgt bezüglich dem Zeitraum  $\Delta t$  eine Annäherung der Fahrzeugbewegung durch zwei Rotationen und einer Translation.  $\Delta t$  entspricht der zeitlichen Dauer zwischen den diskreten Zeitpunkten  $k-1$  und  $k$ . Die Rotationsmatrix  $\mathbf{R}_{\psi}(0,5 \cdot \dot{\psi} \cdot \Delta t)$  dreht das Fahrzeug um den halben Gierwinkel auf die Sekante eines Kreissegmentes. Anschließend erfolgt die Translation  $\mathbf{t}(v_x \cdot \Delta t, v_y \cdot \Delta t, 0)$  entlang der Kreissekante. Abschließend findet die Durchführung der zweiten Hälfte der Fahrzeugrotation statt, in dem die Drehbewegung  $\mathbf{R}_{\psi}(0,5 \cdot \dot{\psi} \cdot \Delta t)$  erneut durchgeführt wird.

## A.5 Visuelle Markererkennung

In diesem Kapitel wird ein performantes Verfahren zur visuellen Erkennung eines bekannten Markers vorgestellt. Die Beschreibung des Algorithmus erfolgt in Anlehnung an die Erkennung eines symmetrischen Raute-Markers. Dieser ist in der Abbildung A.6 dargestellt, besteht aus einer weißen Grundfläche, einer mittig aufgeklebten schwarze Raute und vier Befestigungsbändern.



(a)



(b)

**Abbildung A.6:**

Visuelle Kennzeichnung von militärischen Lkws durch  $50 \times 50$  cm große Raute-Marker.

### A.5.1 Detektionsalgorithmus

Der Algorithmus zur Erkennung eines bekannten Markers ist kompatibel mit mehreren Kameras und benötigt Farb- oder Grauwertbilder als Eingabe. Insgesamt müssen für jedes Kamerabild die im Ablaufdiagramm A.7 dargestellten fünf Verarbeitungsphasen durchgeführt werden. Die kompletten Phasen sind parallel ausführbar, sodass die Verarbeitung von  $n$  Kamerabildern in  $n$  Threads erfolgen kann.

Im Nachfolgenden wird zum einen auf die Grundschrte der visuellen Markererkennung und des Weiteren auf die optionale Berechnung der ego-relativen 3D-Markerposition eingegangen.

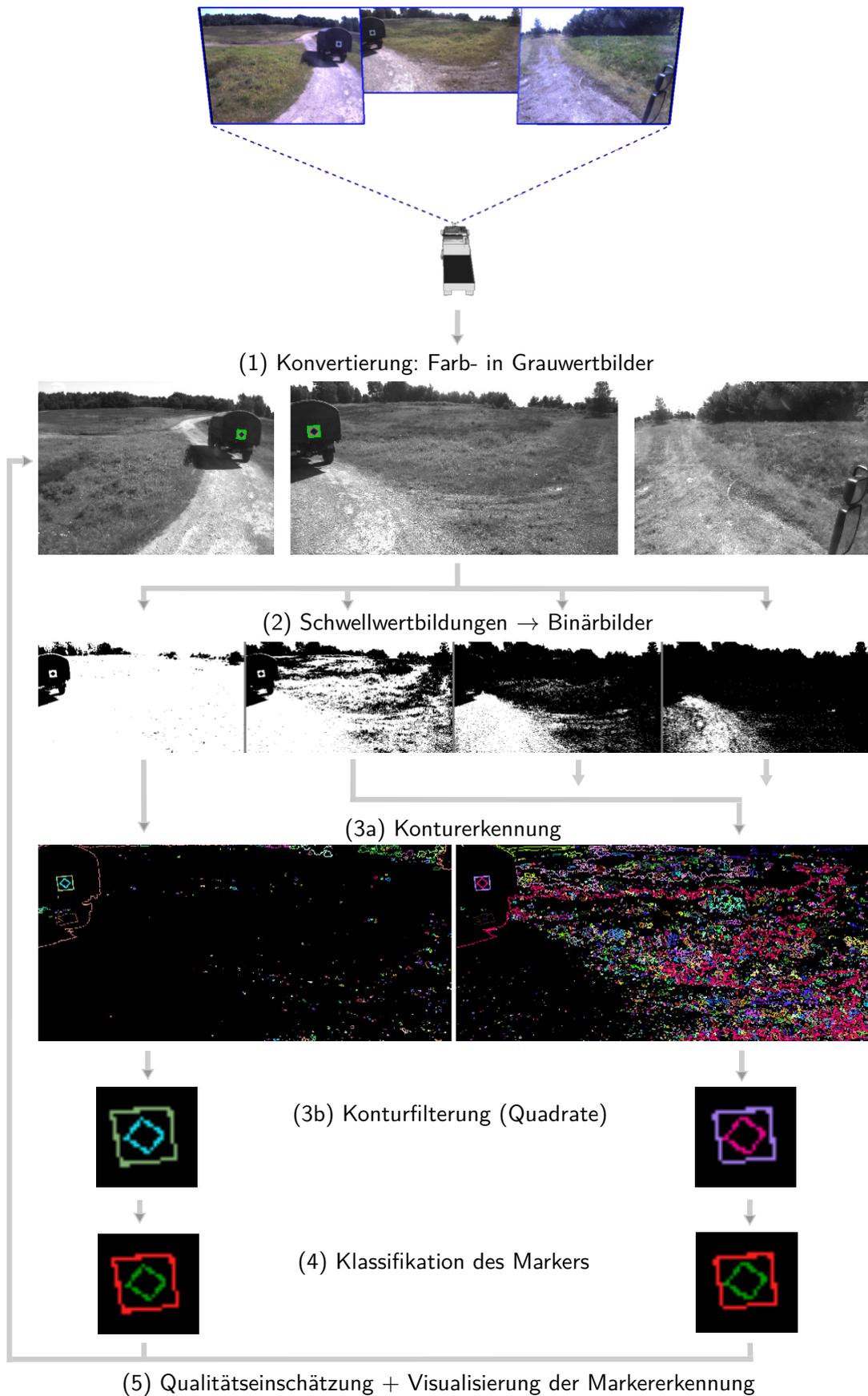


Abbildung A.7: Ablaufdiagramm der visuellen Markererkennung.

**(1a) Optionale Berücksichtigung von Vorwissen:** Die Bildverarbeitung von einem Kamerabild kann oftmals auf einen Teilbereich reduziert werden. Zum Beispiel hat eine nebenläufige Applikation bereits das Fahrzeug - an dem sich der Marker befindet - erkannt und dessen ego-relative 3D-Position ermittelt. In diesem Fall kann diese zusätzliche Information für eine effizientere Markersuche in einem kleineren Bildbereich verwendet werden. Ein solcher reduzierter Auswertungsbereich wird als sogenannter Interessensbereich, engl. *Area of Interest* (AOI) bezeichnet und ist in diesem Fall durch zwei Schritte ermittelbar. Im ersten Schritt erfolgt ausgehend der bereitgestellten ego-relativen 3D-Fahrzeugposition die Berechnung der konvexen 3D-Fahrzeughülle. Dafür sind die bekannten Fahrzeugdimensionen (Breite, Höhe, Länge) oder alternativ verallgemeinerte Größenangaben zu verwenden. Anschließend erfolgt die Projektion der 3D-Fahrzeughülle bzw. dessen acht 3D-Eckpunkten in das Kamerabild. Diese Projektion besteht aus verketteten Transformationen zwischen verschiedenen Koordinatensystemen. Die Positionsdaten der acht 3D-Eckpunkte befindet sich im Egokoordinatensystem. Von dort aus müssen die 3D-Punkte anhand der extrinsischen Kameraparameter - und damit mit dem Wissen über die ego-relative Kamerapose - in das Kamerakoordinatensystem transformiert werden. Anschließend erfolgt mit den intrinsischen Kameraparametern die Abbildung der 3D-Punkte in das 2D-Bildkoordinatensystem. Dort führt die konvexe Hülle der acht 2D-Punkte zu der gesuchten AOI. Die Berechnung der konvexen Hülle kann mit dem sogenannten *Graham Scan* [Graham, 1972] durchgeführt werden. In diesem Anwendungsfall besitzt der Algorithmus eine asymptotische Laufzeit von  $\mathcal{O}(n \cdot \log n)$ , mit  $n = 8$ .

Bei der Verwendung von kommunizierten 3D-Positionsdaten ist deren Fehlerbehaftung zu berücksichtigen. Die Berücksichtigung erfolgt durch eine Vergrößerung der berechneten AOI. In dem Fall, dass Genauigkeitsinformationen der bereitgestellten 3D-Positionsdaten zur Verfügung stehen, kann eine Skalierung der AOI auf deren Basis erfolgen, andernfalls ist ein benutzerdefinierter Skalierungsfaktor zu definieren.

**(1b) Konvertierung von Farb- in Grauwertbilder:** Der nächste Schritt ist die Konvertierung der Farbinformationen eines Farbbildes  $F$  in Grauwerte. Ein gängiges und effizientes Vorgehen ist es, die Werte der einzelnen Farbkanäle gewichtet aufzusummieren und anschließend die Summe zu mitteln. In dieser Arbeit kommt die aus Burger und Burge [2006] empfohlene Gewichtung von RGB-Pixelwerten zum Einsatz:

$$I(u, v) = \frac{0,299 \cdot F(u, v)_R + 0,587 \cdot F(u, v)_G + 0,114 \cdot F(u, v)_B}{3} \quad (\text{A.10})$$

Die Gleichung A.10 beschreibt die Konvertierung von einem Bildpunkt  $(u, v)$  aus einem Farbbild in den zugehörigen Pixelwert im Grauwertbild  $I$ . Dabei wird berücksichtigt, dass die wahrnehmbare Farbhelligkeit von Grün und Rot deutlich höher ist als die von Blau.

- (2) **Erstellung von Binärbildern:** Ein dominanter Marker weist starke Intensitätsübergänge (Kanten) und große unterschiedliche Farbflächen auf, ansonsten ist eine zuverlässige visuelle Wiedererkennung nicht gewährleistet. Ein Beispiel eines dominanten Markers ist der in dieser Arbeit verwendete schwarz/weiße Raute-Marker. Dieser hat keine Farbmerkmale, jedoch markante Markerkanten sowie helle und dunkle Flächen, welche in einem Grauwert- oder Binärbild detektiert werden können. In dieser Arbeit wird aus Effizienzgründen kein Grauwertbild, sondern ein daraus erstelltes Binärbild für die Suche nach den Markermerkmalen verarbeitet. Die Erstellung eines Binärbildes  $I_B$  aus einem Grauwertbild  $I$  ist mit einem Schwellwert  $g$  möglich:

$$I_B(u, v) = \begin{cases} 1, & \text{wenn } I(u, v) > g \\ 0, & \text{sonst} \end{cases} \quad (\text{A.11})$$

Der in der Gleichung A.11 dargestellte Vorgang wird als Schwellwertbildung bezeichnet, bei der die Wahl des Schwellwertes für die Markererkennung entscheidend ist. Wird dieser fest definiert, dann kann eine zuverlässige Markererkennung bei unterschiedlichen Wetter- und Belichtungsbedingungen nicht gewährleistet werden. Aus diesem Grund erfolgt die Bestimmung des Schwellwertes oft adaptiv [Sezgin und Sankur, 2004]. Eine Möglichkeit ist die Berechnung des Schwellwertes anhand des Mittelwertes  $\bar{m} = 1/n \cdot \sum_{i=1}^n I(u, v)_i$  und der Standardabweichung  $\sigma$  (unkorrigierte Stichprobenvarianz) aller  $n$  Pixelwerte:

$$g = \bar{m} - \alpha \cdot \sigma \quad (\text{A.12})$$

Dabei definiert  $\alpha$  den Einbeziehungsfaktor der Standardabweichung, die sich durch die Gleichung A.13 berechnen lässt.

$$\sigma = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (I(u, v)_i - \bar{m})^2} \quad (\text{A.13})$$

Diesbezüglich evaluierte diese Arbeit die Faktoren im Bereich von  $[0 \dots 1]$  und kein  $\alpha$  ermöglichte eine Markererkennung bei unterschiedlichen Belichtungen. Insbesondere bei sonnigen Tagen während der Durchfahrt von Waldgebieten schlug die Binärbilderstellung fehl und folglich war die Markerkontur nach der Schwellwertbildung nicht mehr eindeutig erkennbar. Weiterführend wurden noch weitere adaptive Verfahren zur Bestimmung eines optimalen Schwellwertes evaluiert. Dazu gehört das Cluster-basierte Verfahren namens *Otsu's method* [Otsu, 1979]. Doch auch dort wurden Binärbilder bei wechselhaften Lichtverhältnissen erzeugt, in denen der Marker komplett schwarz oder weiß erschien. Aus diesem Grund müssen mehrere Schwellwerte definiert und verwendet werden. Für die Bestimmung von geeigneten Schwellwerten erfolgte die Aufnahme von über 100 000 Bildern, mit drei Kameras und unter ver-

schiedenen Lichtbedingungen. Anschließend war die Anzahl der Schwellwerte und deren Wert zur optimalen Schwellwertbildung iterativ ermittelbar. Als Abbruchkriterium galt, dass eine erfolgreiche Markerdetektion in mindestens jedem zehnten Kamerabild vorhanden sein muss. Die dafür benötigte Anzahl an Schwellwerten wurde im Bereich  $[1 \dots 5]$  evaluiert. Die Pixelwerte eines Intensitätsbildes haben einen ganzzahligen Wertebereich von  $[0 \dots 127 \dots 255]$  (dunkel  $\rightarrow$  grau  $\rightarrow$  hell). Aus zeitlichen Gründen wurde nicht über jeden Schwellwert von 0 bis 255 iteriert, sondern nur für den ersten Schwellwert  $g_1^{(0)}$  und in Fünferschritten:

$$g_1^{(0)} = 0, g_1^{(1)} = 5, g_1^{(2)} = 10, \dots \quad (\text{A.14})$$

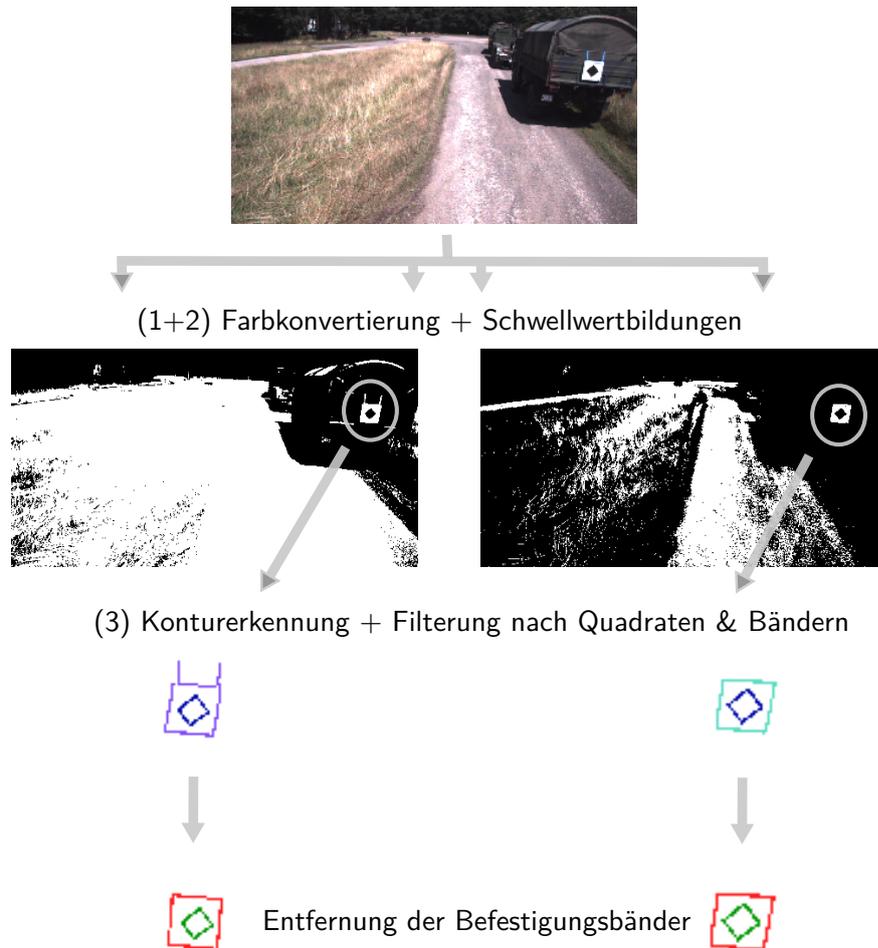
Für eine schnellere Konvergenz des Algorithmus erhielt jeder weitere Schwellwert  $g_{j,j \geq 2}$  einen benutzerdefinierten Offset zugewiesen:

$$g_j = g_1 + (j - 1) \cdot 40 \quad (\text{A.15})$$

Der Algorithmus konvergierte bei vier Schwellwerten mit  $g_1^{(12)} = 60$ . Folglich wurde in dieser Iteration  $g_2 = 100$ ,  $g_3 = 140$  und  $g_4 = 180$  zugewiesen. Diese vier Schwellwerte führen zur Erzeugung von vier Binärbildern  $I_{B_1, \dots, 4}$ , in denen mindestens ein Binärbild die Markerkonturen auch bei unterschiedlichen Lichtbedingungen eindeutig darstellt. Ein Praxisbeispiel zeigt die Abbildung A.7, in der ein Bild der mittleren Frontkamera vom TULF-Lkw mit vier Schwellwerten in vier Binärbilder konvertiert wurde. Dabei konnte der Raute-Marker in zwei der vier Binärbilder eindeutig wiedererkannt werden. Ebenfalls wurde der Marker in dem Bild der linken Frontkamera erfolgreich detektiert.

- (3a) Konturerkennung:** Der nächste Schritt ist die Untersuchung der Binärbilder auf markante Konturen. Ein effizienter Algorithmus ist die sogenannte *Connected Component-Analyse* von Suzuki und Abe [1985]. Das Verfahren verfolgt die Pixel mit einem Pixelwert ungleich Null, identifiziert dabei verbundene Pixel, nummeriert Konturbereiche und gliedert Konturen in eine Baumstruktur hierarchisch ein. Dabei erfolgt die Verfolgung von Pixeln durch die Betrachtung der horizontalen, vertikalen und diagonalen Nachbarpixel, sodass auch komplexe Konturen erkannt werden.
- (3b) Konturfilterung (Quadrate):** Im nächsten Schritt sind die einzelnen Markerkonturen zu extrahieren. Auf den ersten Blick weist der in der Abbildung A.6 dargestellte Raute-Marker zwei markante Konturen auf: Ein äußeres Quadrat und ein um  $45^\circ$ -rotiertes inneres Quadrat. Das innere Quadrat besitzt einen starken Intensitätsübergang zwischen der schwarzen Raute und der weißen Markergrundfläche. Das äußere Quadrat weist bei Fahrzeugen mit einer dunklen Karosserie ebenfalls vier dominante Kanten auf. Allerdings können zum Einsatz kommende Befestigungsbänder die Erkennung der äußeren Markerkontur

„stören“. Die Abbildung A.8 verdeutlicht die Problematik anhand von vier hellblauen Befestigungsbändern. In diesem Anwendungsbeispiel werden während der Schwellwertbildung Pixelbereiche der Befestigungsbänder als hell klassifiziert. Dies führt zu nicht-quadratischen Konturen und ein Suchalgorithmus für Quadrate schlägt fehl. Eine Lösung ist die gesonderte Detektion der Befestigungsbänder, sodass die Konturen der Bänder aus dem Konturbild extrahiert werden und ein Quadratdetektor korrekt arbeiten kann.



**Abbildung A.8:** Erkennung und Entfernung von Befestigungsbändern.

Im Nachfolgenden wird auf die Erkennung der Quadrate und Befestigungsbänder näher eingegangen. Zu Beginn erfolgt ein Vorverarbeitungsschritt bzgl. der gefundenen Konturen. Dabei besteht eine Kontur aus Stützpunkten, welche jeweils eine Richtungsänderung kennzeichnen. Im Praxisfall weist jedes unvorverarbeitete Kamerabild viele Kanten mit vielen Richtungswechseln auf. Wird beispielsweise eine gerade Tischkante fotografiert und deren Kontur bestimmt, dann kann aufgrund von Bildrauschen und Abbildungsungenauigkeiten die diskretisierte Kante nichtlinear sein und damit Pixelsprünge beinhalten. Dies

führt bei einer Suche nach den vier  $90^\circ$ -Winkeln eines Quadrates zu Problemen, da bei den Pixelsprüngen ebenfalls  $90^\circ$ -Richtungsänderungen erkannt werden. Aus diesem Grund wird der iterative Ramer–Douglas–Peucker (RDP)-Algorithmus [Ramer, 1972, Douglas und Peucker, 1973] zur Punktreduzierung und Linienglättung verwendet. Anschließend erfolgt die Flächenberechnung mit dem Lemma von Green, sodass Konturen mit zu kleinen und zu großen Flächen verworfen werden können. Als Schwellwerte für die Mindest- und Maximalfläche haben sich  $9 \times 9$  und  $300 \times 300$  Quadratpixel als nutzbar herausgestellt. Die nächste Filterung betrachtet den absoluten Winkel  $\alpha$  zwischen drei aufeinanderfolgenden Stützstellen  $(x, y)_{1,\dots,3}$  einer Kontur:

$$\alpha = \arccos \left( \frac{(x_2 - x_1) \cdot (x_2 - x_3) + (y_2 - y_1) \cdot (y_2 - y_3)}{\left( \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \cdot \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \right)} \right) \quad (\text{A.16})$$

An dieser Stelle sollte aufgrund von perspektivischen Verzeichnungen nicht exakt nach  $90^\circ$ -Winkeln, sondern nach Winkeln im Bereich von  $[80 \dots 100]^\circ$  gesucht werden. Nach der Winkelüberprüfung steht die Anzahl an  $90^\circ$ -Winkeln fest und Konturen mit weniger als vier Winkeln werden verworfen. Besitzt dagegen eine Kontur mehr als vier  $90^\circ$ -Winkel, dann könnte es sich hierbei um den Raute-Marker inkl. der Befestigungsbänder handeln. Demzufolge erfolgt eine Bänderdetektion und die Entfernung des Bandes aus der Kontur, sodass nur noch die Winkel vom quadratischen Raute-Marker erkannt werden.

Im Nachfolgenden wird eine Kontur bestehend aus vier  $90^\circ$ -Winkeln untersucht. Als Kriterium gilt zunächst die Kontursymmetrie und es erfolgt ein Vergleich zwischen der Konturbreite und -höhe. Ist das Verhältnis nicht nahezu identisch, dann gehört die Kontur nicht zum Raute-Marker und wird verworfen. Andernfalls erfolgt anhand der vier  $90^\circ$ -Winkel die Einordnung, ob es sich bei der Kontur um ein rotiertes oder aufrechtes Quadrat handelt. Anschließend ist die Farbe innerhalb der Kontur ein weiteres Klassifikationskriterium. Demzufolge muss ein rotiertes Quadrat dunkle Pixel aufweisen. Ein aufrechtes Quadrat sollte dagegen im Zentrum dunkle Pixel und am inneren Rand helle Pixel aufweisen.

- (4) Klassifikation des Markers:** Bis zu diesem Abschnitt erfolgte die Klassifizierung von rotierten und aufrechten Quadraten. In diesem Abschnitt folgt die übergeordnete Markerdetektion indem alle Einzelklassifikationen aus allen Binärbildern gegeneinander evaluiert werden. Das hat den Vorteil, dass nicht in einem Binärbild ein inneres und ein äußeres Quadrat eines Markers erkannt werden müssen, sondern dass unterschiedliche Schwellwertbildungen zu einer gemeinsamen Markerdetektion führen können. Abschließend wird ein Marker als „erfolgreich wiedererkannt“ klassifiziert, wenn sich ein rotiertes Quadrat in einem aufrechten Quadrat befindet und deren Größen in einem realistischen Verhältnis zueinander stehen.

**(5) Qualitätseinschätzung:** Eine Qualitätseinschätzung kann anhand der erkannten Markersegmente durchgeführt werden. Beispielsweise sind die nachfolgend aufgezählten Qualitätsaussagen möglich:

- Alle Markersegmente wurden wiedererkannt → Sichere Qualität
- Lediglich das äußere Quadrat wurde erkannt → Geringe Qualität
- Nur die innere Raute wurde erkannt → Geringe Qualität
- Kein Markersegment erkennbar → Schlechte Qualität

**(6) Optionale 3D-Entfernungsbestimmung:** Dieser Abschnitt beschreibt die optionale Bestimmung der relativen Lage der Kamera zum detektierten Marker. Dabei sind einerseits Informationen zum Sensorchip der Kamera und andererseits Wissen über die Breite, Höhe oder Diagonale des Markers erforderlich. Im Nachfolgenden zeigt die Gleichung A.17 die Berechnung des Markerabstandes  $d_m$  anhand einer bekannten Markerhöhe  $h_m$ , der vertikalen Brennweite  $f_{vm}$ , der Anzahl an Bildzeilen  $r$  und der Sensorchiphöhe  $s_m$ .

$$d_m = \frac{f_{vm} \cdot h_m \cdot r}{h_{px} \cdot s_m} \quad (\text{A.17})$$

Des Weiteren sind rekursive Schätzverfahren (siehe Abschnitt 3.2) für die 3D-Entfernungsbestimmung sowie zur zeitlichen Verfolgung von Markern geeignet. Hierbei ist die räumliche Abbildung zwischen der Markerposition im 3D-Raum auf die Bildebene zu modellieren. Anschließend kann jede Markerdetektion sequentiell in einem Filter eingefiltert werden.

### A.5.2 Bewertung

Der vorgestellte Algorithmus zur visuellen Markererkennung wurde für die eindeutige Identifikation von militärischen texturarmen Fahrzeugen sowie zur Erkennung von Gefahrgutschildern eingesetzt. In diversen Praxisdurchläufen erfolgte die visuelle Detektion von einem Raute-Marker der am Heck eines ausgewählten Führungsfahrzeuges befestigt war. Dabei wurden einerseits unterschiedliche Markergrößen und andererseits ein Pkw und zwei Lkw als Führungsfahrzeug erprobt. Hierbei konnte der Marker sowohl im Nahbereich als auch in großen Konvoiabständen zuverlässig wiedererkannt werden. Diesbezüglich listet die Tabelle A.5 den evaluierten visuellen Erfassungsbereich auf. Die hierbei zum Einsatz kommende Kamera war die Basler acA2040-25gc mit einem Weitwinkelobjektiv.

Markertyp	Größe (B × H)(m)	Detektionsbereich (m)		
		Äußeres Quadrat	Inneres Quadrat	Kompletter Marker
Raute	0,5 × 0,5	50	35	22
	1,0 × 1,0	100	70	44

**Tabelle A.5:**

Visueller Detektionsbereich eines Markers und dessen Segmente unter der Verwendung der Farbkamera Basler acA2040-25gc.

Die visuelle Detektion ist gegenüber wechselhaften Lichtverhältnissen sehr robust. Zum Beispiel wird ein Marker auch bei sonnigen Walddurchfahrten zuverlässig im Nahbereich erkannt. Der Grund dafür liegt in der Anwendung von mehreren Schwellwertbildungen, wodurch unterschiedliche Binärbilder auf die Markerkonturen untersucht werden. Des Weiteren ist der Algorithmus modular aufgebaut und damit auf beliebig vielen Kameras parallel anwendbar. Der Algorithmus verarbeitet mit einer einzelnen 3,4 GHz Intel CPU ein Kamerabild in Echtzeit. Im Detail benötigt die Verarbeitung eines kompletten Kamerabildes der Größe  $1294 \times 964$  Pixel grob 80 ms. In der Verarbeitungszeit nimmt die Konturerkennung und -filterung mit circa 60 ms die meiste Zeit in Anspruch. Allerdings ist die Verarbeitung eines kompletten Kamerabildes nur in seltenen Fällen notwendig. Wurde beispielsweise ein Marker bereits erkannt, dann ist im nächsten Zeitschritt lediglich ein kleiner Bildausschnitt auszuwerten. Dies reduziert die Verarbeitungszeit eines Kamerabildes bzw. dessen Bildausschnittes auf durchschnittlich 7 ms.

## A.6 Inbetriebnahme der Software

Dieses Kapitel beschreibt die Inbetriebnahme des Folgeassistenten und gibt dem Leser einen Eindruck über die Handhabbarkeit der entwickelten Software. Insgesamt müssen von einem Benutzer vier Grundschritte durchgeführt werden:

1. Einschalten des Fahrzeugrechners und der Sensorik
2. Starten der gewünschten Softwareapplikation mit einer optionalen Einstellung von Konvoiparametern
3. Entriegelung des Notauschalters
4. Aktivierung der Autonomiefunktion via Taster

Die Abbildung A.9 zeigt den Innenraum von MuCAR-3 und kennzeichnet die vier Orte, an denen die jeweiligen Grundschritte durchgeführt werden.

1.) Einschalten des Fahrzeugrechners und der Sensorik



2.) Starten der Softwareapplikationen inkl. Konvoikonfiguration



4.) Aktivierung der Autonomiefunktion

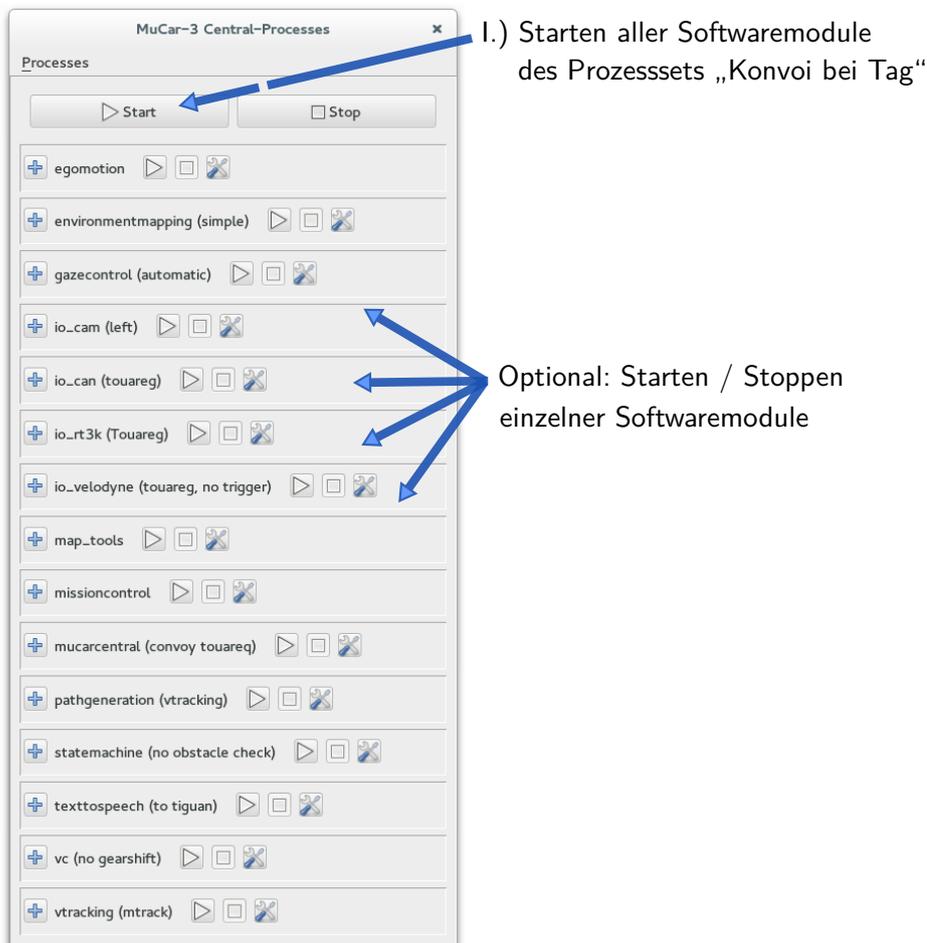


3.) Entriegelung des Notauschalters

**Abbildung A.9:** Vier Grundschritte für die Inbetriebnahme des Folgeassistenten.

Der erste Schritt ist das Einschalten des Fahrzeugrechners und der Sensorik. In den Fahrzeugen MuCAR-3 und MuCAR-4 erfolgt dies über Schalter und Knöpfe, die in der Mittelkonsole integriert sind. Im TULF und StrAsRob befindet sich eine Schalttafel im hinteren Bereich der Mittelkonsole.

Nach dem Einschaltvorgang wird das Linux-Betriebssystem geladen und die verfügbaren Applikationen können angezeigt werden. Dafür muss in MuCAR-3 der Kommandozeilenauf Ruf „mcp-touareg“ und in MuCAR-4 entsprechend „mcp-tiguan“ aufgerufen werden. Daraufhin kann der Benutzer einzelne Applikationen sowie gesamte Sets von Softwaremodulen auswählen. Beispielsweise existiert die Applikation für das autonome Folgen im Konvoi oder die Navigation mit aktiver Weg- und Kreuzungserkennung. Die Abbildung A.10 zeigt das Prozessset für den Folgeassistenten. In diesem sind die Softwaremodule aufgelistet, welche zum autonomen Nachfahren eines ausgewählten Führungsfahrzeuges gestartet werden können. Für das Starten aller Softwaremodule wird einmal der „Start“-Knopf betätigt.

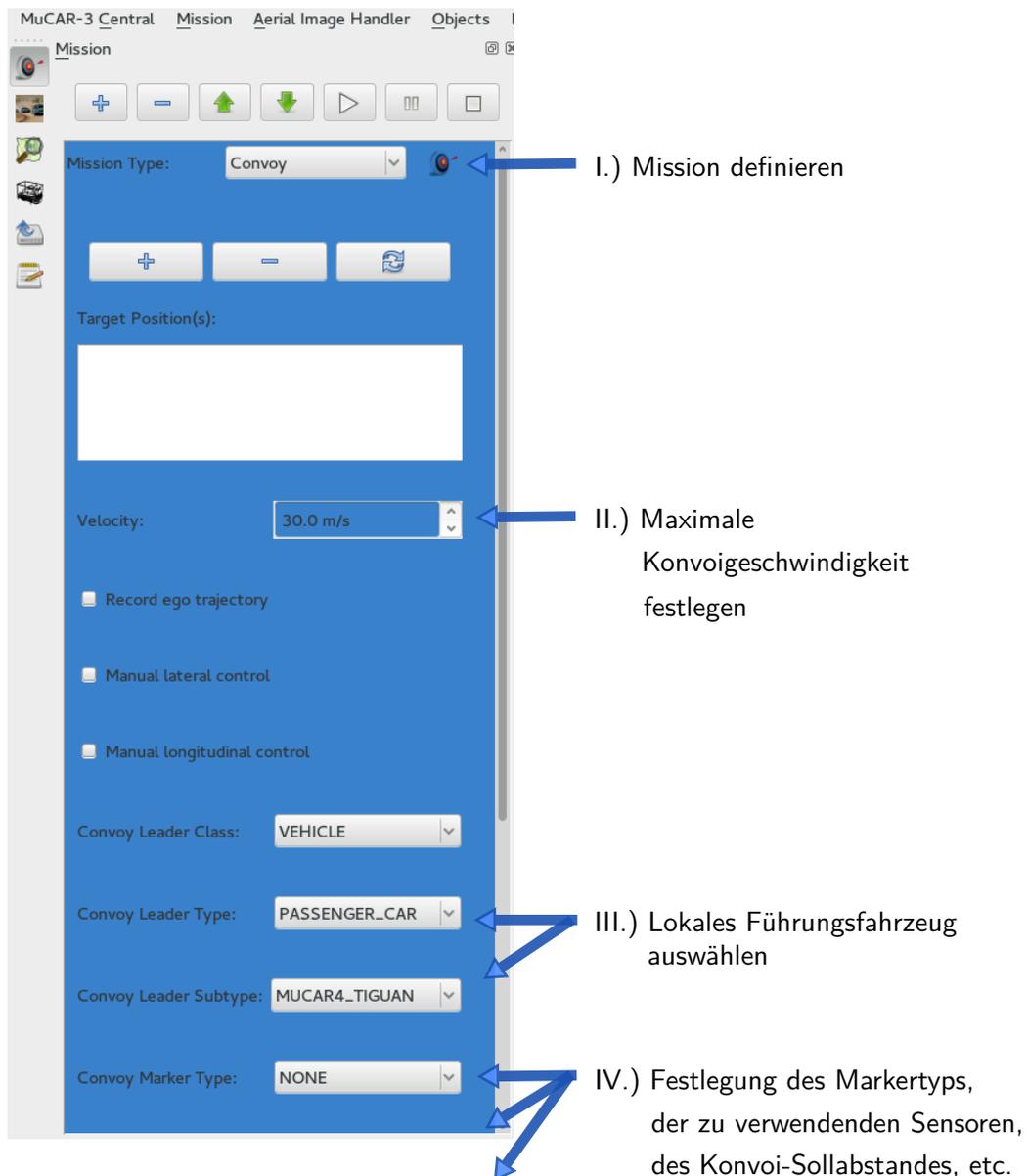


**Abbildung A.10:**

Starten des gesamten Prozesssets „Autonomes Folgen bei Tag“ oder von einzelnen Softwaremodulen.

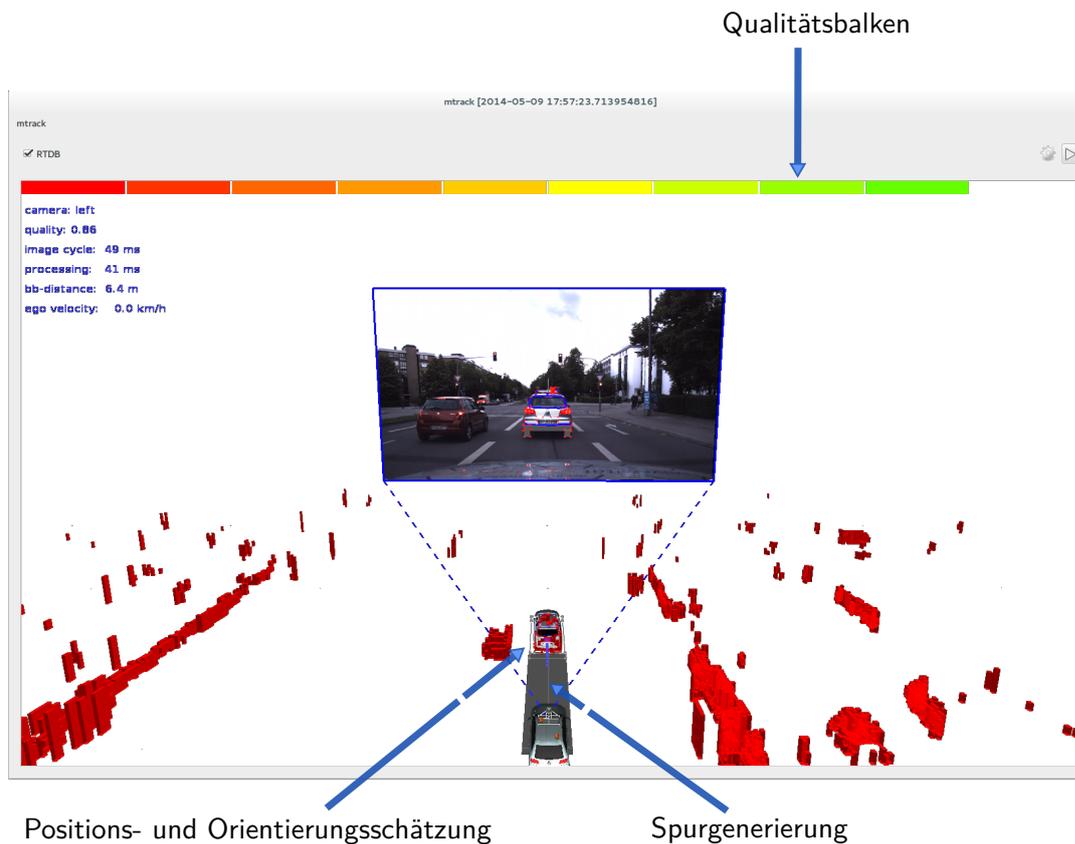
Die einzelnen Module können allerdings auch separat gestartet oder gestoppt werden. Beispielsweise ist beim Folgeassistenten das Sprachausgabemodul „texttospeech“ über die aktuelle Schätzqualität der Fahrzeugerkennung nicht erforderlich und kann aus diesem Grund abgeschaltet werden.

Insgesamt erfolgt das Starten der einzelnen Module sehr schnell, sodass beim Betätigen des „Start“-Knopfes alle Programme in wenigen Sekunden geladen und einsatzbereit sind. Ein wichtiges Modul, in dem zentrale Missionseinstellungen vorgenommen werden können, ist in der Abbildung A.11 zu sehen.



**Abbildung A.11:** Mögliche Missionsparameter.

Hinsichtlich des Folgeassistenten ist dort das passende Führungsfahrzeug auswählbar, der gewünschte Sollabstand zwischen zwei Fahrzeugen einstellbar und es können die Sensoren definiert werden, welche für die Fahrzeugerkennung genutzt werden dürfen. Die Änderungen der Missionsparameter wirken sich dabei sofort auf die anderen Softwaremodule aus, da diese über eine Echtzeitdatenbank [Goebl, 2009] kommuniziert werden. Das Softwaremodul zur Fahrzeugerkennung, welches die Missionsparameter kontinuierlich liest und auf Basis dieser Informationen (Führungsfahrzeug, Markertyp, Sensorbegrenzung, etc.) eine Fahrzeugerkennung durchführt, wird in der Abbildung A.12 gezeigt. Hinsichtlich der gestarteten Softwaremodule gibt die grafische Oberfläche der Fahrzeugerkennung einen guten Überblick über die Einsatzbereitschaft des Folgeassistenten wieder. Denn zum einen wird die geschätzte ego-relative Pose vom Führungsfahrzeug und die daraus ermittelte Spurgenerierung visualisiert und zum anderen befindet sich ein farbiger Qualitätsbalken am oberen Rand des Softwaremoduls. Diese grafischen Informationen geben dem Benutzer einen guten Eindruck, wie zuverlässig das System gerade ist und ob die Freischaltung zum autonomen Folgen erfolgen sollte.



**Abbildung A.12:**

Grafische Visualisierung der Fahrzeugerkennung und der Spurgenerierung.

Für die Freischaltung der Autonomiefunktion sind zwei Tätigkeiten notwendig. Zum einen muss der Notauschalter, der sich in der Mittelkonsole befindet, entriegelt werden und zum anderen ist eine manuelle Betätigung eines grünen Schalters erforderlich (siehe Abbildung A.9). Der grüne Schalter befindet sich ebenfalls in der Mittelkonsole, aktiviert die Autonomiefunktion und erlaubt der Software die Ansteuerung des Lenkrades, der Schaltung sowie des Gas- und Bremspedals.

# Index

- Ackermann Steering Model, 168
- Binärbild, 88
- Closing-Operator, 57
- Connected-Component-Analyse, 56
- Ego-relative Fahrzeugpose, 72
- Farbräume
  - HSV, 166
  - RGB, 165
- Gauß-Verteilung, 79
- Hindernisgitter, 91
- Hinderniszellen, 91
- Kaskadierter Klassifikator, 40
- Klothoide, 113
- Linienintegralbild, 88
- MarVEye, 101, 119
- Modellgenerierung, 51–70, 125
  - 3D-Merkmalmodell, 54
  - Automatisch, 54, 130
  - Manuell, 51, 125
- MuCAR, 118
- Normalverteilung, 79
- Object of Interest (OOI), 54
- Region Growing, 46
  - Cell Flooding, 47
  - Multi-Dynamic Region Growing, 47
- Rekursive Zustandsschätzung, 29–40
  - Kalman-Filter, 29
  - Partikel-Filter, 36
  - Scaled Unscented Transform, 34
  - Sequentielle Innovation, 32
- Schwellwertbildung, 60
- Sensorkalibrierung, 17–27
  - Extrinsisch, 18–20
  - Intrinsisch, 20
  - Intrinsische Kameraparameter, 20
    - Bildweite, 20
    - Hauptpunktverschiebung, 20
    - Linsenverzeichnung, 20
    - Scherungskoeffizient, 20
- Sensorsynchronisierung, 27
- Strahlensatz, 22
- StrAsRob, 123
- TULF, 122
- Zylinderkoordinaten, 72

# Abbildungsverzeichnis

1.1	Statistik über die am häufigsten vorkommenden Unfallarten auf deutschen Straßen . . . . .	2
2.1	Forschungsprojekt KONVOI . . . . .	7
2.2	Künstliche Marker . . . . .	13
3.1	Extrinsische Sensorpose . . . . .	18
3.2	Manuelle extrinsische Kalibrierung einer Kamera in Bezug zu einem LiDAR-Sensor . . . . .	19
3.3	Zwei Verzeichnungsarten einer Kamera . . . . .	21
3.4	Zentralprojektion . . . . .	22
3.5	Kalibriermuster zur Ermittlung der intrinsischen Kameraparameter .	25
3.6	Visuelle Detektion eines bekannten Schachbrettmusters . . . . .	26
3.7	Blickrichtungsabhängige Synchronisierung von Sensordaten . . . . .	28
3.8	<i>Unscented Transform</i> . . . . .	34
3.9	Approximation einer multimodalen Wahrscheinlichkeitsverteilung . .	37
3.10	Partikel-Filter: Prädiktion und Innovation . . . . .	39
3.11	Kaskadierung von schwachen Klassifikatoren . . . . .	40
3.12	Aufnahme von negativen Kamerabildern . . . . .	42
3.13	Zeitaufwendiges manuelles Labeln von mehreren tausend Bildern . .	42
3.14	Halb-automatisches Labeln von mehreren tausend Kamerabildern. .	43
3.15	Flächenbasierte Haar-Merkmale . . . . .	44
3.16	Illustrierung von Haar-Merkmalen, die ein maschineller Lernprozess an vier Bildpositionen als Alleinstellungsmerkmal identifizierte. . . .	45

3.17	Objektbezogene Pixeleliminierung mit dem <i>Multi-Dynamic Region Growing</i> -Verfahren . . . . .	48
4.1	Erstellung eines 3D-Merkmalmodells . . . . .	52
4.2	Sensorabhängige 3D-Merkmalmodelle eines VW Tiguan . . . . .	53
4.3	Exemplarische Trajektorie zur Datenaufzeichnung . . . . .	55
4.4	Der morphologische Bildverarbeitungsoperator <i>Closing</i> . . . . .	57
4.5	Ablaufdiagramm über die vier Phasen der automatischen Modellerzeugung . . . . .	59
4.6	Suche nach dominanten Kanten . . . . .	62
4.7	Merkmalstransformation in ein globales Referenzkoordinatensystem	68
4.8	Suche nach dominanten Kanten mittels statistischer Auswertung einer zeitlichen Merkmalsbeobachtung . . . . .	69
5.1	Relative Fahrzeugpose . . . . .	72
5.2	Filter-Initialisierung . . . . .	73
5.3	Klassifikatoren zur Detektion von Fahrzeugseiten . . . . .	74
5.4	Fahrzeug umschließendes Quader beschrieben durch 8 Eckpunkte .	75
5.5	Prädiktion der ego-relativen Fahrzeugpose . . . . .	76
5.6	Zufallszahlengenerator für eine Normalverteilung . . . . .	79
5.7	Partikelgewichtung . . . . .	82
5.8	Kantenextraktion mittels Canny-Algorithmus . . . . .	84
5.9	Größe der Faltungsmaske . . . . .	85
5.10	Berechnung der Kantenabstände mittels definierter Abstandstransformation . . . . .	86
5.11	Filterung nach roten Bildpixeln . . . . .	89
5.12	Hohe Partikelgewichtung . . . . .	91

5.13	Erstellung einer Hinderniskarte aus Laserdaten . . . . .	92
5.14	Auswertung von hindernisbehafteten und hindernisfreien Zellen . . . . .	94
5.15	Merkmalsauswertung im 2D- und 2,5D-Hindernisgitter . . . . .	95
5.16	Prädiktionsschritt des Partikel-Filters . . . . .	96
5.17	Robustheitssteigerung des Gesamtsystems durch die Kopplung von zwei unterschiedlichen Verfahren zur Fahrzeu­gerkennung . . . . .	99
5.18	Aktive Blickrichtungssteuerung mit der bewegbaren Kameraplattform MarVEye-8 . . . . .	101
5.19	Visuelle Aufmerksamkeitssteuerung im Bereich der Objekterkennung	102
5.20	Softwarebasierte AOI . . . . .	103
5.21	<i>Multi-Dynamic Region Growing</i> zur Aufmerksamkeitssteuerung . . . . .	104
5.22	Naiver Bayes-Klassifikator zur Auswertung von roten Flächen . . . . .	109
5.23	Naiver Bayes-Klassifikator zur Auswertung von Kantenabständen . . . . .	110
5.24	Spurgenerierung zum Folgen eines Führungsfahrzeuges . . . . .	112
6.1	MuCAR-3: Umgebauter VW Touareg und dessen Sensorausstattung	118
6.2	Fahrer- und Beifahrerraum von MuCAR-3 . . . . .	119
6.3	MuCAR-4: Umgebauter VW Tiguan und dessen Sensorausstattung	121
6.4	TULF: Sensorausstattung eines umgebauten Lkw . . . . .	122
6.5	StrAsRob: Sensorausstattung eines umgebauten Lkw . . . . .	123
6.6	Führungsfahrzeuge dieser Arbeit . . . . .	124
6.7	3D-Merkmalsmodelle eines Lkw vom Typ MAN mit gl . . . . .	126
6.8	Sensorabhängiges Merkmalsmodell eines Lkw . . . . .	129
6.9	Sensorabhängige 3D-Merkmalsmodelle zur Auswertung von Farb-, Wärme- und Hindernismerkmalen . . . . .	131
6.10	Visuelle Fahrzeu­gerkennung mit kaskadierten Klassifikatoren . . . . .	133

6.11	Modellbasierte Fahrzeugerkennung mit einem Partikel-Filter . . . . .	134
6.12	Modellbasierte Fahrzeugerkennung eines VW Golf trotz schlechter Sichtbedingungen . . . . .	135
6.13	Autonomes Folgen in einer unstrukturierten Umgebung . . . . .	136
6.14	Visuelle Eindrücke der modellbasierten Fahrzeugerkennung . . . . .	137
6.15	Visuelle Eindrücke der modellbasierten Fahrzeugerkennung eines VW Tiguan . . . . .	138
6.16	Modellbasierte Erkennung eines ausgewählten Führungsfahrzeuges bei Nacht . . . . .	140
6.17	Überblick einer autonomen zwölf Kilometer langen Fahrt bei Nacht	142
6.18	Anspruchsvolle Verkehrssituationen während einer autonomen Nachtfahrt . . . . .	143
6.19	Autonomes Fahren im Dreierkonvoi . . . . .	144
6.20	Ergebniskarte vom Team MuCAR-3. . . . .	147
6.21	Anspruchsvolle Situationen während des Konvoiszenarios der ELROB'16.	148
A.1	Intrinsische Kameraparameter: Bildhauptpunkt, tangentielle und radiale Verzeichnung . . . . .	164
A.2	Visualisierung des RGB-Farbraumes . . . . .	165
A.3	Visualisierung des HSV-Farbraumes . . . . .	166
A.4	Farbtonskala vom HSV-Farbraum . . . . .	166
A.5	Einspurmodel eines vorderachsgelenkten Fahrzeuges . . . . .	168
A.6	Visuelle Kennzeichnung von militärischen Lkws . . . . .	170
A.7	Ablaufdiagramm der visuellen Markererkennung . . . . .	171
A.8	Erkennung und Entfernung von Befestigungsbändern . . . . .	175
A.9	Vier Grundschritte für die Inbetriebnahme des Folgeassistenten . . . .	179
A.10	Starten des gesamten Prozesssets oder von Softwaremodulen . . . .	180

A.11 Mögliche Missionsparameter . . . . .	181
A.12 Grafische Visualisierung der Fahrzeugerkennung und der Spurgenerierung . . . . .	182

## Tabellenverzeichnis

5.1	Hinzufügen von zufälligem Rauschen auf prädierte Zustandswerte	80
6.1	Jurybewertung über die erfolgreichen Teilnehmer des Konvoiszenarios der ELROB'16. . . . .	148
6.2	Evaluierung der Schätzgenauigkeit . . . . .	150
A.1	Sensoreigenschaften der in dieser Arbeit zum Einsatz kommenden LiDAR-Sensoren . . . . .	156
A.2	Auflistung der in dieser Arbeit verwendeten Kameras. . . . .	159
A.3	Inertialsensoren . . . . .	160
A.4	Ermittelte intrinsische Kameraparameter . . . . .	163
A.5	Visueller Detektionsbereich eines Markers . . . . .	178

## Literaturverzeichnis

- Adams, R. und Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647.
- Alam, A. (2011). *Fuel-Efficient Distributed Control for Heavy Duty Vehicle Platooning*. Diplomarbeit, KTH School of Electrical Engineering, Stockholm, Sweden.
- Alcantarilla, P. F., Bergasa, L. M., Jiménez, P., Sotelo, M. Á., Parra, I., Llorca, D. F., und Mayoral, S. (2008). Night Time Vehicle Detection for Driving Assistance Lightbeam Controller. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 291–296.
- Bargende, M. und Wiedemann, J. (2001). *Kraftfahrwesen und Verbrennungsmotoren – Automotive and Engine Technology*. expert-Verlag. <https://books.google.de/books?id=8y0Bbr18ke8C>.
- Barth, A. und Franke, U. (2010). Tracking Oncoming and Turning Vehicles at Intersections. In *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, Seiten 861–868.
- Bay, H., Tuytelaars, T., und Gool, L. V. (2006). Surf: Speeded Up Robust Features. In *ECCV*, Seiten 404–417.
- Bergenheim, C., Shladover, S., Coelingh, E., Englund, C., und Tsugawa, S. (2012). Overview of Platooning Systems. In *Proceedings of the 19th ITS World Congress*, Vienna, Austria.
- Bouguet, J.-Y. (1999). *Visual Methods for Three-Dimensional Modeling*. PhD thesis, California Institute of Technology, Pasadena, California.
- Bouguet, J.-Y. (2015). Camera Calibration Toolbox for Matlab. Website. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/) Zugriffsdatum: 20.12.2016, Stand 14.10.2015.
- Bradski, G. und Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA.
- Bresenham, J. E. (1965). Algorithm for Computer Control of a Digital Plotter. *IBM Systems Journal*, 4(1):25–30.
- Broggi, A., Cerri, P., Felisa, M., Laghi, M. C., Mazzei, L., und Porta, P. P. (2012). The VisLab Intercontinental Autonomous Challenge: an Extensive Test for a Platoon of Intelligent Vehicles. *International Journal of Vehicle Autonomous Systems, special issue for 10<sup>th</sup> Anniversary*, 10(3).
- Brown, D. C. (1966). Decentering Distortion of Lenses. *PHOTOGRAMMETRIC ENGINEERING*, 32(3):444–462.

- Brown, D. C. (1971). Close-Range Camera Calibration. *PHOTOGRAMMETRIC ENGINEERING*, 37(8):855–866.
- Buehler, M., Iagnemma, K., und Singh, S., Herausgeber (2009). *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, volume 56 of *Springer Tracts in Advanced Robotics*. Springer.
- Burger, W. und Burge, M. J. (2006). *Digitale Bildverarbeitung: Eine Einführung mit Java und ImageJ*. X.media.press. Springer Vieweg.
- Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698.
- Catling, I. und McQueen, B. (1991). Road Transport Informatics in Europe-Major Programs and Demonstrations. *IEEE Transactions on Vehicular Technology*, 40(1):132–140.
- Chan, E., Gilhead, P., Jelinek, P., Krejci, P., und Robinson, T. (2012). Cooperative Control of SARTRE Automated Platoon Vehicles. In *Proceedings of the 19th ITS World Congress*, Vienna, Austria.
- Chen, C.-Y., Chien, H.-J., Huang, P.-S., Hong, W.-B., und Chen, C.-F. (2012). Intrinsic Parameters Calibration for Multi-beam LiDAR Using the Levenberg-Marquardt Algorithm. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand, IVCNZ*, Seiten 19–24, New York, NY, USA. ACM.
- Chen, Y.-L. und Chiang, C.-Y. (2010). Embedded On-Road Nighttime Vehicle Detection and Tracking System for Driver Assistance. In *In Proceedings of IEEE International Conference on Systems Man and Cybernetics (SMC)*, Seiten 1555–1562, Istanbul, Turkey.
- Chen, Y.-L., Wu, B.-F., Huang, H.-Y., und Fan, C.-J. (2011). A Real-Time Vision System for Nighttime Vehicle Detection and Traffic Surveillance. *IEEE Transactions on Industrial Electronics*, 58:2030–2044.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., und Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Dalal, N. und Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 886–893.
- Danielsson, P.-E. und Seger, O. (1990). *Generalized and Separable Sobel Operators*. Machine vision for three-dimensional scenes. Herbert Freeman (ed), Academic Press.
- Deuschle, S., Kessler, G., Lank, C., Hoffmann, G., Hakenberg, M., und Brummer, M. (2010). Einsatz elektronisch gekoppelter Lkw-Konvois auf Autobahnen. *ATZ - Automobiltechnische Zeitschrift*, Seiten 570–575.

- Dickmanns, E., Behringer, R., Dickmanns, D., Hildebrandt, T., Maurer, M., Thomanek, F., und Schiehlen, J. (1994). The Seeing Passenger Car 'VaMoRs-P'. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 68–73.
- Dickmanns, E. D. (1989). Subject-object discrimination in 4D dynamic scene interpretation for machine vision. In *In Proceedings of Workshop on Visual Motion*, Seiten 298–304.
- Dickmanns, E. D. (2007). *Dynamic Vision for Perception and Control of Motion*. Springer Verlag.
- Dominguez, R., Onieva, E., Alonso, J., Villagra, J., und Gonzalez, C. (2011). LIDAR Based Perception Solution for Autonomous Vehicles. In *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, Seiten 790–795.
- Douglas, D. H. und Peucker, T. K. (1973). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.
- Dudenhöffer, F. (2016). *Wer kriegt die Kurve?: Zeitenwende in der Autoindustrie*. Campus Verlag.
- Eigel, T. (2010). *Integrierte Längs- und Querführung von Personenkraftwagen mittels Sliding-Mode-Regelung*. AutoUni-Schriftenreihe. Logos-Verlag. <http://books.google.de/books?id=2X66ygAACAAJ>.
- Fanto, P. L. (2012). *Automatic Positioning and Design of a Variable Baseline Stereo Boom*. Masterarbeit, Staatsuniversität, Virginia, USA.
- Fassbender, D., Fries, C., Luettel, T., und Wuensche, H.-J. (2014). TULF - Automatisiertes Folgen im Konvoi. In *Wehrwissenschaftliche Forschung - Jahresbericht 2013*, Seiten 52–53. Bundesministerium der Verteidigung.
- Fiala, M. (2005). *ARTag, a Fiducial Marker System Using Digital Techniques*. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, Washington, DC, USA.
- Freund, Y. und Schapire, R. E. (1995). A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. In *Proceedings of the Second European Conference on Computational Learning Theory, EuroCOLT*, Seiten 23–37, London, UK. Springer-Verlag.
- Friedman, J., Hastie, T., und Tibshirani, R. (2000). Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics*, 38(2).
- Fries, C., Burger, P., Kallwies, J., Naujoks, B., Luettel, T., und Wuensche, H.-J. (2017). How MuCAR Won the Convoy Scenario at ELROB 2016. In *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, Yokohama, Japan.

- Fries, C., Luettel, T., und Wuensche, H.-J. (2013). Combining Model- and Template-based Vehicle Tracking for Autonomous Convoy Driving. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, Queensland, Australia.
- Fries, C. und Schuhfuß, M. (2010). *Realisierung eines 3D-Laserscanners: auf Basis der Lichtschnitttechnik*. VDM Verlag Dr. Müller.
- Fries, C. und Wuensche, H.-J. (2014). Monocular Template-based Vehicle Tracking for Autonomous Convoy Driving. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, IL, USA.
- Fries, C. und Wuensche, H.-J. (2015). Autonomous Convoy Driving by Night: The Vehicle Tracking System. In *Proceedings of IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, Boston, MA, USA.
- Fries, C. und Wuensche, H.-J. (2016). Real-time Unsupervised Feature Model Generation for a Vehicle Following System. In *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, Rio de Janeiro, Brazil.
- Gaudet, B. (2014). Review of cooperative truck platooning systems. Technical report, National Research Council Canada. Automotive and Surface Transportation.
- Glennie, C. und Lichti, D. D. (2010). Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning. *Remote Sensing*, 2(6):1610–1624.
- Goebel, M. (2009). *Eine realzeitfähige Architektur zur Integration kognitiver Funktionen*. Dissertation, Technische Universität München, München.
- Goerick, C., Noll, D., und Werner, M. (1996). Artificial Neural Networks in Real-Time Car Detection and Tracking Applications. *Pattern Recognition Letters*, 17(4):335–343.
- Gonzalez, R. C. und Woods, R. E. (2008). *Digital Image Processing*. Prentice Hall, USA.
- Gordon, N. J., Salmond, D. J., und Smith, A. F. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEEE Proceedings F Radar and Signal Processing*, 140(2):107–113.
- Graham, R. L. (1972). An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. In *Information Processing Letters 1*, Seiten 132–133, New Jersey, USA.
- Guo, L. und Shen, X. H. (2013). Vehicle Detection Method under Night Circumstance. In *Applied Mechanics and Materials*, volume 380, Seiten 3870–3873, Switzerland.
- Hajebi, K. (2007). *Structure from Infrared Stereo Images*. Masterarbeit, Universität Waterloo, Canada.
- Harrer, M. und Pfeffer, P. (2016). *Steering Handbook*. Springer International Publishing.

- Harris, C. und Stephens, M. (1988). A Combined Corner and Edge Detector. In *In Proceedings of Fourth Alvey Vision Conference*, Seiten 147–151.
- Hartley, R. I. und Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition.
- Hastie, T., Tibshirani, R., und Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer.
- Hecker, F., Luettel, T., und Wuensche, H.-J. (2010). Object Related Reactive Offset Maneuver. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 1092 – 1097, San Diego, CA, USA.
- Heinrich, B. C. und Wuensche, H.-J. (2017). Autonomous Cooperative Multi-Convoy Control for Off-Road Scenarios. In *8. VDI/VDE Fachtagung, AUTOREG 2017 - Automatisiertes Fahren und vernetzte Mobilität*, Berlin, Germany.
- Himmelsbach, M., Luettel, T., und Wuensche, H. (2009a). Real-time Object Classification in 3D Point Clouds Using Point Feature Histograms. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seiten 994–1000.
- Himmelsbach, M., Schneider, S., und Wuensche, H.-J. (2011). A Comparison of Error Metrics for Extrinsic Calibration and Fusion of Camera and Multi-Layer LIDAR. In *Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA)*, Innsbruck, Austria.
- Himmelsbach, M., von Hundelshausen, F., Luettel, T., Manz, M., Mueller, A., Schneider, S., und Wuensche, H.-J. (2009b). Team MuCAR-3 at C-ELROB 2009. In *Proceedings of 1st Workshop on Field Robotics, Civilian European Land Robot Trial*.
- Hofmann, U., Rieder, A., und Dickmanns, E. D. (2000). EMS-Vision: Application to Hybrid Adaptive Cruise Control. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 468–473.
- Hota, R. N., Jonna, K., und Krishna, P. R. (2010). On-road Vehicle Detection by Cascaded Classifiers. In *Proceedings of the Third Annual ACM Bangalore Conference, COMPUTE '10*, Seiten 27:1–27:5, New York, NY, USA. ACM.
- Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., Mujica, F., Coates, A., und Ng, A. Y. (2015). An Empirical Evaluation of Deep Learning on Highway Driving. *CoRR*.
- ISO 15622 (2018). Intelligent transport systems – Adaptive cruise control systems – Performance requirements and test procedures. International Organization for Standardization. <https://www.iso.org/standard/71515.html> Zugriffdatum: 28.09.2018.

- Jahnke, B., Noack, P. O., Happich, G., und Geimer, M. (2012). Elektronische Deichsel für landwirtschaftliche Arbeitsmaschinen - Folgsame Traktoren. In *KEM - Informationsvorsprung für Konstrukteure*, Seite 4. Institut für Fahrzeugtechnik und Mobile Arbeitsmaschinen (IFFMA), Konradin Verlag.
- Julier, S. J. (2002). The Scaled Unscented Transformation. In *Proceedings of the American Control Conference*, volume 6, Seiten 4555–4559.
- Julier, S. J., Uhlmann, J. K., und Durrant-Whyte, H. F. (1995). A New Approach for Filtering Nonlinear Systems. In *Proceedings of the American Control Conference*, Seiten 1628–1632, Seattle, WA.
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the American Society of Mechanical Engineers (ASME) – Journal of Basic Engineering*, 82:35–45.
- Kammel, S., Ziegler, J., Pitzer, B., Werling, M., Gindele, T., Jagszent, D., Schröder, J., Thuy, M., Goebel, M., Hundelshausen, F., Pink, O., Frese, C., und Stiller, C. (2008). Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge. *Journal of Field Robotics*, 25:615–639.
- Kang, C.-C. und Wang, W.-J. (2009). Fuzzy based Seeded Region Growing for Image Segmentation. In *Fuzzy Information Processing Society, NAFIPS. Annual Meeting of the North American*.
- Kato, H. und Billinghurst, M. (1999). *Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System*. Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality - S. 85 ff. IEEE Computer Society, Washington, DC, USA.
- Kim, S., young Oh, S., Kang, J., Ryu, Y., Kim, K., Park, S.-C., und Park, K. (2005). Front and Rear Vehicle Detection and Tracking in the Day and Night Times using Vision and Sonar Sensor Fusion. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seiten 2173–2178.
- Krishnan, A. und Larsson, J. (2016). *Vehicle Detection and Road Scene Segmentation using Deep Learning*. Masterarbeit, Chalmers University of Technology, Sweden.
- Krizhevsky, A., Sutskever, I., und Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, Seiten 1097–1105. Curran Associates, Inc.
- Kämpchen, N., Waldmann, P., Homm, F., und Ardelt, M. (2010). Umfelderkennung für den Nothalteassistenten – ein System zum automatischen Anhalten bei plötzlich reduzierter Fahrfähigkeit des Fahrers. In *Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel (AAET)*, Braunschweig.
- Lammert, M. P., Duran, A., Diez, J., Burton, K., und Nicholson, A. (2014). Effect of Platooning on Fuel Consumption of Class 8 Vehicles Over a Range of Speeds,

- Following Distances, and Mass. *SAE International Journal of Commercial Vehicles*, 7:626–639.
- Landsiedel, C. und Wollherr, D. (2017). Road Geometry Estimation for Urban Semantic Maps using Open Data. *Advanced Robotics*, 31(5):282–290.
- Lange, S., Ulbrich, F., und Goehring, D. (2016). Online Vehicle Detection using Deep Neural Networks and Lidar based Preselected Image Patches. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 954–959.
- Langerwisch, M., Reimer, M., Hentschel, M., und Wagner, B. (2010). Control of a Semi-Autonomous UGV Using Lossy Low-Bandwidth Communication. In *2nd IFAC Symposium on Telematics Applications, TA 2010, Bucharest, Romania, October 5-8, 2010.*, Seiten 184–190.
- Lank, C., Haberstroh, M., und Wille, M. (2011a). Interaction of Human, Machine, and Environment in Automated Driving Systems. In *Transportation Research Record: Journal of the Transportation Research Board*, Seiten 138–145.
- Lank, C., Haberstroh, M., und Wille, M. (2011b). KONVOI-Projekt: Einflüsse automatisierter Lkw auf Fahrer und Umgebungsverkehr. *Zeitschrift für Verkehrssicherheit*, Seiten 7–13.
- Lepetit, V., Moreno-Noguer, F., und Fua, P. (2009). EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81:155–166.
- Lienhart, R. und Maydt, J. (2002). An Extended Set of Haar-like Features for Rapid Object Detection. In *Proceedings of the International Conference on Image Processing*, volume 1, Seiten 900–903.
- Lou, J., Tan, T., Hu, W., Yang, H., und Maybank, S. J. (2005). 3-D Model-Based Vehicle Tracking. *IEEE Transactions on Image Processing*, 14:1561–1569.
- Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. In *Proceedings of IEEE International Conference on Computer Vision*, Seiten 1150–1157, Washington, DC, USA.
- Luettel, T., Ebert, F., Fassbender, D., Fries, C., Heinrich, B. C., Jaspers, H., Kallwies, J., Naujoks, B., und Wuensche, H.-J. (2017). Assistenz- und Autonomiefunktionen zur Entlastung des Fahrers in Offroad-Szenarien. *at - Automatisierungstechnik*, 65:495–511.
- Luettel, T., Himmelsbach, M., von Hundelshausen, F., Manz, M., Mueller, A., und Wuensche, H.-J. (2009). Autonomous Offroad Navigation Under Poor GPS Conditions. In *Proceedings of 3rd Workshop On Planning, Perception and Navigation for Intelligent Vehicles (PPNIV), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Maiero, J. (2009). *Image-based Tracking*. Masterarbeit, Hochschule Bonn-Rhein-Sieg.

- Manz, M. (2013). *Modellbasierte visuelle Wahrnehmung zur autonomen Fahrzeugführung*. Dissertation, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Neubiberg.
- Manz, M., Luettel, T., von Hundelshausen, F., und Wuensche, H.-J. (2011). Monocular Model-Based 3D Vehicle Tracking for Autonomous Vehicles in Unstructured Environment. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.
- Maurer, M. (2000). *Flexible Automatisierung von Straßenfahrzeugen mit Rechnerssehen*. Dissertation, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Neubiberg.
- Mitschke, M. und Wallentowitz, H. (2004). *Dynamik der Kraftfahrzeuge*. Springer.
- Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., und Thrun, S. (2008). Junior: The Stanford Entry in the Urban Challenge. *Journal of Field Robotics*.
- Mueller, A., Manz, M., Himmelsbach, M., und Wuensche, H.-J. (2009). A Model-Based Object Following System. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, China.
- Mueller, A. und Wuensche, H.-J. (2012). Object-Related-Navigation for Mobile Robots. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 603–610, Alcalá de Henares, Spain.
- Mueller, G. R. und Wuensche, H.-J. (2016). Continuous Extrinsic Online Calibration for Stereo Cameras. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 1047–1052, Gothenburg, Sweden.
- Neckermann, L. (2017). *Corporate Mobility Breakthrough 2020*. Corporate Vehicle Observatory.
- Ojala, T., Pietikainen, M., und Harwood, D. (1994). Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 1 – Conference A: Computer Vision & Image Processing*, Seiten 582–585.
- Okutomi, M. und Kanade, T. (1993). A Multiple-Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363.
- O'Malley, R., Jones, E., und Glavin, M. (2010). Rear-lamp Vehicle Detection and Tracking in Low-exposure Color Video for Night Conditions. *Transactions on Intelligent Transportation Systems*, 11(2):453–462.

- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 9(1):62–66.
- Pantilie, C. D. und Nedevschi, S. (2010). Real-time Obstacle Detection in Complex Scenarios Using Dense Stereo Vision and Optical Flow. In *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, Seiten 439–444.
- Petermann, T. und Grünwald, R. (2011). Stand und Perspektiven der militärischen Nutzung unbemannter Systeme. In *Endbericht zum TA-Projekt, Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag*.
- Piccardi, M. (2004). Background subtraction techniques: a review. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, volume 4, Seiten 3099–3104, Sydney, NSW, Australia.
- Ramer, U. (1972). An Iterative Procedure for the Polygonal Approximation of Plane Curves. In *Computer Graphics and Image Processing*, Seiten 244–256. Academic Press Inc.
- Rauch, S., Aeberhard, M., Ardelt, M., und Kämpchen, N. (2012). Autonomes Fahren auf der Autobahn – eine Potentialstudie für zukünftige Fahrerassistenzsysteme. In *5. Tagung Fahrerassistenz, München*.
- Reif, K. (2010). *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. Bosch Fachinformation Automobil. Vieweg+Teubner Verlag. <https://books.google.de/books?id=N84R5SBct5IC>.
- Rezaei, M., Terauchi, M., und Klette, R. (2015). Robust Vehicle Detection and Distance Estimation Under Challenging Lighting Conditions. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2723–2743.
- Robert, K. (2009). Video-based Traffic Monitoring at Day and Night - Vehicle Features Detection Tracking. In *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, Sydney, Australia.
- Rosenfeld, A. und Pfaltz, J. L. (1966). Sequential Operations in Digital Picture Processing. *J. ACM*, 13(4):471–494.
- Rosten, E. und Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. In *In European Conference on Computer Vision*, Seiten 430–443.
- Rumelhart, D. E., Hinton, G. E., und Williams, R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature*, 323(Oct):533–536.
- SAE J3016 (2018). Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. SAE International Standard. [https://www.sae.org/standards/content/j3016\\_201806](https://www.sae.org/standards/content/j3016_201806) Zugriffsdatum: 28.09.2018.

- Schaub, G. W., Pfaendner, A. H., und Schaefer, C. (2004). PRIMUS: Autonomous Navigation in Open Terrain with a Tracked Vehicle. In *Unmanned Ground Vehicle Technology VI*, volume 5422, Seiten 156–165.
- Schneider, F. E., Wildermuth, D., Brüggemann, B., und Röhling, T. (2010a). European Land Robot Trial (ELROB) - Towards a Realistic Benchmark for Outdoor Robotics. In *Proceedings of the International Conference on Robotics in Education*.
- Schneider, S., Himmelsbach, M., Luettel, T., und Wuensche, H.-J. (2010b). Fusing Vision and LIDAR - Synchronization, Correction and Occlusion Reasoning. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 388–393.
- Schneider, S., Himmelsbach, M., Lüttel, T., und Wünsche, H.-J. (2010c). Fusing vision and LIDAR - Synchronization, correction and occlusion reasoning. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 388–393, San Diego, CA.
- Schneider, S., Luettel, T., und Wuensche, H.-J. (2013). Odometry-based Online Extrinsic Sensor Calibration. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seiten 1287–1292.
- Schweiger, R., Neumann, H., und Ritter, W. (2005). Multiple-Cue Data Fusion with Particle Filters for Vehicle Detection in Night View Automotive Applications. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 753–758.
- Schweitzer, M. und Wuensche, H.-J. (2009). Efficient Keypoint Matching for Robot Vision using GPUs. In *Proceedings of IEEE International Conference on Computer Vision, 5th IEEE Workshop on Embedded Computer Vision*.
- Sezgin, M. und Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *J. Electronic Imaging*, 13:146–168.
- Shladover, S. E. (2007). PATH at 20 - History and Major Milestones. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):584–592.
- Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience.
- Sivaraman, S. und Trivedi, M. M. (2011). Combining Monocular and Stereo-Vision for Real-Time Vehicle Ranging and Tracking on Multilane Highways. In *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*.
- Sivaraman, S. und Trivedi, M. M. (2012). Real-Time Vehicle Detection Using Parts at Intersections. In *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*.
- Smith, R. (2007). An Overview of the Tesseract OCR Engine. In *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, Seiten 629–633.

- Statistisches Bundesamt Deutschland (2015). Verkehrsunfälle - Fachserie 8, Reihe 7 (Jahresbericht). <https://www.destatis.de/DE/Publikationen/Thematisch/TransportVerkehr/Verkehrsunfaelle/VerkehrsunfaelleJ.html> Stand: 11.08.2016, Zugriffsdatum: 30.09.2016.
- Steyer, S., Tanzmeister, G., und Wollherr, D. (2018). Grid-Based Environment Estimation Using Evidential Mapping and Particle Tracking. *IEEE Transactions on Intelligent Vehicles*, 3:384–396.
- Suzuki, S. und Abe, K. (1985). Topological Structural Analysis of Digitized Binary Images by Border Following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46.
- Szegedy, C., Toshev, A., und Erhan, D. (2013). Deep Neural Networks for Object Detection. In *Advances in Neural Information Processing Systems 26*, Seiten 2553–2561. Curran Associates, Inc.
- Tanzmeister, G. (2016). *Grid-based Environment Estimation for Local Autonomous Vehicle Navigation*. Dissertation, Technische Universität München, München.
- Tanzmeister, G. und Wollherr, D. (2017). Evidential Grid-Based Tracking and Mapping. *IEEE Transactions on Intelligent Transportation Systems*, 18:1454–1467.
- Thomanek, F. (1996). *Visuelle Erkennung und Zustandsschätzung von mehreren Straßenfahrzeugen zur autonomen Fahrzeugführung*. Fortschritt-Berichte VDI: Reihe 12, Verkehrstechnik/Fahrzeugtechnik. VDI-Verlag.
- Thomanek, F. und Dickmanns, D. (1992). Obstacle Detection, Tracking And State Estimation For Autonomous Road Vehicle Guidance. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, Seiten 1399–1406.
- Thomanek, F., Dickmanns, E., und Dickmanns, D. (1994). Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 231–236.
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekirk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., und Mahoney, P. (2006). Winning the DARPA Grand Challenge. *Journal of Field Robotics*.
- Tipping, M. E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine. *The Journal of Machine Learning Research*, 1:211–244.
- Tsugawa, S., Jeschke, S., und Shladover, S. E. (2016). A Review of Truck Platooning Projects for Energy Savings. *IEEE Transactions on Intelligent Vehicles*, 1(1):68–77.

- Tsugawa, S., Kato, S., und Aoki, K. (2011). An Automated Truck Platoon for Energy Saving. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seiten 4109–4114.
- Unterholzner, A. (2016). *Sensor Orientation Selection and Adaptive Control of an Actuated Sensor Platform for Autonomous Vehicles*. Dissertation, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Neubiberg.
- Unterholzner, A. und Wuensche, H.-J. (2010). Hybrid Adaptive Control of a Multi-Focal Vision System. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 534–539, USA.
- Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T. M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S., Snider, J., Stentz, A., Whittaker, W., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., und Ferguson, D. (2008). Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466.
- Viola, P. und Jones, M. (2001a). Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Viola, P. und Jones, M. (2001b). Robust Real-time Object Detection. In *International Journal of Computer Vision*.
- Wagner, D., Langlotz, T., und Schmalstieg, D. (2008). *Robust and Unobtrusive Marker Tracking on Mobile Phones*. IEEE / ACM International Symposium on Mixed and Augmented Reality - S. 121–124. IEEE Computer Society, Los Alamitos, CA, USA.
- Wan, E. A. und Merwe, R. V. D. (2000). The Unscented Kalman Filter for Nonlinear Estimation. In *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC)*, Seiten 153–158, Alberta, Canada.
- Wang, H. und Brady, M. (1995). Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9):695–703.
- Watzenig, D. und Horn, M., Herausgeber (2017). *Automated Driving: Safer and More Efficient Future Driving*. Springer.
- Westhofen, D., Grundler, C., Doll, K., Brunsmann, U., und Zecha, S. (2012). Transponder- and Camera-based Advanced Driver Assistance System. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, Seiten 293–298.

- Winner, H., Hakuli, S., Lotz, F., und Singer, C. (2015). *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. ATZ/MTZ-Fachbuch. Springer Fachmedien Wiesbaden.
- Wuensche, H.-J. (1987). *Bewegungssteuerung durch Rechnersehen*. Dissertation, Universität der Bundeswehr München.
- Zhang, X., Geimer, M., Noack, P. O., und Grandl, L. (2010). Development of an Intelligent Master-Slave System between Agricultural Vehicles. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, University of California, San Diego, CA, USA.
- Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C., Kaus, E., Herrtwich, R., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knöppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M., und Zeeb, E. (2014). Making Bertha Drive - An Autonomous Journey on a Historic Route. *Proceedings of the IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20.