Uni-DAS

12. Workshop Fahrerassistenzsysteme und automatisiertes Fahren

FAS 2018

Walting im Altmühltal,

26.09. – 28.09.2018

Copyright Uni-DAS e. V.
Alle Rechte, auch das des auszugsweisen Nachdrucks, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-00-060371-6

Uni-DAS e. V
Otto-Berndt-Straße 2
64287 Darmstadt
Telefon: +49 6151 16 24200
Telefax: +49 6151 16 24205

Vorwort

Der Workshop „Fahrerassistenzsysteme und automatisiertes Fahren 2018“ knüpft als zwölfte Veranstaltung dieser Art an die erfolgreichen Vorgänger an. Und zum zehnten Mal findet er in Walting statt, wodurch dieser Ort in der Community nahezu synonym für diese Veranstaltung steht. Auch wenn die Zahl der in diesem Bereich forschenden und entwickelnden Personen seit Beginn der Reihe um mindestens eine Größenordnung gewachsen ist, bleiben wir dem Konzept der Kombination von wissenschaftlicher Tagung und interaktivem Workshop mit beschränkter Teilnehmerzahl treu, um weiterhin den Tiefgang der Diskussion zu ermöglichen.

Fahrerassistenzsysteme haben sich mittlerweile etabliert. Auch wenn sie noch viele Weiterentwicklungspotenziale besitzen und vor allem im Markt noch breiter zu finden sein könnten, ist die aktuelle Diskussion von der Einführung des automatisierten Fahrens ohne Fahrerüberwachung bestimmt. Den großen Versprechungen über die Zukunft der automatisch fahrenden Fahrzeuge stehen aber immer noch viele ungelöste Aufgaben gegenüber. Dazu gehört insbesondere die Freigabe als Abschluss der Entwicklungsphase. Unter dem Titel „Wie kommen wir aus der Freigabefalle?“ wollen wir im Workshop-Teil der Veranstaltung nach Ideen und Lösungsansätze suchen, damit die Technik nicht kurz vor der Markteinführung in der Freigabefalle stecken bleibt.

Die einzelnen Vorträge des Workshops und Beiträge dieses Tagungsbandes beleuchten ein breites Spektrum des aktuellen Wissensstandes im Bereich der Fahrerunterstützung und des automatischen Fahrens. In kaum einer anderen Reihe lässt sich so klar nachvollziehen, wie sich die Fahrerassistenzsysteme und das automatisierte Fahren seit etwa 2000 entwickelt haben.

Mein Dank gilt allen Helfern bei der Organisation des Workshops, insbesondere den Herren Philipp Junietz, Stefan Ackermann, Björn Klamann sowie den Mitgliedern von Uni-DAS e.V.. Ich wünsche allen Teilnehmern neue Einsichten, spannende Diskussionen und das Knüpfen neuer und Vertiefen alter Kontakte.

Darmstadt im September 2018

Hermann Winner

Inhaltsverzeichnis

I Absicherung

1 Wissensbasierte Szenariengenerierung für Betriebsszenarien auf deutschen Autobahnen 1
G. Bagschik, T. Menzel, C. Körner und M. Maurer, TU Braunschweig

2 Detaillierung einer stichwortbasierten Szenariobeschreibung für die Durchführung in der Simulation am Beispiel von Szenarien auf deutschen Autobahnen 15
T. Menzel, G. Bagschik, L. Isensee, A. Schomburg, und M. Maurer, TU Braunschweig

3 Funktionale Dekomposition - Ein Beitrag zur Überwindung der Parameterraumexplosion bei der Validation von höher automatisiertem Fahren 27
C. Amersbach, H. Winner, TU Darmstadt

4 A Traffic-based Method for Safety Impact Assessment of Road Vehicle Automation 37
C. Rösener, F. Henncke, J. Sauerbier, A. Zlocki, D. Kemper, L. Eckstein, M. Oeser, RWTH Aachen

5 Ein Fehlermodell zur Manipulation des Fahrerverhaltens in virtuellen Umgebungen zum Testen automatisierter Fahrfunktionen 47
B. Huber, C. Sippl, S. Weber, R. German, Audi AG

II Human Factors

6 Mensch-Maschine-Interaktion bei manuellem und automatisiertem kooperativen Fahren an Auffahrten und Kreuzungen 56
A. Kraft, C. Maag, A. Neukum, M. Baumann, WIVW

7 Der Toleranzansatz als Methodik zur Bewertung der Kontrollierbarkeit höherer Eingriffsstärken bei Falschauslösung eines systeminitiierten Ausweichmanövers mit ESP-Aktorik 67
N. Schneider, M. Schmitz, L. Ahrens, C. Löffler, A. Neukum, WIVW

8 Cooperative Approach to Overcome Automation Effects During the Transition Phase of Conditional Automated Vehicles 79
C. Karakaya, L. Kalb, K. Bengler, TUMDonnerstag

III Fahrerzustand

9 Die Erfassung des Fahrerzustands als Voraussetzung für höher automatisierte Fahrfunktionen – Eine kritische Diskussion und ein Lösungsvorschlag 87
M. Schwalm, N. Herzberger, RWTH Aachen

10 Tracking des Aufmerksamkeitsziels des Fahrers mittels Multi-Hypothesen Multi-Modell Filter 95
J. Schwehr, V. Willert, TU Darmstadt

11 Using Past Maneuver Executions for Personalization of a Driver Model 106
H. Dang, J. Fürnkranz, TU Darmstadt

IV Wahrnehmung

12 Modeling of Drivable Free Space with Fused Camera Data for Autonomous Driving 116
O. Speidel, F. Gies, K. Dietmayer, Universität Ulm

13 Extended Object Tracking With Transparency Mitigation 126
T. Hirscher, H. Herrmann, M. Heimberger, J. Honer, Valeo GmbH

14 Fusion of Camera and Lidar Data for Object Detection using Neural Networks 138
E. Schröder, M. Mählisch, F. Hamker, Audi AG

V Prädiktion und Planung

15 Predicting Objects of Interest with Deep LearningM. 147
S. Hoermann, K. Dietmayer, Universität Ulm

16 Differentielle Dynamische Programmierung in der Trajektorienstabilisierung 157
A. Hildebrandt, G.Witt, M. Kumar, S. Kraus, MAN

Wissensbasierte Szenariengenerierung für Betriebsszenarien auf deutschen Autobahnen

G. Bagschik, T. Menzel,* C. Körner† und M. Maurer‡

Zusammenfassung: Szenarienbasierte Testkonzepte, vorrangig für Simulationsumgebungen, können einen Beitrag zur Freigabe und Markteinführung automatisierter Fahrzeuge leisten. Kernbestandteil der Ansätze sind relevante Betriebsszenarien, welche in vielfältiger Variation die Repräsentativität der simulierten Testfälle gegenüber der realen Einsatzumgebung erhöhen. Die notwendigen Szenarien können durch Messfahrten, Expertenanalysen oder systematische Prozesse gewonnen werden. Eine breite Variation umfassend beschriebener Szenarien kann im Rahmen von Expertenanalysen oder Messfahrten nur mit hohem manuellem Aufwand erzeugt werden. In diesem Beitrag werden Ansätze zur Formalisierung von Wissen und zur wissensbasierten automatisierten Szenariengenerierung für Betriebsszenarien auf deutschen Autobahnen vorgestellt.

Schlüsselwörter: Automatisiertes Fahren, Ontologie, Szenarien, Testprozess

1 Einleitung

Wachenfeld und Winner [1] sagen Milliarden von notwendigen Kilometern für den streckenbasierten Test von automatisierten Fahrzeugen voraus. Das Gedankenexperiment geht von Testansätzen für Fahrerassistenzsysteme aus und prädiziert unter definierten Annahmen die benötigten Testumfänge für automatisierte Fahrfunktionen. Die resultierende hohe Anzahl zu fahrender Kilometer lässt sich als Notwendigkeit eines vielfältigen Szenarienraums interpretieren. Für verschiedene Testziele (z.B. Sicherheit oder Komfort) von automatisierten Fahrzeugen werden unterschiedlich ausgeprägte Szenarien benötigt. In einem risikobasierten Testprozess werden kritische Szenarien analysiert, wohingegen normale Betriebsszenarien für die Auslegung des Komforts einer Fahrfunktion benötigt werden. Für beide beispielhaften Testziele können aus einem allgemein formulierten Katalog von Betriebsszenarien relevante Szenarien als Grundlage für Testfälle identifiziert werden. Die Vorteile einer automatisierten Szenariengenerierung gegenüber der Sammlung von Szenarien, beispielsweise durch Messfahrten, sind zum einen die Nachvollziehbarkeit der Inhalte im erstellten Szenarienraum und zum anderen eine bessere Kosten- und Zeiteffizienz.

*G. Bagschik und T. Menzel sind wissenschaftliche Mitarbeiter am Institut für Regelungstechnik der Technischen Universität Braunschweig (Email: {bagschik, menzel}@ifr.ing.tu-bs.de).

†C. Körner ist Masterabsolvent der Technischen Universität Braunschweig (Email: c.koerner@online.de).

‡M. Maurer ist Professor und Institutsleiter am Institut für Regelungstechnik der Technischen Universität Braunschweig (Email: maurer@ifr.ing.tu-bs.de).

Schuldt [2] stellt ausgehend von der Notwendigkeit neuer Testansätze einen Beitrag zu szenarienbasierten Testkonzepten für automatisierte Fahrzeuge in simulativen Umgebungen vor. Der von Schuldt et al. [3] vorgestellte *modulare virtuelle Testbaukasten* sieht vor, durch eine Sensitivitätsanalyse Einflüsse von Szenarien-Parametern auf den Testgegenstand abzuleiten. Ulbrich et al. [4] untersuchen die Begriffe Szene, Szenario und Situation und schlagen aufbauend auf diversen Vorarbeiten konsolidierte Definitionen im Bereich des automatisierten Fahrens vor. Weiterführend leiten Bagschik et al. [5] ausgehend von der Verwendung von Szenarien im Entwicklungsprozess drei Beschreibungsformen für Szenarien ab. *Funktionale Szenarien* stellen die höchste Abstraktionsebene dar. Die Inhalte eines funktionalen Szenarios sind sprachlich formuliert und somit für menschliche Experten im Entwicklungsprozess verständlich. Während des Entwicklungsprozesses kann der Inhalt eines Szenarios fortlaufend detailliert werden und logisch (Parameterraum in Wertebereichen) oder konkret (Auswahl konkreter Repräsentanten der Wertebereiche) beschrieben werden.

In der Fallstudie von Schuldt [2] werden in der Nomenklatur nach Bagschik et al. [5] mehrere konkrete Szenarien aus einem logischen Szenario erstellt, welches auf ein grob beschriebenes funktionales Szenario („Arbeitsstelle“) zurückzuführen ist [2, S. 184]. Schuldt [2] behandelt somit die Konkretisierung von logischen Szenarien zu konkreten Szenarien (vgl. [5]) im Rahmen der Testfallerstellung. Ein wichtiger Bestandteil von szenarienbasierten und simulativen Testansätzen ist die vorgelagerte Szenariengenerierung [2, S. 205].

In diesem Beitrag wird ergänzend zu Schuldt [2] ein Ansatz vorgestellt, der sprachlich gefasste Informationsquellen in einer Ontologie als Wissensbasis modelliert. Aus dieser Wissensbasis können automatisiert funktionale Szenarien generiert werden. Die erstellten funktionalen Szenarien bilden einen Beitrag für die Entwicklung und den Test automatisierter Fahrfunktionen.

Im Folgenden werden zunächst verwandte Arbeiten wissensbasierter Systeme im Bereich des automatisierten Fahrens vorgestellt. Darauf aufbauend wird die *wissensbasierte Szenariengenerierung* als Ansatz eingeführt. Im ersten Teil des Abschnitts 3 wird der Aufbau der Wissensbasis und anschließend der Prozess der Szenariengenerierung beschrieben. Der Artikel schließt mit einem Fazit und einem Ausblick auf zukünftige Arbeiten.

2 Stand der Technik und verwandte Arbeiten

Bereits in den 1980er Jahren wurden wissensbasierte Ansätze zur natürlichsprachlichen Beschreibung von Verkehrsszenarien vorgestellt. Das System *Natürlichsprachliche Beschreibung von Objektbewegungen in einer Straßenszene* (NAOS) [6] interpretiert Bildfolgen ausgehend von einer extrahierten geometrischen Szenenbeschreibung. Aus diesen Daten werden anschließend Ereignisse extrahiert und diese unter Verwendung einer Wissensbasis interpretiert. Das Ergebnis der Interpretation wird zu einer sprachlichen Beschreibungen der Handlung des Szenarios überführt. Mehrere verwandte Arbeiten zur natürlichsprachlichen Beschreibung von Verkehrsszenarien in Bildfolgen stammen aus dem Sonderforschungsbereich (SFB) 314 der Deutschen Forschungsgemeinschaft (DFG) *Künstliche Intelligenz und wissensbasierte Systeme* von 1985 bis 1995. Dort sind vor allem die Arbeiten um Prof. Nagel vom Institut für Algorithmen und Kognitive Systeme der Universität Karlsruhe hervorzuheben (z.B. [7], [8] und zusammenfassend [9]). Die Vorarbeiten zu natürlichsprachlichen Beschreibungen von Bildfolgen bilden thematisch eine der Grundla-

gen zur Generierung von funktionalen Szenarien. In diesen Arbeiten wird die Bedeutung von Bildfolgen sprachlichen Ausdrücken zugeordnet, um diese für menschliche Experten von einer mathematischen Modellierung zu abstrahieren.

Dickmanns [10] stellt einen Ansatz zur Echtzeit-Interpretation von Bildfolgen in Fahrzeugsystemen vor. Dabei werden in weiteren Arbeiten [11] Annahmen über den Aufbau des Straßennetzwerks und die Bewegungen von Verkehrsteilnehmern in mathematische Modelle überführt. Welche Modelle miteinander kombiniert werden dürfen, wird durch Regeln in einer Wissensbasis hinterlegt. Aufgrund dieser Wissensbasis kann für konkrete Beobachtungen in einem Kamerabild die Zusammenstellung der gültigen Modelle gewählt werden. Mit der Zuordnung von mathematischen Modellen zu einem Objekt in der wahrgenommenen Welt kann eine Kopplung der Modelle zu einer *höhere* Bedeutung dargestellt werden. Diese Verknüpfung gibt einen Hinweis darauf, dass das zu modellierende sprachlich gefasste Hintergrundwissen des hier vorgestellten Ansatzes möglichst eindeutig auf Parametersätze abbildbar sein sollte.

Weiterhin werden wissensbasierte Systeme im Bereich des Szenenverständnisses von automatisierten Fahrzeugen angewandt. Die wissensbasierte Herangehensweise beim Szenenverständnis erlaubt es, gemessene Sensordaten um Kontextinformationen zu erweitern und somit logische Inferenz zu ermöglichen. Im Grundkonzept kann eine Verhaltensentscheidung auf eine Ontologie als Wissensrepräsentation zugreifen, um eine Szene auszuwerten und somit beispielsweise Fahrmanöver aufgrund semantischer Regeln einzuleiten [12]. In der Forschung und Entwicklung unterscheiden sich die vorgestellten Ontologien in ihren Inhalten sowie ihrer Komplexität. Nach Lattner et al. [12] können sprachlich gefasste Bestandteile einer Verkehrsszenenbeschreibung in einer Wissensbasis wie folgt gegliedert werden:

- Objekt-Klassen
- Topologische Information
- Räumliche Relationen
- Geschwindigkeitsinformation
- Distanz-Information
- Straßennetzwerk
- Verkehrssituationen
- Hintergrundwissen

Hummel et al. [13] nutzen ein wissensbasiertes System für das Szenenverständnis an urbanen Kreuzungen. Im Hintergrundwissen sind dafür Bestandteile von Kreuzungen mit geometrischen Klassen (Kreis, Gerade, Klothoide) verknüpft und die Relationen der Bestandteile über ein Regions-Verknüpfungskalkül (engl.: region connection calculus (RCC8)) beschrieben. Das Verknüpfungskalkül beschreibt Überschneidungsmöglichkeiten der geometrischen Klassen ähnlich der Mengenlehre aus der Mathematik. Mit diesem Konzept werden in dem vorgestellten Ansatz Teile-Ganze-Beziehungen der geometrischen Bestandteile modelliert. Beispiel einer konkreten Beobachtung dieser Beziehungen wäre eine Markierung die logisch einen *Bestandteil* eines Fahrstreifens darstellt und geometrisch daher *innerhalb* der Fahrbahnbegrenzung liegen muss. Um komplexe Kreuzungen zu erfassen, können einzelne Bestandteile semantisch miteinander verknüpft werden. Die Verknüpfungen sind beispielsweise *parallel zu, Nachbar von, Nachfolger von, gespiegelt gegenüber* und *nicht verbunden*. Diese Konzepte ermöglichen die Überprüfung von Sensormessungen gegenüber dem modellierten Hintergrundwissen, welche hier nicht weiter

betrachtet wird, da die Zuordnung von Sensordaten in dem hier gezeigten Ansatz keine Verwendung findet.

Armand et al. [14] nutzen ebenfalls ein wissensbasiertes System für das Szenenverständnis automatisierter Fahrzeuge. Der Fokus des modellierten Hintergrundwissens liegt auf Verkehrsteilnehmern und deren Interaktionen sowie Objekteigenschaften, die sich aus Objekthypothesen aus Sensordaten ableiten lassen. Objekteigenschaften bestehen aus einem Katalog von Fahrmanövern mit Bezug zur Infrastruktur oder anderen Verkehrsteilnehmern (*bewegt sich zu, folgt, ist nahe bei*) und mögliche Handlungen, die sich aus Regeln schlussfolgern lassen (*muss anhalten, wird erreichen, wird bremsen*).

Zhao et al. [15] verknüpfen die vorher vorgestellten Arbeiten zu einer einheitlichen Modellierung des Hintergrundwissens über städtische Szenarien. Darüber hinaus erweitern die Autoren das Wissen um formalisierte Verkehrsregeln wie Vorfahrtsregeln. Die Schlussfolgerungen aus Sensordaten werden in der Veröffentlichung mittels einer Anfragesprache (SPARQL) für die Verwendung unter Echtzeitbedingungen im Fahrzeug optimiert und evaluiert, was für den hier vorgestellten Ansatz nicht weiter von Bedeutung ist.

Aus den vorgestellten Arbeiten können zusammenfassend Teile der Wissensbasen für den hier vorgestellten Ansatz Beiträge zur Modellierung des Hintergrundwissens liefern. So werden in Teilen Objekteigenschaften und hierarchische Klassenkonzepte ähnlich zu Armand et al. und Hummel et al. verwendet. Die Formalisierung von Verkehrsregeln als Bestandteil der Wissensbasis ist von Zhao et al. inspiriert. Methodisch verfolgen die bisherigen Ansätze das Ziel, aus Szenarien (Beobachtungen zu Instanzen) des Verkehrsgeschehens Wissen über die aktuelle Situation des eigenen Fahrzeugs aus einer Wissensbasis (Hintergrundwissen) abzuleiten. Der hier beschriebene Ansatz leitet dagegen aus einer bestehenden Wissensbasis (Hintergrundwissen) mögliche funktionale Szenarien (Beobachtungen zu Instanzen) ab. Nach Kenntnisstand der Autoren existieren keine verwandten Ansätze zu diesem Vorgehen, insbesondere für die Generierung von sprachlich beschriebenen Verkehrsszenarien. Daher ist es notwendig, die Modellierung des Hintergrundwissen für die Erstellung von Szenarien zu erweitern, wie es im Folgenden vorgestellt wird.

3 Wissensbasierte Szenariengenerierung

Der Prozess zur wissensbasierten und systematischen Szenariengenerierung ist, wie in Abbildung 1 dargestellt, in zwei Teile gegliedert. Im ersten Teil wird sprachlich gefasstes Wissen über das Verkehrsgeschehen auf deutschen Autobahnen identifiziert und formalisiert. Das Ergebnis dieses Teilschritts ist eine Wissensbasis, die im hier vorgestellten Ansatz durch eine Ontologie in der *Web Ontology Language* (OWL) implementiert ist. Das Hintergrundwissen wird durch hierarchische Klassen und semantische Beziehungen und Restriktionen (Klassenbeziehungen) zwischen diesen Klassen modelliert. Ontologien ermöglichen weiterhin logische Inferenz (Schlussfolgerungen) durch sogenannte Reasoner[1]. Als Grundlage für die Inferenz sind semantische Regeln (Semantic Web Rules) in Form von wenn-dann-Ausdrücken in der Ontologie hinterlegt. Reasoner sind in der Lage, Instanzen auf das Hintergrundwissen zuzuordnen und auf diese Weise aktuell geltende Regeln zu ermitteln. Dadurch können neue Beziehungen zwischen konkreten Instanzen in einem Szenario aus dem Hintergrundwissen geschlussfolgert werden.

[1] Für den Begriff *Reasoner* existiert keine passende deutsche Übersetzung.

In diesem Beitrag wird die *Richtlinie zur Anlage von Autobahnen* (RAA) [16] als Fallbeispiel für die Formalisierung des Wissens über den Streckenbau auf deutschen Autobahnen genutzt. Aus der Wissensbasis werden im zweiten Teil der Szenariengenerierung automatisch und systematisch Szenarien durch Variation der Klassen und ihrer Instanzen abgeleitet. Die Variation geschieht unter Berücksichtigung von möglichen Zusammenstellungen und Einschränkungen, die durch die Wissensbasis vorgegeben sind. Die generierten Szenarien sind zum einen für Menschen lesbar und zum anderen für nachfolgende automatisierte Verarbeitungsschritte durch Rechnersysteme interpretierbar.

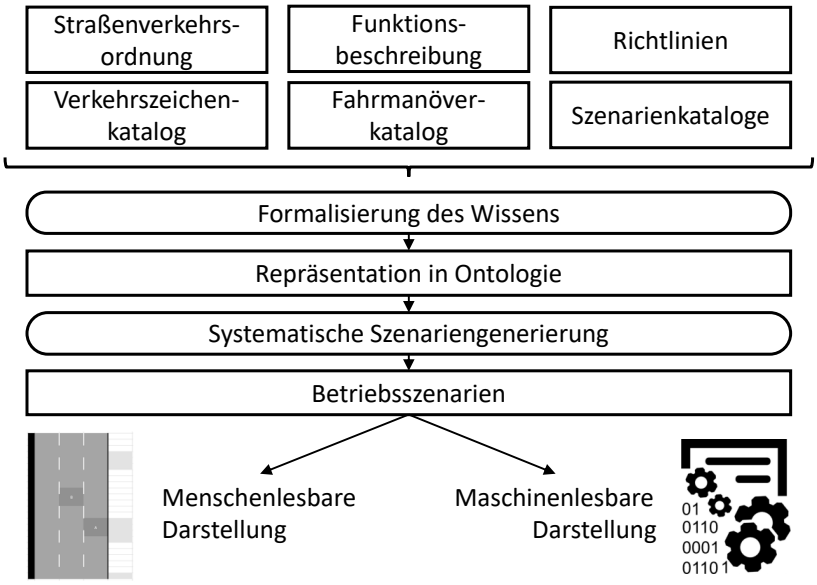


Abbildung 1: Prozessübersicht der wissensbasierten Szenariengenerierung. Abgerundete Rechtecke stellen Arbeitsschritte und normale Rechtecke Prozessergebnisse und Wissensquellen dar.

3.1 Modellierung von Hintergrundwissen

Das Hintergrundwissen in der Wissensbasis ist nach einem 5-Ebenen-Modell (Abbildung 2) strukturiert. Das Modell wurde aus den Vorarbeiten von Schuldt [3] für die Repräsentation in einer Ontologie und die automatisierte Generierung von funktionalen Szenarien angepasst.

Straßenebene Die erste Ebene beschreibt die Basisstrecke, auf der ein Szenario durchgeführt werden kann. Entgegen des Vorschlags von Schuldt [2] wird im hier vorgestellten Ansatz auf der ersten Ebene nur die Geometrie und die Topologie der Streckenabschnitte modelliert. Die Straßenausstattung wie statische Objekte oder Verkehrszeichen werden im Gegensatz zu Schuldt [2] gesondert auf der zweiten Ebene E2 betrachtet. Dies lässt die Generierung mehrerer Szenarien auf einer Basisstrecke durch Anpassung der statischen Bebauung und verschiedenen Verkehrsregeln wie Geschwindigkeitsbegrenzungen zu. In der Aufteilung der Ebenen verbirgt sich die Annahme, dass optionale Verkehrsregeln, die häufig durch Verkehrszeichen instanziiert werden, semantische Unterschiede für automatisierte Fahrzeuge mit Umfeldwahrnehmung und -interpretation und somit ein anderes

Abbildung 2: 5-Ebenen-Modell zur Strukturierung des Wissens über Szenarien basierend auf Schuldt [2]

Szenario ausmachen. Die Existenz der Elemente der Basisstrecke und ihre Strukturierung ist durch hierarchische Klassen in der Ontologie modelliert. Die Auslegung von Autobahnen wird in Entwurfsklassen (EKA), die jeweils mehrere Regelquerschnitte und ihre Elemente enthalten, unterteilt. Ein Streckenabschnitt besteht aus einer Richtungsfahrbahn und einigen teilweise optionalen Ausstattungselementen. Die Richtungsfahrbahn wird durch einen Regelquerschnitt der RAA instanziiert. Elemente der Regelquerschnitte wie Fahrstreifen und Seitenstreifen sind als gesonderte Klassen modelliert und werden über die Objektbeziehung *besteht aus* mit einer genauen Anzahl modelliert. So *besteht* ein Regelquerschnitt 36 beispielsweise aus *genau drei* Fahrstreifen pro Fahrtrichtung.

Straßenausstattung Die zweite Ebene modelliert die Straßenausstattung[2], welche die Begrenzungen der Fahrbahn, Verkehrsregeln und ihre Instanziierungen (Verkehrszeichen, Markierungen, Wechselzeichen) und statische Objekte des Szenarios enthält. In der Wissensbasis sind semantische Regeln (Semantic Web Rules) hinterlegt, welche Verkehrszeichen unter welchen Umständen auf den jeweiligen Regelquerschnitten angebracht sein dürfen.

Temporäre Beeinflussung Die dritte Ebene sieht konzeptionell die Modellierung von temporären Veränderungen der Ebenen E1 und E2 vor, welche in der vorliegenden Arbeit noch nicht umgesetzt ist. Arbeitsstellen wurden in den Vorarbeiten von Schuldt [2] als Beispiel zur Generierung von *logischen Szenarien* [5] genutzt. Teile des Regelwerks für Arbeitsstellen sind in der RAA enthalten und in den Richtlinien für die Absicherung von Arbeitsstellen an Straßen (RSA) [17] detailliert beschrieben.

[2]Benennung nach Hinweis durch Prof. Lutz Eckstein in persönlicher Kommunikation.

Bewegliche Objekte In der vierten Ebene sind bewegliche Objekte (vorrangig Verkehrsteilnehmer) und ihre Interaktionen beschrieben. Kategorien von Verkehrsteilnehmern (z.B. Pkw und Lkw) sind als Klassen der Ontologie modelliert. Interaktionen zwischen Verkehrsteilnehmern sind als Fahrmanöver hinterlegt. Einen Katalog von notwendigen Fahrmanövern für ein vollständig automatisiertes Fahrzeug auf Abruf wurde von Reschka [18] basierend auf Tölle [19] und Nagel und Enkelmann [8] vorgestellt. Für den hier gezeigten Ansatz wurde der Katalog von Reschka um das Manöver *Zurückfallen* als Gegenteil zum *Annähern* erweitert. Mit dieser Erweiterung lassen sich Bedingungen von Relativgeschwindigkeiten für eine spätere Überführung in logische Szenarien explizit unterscheiden.

Umweltbedingungen Die fünfte Ebene beschreibt Umweltbedingungen bestehend aus Wetter, Jahreszeit und Tageszeit. Diese Elemente wurden durch hierarchische Klassen grob gegliedert (z.B. Regen, Schnee, Sonnenschein) oder diskretisiert (Morgen, Mittag, Abend, Nacht). Feinere Diskretisierungen sind in der Überführung zu logischen Szenarien vorgesehen, da die hier beschriebenen Stufen durch Parameter wie Stunden und Minuten quasi-kontinuierlich beschrieben werden können.

Klassenbeziehungen Neben der Modellierung von hierarchischen Klassenkonzepten zur Beschreibung des Straßenverkehrs werden Beziehungen zwischen diesen Klassen modelliert. In diesem Ansatz sind drei Arten von Beziehungen unterschieden: Implementierungsbeziehungen, Konstellationsbeziehungen und Manöverbeziehungen.

Implementierungsbeziehungen modellieren die einstellbaren Parameter Komplexitätslevel und Abstraktionslevel sowie die dazugehörigen Standardklassen. Jeder Klasse kann somit die Zugehörigkeit zu einem Komplexitäts- und Abstraktionslevel zugewiesen werden, was in Abschnit 3.2 näher beschrieben ist.

Konstellationsbeziehungen sind weiter in Erstellungs- und Anordnungsbeziehungen unterschieden. Erstellungsbeziehungen beschreiben abstrakte (nicht „greifbare“) Klassen und deren Repräsentanten in der realen Umwelt. Die Konzepte dafür sind Teile-Ganze-Beziehungen (Regelquerschnitt *besteht aus* einer Anzahl von Elementen), Instanziierungen (Richtungsfahrbahn *wird instanziiert durch* Regelquerschnitt) und die Beschreibung von optionalen Bestandteilen wie Geschwindigkeitsbegrenzungen. Anordnungsbeziehungen beschreiben räumliche Relationen von Instanzen wie *folgt auf/gefolgt von*, *rechter/linker Nachbar* und *angebracht auf*.

Manöverbeziehungen sind für die Generierung von Fahrmanövern als Interaktion zwischen Verkehrsteilnehmern notwendig. Diese Beziehungen beschreiben Positionsrelationen, Geschwindigkeitsrelationen und daraus resultierende Abstände im Umfeld des jeweiligen Verkehrsteilnehmers. Auf Grundlage dieser Informationen können bei der Szenarienerstellung Fahrmanöver automatisch geschlussfolgert werden (vgl. Abschnitt 3.2). Während Implementierungs- und Erstellungsbeziehungen vor der Szenariengenerierung als Hintergrundwissen modelliert werden, können Anordnungs- und Manöverbeziehungen erst nach Erstellung der Szenerie und Instanziierung der Verkehrsteilnehmer für jedes Szenario einzeln ausgewertet werden.

3.2 Systematische Szenariengenerierung

Das im vorherigen Abschnitt beschriebene Hintergrundwissen liefert die Grundlage für die systematische Szenariengenerierung. Im Folgenden werden zunächst die bei der Generierung einstellbaren *Parameter* beschrieben. Anschließend folgt eine Darstellung der Prozessschritte *Szeneriegenerierung, Verkehrsteilnehmergenerierung, Szenarienentwicklung* und *Export der Szenarien.*

Parameter Die Szenariengenerierung kann durch vier Parameter beeinflusst werden: Komplexitätsgrad, Abstraktionsgrad, Anzahl der Positionen pro Fahrstreifen und Anzahl der Verkehrsteilnehmer. Der Komplexitätsgrad der Szenariengenerierung gibt an, welche Klassen der 5 Ebenen in die Variationsmethode einbezogen werden. Bei einem niedrigen Komplexitätsgrad werden beispielsweise Verkehrsteilnehmer der Klasse Pkw und Knotenpunkt-freie Streckenabschnitte generiert. Mit steigendem Komplexitätsgrad werden Verkehrsteilnehmer wie Lkw und Busse und Regelquerschnitte auf Brücken und in Tunneln während der Generierung berücksichtigt.

Der Abstraktionsgrad gibt an, welche Subklassen der durch den Komplexitätsgrad zulässigen Superklassen variiert werden. In der abstraktesten Konfiguration werden beispielsweise nur Streckenabschnitte mit zwei Fahrstreifen und nachgiebigen Rückhaltesystemen generiert. Mit sinkendem Abstraktionsgrad werden mehr Variationsmöglichkeiten (drei Fahrstreifen und starre Begrenzungen) in die Variation integriert. Für jede Superklasse kann eine Subklasse dabei als Standard-Ausprägung gewählt werden, welche zum Beispiel durch statistische Analysen („Standard-Autobahn“) oder eine besondere Bedeutung für die Szenariengenerierung bestimmt werden kann. Die Standardklassen werden genutzt, um für jede Ebene in der Klassenhierarchie mindestens einen Repräsentanten auszuwählen, auch wenn die Szenariengenerierung mit einem hohen Abstraktionsgrad konfiguriert ist. Abbildung 3 zeigt die Parameter Komplexitätsgrad und Abstraktionsgrad in der hierarchischen Modellierung des Hintergrundwissen. Die Parameter Anzahl der Positionen und Anzahl der Verkehrsteilnehmer werden in den nächsten Absätzen erläutert.

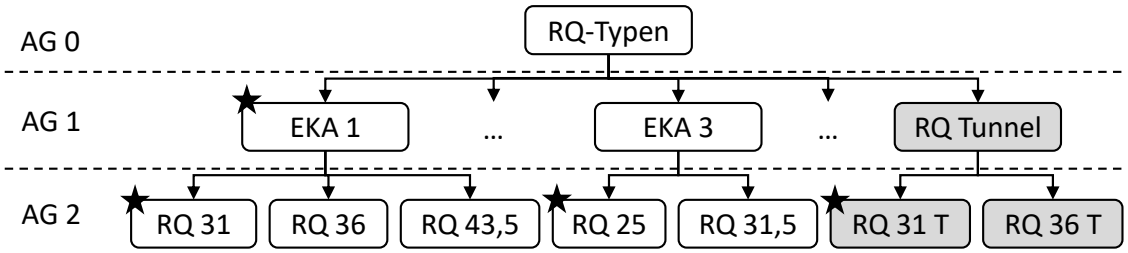


Abbildung 3: Einfluss der Parameter der Szenariengenerierung. Rechtecke (weiß = Komplexitätsgrad 1, grau = Komplexitätsgrad 2) stellen Klassen der Ontologie dar. Ebenen im Bild bilden Abstraktionsgrad (0 = größte Abstraktion) ab. Mit Stern markierte Klassen sind Standardklassen. Abkürzungen: Entwurfsklasse (EKA), Regelquerschnitt (RQ), Abstraktionsgrad (AG)

Szeneriegenerierung Im ersten Schritt der Szeneriegenerierung wird ermittelt, welche zulässigen Bestandteile die Szenerien enthalten. Welche Bestandteile in der Szenariengene-

rierung berücksichtigt werden sollen, hängt von der Konfiguration des Komplexitäts- und des Abstraktionsgrads ab. Dazu werden die Ebenen E1 und E2 des 5-Ebenen-Modells ausgewertet. Die Auswertung des Hintergrundwissens beginnt bei der abstrakten Klasse *Szene*, welche unter anderem mindestens einen Streckenabschnitt zur Instanziierung benötigt. Streckenabschnitte sind über die Klassenbeziehung *instaziiert durch* semantisch zu Richtungsfahrbahnen und somit mit den Regelquerschnitten verknüpft [16, Kap. 4]. Die Kombinationen der Elemente werden gemäß der Konfiguration erzeugt und in einem *Instanziierungsgraphen* gespeichert. Mehrere Möglichkeiten der Kombination in einer Klasse ergeben dabei mehrere Verzweigungen im resultierenden Graphen. Für die in Abbildung 3 gezeigten Konfiguration würden für die Parameter Abstraktionsgrad AG 1 und Komplexitätsgrad KG 1 Szenerien mit den Regelquerschnitten RQ 31 (aus der Entwurfsklasse EKA 1) und RQ 25 (aus der Entwurfsklasse EKA 3) erzeugt werden. Bei Abstraktionsgrad AG 2 hingegen wären alle Regelquerschnitte der Entwurfsklassen EKA 1 und EKA 3 in den resultierenden Szenerien enthalten.

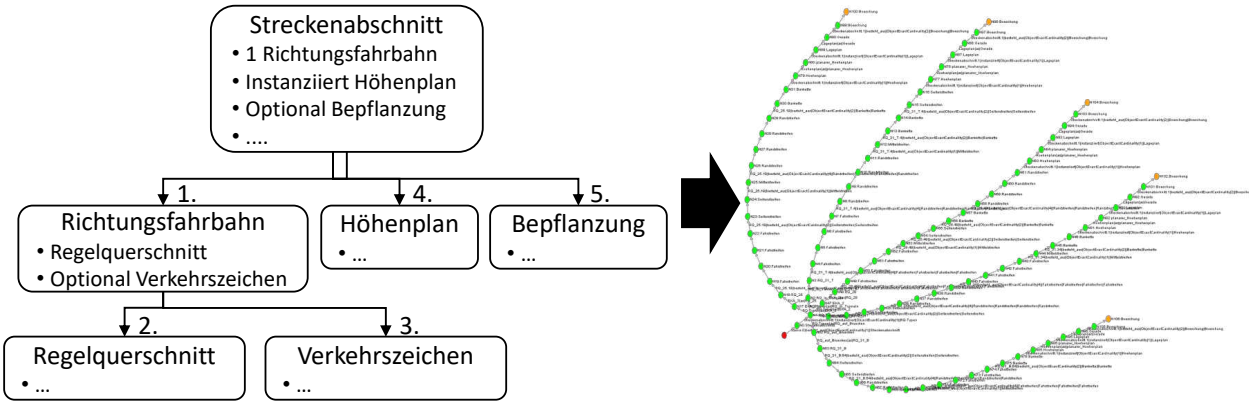


Abbildung 4: Auswertung des Hintergrundwissens mit Reihenfolge (Zahlen an Pfeilen) und Ausschnitt des resultierenden Instanziierungsgraphen

Abbildung 4 zeigt beispielhaft die Reihenfolge der Auswertung und einen Ausschnitt des resultierenden Graphen der Auswertung des Hintergrundwissens gemäß des konfigurierten Abstraktions- und Komplexitätsgrads. Neben den Bestandteilen der Szenerie aus der RAA wird die konfigurierte Anzahl von Positionen pro Fahrstreifen als Instanzen in den Graphen eingefügt. Diese Positionen sind für die nachfolgende Generierung von Verkehrsteilnehmern notwendig und werden räumlich den Fahrstreifen und sich selbst zugeordnet. Jeder Pfad durch den Graphen stellt abschließend die enthaltenen Elemente einer Szenerie dar, die für die Szenariengenerierung relevant sind. Für die weiteren Prozessschritte werden aus allen Bestandteilen des Graphen konkrete Instanzen mit eindeutiger Kennung (beispielsweise Fahrstreifen$1) erstellt.

Die Beschreibung eines Szenarios beinhaltet neben der Existenz von Instanzen der Bestandteile auch deren räumliche Relationen [20, Kap. 4.2]. Dafür werden Instanzen des Graphen überprüft, welche mit einer oder mehreren Anordnungsbeziehungen modelliert sind. Die Reihenfolge der Auswertung der Klassenbeziehungen stellt eine entscheidende Rolle dar. Zunächst wird eine einfache Anordnung durchgeführt und die Individuen identifiziert, welche „Ganze“-Konzepte (Superklassen) von Teile-Ganze-Beziehungen darstellen. „Teile“-Konzepte sind durch die jeweiligen Subklassen modelliert. In einem Teile-Ganze-Konstrukt sind Prioritäten der Anordnung modelliert, welche für die Bestimmung der

Reihenfolge genutzt werden. Auf diese Weise steht fest, in welcher Reihenfolge die Restriktion für die Anordnung der Instanzen ausgewertet werden müssen. Beispielsweise werden in einem Regelquerschnitt erst die Anordnungen der Fahrstreifen zueinander, danach die Anordnung von Seiten- und Randstreifen und abschließend optionale Elemente generiert. Im zweiten Schritt der Anordnung werden übergreifende Zusammenhänge der einzelnen Teile-Ganze-Konstrukte erzeugt. Dazu gehören die fahrstreifenübergreifenden Relationen von Positionen in den Fahrstreifen und Markierungen. Die übergreifenden Beziehungen können mithilfe von modellierten semantischen Regeln durch einen Reasoner automatisch geschlussfolgert werden.

Als Ergebnis der Teilschritte der Szenerieerstellung liegen Instanzen der zulässigen Ausprägungen der Szenerie mit räumlichen Relationen vor (Abbildung 5).

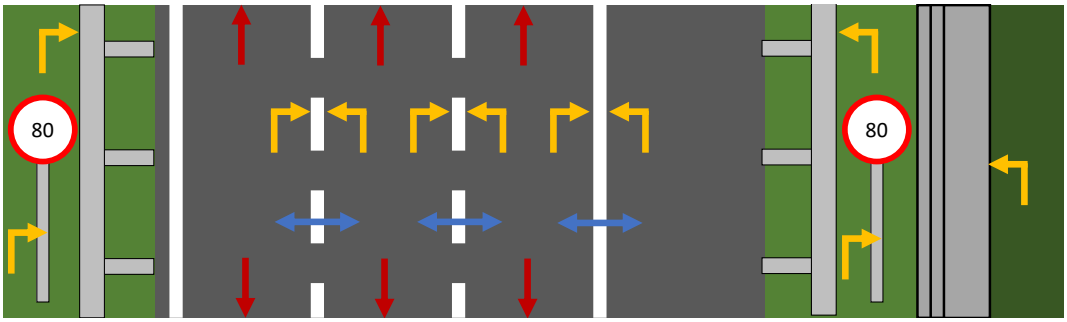


Abbildung 5: Ausschnitt der Anordnung von Szenerieelementen. Blaue Pfeile: *rechts/links*, orange Pfeile: *angebracht auf*, rote Pfeile: *folgt auf/gefolgt von*.

Verkehrsteilnehmergenerierung Die Generierung von Verkehrsteilnehmern beginnt analog zur Erstellung der Szenerie. Dabei werden zulässige Fahrzeugklassen und deren Kombinationen als Instanziierungsgraph gespeichert und in Instanzen überführt. Bei drei Fahrzeugen mit Pkw und Lkw ist es beispielsweise möglich, drei Pkw, zwei Pkw und einen Lkw, einen Pkw und zwei Lkw und drei Lkw zu erzeugen. Anschließend werden die Instanzen der Verkehrsteilnehmer in allen Permutationen auf den vorhandenen Positionen der Fahrstreifen verteilt. Dopplungen werden durch die Permutation mit Wiederholung (ohne Reihenfolge) vermieden. Semantische Dopplungen liegen vor, wenn auf funktionaler Ebene kein Unterschied der Szenarien vorhanden ist: Bei zwei hintereinander fahrenden Fahrzeugen ist kein Unterschied vorhanden, wenn die Fahrzeuge der gleichen Fahrzeugklasse angehören (PKW\$1 vor PKW\$2 = PKW\$2 vor PKW\$1). Wenn beide Fahrzeuge unterschiedlichen Klasse angehören, bedeutet dies einen Unterschied für das entstehende Szenario (LKW\$1 vor PKW\$1 $\neq$ PKW\$1 vor LKW\$1). Dieses Vorgehen führt zu einer Reduktion der Szenarienmenge bei gleicher qualitativer Aussagekraft des Szenarienkatalogs. Aufgrund der resultierenden räumlichen Verteilung der Verkehrsteilnehmer können Fahrmanöver geschlussfolgert werden. Jedes Fahrmanöver aus dem Fahrmanöverkatalog (Abschnitt 3.1) ist in einem Satz von semantischen Regeln (Semantic Web Rules) formalisiert. Durch die Einzelbetrachtung jedes Fahrzeugs können mehrere resultierende Fahrmanöver für ein Szenario erzeugt werden. In Abbildung 6 hat das vordere Fahrzeug 2 nach Auswertung der Regelsätze nur die Möglichkeit, dem Fahrstreifen weiter zu folgen (grüner Pfeil), da das Rechtsfahrgebot in den Manövern berücksichtigt wird. Fahrzeug 1 hingegen hat die Möglichkeiten, Fahrzeug 2 zu folgen (grüner Pfeil), sich Fahrzeug 2 anzunähern

(violetter Pfeil) und einen Fahrstreifenwechsel nach links durchzuführen (roter Pfeil). Die Manöver der Fahrzeuge sind ebenfalls durch einen Reasoner automatisch geschlussfolgert.

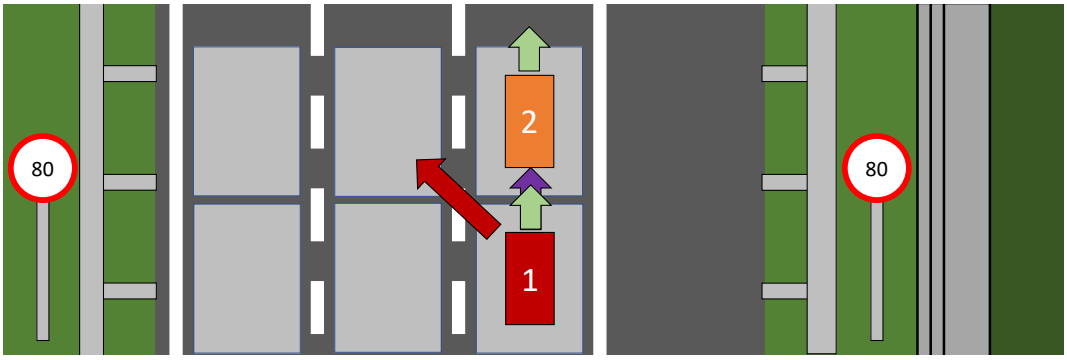


Abbildung 6: Mögliche Fahrmanöver in überladener Startszene. Grün: Folgen, violett: Annähern, rot: Fahrstreifenwechsel.

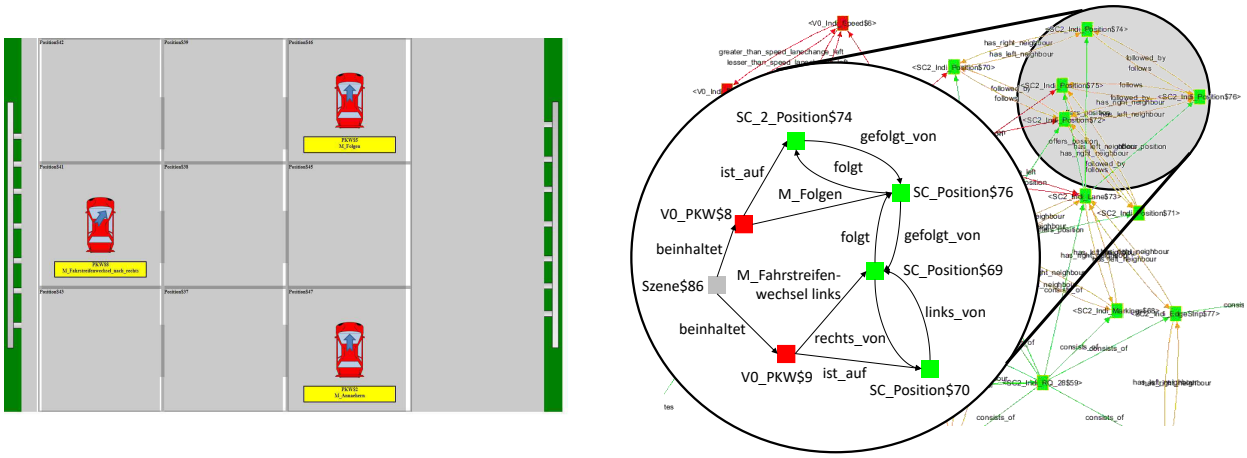
Szenarienentwicklung Aus der *überladenen* Startszene können im nächsten Prozessschritt mögliche Endszenen abgeleitet werden, wodurch nach der Nomenklatur von Ulbrich et al. [4] Szenarien entstehen. Aus den überladenen Startszenen werden dazu erst alle Variationen von einzelnen Manövern der Verkehrsteilnehmer ausgewählt, woraus eine Vielzahl von potenziellen *Startszenen* entsteht. Im Beispiel aus Abbildung 6 ergeben sich drei Startszenen bei denen Fahrzeug 2 immer das Manöver *Folgen* zugeordnet wird und Fahrzeug 1 jeweils eins der aufgezählten Manöver ausführt. Für die Inferenz zu einer Endszene müssen den Verkehrsteilnehmern aufgrund ihrer aktuellen Manöver Relativgeschwindigkeiten zugeordnet werden. Fahrzeug 2 bekommt im Beispiel die Geschwindigkeit v_2 zugeordnet. Aus den Manövern von Fahrzeug 1 ergeben sich dann Bedingungen für die Relation von v_1 zu v_2 wenn Fahrzeug 2 dem Fahrstreifen folgt:

- Fahrzeug 1 folgt Fahrzeug 2: $v_1 = v_2$
- Fahrzeug 1 nähert sich Fahrzeug 2 an: $v_1 > v_2$
- Fahrzeug 1 leitet Fahrstreifenwechsel nach links ein: $v_1 > v_2$

Auf Grundlage der Relativgeschwindigkeiten kann die zugehörige *Endszene* ermittelt werden. Das langsamste Fahrzeug im Szenario wird dazu eine logische Position in Fahrtrichtung weiter gesetzt und alle anderen Fahrzeug aufsteigend ihrer Geschwindigkeiten mehrere Positionen. Die virtuellen Pfade der Verkehrsteilnehmer über das Positionsgitter werden auf Kollisionen überprüft und bei Vorhandensein einer Kollision als nicht valide Endszene markiert.

Abschließend liefert die Szenariengenerierung umfassend sprachliche beschriebene Szenarien mit einer Start- und einer Endszene sowie Relationen der Geschwindigkeiten der Verkehrsteilnehmer ohne dabei konkrete Werte festzulegen. Die Parametrierung von Wertebereichen und einzelnen Repräsentanten erfolgt im Konzept nach Bagschik et al. [5] bei der Überführung zu logischen und konkreten Szenarien. Ein Beitrag dazu findet sich in diesem Tagungsband mit dem Titel „Detaillierung einer stichwortbasierten Szenariobeschreibung für die Durchführung in der Simulation am Beispiel von Szenarien auf deutschen Autobahnen“ von Menzel et al. [21].

Export der Szenarien Wie eingangs motiviert sollen die Szenarien in eine Menschen-lesbare und eine technische Darstellung exportiert werden. Für die Menschen-lesbare Darstellung ist eine abstrahierte HTML-basierte Visualisierung umgesetzt (Abbildung 7a). Die Anordnung der Bestandteile des Szenarios werden in ein HTML-Raster überführt und jeweils eine Grafik für eine Klasse der Ontologie angezeigt. Die Grafiken sind für jede Klasse einzeln erstellt worden (Fahrstreifen, Fahrzeuge und Rückhaltesysteme) oder aus vorhandenen Katalogen (Verkehrszeichen) übernommen. Verkehrsteilnehmer werden je nach Manöver mit einem Richtungspfeil markiert und alle Elemente sind mit der eindeutigen Kennung annotiert. Die technische Darstellung ist mithilfe eines vollständig ver-



(a) HTML-basierte Visualisierung für abstrahierte Übersicht

(b) Szenariengraph als detaillierte sprachliche Beschreibung

Abbildung 7: Visualisierung der funktionalen Szenarien

knüpften *Szenariengraph* (Abbildung 7b) dargestellt. Jeder Knoten des Graphen stelle eine Instanz mit eindeutiger Kennung dar. Die Verknüpfungen zwischen den Knoten zeigen die erstellten und geschlussfolgerten Objektbeziehungen. In dieser Darstellung ist ein Szenario über die Start- und die Endszene auf dem größten Detaillevel beschrieben und aufgrund der zahlenmäßigen Komplexität für den Menschen zwar verständlich jedoch nicht intuitiv zu verarbeiten.

4 Fazit und Ausblick

Mit dem hier vorgestellten Ansatz ist es möglich, funktionale Szenarien nach Bagschik et al. [5] automatisiert in großer Vielfalt systematisch zu erstellen. In einer ersten Studie wurden 12.475 Szenarien mit jeweils vier Verkehrsteilnehmern und vier Positionen pro Fahrstreifen auf mehreren Regelquerschnitten erzeugt. Diese Zahl ließe sich durch Variation der Infrastruktur und einer größeren Anzahl von Verkehrsteilnehmern fast beliebig erhöhen. Aus der Studie wurden 700 Szenarien einzeln auf Richtigkeit und Vollständigkeit gegenüber dem modellierten Wissen verifiziert. Fokus der Auswertung liegt einerseits auf Szenarien mit komplexer Infrastruktur und andererseits auf Szenarien mit komplexen Interaktionen der Verkehrsteilnehmer.

Die Vorteile der Szenariengenerierung sind die automatische Erstellung und der nachvollziehbare Inhalt der Szenarien. So können die etwa 300 modellierten Klassen effizienter

auf fehlende Aspekte untersucht werden als der resultierende Szenarienkatalog. Der Fokus der aktuell modellierten Wissensbasis liegt auf *Normalszenarien* auf deutschen Autobahnen. Im Gegensatz zu potenziell kritischen Szenarien werden sich überschneidende Pfade von Fahrmanövern bei der Szenarienentwicklung und nicht zu erwartende Aktionen (plötzliche Fahrstreifenwechsel bei Rechtsfahrgebot) nicht generiert. Solche Fälle können jedoch durch Anpassung der Regeln der Ontologie nachvollziehbar hinzugefügt werden. Kritische Szenarien, in denen geringe Abstände und hohe Relativgeschwindigkeiten von Interesse sind, können im Rahmen der Parametrierung der Fahrmanöver bei der Konvertierung zu logischen und konkreten Szenarien erzeugt werden [21].

Einschränkungen des Ansatzes ist, dass für die Modellierung des Hintergrundwissens und den systematischen Erstellungsprozess Vereinfachungen und Abstraktionen notwendig sind. So werden bei der Generierung von Verkehrsteilnehmern beispielsweise keine Fahrzeugdimensionen berücksichtigt, sodass zwei Pkw neben einem Lkw nicht in der logischen Positionierung erstellt werden können.

In Zusammenhang mit den Arbeiten von Menzel et al. kann der hier vorgestellte Ansatz einen Beitrag zur Entwicklung und zum szenarienbasierten Testansatz von automatisierten Fahrzeugen leisten. Die Verwendung von quantitativen Szenarienkatalogen als Grundlage für Testfälle ist aktuell Fokus der weiteren Arbeiten am Institut für Regelungstechnik.

Literatur

[1] W. Wachenfeld und H. Winner. “Die Freigabe des autonomen Fahrens”. In: *Autonomes Fahren*. Berlin, Heidelberg: Springer Vieweg, 2015, S. 439–464.

[2] F. Schuldt. “Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen”. Diss. Braunschweig: Technische Universität Braunschweig, 2017.

[3] F. Schuldt, F. Saust, B. Lichte, M. Maurer und S. Scholz. “Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen”. In: *Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel (AAET)*. Braunschweig, Deutschland, 2013.

[4] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt und M. Maurer. “Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren”. In: *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*. Bd. 10. Walting im Altmühltal, Deutschland: Uni-DAS e. V, 2015, S. 105–117.

[5] G. Bagschik, T. Menzel, A. Reschka und M. Maurer. “Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen”. In: *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*. Bd. 11. Walting im Altmühltal, Deutschland: Uni-DAS e.V., 2017, S. 125–135.

[6] B. Neumann und H.-J. Novak. “NAOS: Ein System zur natürlichsprachlichen Beschreibung zeitveränderlicher Szenen”. In: *Informatik Forschung und Entwicklung* 1 (1986), S. 83–92.

[7] G. Herzog, C.-K. Sung, E. André, W. Enkelmann, H.-H. Nagel, T. Rist, W. Wahlster und G. Zimmermann. "Incremental Natural Language Description of Dynamic Imagery". In: *Wissensbasierte Systeme* 227 (1989), S. 153.

[8] H.-H. Nagel und W. Enkelmann. "Generic road traffic situations and driver support systems". In: *Proceedings of the 5th PROMETHEUS Workshop*. München, Deutschland, 1991.

[9] M. Arens, R. Gerber und H.-H. Nagel. "Conceptual Representations between Video Signals and Natural Language Descriptions". In: *Image and Vision Computing* 26.1 (2008), S. 53–66.

[10] E. D. Dickmans. "Subject-Object Discrimination in 4D Dynamic Scene Interpretation for Machine Vision". In: *Workshop on Visual Motion 1989 Proceedings*. 1989, S. 298–304.

[11] D. Dickmanns. "Knowledge Based Real-Time Vision". In: *IFAC Proceedings Volumes* 28.11 (1995), S. 13–18.

[12] A. D. Lattner, J. D. Gehrke, I. J. Timm und O. Herzog. "A Knowledge-Based Approach to Behavior Decision in Intelligent Vehicles". In: *2005 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NEV, USA, 2005, S. 466–471.

[13] B. Hummel, W. Thiemann und I. Lulcheva. "Scene Understanding of Urban Road Intersections with Description Logic". In: *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.

[14] A. Armand, D. Filliat und J. Ibañez-Guzman. "Ontology-Based Context Awareness for Driving Assistance Systems". In: *2014 IEEE Intelligent Vehicles Symposium (IV)*. Dearborn, MI, USA, 2014, S. 227–233.

[15] L. Zhao, R. Ichise, Y. Sasaki, Z. Liu und T. Yoshikawa. "Fast Decision Making Using Ontology-Based Knowledge Base". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. Götheborg, Schweden, 2016, S. 173–178.

[16] Forschungsgesellschaft für Straßen- und Verkehrswesen. *Richtlinie für die Anlage von Autobahnen*. 2008.

[17] Bundesministerium für Verkehr. *Richtlinien für die Absicherung von Arbeitsstellen an Straßen*. 1995.

[18] A. Reschka. "Fertigkeiten- und Fähigkeitengraphen als Grundlage des sicheren Betriebs von automatisierten Fahrzeugen im öffentlichen Straßenverkehr in städtischer Umgebung". Diss. Braunschweig: Technische Universität Braunschweig, 2017.

[19] W. Tölle. *Ein Fahrmanöverkonzept für einen maschinellen Kopiloten*. Fortschritt-Berichte VDI. Düsseldorf: VDI-Verlag, 1996.

[20] H.-J. Novak. *Textgenerierung Aus Visuellen Daten: Beschreibungen von Straßenszenen*. Hrsg. von W. Brauer. Bd. 142. Informatik-Fachberichte. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987.

[21] T. Menzel, G. Bagschik, L. Isensee, A. Schomburg und M. Maurer. "Detaillierung einer stichwortbasierten Szenariobeschreibung für die Durchführung in der Simulation am Beispiel von Szenarien auf deutschen Autobahnen". In: *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*. Bd. 12. Walting im Altmühltal, Deutschland: Uni-DAS e.V., 2018.

Detaillierung einer stichwortbasierten Szenariobeschreibung für die Durchführung in der Simulation am Beispiel von Szenarien auf deutschen Autobahnen

T. Menzel, G. Bagschik,* L. Isensee, A. Schomburg,† und M. Maurer‡

Zusammenfassung: Szenarienbasierte Entwicklungs- und Testprozesse stellen einen vielversprechenden Ansatz für die Verifikation und Validierung automatisierter Fahrfunktionen dar und sind somit voraussichtlich im Stande, einen Beitrag zur Freigabe automatisierter Fahrzeuge zu leisten. Um diesen Beitrag dokumentieren und bewerten zu können, müssen Szenarien nachverfolgbar entlang des Entwicklungs- und Testprozesses erstellt werden. In den frühen Entwicklungsphasen liegen die Betriebsszenarien des Entwicklungsgegenstandes üblicherweise in einer abstrakten sprachlichen Beschreibungsform vor. Spätestens im Rahmen einer Testfallableitung für den simulativen Test müssen die sprachlich dokumentierten Betriebsszenarien in eine Parameterraumdarstellung überführt und in die Formate für die jeweilige Simulationsumgebung konvertiert werden. Dieser Schritt bedeutet zur Zeit einen hohen manuellen Aufwand. Zudem ist bei einer manuellen Überführung von sprachlich beschriebenen Szenarien in die Formate für die Simulationsdurchführung durch unterschiedliche Bearbeiter nicht sichergestellt, dass eine einheitliche Interpretation und Umsetzung der Szenarien erfolgt. In diesem Beitrag wird ein Ansatz zur automatisierten Überführung von stichwortbasierten Szenarien in eine Parameterraumdarstellung sowie die Konvertierung in die Formate OpenDRIVE (Beschreibung des Streckennetzes) und OpenSCENARIO (Beschreibung der Verkehrsteilnehmer und Umweltbedingungen) am Beispiel von Szenarien auf deutschen Autobahnen vorgestellt.

Schlüsselwörter: Automatisiertes Fahren, Testprozess, Szenarien, OpenSCENARIO, OpenDRIVE

1 Einleitung

Vor der Einführung automatisierter Fahrzeugführungssysteme muss ein definiertes Sicherheitsniveau der Systeme sichergestellt werden. Bei Fahrerassistenzsystemen erfolgt diese Prüfung nach Stand der Technik über einen streckenbasierten Ansatz. Für Systeme höherer Automatisierungsgrade sagen Wachenfeld und Winner [1] bei einem streckenbasiertenbasierten Ansatz mehrere Milliarden notwendige Testkilometer voraus. Schuldt [2] folgert daraus, dass ein streckenbasierter Ansatz für die Freigabe von Systemen höherer Auto-

*T. Menzel und G. Bagschik sind wissenschaftliche Mitarbeiter am Institut für Regelungstechnik der Technischen Universität Braunschweig (Email: {menzel, bagschik}@ifr.ing.tu-bs.de).

†L. Isensee und A. Schomburg sind Masterstudenten an der Technischen Universität Braunschweig.

‡M. Maurer ist Professor und Institutsleiter an selbigem Institut (Email: maurer@ifr.ing.tu-bs.de).

matisierungsgrade ökonomisch nicht vertretbar ist und schlägt einen szenarienbasierten Ansatz als vielversprechende Alternative vor.

Bei szenarienbasierten Ansätzen stellt die Szenariengenerierung einen sensiblen Schritt für die Sicherheitsargumentation und somit die Freigabe des Systems dar. Die Szenarien müssen daher systematisch und nachverfolgbar entlang des Entwicklungsprozesses generiert und dokumentiert werden. Hierzu stellen Bagschik et al. [3] die drei Abstraktionsgrade der *funktionalen*, *logischen* und *konkreten Szenarien* vor. Diese Abstraktionsgrade unterstützen die Aktivitäten in der Konzept-, Systementwurfs- und Testphase eines V-Modell-basierten Entwicklungsprozesses und bilden so die Grundlage für eine strukturierte Generierung sowie Verwendung von Szenarien.

Zur automatisierten Erstellung von funktionalen Szenarien stellen Bagschik et al. [4] einen wissensbasierten Ansatz vor. Die generierten Szenarien nutzen ein definiertes Vokabular, um die Strecke sowie die Verkehrsteilnehmer sprachlich zu beschreiben. Dabei besteht jedes Szenario aus einer Start- und einer Endszene. Für die Überführung der funktionalen Szenarien zu logischen Szenarien muss die stichwortbasierte Notation mithilfe einer Parameterraumdarstellung detailliert und in die Formate für die Simulationsdurchführung konvertiert werden. Dieser Schritt beinhaltet im Hinblick auf eine Simulationssteuerung zudem die Spezifikation der zeitlichen Abläufe der Fahrzeuginteraktionen zwischen der Start- und der Endszene. Damit die Strecke und die zeitlichen Abläufe der Fahrzeuginteraktionen in den Datenformaten konsistent modelliert werden, müssen verschiedene Beziehungen und Abhängigkeiten zwischen den Szenarioelementen und -parametern berücksichtigt werden.

Die Schritte der Szenariendetaillierung und -konvertierung bedeuten aktuell einen hohen manuellen Zeitaufwand. Zudem sind insbesondere bei unterschiedlichen Bearbeitern keine einheitliche Interpretation der sprachlichen Szenarien und keine konsistente Modellierung in den Formaten für die Simulation sichergestellt. In diesem Beitrag wird daher ein Ansatz zur automatisierten Überführung einer stichwortbasierten Szenariobeschreibung in eine Parameterraumdarstellung sowie die Konvertierung in die Formate für die Simulationsdurchführung am Beispiel von Szenarien auf deutschen Autobahnen vorgestellt. Innerhalb der Parameterraumdarstellung werden Abhängigkeiten und Beziehungen zwischen den Elementen und Parametern des Szenarios modelliert. Dies ist notwendig, um die zeitlichen Abläufe der Fahrzeuginteraktionen in der Simulation entsprechend der Vorgaben der funktionalen Szenarien aufzulösen und eine konsistente Umsetzung der Szenarien in den Datenformaten sicherzustellen. Als Format für die Streckenbeschreibung wird OpenDRIVE verwendet. Als Format für die Beschreibung der Verkehrsteilnehmer und der Umwelt wird OpenSCENARIO genutzt.

2 Stand der Technik und verwandte Arbeiten

Schuldt et al. [5] schlagen einen szenarienbasierten Testansatz sowie eine Strukturierung von Szenarien in einem 4-Ebenen-Modell vor. Ulbrich et al. [6] definieren die Begriffe Szene, Situation und Szenario. Bagschik et al. [3] motivieren unterschiedliche Darstellungsformen von Szenarien entlang des Entwicklungsprozesses am Beispiel der ISO 26262 [7] und schlagen die Abstraktionsgrade funktionale Szenarien, logische Szenarien und konkrete Szenarien vor. Dieser Beitrag verwendet die Begriffe entsprechend der Definitionen von Ulbrich et al. [6] und Bagschik et al. [3].

Waymo LLC [8] stellt einen Testansatz vor, mit dem Ausschnitte durchgeführter Versuchsfahrten in der Simulation reproduziert und Parameter der Verkehrsteilnehmer und der Leitinfrastruktur variiert werden können. Neben den Messdaten der Realfahrten wird für diesen Ansatz eine hochdetaillierte Karte des realen Streckennetzes benötigt. Dieser Ansatz konvertiert somit Messdaten in konkrete Szenarien, die in der Simulation ausführbar sind, und verallgemeinert diese konkreten Szenarien für eine nachträgliche Variation der Parameter zu logischen Szenarien. Eine Generierung synthetischer Strecken und Szenarien auf Basis von formalisiertem Wissen wird nicht beschrieben.

Bach et al. [9] motivieren eine modellbasierte Darstellung von Szenarien. Des Weiteren beschreiben sie ein Konzept und eine prototypische Umsetzung zur abstrakten Modellierung von konkreten Szenarien insbesondere der Handlungsabläufe der Verkehrsteilnehmer. Das Konzept basiert auf der Unterteilung eines Szenarios in unterschiedliche Akte und Events und setzt somit eine ähnliche Strukturierung der Handlungsabläufe wie das Format OpenSCENARIO um. Xiong [10] stellt eine Orchestrierung von Szenarien für die Simulation auf Grundlage einer Wissensbasis vor. Hierzu beschreibt Xiong den Ablauf eines Szenarios in der Simulation mithilfe einer Ontologie. In der Ontologie sind sowohl die Bestandteile der Strecke und die Interaktionen der Verkehrsteilnehmer als auch die einzusetzenden Fahrermodelle detailliert modelliert. Das Szenario wird von einer Ablaufsteuerung in der Simulation umgesetzt und gegebenenfalls zur Laufzeit an das Verhalten der Fahrfunktion angepasst. Dieser Ansatz legt somit explizit ein Fahrzeug als Testobjekt fest und steuert die übrigen Verkehrsteilnehmer mithilfe des in der Ontologie modellierten Wissens, sodass definierte Szenen im Szenario erreicht werden. Nach der Terminologie von Bagschik et al. [3] beschreibt Xiong aufgrund der Definition eines Testobjektes keine Szenarien sondern Testfälle. Bach et al. und Xiong stellen somit Notationen für konkrete Szenarien bzw. Testfälle vor. Die Generierung aller möglichen funktionalen Szenarien in Bezug auf eine Wissensbasis und eine Überführung unterschiedlicher Darstellungsformen von Szenarien innerhalb des Entwicklungs- und Testprozesses werden weder von Bach et al. noch von Xiong näher beschrieben. Beide Arbeiten können aber zu der Modellierung von konkreten Szenarien sowie Testfällen beitragen.

Bagschik et al. [11] stellen ein Konzept für die wissensbasierte Generierung von Startszenen mithilfe einer Ontologie vor. Für die Strukturierung des Wissens innerhalb der Ontologie nutzen Bagschik et al. [11] ein 5-Ebenen-Modell basierend auf dem 4-Ebenen-Modell nach Schuldt et al. [5]. In einer späteren Veröffentlichung erweitern Bagschik et al. [4] das Konzept der wissensbasierten Generierung von Startszenen und stellen eine Umsetzung für die wissensbasierte Generierung von Betriebsszenarien auf deutschen Autobahnen vor. Die Betriebsszenarien enthalten jeweils eine Start- und eine Endszene und werden im Rahmen von funktionalen Szenarien sprachlich auf semantischer Ebene beschrieben und mithilfe der Web Ontology Language (OWL) modelliert. Die in diesem Beitrag vorgestellten Arbeiten nutzen die von Bagschik et al. [4] generierten funktionalen Szenarien als Ausgangsbasis für die Szenariendetaillierung und Überführung in die Formate für die Simulationsdurchführung.

Für den simulativen Test von automatisierten Fahrfunktionen steht ein breites Angebot an Simulationsumgebungen wie ASM Traffic[1] von der dSPACE GmbH, CarMa-

[1] `https://www.dspace.com/de/gmb/home/products/sw/automotive_simulation_models/produkte_asm/asm_traffic.cfm`, abgerufen am 13.04.2018

ker[2] von der IPG Automotive GmbH oder Virtual Test Drive[3] von der VIRES Simulationstechnologie GmbH zur Verfügung. Die verschiedenen Simulationsumgebungen nutzen unterschiedliche Formate für die Spezifikation von Szenarien. Das in diesem Beitrag beschriebene Werkzeug überführt funktionale Szenarien in die Formate für die simulative Durchführung mithilfe der Simulationsumgebung Virtual Test Drive. Virtual Test Drive fordert für die Beschreibung des Straßennetzes das Format OpenDRIVE[4]. Für die Beschreibung der Interaktionen der Verkehrsteilnehmer sowie der Umweltbedingungen nutzt Virtual Test Drive zukünftig das Format OpenSCENARIO[5].

3 Detaillierung einer stichwortbasierten Szenariobeschreibung für die Durchführung in der Simulation

Die Überführung einer stichwortbasierten Szenariobeschreibung in die Formate für die Simulationsdurchführung erfolgt, wie in Abbildung 1 dargestellt, in zwei Schritten. Die funktionalen Szenarien liegen als semantische, stichwortbasierte Beschreibungen vor und sind mithilfe der Web Ontology Language modelliert. Die Generierung der funktionalen Szenarien wird von Bagschik et al. [4] beschrieben.

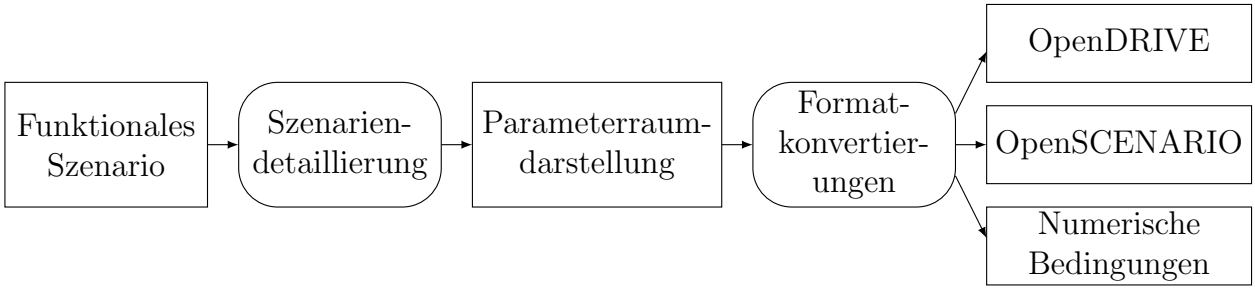


Abbildung 1: Prozessübersicht der Szenarienüberführung. Rechtecke stellen Arbeitsergebnisse, abgerundete Ecken stellen Prozessschritte dar.

Im Rahmen der Szenariendetaillierung werden die funktionalen Szenarien eingelesen und in eine Parameterraumdarstellung überführt. Hierzu werden jeder Vokabel der funktionalen Szenarien Parameter zugeordnet, die für die Definition des jeweiligen Elementes in den Formaten für die Simulationsdurchführung gefordert werden. Zusätzlich werden Abhängigkeiten der Parameter sowie einzuhaltende Rahmenbedingungen für die Wahl und Kombination von Parameterwerten modelliert. Diese Abhängigkeiten und Rahmenbedingungen ergeben sich zum einen aus physikalischen Zusammenhängen der Parameter und zum anderen auf Basis der in den funktionalen Szenarien spezifizierten Interaktionen der Verkehrsteilnehmer sowie den Anforderungen durch die Formate für die Simulationsdurchführung.

[2] `https://ipg-automotive.com/products-services/simulation-software/carmaker/`, abgerufen am 13.04.2018

[3] `https://vires.com/vtd-vires-virtual-test-drive/`, abgerufen am 13.04.2018

[4] `http://www.opendrive.org/`, abgerufen am 13.04.2018

[5] `http://www.openscenario.org/`, abgerufen am 13.04.2018

Im Rahmen der Formatkonvertierungen werden die für die einzelnen Formate benötigten Informationen aus der Parameterraumdarstellung extrahiert und in die Struktur des jeweiligen Formats überführt. Zudem werden die bei der Wahl der Parameterwerte einzuhaltenden Bedingungen in einer zusätzlichen Datei als Regelwerk dokumentiert, um in späteren Prozessschritten (beispielsweise im Rahmen einer Testfallerstellung) nur gültige Wertekombinationen zu erzeugen. Anschließend wird für jeden Parameter ein Standardwert entsprechend des dokumentierten Regelwerks gewählt. Die extrahierten Szenarien spezifizieren somit die variierbaren Parameter mithilfe der Formate OpenDRIVE und OpenSCENARIO, formalisieren ihre Bedeutung für die Simulationsdurchführung und geben einen entsprechend eines dokumentierten Regelwerkes gültigen Standardparameter an. Damit stellen die extrahierten Szenarien die Grundlage für logische Szenarien nach der Definition von Bagschik et al. [3] bereit. Die Generierung einer repräsentativen Stichprobe konkreter Szenarien für jedes logische Szenario entsprechend eines definierten Testkonzeptes ist nicht Teil dieses Beitrags.

3.1 Szenariendetaillierung

Als Ausgangsbasis für die Szenariendetaillierung wird eine semantische Beschreibung von Szenarien verwendet, die im OWL-Format vorliegt. Diese funktionalen Szenarien nutzen ein definiertes Vokabular sowie semantische Beziehungen zwischen den Begriffen des Vokabulars zur Beschreibung der Straßenebene, der Straßenausstattung, der beweglichen Objekte und der Umweltbedingungen. Das Vokabular der Straßenebene und Straßenausstattung wurde von Bagschik et al. [4] in Anlehnung an die Richtlinien zur Anlage von Autobahnen [12] erstellt. Für die Beschreibung der Interaktionen der Verkehrsteilnehmer verwenden Bagschik et al. [4] die Manöver nach Reschka [13] basierend auf Tölle [14] und Nagel und Enkelmann [15].

Im ersten Schritt der Szenariendetaillierung werden, wie in Abbildung 2 dargestellt, die zur Beschreibung des funktionalen Szenarios genutzten Vokabeln in das 5-Ebenen-Modell [11] eingeordnet und um Parameter, die die einzelnen Elemente spezifizieren, erweitert. Die zur Beschreibung der einzelnen Elemente notwendigen Parameter sind von den Formaten für die Simulationsdurchführung abhängig. Im Rahmen der in diesem Beitrag vorgestellten Arbeiten wurden die Parameter daher aus den Spezifikationen der Simulationsformate OpenDRIVE [16] und OpenSCENARIO [17] abgeleitet. Das Format OpenDRIVE beschreibt das Straßennetz und somit die Elemente der ersten bis dritten Ebene des 5-Ebenen-Modells. Das Format OpenSCENARIO beschreibt die Interaktionen der Verkehrsteilnehmer sowie die Umweltbedingungen und somit die vierte und fünfte Ebene des 5-Ebenen-Modells.

Die Bewegungen der Fahrzeuge werden in OpenSCENARIO mithilfe von Aktionen (Manöver und weitere Befehle zur Simulationssteuerung) und Events (Ereignisse, bei denen eine Aktion ausgelöst wird) beschrieben. In den funktionalen Szenarien werden die Bewegungen eines Fahrzeuges durch seine relative Position zu den übrigen Verkehrsteilnehmern in der Start- und Endszene sowie das auszuführende Manöver beschrieben. Damit die in den funktionalen Szenarien definierten Manöver mithilfe von Parametern aus OpenSCENARIO darstellbar sind, müssen die Interaktionen der Verkehrsteilnehmer, wie in Abbildung 3 dargestellt, zunächst in eine eventbasierte Darstellung überführt werden. Hierzu werden die Manöver der funktionalen Szenarien durch zusätzliche Beschreibungselemente, wie Aktionen und Events, detailliert.

Abbildung 2: 5-Ebenen-Modell zur Strukturierung von Szenarien nach Bagschik et al. [11] auf Basis von Schuldt [2]. Die blauen Kästen symbolisieren beispielhaft Vokabeln aus den funktionalen Szenarien. Die gelben Kästen symbolisieren beispielhaft aus den Formatspezifikationen abgeleitete Parameter.

Im zweiten Schritt der Szenariendetaillierung werden die Beziehungen und Abhängigkeiten zwischen den Elementen und Parametern der Elemente modelliert. Hierbei wurden, wie in Abbildung 4 dargestellt, drei Arten von Beziehungen identifiziert und modelliert: Anordnungsbeziehungen, Objektabhängigkeiten und Parameterabhängigkeiten. Die drei Beziehungsebenen sind unabhängig voneinander und haben unterschiedlichen Aussagencharakter.

Die Anordnungsbeziehungen charakterisieren die topologischen Lagebeziehungen der Objekte zueinander. Hierzu werden alle im Szenario modellierten Objekte als Knoten in einen Graphen eingetragen. Benachbarte oder im Szenario miteinander interagierende Objekte werden über die Kanten des Graphen in Relation gesetzt. Die Anordnungsbeziehungen beinhalten unter anderem die Zuordnung von Fahrstreifen zu Streckenabschnitten, die Anordnung von Fahrstreifen innerhalb eines Streckenabschnittes, die Positionen der Verkehrsteilnehmer auf der Strecke sowie die Lagebeziehungen der Verkehrsteilnehmer zueinander. Die Anordnungsbeziehungen werden mithilfe von Vokabeln bereits in den funktionalen Szenarien beschrieben und können somit direkt modelliert werden.

Die Objektabhängigkeiten beschreiben die Abhängigkeiten zwischen den Objekten des Szenarios. Jede Objektabhängigkeit umfasst einen Regelsatz, der einzuhaltende Bedingungen zwischen den Parametern der verknüpften Objekte beschreibt. Diese Bedingungen sind notwendig, um Vorgaben aus den funktionalen Szenarien oder Anforderungen durch die Formate zu berücksichtigen. Beispielsweise werden in den funktionalen Szenarien nach Bagschik et al. [4] die Interaktionen der Verkehrsteilnehmer auf semantischer Ebene mithilfe von Manövern beschrieben. Aus dem Manöver *Annähern* resultiert auf Pa-

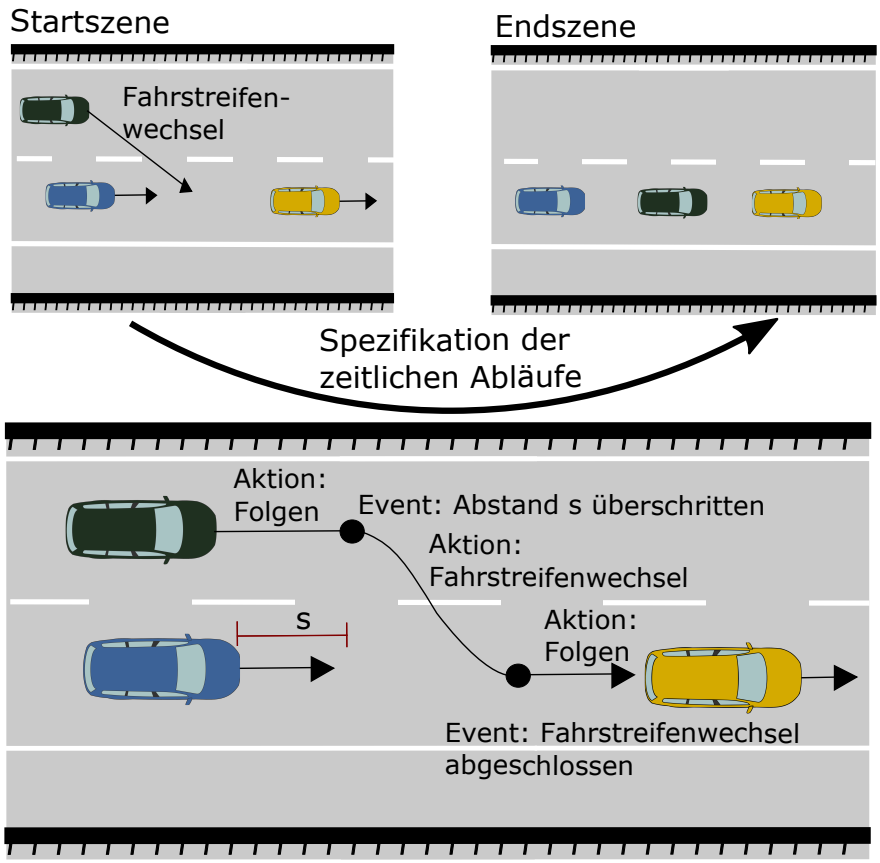


Abbildung 3: Spezifikation der zeitlichen Abläufe mithilfe einer eventbasierten Darstellung

rameterebene die Bedingung, dass die Geschwindigkeit des sich annähernden Fahrzeuges größer sein muss als die Geschwindigkeit des vorausfahrenden Fahrzeuges. In OpenDRIVE werden verschiedene Fahrstreifen mithilfe von Streckenabschnitten gegliedert und zu einem Straßennetz zusammengesetzt. Aus diesem Aufbau resultiert die Bedingung, dass aufeinanderfolgenden Fahrstreifen an den Schnittstellen die gleiche Fahrstreifenbreite zugeordnet sein muss.

Die Parameterabhängigkeiten beschreiben die Abhängigkeiten zwischen den Parametern eines Objekts. Dabei können auf einen Parameter ein oder mehrere andere Parameter desselben Objektes Einfluss nehmen. Die Parameterabhängigkeiten berücksichtigen Vorgaben bei der Wahl von Parameterwerten durch Normen und Richtlinien oder physikalische Zusammenhänge. Beispielsweise wird in den funktionalen Szenarien nach Bagschik et al. [4] das Höhenprofil der Strecke mithilfe von Entwurfselementen aus der Richtlinie zur Anlage von Autobahnen [12], wie planarer Höhenplan, Kuppe oder Wanne, beschrieben. Die Länge einer Kuppe darf dabei nicht unabhängig von der Anfangs- und Endneigung gewählt werden, sondern muss, wie in Abbildung 4 dargestellt, auf Basis einer Formel aus der Richtlinie zur Anlage von Autobahnen berechnet werden. Mithilfe der Parameterabhängigkeiten werden die Parameterwerte einzelner Objekte somit konsistent im Sinne regulatorischer Vorgaben oder physikalischer Zusammenhänge gewählt.

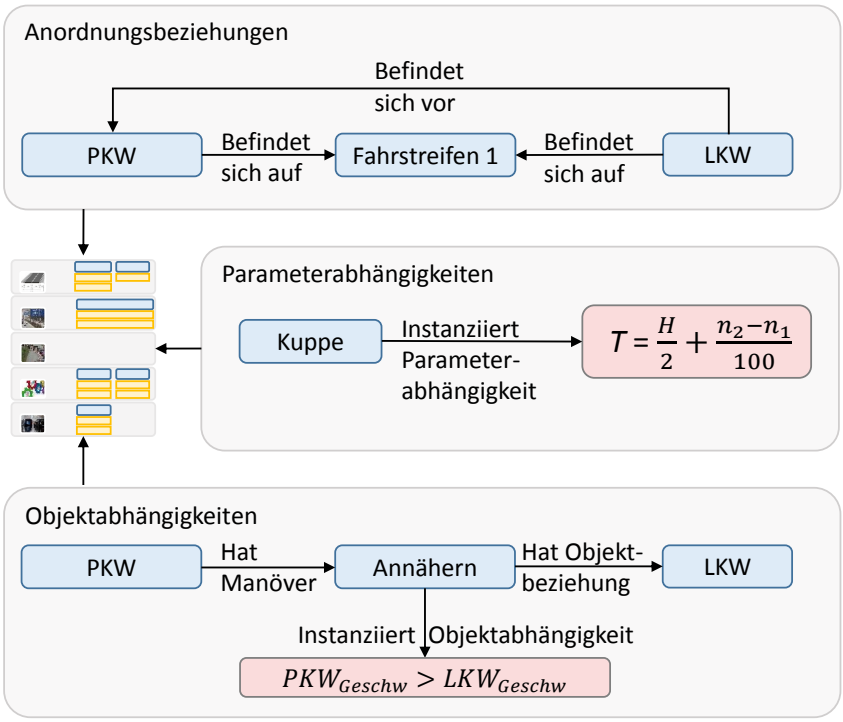


Abbildung 4: Erweiterung des Parameterraumes um Anordnungsbeziehungen sowie Objekt- und Parameterabhängigkeiten. Die Länge T einer Kuppe wird mithilfe des Halbmessers H sowie der Anfangsneigung n_1 und Endneigung n_2 berechnet.

3.2 Export der Formate

Im Rahmen der Formatkonvertierungen werden die in der Parameterraumdarstellung modellierten Informationen in die Formate OpenDRIVE und OpenSCENARIO überführt.

Im ersten Schritt erfolgt die Erstellung der Straßenbeschreibung im OpenDRIVE-Format. Hierzu werden die Elemente der ersten bis dritten Ebene des 5-Ebenen-Modells in die Syntax des Formats OpenDRIVE übersetzt. Dabei werden den Parametern gültige Werte zugewiesen, die auf Basis der modellierten Abhängigkeiten und Bedingungen berechnet werden.

Im zweiten Schritt erfolgt die Erstellung der Verkehrsteilnehmer sowie der Umweltbedingungen im OpenSCENARIO-Format. Hierzu werden die Elemente der vierten und fünften Ebene des 5-Ebenen-Modells in die Syntax des Formats OpenSCENARIO überführt und mit gültigen Parameterwerten initialisiert. Die im vorherigen Schritt erstellte OpenDRIVE-Datei wird dabei als Referenz für die Straßenbeschreibung angegeben. Weiterhin wird das Regelwerk, das für die Ableitung der Standardwerte der Parameter herangezogen wurde, in einer zusätzlichen Datei dokumentiert und so für eine Parametervariation in nachfolgenden Prozessschritten zugänglich gemacht.

4 Evaluierung

In diesem Kapitel erfolgt eine Evaluierung der erzeugten OpenDRIVE- und OpenSCENARIO-Dateien. Hierzu werden funktionale Szenarien mit dem von Bagschik et al. [4] beschriebenen wissensbasierten Ansatz generiert und mit dem in diesem Beitrag vorge-

stellten Ansatz in eine Parameterraumdarstellung überführt sowie in die Formate OpenDRIVE und OpenSCENARIO konvertiert. Die Auswahl der Testszenarien und die Evaluierung erfolgen für beide Formate separat.

4.1 OpenDRIVE

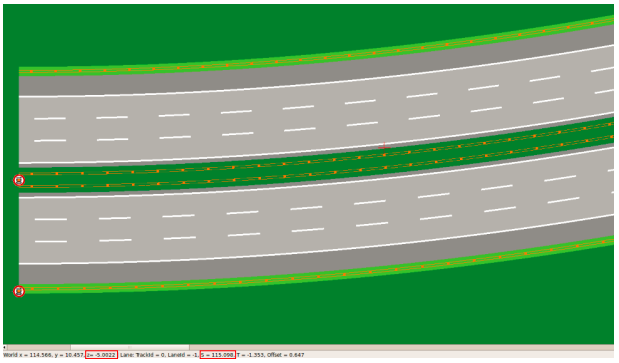
Das Vokabular der funktionalen Szenarien für die Streckenbeschreibung umfasst mehrere Ausprägungen von Höhenplänen (wie Anstieg und Gefälle), Lageplänen (wie Gerade oder Kreisbogen) und Fahrstreifentypen (wie Seitenstreifen oder Böschung). Des Weiteren werden unterschiedliche Elemente der Leitinfrastruktur (wie Verkehrszeichen und Rückhaltesysteme) beschrieben. Durch die Kombination dieser Elemente werden viele unterschiedliche Ausprägungen von Streckenabschnitte ermöglicht. In der aktuellen Umsetzung der wissensbasierten Szenariengenerierung nach Bagschik et al. [4] besteht die Straßenebene jedes funktionalen Szenarios aus nur einem Streckenabschnitt. Eine Kombination aufeinanderfolgender Abschnitte zu längeren Strecken ist in der aktuellen Implementierung nicht umgesetzt und begrenzt somit den Umfang der Eingangsdaten für die Evaluierung.

Innerhalb des OpenDRIVE-Formates existieren ausschließlich Abhängigkeiten zwischen aufeinanderfolgenden Streckenabschnitten und Fahrstreifen. Die Entwurfselemente eines Streckenabschnittes, wie der Lageplan, der Höhenplan und der Fahrstreifenbereich, können unabhängig voneinander spezifiziert werden. Da die generierten Strecken der funktionalen Szenarien nur einen Streckenabschnitt beinhalten, müssen nicht alle Kombinationsmöglichkeiten der Vokabeln getestet werden. Für die Evaluierung muss lediglich geprüft werden, ob jede Vokabel im einzelnen korrekt (entsprechend der im OpenDRIVE-Format vorgesehenen Klasse und Attribute) auf das OpenDRIVE-Format abgebildet wird. Für die Evaluierung wurden somit insgesamt fünf funktionale Szenarien ausgewählt, die das Vokabular für die Streckenbeschreibung vollständig nutzen.

Die Bewertung der OpenDRIVE-Dateien erfolgte in zwei Schritten. Zunächst wurden die Strecken im OpenDRIVE-Viewer (siehe Abbildung 5a) auf korrekte Umsetzung entsprechend der Vorgaben aus den funktionalen Szenarien und normgerechter Parametrierung untersucht. Hierbei wurde über den Koordinatenverlauf der Strecke beispielsweise die Geometrie (Längs- und Höhenverlauf) bewertet. Anschließend wurden die Strecken in Virtual Test Drive (siehe Abbildung 5b) geladen und geprüft, ob alle 3D-Grafiken angezeigt werden und die Fahrstreifen korrekt miteinander verknüpft sind. Hierzu wurde ein virtuelles Fahrzeug definiert, dass die jeweilige Strecke komplett durchfahren musste. Mit diesem Vorgehen konnte gezeigt werden, dass die Straßenbeschreibung entsprechend der Vorgaben des jeweiligen funktionalen Szenarios sowie der geltenden Normen spezifiziert und korrekt in das Datenformat OpenDRIVE konvertiert wird.

4.2 OpenSCENARIO

Das Vokabular der funktionalen Szenarien für die Beschreibung der Verkehrsteilnehmer umfasst verschiedene Fahrzeugklassen (wie PKW oder LKW) und Fahrmanöver (wie Fahrstreifenwechsel oder Annähern). Die Positionen der Fahrzeuge werden über ein fahrstreifenrelatives Positionsgitter miteinander verknüpft. Des Weiteren werden unterschiedliche Umwelteigenschaften mithilfe von verschiedenen Wetterbedingungen (wie Regen oder klarer Himmel) und verschiedenen Tageszeiten (wie Morgens oder Mittags) beschrieben.



(a) OpenDRIVE-Strecke im OpenDRIVE-Viewer

(b) OpenDRIVE-Strecke in Virtual Test Drive

Abbildung 5: Darstellung einer dreistreifigen Autobahn in einer Kurve mit Gefälle, zulässiger Höchstgeschwindigkeit und nachgiebigem Rückhaltesystem im OpenDrive-Viewer (a) und in Virtual Test Drive (b)

Die Bewertung der OpenSCENARIO-Dateien erfolgt in zwei Schritten. Im ersten Schritt erfolgt ein statischer Test mithilfe einer Schema-Datei. Hierbei wird die Struktur der erstellen OpenSCENARIO-Datei mit den Vorgaben des Formates verglichen. Auf diese Weise können fehlende Attribute oder Elemente identifiziert werden. Für die Überprüfung der OpenSCENARIO-Dateien mit der Schema-Datei wurden sechs Szenarien ausgewählt, die zusammen alle Entwurfselemente von OpenSCENARIO beinhalten, die für die Umsetzung des Vokabulars benötigt werden. Durch die Überprüfung dieser Szenarien mithilfe der Schema-Datei wurde gezeigt, dass die funktionalen Szenarien auf Grundlage des definierten Vokabulars in OpenSCENARIO-Dateien konvertiert werden, die den Formatvorgaben entsprechen. Dieser statische Test erlaubt allerdings noch keine Aussage, ob alle Elemente der funktionalen Szenarien umgesetzt und mit validen Parameterwerten initialisiert werden.

Im zweiten Schritt erfolgt daher eine Ausführung der OpenSCENARIO-Dateien in der Simulation. Hierbei wird geprüft, ob die mithilfe von OpenSCENARIO modellierten Szenarien in der Simulation so ablaufen, wie in den funktionalen Szenarien spezifiziert. Zentrale Fragen während der Tests sind:

- Starten alle Fahrzeuge auf der in der Startszene definierten Position?
- Entspricht das Verhalten aller Fahrzeuge den in der Startszene definierten Manövern?
- Starten und enden die Manöver zum richtigen Zeitpunkt, sodass keine Kollisionen verursacht werden?
- Befinden sich alle Fahrzeuge nach den Manöverausführungen auf den in der Endszene definierten Positionen?

Da nicht alle erstellten funktionalen Szenarien manuell geprüft werden können, wurden basierend auf den Erfahrungen der Autoren relevante Testfälle ausgewählt. Hierzu wurden neun Testziele definiert. Jedes Testziel beschreibt einen Programmteil der Szenariendetaillierung und -konvertierung (beispielsweise die Berechnung der Startgeschwindigkeiten der Verkehrsteilnehmer), der explizit geprüft werden soll. Zusätzlich definiert jedes Testziel eine Testskizze, in der der grobe Aufbau und Ablauf der Szenarien beschrieben wird, die zur Prüfung des Testzieles eingesetzt werden können. Insgesamt wurden 30 Szenarien entsprechend der Testskizzen ausgewählt und korrekt (wie in den funktionalen Szenarien spezifiziert) in der Simulation mit Virtual Test Drive ausgeführt.

5 Fazit und Ausblick

In diesem Beitrag wurde ein Ansatz vorgestellt, mit dem eine stichwortbasierte Szenarienbeschreibung in die Formate für die Simulation überführt wird. Hierzu werden die stichwortbasierten Szenarien zunächst durch eine Parameterraumdarstellung detailliert und anschließend in die Formate OpenDRIVE und OpenSCENARIO konvertiert. Die automatisiert erstellten Szenarien wurden evaluiert und laufen den Vorgaben der funktionalen Szenarien entsprechend in der Simulation ab.

Aufbauend auf den vorgestellten Ansätzen sind weitere Arbeiten notwendig, um Szenarien entlang der von Bagschik et al. [3] beschriebenen Abstraktionsgrade zu generieren. Sowohl in den funktionalen Szenarien als auch in den erstellten Formaten für die Simulation werden aktuell nur Fahrzeuge beschrieben, die durch die Simulation gesteuert werden. Die Definition eines Fahrzeuges, dass durch eine automatisierte Fahrfunktion geregelt wird, muss im Rahmen einer Testfallerstellung zukünftig ergänzt werden. Der durch das undefinierte Verhalten der Automation im Szenario entstehende Freiheitgrad könnte zu ungewolltem Verhalten im Szenarioablauf führen. Der Übergang zu Testfällen sowie die Anwendung der erstellen Formate im Rahmen einer Testfallbeschreibung müssen daher in nachfolgenden Arbeiten untersucht werden.

Literatur

[1] W. Wachenfeld und H. Winner. “Die Freigabe des autonomen Fahrens”. In: *Autonomes Fahren.* Hrsg. von M. Maurer, J. C. Gerdes, B. Lenz und H. Winner. Springer Vieweg, Berlin, Heidelberg, 2015, S. 439–464.

[2] F. Schuldt. “Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen”. Diss. Braunschweig: Technische Universität Braunschweig, 2017.

[3] G. Bagschik, T. Menzel, A. Reschka und M. Maurer. “Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen”. In: *Workshop Fahrerassistenzsysteme und automatisiertes Fahren.* Bd. 11. Walting im Altmühltal: Uni-DAS e.V., 2017, S. 125 –135.

[4] G. Bagschik, T. Menzel, C. Körner und M. Maurer. “Wissensbasierte Szenariengenerierung für Betriebsszenarien auf deutschen Autobahnen”. In: *Workshop Fahrerassistenzsysteme und automatisiertes Fahren.* Bd. 12. Walting im Altmühltal: Uni-DAS e.V., 2018. angenommen zur Veröffentlichung.

[5] F. Schuldt, F. Saust, B. Lichte und M. Maurer. “Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen”. In: *AAET - Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel.* Braunschweig, 2013, S. 114 –134.

[6] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt und M. Maurer. “Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren”. In: *Workshop Fahrerassistenzsysteme und automatisiertes Fahren.* Bd. 10. Walting im Altmühltal: Uni-DAS e.V., 2015, S. 105 –117.

[7] International Organization for Standardization. *ISO 26262: Road vehicles – Functional Safety*. 2011.

[8] Waymo LLC. *Waymo Safety Report*. 2017. URL: `https://waymo.com/safetyreport/` (abgerufen am 13. 04. 2018).

[9] J. Bach, S. Otten und E. Sax. "Model based scenario specification for development and test of automated driving functions". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, S. 1149–1155.

[10] Z. Xiong. "Creating a Computing Environment in a Driving Simulator to Orchestrate Scenarios with Autonomous Vehicles". Diss. Leeds: University of Leeds, 2013.

[11] G. Bagschik, T. Menzel und M. Maurer. "Ontology based Scene Creation for the Development of Automated Vehicles". In: *2018 IEEE Intelligent Vehicle Symposium (IV)*. Changshu, 2018.

[12] *Richtlinie für die Anlage von Autobahnen*. Forschungsgesellschaft für Straßen und Verkehrswesen, 2009.

[13] A. Reschka. "Fertigkeiten- und Fähigkeitengraphen als Grundlage für den sicheren Betrieb von automatisierten Fahrzeugen in städtischer Umgebung". Diss. Braunschweig: Technische Universität Braunschweig, 2017.

[14] W. Tölle. *Ein Fahrmanöverkonzept für einen maschinellen Kopiloten*. Fortschritt-Berichte VDI. Düsseldorf: VDI-Verlag, 1996.

[15] H.-H. Nagel und W. Enkelmann. "Generic road traffic situations and driver support systems". In: *Proceedings of the 5th PROMETHEUS Workshop*. München, Germany, 1991.

[16] VIRES Simulationstechnologie GmbH. *OpenDRIVE V 1.4 Format Specification, Revision H*. 2015. URL: `http://www.opendrive.org/docs/OpenDRIVEFormatSpecRev1 4H.pdf` (abgerufen am 14. 04. 2018).

[17] VIRES Simulationstechnologie GmbH. *OpenSCENARIO v0.9 XML Schema*. 2017. URL: `http://www.openscenario.org/docs/OpenSCENARIO_v0.9_schemata.zip` (abgerufen am 14. 04. 2018).

Funktionale Dekomposition - Ein Beitrag zur Überwindung der Parameterraumexplosion bei der Validation von höher automatisiertem Fahren

Christian Amersbach* und Hermann Winner†

Zusammenfassung: Ein statistischer Sicherheitsnachweis für hochautomatisiertes Fahren ist mit aktuellen Methoden, wie der Felderprobung, nicht möglich. Neuartige Methoden, bspw. szenario-basiertes Testen, könnten helfen dieses Dilemma zu lösen. Trotzdem führen die Vielzahl an Einflussparametern und deren Werteraum zu einer riesigen Menge an möglichen Kombinationen innerhalb des Parameterraums. Da es aktuell noch zu wenig Erfahrung mit automatisierten Fahrfunktionen gibt, kann der benötigte Testraum nicht durch wissensbasierte Testfallerstellung reduziert werden. Partikuläres Testen mittels Funktionaler Dekomposition der Fahrfunktion kann potentiell dazu beitragen, diese Parameterraumexplosion zu überwinden. Basierend auf Beispielszenarien und Annahmen für die benötigte Testraumabdeckung und Diskretisierungsstufen der Einflussparameter scheint damit eine Reduktion des Testumfanges um bis zu zwei Größenordnungen möglich.

Schlüsselwörter: Automatisiertes Fahren, Funktionale Dekomposition, Parameterraum, PEGASUS

1 Einleitung

Die technische Entwicklung automatisierter Fahrzeuge hat ein Niveau erreicht, das eine baldige Markteinführung erlauben würde. Der Markteintritt höher automatisierter Fahrzeuge (d. h. SAE-Level 3 und höher [1], HAF) wird von verschiedenen Herstellern angekündigt (z.B. [2]). Dafür sind aber noch die hohen Sicherheitsanforderungen an automatisierte Fahrzeuge und die Herausforderung, dass ein solches System als sicher genug nachzuweisen ist.

1.1 Milliarden von Testkilometern bis zur Einführung von höher automatisiertem Fahren

Wenn die aktuellen Test Konzepte für HAF beibehalten werden, würden die benötigten Testdistanzen dramatisch ansteigen. Unter der Annahme, dass die automatisierte Fahrfunktion doppelt so sicher - bezogen auf die Anzahl tödlicher Unfälle - wie ein

* Christian Amersbach ist Wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail:amersbach@fzd.tu-darmstadt.de).

† Hermann Winner ist Leiter des Fachgebiet Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail:winner@fzd.tu-darmstadt.de).

menschlicher Fahrer ist, müssten laut Wachenfeld und Winner [3] ca. 6,6 Milliarden Testkilometer unter repräsentativen Bedingungen im Realverkehr für den Sicherheitsnachweis des sog. „Autobahn Chauffeur“ [4], ein SAE Level 3 [1] System, zurückgelegt werden. Dieses Beispiel zeigt, dass ein streckenbasierter, statistischer Sicherheitsnachweis für HAF in der Praxis vor Markteinführung nicht durchführbar ist; insbesondere, da die Tests wiederholt werden müssten, wenn Änderungen am System vorgenommen werden. Deshalb werden alternative Methoden für den Sicherheitsnachweis benötigt.

1.2 Das Forschungsprojekt PEGASUS

Um neue Standards und Methoden zur Absicherung von HAF zu entwickeln, wurde das ***P**rojekt zur **E**tablierung von **g**enerell **a**kzeptierten Gütekriterien, Werkzeugen und Methoden sowie **S**zenarien **u**nd **S**ituationen zur Freigabe hochautomatisierter Fahrfunktionen (PEGASUS)* 2016 ins Leben gerufen. Im Projekt, das durch das Bundesministerium für Wirschaft und Energie (BMWi) aufgrund eines Beschlusses des Deutschen Bundestages gefördert wird, definieren 17 Partner aus Wissenschaft und Industrie einen Stand der Technik zur Absicherung von HAF [5]. Die vorliegende Arbeit ist im Rahmen von PEGASUS entstanden.

1.3 Szenariobasierter Ansatz

Im Projekt PEGASUS wir ein sog. szenariobasierter Ansatz verfolgt. Es wird angenommen, dass der Großteil der Fahrstrecken auf Autobahnen ohne Probleme oder kritische Situationen zurückgelegt werden kann, während kritische Szenarien im Realverkehr selten auftreten und zufällig verteilt sind. Testen der an erster Stelle genannten gewöhnlichen Szenarien liefert keinen relevanten Erkenntnisgewinn für den Absicherungsprozess. Deshalb sollte es möglich sein, durch Identifikation kritischer Szenarien - die in der Simulation oder auf dem Prüfgelände reproduziert werden können - den nötigen Testumfang signifikant zu reduzieren. In PEGASUS werden relevante Szenarien mit Metriken identifiziert [6] oder, basierend auf Ontologien [7], synthetisch erzeugt und in einer zentralen Szenariodatenbank [8] abgelegt.

2 Parameterraumexplosion

Der in 1.3 eingeführte szenariobasierte Ansatz kann zwar den Absicherungsaufwand für HAF potentiell reduzieren, führt aber immer noch zu einer riesigen Anzahl konkreter Testfälle, die für ein logisches Szenario evaluiert werden müssen. Beispielsweise behauptet Waymo in seinem Safety Report, „tausende Variationen“ eines einzelnen Szenarios zu erzeugen [9]. Diese sog. Parameterraumexplosion wird maßgeblich von drei entscheidenden Faktoren verursacht, die in den folgenden Abschnitten erläutert werden:

2.1 Einflussparameter

Bereits einfache logische Szenarien werden von einer Vielzahl an Parametern beschrieben, die das Testobjekt beeinflussen. Bagschik et al. definieren ein Fünf-Ebenen-Modell, um diese Parameter zu strukturieren [7]. Schuldt, der das Modell ursprünglich definiert hat, beschreibt auch, wie diese Einflussparameter ermittelt und analysiert werden können und

nennt verschiedene Informationsquellen, bspw. Normen und Richtlinien zur Anlage von Straßen, Fahrzeugkataloge oder Expertenwissen [10, S. 85 ff.]. Jedem Parameter p_i können v_i verschiedene Werte zugewiesen werden. Da die Mehrheit der Parameter einen kontinuierlichen Werteraum hat (z.B. Geschwindigkeit), hängt v_i von der gewählten Diskretisierung ab. Die Wahl der richtigen Diskretisierung ist eine Herausforderung, die es noch zu lösen gilt, da eine zu grobe Diskretisierung zu Lücken im Parameterraum und damit zu unentdeckten Problemen führen kann, während eine zu feine Diskretisierung den Testumfang unnötig in die Höhe treiben würde (siehe 2.4). Die Herausforderung der Diskretisierung soll hier nicht gelöst werden, stattdessen werden für die vorliegende Studie Werte für v_i angenommen, die bspw. aufgrund von Expertenwissen als geeignet gelten sollen.

2.2 Systematische Testfallerzeugung

Eine Szenario, das von *N* Parametern ($p_1, \dots, p_i, \dots, p_N$) mit jeweils v_i Instanzen definiert wird, führt zu

$$S_N = \prod_{i=1}^{N} v_i \qquad (1)$$

möglichen Testfällen [11, S. 169]. Deshalb wächst S_N mit jedem zusätzlichen Parameter oder Diskretisierungsschritt stark progressiv an. Aus Zeit- und Kostengründen ist es einerseits nicht möglich, alle theoretisch möglichen Testfälle zu testen [12, S. 539]. Andererseits gibt es aber zu wenig Erfahrung mit HAF, um mit wissensbasierter Testfallerzeugung eine vollständige Testabdeckung zu erreichen. Deshalb ist eine systematische Testfallerzeugung unumgänglich. Grindal et al. [11] geben eine Übersicht über Kombinationsstrategien, die für eine Testfallerzeugung genutzt werden können. Um die Anforderung der Reproduzierbarkeit, die von Wachenfeld und Winner [13, S. 433] gefordert wird, zu erfüllen, reduziert sich hier der Lösungsraum auf deterministische Verfahren. Die wichtigste Randbedingung bei der Wahl einer Kombinationsstrategie zur systematischen Testfallerzeugung ist die erforderliche Testabdeckung, die im nächsten Abschnitt behandelt wird.

2.3 Testabdeckung

In diesem Paper wird die sog. *t*-weise Abdeckung als Metrik zur Beschreibung der Abdeckung einer bestimmten Testsuite, d.h. einer Menge von Testfällen genutzt. Grindal et al. [11, S. 171 f.] definieren eine *t*-weise Abdeckung wie folgt:

> *Für eine t-weise Abdeckung muss jede mögliche Kombination aus allen Werten von jeweils t Parametern in (mind.) einem Testfall der Testsuite enthalten sein.*

Die Menge aller möglichen Kombinationen aus allen Parametern, wie in (1) berechnet, entspricht dem Sonderfall der *N*-weisen Abdeckung.

Die Frage, welche Testabdeckung für die Sicherheitsbewertung von HAF nötig ist, muss jedoch noch geklärt werden. Gründel [14, S. 19 ff.] legt dar, dass Verkehrsunfälle typischerweise durch eine Kombination verschiedener Faktoren verursacht werden und somit dem sog. *Schweizer-Käse-Modell* nach Reason [15] folgen. Unter der Annahme, dass Fehler in HAF ebenfalls multikausal sind, ist eine einfache (1-weise) Abdeckung also nicht ausreichend. Kuhn et al. [16] haben empirische Daten aus Fehlerberichten verschiedener

Domänen untersucht und führen die *failure-triggering fault interaction* (FTFI) Zahl ein, die der Anzahl von Bedingungen entspricht, die nötig sind, um einen bestimmten Fehler auszulösen. Das bedeutet, dass alle Fehler, deren FTFI Zahl kleiner oder gleich *t* ist, bei einem Test mit *t*-weiser Abdeckung entdeckt werden. In den von Kuhn et al. [16] analysierten Daten ist die größte FTFI Zahl sechs (siehe Abb. 1). Da für HAF aber nicht ausreichend empirische Daten existieren, sind die FTFI Zahlen solcher Systeme unbekannt. Deshalb wird in der vorliegenden Untersuchung angenommen, dass die Erkenntnisse aus anderen Domänen zumindest teilweise übertragbar sind und die FTFI Zahlen in der gleichen Größenordnung liegen und somit im besten Fall eine *3*-weise und im schlechtesten Fall eine *10*-weise Abdeckung nötig ist, um einen hinreichend großen Anteil aller Fehler zu finden.
Kuhn et al. [16] leiten ebenfalls her, dass die Größe einer Testsuite für *t*-weise Abdeckung unter den Annahmen, dass jeder Parameter p_i die gleiche Anzahl diskreter Werte $v_i = v$ annehmen kann und die Testfallableitung „perfekt effizient" ist (d.h. keine Duplikate in der Testsuite enthalten sind), wie folgt abgeschätzt werden kann:

$$S_t = v^t \qquad (2)$$

2.4 Resultierende Parameterraumexplosion

Nachdem die Einflussfaktoren der Parameterraumexplosion bekannt sind, kann die Größe der benötigten Testsuite in Abhängigkeit der geforderten Abdeckung und der Wertebereiche der *N* Einflussparameter abgeschätzt werden. Da in einer realen Anwendung die vereinfachende Annahme, dass jeder Einflussparameter p_i gleich viele diskrete Werte annehmen kann, nicht gegeben ist (bspw. hat die Fahrstreifenbreite weniger Diskretisierungsstufen als die Sonnenposition), muss (2) für den allgemeinen Fall angepasst werden:

$$S_t = \prod_{i=1}^{t} \max_i(v_1, \dots, v_N) \qquad (3)$$

wobei $\max_i(V)$ als das *i*-größte Element der Menge V definiert ist.

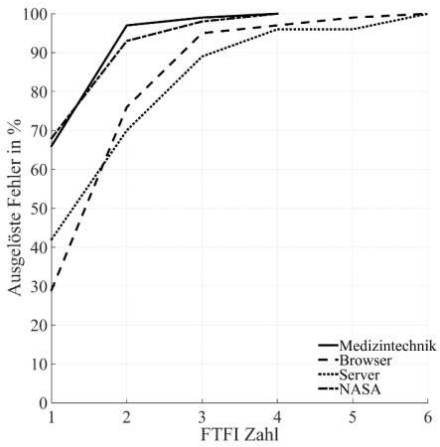


Abbildung 1: FTFI, kumulierte Verteilung nach [16]

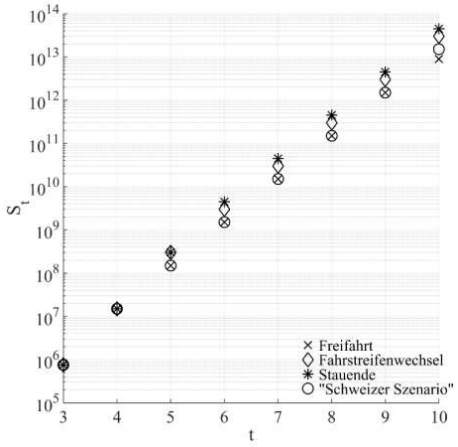


Abbildung 2: S_t für einige Beispielszenarien

Das heißt, dass für einen gegebenen Wert t für die Testabdeckung nur die t Parameter mit den größten Wertebereichen einen Einfluss auf die Größe der Testsuite haben und diese durch Elimination der Parameter mit der feinsten Diskretisierung oder durch Reduktion der Diskretisierungsschritte reduziert werden kann. Abbildung 2 zeigt die Werte von S_t für einige Beispielszenarien (siehe 3.3) in Abhängigkeit der geforderten Testabdeckung.

3 Funktionale Dekomposition zum Überwinden der Parameterraumexplosion

Ein möglicher Ansatz zum Überwinden der skizzierten Parameterraumexplosion ist die Funktionale Dekomposition der Fahraufgabe zum Ableiten partikulärer Testfälle, der von den Autoren in [17] vorgestellt wird.

3.1 Methodik

Die Methode der Funktionalen Dekomposition wird in verschieden Bereichen (z.B. Mathematik, Informatik) bereits ausgiebig genutzt, um komplexe Problem, Funktionen oder Systeme in weniger komplexe Teilprobleme, -funktionen oder -systeme zu zerlegen. Die Autoren schlagen vor, diese Methode auf das Testen von HAF zu übertragen und dabei partikulärer Testfälle für die einzelnen funktionalen Ebenen (d. h. Teilfunktionen) der Fahrfunktion zu erstellen. Hierzu wird die Fahrfunktion zunächst in voneinander unabhängige Ebenen (siehe Abb. 3) zerlegt. Die genutzten funktionalen Ebenen und deren Schnittstellen basieren auf der Dekomposition der menschlichen Fahraufgabe nach Graab et al. [18] und werden in [17] ausführlich vorgestellt.

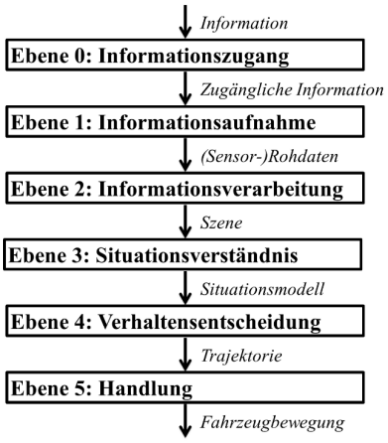


Abbildung 3: Dekompositionsebenen

3.2 Effekte zur Reduktion des Parameterraumes durch partikuläres Testen

Neben anderen in [17] skizzierten Vorteilen gegenüber dem Gesamtfahrzeugtest, kann partikuläres Testen den Parameterraum signifikant schrumpfen. Dies basiert maßgeblich auf drei Effekten:

3.2.1 Der Parameterraum einer einzelnen funktionalen Ebene ist kleiner als der Parameterraum der gesamten HAF-Funktion

Die Einflussparameter beeinflussen nur einige der funktionalen Ebenen. Beispielsweise beeinflusst die Mehrheit der Parameter, die die Umwelt repräsentieren, nur die funktionalen Ebenen 1 und 2. Deshalb sind die Parameterräume der einzelnen funktionalen Ebenen jeweils Teilmengen des Gesamtparameterraumes. Dies hat jedoch nur einen Einfluss auf die

Größe der Testsuite, wenn Parameter mit einer großen Anzahl möglicher Werte für einzelne Ebenen nicht relevant sind (vgl. 2.4). Dieser Effekt führt für die in dieser Studie untersuchten Beispielszenarien (siehe 3.3) zu einer Reduktion des Umfangs der Testsuite um ca. 50% bei großen Testabdeckungen (d. h. $t \geq 6$); für kleine Testabdeckungen ($t \leq 3$) wird der Umfang jedoch erhöht. Dennoch kann unter der Annahme, dass partikuläre Tests im Vergleich zu Gesamtsystemtests einen geringeren Aufwand erfordern, der Gesamttestaufwand auch für kleine Abdeckungen reduziert werden.

3.2.2 Systeme mit geringerer Komplexität erfordern eine kleinere Testabdeckung

Die maximale FTFI Zahl eines Systems hängt von dessen Komplexität ab. Da für HAF (noch) keine empirischen Daten existieren, mit denen die FTFI Zahlen für das Komplettsystem und die einzelnen funktionalen Ebenen bestimmt werden könnten, wird hier angenommen, dass die FTFI Zahl für die einzelnen Ebenen im Vergleich zum Komplettsystem um mindestens eins reduziert wird ($t_{\text{part.}} \leq t_{\text{system}} - 1$). Dies würde zu einer Reduktion des Parameterraumes um ca. eine Größenordnung führen.

3.2.3 Der Test der Wahrnehmungsebenen kann für Mengen ähnlicher Szenarien zusammengefasst werden.

Bei der Analyse der Einflussparameter einer HAF-Funktion wird klar, dass die Mehrheit der Parameter mit einem großen Wertebereich ausschließlich die Wahrnehmung, die in den funktionalen Ebenen 0-2 repräsentiert wird, beeinflusst. Ein offensichtliches Beispiel hierfür ist die Sonnenposition, die ausschließlich Einfluss auf die Ebenen eins und zwei hat. Zusätzlich ist hier der Wertebereich sehr groß, wenn man berücksichtigt, dass der Azimuthwinkel (relativ zum Testobjekt, OUT) Werte zwischen 0° und 360° annehmen kann und der Elevationswinkel ja nach Topologie und Position zwischen ungefähr -10° und 90° liegt. An dieser Stelle könnte man argumentieren, dass lediglich Sonnenpositionen im Sichtbereich von Kamera und LIDAR relevant sind. Dies gilt aber nur für direkte Blendeffekte. Um Wahrnehmungsfehler, die durch indirekte Blendung, d.h. Reflektionen an statischen oder dynamischen Objekten in der Szene, hervorgerufen werden auszuschließen, müssen alle möglichen Sonnenpositionen in Kombination mit Objektattributen und -positionen in der Testsuite enthalten sein. Das Zusammenfassen der Parameter mit Einfluss auf die Wahrnehmung in Äquivalenzklassenszenarien für Gruppen ähnlicher Szenarien (bspw. Szenarien auf Autobahnen mit zwei Fahrstreifen pro Richtungsfahrbahn) könnte somit den Umfang der Testsuite noch weiter reduzieren. Die Wahrnehmungsebenen würden lediglich in den Äquivalenzklassenszenarien getestet, die den gesamten Parameterraum der Szenariengruppe abdecken.

In der beispielhaften Szenariengruppe, die in dieser Studie analysiert wurde, konnte so die Testsuite für Testabdeckungen mit $t \leq 7$ um über 50 % reduziert werden. Bei großen Abdeckungen mit $t \geq 9$ würde das Zusammenfassen der mit Einfluss auf die Wahrnehmung in Äquivalenzklassenszenarien die Anzahl der Testfälle jedoch erhöhen, da die Äquivalenzklassenszenarien eine höhere Komplexität besitzen als die einzelnen Szenarien und deshalb mehr Einflussparameter mit großen Wertebereichen haben.

3.3 Untersuchte Beispielszenarien

In dieser Studie wurde die Methode der Funktionalen Dekomposition exemplarisch auf eine Gruppe von 9 Szenarien angewandt, um deren Potential zur Überwindung der skizzierten Parameterraumexplosion zu evaluieren. Hierfür wurden die Umfänge der jeweils benötigten Testsuites sowohl für den Gesamtsystemtest als auch für das partikuläre Testen mit Formel (3) abgeschätzt und verglichen.

Vier der genutzten Szenarien sind in Abbildung 4 illustriert, einige der zugeordneten Parameter sind in Tabelle 1 zusammengefasst. Zusätzlich wurden die logischen Szenarien „Folgefahrt“, „Einscherer“, „Ausscherer“, „Durchscherer“ und „Hindernis“ analysiert. Das „Schweizer Szenario“ ist die Repräsentation eines realen Unfalls mit einem Tesla Model S im Mai 2016 in der Schweiz [19], alle anderen Szenarien sind Teil des Szenarienkatalogs im Projekt PEGASUS. Die Wahl der Parameter und deren zugeordneten Wertebereichen und Diskretisierungsschritten bleibt grundsätzlich willkürlich, da sie aufgrund fehlender empirischer Daten auf Annahmen und Expertenwissen beruht. Die Wahl anderer Parameterräume würde zwar die Absolutwerte beeinflussen, aber nicht den relativen Vergleich zwischen partikulärem Testen und dem Gesamtsystemtest.

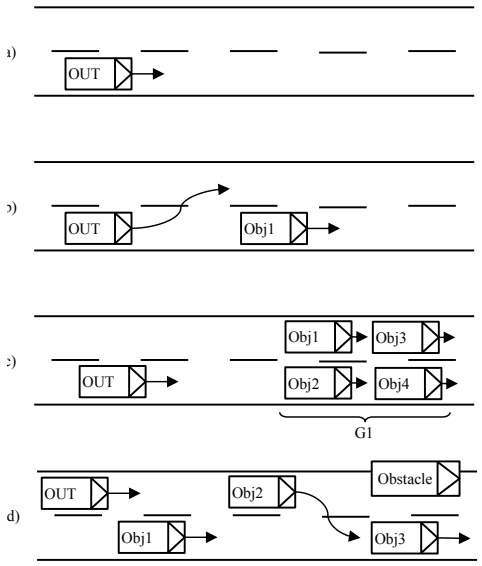


Abbildung 4: Beispielszenarien:
a) Freifahrt, b) Fahrstreifenwechsel,
c) Stauende, d) „Schweizer Szenario“

Tabelle 1: Exemplarischer Parameterraum (Auszug)

	Parameter p_i	Scenario				Influence on functional layer					
		free driving	lane change	traffic jam	Swiss scenario	0 Inf. Access	1 Inf. Reception	2 Inf. Processing	3 Sit. Under-standing	4 Behavior. Decision	5 Action
		number of discretization steps v_i									
1 Road Level	width lane 1	2				x	x			x	
	width lane 2	3				x	x			x	
2 Traffic Infra-structure	type of boundaries/ barriers	10				x	x	x			
	speed limit	5							x	x	
4 Objects	type of object 1	-	5	5	5	x	x	x	x		
	position of obstacle (x,y)	-	-	-	50	x	x		x	x	
	manoever of object 1	-	1	1	5	x	x		x	x	
5 Environment	sun position	250	250	250	250		x	x			
	precipitation (rain, snow, etc.)	10	10	10	10	x	x	x			x

3.4 Potential zur Reduktion des Parameterraumes mittels Funktionaler Dekomposition

Die in 3.2 beschrieben Effekte führen in Summe zu einer Reduktion des Umfangs der Testsuite um einen Faktor zwischen 130 für *3*-weise Abdeckung und 20 für *10*-weise Abdeckung, was einer Elimination von 99 % - 95 % der Testfälle im Vergleich zum Gesamtsystemtest entspricht (siehe Abbildung 5). Das Reduktionspotential ist insbesondere bei kleinen Testabdeckungen sehr groß, da hier der in 3.2.3 beschriebene Effekt besonders stark ausgeprägt ist, während er bei großen Abdeckungen keinen $(t \geq 8)$ keine Auswirkungen mehr hat.

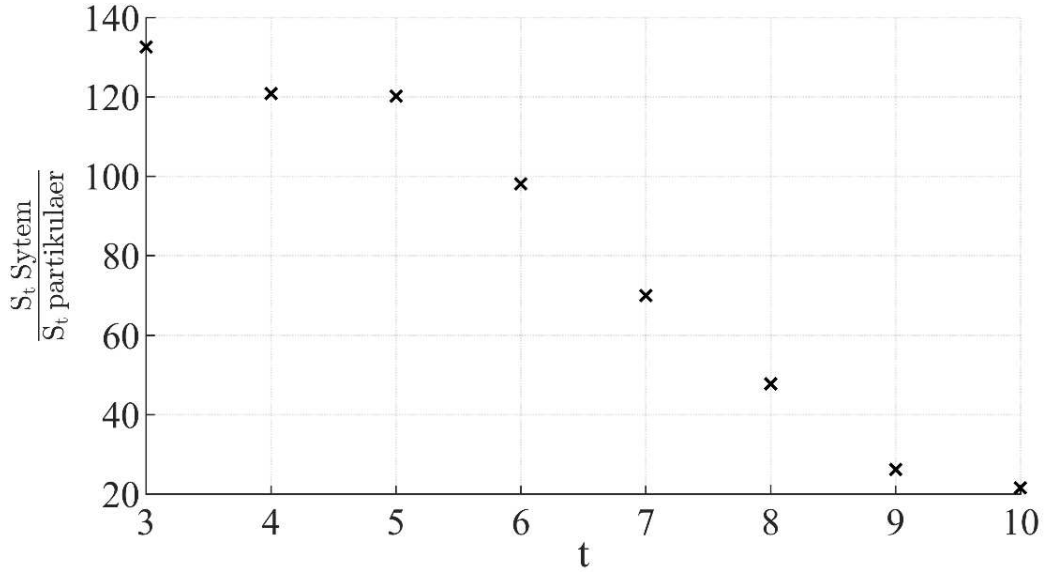


Abbildung 5: Reduktion des Testumfangs durch partikuläres Testen

4 Fazit

In diesem Artikel wurde die benötigte Größe von Testsuiten für die szenariobasierte Absicherung von HAF für Beispielszenarien sowohl für das Testen des Gesamtsystems als auch für partikuläres Testen analysiert und verglichen.

Die Diskretisierungsschritte der Einflussparameter und die erforderliche Testabdeckung haben weiterhin den größten Einfluss auf den Umfang der Testsuiten. Dies führt einerseits zur Herausforderung der Diskretisierung, die in dieser Arbeit nicht gelöst wird. Andererseits kann die erforderliche Testabdeckung nur durch Analyse empirischer Daten, die für HAF noch nicht in hinreichendem Umfang existieren, ermittelt werden. Jedoch konnte, basierend auf Studien mit Daten aus anderen Domänen, der mögliche Bereich für die erforderliche Testabdeckung eingeschränkt werden. Zusätzlich wurden Wertebereiche und Diskretisierungsschritte für die einzelnen Einflussparameter angenommen. Mit diesen Annahmen konnte das Potential den Parameterraum mit der Methode der funktionalen Dekomposition

und partikulärem Testen von HAF-Funktionen zu schrumpfen, abgeschätzt werden. Während hierbei die Absolutwerte sehr stark von den getroffenen Annahmen abhängen, werden die relativen Vergleichswerte nur wenig beeinflusst, da die gleichen Beispieldaten für den Vergleich beider Methoden, partikuläres Testen und Gesamtsystemtest herangezogen wurden.
Durch partikuläres Testen, kann die Anzahl der Testfälle potentiell um einen Faktor von 20 … 130 (abhängig von der geforderten Testabdeckung) reduziert werden und somit ein wertvoller Beitrag zur Überwindung der Parameterraumexplosion erreicht werden. Diese Studie basiert jedoch nur auf einer kleinen Anzahl von Beispielszenarien und auf vielen Annahmen, weshalb die gewonnenen Erkenntnisse zukünftig durch eine praktische Implementierung der Methode und einen repräsentativen Vergleich validiert werden müssen.

Literatur

[1] *SAE J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, 2014.
[2] Audi MediaInfo, *Automated driving at a new level: the Audi AI traffic jam pilot.* [Online] Verfügbar unter: https://www.audi-mediacenter.com/en/press-releases/automated-driving-at-a-new-level-the-audi-ai-traffic-jam-pilot-9300. Zugriff am: Mrz. 13 2018.
[3] W. Wachenfeld und H. Winner, „Die Freigabe des Autonomen Fahrens“ in *Autonomes Fahren*, Springer, 2015, S. 439–464.
[4] F. Köster, T. Form, K. Lemmer und J. Plättner, „Wie gut müssen automatisierte Fahrzeuge fahren-PEGASUS“, *Tagungsband: AAET 2016*, S. 292–300, 2016.
[5] Deutsches Zentrum für Luft- und Raumfahrt e. V., *Forschungsprojekt PEGASUS.* [Online] Verfügbar unter: http://pegasus-projekt.info/de/home. Zugriff am: Apr. 25 2018.
[6] P. Junietz, J. Schneider und H. Winner, „Metrik zur Bewertung der Kritikalität von Verkehrssituationen und - szenarien“ in *Workshop Fahrerassistenz und automatisiertes Fahren*, Walting, 2017.
[7] G. Bagschik, T. Menzel und M. Maurer, „Ontology Based Scene Creation for the Development of Automated Vehicles“ in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, Suzhou, China.
[8] A. Pütz, A. Zlocki, J. Bock und L. Eckstein, „System validation of highly automated vehicles with a database of relevant traffic scenarios“ in *12th ITS European Congress*, Strasbourg, 2017.
[9] WAYMO, *On the Road to Fully Self-Driving: Waymo Safety Report*, 2017.
[10] F. Schuldt, „Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen-English title: Towards testing of automated driving functions in virtual driving environments“, Ph. D. dissertation, Technische Universität Braunschweig, 2017.
[11] M. Grindal, J. Offutt und S. F. Andler, „Combination testing strategies: a survey“, *Software Testing, Verification and Reliability*, Jg. 15, Nr. 3, S. 167–199, 2005.
[12] I. Sommerville, *Software engineering,* 8. Aufl. New York: Addison-Wesley, 2006.

[13] W. Wachenfeld und H. Winner, „The Release of Autonomous Vehicles“ in *Autonomous Driving: Technical, Legal and Social Aspects*, H. Winner, M. Maurer, J. C. Gerdes und B. Lenz, Hg., Berlin, Heidelberg: Springer, 2016, S. 425–449.
[14] M. Gründl, „Fehler und Fehlverhalten als Ursache von Verkehrsunfällen und Konsequenzen für das Unfallvermeidungspotenzial und die Gestaltung von Fahrerassistenzsystemen“, 2005.
[15] J. Reason, „The Contribution of Latent Human Failures to the Breakdown of Complex Systems“, *Philosophical Transactions of the Royal Society B: Biological Sciences*, Jg. 327, Nr. 1241, S. 475–484, 1990.
[16] D. R. Kuhn, D. R. Wallace und A. M. Gallo, „Software fault interactions and implications for software testing“, *IEEE transactions on software engineering*, Jg. 30, Nr. 6, S. 418–421, 2004.
[17] C. Amersbach und H. Winner, „Functional Decomposition: An Approach to Reduce the Approval Effort for Highly Automated Driving“ in *8. Tagung Fahrerassistenz*, 2017.
[18] B. Graab, E. Donner, U. Chiellino und M. Hoppe, „Analyse von Verkehrsunfällen hinsichtlich unterschiedlicher Fahrerpopulationen und daraus ableitbarer Ergebnisse für die Entwicklung adaptiver Fahrerassistenzsysteme“ in *TU München & TÜV Süd Akademie GmbH (Eds.), Conference: Active Safety Through Driver Assistance. München*, 2008.
[19] F. Lambert, *Tesla Model S driver crashes into a van while on Autopilot [Video]*. [Online] Verfügbar unter: https://electrek.co/2016/05/26/tesla-model-s-crash-autopilot-video/. Zugriff am: Aug. 31 2017.

A Traffic-based Method for Safety Impact Assessment of Road Vehicle Automation

Christian Rösener*, Friederike Hennecke†, Jan Sauerbier*,
Adrian Zlocki*, Dirk Kemper†, Lutz Eckstein*
and Markus Oeser†

Abstract: One of the major challenges for enabling market introduction of automated driving is to identify risks and benefits of these functions. For this purpose, a new framework for assessing the safety impact of automated driving functions has been investigated. The developed framework takes the characteristics of automated driving functions into account. Automated driving functions - in contrast to active safety systems - continuously control the behaviour of the vehicle. Thus, it is possible that automated driving functions will get involved less frequently in accident scenarios playing a major role at human driving, e.g. rear-end accident scenarios. On the other hand, it is likely that other previously irrelevant accident types will rise. Therefore, besides investigating the change of severity of an accident by using accident re-simulations, the changes of frequency of occurrence of driving scenarios induced by automated driving are considered as well. These changes in frequency of occurrence of driving scenarios are analysed by using traffic simulations. After determining the effectiveness of the automated driving function, it is projected and depicted over the whole territory of the Federal Republic of Germany. The methodology is applied on five generic automated driving functions as for example a generic "Motorway-Chauffeur" (SAE level 3) and a generic "Urban Robot-Taxi" (SAE level 4). This paper provides the results of the safety impact assessment of these automated driving functions.

Keywords: Safety impact assessment, traffic simulation, automated driving

1 Introduction

Due to technological progress in microelectronics and computing power, various automotive functions for supporting the driver have been developed during the last decade. These so-called advanced driver assistance systems (ADAS) are supporting the driver on different levels of the driving task. Driven by recent developments in algorithms for environment perception and decision making, the ultimate goal of vehicle automation seems to be a solvable task as shown by several demonstrations [1].

* the authors are with the Institute for Automotive Engineering (ika) of RWTH Aachen University (e-mail:{christian.roesener, jan.sauerbier, adrian.zlocki, lutz.eckstein}@ika.rwth-aachen.de)

† the authors are with the Chair and Institute of Highway Engineering (isac) of RWTH Aachen University (e-mail: {hennecke, kemper, oeser}@isac.rwth-aachen.de)

However, due to an increasing complexity of decision making algorithms, identifying benefits and drawbacks will be challenging. Hence, new safety impact assessment methods have to be designed which are based on detailed accident-, FOT- and simulation data and that are assessing the automated driving functions with respect to a certain baseline. Since automated driving will not be able to avoid all accidents on roads, e.g. due to the misbehaviour of other traffic participants and physical limits, a baseline for assessment has to be defined. According to the *German Ethics Commission for Automated and Connected Driving*,

> *"[..] the licensing of automated systems is not justifiable unless it promises to produce at least a diminution in harm compared with human driving, in other words a positive balance of risks [..]"[2]*

Consequently, the reference for safety impact assessment needs to be human driver performance. In order to assess automated driving functions with respect to human driver performance, this paper introduces a method for safety impact assessment that takes the characteristics of continuous road vehicle automation according to [3] into account. Human driver performance is used as a baseline.

2 Background

For safety impact assessment of (advanced) driver assistance systems with environment perception, many different methods have been used in the past. All these methods have in common, that they compare driving situations without the system with driving situations, in which the system is activated. One valid approach for determining the effectiveness of ADAS is the accident re-simulation on basis of in-depth accident data, e.g. as applied in [4]. In this case, reconstructed accident scenarios from detailed accident data, such as the German-in-depth accident database (GIDAS) [5], are simulated with and without the considered function. The difference in performance in the situation, e.g. probability of severe injuries, is considered as the benefit of the function. A disadvantage of this approach is that new induced driving scenarios by automated driving cannot be considered, because these are not represented in the accident data. Another approach for safety impact assessment based on recorded data is the field operational test (FOT) as presented in [6]. Here, huge amounts of driving data without function (*control condition*) and with activated function (*experimental condition*) are collected. The safety impact of the considered function is analysed by investigating the change in frequency of occurrence of incidents and near-crashes compared to the baseline. For safety impact assessment of a function in defined situations, driving simulator studies can be used as well. This approach allows a detailed investigation of human driver performance with and without the considered function as demonstrated in [7], but requires a selection of situation parameters to be presented to the drivers. As described previously, automated driving functions need to be assessed in the whole entity of possible driving situations in their operational design domain. Hence, simulations of these functions in the whole traffic are a promising approach as presented in [8]. However, validation of these simulations remains challenging because of the variety and complexity of models necessary for safety impact assessment.

Based on the available methods presented previously, a suitable method for assessing the safety impact of road vehicle automation is defined. Although accident re-simulation based on detailed accident data is a valid approach, it will not be suitable for assessing automated driving functions since this approach is based on previously recorded detailed accident data from human driving. In order to identify new driving situations induced by automated driving functions, a FOT would be suitable. However, considering the necessary resources difficult to realize. Thus, a holistic approach including several data sources and in particular traffic simulation as proposed in [8] is realized for safety impact assessment of automated driving functions.

3 Approach for safety impact assessment

While in the past active safety systems were assessed on a set of recorded accident scenarios obtained from human driving [4], this approach will not be sufficient concerning automated driving functions. Automated driving functions – in contrast to active safety systems – are continuously controlling the behaviour of the vehicle. Due to this reason, it is possible that automated driving functions do not get involved in previously important accident scenarios any longer while other, at human driving less relevant accident scenarios, will become more important. Hence, future assessment approaches have to take the change in frequency of occurrence of driving scenarios induced by automated driving functions into account. Therefore, besides re-simulation of detailed accident scenarios for identifying the changes in severity due to automated driving, the changes in frequency of occurrence of relevant driving scenarios have to be modelled and identified as well. The overall approach for safety impact assessment incorporating the prediction of frequencies of driving scenarios is presented in Figure 1.

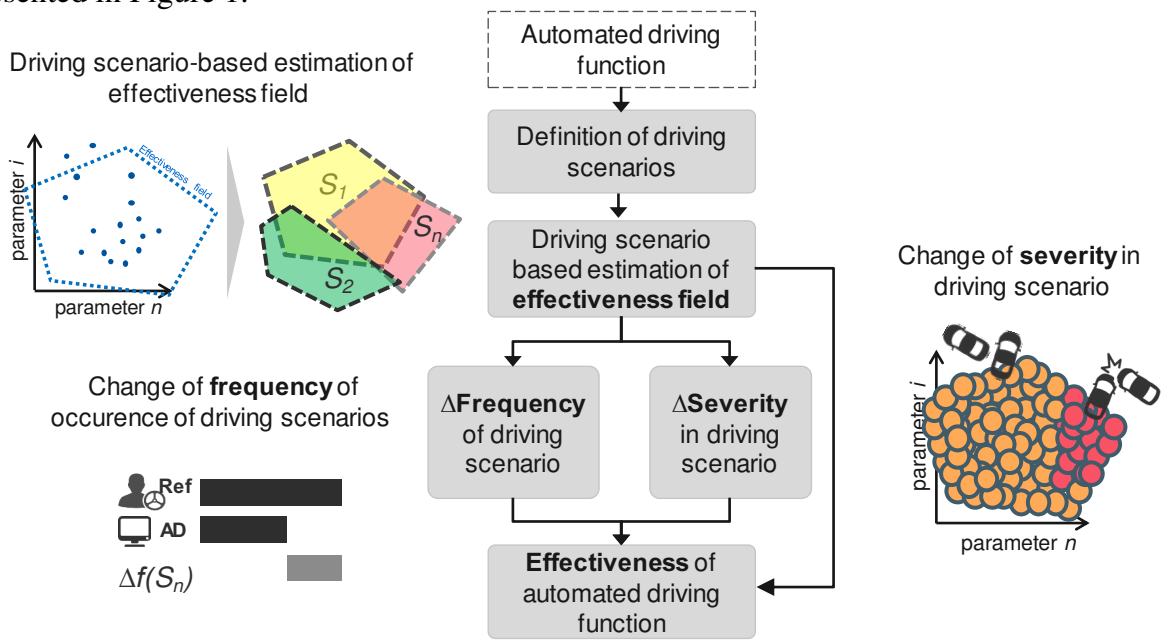


Figure 1: Methodological approach for safety impact assessment of automated driving

Based on a definition of the automated driving functions and the addressed driving scenarios the effectiveness field – all addressed accidents and safety relevant driving situations – in the considered accident data are identified. Afterwards, the changes in frequencies of occurrence of the defined driving scenarios are assessed by using traffic simulation. Of course, these changes of frequencies of driving scenarios can be derived as well from FOT-data. For identifying the changes in severity in the defined driving scenarios they are simulated with and without the automated driving function while the reference performance is modelled by human driver performance models.

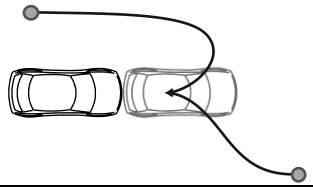
Thus, the overall effectiveness E of an automated driving function in terms of safety can be calculated by multiplication of the change of severity respectively injury risk ΔI per relevant driving scenario S_i with the change of frequency Δf of the considered driving scenario S_i for all driving scenarios n in the effectiveness field of all addressed accidents.

$$E = \sum_{i=1}^{n} \Delta I(S_i) \cdot \Delta f_i(S_i) \tag{1}$$

3.1 Definition of driving scenarios based on accident type

For prospective assessment of the safety impact of automated driving functions on traffic, a *scenario-based* approach is used. First, the automated driving functions and the addressed driving scenarios are described, e.g. a “Motorway-Chauffeur” is addressing amongst others the driving scenario “passive cut-in”. In total, 13 driving scenarios have been derived based on a systematic analysis of potential collision positions of two vehicles. These 13 driving scenarios are validated by checking if all three digit accident types $UTYP3$ are covered by the defined scenarios. In Table 1, an exemplary definition of a driving scenario is given.

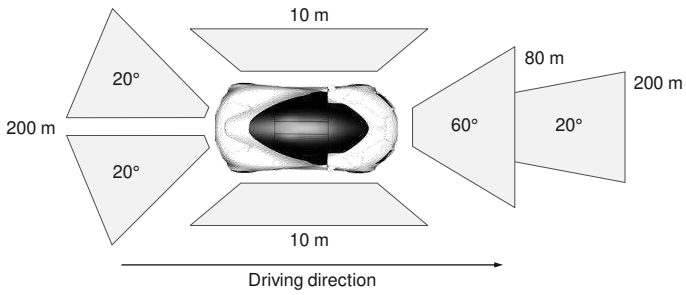
Table 1: Definition of an exemplary driving scenario for safety impact assessment of automated driving. In the illustration, the relative movement of the cutting-in vehicle is displayed.

Driving scenario	Description	Illustration of driving scenario	Covered three digit accident types
Passive cut-in	An object changes (or initiates a lane change) to the lane of another vehicle resulting in a potential collision in longitudinal direction.		204, 233, 631, 632, 634, 635,641, 642, 644, 645, 646

Afterwards the driving scenarios are linked with the accident feature “three digit accident type” $UTYP3$ to the (detailed-) accident data. By using this link, the detailed accident scenarios can be extracted from *GIDAS* (German in-depth accident study) accident data [5] as well as from the national accident statistics *DESTATIS* [9]. Next to the addressed driving scenarios, the sensor view range, addressed road types and the speed range of the automated driving function are included in the definition. Furthermore, limitations of the automated driving function are considered as well. Concerning these functional limitations, it is distinguished between environmental conditions, e.g. heavy precipitation, fog and road

conditions, icy roads, construction sites, limiting the automated driving function. In Table 2 an exemplary definition of an SAE level 3 “Motorway-Chauffeur” is given.

Table 2: Definition of automated driving functions for safety impact assessment on the example of the “Motorway-Chauffeur”

Parameter	**Value**
Name	Motorway-Chauffeur
Level of automation according to [SAE16]	3
Sensor view range	
Adressed driving scenarios	• Driving without influence from leading vehicle • Approaching static object • Approaching leading vehicle • Approaching traffic jam • Passive cut-in • Lane change
Road types and speed range	• Motorways: 0 - 130 km/h
Functional limitations	• Environmental conditions: all, except heavy precipitation (rain and snow) and fog • Road conditions: no icy road, no construction sites

3.2 Identification of effectiveness fields

After defining the assessed automated driving functions including the driving scenarios they are addressing, the accidents in which the automated driving functions have a potential impact are identified. These so-called *effectiveness fields* are estimated in the in-depth accident data and national accident statistics by using the three digit accident type $UTYP3$. Besides, the effectiveness fields are limited based on the addressed road types and limitations defined for the automated driving functions. For example, from all accidents with personal injuries $A(P)$ occurring on Motorways in Germany (6 % of all accidents), a Motorway-Chauffeur is addressing 53 %. The other accidents in the domain cannot be addressed due to the reason that driving scenarios are not covered by the functional scope of the automated driving function (12 %), driver and vehicle related limits such as technical failures or alcohol use (9 %), functional limitations (14 %) and no car involvement in the accident (12 %), see Figure 3.

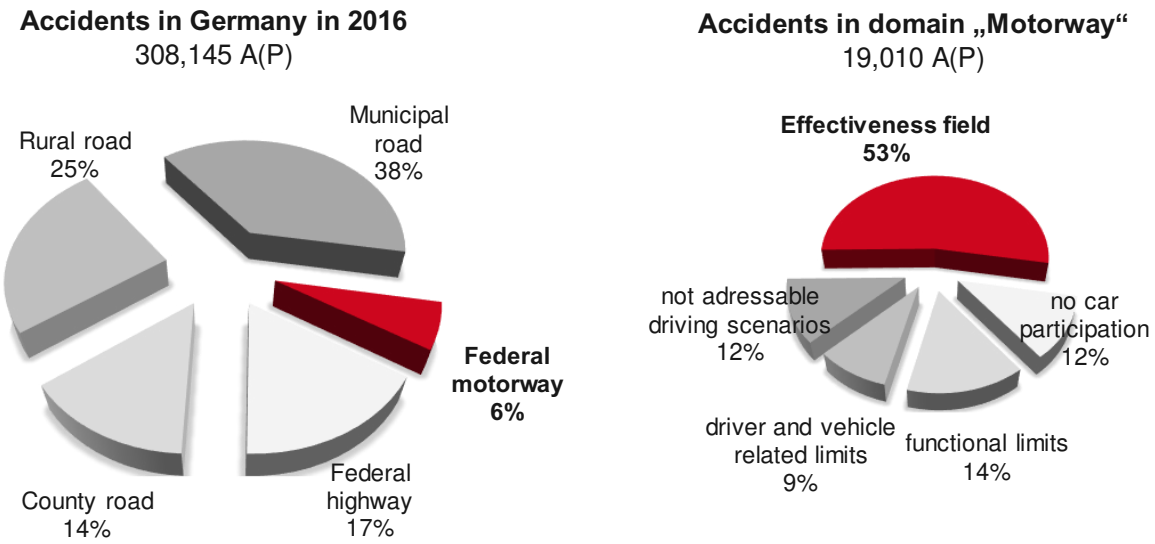


Figure 3: Analysis of accidents with personal injuries $A(P)$ in DESTATIS data with regard to road class (left) and with regard to functional limits, driver- and vehicle related limits, type of participation and addressed driving scenarios (right) on the example of the “Motorway-Chauffeur”

Finally, resulting from the analysis are the number of accidents per driving scenario in the effectiveness field and the distributions of situational variables (e.g. velocities of all participants) necessary for simulation per driving scenario, see Figure 4.

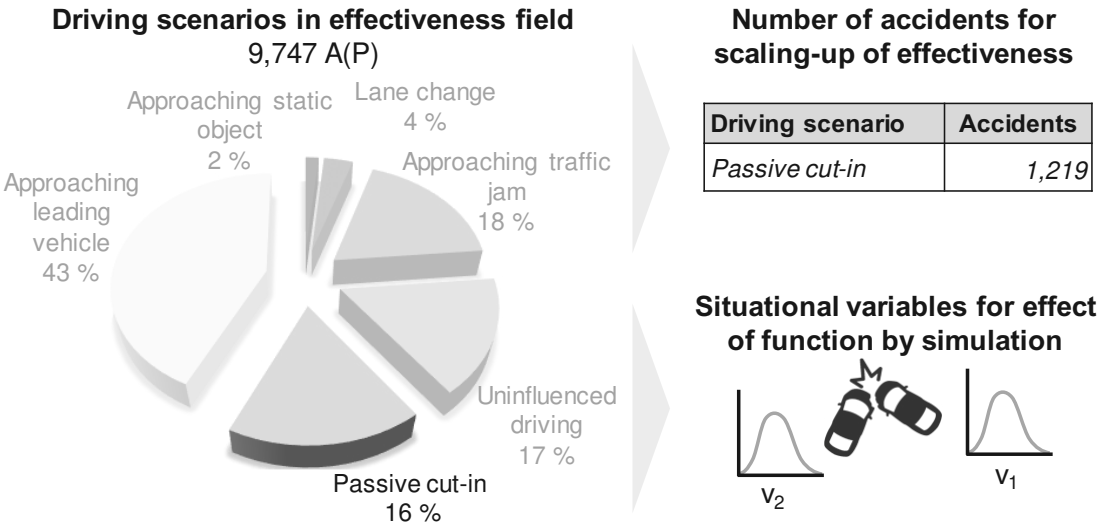


Figure 4: Analysis of accidents with personal injuries $A(P)$ in effectiveness field with respect to driving scenarios on the example of the “Motorway-Chauffeur”

3.3 Change of frequencies of occurrence of relevant driving scenarios

In the identified effectiveness fields, the potential impact of the automated driving functions in terms of safety is estimated. Automated driving functions are operating continuously and consequently may get involved with a higher frequency in certain driving scenarios while the frequency of other driving scenarios may decrease compared to human driving. Hence,

the changes of frequencies of occurrence $\Delta f_i(S_i)$ of the addressed driving scenarios are assessed by using traffic simulations. For considering the effects within mixed traffic conditions of human driven and automated vehicles, it is distinguished whether a human driven or an automated vehicle has induced or "caused" a certain driving scenario. For example, a human driver cutting-in in front of the automated vehicle can cause a "passive cut-in" driving situation. In this case, the human driver induced the driving situation while the automated vehicle was involved in it. Based on this principle, a classification scheme for driving situations is introduced, see Table 3.

Table 3: Types of interactions in mixed traffic conditions

Type of interaction	Type of vehicle driving scenario **induced** by	Type of vehicle **involved:**	Illustration
HUM-HUM	Human driver	Human driver	
HUM-ADF	Human driver	Automated driving function	
ADF-HUM	Automated driving function	Human driver	
ADF-ADF	Automated driving function	Automated driving function	

The changes of frequencies for all four defined "types of interactions" are analysed by using traffic simulation data of human driven and automated vehicles for several market penetration rates of automated vehicles. For traffic simulation, a 26 km long section of the German motorway A2 around Hanover is used. In Figure 5 an exemplary section of the considered traffic scenario (left) and the change of frequency of the driving scenario "approaching leading vehicle" is presented (right).

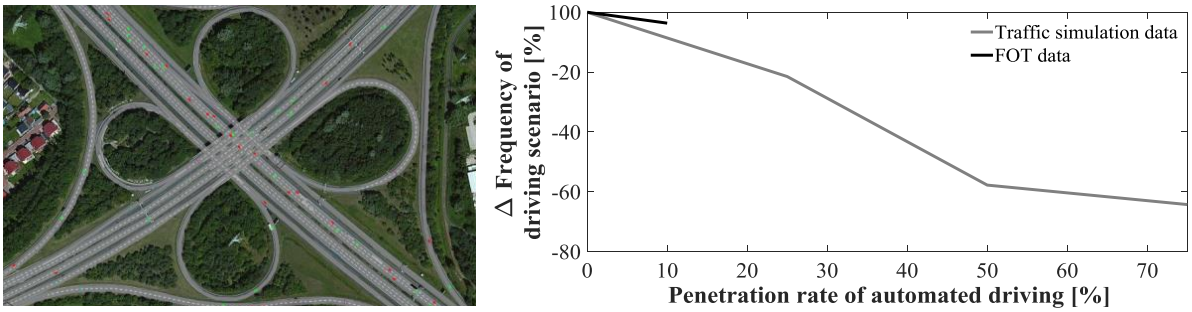


Figure 5: Traffic scenario for estimation of changes in frequencies of driving scenarios (left) and change of frequency of "approaching leading vehicle" driving scenario

3.4 Change of severity in relevant driving scenarios

The changes of severity ΔI in the considered driving scenarios are estimated by re-simulation of the addressed accidents in the effectiveness fields with and without the

considered automated driving function, see Figure 6. For this purpose, the situational variables resulting from the effectiveness field analysis of the GIDAS data are used. For reference performance, driver models parameterized from driving simulator studies such as [7] are used. Resulting is the change in severity per driving scenario. For example, in the driving scenario "passive cut-in" the probability for a severe injury can be reduced by 42.3 % by the "Motorway-Chauffeur".

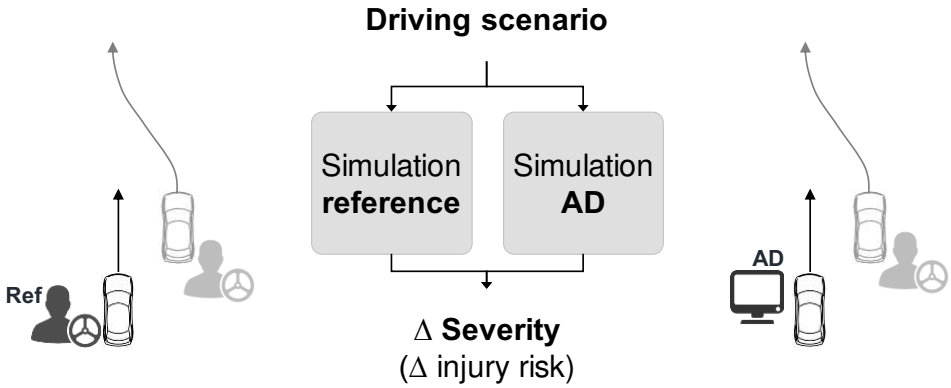


Figure 6: Re-simulation of accident situations for changes in severity in driving scenarios

3.5 Effectiveness of automated driving function

Finally, the effectiveness of the automated driving function in the effectiveness field is derived based on the changes in frequencies of all driving scenarios and the changes in severity in all driving scenarios. This process is illustrated on the example of the "passive cut-in" driving scenario at 50 % market penetration of the Motorway-Chauffeur, see Figure 7. Afterwards, the calculated effects are applied to the target population of accidents in the effectiveness fields.

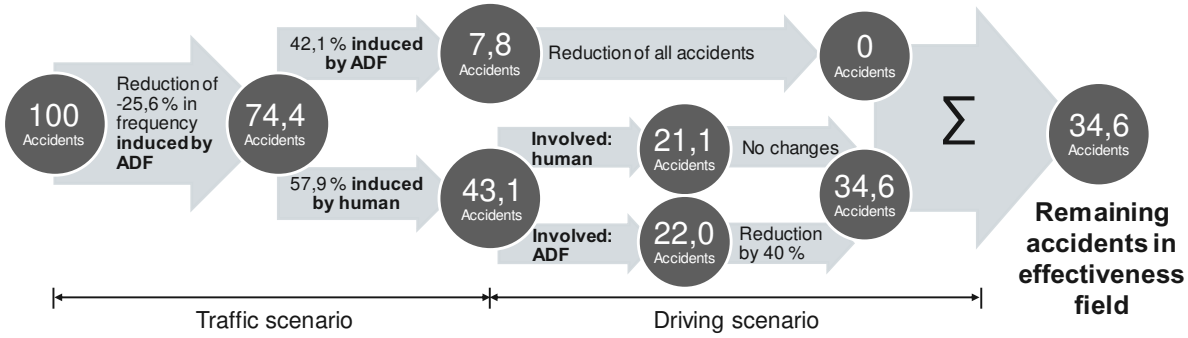


Figure 7: Method for deriving the effectiveness of an automated driving function on traffic scenario and driving scenario level on the example of a "passive cut-in" driving scenario and a market penetration of 50 %.

3.6 Projection of effectiveness on national level in Germany

The simulation-based estimated effectiveness for the different automated driving functions is scaled-up on national level for the Federal Republic of Germany. Since the effectiveness of the automated driving function is determined based on detailed GIDAS accident data that is only available for a limited geographical region in Germany the effects have to be

corrected and projected by using the national accident statistics. For this purpose, the correction factors per driving scenario are derived based on the frequency of occurrence of the defined driving scenarios in GIDAS detailed accident and national accident statistics by using the three-digit accident type.On basis of the Urban Robot-Taxi the results presented in Figure 8 will be explained. In the operation domain of the Urban Robot-Taxi 205,321 accidents with personal injuries occurred in 2016. Since only automated driving functions of passenger cars are considered, just those accidents can be addressed where at least one passenger car is among the first two participants of the accidents. These 36,486 accident cannot be addressed (see light gray area). Furthermore, 47,487 accidents per year are outside the functional limits of the Urban Robot-Taxi (see dark gray area) due to not addressed driving scenarios, alcohol and drug use, technical failures and limitations of the Urban Robot-Taxi (rain, fog, ice, construction sites).

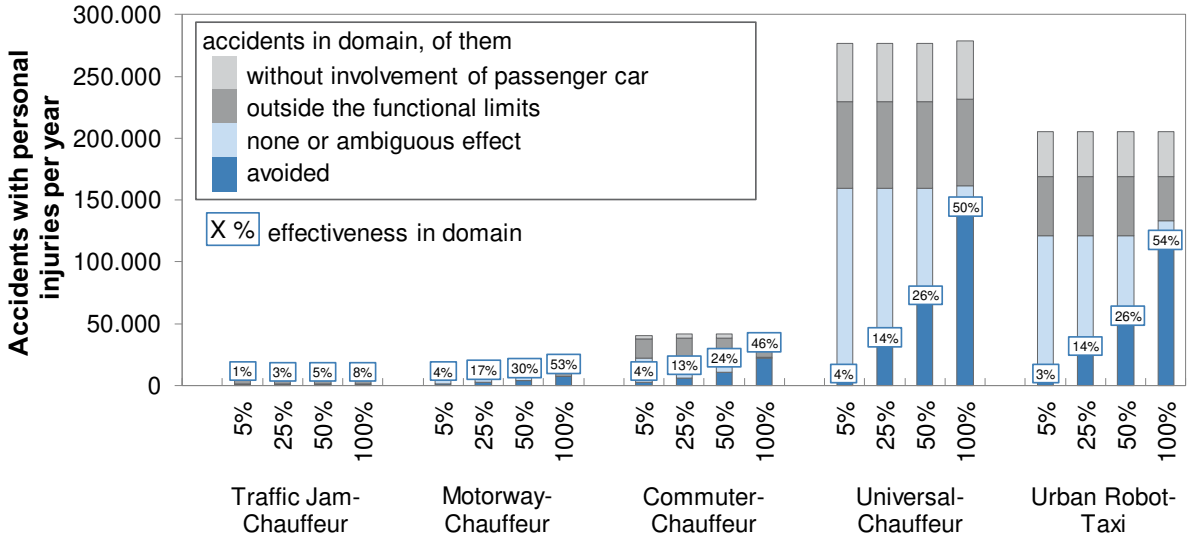


Figure 8: Effectiveness of the considered automated driving functions

The light blue area represents the number of accidents that are potentially addressable, but cannot be avoided according to the simulation results. These are for example accidents that cannot be avoided due to physical constraints. However, the severity of these accidents possibly can be reduced by a reduction of the collision speed. The dark blue area represents the number of avoided accidents. Hence, the Urban Robot-Taxi can avoid 52,517 accidents at a market penetration of 50 %. This equals an effectiveness of 26.5 % of all accidents in the operation domain.

4 Conclusion

According to the statements in [1], automated driving functions need to show a positive risk-balance compared to human driving in terms of traffic safety. Therefore, a framework for safety impact assessment of road vehicle automation has been introduced in this work. In contrast to already existing methods for safety impact assessment of active safety, this framework is considering the changes in frequency of occurrence of driving scenarios due

to the continuous operation principle of automated driving functions. Traffic simulations with automated driving functions have investigated the changes in frequency of occurrence. For determination of the change in severity in relevant driving scenarios, accident re-simulations were used. After determining the effectiveness of the automated driving functions, they are projected and depicted over the whole territory of the Federal Republic of Germany. The results indicate that, e.g. a Motorway-Chauffeur at a market penetration of 50 % has a potential for reducing about 30 % of all accidents on German motorways resulting in personal injury. This equals 2 % of all accidents on German roads. The Urban Robot-Taxi can avoid 26.5 % of all accidents with personal injury within city-limits at a market penetration of 50 %. This equals 17 % of all accidents on German roads.

Acknowledgment

The research leading to these results has received funding from the *German Federal Highway Research Agency (BASt)* within the project "*Potential Societal Benefits due to increasing Vehicle Automation*" *FE 82.0626/2015/ERB*. Responsibility for the information and views set out in this publication lies entirely with the authors. The authors would like to thank all partners within this project for their cooperation and valuable contribution.

References

[1] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen, "Experience, results and lessons learned from automated driving on Germany's highways," ITS Magazine, IEEE, vol. 7, pp. 42-57, 2015.

[2] U. Di Fabio, M. Broy, R.J. Brüngger et al., "Ethics Commission Automated and Connected Driving", Federal Ministry of Transport and Digital Infrastructure of the Federal Republic of Germany, 2017

[3] T. Gasser, A. Frey, A. Seeck et al., "Comprehensive Definitions for Automated Driving and ADAS", 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV), Detroit, USA, 2017.

[4] M.v. Noort, T. Bakri, F. Fahrenkrog, J. Dobberstein, "SIMPATO – the Safety Impact Assessment Tool of Interactive",16th International IEEE Annual Conference on Intelligent Transportation Systems, The Hague, 2013

[5] N.N., "GIDAS (German In-Depth Accident Study) Codebook", BASt/FAT, Bergisch-Gladbach, Germany, 2015

[6] M. Benmimoun, A. Puetz, A. Zlocki and L. Eckstein, "euroFOT: Field Operational Test and Impact Assessment of Advanced Driver Assistance Systems: Final Results", Proceeding of the FISITA 2012 World Automotive Congress, 2013

[7] H. Schittenhelm, "The influence of situational factors on the reaction time of Motorcar Drivers", XV. EVU-Jahrestagung, Dresden, 2006

[8] F. Fahrenkrog, "Traffic safety impact assessment for advanced driver assistance systems, PhD. Thesis, RWTH Aachen University, 2016

[9] N.N., "DESTATIS - Statistisches Bundesamt, Fachserie 8 / Reihe 7 Verkehr, Verkehrsunfälle 2016", No.: 2080700147004, Wiesbaden, 2016

Ein Fehlermodell zur Manipulation des Fahrerverhaltens in virtuellen Umgebungen zum Testen automatisierter Fahrfunktionen

Bernd Huber*, Christoph Sippl*, Stefanie Weber† und Reinhard German‡

Zusammenfassung: Dieser Beitrag beschreibt ein Konzept für ein Fehlermodell für Fahrerverhaltensmodelle zum Test hochautomatisierter Fahrfunktionen. Das vorgestellte Konzept basiert auf Ergebnissen der Unfallforschung. Ziel des Fehlermodells ist es, dem Fahrerverhaltensmodell in der Verkehrssimulation die gleichen Fähigkeiten zu geben, wie sie der reale Mensch besitzt. Das Absichern und Freigeben hochautomatisierter Fahrfunktionen stellt für die Automobilhersteller große Herausforderungen dar. Das Erproben derartiger Systeme ist ohne unterstützende Tests in einer virtuellen Umgebung aufgrund wirtschaftlicher Betrachtungen, technologischer Restriktionen oder aus Sicherheitsgründen nicht umsetzbar. Daher werden für die sinnvolle Anwendung von Verkehrssimulationen neue Konzepte benötigt.

Schlüsselwörter: Automatisiertes Fahren, Fahrerverhaltensmodell, Fehlermodellierung, Simulation, Testen

1 Motivation

In zukünftigen Mobilitätslösungen für den Individualverkehr gewinnt die Automatisierung der Fahraufgabe immer stärker an Bedeutung. Dies kann einerseits auf wirtschaftliche Gründe zurückgeführt werden, so lassen sich durch hochautomatisierte Fahrfunktionen verschiedene Anwendungsfälle ableiten [1], wodurch der Markt entsprechend wächst [2]. Andererseits soll durch den erhöhten Automatisierungsgrad die Sicherheit im Straßenverkehr steigen. Verschiedene Statistiken zeigen, dass in über 90% der Fälle menschliches Versagen als Unfallursache gilt [3], [4]. Um die Anzahl an Unfällen im Straßenverkehr weiter zu reduzieren, soll dem Menschen die Fahraufgabe abgenommen werden.
Zur Absicherung dieser Systeme existieren zwei unterschiedliche Ansätze. Funktionstests können mithilfe von Testfahrten im Fahrzeug durchgeführt und anschließend die aufgezeichneten Daten ausgewertet werden. Der zweite Ansatz nutzt Simulationen, um Funktionstests in der virtuellen Welt im Labor durchführen zu können.
Im Vergleich zu Advanced Driver Assistance Systems (ADAS) ist das Testen und Freigeben hochautomatisierter Fahrfunktionen komplexer und der Testaufwand deutlich erhöht.

*AUDI AG, 85045 Ingolstadt (e-mail: {bernd2.huber}, {christoph.sippl}@audi.de).
†Universitätsklinikum Regensburg, AARU Verkehrsunfallforschung, Am BioPark 13, 93053 Regensburg (e-mail: stefanie.weber@aaru.de).
‡Friedrich-Alexander Universität Erlangen, Lehrstuhl Informatik 7, Rechnernetze und Kommunikationssysteme, Martensstraße 3, 91058 Erlangen (e-mail: reinhard.german@fau.de).

Mit wachsendem Automatisierungsrad [5] werden die zu beherrschenden Situationen vielfältiger und komplexer. Zudem werden die Funktionen im offenen Verkehrsraum ausgeführt, sodass die Systeme unzählige Situationen mit anderen Verkehrsteilnehmern bewältigen müssen. Das Freigeben hochautomatisierter Fahrfunktionen ist ohne unterstützende Tests in der virtuellen Welt aufgrund wirtschaftlicher Betrachtungen, technologischer Restriktionen oder aus Sicherheitsgründen nicht umsetzbar. Zusätzlich zeigen die Zwischenergebnisse des PEGASUS[1] Förderprojektes, dass ein deutlicher Anteil der Erprobung hochautomatisierter Fahrfunktionen in der virtuellen Welt mithilfe von Simulationen umzusetzen sind [6].

2 Problemstellung

Mithilfe sogenannter X-in-the-Loop-Methoden werden vernetzte Funktionen unter Verwendung von Simulationen im Labor getestet und verifiziert. Je nach Phase im Entwicklungsprozess werden unterschiedliche Simulationsmethoden eingesetzt, um entwicklungsbegleitend zu testen [7].
Ulbrich et al. [8] beschreiben die gezielte Anwendung simulativer Methoden beim entwicklungsbegleitenden Testen. Dabei sichern die Autoren die Spurwechselfunktion eines hochautomatisierten Fahrzeuges ab. Hierbei wurden von Experten Simulationsszenarien entwickelt, unter welchen das System getestet wurde. Das manuelle Ausarbeiten der Simulationsszenarien stellt hier eine Limitation für die virtuelle Erprobung der hochautomatisierten Fahrfunktion dar. Das Erstellen von Simulationsszenarien ist aufwendig und kompliziert, da das Verhalten jeder einzelnen Entität in der Simulation durch den Entwickler beschrieben werden muss. Darüber hinaus ist es fraglich, ob eine Gruppe von Experten in der Lage ist, alle notwendigerweise zu testenden Szenarien für einen fehlerfreien Betrieb der hochautomatisierten Fahrfunktion zu definieren.
Das PEGASUS Projekt begegnet dieser Problematik durch das Erstellen einer Datenbank, welche logische Szenarien beinhaltet [9]. Auf Basis der logischen Szenarien und mithilfe eines entsprechend definierten Parameterraums sollen automatisiert konkrete Szenarien für den Test von hochautomatisierten Fahrfunktionen abgeleitet werden können [10].
Die logischen Szenarien der Datenbank können aus unterschiedlichen Datenquellen, wie beispielsweise aus Unfalldaten, Fahrversuchen und Feldversuchen zur Verfügung gestellt werden [11]. Eine weitere Möglichkeit, die Datenbank mit entsprechenden Verkehrssituationen zu füllen, ist das Auswerten möglichst realistischer Verkehrssimulationen in der virtuellen Welt. Die Simulation bringt hier einige Vorteile mit sich, sodass Fahrzeugfunktionen betrachtet werden können, welche noch nicht die Serienreife erreicht haben. So kann beispielsweise das Verhalten unterschiedlich stark automatisierter Fahrfunktionen untereinander untersucht werden. Auch die Ausführung einer hochautomatisierten Fahrfunktion im Grenzbereich ist mithilfe einer Verkehrssimulation gefahrlos analysierbar. Jedoch ist sicherzustellen, dass der Abstraktionsgrad der Simulationen von der Realität nicht zu physikalisch unmöglichen Zuständen führt.
Ebenso stellt das Finden von Verkehrssituationen für den Test hochautomatisierter Fahrfunktionen einen neuen Anwendungsfall für Verkehrssimulationen dar. Zunächst muss allerdings beachtet werden, dass bei diesem Anwendungsfall das Verhalten der simulierten

[1] Forschungsprojekt PEGASUS `http://pegasus-projekt.info`, [abgerufen am 22.04.2018].

Verkehrsteilnehmer nicht durch den Entwickler bestimmt werden kann. Eine Möglichkeit das Verhalten von Verkehrsteilnehmern zu bestimmen, ist das Einsetzen von entsprechenden Verhaltensmodellen. Diese nutzen sowohl statische Daten (z. B. Verkehrsinfrastrukturdaten) als auch dynamische Informationen (z. B. Objektdaten des Fremdverkehrs) aus der Verkehrssimulation, um das Verhalten von Verkehrsteilnehmern nachzubilden. Es wurden bereits unzählige Verhaltensmodelle vorgestellt, jedoch zeigen Reviews, wie zum Beispiel von Levesque et al. [12] über Fahrerverhaltensmodelle, dass die Modelle für andere Anwendungsfälle implementiert wurden und somit anderen Anforderungen genügen.

3 Ziel der Verkehrssimulation

Unter Anwendung einer Verkehrssimulation sollen Situationen generiert und identifiziert werden, unter welchen die hochautomatisierte Funktion nicht ordnungsgemäß funktioniert. Dabei soll die Funktion in der virtuellen Welt in Unfall- oder Beinaheunfall-Situationen gebracht werden, welche physikalisch möglich sind. Aus diesem Grund soll in der Simulation nicht explizit realistischer Verkehr modelliert werden, bei dem es selten zu Unfall- oder Beinaheunfall-Situationen kommt. Vielmehr sollen vermehrt Situationen generiert werden, welche zu Unfällen führen können. Solche Situationen in der virtuellen Welt zu erzeugen, ist mithilfe sogenannter *near-collision driver behavior models* möglich. Markkula et al. analysieren in [13] bereits vorgestellte *near-collision driver behavior models*. Dabei zeigen die Autoren auf, dass einerseits die jeweiligen Modelle für bestimmte Anwendungsszenarien entwickelt wurden und somit nicht allgemein gültig sind. Andererseits sehen die Autoren Herausforderungen in der Validierung der Modelle.

4 Konzept eines Fehlermodells für Fahrerverhaltensmodelle

Wie in Abschnitt 1 beschrieben wurde, ist menschliches Fehlverhalten in über 90% der Unfälle für deren Entstehung verantwortlich. Aus diesem Grund wird die Idee verfolgt, dem Fahrerverhaltensmodell menschliche Fehlverhaltensweisen beizubringen.
Im Folgenden wird das Konzept des Fehlermodells erläutert. Dabei wird zuerst auf die Datenbasis für das Fehlermodell eingegangen, die auf Unfalldaten der AARU (Audi Accident Research Unit[2]) beruht. Die Datenbasis gibt Aufschluss darüber, welche Fehler dem Menschen während der Ausführung der Fahraufgabe unterlaufen. Daraufhin werden entsprechende Anforderungen an das Fahrerverhaltensmodell abgeleitet. Anschließend wird anhand einer imaginativen Unfallsituation beispielhaft dargestellt, nach welcher Methodik die Informationen der AARU genutzt werden, um Beinaheunfall-Situationen in einer Verkehrssimulation zu generieren. Abschließend wird die Anwendung des Fehlermodells am Beispiel des Autobahn-Chauffeurs [14] dargestellt und beschrieben.

[2] Audi Accident Research Unit `http://www.aaru.de/` [abgerufen am 22.04.2018].

4.1 Datenbasis des Fehlermodells

Das Ziel des Fehlermodells ist das Generieren von Unfall- oder Beinaheunfall-Situationen. Um solche Situationen gezielt mithilfe von Fahrerverhaltensmodellen provozieren zu können, ist es nötig, ausreichend Wissen über die Ursachen von Unfällen oder Beinaheunfällen zu haben. Eine Möglichkeit stellen hierbei die Auswertungen der GIDAS (German In-Depth Accident Study[3]) Datenbank dar. Jedoch ist bei der Auswertung der GIDAS Datenbank die Unfallursache nur mithilfe einer allgemeinen Bewertung beschrieben, wie z. B. unangepasste Geschwindigkeit des Unfallverursachers. Detaillierte Informationen über die Unfallursachen liefert die AARU. Neben technischen und medizinischen Faktoren wird bei der Unfallerhebung durch die AARU auch eine psychologische Beurteilung des Unfallhergangs durchgeführt. Dabei wird der menschliche Informationsverarbeitungsprozess der Unfallbeteiligten mithilfe der 5-Step-Methode in fünf Fehlerklassen kategorisiert. Diese gliedern sich in Informationszugang, Informationsaufnahme, Informationsverarbeitung, Zielsetzung und Handlung [4]. Die Ergebnisse der psychologischen Beurteilung bilden die Datenbasis für das Fehlermodell.

4.2 Anforderungen an Fahrerverhaltensmodelle

Um die in Abschnitt 4.1 beschriebenen Informationen in der Fahrerverhaltensmodellierung nutzen zu können, muss das Fahrerverhaltensmodell die Fehlerkategorien der 5-Step-Methode abbilden. Da die 5-Step-Methode auf einem menschlichen Informationsverarbeitungsmodell basiert [15], eignen sich vor allem Fahrerverhaltensmodelle, welche die menschliche Entscheidungslogik abbilden. Ein Beispiel hierfür ist das Fahrerverhaltensmodell von Bellet et al. [16]. Es besitzt ein Wahrnehmungsmodul, welches den Informationszugang und die Informationsaufnahme abstrahiert. Sowie ein Kognitionsmodul, welches die Informationsverarbeitung und die Zielsetzung berechnet, und ein Aktionsmodul, welches die Handlung des Fahrers in der virtuellen Welt umsetzt.

4.3 Methodik zur Entwicklung von Modellen des menschliches Fehlverhaltens auf Basis der Ergebnisse der AARU

Zur Modellierung des menschlichen Fehlverhaltens werden neben den Ergebnissen psychologischer Auswertungen aufgenommener Unfälle durch die AARU auch die Beschreibung des Unfallhergangs, Ergebnisse der simulativen Unfallrekonstruktion sowie Interviews der Unfallbeteiligten berücksichtigt.
Ebendiese Informationen wurden von der AARU bezüglich Unfällen auf Autobahnen im Erhebungsgebiet zur Implementierung eines Fehlermodells für Fahrerverhaltensmodelle ausgewertet und zur Verfügung gestellt. Während der Analyse der bereitgestellten Daten konnten viele Unfälle in der Pre-Crash Phase gemeinsamen Mustern zugeordnet werden. Unfälle, welche dem gleichen oder ähnlichen Muster in der Pre-Crash Phase entsprechen, und das gleiche Fehlverhalten nach der 5-Step-Methode durch den Unfallverursacher aufweisen, werden im Folgenden als Unfallsituationstyp bezeichnet. Nachfolgend soll nun anhand eines imaginativen Unfallsituationstyps beispielhaft die Vorgehensweise zur Entwicklung eines Modells für das menschliche Fehlverhalten aufgezeigt werden.

[3] German In-Depth Accident Study `http://www.gidas.org/`, [abgerufen am 22.04.2018].

Der in Abbildung 1 dargestellte Unfallsituationstyp basiert auf Auswertungen verschiedener Unfallrekonstruktionen durch die AARU. Die abgebildeten Situationen stellen jedoch keine Rekonstruktionen realer Unfälle dar, sie sollen lediglich das Muster eines Unfallsituationstyps verdeutlichen.

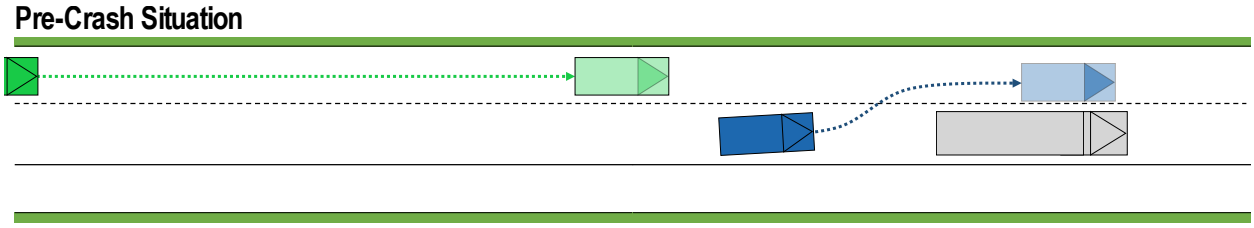


Abbildung 1: Potentielle Pre-Crash Situation auf einer zweispurigen Autobahn.

Abbildung 1 zeigt eine potentielle Pre-Crash Situation auf einer zweispurigen Autobahn mit Seitenstreifen. Die linke Spur wird hierbei von einem PKW (grün) mit sehr hoher Geschwindigkeit (> 180 km/h) befahren. Gleichzeitig läuft ein weiterer PKW (blau) auf einen LKW (grau) auf. Das blaue Fahrzeug hält sich dabei an die Richtgeschwindigkeit (∼ 130 km/h), während der LKW die für ihn vorgeschriebene Geschwindigkeit von 80 km/h einhält. Aufgrund der Differenzgeschwindigkeit zwischen blauen PKW und LKW möchte der PKW den langsameren LKW überholen. Dazu initiiert das blaue Fahrzeug einen Spurwechsel auf die linke Fahrbahn. Dabei nimmt der Fahrer des blauen Fahrzeugs aus unbekannten Gründen nicht wahr, dass sich das grüne Fahrzeug mit hoher Geschwindigkeit von hinten nähert. Dieses Fehlverhalten des Fahrers des blauen PKWs ist der Kategorie Informationsaufnahme zuzuordnen. Durch die sehr hohe Differenzgeschwindigkeit zwischen grünem und blauem PKW ist es dem Fahrer des grünen PKW nicht möglich, durch eine Geschwindigkeitsanpassung einen Aufprall auf das Heck des blauen Fahrzeugs zu verhindern. Nach der Analyse verschiedener Unfallrekonstruktionen dieses Unfallsituationstyps konnten zwei unterschiedliche Verhaltensweisen des grünen Fahrzeugs identifiziert werden.

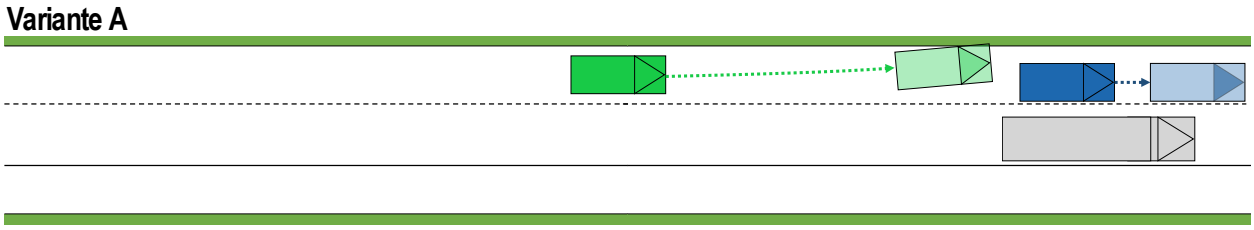


Abbildung 2: Unfallvermeidungsstrategie des grünen Fahrzeugs durch Ausweichen nach links.

Einerseits kann der Fahrer des grünen Fahrzeugs versuchen durch ein Ausweichmanöver nach links einen Unfall zu verhindern. In Abbildung 2 ist das beschriebene Ausweichmanöver visualisiert. Auswertungen verschiedener Unfallrekonstruktionen dieses Unfallsituationstyps zeigen, dass dieses Manöver in einer Kollision mit der Leitplanke und/oder einer Kollision mit dem blauen Fahrzeug enden kann.
Andererseits zeigen die Analysen der AARU eine weitere Unfallvermeidungsstrategie des Fahrers des grünen PKW. So versuchen Autofahrer durch ein Ausweichmanöver über den

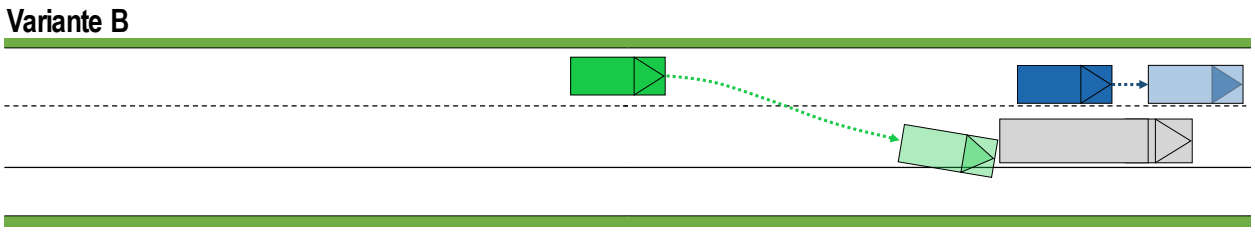


Abbildung 3: Unfallvermeidungsstrategie des grünen Fahrzeugs durch Ausweichen nach rechts über den Standstreifen

Standstreifen einen Unfall zu verhindern. Solch ein Fahrmanöver des grünen Fahrzeugs in dem zuvor beschriebenen Unfallsituationstyp ist in Abbildung 3 dargestellt. Aufgrund des fahrdynamischen Zustands des grünen Fahrzeugs kann es jedoch durch das Fahrmanöver ins Schleudern geraten, wodurch das Fahrzeug für den Fahrer nur sehr schwer kontrollierbar ist. Dabei kann das grüne Fahrzeug den LKW rammen, die Fahrbahn auf den Standstreifen verlassen, oder durch das Schleudern sich auf der Autobahn drehen.

Durch den vorgestellten Unfallsituationstyp (vgl. Abbildung 1) soll verdeutlicht werden, dass durch ein bestimmtes Fehlverhalten eines Fahrers unterschiedliche Situationen entstehen können. Dieser Effekt soll genutzt werden, um Testszenarien für hochautomatisierte Fahrfunktionen in einer virtuellen Welt zu generieren. Dazu muss das Fehlermodell das Fehlverhalten des blauen PKWs nachbilden, als auch die möglichen Unfallvermeidungsstrategien des grünen Fahrzeugs.

4.4 Anwendung des Fehlermodells zum Testen hochautomatisierter Fahrfunktionen

Für die Anwendung des Fehlermodells in einer Verkehrssimulation, ist zuerst ein Anwendungsfall der zu testenden Funktion zu definieren (vgl. Abbildung 4). Anhand des Autobahn-Chauffeurs könnte solch ein Anwendungsfall eine Autobahnfahrt mit definiertem Start und Ziel sein. Aufgrund dieser Beschreibung wird für den Test eine entsprechende Autobahn als Datenbasis für die Verkehrssimulation benötigt. Ebenso ist eine Szenario-Datei zur Verfügung zu stellen, in welcher die weiteren Verkehrsteilnehmer und deren Ziel definiert ist. Zusätzlich ist in der Szenario-Datei zu definieren, welche Fahrzeuge

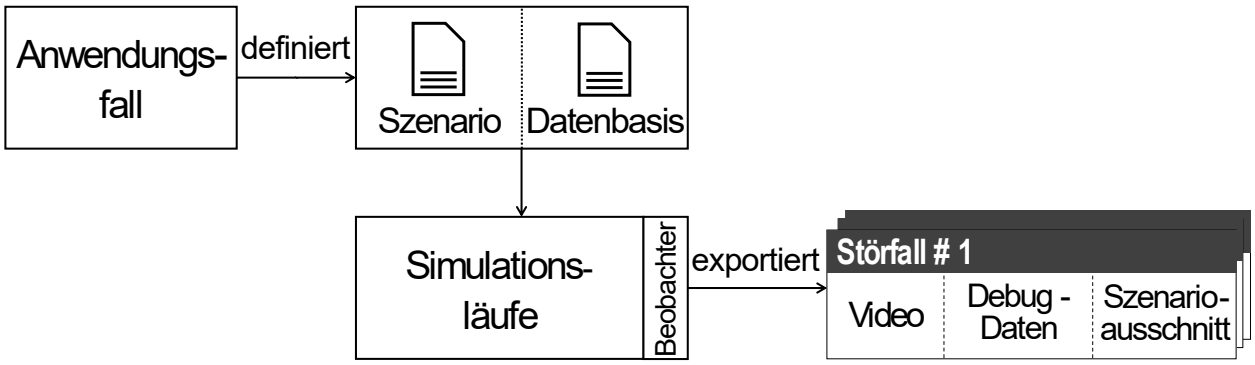


Abbildung 4: Arbeitsschritte für die Anwendung des Fehlermodells in einer nicht-deterministischen Verkehrssimulation.

durch die hochautomatisierte Funktion gesteuert werden und welche Fahrerverhaltensmodelle die übrigen Verkehrsteilnehmer steuern. Auf Basis dieser Dateien werden, mithilfe einer Verkehrssimulation, Simulationsläufe durchgeführt (vgl. Abbildung 4). Dabei fährt die hochautomatisierte Funktion das zu steuernde Fahrzeug zum definierten Ziel. Je nach Konfiguration durch die Szenario-Datei ist das Fehlermodell in verschiedenen Verkehrsteilnehmer aktiv.
Die Funktionsweise des Fehlermodells ist in Abbildung 5 beispielhaft verdeutlicht. Auf der mittleren Fahrspur, der dargestellten Autobahn, ist ein hochautomatisiertes Fahrzeug (rot) abgebildet. Während auf der rechten Fahrspur ein PKW (blau) auf einen LKW (grau) aufläuft. Hierbei manipuliert das Fehlermodell die Wahrnehmung des Fahrerverhaltensmodell, sodass das Umfeld des blauen PKW nur teilweise beschrieben wird (vgl. Abbildung 5, rechts). Durch die manipulierte Wahrnehmung würde das blaue Fahrzeug einen

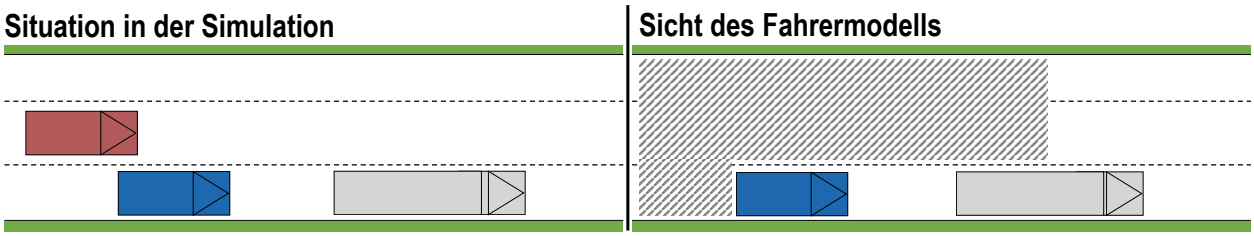


Abbildung 5: Beispielsituation in der Simulation unter Manipulation der Wahrnehmung. Links: Situation in der Simulation, Rechts: manipulierte Wahrnehmung des Fahrermodells

Spurwechsel durchführen um den langsameren LKW zu überholen, trotz des roten PKW auf der mittleren Spur. Auf dieses Verhalten muss die hochautomatisierte Fahrfunktion entsprechend reagieren.
Die Situation wird gleichzeitig von einem Simulationsbeobachter analysiert (siehe Abbildung 4). Je nach Verletzung funktionsspezifischer Anforderungen wird sowohl ein Szenario-Ausschnitt, das entsprechende Video der Situation als auch aufgezeichnete Debug-Daten der hochautomatisierten Fahrfunktion als Paket zur späteren Untersuchung abgelegt.

5 Zusammenfassung und Ausblick

Dieser Beitrag beschreibt ein Konzept für ein Fehlermodell für Fahrerverhaltensmodelle zum Test hochautomatisierter Fahrfunktionen unter der Verwendung einer Verkehrssimulation. Das vorgestellte Konzept basiert auf Ergebnisse der Unfallforschung. Des Weiteren wurde die Methodik verdeutlicht, in welcher Art und Weise die Daten der Unfallforschung zur Modellentwicklung genutzt werden. Dabei wurde der Begriff Unfallsituationstyp eingeführt und definiert. Anschließend wurde die Funktionsweise des Fehlermodells anhand eines Beispiels verdeutlicht. Ebenso wurden die Voraussetzungen für das Fehlermodell, als auch die Arbeitsschritte für die Anwendung des Fehlermodells in einer nicht-deterministischen Verkehrssimulation beschrieben.
Das beschriebene Konzept des Fehlermodells für Fahrerverhaltensmodelle wird anhand psychologischer Beurteilung von dokumentierten Unfällen der AARU als Konformitätsnachweis implementiert. Die jeweiligen Fehlverhaltensmodelle werden anschließend unter Zuhilfenahme weiterer Unfälle validiert. Abschließend soll die Nutzbarkeit des Fehlermodells anhand einer hochautomatisierten Fahrfunktion erprobt werden.

Literatur

[1] W. Wachenfeld, H. Winner, J. C. Gerdes, B. Lenz, M. Maurer, S. Beiker, E. Fraedrich, and T. Winkle, "Use cases for autonomous driving," in *Autonomous Driving*. Springer, 2016, pp. 9–37.

[2] A. Cacilo, S. Schmidt, P. Wittlinger, F. Herrmann, W. Bauer, O. Sawade, H. Doderer, M. Hartwig, and V. Scholz, "Hochautomatisiertes Fahren auf Autobahnen–Industriepolitische Schlussfolgerungen," Fraunhofer-Institut für Arbeitswissenschaft und Organisation IAO, Tech. Rep., 2015.

[3] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," U.S. Department of Transportation, Tech. Rep., 2015.

[4] S. Weber, A. Ernstberger, E. Donner, and M. Kiss, "Interdisziplinaere Unfallforschung - ein Zusammenschluss von Technik, Medizin und Psychologie zur Steigerung der Verkehrssicherheit," *VKU Verkehrsunfall und Fahrzeugtechnik*, vol. 52, no. 2, 2014.

[5] SAE International, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," *Standard J3016*, 2014.

[6] Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) Institut für Verkehrssystemtechnik, "Testen von L3 Systemen - ein Überblick über die aktuelle PEGASUS Vorgehensweise," [ONLINE]. Verfügbar: http://www.pegasus-projekt.info/files/tmpl/HZE-Vortraege/04_PEGASUS_HZE_Testen%20von%20L3%20Systemen.pdf, 2017, [abgerufen am 22.04.2018].

[7] S. Hakuli and M. Krug, "Virtuelle Integration," in *Handbuch Fahrerassistenzsysteme*. Springer, 2015, pp. 125–138.

[8] S. Ulbrich, F. Schuldt, K. Homeier, M. Steinhoff, T. Menzel, J. Krause, and M. Maurer, "Testing and validating tactical lane change behavior planning for automated driving," in *Automated Driving*. Springer, 2017, pp. 451–471.

[9] G. Bagschik, T. Menzel, A. Reschka, and M. Maurer, "Szenarien für entwicklung, absicherung und test von automatisierten fahrzeugen," in *11. Workshop Fahrerassistenzsysteme. Hrsg. von Uni-DAS e. V*, 2017, pp. 125–135.

[10] Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) Institut für Verkehrssystemtechnik, "Testfallgenerierung, Testraumabdeckung und Reduktion des Testaufwandes," [ONLINE]. Verfügbar: http://www.pegasus-projekt.info/files/tmpl/PDF-HZE/13_Testfallgenerierung-Testraumabdeckung-und-Reduktion-des-Testaufwandes.pdf, 2017, [abgerufen am 23.04.2018].

[11] ——, "Testspezifikations-Datenbank," [ONLINE]. Verfügbar: http://www.pegasus-projekt.info/files/tmpl/PDF-HZE/12_Testspezifkations-Datenbank.pdf, 2017, [abgerufen am 23.04.2018].

[12] A. Levesque and J. Johrendt, "The state of the art of driver model development," SAE Technical Paper, Tech. Rep., 2011.

[13] G. Markkula, O. Benderius, K. Wolff, and M. Wahde, "A review of near-collision driver behavior models," *Human factors*, vol. 54, no. 6, pp. 1117–1143, 2012.

[14] Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) Institut für Verkehrssystemtechnik, "Der Autobahn-Chauffeur," [ONLINE]. Verfügbar: http://www.pegasus-projekt.info/files/tmpl/PDF-HZE/03_Der-Autobahn-Chauffeur.pdf, 2017, [abgerufen am 23.04.2018].

[15] A. Zimmer, "Wie intelligent darf/muss ein Auto sein? Anmerkungen aus ingenieurpsychologischer Sicht," in *Kraftfahrzeugführung*. Springer, 2001, pp. 39–55.

[16] T. Bellet, P. Mayenobe, J.-C. Bornard, J.-C. Paris, D. Gruyer, and B. Claverie, "Human driver modelling and simulation into a virtual road environment," in *Human modelling in assisted transportation*. Springer, 2011, pp. 251–262.

Mensch-Maschine-Interaktion bei manuellem und automatisiertem kooperativen Fahren an Auffahrten und Kreuzungen

Ann-Kathrin Kraft*, Christian Maag*, Alexandra Neukum* und Martin Baumann†

Zusammenfassung: Kooperatives Fahren erleichtert an Auffahrten und Kreuzungen das Einfahren anderer Fahrzeuge, indem für diese eine Lücke erstellt wird. Dies kann durch eine technische Kommunikation und Abstimmung zwischen den beteiligten Fahrzeugen unterstützt werden. Die Probandenstudie im Fahrsimulator evaluiert verschiedene Konzepte von Mensch-Maschine-Schnittstellen eines Assistenzsystems für manuelles und automatisiertes kooperatives Fahren beim Auffahren auf die Autobahn und Linkseinbiegen auf Landstraßen. Die Evaluationsergebnisse zeigen, dass alle untersuchten Systemvarianten hilfreich und verständlich sind. Allerdings bevorzugen die Fahrer eine Darstellung, die das natürliche Blickverhalten unterstützt, indem Informationen in der Windschutzscheibe und der Fahrumgebung angezeigt werden. Insgesamt führt das System zu häufigeren erfolgreichen kooperativen Interaktionen.

Schlüsselwörter: Fahrsimulation, Kooperatives Fahren, Mensch-Maschine-Interaktion, Probandenstudie

1 Einleitung

Kooperatives Fahren ist ein Verhalten im Straßenverkehr, bei dem Verkehrsteilnehmer Fahrmanöver gemeinsam planen und umsetzen. Dies kann bedeuten, dass ein Verkehrsteilnehmer auf das eigene Recht verzichtet und dem Mobilitätswunsch eines Anderen Vorrang gewährt [1] [2]. Statt nur auf seine eigenen Ziele fokussiert zu sein, verzichtet der kooperative Fahrer für andere auf seine Vorfahrt, um das gesamte System möglichst leistungsfähig zu machen [3]. Auf diese Weise kann die Effizienz verbessert und die Beanspruchung beim Fahren reduziert werden – insbesondere in schwierigen Situationen wie z.B. dem Einfädeln auf eine Autobahn [4] [5] oder dem Linksabbiegen [6] [7].

Eine kooperative Abstimmung zwischen Fahrern findet aktuell hauptsächlich über Gesten, Signale (z.B. Lichthupe) und entsprechendes Fahrverhalten (z.B. Öffnen einer Lücke) statt

* Würzburger Institut für Verkehrswissenschaften (WIVW) GmbH, Robert-Bosch-Str. 4, 97209 Veitshöchheim (kraft@wivw.de, maag@wivw.de, neukum@wivw.de).

† Abteilung Human Factors, Universität Ulm, Albert-Einstein-Allee 43, D-89069 Ulm (martin.baumann@uni-ulm.de).

[1]. Dennoch kommt es immer wieder zu Missverständnissen zwischen den Beteiligten [3]. Frühere Arbeiten zu kooperativen Spurwechselmanövern auf der Autobahn weisen darauf hin, dass eine technisch unterstützte Kommunikation zwischen den potentiellen Kooperationspartnern die Häufigkeit einer Kooperation erhöhen kann [8] [9]. Demzufolge könnte eine technisch unterstützte Kommunikation und Abstimmung zwischen den beteiligten Fahrzeugen an Auffahrten und Kreuzungen ebenfalls kooperatives Fahren erleichtern und helfen, kritische Situationen zu vermeiden. Um dieses Ziel zu erreichen, sind intuitive Konzepte für die Mensch-Maschine-Interaktion (MMI) notwendig, damit Fahrer die kooperative Manöverabstimmung akzeptieren und Fahrempfehlungen korrekt und sicher umsetzen können [10] [11].

Neben der Verbesserung des Verkehrsablaufs bei manuell gesteuerten Fahrzeugen, kommt der kooperativen Manöverabstimmung vor dem Hintergrund zunehmender Fahrzeugautomatisierung eine wachsende Bedeutung zu [12]. Da zunehmende Automatisierungsgrade auf maximale Sicherheit und hohen Komfort zielen sollten [13], verhalten sich automatisierte Fahrzeuge voraussichtlich defensiver als der Durchschnittsfahrer. So können bei hohen Verkehrsdichten in bestimmten Situationen (z.B. an Auffahrten und Kreuzungen) nur durch kooperatives Verhalten anderer Verkehrsteilnehmer ausreichend große Lücken für automatisierte Fahrzeuge bereitgestellt werden. Auch hier ergeben sich Herausforderungen für die Mensch-Maschine-Schnittstelle, um beim Fahrer Akzeptanz und Vertrauen in die automatisierten Fahrfunktionen zu gewährleisten [14].

Im Mittelpunkt der dargestellten Probandenstudie im Fahrsimulator steht die Entwicklung der Mensch-Maschine-Schnittstelle eines Assistenzsystems für manuelles und automatisiertes kooperatives Fahren. Weitere Fragestellungen betreffen den Vergleich der clusterbasierten MMI-Varianten mit einer Benchmark-Variante. Diese wurde in einer Vorstudie entwickelt. Die Darstellung erfolgt über ein Head-Up-Display (HUD) sowie Augmented-Reality (AR)-Elementen. Der Vergleich bezieht neben dem subjektiven Feedback der Probanden (Akzeptanz, Vertrauen), Analysen des Blickverhaltens, der Bedienhandlungen sowie Fahrdaten ein.

Die Studie wurde im Rahmen des vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderten Verbundprojekts IMAGinE (Intelligente Manöver Automatisierung - kooperative Gefahrenvermeidung in Echtzeit) durchgeführt (http://imagine-online.de).

2 Methodik

2.1 Versuchsdesign und -ablauf

Der Proband durchfährt manuell oder automatisiert mit Quer- und Längsregelung (L3 conditional automation, SAE Level 3) [15] prototypische Einfädel- und Abbiegesituationen (siehe Abbildung 1). Das Einfädeln findet von einer ca. 200m langen Auffahrspur auf eine zweispurige Autobahn mit Verkehr auf beiden Spuren statt. Beim Abbiegen biegt der Proband an einer mit einem Stoppschild geregelten T-Kreuzung nach links auf die Hauptstraße ein, während von links und rechts vorfahrtsberechtigte

Fahrzeuge die Kreuzung befahren. Beide prototypischen Szenarien durchfahren die Probanden sowohl aus kooperationsbedürftiger (Einfädler, Einbieger) als auch aus potentiell kooperationsgewährender (Hauptfahrbahn) Perspektive. Die Probanden erleben die Situationen ohne (Baseline) oder mit Assistenzsystem für die kooperative Manöverabstimmung. Wird mit kooperativem Assistenzsystem gefahren, lernen die Fahrer drei unterschiedliche MMI-Varianten kennen.
Insgesamt werden folgende unabhängigen Variablen variiert: (1) Automatisierung (between-Faktor: manuell vs. automatisiert), (2) MMI-Variante (within-Faktor: HUD&AR, Cluster Dynamisch, Cluster Statisch), (3) Situation (within-Faktor: Einfädeln vs. Linksabbiegen), (4) Fahrerperspektive (within-Faktor: kooperationsgewährend vs. -bedürftig).

Abbildung 1: Kooperationssituation Einfädeln (links) und Linksabbiegen (rechts) jeweils aus Sicht des Kooperationsbedürftigen.

Jeder Proband durchfährt jede Strecke zwei Mal. In der ersten Fahrt lernt der Proband das System kennen. Der Proband wird in dieser Fahrt gebeten laut mitzudenken. In der zweiten Fahrt (Evaluationsfahrt) werden neben Fragebogendaten zur Akzeptanz objektive Daten (Fahrdaten, Blickdaten) erfasst. Der Versuch dauert ca. 2.5 Stunden und die Probanden erhalten eine Aufwandsentschädigung.

2.2 Versuchsaufbau

2.2.1 Fahrsimulator

Die Studie wurde im statischen High-End-Fahrsimulator der WIVW GmbH durchgeführt (siehe Abbildung 2). Das frontale Sichtfeld deckt einen Winkel von 300° (horizontal) bzw. 47° (vertikal) ab. Das Sichtfeld besteht aus fünf Bildkanälen mit einer Auflösung von je 1400×1050 Punkten, die auf eine flache Leinwand projiziert werden. Die Außenspiegel sind als LCD-Displays ausgeführt. Im Kofferraum simuliert ein Bildschirm die Sicht nach hinten. Der Innenspiegel entspricht daher dem Spiegel im Serienfahrzeug. Die Update-Rate der Grafik beträgt 60 Hz.
Ein Opel Insignia dient als Mockup für den Fahrsimulator. Im Lenkrad wurde eine Lenkkraftsimulation (SensoDrive Lenkmotor) integriert. Zwei LCD-Displays mit je 1024×768 Punkten werden für das Instrumentenfeld, in dem das clusterbasierte MMI dargestellt wird, und für die Navigationshinweise verwendet. Die Soundsimulation verwendet Original-Lautsprecher des Fahrzeugs. Zur Durchführung der Simulation wird ein Rechnernetz bestehend aus neun PCs verwendet, das über 1 Gbit Ethernet verbunden ist. Als Fahrsimulationssoftware wird SILAB® der WIVW GmbH verwendet.

Abbildung 2: Mockup und Sichtsystem des statischen High-End-Simulators (links) und beispielhafte Fahrersicht inkl. Lenkradbutton (durch Pfeil hervorgehoben) zur Bestätigung einer Kooperationsanfrage (rechts).

Der Versuchsleiter im Operatorraum steuert die Simulation und überwacht anhand von Mitschau-Monitoren den Versuch. Über eine Gegensprechanlage kann er jederzeit mit dem Probanden kommunizieren. Neben der Aufzeichnung objektiver Fahrparameter (z.B. Geschwindigkeit, Abstände) und Bedienhandlungen (z.B. Gaspedal, Blinker) werden Blickdaten sowie ein Video der Fahrt aufgezeichnet.

2.2.2 MMI-Varianten

Die MMI-Varianten sind beispielhaft in Abbildung 3 dargestellt. In allen drei untersuchten MMI-Varianten (neben der Baseline) wird der Fahrer über den Status der kooperativen Manöverabstimmung informiert (z.B. für das Einfädeln aus kooperationsbedürftiger Perspektive: „Partner wird gesucht“, „Lücke wird erstellt“, „Umgebung beobachten und einfädeln“, „Kooperation erfolgreich“). Zusätzlich werden in der Variante ‚Cluster Dynamisch‘ die Verkehrssituation und die kooperierenden Partner im Cluster dynamisch visualisiert. Dabei stimmen die zeitlichen (nicht aber die räumlichen) Verhältnisse mit der tatsächlichen Trajektorie der kooperierenden Partner überein (Abbildung 3, links). Die Variante ‚Cluster Statisch‘ arbeitet mit statischen Bildern. Dadurch werden die kooperierenden Partner nicht in Übereinstimmung mit der tatsächlichen Verkehrssituation dargestellt, sondern die Bilder illustrieren die Statusmeldungen der Manöverabstimmung (Abbildung 3, Mitte). Die Variante ‚HUD&AR‘ stellt im Cluster lediglich die gefahrene Geschwindigkeit dar und projiziert den Status direkt vor dem Fahrzeug in die Szenerie und markiert die Partnerfahrzeuge direkt in der Fahrumgebung mit Hilfe eines AR-Elements über den kooperierenden Fahrzeugen.

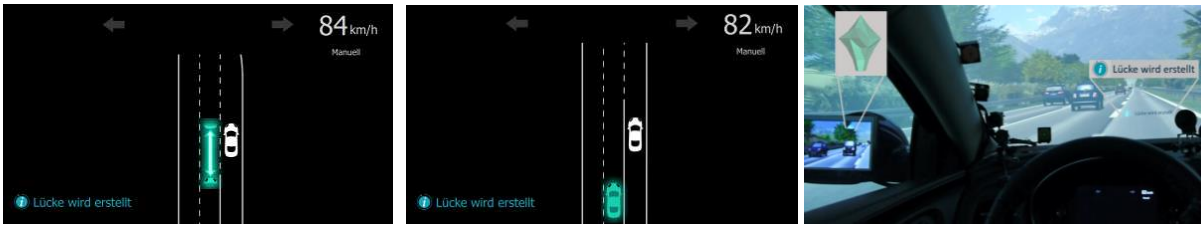


Abbildung 3: MMI-Variante ‚Cluster Dynamisch‘ (links), ‚Cluster Statisch‘ (Mitte) und ‚HUD&AR‘ (rechts) für die Situation Einfädeln aus kooperationsbedürftiger Perspektive und bei manueller Fahrt.

Im Vergleich zum manuellen Fahren wird das automatisierte Fahren im Cluster vermittelt durch eine blaue Darstellung der gefahrenen Geschwindigkeit (statt weiß), den Text „Automatisiert“ (statt „Manuell“), die Statusinformation z.B. "Ich wechsle die Spur" (statt "Umgebung beobachten und einfädeln").

2.3 Stichprobe und Datenanalyse

An der Studie nahmen N=24 Probanden (davon 13 weiblich) im Alter von 19 bis 60 Jahren teil (M = 41.8 Jahre, SD = 13.9 Jahre). Die Probanden wurden aus einem Pool im Fahrsimulator trainierter Probanden rekrutiert. Die Studienteilnehmer berichten im Mittel jährliche Fahrleistungen von ca. 15500 km (SD = 11800 km/Jahr).
Neben deskriptiven Auswertungen werden parametrische Signifikanztests (Varianzanalysen) gerechnet.

3 Ergebnisse

3.1 Akzeptanz und Präferenz der MMI-Varianten

Am Ende des Versuchs wurden die Probanden gebeten, Ränge für die verschiedenen MMI-Varianten zu vergeben (siehe Abbildung 4). Das Ranking zeigt, dass die Benchmark-Variante ‚HUD&AR' sowohl beim manuellen als auch beim automatisierten Fahren am häufigsten präferiert wird. Der Vorsprung der Variante ‚HUD&AR' vor der Variante ‚Cluster Dynamisch' ist für das automatisierte Fahren deutlich größer als beim manuellen Fahren. Die Probanden begründen ihre Präferenz damit, dass die Variante ‚HUD&AR' nicht ablenkt, da die Infos direkt im Verkehrsgeschehen angezeigt werden, der Partner deutlich zu erkennen ist und das MMI am einfachsten zu verstehen ist.

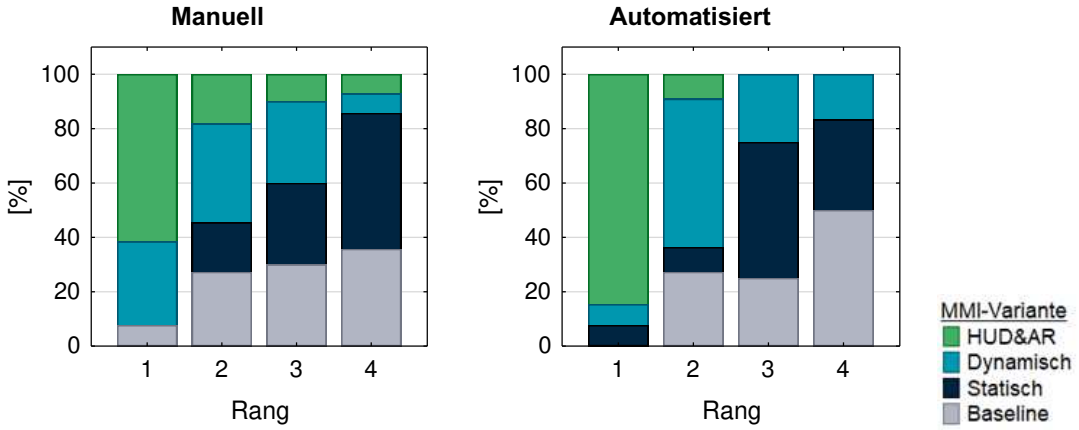


Abbildung 4: Präferenz der MMI-Varianten für das manuelle (links) und automatisierte (rechts) Fahren.

Am zweithäufigsten wird die Variante ‚Cluster Dynamisch' präferiert. Die Probanden merken positiv an, dass die Abstände bzw. die Lückengröße realistisch dargestellt werden und angezeigt wird, wo die Lücke ist. Die Variante ‚Cluster Statisch' wird am seltensten präferiert, da der Partner nicht eindeutig zu identifizieren ist, manchmal zu wenig Zeit ist, um die Anzeige zu erkennen, zu viel Information angezeigt wird und das System vom Verkehrsgeschehen ablenkt.
Sowohl nach der initialen Kennenlernfahrt als auch nach der Evaluationsfahrt genannten Wiederholung schätzen der Proband und der Versuchsleiter unabhängig voneinander das Systemverständnis des Probanden ein. Die Analyse zeigt, dass nach der Evaluationsfahrt das Systemverständnis im Vergleich zur Abfrage nach der Kennenlernfahrt signifikant

höher ist (Proband: $F_{1,21} = 56.78$, $p < .001$; Versuchsleiter: $F_{1,22} = 93.97$, $p < .001$). Weiterhin unterscheidet sich das Systemverständnis zwischen den MMI-Varianten nach der Kennenlernfahrt signifikant (Proband: $F_{2,42} = 7.45$, $p = .002$; Versuchsleiter: $F_{2,44} = 14.24$, $p < .000$). Während bei der Variante ‚HUD&AR' das Systemverständnis sehr hoch ist, sind die Clustervarianten weniger intuitiv verständlich. Dabei ist die Variante ‚Cluster Statisch' weniger verständlich als die Variante ‚Cluster Dynamisch'.

3.1.1 Perspektive Kooperationsbedürftig

In den Fahrten mit MMI-Unterstützung werden die Probanden nach jeder Situation befragt, wie hilfreich die Anzeige war (siehe Abbildung 5, links). Die Auswertung für den kooperationsbedürftigen Fahrer zeigt einen Effekt der MMI-Variante ($F_{2,44} = 17.67$, $p < .001$). Die Variante ‚Cluster Statisch' wird durchgehend als weniger hilfreich bewertet als die Variante ‚Cluster Dynamisch'. Am hilfreichsten wird die Variante ‚HUD&AR' bewertet. Dies gilt für das manuelle wie das automatisierte Fahren.

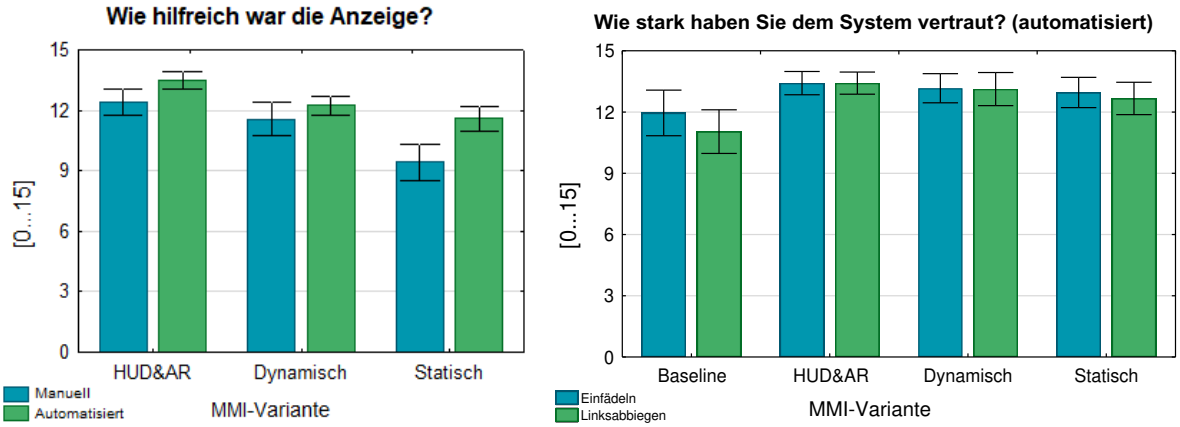


Abbildung 5: Beurteilung, wie hilfreich (links) die Anzeige (links) und wie stark das Systemvertrauen (rechts) war aus der Perspektive des Kooperationsbedürftigen (Mittelwert ± 95%-Konfidenzintervall).

Die Fahrer beurteilen beim automatisierten Fahren alle Varianten als hilfreicher als beim manuellen Fahren ($F_{1,22} = 5.52$, $p = .028$). Die Anzeige wurde in beiden Situationen als gleich hilfreich beurteilt.

Beim automatisierten Fahren ist das Vertrauen in die Automatisierung mit MMI-Unterstützung höher als ohne Unterstützung ($F_{3,33} = 13.97$, $p < .001$). Zwischen den drei MMI-Varianten und den beiden Situationen gibt es keine signifikanten Unterschiede (siehe Abbildung 5, rechts).

3.1.2 Perspektive Kooperationsgewährend

Ähnlich wie aus der Perspektive des Kooperationsbedürftigen wird die MMI-Variante ‚HUD&AR' als am hilfreichsten bewertet. Es folgen die Varianten ‚Cluster Dynamisch' und ‚Cluster Statisch' ($F_{2,44} = 6.90$, $p = .002$). Die Bewertung, wie hilfreich die Anzeige ist, hängt neben der MMI-Variante auch vom Automatisierungsgrad ab (Interaktion MMI-Variante x Automatisierung: $F_{2,44} = 4.63$, $p = .015$). Beim automatisierten Fahren wird die Variante ‚HUD&AR' als am hilfreichsten beurteilt. Beim manuellen Fahren werden die Varianten ‚HUD&AR' und ‚Cluster Dynamisch' als ähnlich hilfreich bewertet.

3.2 Blickverhalten und Ablenkung

Die Auswertung der Blickverteilung erfolgt für den Zeitraum von der Aktivierung des Systems (d.h. ab der Meldung „Partner wird gesucht“) bis zum Abschluss der kooperativen Interaktion (d.h. dem Beginn der Meldung „Kooperation erfolgreich“). Um die Ablenkung durch das MMI zu analysieren, wird die Blickverteilung des kooperationsbedürftigen Fahrers auf unterschiedliche Sichtbereiche analysiert (Cluster, Innenspiegel, aus linkem Seitenfenster, aus rechtem Seitenfenster, nach vorne). Die fahrrelevanten Informationen sind dabei in Abhängigkeit von der Situation unterschiedlich und befinden sich für den Einfädler links (Lücke auf Hauptfahrbahn) und frontal (Ende der Einfädelspur) sowie für den Linksabbieger links und rechts (Lücken im Verkehrsstrom auf der Hauptstraße).

Die Blickverteilung weist für beide Situationen auf, dass die Fahrer relativ lange auf den fahrrelevanten Bereich schauen (siehe Abbildung 6). Beim Fahren mit den Clustervarianten wird die Blickdauer auf das Cluster länger (Einfädeln: $F_{3,27} = 14.29$, $p < .001$; Linksabbiegen: $F_{3,54} = 54.56$, $p < .001$). Diese Zuwendung geht auf Kosten fahrrelevanter Sichtbereiche. Das natürlichste Blickverhalten wird bei der MMI-Variante ‚HUD&AR‘ gezeigt.

Beim automatisierten Fahren ist die gesamte Blickdauer in der jeweiligen Situation länger auf das Cluster als beim manuellen Fahren (dies gilt beim Einfädeln nur für die Clustervarianten: $F_{3,27} = 3.88$, $p = .020$; Linksabbiegen: $F_{1,18} = 6.86$, $p = .017$).

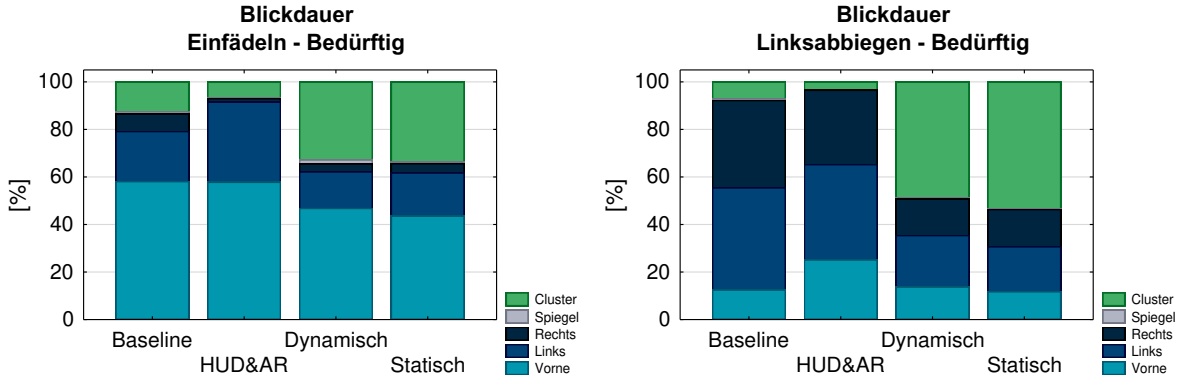


Abbildung 6: Blickverteilung des Fahrers beim Einfädeln (links) und beim Linkseinbiegen (rechts).

3.3 Fahrverhalten

3.3.1 Perspektive Kooperationsbedürftig

Aus Perspektive des Kooperationsbedürftigen erlebt der Proband mit jeder MMI-Variante und in der Baseline-Fahrt (d.h. ohne MMI-Unterstützung) jeweils zwei Einfädel- und zwei Linksabbiegesituationen, in denen ein simuliertes Fahrzeug auf der Hauptfahrbahn eine Lücke erstellt. Diese vorbereitete Lücke wird im Linksabbiegeszenario vom manuellen Fahrer mit MMI-Unterstützung häufiger genutzt als ohne MMI-Unterstützung ($Chi2_{df=3} = 44.82$, $p = .000$). Die meisten Probanden haben ohne Unterstützung durch ein MMI die Lücke auf der Hauptfahrbahn im Landstraßenszenario als ungeeignet eingeschätzt. Beim Einfädeln auf die Autobahn wird der Einfluss des MMI nicht signifikant ($Chi2_{df=3} = 3.56$,

p = .314).
Die Unterschiede zwischen den MMI-Varianten sind für beide untersuchte Szenarien nicht sehr groß, d.h. alle Varianten sind in der Lage, die Probanden zur Wahl der vorbereiteten Lücke zu animieren. Beim Auffahren auf die Autobahn wird die vorbereitete Lücke zu 79-92% gewählt, beim Linksabbiegen zu 92-100% (siehe Abbildung 7 links).
In der Bedingung automatisierten Fahrens nutzt die automatisierte Quer- und Längsregelung in allen Situationen die von den Kooperationspartnern erstellte Lücke. Keiner der Probanden greift ein und übersteuert die Automatisierung.

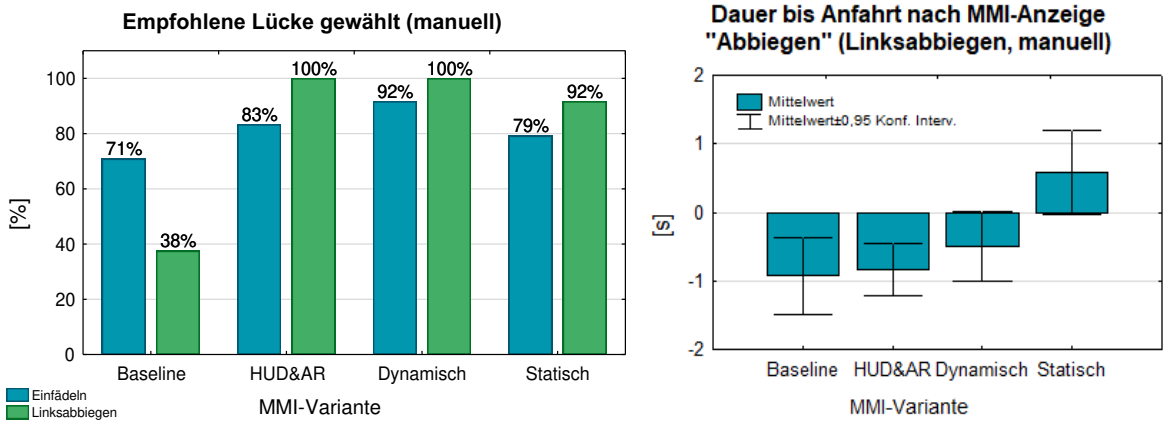


Abbildung 7: Anteil der Fahrer, die die empfohlene Lücke wählen (links), und Latenz der Anfahrt im Szenario Linksabbiegen (rechts; Mittelwert ± 95%-Konfidenzintervall).

Die Latenz mit der die Probanden beim Linksabbiegen an der Landstraße losfahren – im Vergleich zum Zeitpunkt, an dem das MMI den Status „Umgebung beobachten und abbiegen“ anzeigt – ist abhängig von der MMI-Variante ($F_{3,18}$ = 3.44, p = .039). Während die zeitliche Charakteristik des Losfahrens mit der MMI-Variante ‚HUD&AR‘ und ‚Cluster Dynamisch‘ dem natürlichen Anfahren in der Baseline ähnelt, verzögert sich das Losfahren bei der Variante ‚Cluster Statisch‘ deutlich. In dieser Variante reagieren die Probanden auf den Status des MMI (mit einer Reaktionszeit von ca. 0.6s), während sie bei den anderen Varianten und in der Baseline die Lücke antizipieren und bereits 0.5-1.0s früher losfahren als das MMI den Text „Umgebung beobachten und abbiegen“ anzeigt (siehe Abbildung 7, rechts).

3.3.2 Perspektive Kooperationsgewährend

Im Fall manuellen Fahrens bewertet der Versuchsleiter nach jeder Situation, ob der Proband kooperativ war. Die Ergebnisse zeigen, dass die Unterstützung durch ein MMI die Kooperationshäufigkeit der Fahrer erhöht. Während die Fahrer in 88% der Situationen in der Baseline für einen Einfädler eine Lücke erstellen, steigt dieser Wert mit kooperativer Assistenz und unabhängig von der MMI-Variante auf 100% ($Chi2_{df=3}$ = 9.29, p = .026). Beim Verzögern für einen Linkseinbieger auf die Landstraße sind die Probanden ohne Assistenz nur in ca. jeder zweiten Situation kooperativ (54%). Mit MMI-Unterstützung steigt der Wert in Abhängigkeit von der MMI-Variante auf 88-100% ($Chi2_{df=3}$ = 23.62, p = .000). Die kooperative Assistenz weist den Probanden auf die Kooperationsbedürftigkeit eines anderen Fahrers hin und animiert den Fahrer

wirkungsvoll, eine Lücke zu erstellen (siehe Abbildung 8, links).
Die kooperationsgewährenden Fahrer bestätigen ihre Kooperationsabsicht durch Betätigung eines Lenkradbuttons. Die Dauer bis zum Drücken des Buttons nach der Kooperationsanfrage im MMI beträgt 1.9-2.5s und unterscheidet sich nicht signifikant zwischen den MMI-Varianten und Situationen (siehe Abbildung 8, rechts).

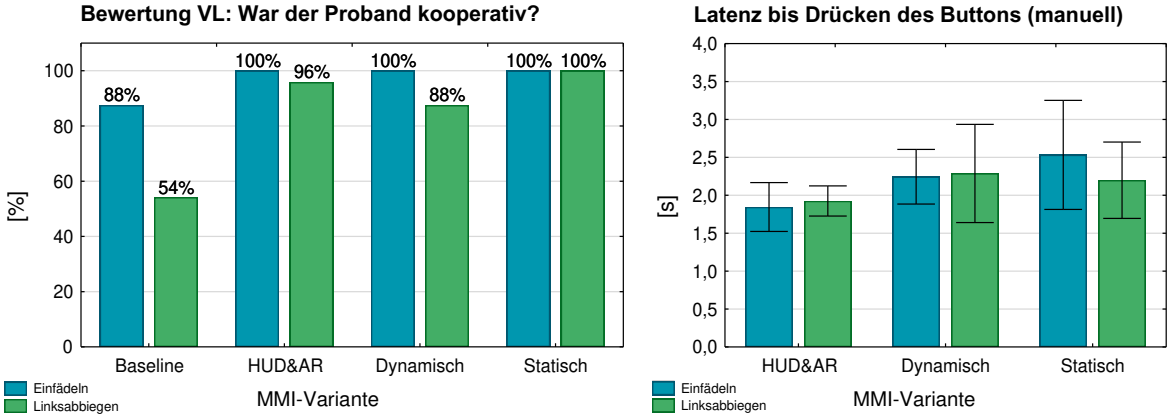


Abbildung 8: Anteil der Fahrer, die laut Versuchsleiter (VL) kooperativ fahren (links), und Latenz bis zur Bestätigung (rechts; Mittelwert ± 95%-Konfidenzintervall).

4 Zusammenfassung und Diskussion

Die Probandenstudie im Fahrsimulator betrachtet das manuelle und automatisierte kooperative Fahren an Kreuzungen und Auffahrten. Dabei werden die Auswirkungen unterschiedlicher MM-Konzepte einer Fahrerassistenz zur Unterstützung kooperativen Fahrens auf die Akzeptanz, die Ablenkung und das Fahrverhalten untersucht. Als Untersuchungssituationen dienen das Einfädeln auf die Autobahn und das Linksabbiegen auf eine Landstraße. Es werden drei verschiedene MMI-Konzepte evaluiert: ‚HUD&AR‘, ‚Cluster Dynamisch‘ und ‚Cluster Statisch‘. Bei der MMI-Variante HUD&AR werden die Informationen in einem HUD mit AR-Elementen dargestellt. Diese Variante dient als Benchmark zum Vergleich der anderen beiden Varianten, bei denen Informationen im Instrumentencluster angezeigt werden.
Die Ergebnisse zeigen, dass die Probanden grundsätzlich die Unterstützung durch ein Assistenzsystem zum kooperativen Fahren akzeptieren. Dies gilt sowohl für den Fall des manuellen Fahrens, bei dem das System häufiger zu erfolgreicher Kooperation führt, als auch beim automatisierten Fahren, bei dem der Fahrer das von der Automation durchgeführte Manöver akzeptiert. Die Ergebnisse zeigen, dass durch alle Varianten ein ausreichend großes Vertrauen in die kooperative Manöverabstimmung erreicht wurde. Dabei bezeichnen die Probanden alle getesteten MMI-Varianten als hilfreich und verständlich. Allerdings wird die Variante ‚HUD&AR‘ intuitiv am besten verstanden und folglich am besten bewertet. In der subjektiven Bewertung durch die Fahrer folgt die dynamische Clustervariante vor der statischen Variante.
Auch die Analyse des Blickverhaltens zeigt für die Variante ‚HUD&AR‘ eine Verteilung

der Aufmerksamkeit, die am ehesten dem natürlichen Blickverhalten entspricht. Mit den Clustervarianten steigt der Anteil der Blicke ins Instrumentencluster deutlich an. Dies geht auf Kosten der fahrrelevanten Blicke nach links, rechts und vorne. Trotz der Darstellung bewegter Objekte in der dynamischen Variante zeigt sich im Blickverhalten kein Unterschied zwischen beiden Clustervarianten.
Das Verhalten der Fahrer wird sowohl aus Perspektive des Kooperationsbedürftigen (Einfädler, Linkseinbieger) als auch aus Perspektive des kooperationsgewährenden Fahrers auf der Hauptfahrbahn betrachtet. Dabei zeigt sich, dass ein kooperationsbedürftiger Linkseinbieger mit der statischen Clustervariante mehr Zeit bis zum Losfahren benötigt. Anscheinend warten die Fahrer mit dieser MMI-Variante auf eine Meldung des kooperativen Assistenzsystems, während sie beim Fahren in der Baseline-Bedingung und mit den beiden anderen MMI-Varianten die erstellte Lücke antizipieren und früher anfahren. Die statische Clustervariante behindert somit das natürliche Fahrverhalten. Entsprechend wird diese Variante als deutlich weniger hilfreich bewertet. Die Analyse des Bedienverhaltens zeigt, dass ein kooperationswilliger Fahrer auf der Hauptfahrbahn ca. 2.0-2.5 s braucht, um die Kooperation durch Drücken eines Lenkradbuttons zu bestätigen.
Insgesamt zeigen die Resultate, dass MMI-Konzepte zur Unterstützung kooperativen Fahrens akzeptiert werden. Dabei werden Konzepte bevorzugt, die das natürliche Blickverhalten des Fahrers beachten und Informationen mit Hilfe von AR-Elementen im HUD darstellen. Ein clusterbasiertes MMI sollte dynamisch gestaltet sein, so dass eine Antizipation der Situation ermöglicht wird.

Literatur

[1] Maag, C. (2004). Fahrer und Verkehrsklima - Eine wissenschaftliche Untersuchung zu den Ursachen und Folgen von Emotionen im Straßenverkehr. Düsseldorf: VDI.
[2] Ellinghaus, D. (1986). Rücksichtslosigkeit und Partnerschaft. Eine sozialpsychologische Untersuchung über den Umgang unter Kraftfahrern im Straßenverkehr (12). Köln: Uniroyal-Verkehrsuntersuchung.
[3] Benmimoun, A., Neunzig, D. & Maag, C. (2004). Effizienzsteigerung durch professionelles/partnerschaftliches Verhalten im Straßenverkehr. FAT-Schriftenreihe Nr. 181. Frankfurt/Main: Forschungsvereinigung Automobiltechnik e.V.
[4] Buld, S., Hoffmann, S. & Krüger, H.-P. (2006). Fahrerassistenz auf der Grundlage von Umgebungs- und Fahrervariablen. 4. Fahrerassistenzsysteme-Workshop, Löwenstein.
[5] de Waard, D., Dijksterhuis, C. & Brookhuis, K. A. (2009). Merging into heavy motorway traffic by young and elderly drivers. Accident Analysis and Prevention 41, S. 588–597.
[6] Lord-Attivor, R. & Jha, M. (2012). Modeling gap acceptance and driver behavior at stop controlled (priority) intersections in developing countries. In Proceedings of the American Conference on Applied Mathematics (American-Math'12), S. 29-38.
[7] Chovan, J. D., Tijerina, L., Everson, J. H., Pierowicz, J. A. &Hendricks, D. L. (1994). Examina-tion of intersection, left turn across path crashes and potential IVHS countermeasures (No. DOT-VNTSC-NHTSA-94-4).

[8] Heesen, M., Baumann, M., Kelsch, J., Nause, D., & Friedrich, M. (2012). Investigation of Cooperative Driving Behaviour during Lane Change in a Multi-Driver Simulation Environment. In D. de Waard, K. A. Brookhuis, F. Dehais, C. Weikert, S. Röttger, D. Manzey, P. Terrier (Eds.), Human Factors: a view from an integrative perspective. Proceedings HFES Europe Chapter Conference Toulouse.

[9] Zimmermann, M., Schopf, D., Lütteken, N., Liu, Z., Storost, K., Baumann, M., Bengler, K. J. (2018). Carrot and Stick: A Game-theoretic Approach to Motivate Cooperative Driving through Social Interaction. Transportation Research Part C: Emerging Technologies, 88(January), 159–175.

[10] Maag, C. & Mark, C. (2012). Assisting drivers' merging onto the motorway: Evalua-tion of a new advanced driver assistance system (ADAS) using multi-driver simu-lation. Präsentation beim 19. ITS Weltkongress, Wien, 25.10.2012.

[11] Baumann, M., Tango, F., Kelsch, J. & Dziennus, M. (2014). A framework for driver-vehicle cooperation for highly automated intelligent vehicles. 5th International Conference on Applied Human Factors and Ergonomics (AHFE 2014).

[12] Maurer, M., Gerdes, J. C., Lenz, B., Winner, H. (Hrsg.) (2015). Autonomes Fahren Technische, rechtliche und gesellschaftliche Aspekte. Berlin: Springer.

[13] Fraedrich, E. & Lenz, B. (2015). Gesellschaftliche und individuelle Akzeptanz des autonomen Fahrens. In: Maurer, M., Gerdes, J. C., Lenz, B., Winner, H. (Hrsg.) (2015). Autonomes Fahren – Technische, rechtliche und gesellschaftliche Aspekte. Berlin: Springer, S. 639-660.

[14] Wolf, I. (2015). Wechselwirkung Mensch und autonomer Agent. In: Maurer, M., Gerdes, J. C., Lenz, B., Winner, H. (Hrsg.) (2015). Autonomes Fahren – Technische, rechtliche und gesellschaftliche Aspekte. Berlin: Springer, S. 103-125.

[15] SAE (2014). Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, SAE J3016.

Der Toleranzansatz als Methodik zur Bewertung der Kontrollierbarkeit höherer Eingriffsstärken bei Falschauslösung eines systeminitiierten Ausweichmanövers mit ESP-Aktorik

Norbert Schneider*, Marcus Schmitz†, Lenne Ahrens‡,
Christian Löffler§ und Alexandra Neukum**

Zusammenfassung: Aktuell gelten für aktiv in die Querführung eingreifende Assistenzsysteme geschwindigkeitsabhängige Grenzwerte, welche die Wirksamkeit im Nutzenfall beschränken. Die Grundlage dieser Grenzwerte bilden Toleranzgrenzen für fahrdynamische Kennwerte, die anhand von subjektiven Bewertungen der Kontrollierbarkeit von Falschauslösungen ermittelt wurden (Toleranzansatz). Um die Wirksamkeit neuer Systeme zu maximieren, sind jedoch höhere Eingriffsstärken und Grenzwerte notwendig. Vorliegende Befunde zur Kontrollierbarkeit von höheren Grenzwerten sind allerdings sehr heterogen und wenig vergleichbar. In einer Realfahrzeugstudie mit N = 65 wurde daher die Anwendung des Toleranzansatzes zur Bewertung der Kontrollierbarkeit überprüft. Die Ergebnisse zeigen einerseits, dass der Toleranzansatz geeignet ist, um höhere Eingriffsstärken zu untersuchen und andererseits, dass unter Berücksichtigung bestimmter Gestaltungsmerkmale höhere Eingriffsstärken toleriert und beherrscht werden können.

Schlüsselwörter: Ausweichmanöver, ESP-Aktorik, Falschauslösung, Kontrollierbarkeit, Untersuchungsmethodik.

1 Einleitung

Aktiv in die Querführung eingreifende Assistenzsysteme könnten zur Vermeidung von Unfällen eingesetzt werden, die bisher (allein) durch Eingriffe in die Längsführung, wie z.B. einen automatischen Notbremseingriff, nicht adressiert werden können.

* Norbert Schneider ist Mitarbeiter des Würzburger Instituts für Verkehrswissenschaften (WIVW) im Bereich Human Factors (schneider@wivw.de)

† Marcus Schmitz ist Mitarbeiter des Würzburger Instituts für Verkehrswissenschaften (WIVW) im Bereich Human Factors (schmitz@wivw.de)

‡ Lenne Ahrens ist Mitarbeiter der Robert Bosch GmbH in der Vorentwicklung des Geschäftsbereich Chassis Systems Control (Lenne.Ahrens@de.bosch.com)

§ Christian Löffler ist Mitarbeiter der Robert Bosch GmbH in der Vorentwicklung des Geschäftsbereich Chassis Systems Control (Christian.Loeffler@de.bosch.com)

** Alexandra Neukum ist Leiterin des Würzburger Instituts für Verkehrswissenschaften (WIVW) (neukum@wivw.de)

Dazu ist, je nach Ausgangsituation, eine hohe Eingriffsdynamik erforderlich, um die Kollision sicher vermeiden zu können. Da aufgrund potenzieller funktionaler Unzulänglichkeiten der Umfeldsensorik und situativer Unsicherheiten nicht zu 100% gewährleistet werden kann, dass Falschauslösungen nicht auftreten, muss sichergestellt werden, dass diese durch den Fahrer übersteuert werden können. Bisher finden sich nur wenige Studien in der Literatur, die sich mit der Frage der Kontrollierbarkeit von aktiv in die Querführung eingreifender Assistenzsystemen im Falle einer Falschauslösung oder höheren Eingriffsstärken im allgemeinen beschäftigen (Bräuchle, Flehming, Rosenstiel & Kropf, 2013; Fricke et al., 2015; Gußner, Bürkle & Marberger, 2015; Hesse et al., 2013; Schieben et al., 2014; Sieber et al., 2015a; Sieber, Siedersberger, Siegel & Färber, 2015b). Allerdings ist die Befundlage insgesamt sehr heterogen. Eine mögliche Ursache hierfür können die bestehenden Unterschiede in der Eingriffs- und Situationsgestaltung sein, sodass die die Ergebnisse der Studien nicht miteinander verglichen werden können. Zwar werden mögliche Einflussfaktoren, wie die Eingriffscharakteristik oder die Wahl des Aktorkonzepts diskutiert (siehe hierzu auch Schneider et al., 2017), allerdings fehlt eine systematische, zusammenfassende Erarbeitung und Beschreibung der Einflussfaktoren sowie eine vergleichbare Untersuchungs- und Bewertungsmethodik. Dies erschwert eine Einschätzung, wie sich die untersuchten Einflussfaktoren auf die Kontrollierbarkeit auswirken.

Bei der Gestaltung der Notausweichassistenten werden in der Regel starke Eingriffe in die Querdynamik vorgenommen, die anhand der aus den Grundlagenuntersuchungen gewonnenen Erkenntnisse als nicht kontrollierbar eingestuft werden müssten. Allerdings unterscheidet sich die Eingriffsgestaltung hinsichtlich Dauer und zeitlichem Verlauf teils deutlich von den Fehlerbildern, die im Rahmen der Grundlagenstudien zur Ermittlung der Grenzwerte untersucht wurden. Zwar lassen sich diesen Studien auch Hinweise darauf entnehmen (vgl. Neukum et al., 2010a; Schmidt, Kiss, Switkes & Gerdes, 2007), dass die Gestaltung der Eingriffscharakteristik einen Einfluss auf die Kontrollierbarkeit hat. Allerdings wurde dies nicht bei der Ermittlung der Grenzwerte berücksichtigt, da diese möglichst allgemeingültig sein sollten. Dies gilt insbesondere für die Dauer der Falschauslösungen und die Gestaltung des zeitlichen Verlaufs, wie z.B. die Anstiegssteilheit, oder die Implementierung einer Übersteuerungsmöglichkeit. Daher besteht grundsätzlich die Möglichkeit, dass je nach Eingriffsgestaltung auch höhere Eingriffsstärken vom Fahrer toleriert und beherrscht werden können. Um die Vergleichbarkeit der Ergebnisse zu gewährleisten, sollte bei der Bewertung jedoch eine möglichst standardisierte Prüfmethodik eingesetzt werden. Aufgrund der hohen Anzahl an verfügbaren Studien, sowie den allgemein anerkannten Ergebnissen, bietet sich daher die von Neukum und Krüger (2003) entwickelte Methodik an, um zu überprüfen ob auch höhere Eingriffsstärken unter gleichen Testbedingungen toleriert und beherrscht werden können.

Zu diesem Zweck wurde eine Realfahrzeugstudie durchgeführt, welche mit Hilfe des Toleranzansatzes überprüfen soll, ob und bis zu welcher Stärke höhere Eingriffsstärken bei Falschauslösung eines systeminitiierten Ausweichmanövers mit ESP-Aktorik kontrollierbar sind.

2 Methode

2.1 Beschreibung des Toleranzansatzes

In bisherigen Studien zur Bewertung der Kontrollierbarkeit von Querdynamikstörungen wurde überwiegend der von Neukum und Krüger (2003) entwickelte Toleranzansatz eingesetzt (vgl. Neukum, 2015; Neukum, Paulig, Frömmig & Henze, 2010a; Neukum, Ufer & Schwab, 2010b; Schmidt, 2009). Dabei werden mit Hilfe des subjektiven Urteils des Fahrers Toleranzgrenzen für fahrdynamische Kennwerte ermittelt. Dafür muss die eingesetzte Befragungsskala verschiedene Voraussetzungen erfüllen, um valide Ergebnisse zu erhalten (vgl. Neukum, 2015). Zentraler Bestandteil des Toleranzansatzes ist die von Neukum und Krüger (2003) entwickelte Störungsbewertungsskala (Abbildung 1; siehe auch Neukum, 2015).

Abbildung 1: Störungsbewertungsskala zur Bewertung von Querdynamikstörungen nach Neukum und Krüger (2003).

Die hierarchisch aufgebaute Kategorienskala verlangt im ersten Schritt eine Einstufung der Störung anhand der Verbalkategorien und im Anschluss eine genauere Differenzierung anhand der numerischen Skalenstufen. Sie zeichnet sich aus durch eine eindeutig definierte Toleranzgrenze: Vom Fahrer nicht mehr tolerierte Störungen sollen mit Skalenwerten ≥ 7 („ gefährlich“) beurteilt werden. Die Verbalkategorien der Skala werden wie folgt erläutert (vgl. Neukum & Krüger, 2003):

- In die Kategorie "spürbar" (Urteile 1-3) fallen Störungen, die vom Fahrer bemerkt werden, die jedoch keine oder nur leichte Auswirkungen auf die Fahraufgabe haben. Verletzt sind im Wesentlichen Komfortanforderungen und kompensatorisches Handeln ist nicht oder nur in geringem Maße erforderlich.
- Die Kategorie "störend" (Urteile 4-6) betrifft Fehler, die wegen auftretender Fahrzeugreaktionen bzw. Spurabweichungen einen deutlichen, aber vom Fahrer als vertretbar eingeschätzten kompensatorischen Aufwand erfordern.
- Der Fehler soll als "gefährlich" (Urteile 7-9) eingestuft werden, wenn der Aufwand zur Kompensation des Fehlers sehr hoch und für den Fahrer nicht mehr tolerierbar ist.

Ein entscheidender Unterschied der Störungsbewertungsskala im Vergleich zu anderen subjektiven Beurteilungsinstrumenten ist, dass die Ausprägung der erlebten Störung im

Vordergrund steht und nicht der Zustand des Fahrers, welcher durch Fragen bezüglich der erlebten Beanspruchung oder der gefühlten Sicherheit erfasst werden soll (vgl. Neukum, 2015; Neukum & Krüger, 2003). Dies erlaubt es, die subjektive Bewertung des Fahrers den objektiven Kennwerten gegenüber zu stellen um zu überprüfen, bei welchen objektiven Kennwerten die subjektive Toleranzgrenze des Fahrers überschritten wurde. Wird ein Überschreiten der subjektiven Toleranzgrenze für einen bestimmten Wertbereich der objektiven Kennwerte festgestellt, dient dies als Indikator für die Festlegung einer objektiven Toleranzgrenze. Hierbei muss jedoch beachtet werden, dass dieses Verfahren in der Regel zu konservativeren Ergebnissen führt als die alleinige Betrachtung objektiver Kennwerte oder auf Basis der Auftretenshäufigkeit von sicherheitsrelevanten Ereignissen (vgl. hierzu auch Neukum, 2015). Im Rahmen verschiedener Studien wurden mit Hilfe des Toleranzansatzes fahrdynamische Grenzwerte für Querdynamikstörungen ermittelt (Gierate von $\leq$ 5°/s bei 50 km/h und $\leq$ 3°/s bei 100-130 km/h). Die Studien belegen auch, dass die ermittelten Grenzwerte grundsätzlich nicht von der Wahl des Aktors beeinflusst werden (vgl. Neukum et al., 2010b). Dies und die Beobachtung, dass Querdynamikstörungen bei der Geradeausfahrt mit beschränkten Platzverhältnissen kritischer beurteilt werden als bei instationären Manövern oder einer Kurvenfahrt (vgl. Neukum, 2015; Neukum et al., 2010b), hat die Bewertung der Kontrollierbarkeit von Querdynamikstörungen insgesamt deutlich vereinfacht und führte zu den oben genannten, allgemein anerkannten Grenzwerten. Im Folgenden wird dieser Ansatz herangezogen, um die Kontrollierbarkeit von fehlerhaften Eingriffen mittels einer ESP-Aktorik bei höheren Eingriffsstärken (> 3°/s bei 100 km/h) zu bewerten.

2.2 Studienaufbau

Als Prüfszenario wurde eine Gasse mit einer Breite von 2.75 Meter gewählt, die mit einer Geschwindigkeit von 100 km/h durchfahren wurde. Die Breite der Gassen orientiert sich an der Richtlinie zur Anlage von Straßen und Autobahnen (RAS-Q). Die gewählte Gassenbreite betrug 2,75 Meter für Geschwindigkeiten von bis zu 100 km/h und 3,00 Meter für Geschwindigkeiten bis zu 120 km/h. Die Länge der aufgestellten Gassen betrug ca. 150 Meter, bei einem Längsabstand von ca. 5,00 Meter zwischen den Pylonen (vgl. Abbildung 2). Die erste Falschauslösung mit einer Zielgierrate von 6°/s erfolgte innerhalb der Gasse, ohne dass der Proband vorab über den Inhalt der Untersuchung aufgeklärt wurde.

Abbildung 2: Aufbau Prüfszenarien auf der Teststrecke.

Um sicherzustellen, dass die Probanden bei Gasseneinfahrt bereits die gewünschte Geschwindigkeit erreicht haben, wurde eine ca. 400 Meter lange Beschleunigungszone eingerichtet.

Zusätzlich wurde auf ausreichend große Sicherheitsbereiche links und rechts der Pylonengassen geachtet sowie auf eine entsprechende Auslaufzone (vgl. Abbildung 3).

Abbildung 3: Schematische Darstellung des Aufbaus der Prüfszenarien auf dem Prüfgelände.

Als Versuchsfahrzeug wurde ein Honda Acura RLX eingesetzt (vgl. Abbildung 4). Die Breite des Fahrzeugs betrug 1.89 Meter (ohne Außenspiegel). Die Auswahl und Auslösung des Fehlerbilds erfolgte durch den Versuchsleiter an zufälligen Positionen innerhalb der Pylonengasse.

Abbildung 4: Versuchsfahrzeug innerhalb der Pylonengasse (links, Mitte) und Innenansicht inklusive des Bildschirms zur Auswahl und Auslösung des Fehlerbilds durch den Versuchsleiter (rechts).

Im Rahmen von Vorversuchen wurden verschiedene Eingriffsparametrierungen erprobt, sowie eine Möglichkeit zur Ansteuerung der ESP-Aktorik entwickelt, welche es erlaubt die Zielgierrate als Parameter vorzugeben. Zusätzlich wurde eine Möglichkeit zur Erkennung der Übersteuerungsintention mit Hilfe von lenkradwinkelgeschwindigkeits-basierten Schwellwerten implementiert, welche es erlauben das System zu degradieren und damit die Kontrollierbarkeit zu erhöhen.

2.3 Studiendurchführung

Die externen Studienteilnehmer wurden vorab nicht über den Zweck der Studie aufgeklärt. Ihnen wurde zunächst mitgeteilt, dass sie an einer Studie zum Thema „automatisiertes Fahren“ in Engstellen teilnehmen, um den experimentellen Aufbau zu plausibilisieren.
Vor Versuchsbeginn hatten die Probanden 10 Minuten Gelegenheit sich mit dem Fahrzeug und der Fahraufgabe vertraut zu machen. Dazu wurden die Probanden instruiert, die aufgebaute Gasse mit schrittweise erhöhter Geschwindigkeit so lange zu durchfahren, bis die Zielgeschwindigkeit von 100 km/h erreicht wurde und die Probanden dazu in der Lage waren, die Gasse einhändig zu durchfahren. Danach wurden die Probanden instruiert, die Gasse nochmals mit 100 km/h zu durchfahren. Während der Durchfahrt erfolgte die erstmalige Aufschaltung des Fehlerbilds (Erstkontakt).

Anschließend wurden die Probanden gebeten, die erlebte Situation zu beschreiben und den Eingriff anhand der Störungsbewertungsskala nach Neukum und Krüger (2003a) zu bewerten. Im weiteren Verlauf der Untersuchung wurden zusätzliche experimentelle Variationen der Fehleraufschaltung durchgeführt, sodass die Fahrer insgesamt 21 Aufschaltungen eines Fehlerbilds erlebten. Hierbei wurde zudem die Aufschaltung aus dem Erstkontakt wiederholt, um auf mögliche Lerneffekte prüfen zu können. Der gesamte Versuch dauerte ca. 90 Minuten. Insgesamt nahmen N = 65 Fahrer an der Studie teil (davon 30 weiblich und 35 männlich). Das Alter der Fahrer variierte zwischen 18 und 80 Jahren (M = 43.54).

3 Ergebnisse

Von insgesamt N = 65 Fahrern, die an der Studie teilnahmen, wurden N = 61 für die Auswertungen berücksichtigt. Vier Fahrer wurden von den Auswertungen ausgeschlossen, da sie entweder Probleme bei der Umsetzung der Fahraufgabe hatten (z.B. unfallfreies Durchfahren der Pylonengasse ohne Fehleraufschaltung, Erreichen und Einhalten der vorgegebenen Zielgeschwindigkeit) oder die Instruktionen des Versuchsleiters nicht korrekt befolgten.

3.1 Kontrollierbarkeitsbewertung Erstkontakt

Beim erstmaligen Erleben des Fehlerbilds (Erstkontakt) bewertete bereits bei einer initialen Gierrate von 4.5°/s ein Fahrer (13% der Fälle bei 4.5°/s) den fehlerhaften Eingriff als „gefährlich“. Weitere „gefährlich“ Urteile wurden erst ab einer initialen Gierrate von 6.5°/s beobachtet (siehe Abbildung 5). Keine der „gefährlich“ Bewertungen im Erstkontakt lässt sich auf eine nicht erfolgte Abschaltung zurückführen. Im Mittel bewerteten die Fahrer den Eingriff als „störend“ (M = 4.25), wobei die Spannbreite der Urteile zwischen „spürbar“ (1-3) und „gefährlich“ (7-9) liegt (siehe Abbildung 5).

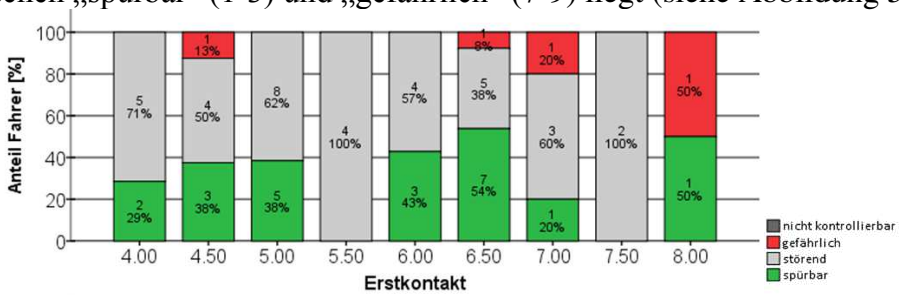


Abbildung 5: Häufigkeitsverteilung der subjektiven Urteile entsprechend der Skalenkategorien in Abhängigkeit von der initial erreichten Gierrate im Erstkontakt.

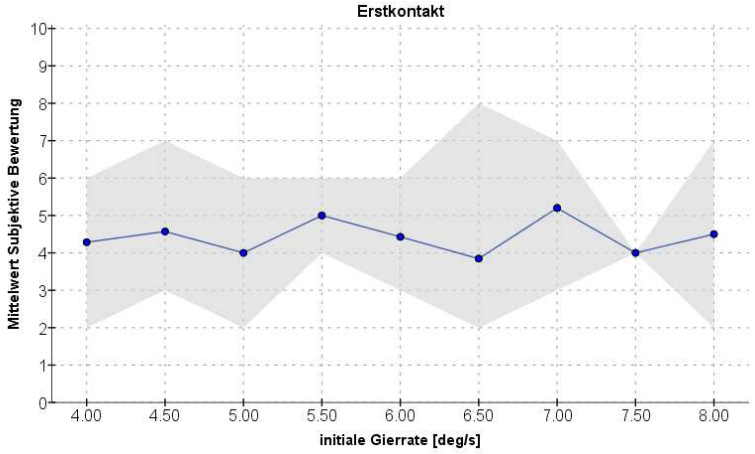


Abbildung 6: Mittelwert (blau) und Streuungsbereich (grau) der subjektiven Urteile in Abhängigkeit von der initial erreichten Gierrate im Erstkontakt. Eingriffe mit nicht erfolgter Abschaltung werden bei der Darstellung nicht berücksichtigt. Für die Darstellung wurden die tatsächlich erreichten initialen Gierraten post-hoc klassifiziert und den entsprechenden Kategorien zugeordnet.

Bei einer Gierrate von 8°/s wurde zudem die vorgegebene Pylonengasse verlassen und es traten Pylonenfehler auf. Einzelfallüberprüfungen kommen daher sowohl für den Fall bei 4.5°/s als auch bei 6.5°/s in Betracht, um zu überprüfen, ob ein ungerechtfertigtes Urteil vorliegt oder nicht (vgl. Neukum, 2015). Bei Gierraten ab 7°/s liegen zu wenige Fälle vor und der Anteil an „gefährlich“ Urteilen überschreitet den Grenzwert von 15% sodass eine Einzelfallprüfung für diese Fälle nicht in Betracht kommt (vgl. Neukum, 2015). Die Einzelfallüberprüfung nach Neukum (2015) ergab bei einer Gierrate von 4.5°/s und 6°/s für beide Fälle keine auffälligen fahrdynamischen Kennwerte. Sowohl die Kompensationsgierrate, als auch das Lenkverhalten liegen innerhalb der Kennwertverteilung von Fällen, in denen keine „gefährlich“ Bewertung abgegeben wurde. Daher wird in beiden Fällen davon ausgegangen, dass es sich bei dem Urteil um ein „ungerechtfertigtes“ Urteil handelt. Dies ist möglicherweise darauf zurückführen, dass die Fahrer sich erschreckt haben und dies, entgegen der Instruktion, mit in ihre Bewertung einbezogen haben. Daher werden beide Fälle als kontrollierbar (Pass) bewertet. Die ermittelte Toleranzgrenze im Erstkontakt liegt damit bei 6.5°/s (vgl. Abbildung 7).

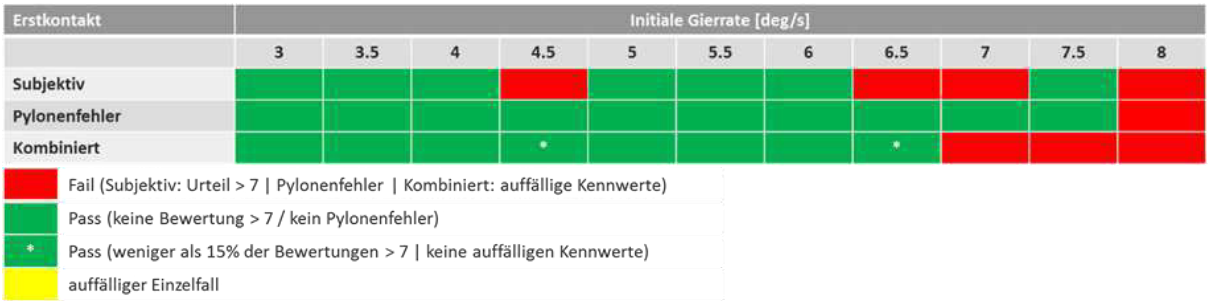


Abbildung 7: Ergebnis der finalen Kontrollierbarkeitsbewertung für den Erstkontakt entsprechend des von Neukum (2015) beschriebenen Bewertungsvorgehens.

3.2 Kontrollierbarkeitsbewertung Wiederholung

In der Wiederholung wurden Eingriffe ab einer Gierrate von 4.5°/s als „gefährlich“ bewertet. Weitere „gefährlich“ Urteile liegen bei Gierraten von 5°/s, 6°/s und 6.5°/s vor. Die „gefährlich“ Urteile bei Gierraten von 4.5°/s und 5°/s lassen sich auf eine nicht erfolgte Abschaltung zurückführen, während zwei der „gefährlich“ Urteile bei 6.5°/s (entspricht einem Anteil von 18% der verbleibenden Fälle bei 6.5°/s; siehe Abbildung 8) und ein weiteres Urteil bei 6°/s trotz erfolgter Abschaltung auftraten. Im Mittel bewerteten die Fahrer den Eingriff als „störend“ ($M = 4.39$; siehe Abbildung 9) mit einer Spannbreite der Urteile zwischen „spürbar“ (1-3) und „gefährlich“ (7-9).

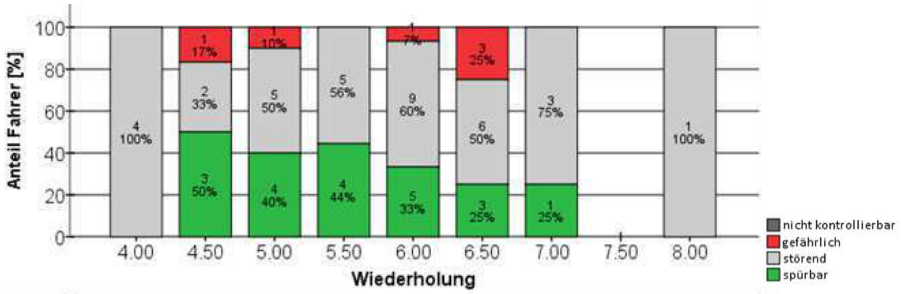


Abbildung 8: Häufigkeitsverteilung der subjektiven Urteile entsprechend der Skalenkategorien in Abhängigkeit von der initial erreichten Gierrate im Erstkontakt.

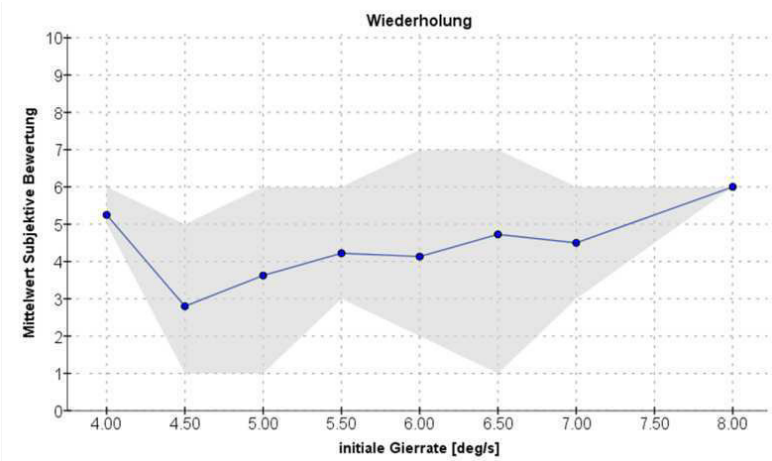


Abbildung 9: Mittelwert (blau) und Streuungsbereich (grau) der subjektiven Urteile in Abhängigkeit von der initial erreichten Gierrate in der Wiederholung. Eingriffe mit nicht erfolgter Abschaltung werden bei der Darstellung nicht berücksichtigt. Für die Darstellung wurden die tatsächlich erreichten initialen Gierraten post-hoc klassifiziert und den entsprechenden Kategorien zugeordnet.

Der Grenzwert für den Anteil an „gefährlich“ Urteilen, bei dem eine Einzelfallüberprüfung noch in Betracht kommt, wird bereits bei einer initialen Gierrate von 6.5°/s überschritten. Zusätzlich wurde bereits bei einer initialen Gierrate von 6°/s ein Verlassen der Pylonengasse beobachtet (ohne „gefährlich“ Bewertung durch den Probanden) und ein „gefährlich“ Urteil abgegeben. Das Verlassen der Pylonengasse lässt sich mutmaßlich auf die schwache und späte Kompensationsreaktion des Fahrers zurückführen.

In Folge dessen wurde die Abschaltung erst nach 0.8 Sekunden und damit zum Zeitpunkt der Umschaltung ausgelöst. Eine vergleichbar späte und schwache Reaktion wurde im Erstkontakt nicht beobachtet und der Fahrer war hier in der Lage eine initiale Gierrate von 6.5°/s fehlerfrei zu kompensieren. Daher wird davon ausgegangen, dass der Fahrer grundsätzlich dazu in der Lage war, einen entsprechenden fehlerhaften Eingriff zu beherrschen. Die Überprüfung der fahrdynamischen Kennwerte für den Fall mit „gefährlich“ Urteil ergab keine Auffälligkeiten. In Folge dessen liegt der ermittelte Toleranzwert in der Wiederholung bei 6°/s (siehe Abbildung 10)

Wiederholung	Initiale Gierrate [deg/s]										
	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8
Subjektiv											
Pylonenfehler											
Kombiniert							*				

Fail (Subjektiv: Urteil > 7 | Pylonenfehler | Kombiniert: auffällige Kennwerte)
Pass (keine Bewertung > 7 / kein Pylonenfehler)
* Pass (weniger als 15% der Bewertungen > 7 | keine auffälligen Kennwerte)
auffälliger Einzelfall

Abbildung 10: Ergebnis der finalen Kontrollierbarkeitsbewertung für die Wiederholung entsprechend des von Neukum (2015) beschriebenen Bewertungsvorgehens.

3.3 Vergleich von Erstkontakt und Wiederholungsmessung

Im Folgenden wird überprüft, ob die subjektive Bewertung der Probanden Lern- oder Gewöhnungseffekten in Folge des mehrfachen Erlebens des Fehlerbilds unterliegt. Hierzu werden die Ergebnisse von Erstkontakt und Wiederholung miteinander verglichen, wobei nur Eingriffe mit erfolgter Abschaltung berücksichtigt werden. Voraussetzung hierfür ist, dass kein systematischer Unterschied der initialen Gierrate zwischen Erstkontakt und Wiederholung besteht. Besteht ein Unterschied ist ein direkter Vergleich der subjektiven Bewertungen nicht ohne weiteres möglich. Im Erstkontakt betrug die initiale Gierrate im Mittel 5.61°/s ($SD = 1.09$) und in der Wiederholung 5.74°/s ($SD = 0.09$; vgl. Abbildung 11). Es gab keinen signifikanten Unterschied (t-Test) zwischen der initialen Gierrate beim erstmaligen Erleben und beim wiederholten Erleben des Fehlerbilds ($t(54) = -0.856$, $p = .396$). Somit ist die Voraussetzung erfüllt, dass kein systematischer Unterschied der initialen Gierrate zwischen Erstkontakt und Wiederholung besteht.

Im Erstkontakt wurde das Fehlerbild im Mittel mit $M = 4.29$ ($SD = 1.51$) bewertet, in der Wiederholung mit $M = 4.24$ ($SD = 1.55$). Es gab keinen signifikanten Unterschied (t-Test) der subjektiven Bewertung zwischen erstmaligem und wiederholtem Erleben des Fehlerbilds ($t(54) = .277$, $p = .783$). Somit liefert der Vergleich zwischen Erstkontakt und Wiederholung keine Hinweise auf einen Einfluss möglicher Lern- oder Gewöhnungseffekte auf die Bewertung der Fahrer.

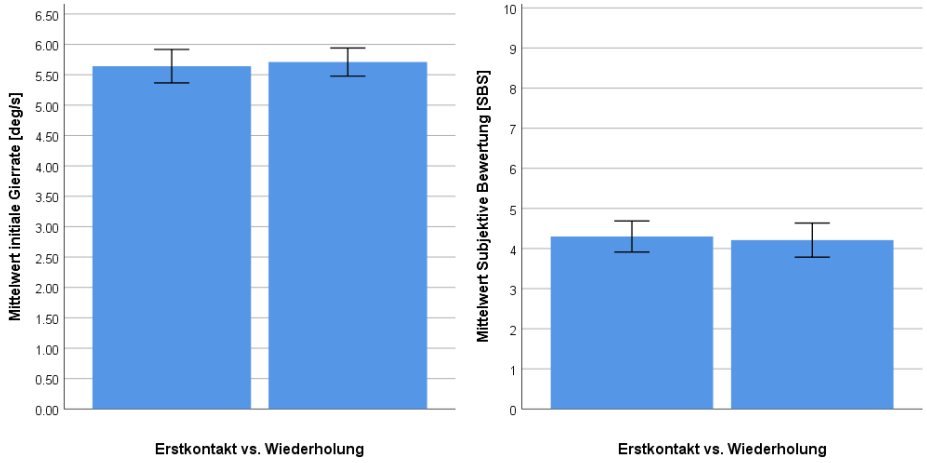


Abbildung 11: Mittelwert der initialen Gierrate für Erstkontakt und Wiederholung (links) und Mittelwert der subjektiven Bewertung für Erstkontakt und Wiederholung (rechts). Fehlerbalken entsprechen dem 95%-Konfidenzintervall. Für die Berechnung wurden nur Eingriffe mit erfolgter Abschaltung berücksichtigt. Fahrer mit fehlenden Messungen (Erstkontakt o. Wiederholung) wurden in der Darstellung ebenfalls nicht berücksichtigt.

4 Diskussion

Die Ergebnisse der Studie zeigen, dass der von Neukum und Krüger (2003) entwickelte Toleranzansatz auch bei höheren Eingriffsstärken eingesetzt werden kann, um einen objektiven Grenzwert für fehlerhafte Eingriffe mit Hilfe subjektiver Maße zu ermitteln. In diesem Fall konnte gezeigt werden, dass höhere Eingriffsstärken von bis zu 6.5°/s im Erstkontakt von den Probanden toleriert und beherrscht werden. Dies gilt unter der Voraussetzung einer zuverlässig erkannten Übersteuerungsintention des Fahrers und einer daraufhin erfolgten Abschaltung des Eingriffs.

Hierbei ist zu beachten, dass subjektiv ermittelte Grenzwerte in der Regel konservativer ausfallen als Grenzwerte, welche allein anhand objektiver Maße ermittelt werden. Dies stellt nicht zwingend einen Nachteil dar, da für die Grenzwertermittlung aus ökonomischen Gründen in der Regel nur kleine Stichproben (20-30 Fahrer pro Fehlerbild) untersucht werden. Die untersuchten Fahrer stellen damit nur einen Teil der Fahrerpopulation dar, wodurch die tatsächlich zu erwartenden Streubreite der Reaktionen nur teilweise abgedeckt werden kann. Mit Hilfe des Toleranzsansatzes werden jedoch objektive und subjektive Maße gegenübergestellt, um einen Grenzwert zu ermitteln, welchem auch die subjektive Einschätzung der eigenen Fahrerfertigkeiten zu Grunde liegt. Dadurch wird die ausbleibende Streubreite der Reaktionen zu einem gewissen Maß kompensiert.

Allerdings müssen bei der Anwendung des Toleranzansatzes zur Ermittlung fahrdynamischer Grenzwerte verschiedene Aspekte beachtet werden. Zum einen sind die Instruktion und die darin enthaltene Vorgabe der Verankerung der Befragungsskala entscheidend für die korrekte Anwendung.

Erfolgt diese nicht oder nur unvollständig, verankern die Fahrer die Items der Befragungsskala möglicherweise unterschiedlich, wodurch die Vergleichbarkeit und die Interpretierbarkeit der Subjektivurteile nicht mehr gewährleistet sind. Zum anderen muss bei der anschließenden Grenzwertermittlung eine detaillierte Prüfung der resultierenden fahrdynamischen Kennwerte erfolgen. Damit soll einerseits sichergestellt werden, dass die unterschiedlichen Fahrer tatsächlich vergleichbare Fehlerbilder erlebt haben (insbesondere bei Eingriffen in die Querführung kann es hier trotz identischer Soll-Anforderungen zu Abweichungen bei den erreichten Werten kommen). Andererseits bildet es die Grundlage für die Überprüfung, ob vereinzelte „gefährlich“ Urteile durch die aufgetretenen Fahrzeugreaktionen erklärt werden können oder ob es sich um „ungerechtfertigte“ Urteile handelt (vgl. Neukum, 2015). Werden diese Aspekte beachtet, stellt das standardisierte Vorgehen des Toleranzansatzes ein gutes Instrument zur Bewertung der Kontrollierbarkeit dar, welches auch bei höheren Eingriffsstärken eingesetzt werden kann. Zudem wird bei dem Vergleich des subjektiven Urteils zwischen Erstkontakt und Wiederholungsmessung deutlich, dass das subjektive Urteil der Fahrer bei vergleichbaren Störgrößen keinen Lern- oder Gewöhnungseffekten unterliegt. Dies ermöglicht zum einen effiziente Prüfanordnungen, mit deren Hilfe auch weitere Einflussfaktoren untersucht werden können. Zum anderen kann die Datenbasis für die Festlegung von fahrdynamischen Grenzwerten vergrößert werden. Ein entscheidender Vorteil bei der Anwendung des Toleranzansatzes zur Ermittlung fahrdynamischer Grenzwerte bei höheren Einsatzstärken ist zudem, dass die Vergleichbarkeit mit bereits vorliegenden Studien gewährleistet ist. Somit ist sichergestellt, dass bei der Ermittlung von neuen fahrdynamischen Grenzwerten die Ergebnisse nicht durch die eingesetzte Methodik beeinflusst werden.

Literatur

[1] Bräuchle, C., Flehming, F., Rosenstiel, W. & Kropf, T. (2013). Das Fahrerverhalten als Indikator für Fehlauslösungen von aktiven Kollisionsvermeidungsmanövern. VDI-Berichte, 2205, 3-14.

[2] Fricke, N., Griesche, S., Hesse, T., Schieben, A. & Baumann, M. (2015). Driver behavior following an automatic steering intervention. Accident Analysis & Prevention, 83, 190–196.

[3] Gußner, T., Bürkle, L. & Marberger, C. (2015). Erweiterung aktiver Fußgängerschutzsysteme durch eine fahrerinitiierte Ausweichunterstützung. 10. Uni-DAS e.V. Workshop Fahrerassistenzsysteme. Walting.

[4] Hesse, T., Schieben, A., Heesen, M., Dziennus, M., Griesche, S. & Köster, F. (2013). Interaction design for automation initiated steering manoeuvres for collision avoidance. 6. Tagung Fahrerassistenz. München.

[5] Neukum, A. (2015). Beherrschbarkeit fehlerhafter Eingriffe in die Fahrzeugquerdynamik. In K. Kompaß (Hrsg.), Fahrerassistenz und Aktive Sicherheit. Haus der Technik Fachbuch. 137, S. 122-138. Renningen: expert verlag.

[6] Neukum, A. & Krüger, H. (2003). Fahrerreaktionen bei Lenksystemstörungen– Untersuchungsmethodik und Bewertungskriterien. VDI-Berichte, 1791, 297-318.

[7] Neukum, A., Paulig, J., Frömmig, L. & Henze, R. (2010a). Untersuchung zur Wahrnehmung von Lenkmomenten bei Pkw. FAT-Schriftenreihe (222).

[8] Neukum, A., Ufer, E. & Schwab, M. (2010b). Kontrollierbarkeit potenzieller Fehleingriffe des elektronischen Stabilitätsprogramms (ESP). In Ika (Hrsg.), 7. Tag des Fahrwerks 2010. S. 108–123.

[9] Schieben, A., Griesche, S., Hesse, T., Fricke, N. & Baumann, M. (2014). Evaluation of three different interaction designs for an automatic steering intervention. Transportation Research Part F: Traffic Psychology and Behaviour, 27, Part B, 238-251.

[10] Schmidt, G. (2009). Haptische Signale in der Lenkung: Controllability zusätzlicher Lenkmomente. Dissertation, TU Braunschweig. http://rzbl04.biblio.etc.tu-bs.de:8080/docportal/servlets/MCRFileNodeServlet/DocPortal_derivate_00007645/schmidtg_dissertation.pdf

[11] Schmidt, G., Kiss, M., Switkes, J. & Gerdes, C. (2007). Auswirkungen haptischer Signale in der Lenkung auf das Fahrer-Fahrzeug System. AAET.

[12] Schneider, N., Berg, G., Paradies, S., Zahn, P., Huesmann, A. & Neukum, A. (2017). Designing Emergency Steering and Evasion Assist to Enhance Safety in Use and

[13] Controllability. In K. Bengler, S. Hoffmann, D. Manstetten, A. Neukum & J. Drüke (Eds.), UR:BAN Human Factors in Traffic Approaches for Safe Efficient and Stressfree Urban Traffic. Wiesbaden, Germany: Springer Vieweg.

[14] Sieber, M., Schneider, M., Siegel, A., Siedersberger, K.-H. & Färber, B. (2015a). Wirksamkeit von Ausweichempfehlungen in Kombination mit Teilbremsung beim Notausweichen. In C. C. Stiller, K. Bengler, K. Dietmayer, L. Eckstein, B. Färber, M. Maurer & H. Winner (Eds.), 10. Workshop Fahrerassistenzsysteme - FAS 2015. S. 39–49. Darmstadt: Uni-DAS e.V.

[15] Sieber, M., Siedersberger, K.-H., Siegel, A. & Färber, B. (2015b). Automatic Emergency Steering with Distracted Drivers: Effects of Intervention Design. In 2015 IEEE 18th International Conference on Intelligent Transportation Systems. S. 2040–2045. Los Alamitos, California: IEEE Conference Publishing Services.

[16]

Cooperative Approach to Overcome Automation Effects During the Transition Phase of Conditional Automated Vehicles

Burak Karakaya* Luis Kalb* Klaus Bengler*

Abstract: One of the safety critical problems for conditional automated vehicles appears during the transition phase. Whenever the system is about to reach its boundaries the take-over request triggers the transition of the driving task to the driver. This transition is often designed by simply switching off the automation. Therefore, this paper proposes to design the transition phase cooperatively in order to establish a higher driving performance throughout the whole process. Further, recommendations for a proper cooperative transition phase are given.

Key words: cooperation, conditional automation, take-over, transition

1 Introduction

The indispensable technological progress affects todays series vehicles and emerges space for innovations. As it can be seen on the future projects of many original equipment manufacturers (OEMs), e.g. Audi [1], BMW [5, 16], Daimler [8], etc., this development is used as a chance to improve driver assistance systems which strides towards automated driving. On this path, the drivers' role is changing and raises new questions to human factor engineers [6, p. 687]. Between these two extremes, i.e manual and automated driving or in other words: full driving task and no driving task at all, the human driver cannot be fully excluded. There is still the need for a driver to monitor the environment up to a certain level entailing known problems with automated systems from the aviation sector, such as loss of skill, higher workload, loss of situation awareness, etc. [10, 19]. To classify the tasks in vehicles with increasing automation the Society of Automotive Engineers defined different levels of automated driving [17]. Current series vehicles are already capable of assisted driving which corresponds to the levels 1 and 2. Therefore, the focus on research is on the upper levels and especially the level 3 systems, which is further referred to as conditional automated driving (CAD). At this level, the driver is allowed to perform non-driving related tasks but is still the fallback level of performance if the automated system is unable to handle the situation. The driver needs a certain time budget to take over the driving task which is initiated by a take-over request (TOR) and restricted by the capability of the automated system and its sensors to assess the present situation. Within this short period of time the driver has to safely resume control and perform a maneuver that the automated system was not able to. Because of the named

*Chair of Ergonomics at the Technical University of Munich, Boltzmannstr. 15, 85748 Garching (e-mail addresses: burak.karakaya@tum.de, luis.kalb@tum.de and klaus.bengler@tum.de)

automation effects the goal is to ensure a good take-over performance. Therefore, this paper proposes a different approach to design the transition phase compared to common studies in this field.

2 Transition process

Whenever a level 3 automated vehicle is about to reach its system boundaries, a TOR is triggered to transfer the driving task to the driver. Figure 1 shows a schematic course of the driving performance during this transition process. Percentage values are only qualitative and represent a drive which obeys the traffic rules and without mistakes that could lead to an accident (100 %) and no driving performance at all (0 %). The automation ideally performs the driving task with 100% efficiency until the TOR is triggered. At this point, the transition of the driving task is often designed as a simple switch as shown in Figure 2. So either the automation or the driver is performing which is why the driving performance of the automation is reduced to 0% at the TOR. But concentrating mainly on secondary tasks and not being responsible to monitor the environment decouples the human from the control loop, also known as the out-of-the-loop performance problem [11]. These automation effects do not form a reliable basis to take over with high driving task performance and accordingly the starting point of the driver is marked very low in Figure 1. The further course to regain 100% efficiency is unknown and depending on the drivers individual characteristics and state, wherefore the end could be reached either before or after the system boundary.

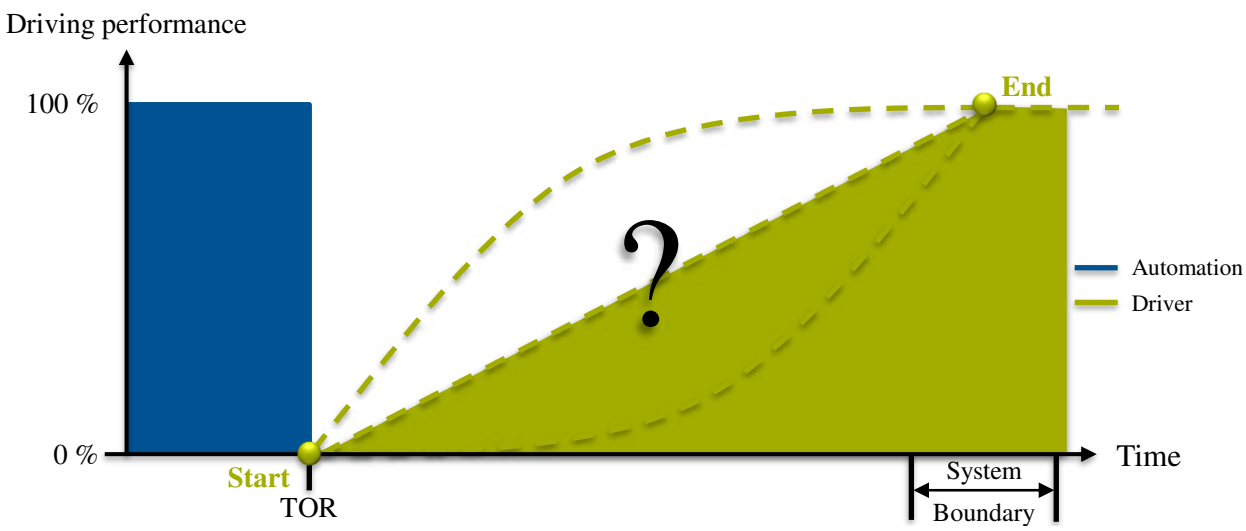


Figure 1: Schematic course of the driving performance during the transition process

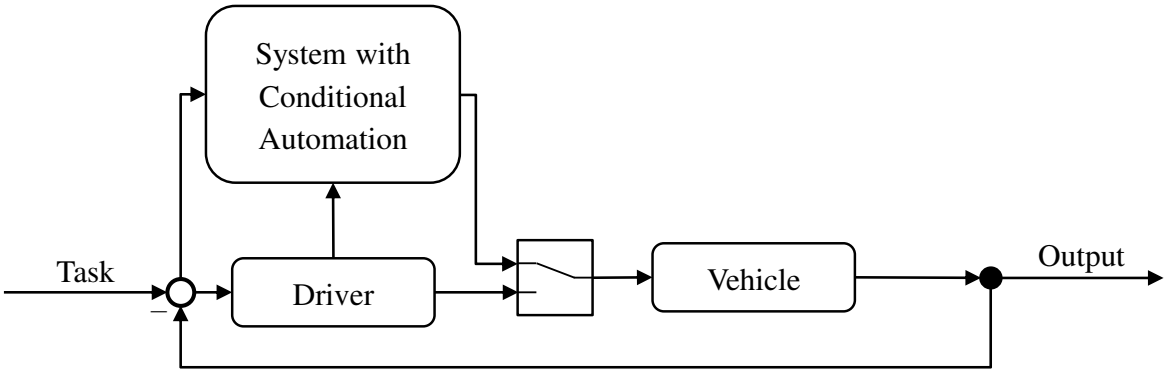


Figure 2: Block diagram of a drive with assistance systems. Adapted from Kienle [14]

The block diagram as shown in Figure 2 is not suitable for the transition phase because of the critical situation that comes along with a TOR. Figure 1 shows the resulting course of the driving performance because of the applied method of a switch. Yet, the drivers course provides great potential to reach 100% of driving performance. Therefore, this paper proposes a different approach of the driving task allocation for critical situations such as the take-over (see Fig. 3). In more specific terms, the idea is to replace the switch in critical situations by a summing point where both, the automation and the driver, act together towards the common goal (or in other words: cooperate) of safe take-over. Before and after a TOR the transition of the driving task can be designed by a simple switch since the criticality in those situations is lower and the driver does not necessarily need additional support. The ideal course of the driving task performance after this adjustment is given in Figure 4. The transition process is classified in two phases, the transition phase (TP) 1 and 2, divided by the hands on steering wheel time. The general approach to establish 100% driving performance for the whole process is addressed during the TP 1 only by the automated system since the driver is obviously not intervening in the driving task yet. After having the hands on the steering wheel, i.e. TP 2, the automation is supporting the driver, e.g. by assisting the maneuver decision and execution.

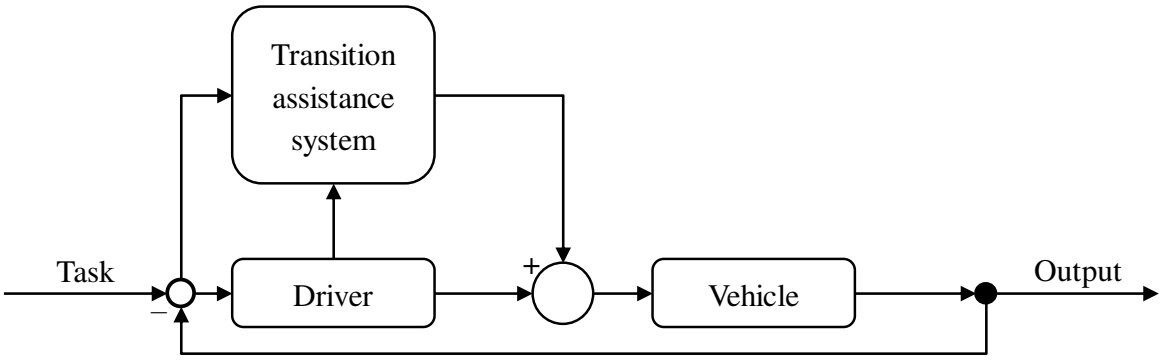


Figure 3: Block diagram of the cooperative transition phase

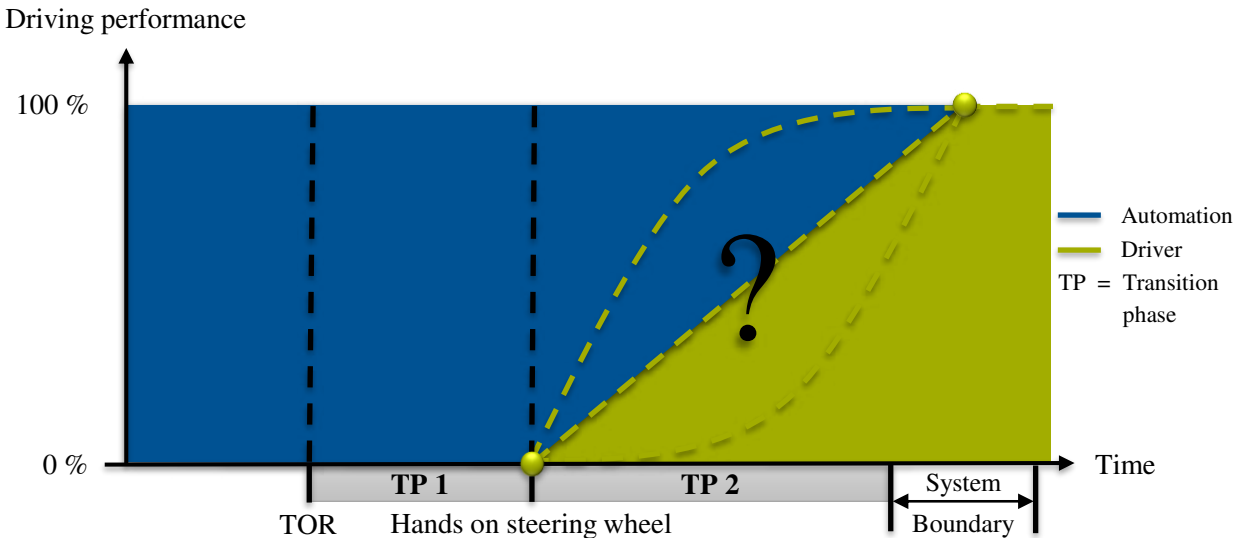


Figure 4: New approach applied on the schematic course of the driving performance during the transition process

3 Cooperative transition

3.1 General definition of cooperation

Before proceeding with redesigning the transition phase via cooperation between driver and automated system it is essential to define the term itself. Different application fields may have distinct views on cooperation. Yet, a domain independent definition was formulated by Hoc [13] as follows:

Two agents are in a cooperative situation if they meet two minimal conditions.
(1) Each one strives towards goals and can interfere with the other on goals, resources, procedures, etc.
(2) Each one tries to manage the interference to facilitate the individual activities and/or the common task when it exists.
The symmetric nature of this definition can be only partly satisfied.

Agents in this case can be two or more human or artificial agents [13]. In other fields such as the human-robot interaction the interaction is classified by four criteria, i.e. *working time*, *workspace*, *aim* and *contact* [18]. Here it is called *Coexistence* if the human and robot share the resources *working time* and *workspace* but they do not inevitably have to have the same *aim*. In turn, if the *aims* are matching it is classified as *Cooperation* and if additionally a direct contact is present it is characterized as *Collaboration* [18]. However, during the proposed transition phase from CAD to manual driving with the automated system actively supporting the driver the difficulty becomes clear. This support occurs by accessing the same resources as the driver and although it should strive towards the same goal of a safe take-over, the drivers expectation of it can be fairly different. This is why the aim of both sides do not have to match. Hence, this process of the automated system and human driver solving the critical situation of a take-over is labeled as *Cooperation*.

3.2 Designing the transition process with cooperation

The idea of designing systems where humans interact with automation in a cooperative manner is not new. Especially during the recent years with increasing interest in automated vehicles, questions arose about the way to do so adequately in the automotive sector. It is important to underline here that this paper is focusing on the cooperation during the transition process (see chapter 2) and not for the whole drive. An example for the latter is the project *H-Mode* which was funded by the German Research Foundation (DFG) [2]. Various recommendations to design cooperative systems can be found in literature, e.g. Flemisch et al. [12] named four cornerstones for cooperative control situations, i.e. *ability*, *authority*, *control* and *responsibility*. Biester [4, p. 11] gives an overview of the 15 most important characteristics of cooperation in human-automobile interaction. Considering that during cooperative situations the automated system and driver are handling the situation together, the automation can or should be seen as a team player. Therefore, Walch et al. [20] suggests four basic requirements for driver and automated system that have to be fulfilled in order to become an effective team player, i.e. *Mutual predictability*, *Directability*, *Shared situation representation* and *Calibrated trust in automation* [20, p. 7]. Further aspects to consider can be found in Klein et al. [15] and Christoffersen and Woods [7]. Bengler et al. [3] structured the most important aspects of cooperation for human-machine interaction and summarized it under the term of *Layers of cooperation*, i.e. *Intention*, *Mode of cooperation*, *Allocation*, *Interface* and *Contact*. These layers are considered as "requirements for successful cooperation" [3, p. 6]

All of the mentioned recommendations were considered for the approach of designing a cooperative transition process. Nevertheless, only those who are important for the transition process are given in the following together with recommendations to implement them. Since many of them are overlapping in their meaning similar ones were summarized as one.

Intention *or* **(Mutual) Predictability.** For intention inference it is necessary to determine on which level the cooperative activity is happening. Hoc [13] differentiated between *action*, *plan* and *meta* levels which can be allocated respectively to the three levels of driving task by Donges [9]: *stabilization*, *guidance* and *navigation*. In the transition phase the main encountered and most important level for intention inference and mutual predictability is the *plan* level because of an upcoming maneuver, wherefore a *shared guidance* system needs to be designed. Because of a time critical event, the intention inference needs to only consider the plan level contrary to the recommendation from Bengler et al. [3] to regard all three. In general, the automated system should execute its intention in TP 2, when the driver has at least the hands on the steering wheel. Therefore, the TP 1, right after the driver receives the TOR, is suitable to communicate the intention of both agents. A confirmation channel for the driver, e.g. through the human machine interface (HMI), is required and recommended [3].

Shared situation representation *or* **Common knowledge base.** The situation representation for both, human and machine, has not mandatory to be the same due to many reasons, e.g. perception of information because of different sensory systems, distinct information processing, etc. [20]. This aspect gets especially important when the driver needs to understand and predict the automated systems actions. Hence, whenever the system communicates its intention it should also communicate its view on the current situation

in order to avoid confusion.

Authority *or* **Allocation of tasks & responsibilities.** Per definition of the automation level 3 by SAE [17] the driver has to take over the driving task in case of a TOR. To be in accordance with this, the drivers' input should always have the highest priority during the TP.

Control *or* **Directability.** The concept of *Control* [12] and *Directability* [7, 15, 20] share the idea that the controlling entity has the ability to shape the course of events. During a cooperative TP, which is solely a time and safety critical situation, both control entities will perform an action. Thus, the driver should be able to control or direct the actions of the automated system. The other way round, i.e. the automation overruling the drivers' input, has to be avoided due to the higher authority of the driver (see "Authority *or* Allocation of tasks & responsibilities").

Interface. The interface has to be designed in a way that a continuous communication is established which is recommended to be multi-modal [3, 4].

Contact. Since the TOR is a time critical situation and a contact is not only desired but crucial the interaction has to be ideally developed at vehicle control devices that are required to handle the situation, e.g. the steering wheel and pedals.

4 Conclusion and future research

Most authors developed their concepts and definitions for a cooperation between a human/driver and a technical system/vehicle for continuous interaction. Yet the proposed interaction in this paper aims at transferring the system's capabilities, including full knowledge about the situation and full control over all resources necessary for the driving task, to the human driver. This poses the question, if existing definitions, i.e. mode of cooperation, need additional stages that focus on the transition between the existing ones. Evaluation of the proposed transition phases will have to deal with the major influence of time on the cooperation. As time is limited, some processes, like negotiations about the system's influence on a control element, might be skipped, leading to the question of how the system's automatic retreat should be designed. It is also yet unknown, whether or not drivers are even willing to share the driving task in such a critical situation. Finally, subjective measurements of driver acceptance should not be neglected as they are crucial for the future implementation of cooperative transitions.

Bibliography

[1] Audi AG. Der neue Audi A8 – hochautomatisiertes Fahren auf Level 3 — Audi MediaCenter, 2017. URL `https://www.audi-mediacenter.com/de/per-autopilot-richtung-zukunft-die-audi-vision-vom-autonomen-fahren-9305/der-neue-audi-a8-hochautomatisiertes-fahren-auf-level-3-9307`. Last checked on August 7, 2018.

[2] K. Bengler and F. Flemisch. Von H-Mode zur kooperativen Fahrzeugführung - Grundlegende ergonomische Fragestellungen. In *Proceedings of the 5. Darmstädter Kolloquium - Zukunft der Fahrzeugführung. Kooperativ oder autonom?* Stuttgart, 2011.

[3] K. Bengler, M. Zimmermann, D. Bortot, M. Kienle, and D. Damböck. Interaction Principles for Cooperative Human-Machine Systems. *it-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik*, 54 (4):157–164, 2012. doi: 10.1524/itit.2012.0680.

[4] L. Biester. *Cooperative Automation in Automobiles.* Doctoral dissertation, Humboldt-Universität zu Berlin, Berlin, Germany, 2008.

[5] BMW Group. 1 COMPANY. 4 BRANDS. ONE FUTURE, 2018. URL `https://www.bmwgroup.com/en/next100/brandvisions.html`. Last checked on August 7, 2018.

[6] H. Bubb, K. Bengler, R. E. Grünen, and M. Vollrath. *Automobilergonomie.* ATZ / MTZ-Fachbuch. Springer Vieweg, Wiesbaden, 2015. ISBN 978-3-8348-1890-4. URL `http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&AN=959251`.

[7] K. Christoffersen and D. D. Woods. How to make automated systems team players. *Advances in human performance and cognitive engineering research*, 2:1–12, 2002.

[8] Daimler AG. Autonomous: Daimler is a pioneer in autonomous driving. Research vehicles already are en route on test tracks., 2018. URL `https://www.daimler.com/case/autonomous/en/`. Last checked on August 7, 2018.

[9] E. Donges. Fahrerverhaltensmodelle. In H. Winner, S. Hakuli, and G. Wolf, editors, *Handbuch Fahrerassistenzsysteme*, pages 15–23. Vieweg+Teubner Verlag, Wiesbaden, 2012. ISBN 978-3-8348-1457-9.

[10] M. R. Endsley. Supporting situation awareness in aviation systems. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pages 4177–4181. IEEE, 1997. ISBN 0-7803-4053-1. doi: 10.1109/ICSMC.1997.637352.

[11] M. R. Endsley and E. O. Kiris. The Out-of-the-Loop Performance Problem and Level of Control in Automation. *Human Factors*, 37(2):381–394, 1995. doi: 10.1518/001872095779064555.

[12] F. Flemisch, M. Heesen, T. Hesse, J. Kelsch, A. Schieben, and J. Beller. Towards a dynamic balance between humans and automation: authority, ability, responsibility and control in shared and cooperative control situations. *Cognition, Technology & Work*, 14(1):3–18, 2012. ISSN 1435-5558. doi: 10.1007/s10111-011-0191-6.

[13] J.-M. Hoc. Towards a cognitive approach to human – machine cooperation in dynamic situations. *International journal of human-computer studies*, 54(4):509–540, 2001. doi: 10.1006/ijhc.2000.0454.

[14] M. M. Kienle. *Kooperative Interaktionskonzepte zum Fahren eines teilautomatisierten Fahrzeugs*. Doctoral dissertation, Technical University of Munich, Munich, 2011.

[15] G. Klein, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich. Ten Challenges for Making Automation a "Team Player" in Joint Human-Agent Activity. *IEEE Intelligent Systems*, 19(06):91–95, 2004. doi: 10.1109/MIS.2004.74.

[16] N. Peter. BMW iNEXT 2021 - SNEAK PREVIEW MODEL YEAR 2018., 2017. URL `https://www.bmwgroup.com/content/dam/bmw-group-websites/bmwgroup_com/ir/technologie_workshops/Presentation_Dr_Nicolas_Peter.pdf`. Last checked on May 10, 2018.

[17] SAE International. Surface Vehicle Recommended Practice: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2018.

[18] J. Schmidtler, V. Knott, C. Hölzel, and K. Bengler. Human Centered Assistance Applications for the working environment of the future. *Occupational Ergonomics*, 12(3):83–95, 2015. doi: 10.3233/OER-150226.

[19] C. Thornton, C. Braun, C. Bowers, and Ben B. Morgan JR. Automation Effects in the Cockpit: A Low-Fidelity Investigation. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 36(1):30–34, 1992. ISSN 1541-9312. doi: 10.1177/154193129203600109.

[20] M. Walch, K. Mühl, J. Kraus, T. Stoll, M. Baumann, and M. Weber. From Car-Driver-Handovers to Cooperative Interfaces: Visions for Driver–Vehicle Interaction in Automated Driving. In G. Meixner and C. Müller, editors, *Automotive User Interfaces*, volume 14 of *Human–Computer Interaction Series*, pages 273–294. Springer International Publishing, Cham, 2017. ISBN 978-3-319-49447-0. doi: 10.1007/978-3-319-49448-7_10.

Die Erfassung des Fahrerzustands als Voraussetzung für höher automatisierte Fahrfunktionen – Eine kritische Diskussion und ein Lösungsvorschlag

Maximilian Schwalm* und Nicolas D. Herzberger†

Zusammenfassung: Vor dem Hintergrund der Entwicklung automatisierter Fahrsysteme, bei denen Fahrer bisweilen die Fahraufgabe wieder übernehmen müssen, stellt sich die Frage ob und inwiefern Systeme zur Erfassung des Fahrerzustands als notwendige Voraussetzung betrachtet werden müssen um eine sichere Übergabe zu gewährleisten. In dem vorliegenden Kapitel soll zunächst diskutiert werden, welche minimalen Anforderungen an den Zustand des Fahrers in Abhängigkeit der Stufe der Automatisierung gestellt werden müssen. Anschließend wird diskutiert, ob die Detektion eines übernahmebereiten Fahrers durch Beobachtung und Messung auf bestimmten Stufen überhaupt valide möglich ist. Alternativ wird diskutiert, dass zunächst die Detektion einer fehlenden Übernahmebereitschaft bereits sehr viel einfacher zu realisieren ist. Dieses wird untermauert mit empirischen Ergebnissen einer Fahrsimulationsstudie bei der die Sollreaktion des Fahrers auf einen Take over Request (TOR) empirisch erfasst und mit der Übernahmeleistung in Verbindung gebracht wird.

Schlüsselwörter: Automationslevel, Fahrerinvolvierung, Fahrerzustand, Mensch-Maschine-Interaktion.

1 Einleitung

Autonomes Fahren wird die Automobilindustrie und die Mobilität der Gesellschaft verändern. Insbesondere die individuelle Fortbewegung im eigenen Fahrzeug wird sich mit zunehmender Automatisierung grundlegend wandeln. Die zukünftigen Kunden eines solchen Fahrzeugs sind bereit für die Möglichkeit die Zeit im Fahrzeug für andere Tätigkeiten nutzen zu können einen erhöhten Anschaffungspreis in Kauf zu nehmen [1]. Für viele Kunden wird der Mehrwert einer erhöhten Automation jedoch erst dann deutlich, sobald das System nicht nur eine Entlastung von der Fahraufgabe bedeutet, sondern eben auch die Möglichkeit bietet, sich anderen Tätigkeiten zuzuwenden. Genau hier liegt derzeit die Entwicklung bei höheren Automationsstufen an einem kritischen Punkt.

* Erster Autor ist seit 2013 Inhaber der Professur für Kraftfahrzeug Systembewertung am Institut für Kraftfahrzeuge der RWTH Aachen University, Steinbachstr. 7, 52070 Aachen (maximilian.schwalm@ika.rwth-aachen.de).

† Zweiter Autor ist wissenschaftlicher Mitarbeiter am Institut für Kraftfahrzeuge der RWTH Aachen University, Steinbachstr. 7, 52070 Aachen (nicolas.herzberger@ika.rwth-aachen.de).

Insbesondere bei SAE Level 3 wird vorausgesetzt, dass der Fahrer prinzipiell auch aus einer Situation heraus die Kontrolle über das Fahrzeug zurückerlangen kann, bei der er sich zuvor anderen Tätigkeiten zugewendet hat [2]. In diesem Kontext ist nach wie vor offen, ob und unter welchen Umständen dieses möglich ist und wie garantiert werden kann, dass der Fahrer übernahmebereit ist. Eine derzeit diskutierte Möglichkeit bestünde darin, den Fahrerzustand technisch zu erfassen und so die Übernahmebereitschaft im Normalbetrieb, an Systemgrenzen oder bei einem Systemausfall zu detektieren [z.B. 3, 4, 5, 6]. Ist der Fahrer übernahmebereit, übergibt das Fahrzeug die Kontrolle an den Fahrer. Detektiert das Fahrzeug, dass der Fahrer nicht übernahmebereit ist, bestünde die Möglichkeit, dass das System mit einem minimal riskanten Manöver das Fahrzeug sicher zum Stehen bringt.
Bevor jedoch die Übernahmebereitschaft eines Fahrers durch ein technisches System erfasst werden kann, erscheint es sinnvoll, zunächst einmal zu definieren worin diese Übernahmebereitschaft eigentlich besteht. Hier macht der vorliegende Artikel folgenden Vorschlag zum Vorgehen: Zunächst werden mögliche auftretende Fahrerzustände beschreiben und definiert. Anschließend wird pro Automatisierungsstufe nach SAE [7] ein Minimalzustand des Fahrers definiert der erforderlich ist um im Normalbetrieb, an der Systemgrenze und bei einem Systemausfall zu garantieren, dass der Fahrer die Übernahmebereitschaft herstellen kann.
Anschließend wird diskutiert, ob die Detektion der erforderlichen Fahrerzustände überhaupt möglich sein wird, und oder ob nicht vielmehr schon die Information einer fehlenden Übernahmebereitschaft also einer nicht-Passung von Fahrerzustand und geforderten Minimalzustand bereits sehr viel einfacher zu realisieren ist.

2 Anforderungen an den Fahrerzustand

Im Folgenden werden zunächst mögliche Fahrerzustände beschreiben und definiert. Anschließend erfolgt eine Beschreibung der Mindestanforderung an den Zustand des Fahrers in Abhängigkeit der Automationsstufen nach SAE [7].

2.1 Mögliche Fahrerzustände

In Abbildung 1 werden die für die Diskussion relevanten Fahrerzustände definiert. Dieses erfolgt anhand der Anforderungen an den Fahrer [8], hier unterteilt in die Wahrnehmung der Fahrsituation (Perzeption), in die Entscheidungsfindung (Kognition) und in die Handlung. Es werden insgesamt 5 Zustände unterschieden: „Vollständig involviert“, „Aktiv involviert“, „Passiv involviert“, „Rückholbar“ und „Nicht unmittelbar rückholbar“. Zentral ist dabei die Annahme, dass bei Transitionen zwischen den Fahrerzuständen – insbesondere in Richtung zunehmender Involvierung in die Fahraufgabe – nur bestimmte Schritte möglich sind. So ist es z.B. nicht möglich, einen schlafenden Fahrer („Nicht unmittelbar Rückholbar“) direkt in den Zustand „Vollständig verantwortlich“ zu überführen. Vielmehr durchläuft der Fahrer dabei mehrere Zwischenstufen. Dies muss

nicht zwangsläufig über eine parallel verlaufende stufenweise Degradation des Automationslevels erfolgen.

Fahrerzustand	Anforderungen		
	Perzeption	Kognition	Handlung
Vollständig verantwortlich	Monitoring aller relevanten Informationen einer Situation	Aktive Entscheidungen auf Basis vorliegender Informationen	Aktive Ausführung notwendiger Handlungen zur Situations- und Fahrzeugkontrolle
Aktiv involviert	Monitoring aller relevanten Informationen einer Situation	Aktive Entscheidungen auf Basis vorliegender Informationen & Überprüfung von Systementscheidungen	Aktive Ausführung eines Teils der Fahraufgabe (Längs- oder Querführung)
Passiv involviert	Monitoring aller relevanten Informationen einer Situation	Aktive Entscheidungen auf Basis vorliegender Informationen & Überprüfung von Systementscheidungen	Keine aktive Ausführung der Fahraufgabe
Rückholbar	Kein Monitoring der Situation, Monitoring aber prinzipiell möglich	Keine aktiven Entscheidungen & Systementscheidungen werden nicht überprüft	Keine aktive Ausführung der Fahraufgabe
Nicht unmittelbar rückholbar (z.B. Fahrer schläft)	Kein Monitoring der Situation	Keine aktiven Entscheidungen & Entscheidungen des Systems werden nicht überprüft	Keine aktive Ausführung der Fahraufgabe

Mögliche Transitionen (zunehmende Involvierung)

Zunehmende Involvierung in die Fahraufgabe

Abbildung 1: Darstellung möglicher Fahrerzustände und Transitionen

2.2 Anforderungen an den Fahrerzustand in Abhängigkeit des Automatisierungslevels

In Abbildung 2 erfolgt die Zuordnung der Mindestanforderungen an den Fahrer in Bezug auf den Fahrerzustand in Abhängigkeit der Automatisierungsstufen nach SAE. Gleichzeitig erfolgt eine Zuordnung der fahrrelevanten Aufgaben in Hinsicht auf Perzeption, Kognition und Handeln entweder zu dem Fahrer oder zu der Automation.

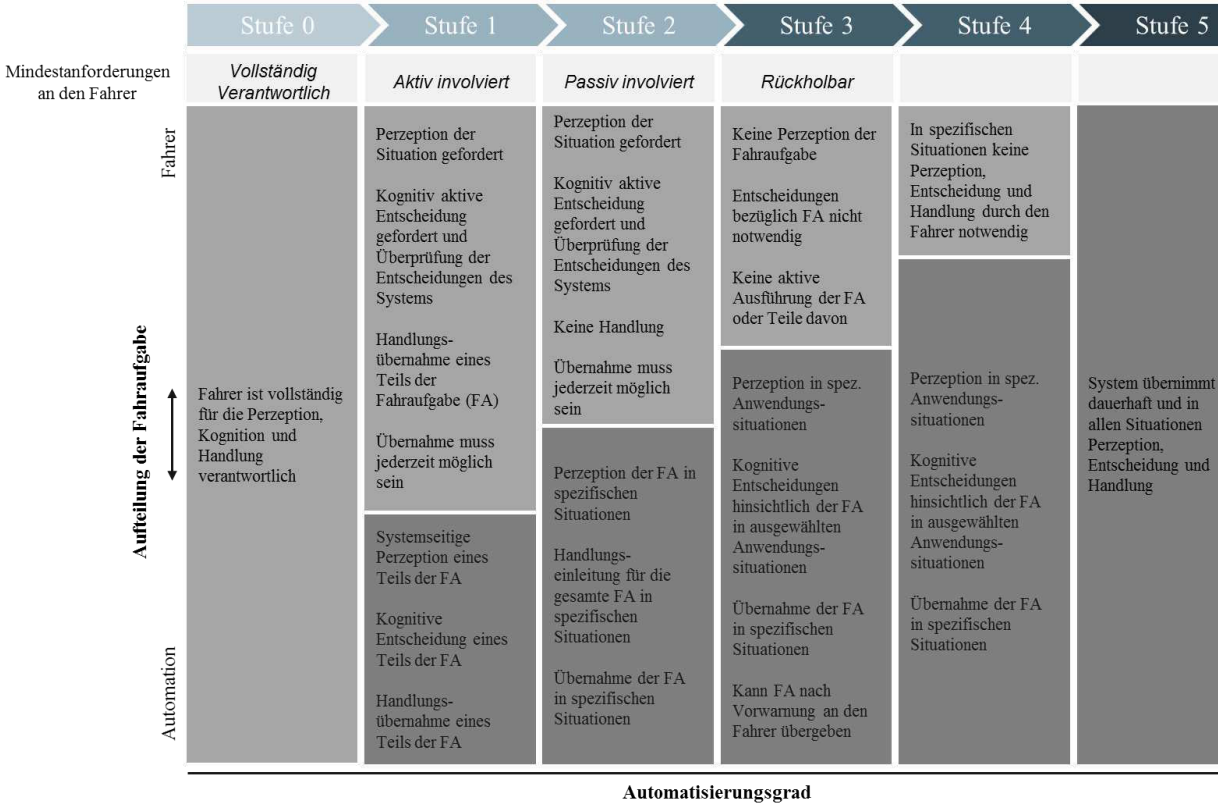


Abbildung 2: Darstellung der Mindestanforderungen an den Fahrer in Abhängigkeit der Automatisierungsstufen nach SAE

2.3 Möglichkeit der Detektion von Fahrerzuständen

Sollen nun die notwendigen Voraussetzungen an den Fahrerzustand in einer Situation durch Beobachtung überprüft werden, wird die Herausforderung insbesondere bei den Zuständen „Passiv involviert“ und „Rückholbar“ deutlich. Im Zustand „Passiv involviert“ ist es erforderlich, dass der Fahrer alle relevanten Informationen einer Situation wahrnimmt und Systementscheidungen überprüft. Die Herausforderung besteht dabei darin, durch Beobachtung fest zu stellen, ob und in wie fern der Fahrer ein vollständiges Bild der Verkehrssituation wahrgenommen und kognitiv verarbeitet hat ohne eine konkrete Handlung zu registrieren die auf eine korrekte Verarbeitung hindeutet. Durch z.B. eine kontinuierliche Erfassung der Blickzuwendung (Eyetracking) kann zwar das aktive physische scannen der Situation erfasst – nicht jedoch sicher die notwendige kognitive Verarbeitung der gesehenen Informationen überprüft werden. Belege für einen solchen Effekt bei dem zwar Informationen in den Fokus der visuellen Aufmerksamkeit rücken, aber dennoch falsch oder unvollständig verarbeitet werden, finden sich in der Literatur unter anderem in der Debatte rund um das Phänomen der „inattentional blindness vgl. u.a. [9]. Noch weniger Anhaltspunkte ergeben sich in dem Zustand „Rückholbar“. Hier ist das Monitoring der Verkehrssituation nicht erforderlich und die Erfassung der Blickzuwendung somit nicht aussagekräftig. Verhaltensbeobachtungen können hier zwar

Anhaltspunkte liefern, aber keine Sicherheit, ob, in wie fern und wie schnell der Fahrer tatsächlich rückholbar ist.
Wenn also die verlässliche Detektion des Fahrerzustands auf Basis einer Beobachtung schwierig erscheint, stellt sich die Frage, ob nicht zunächst die Detektion eines nicht ausreichenden Fahrerzustands in einer Automationsstufe eher möglich ist.

3 Detektion einer fehlenden Übernahmefähigkeit mittels Diagnose einer Sollreaktion auf einen TOR

Im Gegensatz zur Detektion der Übernahmefähigkeit erscheint die Detektion einer fehlenden Übernahmefähigkeit zunächst deutlich einfacher. Folgt man wie weiter oben diskutiert der Idee, dass eine Übergabe an den Fahrer ein kaskadierter Prozess ist, bei dem der Fahrer zunächst z.B. aus dem Zustand ‚Rückholbar' in den Zustand der passiven Involvierung wechselt – um sich z.B. zunächst ein Bild der Verkehrssituation zu machen und die physische Übernahme der Steuerung des Fahrzeugs vorzubereiten (vollständig involviert) – erscheint es zielführend für eben diesen Wechsel des Zustands Indikatoren zu definieren die sich in Beobachtungskriterien überprüfen lassen. Wie beispielsweise der zwingend notwendige Blick in die Verkehrssituation. Der entscheidende Punkt ist dabei, dass es sich hier durch den Zeitpunkt der Übernahmeaufforderung um eine zeitlich festgelegte und empirisch beschreibbare Reaktion des Fahrers handelt. Ist es möglich das Ausbleiben einer solchen Sollreaktion zu detektieren, kann mit sehr hoher Wahrscheinlichkeit zumindest einmal von einer fehlenden Übernahmefähigkeit ausgegangen werden.

Diese Sollreaktion auf ein Take Over Request (TOR) in einer automatisierten Fahrsituation wurde im Rahmen einer empirischen Untersuchung am Fahrsimulator des ika (Instituts für Kraftfahrzeuge Aachen) untersucht (siehe Abbildung 3).

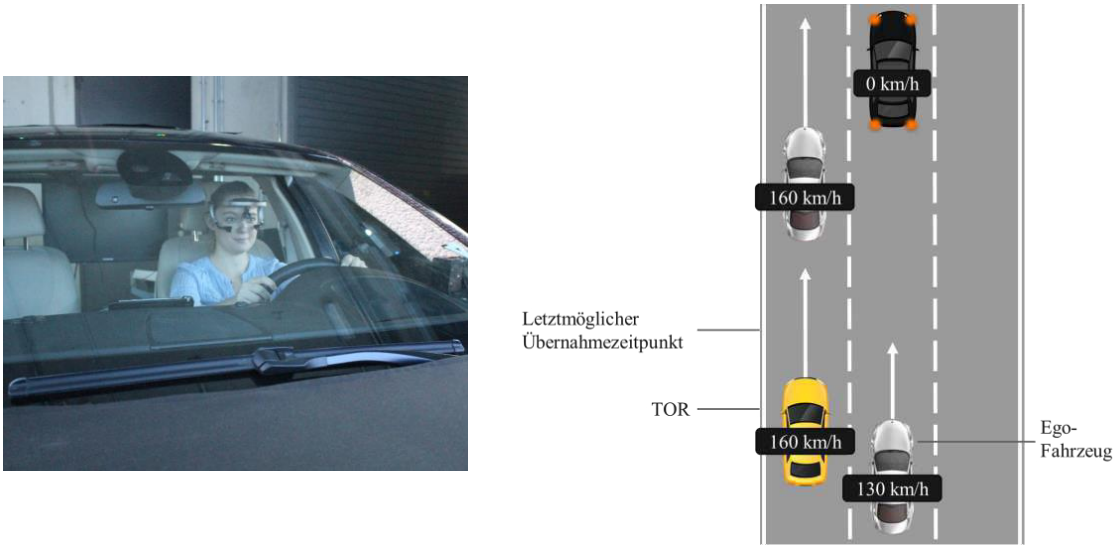


Abbildung 3: Links: Probandin im Fahrsimulator des Instituts für Kraftfahrzeuge der RWTH Aachen University mit dem verwendeten Eyetracking System zur Erfassung der Orientierungsreaktion. Rechts: Schematische Darstellung der kritischen Situation.

Hierbei fuhren N = 50 Probanden in einer 13-minütigen teilautonomen Fahrt bis diese in einer kritischen Situation die Fahraufgabe vom System übernehmen mussten. In dieser Übernahmesituation ließen sich Musterabfolgen in der Orientierungsreaktion der Probanden erkennen, die auf das Gelingen der Übernahme bzw. auf eine mangelnde Übernahmefähigkeit hindeuten – unabhängig davon, ob die Probanden eine fahrfremde Tätigkeit ausführten oder nicht. Abbildung 4 zeigt beispielhafte Orientierungsmuster zweier Probanden auf den Take Over Request. (Detaillierte Ergebnisse und Beschreibung der probandenübergreifenden Muster, wird in [10] erfolgen).

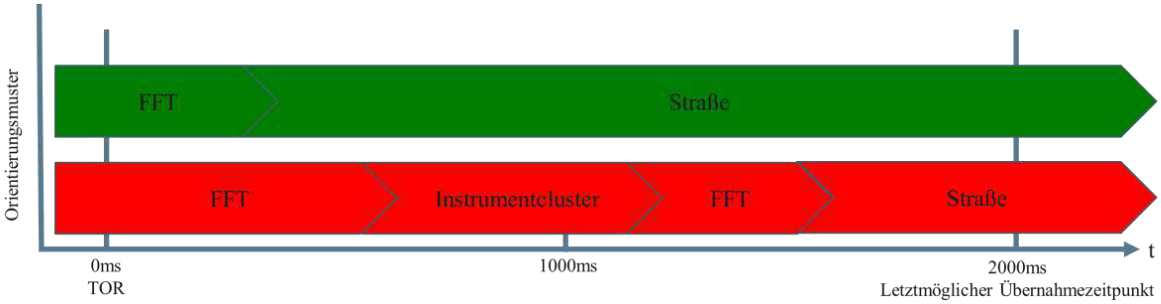


Abbildung 4: Blickmuster von zwei ausgewählten Probanden beginnend bei der fahrfremden Tätigkeit (FFT) auf einen Take Over Request. Das grüne Muster stellt eine Orientierungsreaktion mit erfolgreicher Übernahme dar, das rote Muster stellt eine Orientierungsreaktion mit misslungener Übernahme dar.

Diese ersten gefundenen Orientierungsmuster erscheinen als ein vielversprechender Ansatz um im Falle einer notwendigen Übergabe der Fahraufgabe an den Fahrer einen

Abgleich der gezeigten Abfolge mit einer Sollreaktion durchführen zu können und so frühzeitig eine fehlende Übernahmefähigkeit detektieren zu können.

4 Abschließende Diskussion

In dem vorliegenden Kapitel wird diskutiert, ob und inwiefern vor dem Hintergrund der Entwicklung automatisierter Fahrsysteme, bei denen Fahrer bisweilen die Fahraufgabe wieder übernehmen müssen, Systeme zur Erfassung des Fahrerzustands als notwendige Voraussetzung betrachtet werden müssen um eine sichere Übergabe zu gewährleisten. Dafür werden zunächst auf Basis von einmal mögliche Fahrerzustände definiert. Anschließend erfolgt eine Zuordnung der notwendigen Fahrerzustände zu den Stufen der Automatisierung nach SAE. Es wird argumentiert, dass die Prognose einer erfolgreichen Übernahme der Fahraufgabe insbesondere aus dem Zustand ‚Rückholbar' heraus durch reine Beobachtung des Fahrers fraglich erscheint. Vielmehr wird vorgeschlagen, zunächst auf Basis der Analyse einer Sollreaktion des Fahrers auf den Take Over Request zu bestimmen, ob ein Fahrer wahrscheinlich keine ausreichende Übernahmefähigkeit aufweist. Dieses wird mit einer ersten eigenen empirischen Simulatorstudie untermauert. Die Ergebnisse dieser Studie lassen auch erste Muster erkennen mit denen sich eine misslungene Übernahme der Fahraufgabe durch den Fahrer frühzeitig erkennen lässt. Dabei ist zu berücksichtigen, dass die Orientierungsmuster bislang nur für eine spezifische Fahrsituation mit definierter Time to Collision (TTC) und einer einzigen vorgegebenen fahrfremden Tätigkeit analysiert wurden. Die Übertragbarkeit auf andere Situationen muss in weiteren Erhebungen nachgewiesen werden. Ebenso verbleibt für nachfolgende Studien der Nachweis offen, ob mittels der Analyse der diagnostischen Reaktion auf einen TOR gegebenenfalls auch der Nachweis der Übernahmefähigkeit mit ausreichender Validität getroffen werden kann.
Ferner ist zu diskutieren, wie im Falle einer sicher detektierten fehlenden Übernahmefähigkeit das Systemverhalten auszulegen ist. So könnte beispielsweise die Einleitung eines „minimal risk maneuver" zur Absicherung der Situation und somit zur Sicherstellung der Fahrersicherheit zu einem wesentlich früheren Zeitpunkt erfolgen, als wenn das Ausbleiben der Reaktion des Fahrers vollständig abgewartet werden muss. Es bleibt nachzuweisen welcher konkrete Sicherheitsgewinn durch ein solches Eingreifen in kritischen Fahrsituationen erreicht werden kann.

5 Danksagung

Die vorgestellten Arbeiten sind im Rahmen des Teilprojekts „Systemergonomie für kooperativ interagierende Automobile: Nachvollziehbarkeit des Automationsverhaltens und Eingriffsmöglichkeiten des Menschen im Normalbetrieb, an Systemgrenzen und bei Systemausfall" innerhalb des Schwerpunktprogrammes 1835 „Kooperativ interagierende Automobile" der Deutschen Forschungsgemeinschaft DFG durchgeführt worden.

Literatur

[1] Studienbericht 2018, Enabling the Value of Time, Implikationen für die Innenraumgestaltung autonomer Fahrzeuge. Fraunhofer IAO, Cordence worldwide, Horvat & Partners

[2] Gasser, T. M., Arzt, C., Ayoubi, M., Bartels, A., Bürkle, L., Eier, J., ... & Vogt, W. (2012). Rechtsfolgen zunehmender Fahrzeugautomatisierung. Berichte der Bundesanstalt für Straßenwesen. Unterreihe Fahrzeugtechnik, (83).

[3] Flemisch, F., Altendorf, E., Canpolat, Y., Weßel, G., Baltzer, M., Lopez, D., Herzberger, N. D., Voß, G. M. I., Schwalm, M. & Schutte, P. (2017). Uncanny and unsafe valley of assistance and automation: First sketch and application to vehicle automation. In *Advances in Ergonomic Design of Systems, Products and Processes* (pp. 319-334). Springer, Berlin, Heidelberg.

[4] Herzberger, N. D., Voß, G. M. I., Becker, F., Grazioli, F., Altendorf, E., Canpolat, Y., Flemisch, F. O. & Schwalm, M. (2018). Derivation of a Model of Safety Critical Transitions between Driver and Vehicle in Automated Driving. In *International Conference on Applied Human Factors and Ergonomics*. Springer, Cham.

[5] Flemisch, F., Schieben, A., Schömig, N., Strauss, M., Lüke, S.: Design of Human Computer Interfaces for Highly Automated Vehicles in the EU-Project HAVEit. In Stephanidis, C. (Hrsg.) Universal Access in Human-Computer Interaction, 6767 270--279. Springer, Heidelberg (2011)

[6] Rauch, N., Kaussner, A., Krüger, H.-P., Boverie, S., Flemisch, F.: The importance of driver state assessment within highly automated vehicles. Paper presented at the 16th ITS World Congress, Stockholm, Sweden, 21.-25. (2009)

[7] SAE International, http://www.sae.org/misc/pdfs/automated_driving.pdf (2014)

[8] Herzberger, N. D., Voß, G. M., & Schwalm, M. (2017). Identification of Criteria for Drivers' State Detection. In International Conference on Applied Human Factors and Ergonomics (pp. 798-806). Springer, Cham.

[9] Beanland, V., Pammer, Kristen (2010). Looking without seeing or seeing without looking? Eye movements in sustained inattentional blindness. Vision Research 50 (2010) 977-988

[10] Herzberger, N. D. & Schwalm, M. in prep. (2018)

Tracking des Aufmerksamkeitsziels des Fahrers mittels eines Multi-Hypothesen Multi-Modell Filters

Julian Schwehr* und Volker Willert†

Zusammenfassung: Für eine sichere Übergabe der Fahraufgabe oder adaptive Warnstrategien sind Informationen über die Wahrnehmung einer Situation durch den Fahrer unerlässlich. In dieser Arbeit wurde ein Multi-Hypothesen Multi-Modell Tracking-Algorithmus entworfen, um das Aufmerksamkeitsziel des Fahrers zu schätzen und zeitlich zu verfolgen. Dabei werden sowohl Objektbewegungen als auch raumzeitliche Modellannahmen des menschlichen Blickverhaltens explizit in den Übergangsschritt eines probabilistischen Filters integriert, was zu einer robusten Schätzung des Blickziels führt. Durch die Verwendung dynamischer und statischer potenzieller Blickziele aus einer Objektliste bzw. einem Freiraum-Spline ist der Algorithmus prinzipiell unabhängig von der verwendeten Sensoranordnung. Die Ergebnisse des Filters werden anhand einer beispielhaften Szene aus realen Testdaten analysiert. Zusätzlich wird der Typ des Blickziels mit Hilfe semantischer Segmentierung im Kamerabild bestimmt und das Verfahren mit einem rein bildbasierten Ansatz ohne Tracking verglichen.

Schlüsselwörter: Bayesian Estimation, Fahrer-Umfeld-Fusion, Multi-Hypothesen Multi-Modell Tracking

1 Einführung

1.1 Motivation

Nach aktuellem Stand scheint der Weg zum hochautomatisierten Fahren vorgezeichnet. Mehrere Hersteller planen, ihre ersten Fahrzeuge nach SAE International Standard J3016 Level 4 mit Beginn des nächsten Jahrzehnts zu vermarkten [1]. Aufgrund der Verschiebung der Verantwortung vom Fahrer hin zu den Herstellern und den Herausforderungen durch abgelenkte Fahrer und der damit verbundenen Frage der sicheren Übergabe haben einige Hersteller angekündigt, Level 3 zu überspringen [2]. Dennoch ist davon auszugehen, dass auch zukünftig Fahrerassistenzsysteme (FAS) der Level 0 bis Level 3 weiter verbessert werden. Sowohl diese zukünftigen FAS, als auch das automatisierte Fahren können von einer Schätzung der Situationswahrnehmung des Fahrers profitieren, um ihre Handlungsstrategie besser an die jeweilige Situation anzupassen. Einer der zukünftigen Aspekte von FAS wird ein umfassendes Situationsverständnis der Assistenzfunktion sein, welches sowohl Fahrer als auch Umwelt miteinbezieht [3]. Da das menschliche Blickverhalten den wichtigsten Indikator für das Situationsbewusstsein des Fahrers während des Autofahrens darstellt, wird dieses bereits seit Jahrzehnten sowohl im Fahrsimulator als auch in Realfahrten untersucht.Seitdem wächst das Forschungsinteresse an der automatisierten Erkennung und Schätzung der Aufmerksamkeit und Situationswahrnehmung des Fahrers. Im Falle automatisierten Fahrens des Level 3 ist die Übergabe der Fahraufgabe vom Auto an den Fahrer entscheidend. Aktivitäts-, Aufmerksamkeits- und Wachsamkeitsanalysen sind notwendig, um die erforderliche Zeit für eine sichere Übergabe zu bestimmen. Aber auch Systeme der Level 0 bis 2 können durch

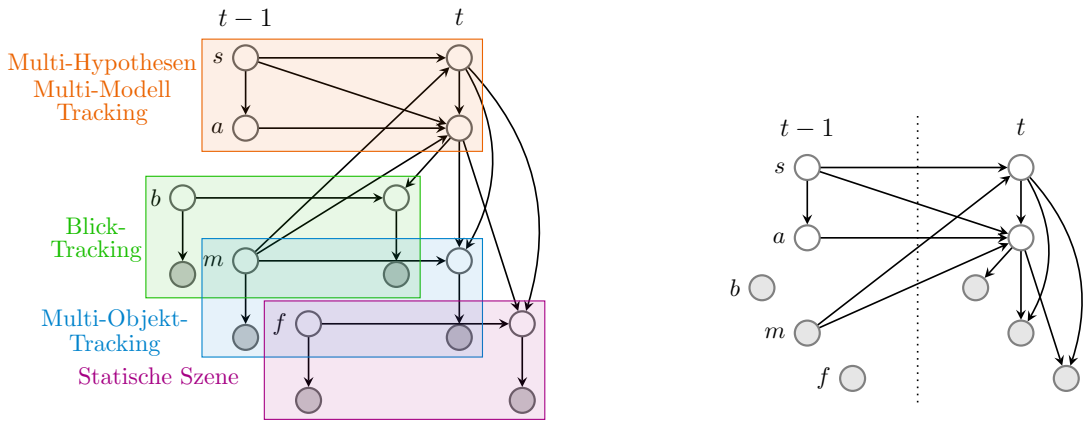
*Julian Schwehr ist Wissenschaftlicher Mitarbeiter am Fachgebiet Regelungsmethoden und Robotik, Technische Universität Darmstadt (e-mail: julian.schwehr@rmr.tu-darmstadt.de).

†Volker Willert ist Akademischer Oberrat am Fachgebiet Regelungsmethoden und Robotik, Technische Universität Darmstadt (e-mail: vwillert@rmr.tu-darmstadt.de).

eine Schätzung der Fahreraufmerksamkeit verbessert werden. Diese Schätzung reicht bis zur Unterscheidung von vom Fahrer gesehene und übersehene Verkehrsteilnehmer. Aus technischer Sicht besteht die Aufgabe dann darin, das Blickziel, also das Objekt der gegenwärtigen visuellen Fixation, zu bestimmen und daraus die Wahrnehmung des Fahrers abzuleiten. Auf dieser Information aufbauend, könnte der Fahrer beispielsweise bei manueller Fahrt im Falle tatsächlicher Ablenkung schon bei geringerem Risiko frühzeitiger auf ein einscherendes Fahrzeug hingewiesen werden. Kreuzungs- und Notbremsassistenten könnten ihre Interventionsstrategie der Situation anpassen, z.B. wenn der Fahrer an einer Rechts-vor-Links Kreuzung nicht ausreichend visuell absichert. Bereits eine Adaption von wenigen 100 ms und ein besseres Verständnis des Fahrers kann die Falsch-Positiv-Rate eines Totwinkelassistenten verbessern und damit die Akzeptanz erhöhen. Systeme dieser Art könnten den Fahrer mit Hilfe des zusätzlichen Wissens demnach im Falle einer kritischen Situation entsprechend seiner Bedürfnisse, ähnlich eines aufmerksamen Beifahrers, warnen und eingreifen und damit zu einer intensiveren Kooperation zwischen Fahrer und Fahrzeug führen.

1.2 Stand der Forschung

Trotz der intensiven Bemühungen Fahrerverhalten und Fahrerblickbewegungen zu verstehen, ist es unmöglich, allein mit Hilfe von tragbaren oder berührungslosen kamerabasierten Eye-Tracking-Systemen, automatisiert valide Ground Truth Informationen darüber zu erhalten, *was* der Fahrer tatsächlich *gesehen* und *wahrgenommen* hat. Dieses Fehlen einer Ground Truth über das *Bewusstsein* des Fahrers über andere Verkehrsteilnehmer ist jedoch wahrscheinlich der Grund, warum die meisten Arbeiten in der Literatur, die sich mit der Blickzielschätzung befassen, auf einfachen Distanzkriterien beruhen [4–10]. Da das Blickziel nicht ohne Weiteres bestimmt werden kann, gilt das Ziel als gesehen, wenn der Blick in eine bestimmte Region fällt. Hierbei werden die Distanzen in unterschiedlichen Koordinatensystemen definiert [11], meist basierend auf der Wahl der Umfeldsensorik. Die meisten Arbeiten verwenden Kameras [4–6, 8, 10, 12], seltener werden Laser Scanner [7, 9] und Radar [11] genutzt. Allein in [7] wird die Modellierung um eine minimale Fixationszeit erweitert und in [8] wird eingeräumt, dass die Blickbewegung des Fahrers nicht allein durch das Auftreten von äußeren Objekten in bestimmten Regionen zuverlässig erklärt werden kann. Nichtsdestoweniger haben die vielen Simulator- und realen Fahrstudien einige allgemein beobachtbare Eigenschaften des Blickverhaltens aufgedeckt [13]. Darunter sind formulierbare Regeln, wie beispielsweise dass Fixationen auf aufgabenrelevante Objekte und Positionen fallen, einzelne Fixationen interpretierbare funktionale Rollen haben und Fixationen unregelmäßig unterbrochen sind. Beim Autofahren werden Fixationen auf relevante Ziele immer wieder durch Blicke nach vorne unterbrochen. Was in aktuellen Arbeiten bisher fehlt, ist die tatsächliche Verwendung dieser bekannten, wenn auch vagen, Verhaltensregeln als Modellwissen für eine gemeinsame Beschreibung des Fahrerverhaltens und der Umgebung zur Schätzung des Blickziels. Hierfür bieten sich Bayessche Filter an, die in zahlreichen Anwendungen gezeigt haben, die Zustandsschätzung, gegeben rauschbehafteter Messungen, mit Modellwissen verbessern zu können. In dieser Arbeit bauen wir auf unseren Ideen aus [11] auf mit dem übergeordneten Ziel, jeweils das Objekt zu extrahieren, dem der aktuelle Blick des Fahrers gilt. Zu diesem Zweck wird in dieser Arbeit ein Multi-Hypothesen Multi-Modell (MHMM) Tracking vorgestellt, welches für jedes Objekt explizit die Wahrscheinlichkeit schätzt, das aktuelle Blickziel zu sein. Zusammenhängende Phasen, die das identische Objekt als Blickziel ausgeben, können dann als Fixation detektiert werden und bei ausreichender Dauer als Indikator dienen, dass sich der Fahrer des entsprechenden Objekts bewusst ist. Vom Standpunkt der Motivation gesehen, ist [14] am ähnlichsten zu dieser Arbeit. Basierend auf ähnlicher Argumentation werden hier die relative Bewegungen der Objekte und des Fixationspunktes als Features genutzt, um das aktuelle Blickziel lernbasiert über ein Klassifikationsproblem zu bestimmen. Die



(a) Kombination des vorgeschlagenen Multi-Hypothesen Multi-Modell Filters zusammen mit probabilistischen Modellen für Blick, mehreren Objekten und statischer Szene.

(b) Vereinfachte Version des vorgeschlagenen Multi-Hypothesen Multi-Modell Filters mit Blick, Objekten und statischer Szene als Messungen.

Abbildung 1: Multi-Hypothesen Multi-Modell Filter in zwei verschieden Varianten mit Umschaltvariable s und Variable a für den Aufmerksamkeitsbereich. b, m, f beschreiben Blick, Objekte und statische Szene. Pfeile zu den Messungen zum Zeitpunkt $t-1$ wurden für bessere Übersicht nicht gezeichnet.

Blickrichtung wird hier jedoch über einen tragbaren Eye-Tracker erhalten.

2 Multi-Hypothesen Multi-Modell Tracking

In dieser Arbeit wird vorausgesetzt, dass die Wahrnehmungen der Verkehrsszene durch den Fahrer und das Fahrzeug konsistent sind. Es wird also einerseits angenommen, dass das Fahrzeug das Fixationsobjekt erfasst und die relative Position des Objektes ausreichend genau bestimmt hat. Andererseits wird angenommen, dass die gemessene Blickbewegung des Fahrers zur relativen Bewegung des jeweils aktuellen Fixationsobjektes passt. Mit dieser Prämisse ist das übergeordnete Ziel die Schätzung der a-posteriori Verbundwahrscheinlichkeit des probabilistischen Netzwerks in Abbildung 1a, welches die Wechselwirkung zwischen Fahrerblick und potenziellen Blickzielen beschreibt. Die Aufgabe des entworfenen Multi-Hypothesen Multi-Modell (MHMM) Filters auf der oberen Filterebene ist, den Aufmerksamkeitsbereich a und das Blickziel s mit geeigneten Bewegungsmodellen sowie den Blick- (b) und Objektschätzungen (m, f) zu schätzen und zu verfolgen. Im Falle exakter und vollständiger Inferenz werden die stochastischen Variablen auf der unteren Ebene des Filters (Blickrichtung, Objektposition und -geschwindigkeit) durch die Schätzung von s und a beeinflusst. Somit ist bei einer gemeinsamen Betrachtung ein Informationsgewinn in beiden Domänen, also Fahrerbeobachtung und Umgebungsschätzung, möglich: Korrekturen der Blickrichtungsmessung sind ebenso möglich wie eine Verbesserung der Positionsschätzung des fixierten Objekts. Statt der umfassenden probabilistischen Beschreibung in Abbildung 1a verwendet diese Arbeit eine vereinfachte Version des vollständigen Filters, welche in Abbildung 1b dargestellt ist und sich allein auf den MHMM-Filter konzentriert. Dieser Filter ist eine Kombination aus einem Multi-Hypothesen (MH) und einem Multi-Modell (MM) Tracking. Ein MH-Tracker stellt eine Verteilung über alle möglichen Objekt-Hypothesen dar und bestimmt die jeweiligen Gewichte jeder Hypothese basierend darauf, wie gut die Hypothesenvorhersagen zur Sensormessung passen. Im Unterschied dazu verfolgt ein MM-Filter verschiedene (Bewegungs-)Modelle für jeweils eine Hypothese und bestimmt das Übergangsmodell ebenfalls basierend darauf, wie gut die verschiedenen Modellvorhersagen zur Sensormessung passen [15].

Das Ziel dieser Arbeit ist die Schätzung der Verbundwahrscheinlichkeit $p(\{s,a\}^t|\{b,m,f\}^{1:t})$[1]. Die Schaltvariable s ist eine $(n+1)$-dimensionale binäre Zufallsvariable $s \in \mathbb{R}^{n+1}$ mit $s_i \in \{0,1\}$, $\sum_i s_i = 1$, wobei n die Anzahl der Objekte in der Umgebung beschreibt und die zusätzliche Dimension die statische Szene enthält. Jedes potenzielle Blickziel i besitzt eine gewisse Wahrscheinlichkeit $p(s_i = 1) = \pi_i$, dass es das aktuelle Blickziel des Fahrers ist und eine gewisse Ausdehnung $p(a|s_i = 1)$ modelliert durch eine Normalverteilung $\mathcal{N}(a|\mu_i, \Sigma_i), a, \mu_i \in \mathbb{R}^2, \Sigma_i \in \mathbb{R}^{2\times 2}$. Der Aufmerksamkeitsbereich des Fahrers wird durch die gewichtete Kombination der einzelnen Normalverteilungen als Gaußsches Mischmodell (engl. mixture of Gaussians)

$$p(\{s,a\}^t|\{b,m,f\}^{1:t}) = \prod_{i=1}^{n+1} \pi_i^{s_i} \mathcal{N}(a^t|\mu_i^t, \Sigma_i^t)^{s_i} \tag{1}$$

dargestellt. Die Messung $b \in \mathbb{R}$ repräsentiert den Gierwinkel der Blickrichtung gemessen in Fahrzeugkoordinaten durch das Eye-Tracking-System. $m \in \mathbb{R}^{n\times 5}$ ist eine geordnete Objektliste, welche Position (x, y), Ausrichtung (φ) und relative Geschwindigkeit $(v_{rel,x}, v_{rel,y})$ der Objekte enthält, basierend auf den Messungen der Front- und Seitenradare. Die Messung der statischen Umgebung $f = \{f_i\}$ ist die Menge der Kontrollpunkte der B-Spline-Kurve, die den Freiraum beschreibt [16].

2.1 Übergangswahrscheinlichkeiten (Prädiktion)

Bei der Modellierung der Übergangswahrscheinlichkeiten muss jede Möglichkeit der Blickbewegung berücksichtigt werden. Dabei können die folgenden Fälle auftreten: Der Blick kann auf einem spezifischen Objekt $i \in 1, ..., n$ haften bleiben (Fixations- und Folgemodell) oder von einem Objekt i zu einem anderen Objekt $j \in 1, ..., n$ oder auch der statischen Umgebung $(j = n + 1)$ springen (Sakkadenmodell). Umgekehrt könnte auch die statische Umgebung das aktuelle Blickziel sein $(i = n + 1)$ und der Fahrer könnte zum nächsten Zeitschritt mit seinem Blick an der gleichen Stelle bleiben, zu einer anderen Stelle in der statischen Umgebung springen (für beide Fälle gilt $j = n+1$) oder ein Objekt fixieren $(j \in 1, ..., n)$. Aufgrund der entworfenen Filterstruktur in Abbildung 1b und der Formulierung der a-priori Verteilung (1) als Gaußsches Mischmodell kann der Übergang aufgeteilt werden in ein skalares Übergangsmodell der Schaltvariable und in ein räumliches Übergangsmodell

$$p(\{s,a\}^t|\{s,a,m\}^{t-1}) = p(s^t|\{s,m\}^{t-1})p(a^t|s^t, \{s,a,m\}^{t-1}). \tag{2}$$

2.1.1 Übergang des Schaltzustands

Normalerweise sind die Übergangsgewichte konstante Entwurfsparameter. Hier wird der Übergang

$$p(s^t|\{s,m\}^{t-1}) = \prod_{j=1}^{n+1} \prod_{i=1}^{n+1} \left(\alpha(s^t, s^{t-1})\beta(s^t, \{s,m\}^{t-1})\right)^{s_i s_j} \tag{3}$$

jedoch durch zwei Potenziale dargestellt, welche einen zeitlichen Aspekt $\alpha(s^t, s^{t-1})$ und einen räumlichen Aspekt $\beta(s^t, \{s,m\}^{t-1})$ modellieren. Dabei berücksichtigt das zeitliche Übergangsmodell α eine Veränderung der Übergangswahrscheinlichkeiten für Fixationen und Sakkaden über der Zeit. Je länger die detektierte Fixationsdauer ist, desto mehr steigt die Wahrscheinlichkeit für eine Sakkade, also einen Sprung hin zu einem anderen Ziel. Das räumliche Übergangsmodell β betrachtet dagegen die geometrischen Beziehungen der Objekte in der Szene und bezieht psychophysikalische Randbedingungen wie maximale Sakkadengeschwindigkeit und mittlere Blickrichtung mit ein.

[1] Wir nutzen die Notation $(\{\cdot,\cdot\}^{1:t})$ als verkürzte Schreibweise um mehrere Variablen mit dem gleichen Zeitindex zu beschreiben, normalerweise dargestellt durch $(\cdot^{1:t}, \cdot^{1:t})$.

2.1.2 Räumlicher Übergang

Wenn der Blick des Fahrers auf einem Objekt verharrt, die Augenbewegung also eine Fixation oder Folgebewegung (engl. smooth pursuit) beschreibt, sollte die entsprechende Wahrscheinlichkeitsverteilung die gleiche Bewegungsdynamik wie das fixierte Objekt aufweisen. Gleichzeitig sollte, für den Fall dass der Fahrerblick zu einem anderen Objekt j wechselt, die prädizierte Verteilung auch an der Stelle des Objekts j sein. Der räumliche Übergang wird daher als Normalverteilung

$$p(a^t|s^t, \{s, a, m\}^{t-1}) = \prod_{j=1}^{n+1} \prod_{i=1}^{n+1} \mathcal{N}(a^t|a^{t-1} + u_{ij}^{t-1}, \Gamma_{ij})^{s_i s_j} \tag{4}$$

modelliert, wobei u_{ij} einen Steuereingang darstellt und das Prozessrauschen durch die Kovarianzmatrix Γ_{ij} gegeben ist. Das Teilergebnis des späteren Inferenzschritts für den Übergang von Objekt i zu Objekt j ist dann die Normalverteilung $\mathcal{N}(a^t|\mu_i^{t-1} + u_{ij}^{t-1}, \Sigma_i^{t-1} + \Gamma_{ij})^{s_i s_j}$. Da für $i = j$ der Blick eine Bewegungsdynamik konsistent zur Messung der Objektdynamik aufweisen soll, wird der Steuereingang durch die relative Bewegung des Blickziels modelliert, also $u_{ij}^{t-1} = v_i^{t-1}\Delta t$, wobei Δt die Abtastzeit des Blicks beschreibt. Analog dazu wird der Steuereingang im Fall $i \neq j$ zu $u_{ij}^{t-1} = x_j^{t-1} - \mu_i^{t-1} + v_j^{t-1}\Delta t$ gewählt, sodass die prädizierte Position der Aufmerksamkeit auf der prädizierten Position des Objekts liegt. Für die statische Umgebung ($j = n+1$) besteht die relative Bewegung allein aus der Kompensation der Eigenbewegung und an Stelle der Objektposition x_j^{t-1} wird die letzte Schätzung μ_{n+1}^{t-1} genutzt. Zusätzlich wird das Prozessrauschen deutlich größer gewählt, da der Fahrerblick von der aktuellen Position auf dem Freiraum-Spline zu fast jeder anderen Position auf dem Spline springen kann.

2.2 Messmodelle

Es wird angenommen, dass die Messungen von Blick, dynamischen Objekten und der statischen Umgebung unabhängig voneinander sind, sodass die Likelihood des Filterschritts in die Terme für die einzelnen Messungen faktorisiert. Für die Blickmessung wird ein Modell verwendet, welches Positionen, die eine geringere Winkeldistanz zur Blickmessung aufweisen, eine höhere Likelihood zuordnet und eine geringe Likelihood an Positionen fern des Blickstrahls.

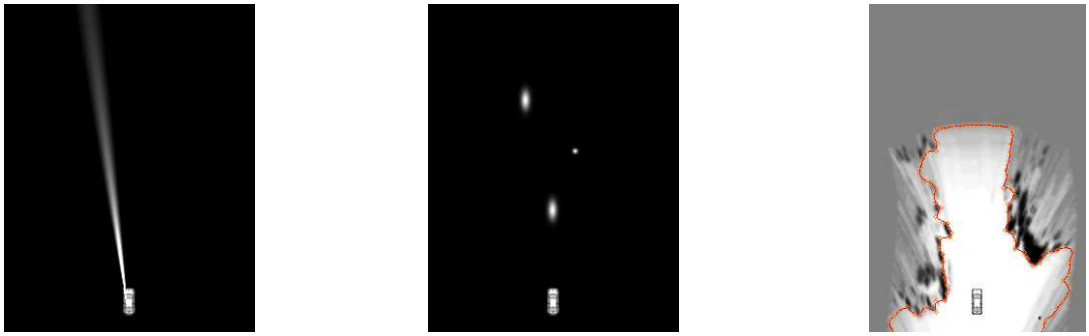
Bei den Objekten wird jedes einzelne Objekt als gewichtete Normalverteilung repräsentiert, deren Mittelwert der Position des jeweiligen Objekts entspricht. Die Kovarianzmatrix wird so gewählt, dass die Hauptachsen der 90% Kovarianzellipse der Breite und Länge des Objekts entsprechen. Die Matrix ist außerdem entsprechend der Ausrichtung des Objekts gedreht.

Die Messung der statischen Umgebung wird durch die Kontrollpunkte einer Freiraum-B-Spline-Kurve repräsentiert [16]. Da diese Spline-Kurve weder in geschlossener Form und schon gar nicht als Normalverteilung formuliert werden kann, dienen Abtastungen der Kurve als diskrete Realisierungen des Likelihood. Der Status der Kontrollpunkte gibt zudem an, ob es sich an der jeweiligen Position um ein statisches Hindernis oder einen Übergang zu unbekannter Umgebung handelt. Exemplarische Bilder der verschiedenen Messmodelle sind in Abbildung 2 dargestellt.

2.3 Inferenz

Im Inferenzschritt des entworfenen MHMM Tracking-Algorithmus treten zwei Schwierigkeiten auf:

1. Sowohl Blick- als auch Freiraum-Likelihood sind keine Normalverteilungen, weshalb die Inferenz nicht in geschlossener Form dargestellt werden kann.



(a) Blickmodell (hellere Bereiche beschreiben höhere Likelihood)

(b) Likelihood für drei dynamische Objekte verschiedener Größe

(c) Freiraum-Spline hinterlegt mit der Belegungskarte

Abbildung 2: Exemplarische Darstellung der Messmodelle.

2. Aufgrund der exponentiell steigenden Anzahl an Gaußschen Mischkomponenten (schon der erste Prädiktionsschritt führt auf $n+1$ Summen von $n+1$ Normalverteilungen) wird der Filter ohne geeignete Approximationen praktisch schnell unberechenbar.

Um dem ersten Problem zu begegnen wird jede Übergangswahrscheinlichkeit von Objekt i zu Objekt j durch Minimierung der Kullback-Leibler-Divergenz gegeben einer Menge an gewichteten Abtastungen x_k mit einer Normalverteilung approximiert. Um den Rechenaufwand des Filters beherrschbar zu halten, wird das zweite Problem angegangen, indem alle Mischkomponenten eines Blickziels mit einer einzelnen Normalverteilung approximiert werden. Dies entspricht dem Vorgehen des *generalisierten pseudo-Bayesschen Schätzers zweiter Ordnung* (engl. generalized pseudo-Bayesian estimator of second order (GPB2)) [15].

3 Bestimmung des Zieltyps mittels semantischer Segmentierung im Bild

Da bei der Modellierung in 2D-Fahrzeugkoordinaten nur der Blick-Gierwinkel verwendet wird, geht bei der zuvor vorgestellten Methode der Blick-Nickwinkel, und damit die Höhe des Fixationspunktes (engl. Point of Regard (PoR)) über dem Boden, verloren. Somit werden Blickmessungen potenziell mit Objekten verknüpft, ohne dass eine tatsächliche Übereinstimmung besteht. Um solche Falschzuordnungen zu detektieren, ist eine Möglichkeit, die Höhe des Blickstrahls am PoR zu überprüfen. Diese Vorgehensweise weist allerdings den Nachteil auf, dass die Höhe eines Objekts ebenso unbekannt sein kann. Stattdessen versuchen wir, den PoR in Fahrzeugkoordinaten ins Bild der Fahrzeugkamera rückzuprojizieren. Der PoR x^* in Fahrzeugkoordinaten ergibt sich als Mittelwert derjenigen Normalverteilung mit dem größten Gewicht

$$x^* = \mu_{i^*} \quad \text{mit} \quad i^* = \text{argmax}_i(\pi_i). \tag{5}$$

Die Höhe des Punkts wird über den gemessenen Blick-Nickwinkel und die Position von x^* im Fahrzeugsystem bestimmt. Anschließend wird dieser 3D-Punkt in das Bild rückprojiziert. Da die Pixelposition alleine jedoch keine Information liefert, wird der PoR im Bildraum mit einer semantischen Segmentierung des Bildes kombiniert. Hierzu verwenden wir das auf dem Cityscapes Datensatz [17] vortrainierte Faltungsnetz aus [18]. Die Art des Blickziels kann nun beispielsweise anhand des Pixellabels an der Stelle der Rückprojektion bestimmt (siehe Abbildung 3) und mit der Objektkategorie aus der Objektliste abgeglichen werden.

In [19] wird argumentiert, dass bei kleiner werdendem Aufmerksamkeitsbereich um den PoR herum, Kategorien mit steigendem proportionalem Anteil das tatsächliche Fixationsziel sind, während Kategorien mit sinkendem Anteil tendenziell nur in den Bereich um den PoR fallen,

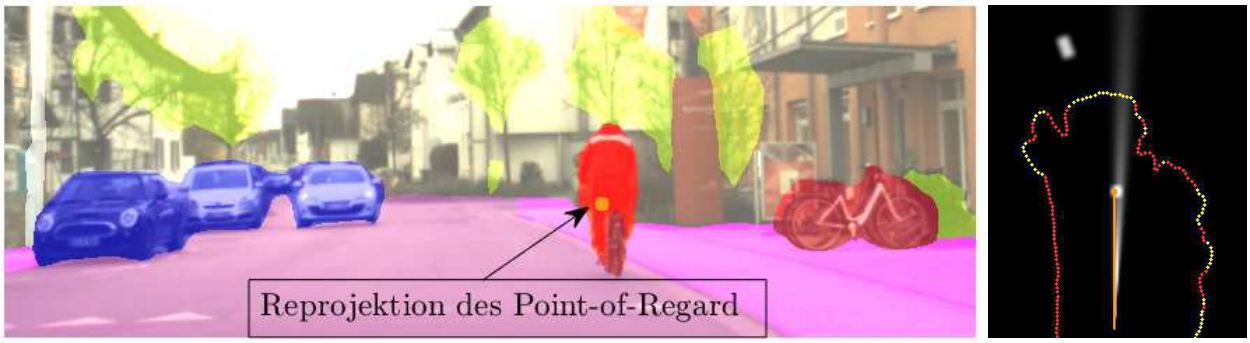


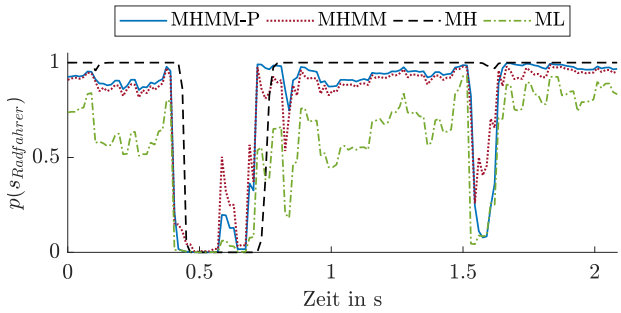
Abbildung 3: Bild der Fahrzeugkamera, überlagert mit dem Ergebnis der semantischen Segmentierung und der Rückprojektion des PoR aus dem MHMM Filter. Rechts die Visualisierung des PoR in Fahrzeugkoordinaten zusammen mit Freiraum-Spline, Objekten und Blickmessung.

da sie sich in der Nähe des eigentlichen Blickziels befinden. Diese Argumentation nutzen wir für eine weitere Möglichkeit zur Bestimmung des Zieltyps. Um den PoR werden Kreise mit den Radien 110 px und 20 px gezogen (entspricht bei der verwendeten Kamera in etwa 10 ° bis 2 ° Blick-Öffnungswinkel) und die proportionalen Anteile jeder Label-Kategorie $p_{k20}, p_{k110},\ k \in m$ bestimmt, wobei m die Anzahl an Kategorien ist. Der Zieltyp K ist dann diejenige Kategorie mit dem stärksten prozentualen Anstieg, d.h. $K^* = \text{label}(\text{argmax}_k(p_{k20}/p_{k110}))$ gegeben, wobei $\text{label}(k)$ das semantische Label der Kategorie k beschreibt.

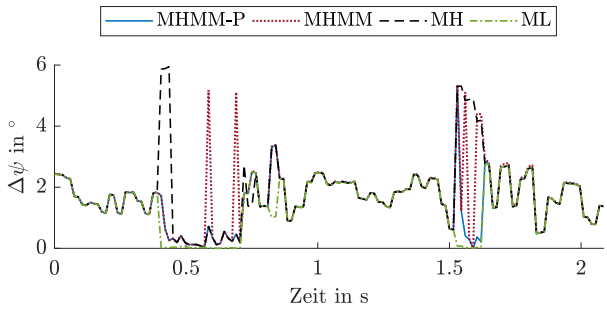
4 Ergebnisse

4.1 Diskussion der Tracking-Ergebnisse anhand eines ausgewählten Szenarios

Wie eingangs erwähnt, ist es bisher nicht möglich, automatisiert valide Ground Truth Informationen darüber zu erhalten, welche Objekte der Fahrer tatsächlich visuell fixiert. In [14] werden daher händische Annotationen von Fixationen und Fixationszielen in den Daten durchgeführt. In dieser Arbeit soll anhand eines ausgewählten Abschnitts aus realen Messdaten, in dem der Fahrerblick klar zwischen zwei Blickzielen wechselt, das MHMM-Tracking argumentativ mit den degenerierten Versionen eines MH-Modells und eines Maximum-Likelihood-Modells (ML) verglichen werden. Im MH-Modell werden die Übergangsparameter von einem Objekt zu einem anderen zu null gesetzt, was in einem separaten Filter für jede Hypothese resultiert. Das ML-Modell besteht dagegen lediglich aus der Multiplikation der einzelnen Messmodelle. Außerdem wird der zusätzliche Nutzen einer komplexen räumlichen und zeitlichen Modellierung der Übergangswahrscheinlichkeiten hervorgehoben werden (das entwickelte Modelle wird hier abgekürzt mit MHMM-P). Das ausgewählte Szenario dauert 2 s und ist in Abbildung 3 dargestellt: in einem Wohngebiet fährt vor dem Fahrer ein Fahrradfahrer während ein anderes Fahrzeug entgegenkommt. In Abbildung 4a sind die Verläufe der Wahrscheinlichkeit $p(s_i)$ des Fahrradfahrers aus jedem der vier Modelle dargestellt und in Abbildung 4b die Abweichung des gemessenen Blick-Gierwinkels zur Richtung des jeweils wahrscheinlichsten Blickziels. Im direkten Vergleich sind folgende Eigenschaften der Modelle zu beobachten: Die ML-Schätzung ist im Vergleich zu den Filteransätzen anfällig für verrauschte Messungen, was eine inhärente Eigenschaft und die grundlegende Motivation für Bayessche Filter ist. Der Schätzungsverlauf des MH-Modells weist dagegen deutlichen Tiefpasscharakter auf, da Sprünge zwischen Blickzielen im direkten Vergleich verzögert auftreten oder sogar unterdrückt werden. Aufgrund der nicht vorhandenen Abwägung zwischen Fixation und Sakkade konvergiert die Wahrscheinlichkeit des „besten“ Modells zu Eins und unterdrückt die Wahrscheinlichkeit für mögliche Alternativen.



(a) Wahrscheinlichkeit $p(s_i)$, dass der Radfahrer vom Fahrer während der ausgewählten Sequenz angeschaut wird.



(b) Abweichung des gemessenen Blick- Gierwinkels ψ zur Richtung des jeweils wahrscheinlichsten Blickziels.

Abbildung 4: Tracking Ergebnis für verschiedene Abwandlungen des vorgestellten Modells.

Aus diesem Grund ist ein geeigneter Regularisierungsterm notwendig [15]. Der vorgeschlagene MHMM-Tracking-Algorithmus überwindet beide der genannten Probleme: Indem er die Möglichkeit verfolgt bei dem aktuellen Objekt zu bleiben, ist die Schätzung glatter als das Messmodell alleine und da er gleichzeitig die Möglichkeit für Sprünge zu anderen Zielen ermöglicht, ist das Modell in der Lage, der flexiblen und schnellen Dynamik des menschlichen Blicks zu folgen. Zu guter Letzt wird die Schätzqualität durch eine zeit- und szenenabhängige Modellierung der Übergänge mittels der Faktoren α und β verbessert (MHMM-P-Modell). Gerade direkt nach einem Sprung in der Wahrscheinlichkeit ist ersichtlich, dass das Modell den Kontrast zwischen den Wahrscheinlichkeiten für verschiedene Blickziele nochmal erhöhen kann.

Betrachtet man die Gierwinkelabweichungen, so ist erkennbar, dass alle Modelle für den Großteil der Sequenz in der Schätzung des Blickziels übereinstimmen (Fixation des Radfahrers), da die Abweichungen praktisch identisch sind. In diesen Abschnitten betragen die Abweichungen in etwa zwischen $0,5\,°$ und $2,5\,°$, was den Annahmen über Messgenauigkeiten und Öffnungswinkel des scharfen Sehens entspricht. Das bedeutet jedoch auch, dass das Messmodell (ML) bereits eine starke lokale Vorgabe über den Ort des Blickziels macht und die Tracking-Modelle vor allem die Gewichte π_i beeinflussen. Während der Phasen, in denen nicht der Radfahrer Ziel des Blicks ist, sondern der Freiraum-Spline, entspricht das ML-Modell praktisch der Messung selbst und beschreibt den Schnitt des Blickstrahls mit dem Freiraum-Spline. Auch bei den anderen Modellen sind die Abweichungen zwischen $t = 0,4\,\mathrm{s}$ und $t = 0,7\,\mathrm{s}$ tendenziell geringer, da im Falle einer Blickfolge auf den Freiraum-Spline kaum Annahmen über die Bewegung des Blicks gemacht werden. Beim MH-Modell ist erneut der Tiefpasscharakter erkennbar, da die Winkeldifferenz nach einer Sakkade, einem Wechsel von einem Blickziel zum anderen, erst verzögert korrigiert wird. Schließlich sind beim MHMM-Modell ebenfalls einzelne Fehlschätzungen erkennbar.

4.2 Verlauf des Blickziels im Bildbereich

Bei der Bestimmung des Zieltyps mittels semantischer Segmentierung im Bild vergleichen wir die Reprojektion des MHMM-P-Filters mit der rein bildbasierten Methode zur Bestimmung des PoR aus [12]: Im Tiefenbild (gemessen von einer Stereokamera) wird die Tiefe des rückprojizierten Blickstrahls mit der gemessenen Tiefe des entsprechenden Pixels im Bild verglichen. Die Stelle der geringsten Tiefendifferenz entspricht dem PoR. In Abbildung 5 ist das semantische Label des Blickziels über der Zeit dargestellt. Hierbei werden einerseits die beiden Methoden zur Bestimmung des PoR (MHMM-P und Stereo) und die beiden Methoden zur Bestimmung des Zieltyps (Pixel-Label an der Stelle des PoR und die Methode aus Abschnitt 3 in Anlehnung an [19]) verglichen. Zunächst ist ersichtlich, dass der rein bildbasiert bestimmte Zieltyp für beide

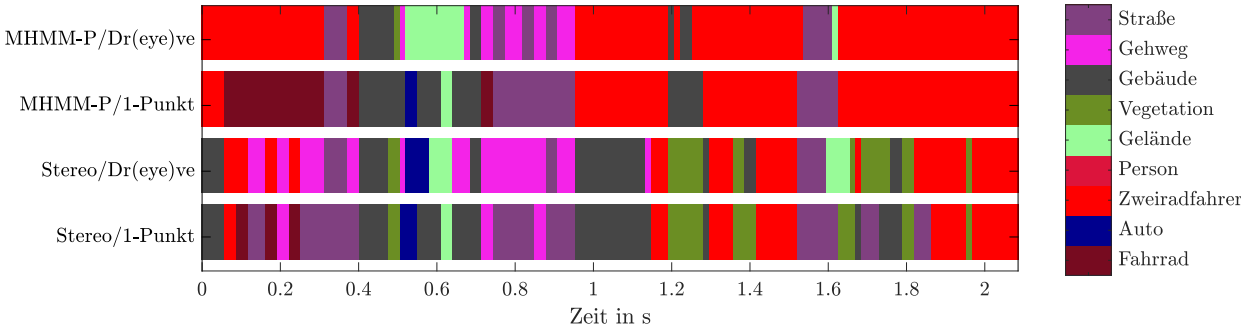


Abbildung 5: Label des Blickziels über der Zeit. MHMM-P: Der PoR stammt aus der Rückprojektion der Filterschätzung in Fahrzeugkoordinaten. Stereo: Der PoR stammt aus der Reprojektion des Blickstrahls in die Tiefenkarte; Dr(eye)ve: Zieltyp über den prozentualen Anstieg des Flächenanteils um den PoR in Anlehnung an [19]. 1-Punkt: Zieltyp entspricht dem Label des PoR-Pixels.

Abbildung 6: PoR im Vergleich, links bei $t = 0,72\,\mathrm{s}$ bis $0,76\,\mathrm{s}$, rechts bei $t = 1,2\,\mathrm{s}$. Orange: Rückprojektion des MHMM-P-Modells. Weiß: Schätzung über Tiefenbild. Zu diesen Zeiten schätzt das Tracking-Modell, dass der Fahrer den Radfahrer fixiert. Im Bildbereich wird unter Berücksichtigung des Blick-Nickwinkels ersichtlich, dass diese Aussage unter Umständen nicht korrekt ist und gegebenenfalls korrigiert werden muss.

Bestimmungsmethoden recht wechselhaft ist. Hieraus lässt sich keine Fixation auf ein Objekt ableiten, da die zusammenhängenden Abschnitte kaum mehr als 100 ms betragen. Dies ist nicht verwunderlich, da die Bestimmung des PoR auf den rauschbehafteten Blickmessungen basiert und keine Modellannahmen getroffen wurden. Im Gegensatz dazu kann aus der Rückprojektion des Filterergebnisses zuverlässig und zusammenhängend der Fahrradfahrer oder das dazugehörige Fahrrad als Blickziel extrahiert werden. Dies liegt zum einen daran, dass der Blick-Gierwinkel über das Tracking korrigiert wird und der PoR im Fahrzeugkoordinatensystem auf den Fahrradfahrer „gezogen“ wird, zum anderen passt jedoch auch der Blick-Nickwinkel zum Fahrradfahrer. Bei $t = 0,72\,\mathrm{s}$ bis $0,95\,\mathrm{s}$ widersprechen sich die Schätzungen des Trackings in Fahrzeug- und Bildkoordinaten jedoch. Wie Abbildung 6 zeigt, ist hier fragwürdig, ob der Blick dem Radfahrer zugeschrieben werden kann oder nicht. Zum Zeitpunkt $t = 1,2\,\mathrm{s}$ stimmen die Schätzungen zwar auch nicht überein, im Bildbereich wird jedoch ersichtlich, dass es sich womöglich nur um einen Fehler der Blick-Nickwinkelmessung handelt.

5 Fazit

In dieser Arbeit wurde ein MHMM-Tracking-Algorithmus zur Schätzung des Aufmerksamkeitsziels des Fahrers vorgestellt. Dieser beinhaltet im Vorhersageschritt sowohl Annahmen über Objektdynamik als auch über Blickverhalten. Die Fixationswahrscheinlichkeit jedes potentiellen Ziels resultiert direkt aus der probabilistischen Beschreibung, wodurch, im Gegensatz zu bestehenden Methoden, auch eine Abwägung zwischen mehreren Hypothesen erfolgt. Erste Evaluierungsergebnisse zeigen, dass die Methode grundsätzlich in der Lage ist, das Blickziel des Fahrers im zweidimensionalen Fahrzeugkoordinatensystem zu bestimmen. Über die Rückprojektion in die Fahrzeugkamera lässt sich das Ergebnis des Filters mit dem semantischen Label an der rückprojizierten Stelle verknüpfen und gegebenenfalls korrigieren.

In zukünftigen Arbeiten soll daher verstärkt auf die Kombination von Fahrzeug- und Bildkoordinaten eingegangen werden. Hierbei gibt es mehrere Ansatzmöglichkeiten: Die Schätzung des PoR in Bildkoordinaten kann über ein eigenes Trackingmodell unter Verwendung von robuster Objektdetektion und Objekttrackings sowie dichten Stereobildern verbessert werden. Robuste Objektdetektionen im Kamerabild könnten nicht nur pragmatischer sein als die detaillierte Darstellung in semantisch segmentierten Bildern, sondern versprechen zudem die Möglichkeit zur Verbesserung der radarbasierten Objektliste. Die Fusion des Aufmerksamkeitsbereichs basierend auf unterschiedlichen Modellen und die Bestimmung des Zieltyps im Bild ist über die Verwendung der Metrik *Intersection over Union (IoU)* denkbar.

Letztlich ist die größte Schwierigkeit jedoch eine ausführliche, sinnvolle und valide Auswertung und der Vergleich verschiedener Modelle. Da Methoden basierend auf berührungslosen kamerabasierten Eye-Tracking-Systemen zwar wie in dieser Arbeit argumentativ untereinander verglichen werden können, der Bezug zu einer vernünftigen Ground Truth jedoch fehlt, ist die Aussagekraft verbesserungswürdig. Die gleichzeitige Verwendung von berührungslosen und tragbaren Eye-Tracking-Systemen inklusive anschließendem manuellen Labeling ähnlich wie in [14, 19] ist zur umfassenden Bewertung der vorgestellten Methoden vermutlich unvermeidbar.

Literatur

[1] J. Walker, "The Self-Driving Car Timeline – Predictions from the Top 11 Global Automakers," 2017. url: techemergence.com/self-driving-car-timeline-themselves-top-11-automakers; Stand 16. Juli 2018.

[2] B. Gain, "Why every car maker should skip level 3," 2017. url: driverless.wonderhowto.com/news/waymo-was-right-why-every-car-maker-should-skip-level-3-0178497; Stand 16. Juli 2018.

[3] R. Klette, "Vision-based Driver Assistance Systems," 2015. url: researchgate.net/publication/272199860_Vision-based_Driver_Assistance_Systems; Stand 16. Juli 2018.

[4] T. Langner, D. Seifert, B. Fischer, D. Goehring, T. Ganjineh, and R. Rojas, "Traffic awareness driver assistance based on stereovision, eye-tracking, and head-up display," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[5] S. J. Zabihi, S. M. Zabihi, S. S. Beauchemin, and M. A. Bauer, "Detection and recognition of traffic signs inside the attentional visual field of drivers," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[6] L. Petersson, L. Fletcher, and A. Zelinsky, "A framework for driver-in-the-loop driver assistance systems," in *Proceedings. IEEE Intelligent Transportation Systems*, pp. 771–776, 2005.

[7] T. Bär, D. Linke, D. Nienhüser, and J. M. Zöllner, "Seen and missed traffic objects: A traffic object-specific awareness estimation," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[8] A. Doshi and M. Trivedi, "Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions," in *IEEE Intelligent Vehicles Symposium, (IV)*, 2009.

[9] M. M. Moniri, D. Merkel, M. Feld, and C. Müller, "Incorporating the Driver's Focus of Attention into Automotive Applications in Real Traffic and in Simulator Setups," in *12th Int. Conf. on Intelligent Environments (IE)*, 2016.

[10] S. Guasconi, M. Porta, C. Resta, and C. Rottenbacher, "A low-cost implementation of an eye tracking system for driver's gaze analysis," in *10th Int. Conf. on Human System Interactions (HSI)*, 2017.

[11] J. Schwehr and V. Willert, "Driver's Gaze Prediction in Dynamic Automotive Scenes," in *20th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC)*, 2017.

[12] T. Kowsari, S. S. Beauchemin, M. A. Bauer, D. Laurendeau, and N. Teasdale, "Multi-depth cross-calibration of remote eye gaze trackers and stereoscopic scene systems," in *IEEE Intelligent Vehicles Symposium Proceedings (IV)*, 2014.

[13] O. Lappi, P. Rinkkala, and J. Pekkanen, "Systematic Observation of an Expert Driver's Gaze Strategy–An On-Road Case Study," *Frontiers in psychology*, vol. 8, p. 620, 2017.

[14] S. Martin and A. Tawari, "Object of Fixation Estimation by Joint Analysis of Gaze and Object Dynamics," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[15] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Theory, Algorithms and Software.* John Wiley & Sons, 2001.

[16] M. Schreier, V. Willert, and J. Adamy, "Compact Representation of Dynamic Driving Environments for ADAS by Parametric Free Space and Dynamic Object Maps," *IEEE Trans. Intell. Transport. Syst.*, vol. 17(2), pp. 367–384, 2016.

[17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] A. Palazzi, D. Abati, S. Calderara, F. Solera, and R. Cucchiara, "Predicting the Driver's Focus of Attention: the DR(eye)VE Project," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Exploiting Maneuver Dependency for Personalization of Driver Assistance Systems

Hien Dang* and Johannes Fürnkranz*

Zusammenfassung:

The problem of personalization in Advanced Driver-Assistance Systems (ADAS) has been addressed in different studies. In general personalization is an approach which aims to develop a system that can adapt on the driver's behavior and thus improve the prediction performance. The individuality of such a system can be used to improve driving experience and comfort. High prediction performance also means that the model can better foresee driver's actions and accordingly raises a warning in case of hazard. In this work, we investigate in the dependency between different maneuvers. We propose an approach to extract information from past maneuver executions and use it as input for predicting impending maneuvers. In particular, we apply our method to predict which gap will be taken at a left-turn scenario where the driver has to wait for an appropriate gap before turning. The results show that by incorporating the previous maneuver execution the prediction performance can be increased by more than 9 % in term of F1 score. By running our approach using different types of past maneuver for a specific application, we can compare the amount of individual information of drivers contained in each type of maneuver.

Schlüsselwörter: Personalization, LSTM, Maneuver Dependency

1 Introduction

It is often the case that users have to adapt themselves to a new system or function to be able to use it. When the users' expectation and preference of a system are not met, their trust in the system will decrease and eventually they may ignore or turn it off . For a standard not-adapting system, there could be a gap between the user's expectation and the outcome of the system. Personalization systems aim at closing this gap.

The problem of personalization is already addressed in different field studies and applications. Recommender systems are an example, in which personalization play a central role. In recommender systems, the data are mostly sequence of actions [1] (i.e. "visiting a website" or "buying an item"...) The problem of personalization in this case is usually formulated as a prediction task of the next action A_i given the past action A_{i-1} or a sequence of the past actions $A_{0...i-1}$:

$$\pi : A_0 ... A_{i-1} \rightarrow A_i \tag{1}$$

*The authors are with Knowledge Engineering Group, TU Darmstadt, Hochschulstraße 10, D-64289 Darmstadt. (e-mails: {hiendang, fuernkranz}@ke.tu-darmstadt.de).

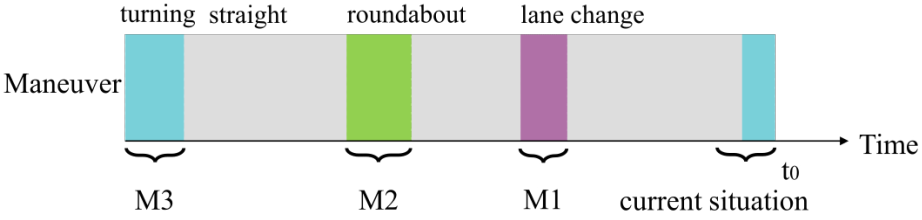


Figure 1: Representation of a test route as consecutive driving maneuvers

In the automotive field, the sequence of maneuvers may by itself reveal very little about the driver. For example, if a driver wants to get to his travel destination, certain maneuvers have to be performed in any case (i.e. entering a roundabout, turning at an intersection...). However, different drivers vary in the way they perform these maneuvers. As a consequence, the problem of personalization must in this case be based on observing how a driver performs the driving maneuvers to infer the driver's preference or to predict how s/he would perform the next maneuver or some specific maneuver of interest. Thus we need to learn a function which maps the driving style $f(.)$ of previous maneuvers to the driving style of an upcoming maneuver:

$$\pi : f(A_0)...f(A_{i-1}) \rightarrow f(A_i) \tag{2}$$

Understanding the driver is a challenging task, there are many factors that can influence a driver on her decision making process. These influencing factors include the driver's habit, her driving skills, her experience, her current physical and mental state and so on. Observing all these factors are complex, and modeling how they will affect the driver and the way s/he drives is also not easy. In this work, we consider a route as a sequence of maneuvers, interleaved with periods of default activities such as driving straight (see Fig 1), and approach the problem of personalization by learning the dependency between maneuver executions. We formalize the problem and show that it is reasonable to exploit the dependency between maneuver executions to adapt the model to individual drivers.

2 Related Work

The idea of personalization is to enable the system to adapt itself to the driver and thus is a key to further improve the safety and driving comfort. The advantages of personalization have already been observed in different applications in automotive field, such as adaptive cruise control (longitudinal driving assistance) or lane changing prediction (lateral driving assistance). Personalization is often combined with the idea of driving style recognition, since both problems address the same issue of understanding the driver behavior.

One of most common approaches is to divide drivers in different groups (e.g., aggressive, moderate and calm). The number of groups varies from two to six, depending on the particular application of interest. The group information could either be used directly to assign individual recommendation to driver or it could also be used as input to the personalization of some specific functionality of ADAS. In [2], Rosenfeld et al. adapt the driver model of [3] to divide drivers into three groups, and use this information together with demographic features for predicting the driver's preferences in the use of adaptive

cruise control (ACC). The driver groups in this study are determined by rules that are manually defined based on empirical observations in the dataset. In contrast to this, Constantinescu et al. [4] present a data-driven approach which makes use of hierarchical clustering and principal component analysis (PCA) to discover six different driver groups. This approach does not make assumptions about driving styles but separates the driver groups by exploring and finding the intrinsic structures of the dataset. Since this method is unsupervised, the discovered clusters do not have an explicit meaning but one can assign suitable labels to each cluster by looking at the features that describe them.

Another approach for identifying driving style is collect and analyze self-reports from drivers [5, 6]. The drivers are asked to complete a driving style questionnaire, based on which different driver groups are formed. This kind of approach, however, suffers from subjective perceptions about driving style. For example, the perception of aggressiveness or calmness may vary from driver to driver.

In many applications in the automotive field, the personalization problem can be reduced to identifying the individual gap acceptance of the driver. In [7], Butakov et al. propose an approach to personalize the lane change prediction by modeling the longitudinal adjustment behavior and the gap acceptance of driver. The gap at a lane change maneuver is defined as the space between the leading vehicle and the following vehicle at the target lane. For adaptive cruise control, an adaptive system will automatically adjust the distance to the leading vehicle to match the driver's preference. Another application of gap acceptance is to predict the decision of the driver at an intersection where driver has to wait for a appropriate gap to perform a turning maneuver [8].

3 Approach on Personalization

Driving style is an abstract concept that is not directly observable from the data. What we can observe is the reflection of driving style on some measurable signals like speed, acceleration, etc. A survey conducted by Martinez et al. [9] shows that the choice of input signals to use for detecting driving style variates from application to application. In general the input signals could be divided into three main groups: vehicle dynamics, energy consumption and personality traits. In this work we focus on the first category of input signal since it is more general and always available in all maneuver execution.

3.1 Problem Formulation

Maneuver Execution Given a situation S_i and a driver D_j, the driving maneuver at this situation could be formulated as a function of S_i and D_j

$$M_i = f(S_i, D_j + \epsilon_i) \tag{3}$$

where S_i represents all environmental factors, such as traffic situation or weather, D_j constitutes the driver's dispositional factors. Both S_i and D_j have an impact on how a maneuver will be performed. However, the same driver could behave differently even if the same situation repeats at some other time. Thus, a variable ϵ_i is introduced to capture the fluctuations in the driver's behavior. It can be seen as a noise term of driver D_j while performing a maneuver M_i.

Personalization of a Driver Model Given a driver D_j at a situation S_i, we now want to predict a specific target value y_i for the current situation. y_i could be, for example, the next action of the driver. To personalize the prediction of y_i, we additionally have to consider the influences of the driver $D_j + \epsilon_i$ in this situation, since y_i depends on these both factors:

$$y_i \leftarrow S_i, D_j + \epsilon_i \tag{4}$$

Now we have the same problem as in modeling maneuver execution, namely that the driver's dispositional factors, which influence y_i, are not directly observable. This make it impossible to learn the individual impacts of driver D_j on y_i. The problem is usually reduced to predict y_i given the current situation S_i. This approach comes with an assumption that the individual influence from a driver on y_i is small or the behavior of the drivers in training set is similar to testing set.

Incorporating Previous Maneuvers As mentioned above, in real-world driving data, the drivers' factors are encoded in each maneuver execution M_i. We therefore propose that instead of learning y_i as a function of S_i and D_j, we can learn y_i as a function of S_i and M_{i-1}, where M_{i-1} is a maneuver that was recently performed by the same driver D_j. This temporal restriction of M_{i-1} allows us to approximate the current impact of the driver $D_j + \epsilon_i$ with $D_j + \epsilon_{i-1}$, which was captured in the previous maneuver.

$$\begin{aligned} y_i &\leftarrow S_i, M_{i-1} \\ &\leftarrow S_i, f(D_j + \epsilon_{i-1}, S_{i-1}) \\ &\approx S_i, f(D_j + \epsilon_i, S_{i-1}) \end{aligned} \tag{5}$$

The prediction of y_i can then be written as a function G of S_i, $D_j + \epsilon_i$ and S_{i-1}:

$$\begin{aligned} y_i &= g(S_i, f(D_j + \epsilon_i, S_{i-1})) \\ &= G(S_i, D_j + \epsilon_i, S_{i-1}) \end{aligned} \tag{6}$$

By taking M_{i-1} into account, we are also using the previous traffic situation S_{i-1} for predicting the current situation. With the assumption that y_i does not depend on other situations than the current S_{i-1}, there should be no information of y_i contained in S_{i-1}. G will be forced to learn to extract useful information about the driver from past maneuvers and then how they will affect the decision of the driver in the current situation.

3.2 Extracting Driver's Information

A maneuver execution (M_i) is characterized by a multivariate time series of sensor values:

$$M_i = (x_{i,1}, x_{i,2}, ..., x_{i,n}), \quad n > 0 \tag{7}$$

where n is the length of the time series and $x_{i,t}$ is a column vector of sensor values at time t. It has to be noted that the length of a maneuver varies, depending on the driver and the traffic condition. To learn from this kind of data we either have to convert them into fixed-length feature vectors or make use of a models that can handle arbitrary length.

In this work, we used two approaches to extract a driver's information from the last executed maneuvers. The first approach extracts features from the time series by computing the statistical information (i.e. minimum, maximum and standard deviation) from

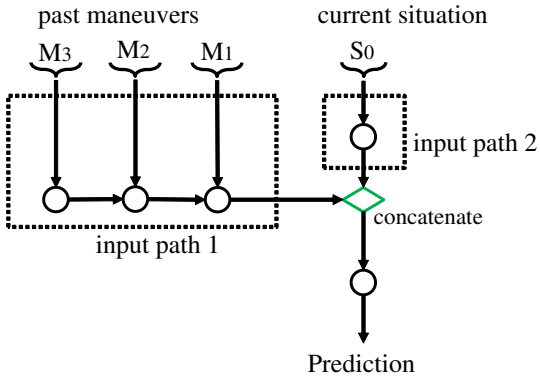


Figure 2: Proposed architecture to capture the driver information from previous maneuver executions and use that for predicting the current situation

each sensor. The second approach makes use of recurrent network layers and takes the whole maneuver execution as an input. While the idea of the first approach is widely used in the literature for extracting driver's information, the second one does not make any assumption about the statistical values but learns to extract useful information direct from raw data.

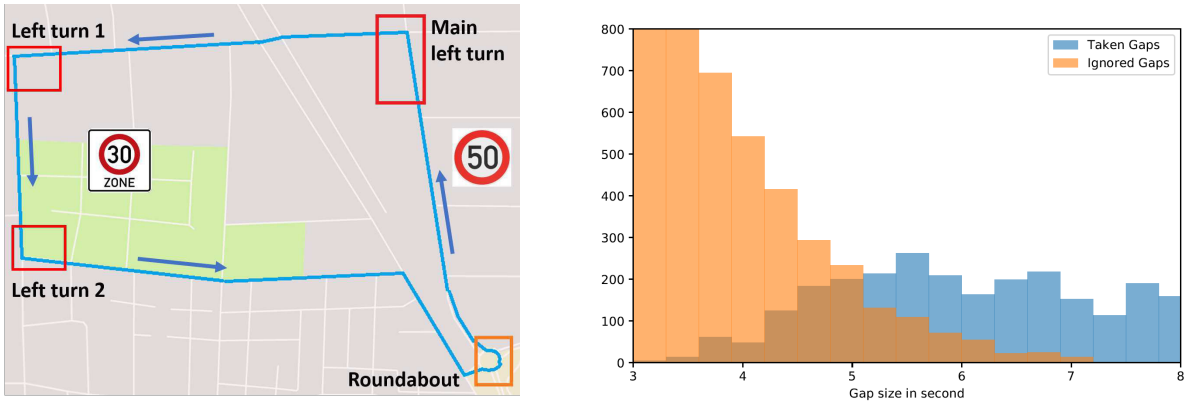
4 Modeling Maneuver Dependency

We have now formulated the prediction of y_i as a function of the current situation S_i and the previous maneuver M_{i-1}. By incorporating the previous maneuver M_{i-1} as input, the individual influence of the current driver will also be considered for predicting y_i. This can be extended by incorporating k last maneuvers that were performed by this driver $(M_{i-1}, M_{i-2}...M_{i-k})$. Fig. 2 shows an example of a network that additionally uses $k = 3$ last maneuvers as features. To validate this concept we test it with $k = 1$ which is theoretically easier to train and requires less data. We use a neural network to model the function G, whose objective is to predict the taken gaps.

4.1 Network Architectures

We design the network using two input layers separately, one takes the current situation (S_i) as input and the other is used for capturing useful information from the previous maneuver (M_{i-1}). Each of these two input layers are then followed by hidden layers and form two separate paths. For the ultimate purpose of predicting y_i, these two paths are then combined in the deeper layers, followed by further hidden layers and lastly the output layer. The whole network is trained by back-propagating the classification error.

Using this network structure allows us to customize the two input paths individually and make it possible to apply different regularization strategies on each path. This is useful to deal with overfitting since it could be a problem when using previous maneuvers. Furthermore, the input path for capturing past maneuver M_{i-1} could be removed from the architecture, which results the common approach that learn to predict y_i directly from S_i.



(a) Test route for recording data. The roundabout and left turn are marked in orange and red

(b) Histograms of taken and ignored gaps

Figure 3: Visualization of driving route and statistics of gaps at the left-turn situation

4.2 Extracting Driver Information with Neural Network

As mentioned in 3.2, extracting driver information from M_{i-1} could be done in two different ways. We modeled the first approach by configuring the first input path using fully connected layers that take statistical values from the last maneuver execution as input. The function G could then be written as $G(S_i, h_s(M_{i-1}))$ or $G_s(S_i, M_{i-1})$ for short. Here h_s compute the statistic information of the given maneuver. The second approach makes use of a recurrent layer to capture the maneuver execution. In particular, we use Long short-term memory networks (LSTM) [10] for capturing past maneuvers M_{i-1}. LSTM Networkss have proven to be quite successful for sequence learning problems [11,12]. The function G in this case is formalized as $G(S_i, h_{lstm}(M_{i-1}))$ or $G_{lstm}(S_i, M_{i-1})$ for short. Here h_{lstm} depicts the LSTM layer that return hidden representation of the given maneuver M_{i-1}.

By using different type of past maneuvers, the performance of the trained model can be used for measuring which maneuver contains more useful information about the driver. Such maneuvers can then be used to characterize the driver and serve as input for further adaptive systems.

5 Experiments

Data set The data used in this work were collected from 32 drivers which cover a wide range of ages and driving experiences. From each driver we recorded 30 rounds of driving on a pre-defined route. The data set was collected in real-world traffic (see Fig 3a). The chosen route is a common urban route which requires the driver to perform different maneuvers like roundabout, left turn at 50 km/h zone (main left turn) , intersection with left yields to right, left turn at 30 km/h zone (left turn 1 and 2), etc. For our experiments, we focused on the two most complex maneuvers: roundabout and left-turn at 50 km/h zone.

For capturing the driving style we use four signals as features for describing a maneuver execution, which include speed, longitude acceleration, latitude acceleration and steering wheel speed. As features for describing the intersection situation we additionally use the

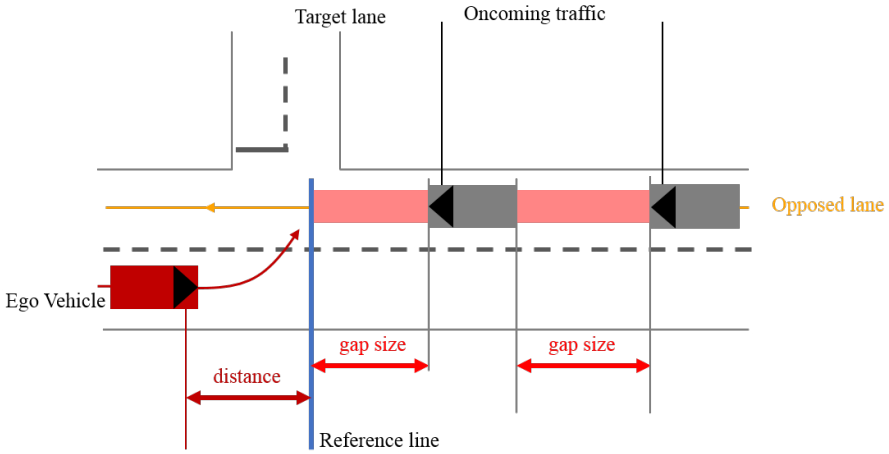


Figure 4: Illustration of the left turn situation with two possible gaps to take

position of the ego vehicle to determine the distance to the middle of the intersection.

Gap Acceptance at Left-Turn Maneuver In an unsignalized intersection scenario, in which the incoming traffic has the right of way. We observe that the preference of the driver in choosing gaps differs from driver to driver. Gaps that are in range from approximately 3 to 7 seconds could either be taken or ignored, depending on the driver's preference, as shown in Fig. 3b. In such a scenario, the problem of predicting the gap acceptance of driver should not only consider the current traffic situation but also the driver's individual preferences.

We detect the oncoming traffic using the equipped front-radar: their the relative position, speed and size. Based this information, each potential gap for turning left is computed. An illustration of the application at an intersection is shown in Fig. 4, where the ego vehicle is depicted as a red box and the incoming vehicles are shown in gray. The size of the first gap is the distance between the closest incoming vehicle and the middle of the intersection (reference line). The last gap is computed based on the furthest detected incoming vehicle and the maximum radar range. All gaps between these two gaps are bounded by a leading vehicle and a following vehicle. For each gap we compute the time (t_i) that is available for the driver to turn left: $t_i = s_i/v_i$, where the s_i is the longitudinal size of gap i and v_i is the current speed of the following vehicle that forms gap i.

6 Evaluation

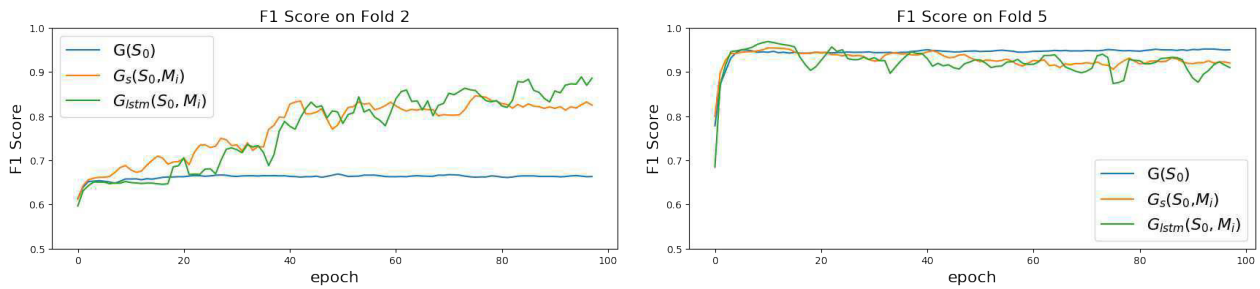
6.1 Evaluation Setup

We evaluate our method using 10-fold cross-validation to construct training and testing sets. The data are split into 10 equal and disjoint folds according to the driver ID. This assures that the model will be evaluated unseen drivers. In total we evaluate four models. As the baseline model, we estimate the best possible threshold that separates the taken gaps from the ignored gaps in the training set. The baseline is then compared to the three models described in the previous sections:

- $G(S_i)$ only use the current situation as input (the first input path is removed).

Table 1: Mean of $F1$ and accuracy score on validation set over 10-fold cross-validation

Model	**F1 Score**	**Accuracy**
Threshold	81.9 %	89.8 %
$G(S_i)$	86.8 %	92.6 %
$G_s(S_i, M_{i-1})$	88.8 %	93.2 %
$G_{lstm}(S_i, M_{i-1})$	91.1 %	94.7 %



(a) Incorporating M_i lead to improvement (b) Overfitting on training data

Figure 5: Learning curve on different validation folds

- $G_s(S_i, M_{i-1})$ uses the statistical values from the roundabout maneuver
- $G_{lstm}(S_i, M_{i-1})$ captures the maneuver execution sequence M_{i-1} using LSTM layer.

6.2 Results of learning from previous maneuvers

The average F1 score and the accuracy of all four models on the validation sets are show in Table 1. Both variants that extract information from M_{i-1} show significant improvements. The overall best score is obtained by G_{lstm}, which uses an LSTM layer for capturing M_{i-1}. Various network configuration are tested with different regularization parameters applied to each input path individually. In all settings, using extra information from M_i always leads to improvement of the prediction performance.

The improvement observed for $G_s(S_i, M_{i-1})$ confirms that the statistical values of a maneuver execution contain information about driving style. This is also the traditional approach for characterizing a driver. The additional improvement obtained by $G_{lstm}(S_i, M_{i-1})$ indicates that there is more information about the driver, which can be extracted by considering the raw maneuver execution instead of only using its statistical values.

6.2.1 Overfitting

Reviewing the cross-validation results, we observe the improvement in terms of F1 score and accuracy of $G_s(S_i, M_{i-1})$ and $G_{lstm}(S_i, M_{i-1})$ over $G(S_i)$ in eight out of ten folds. The maximal improvement reaches 18.5 % in F1 score which translates to 9.4 % accuracy. Fig 5a shows the validation score of all three models in one of these eight folds. Here we can see that $G(S_i)$ gets stuck and its best score is quite low, whereas both, the G_s

Table 2: Comparison on the impact of extracting information from different maneuvers

Used Maneuver	**F1 Score**	**Accuracy**
Lefturn 1	90.2 %	93.7 %
Lefturn 2	90.0 %	93.6 %
Roundabout	91.1 %	94.7 %

and G_{lstm} models benefit from the extra input M_{i-1} and reach much higher accuracy. In Fig 5b we observe the first slight effect of overfitting as the score of G_{lstm} and G_s keeps fluctuating and overfit on training data after 40 epochs. To deal with such effect, early stopping was also used for training the final model.

6.2.2 Evaluation using different past maneuvers

As mentioned in Section 5, our test route also consists of other maneuvers that can be used as input for the model. In this section, we evaluate and compare the impact of using two other left turn maneuvers as M_{i-1}in predicting taken gaps. The results are produced using the LSTM architecture in Section 2 — $G_{lstm}(S_i, M_{i-1})$. Table 2 shows the F1 and Accuracy scores on predicting the taken gap at main left turn scenario. Overall, the system still benefits the most when it extracts information from roundabout maneuvers. Since we are predicting taken gaps at a left turn maneuver, the first intuition would be the system should gain information of we use other past left turn as input. However, we have to note that these two left turn maneuvers are located in a 30-zone with low traffic, whereas the main left turn is located at 50-zone with high-traffic roads. The behavior of the these two left turns are thus different from the main left turn. On the other hand, the roundabout maneuver is a complex one which is longer and require more inputs from driver, therefore there should be more information that can be extract from a roundabout maneuver that is helpful for predicting taken gap

7 Conclusion

In this work we proposed a new approach to learn the dependency between maneuver execution, namely to extract the information about driver behavior and style, and use it to improve the performance of the prediction task in driver assistance systems. We implement our approach using neural networks as building blocks and empirically evaluate the model in a left-turn scenario using on-road data. The results show that the model is able to extract the driver impact from the past maneuver executions and can use it to improve the prediction by more than 9 % in terms of F1 score.

In general our approach can be used to evaluate the amount of information from driver that can be extracted from a maneuver execution. We compared the effect of using different types of past maneuver on predicting taken gap. The results show that roundabout maneuvers contain more individual information from driver. From that the gap prediction task benefit more than from using turning maneuvers in 30-zone.

References

[1] C. C. Aggarwal, *Recommender Systems: The Textbook*. Springer-Verlag, 2016.

[2] A. Rosenfeld, Z. Bareket, C. V. Goldman, D. J. Leblanc, and O. Tsimhoni, "Learning drivers behavior to improve adaptive cruise control," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 19, no. 1, pp. 18–31, jan 2015.

[3] P. S. Fancher and J. R. Sayer, "A Comparison of Manual Versus Automatic Control of Headway as a Function of Driver Characteristics," in *Proceedings of the 3rd Annual World Congress on Intelligent Transport Systems*, 1996.

[4] Z. Constantinescu, C. Marinoiu, and M. Vladoiu, "Driving Style Analysis Using Data Mining Techniques," *International Journal of Computers Communications & Control*, vol. 5, no. 5, p. 654, dec 2010.

[5] M. Ishibashi, M. Okuwa, S. Doi, and M. Akamatsu, "Indices for characterizing driving style and their relevance to car following behavior," in *Proceedings of the SICE Annual Conference*. IEEE, sep 2007, pp. 1132–1137.

[6] D. J. French, R. J. West, J. Elander, and J. M. Wilding, "Decision-making style, driving style, and self-reported involvement in road traffic accidents," *Ergonomics*, vol. 36, no. 6, pp. 627–644, jun 1993.

[7] V. Butakov and P. Ioannou, "Personalized Driver/Vehicle Lane Change Models for ADAS," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4422–4431, oct 2015.

[8] P. J. Cooper and Y. Zheng, "Turning gap acceptance decision-making: the impact of driver distraction," *Journal of Safety Research*, vol. 33, no. 3, pp. 321–335, oct 2002.

[9] C. M. Martinez, M. Heucke, F. Y. Wang, B. Gao, and D. Cao, "Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, mar 2018.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[12] H. Q. Dang, J. Fürnkranz, A. Biedermann, and M. Hoepfl, "Time-to-lane-change prediction with deep learning," in *Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC-17)*. IEEE, oct 2017, pp. 1–7.

Modeling of Drivable Free Space with Fused Camera Data for Autonomous Driving

Oliver Speidel, Fabian Gies and Klaus Dietmayer*

Abstract: The detection and modeling of drivable free space is a major challenge for autonomous driving. A novel approach is to use high resolution cameras especially in order to get semantic information of the environment. However, most approaches fail to use a suitable representation of the free space which is indispensable for subsequent processes like the behavior and motion planning. This paper presents a generic framework for detecting and compactly modeling free space based on fused camera data. For the detection, the disparity image and pixel classification of a stereo-camera are used. Based on that, a new continuous semantic free space model including temporal filtering is introduced. First evaluations show impressive results even in challenging scenarios.

Keywords: B-Splines, Camera Processing, Free Space Modeling, Temporal Filtering

1 Introduction

In the context of automated driving, the estimation of drivable areas is an essential and intensively explored topic. A common strategy is to use a high-precision digital map in which road areas are stored. By localizing the vehicle with a high-precision Global Positioning System (GPS), a safe trajectory can be estimated based on the map and an object tracking algorithm [4, 15]. Nevertheless, these concepts, which heavily depend on digital maps, are not flexible enough to overcome complex scenarios or situations with poor GPS reception, for example in dense urban areas. So for autonomous driving it is inevitable to detect the dynamic drivable space while driving.

In order to do so, a novel approach is to use camera based sensors in combination with Convolutional Neural Networks (CNNs) as a pixel classification of the scene can be obtained [13]. Beside this, to overcome the missing 3D information in a mono-camera setup, there are concepts utilizing depth information yielded by a stereo-camera [14]. However, an appropriate representation of the free space is required in order to plan the vehicles motion for drivable areas. A common approach is to use grid maps where each cell has a probability to be occupied [4]. An according extension are Digital Elevation Maps (DEM) which additionally store height information of a cell [12]. Another strategy is to use so called Stixels, which model objects by vertical structures with according height extensions [8]. To further reduce the complexity for motion planning algorithms there are more compact and efficient mathematical models that explicitly represent the free space

*The authors are with the Institute of Measurement, Control, and Microtechnology at Ulm University, Albert-Einstein-Allee 41, 89081 Ulm (e-mail: firstname.lastname@uni-ulm.de).
The research leading to these results was conducted within the Tech Center a-drive. Responsibility for the information and views set out in this publication lies entirely with the authors.

boundary. Some novel approaches use polynomials, piecewise polynomials or B-splines to represent the boundary. In their fundamental form, these models also allow to use common temporal filtering methods as, for example, the Information or Kalman Filter (KF) for B-splines [9, 5].

To improve the accuracy and robustness of the obtained information, a multi sensor setup in combination with information fusion algorithms can be utilized. A generic approach based on a grid map is presented in [4], where radar and lidar information are fused, however, only free space and occupied cells are distinguished. A sensor specific fusion based on Stixels is used in [6]. By that, a pixel classification of a Neural Network (NN) is fused with the result of an object detection algorithm applied on a disparity image in order to detect unexpected objects.

In contrast, this paper presents a generic concept based on fused camera data in order to model free space. Therefore, the information obtained by a disparity image of a stereo-camera and a pixel classification is fused to estimate the drivable free space in vehicle coordinates. Additionally, a suitable spline model is developed which is not only able to smoothly represent the position of a free space boundary but also holds semantic information. Furthermore, a method for temporally filtering the spline model is introduced. The remainder of this paper is structured as follows. In Section 2 the developed framework is presented. Section 3 shows results based on real world data. Finally, a short summary and outlook is given in Section 4.

2 Free Space Modeling

2.1 Data Acquisition

As input data, the disparity image $\mathcal{C}$ of a stereo-camera is utilized to get depth information of the scene. Further semantic information is obtained by a NN which yields a pixel classification $\mathcal{S}$ of the image. Exemplary input data is depicted in Figure 1.

Generation of a Digital Elevation Map
The depth information is used to generate a DEM and to estimate labels for the single cells by fitting adjacent planes and a street plane into the DEM. The according algorithms are described in [12].In general, each cell receives unary and binary probabilities for belonging to a street or adjacent surface. The unary probability is based on the height of the cell and the binary probability is based on the relative height of neighboring cells. This concept was extended by the label *obstacles*. The according unary *obstacle* probability is formulated as,

$$\begin{aligned} P(z, z_{min}, z_{max}, \lambda_o, P_{min}, P_{max}) &= (P_{max} - P_{min})(s(z - z_{min}, \lambda_o) \\ &\quad -s(z - z_{max}, \lambda_o)) + P_{min} \,, \end{aligned} \tag{1}$$

where

$$s(\Delta z, \lambda_o) = \frac{1}{(1 + exp(-\Delta z/\lambda_o))} \,. \tag{2}$$

Thereby, P_{min} and P_{max} are the minimum and maximum probability for the label *obstacle* and z_{min} and z_{max} denote the minimum and maximum height of an obstacle. Thus, the

obstacle probability increases for DEM cells with a height between z_{min} and z_{max} above the street plane. The steepness of this gating function is controlled with λ_o. The binary probability is equally distributed.

As a result, each DEM cell is described as $\Omega = [\boldsymbol{x}, \boldsymbol{P}_x, h, \boldsymbol{p}]$, where $\boldsymbol{x} = [x_v, y_v]$ is the position of the cell center, $\boldsymbol{P}_x$ represents the according covariances, h is the height and $\boldsymbol{p}$ contains the label probabilities. In general the label set $\mathcal{L} = \{street, obstacle, background\}$ is applied.

Processing of Scene Labeling

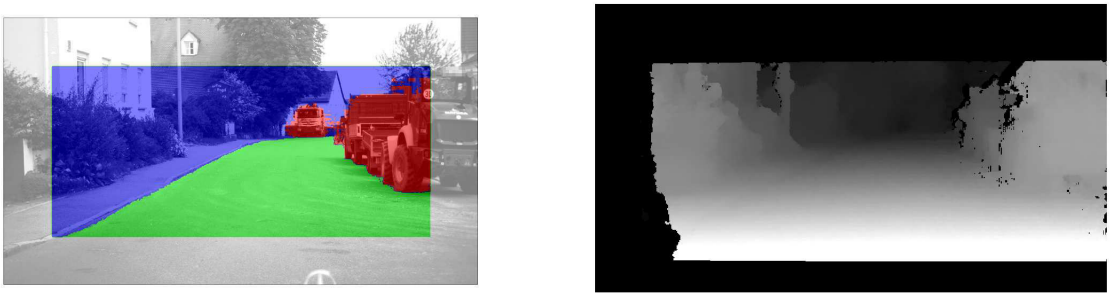
Basically, any pixel classification that provides at minimal the labels in $\mathcal{L}$ can be utilized as input. The algorithm used in this work is described in [13]. In order to reduce noise in $\mathcal{S}$, a morphological opening on each of the probability sets is done followed by a mapping of the labels to $\mathcal{L}$. In the end, the pixel-wise probabilities can be associated to the DEM cells using the stereo depth information. Thereby multiple pixels in the pixel classification image possibly belong to a single DEM cell. Therefore, the mean of all pixel-wise probabilities is used.

Data Fusion

Subsequently, the label probabilities can be fused for each DEM cell independently. Therefore, the single label probabilities for a cell yielded by $\mathcal{C}$ and $\mathcal{S}$ are weighted with a confidence weight w_i. Consequently, the fusion can be calculated by

$$p_i^{\mathcal{F}}(l) = (w_i^{\mathcal{C}}(l) \cdot p_i^{\mathcal{C}}(l) + w_i^{\mathcal{S}}(l) \cdot p_i^{\mathcal{S}}(l)) \cdot w_{norm}\,, \tag{3}$$

where $p_i^{\mathcal{C}}(l)$ and $p_i^{\mathcal{S}}(l)$ represent the probabilities for the label $l \in \mathcal{L}$ yielded by the according sensor and $p_i^{\mathcal{F}}(l)$ is the label probability of the fused DEM which is normalized with w_{norm}. This generic fusion also allows the integration of other information sources as only the weighted probabilities have to be added before normalizing.



(a) Output Scene Labeling (b) Disparity Picture

Figure 1: Input Data Acquired by Existing Data Preprocessing Modules. On the left, the image of a grey-scale camera overlayed by the pixel classification result, where blue denotes the label *background*, green the label *street* and red the label *obstacle*. Thereby, the grey areas are not labeled for performance reasons. On the right, the disparity image is shown, where white pixels encode close data points and black pixels data points far away. Unknown pixels are also encoded with black.

2.2 Free Space Model

In order to be able to model the free space boundary, associated boundary points $\boldsymbol{B} = \boldsymbol{b}_{1...m} = [x_v, y_v, \boldsymbol{p}^t]^{\mathrm{T}}$ have to be extracted of the DEM, where x_v and y_v describe the position and $\boldsymbol{p}^t = [p_s^t, p_{bg}^t, p_{ob}^t]^{\mathrm{T}}$ describes the probability for the transition labels, i.e. the label of the objects beyond the free space boundary. The boundary point estimation is done column wise in the DEM which itself is aligned to the u-columns of the disparity image. Thereby, the last free cell in each column represents a boundary point $\boldsymbol{b}_j$, where $\boldsymbol{p}^t$ is calculated by means of the label probabilities of surrounding cells of $\boldsymbol{b}_j$. This is done by using a mean filter on the fused DEM $\mathcal{M}$ according to

$$p_j^t(l^t = c) = mean(\mathcal{M}_{p(l=c)} \notin street, \boldsymbol{H}_{\boldsymbol{b}_j}) \,, \tag{4}$$

with the boundary point $\boldsymbol{b}_j$ in the center of the mask $\boldsymbol{H}$ and $c \in \mathcal{L}$. Where the size of the mask defines the range of considered neighboring cells. While filtering, the cells labeled as *street* are ignored since only information about the occupied space is required. A schematic illustration of u-columns and boundary points can be seen in Figure 2. For the mathematical model of the boundary, B-splines are utilized. Therefore, the control points $\boldsymbol{C} = [\boldsymbol{c}_1, ..., \boldsymbol{c}_n]$ are estimated based on $\boldsymbol{B}$. In addition, a knot sequence $\boldsymbol{\tau} = [\tau_1, ..., \tau_{n+d}]$ is utilized which is equidistant in the xy-plane and where d represents the order of the spline. In addition to the position, the probabilities for transition labels are approximated by the spline. This is done by adding additional dimensions to the spline for the labels in $\mathcal{L}$. The resulting boundary state model is given as

$$\hat{\boldsymbol{b}}(\tau) = [\hat{x}_v(\tau), \hat{y}_v(\tau), \hat{p}_s^t(\tau), \hat{p}_{bg}^t(\tau), \hat{p}_{ob}^t(\tau)]^{\mathrm{T}} = \hat{\boldsymbol{C}} \cdot \boldsymbol{N}_d(\tau)^{\mathrm{T}} \,, \tag{5}$$

where $\boldsymbol{N}_d(\tau)$ contains the values of the basis functions of the B-spline.

In general, the control points $\boldsymbol{C}$ can be estimated by using a Least Squares Estimator (LSE). With the concept of P-Splines [2] the curvature of the spline can be penalized which enables the generation of smooth boundaries. The according LSE is given as

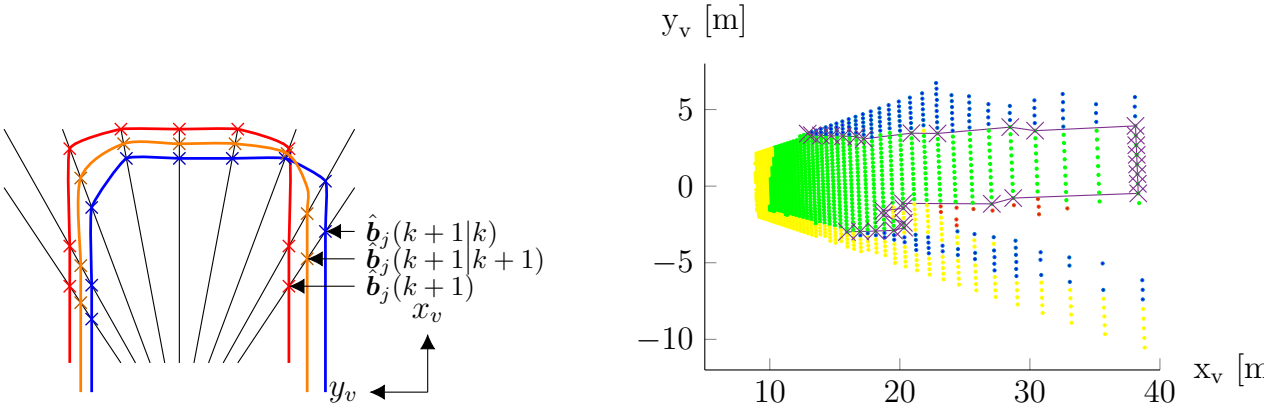
$$\hat{\boldsymbol{C}}(\boldsymbol{B}) = [(\boldsymbol{N_d}^{\mathrm{T}} \cdot \boldsymbol{N_d} + \lambda \cdot \boldsymbol{D_k}^{\mathrm{T}} \cdot \boldsymbol{D_k})^{-1} \cdot \boldsymbol{N_d}^{\mathrm{T}} \cdot \boldsymbol{B}^{\mathrm{T}}]^{\mathrm{T}} \,, \tag{6}$$

where k defines the order of the penalty and λ the intensity.

Thereby, a crucial problem is the association of the boundary points to the corresponding positions τ_i. In [11] and [10] the measurements are assumed to be equidistantly distributed on the spline. In case of camera based sensors this is true for the image plane or respectively the uv-plane. However, this assumption does not hold for complex boundaries in the xy-plane as large distances might be covered by only a few boundary points and thus leads to overshooting behavior of the spline. Therefore, in this work, the positions of the boundary points on the spline are estimated by their Euclidean distance to each other. Thus,

$$\tau_j = \frac{||\boldsymbol{b}_{1,..,j}||}{||\boldsymbol{b}_{1,..,m}||} \cdot \tau_{n+d} \,, \tag{7}$$

where $||\boldsymbol{b}_{1,..,j}||$ denotes the accumulative Euclidean distances from $\boldsymbol{b}_1$ to $\boldsymbol{b}_j$ and τ_{n+d} is the total length of the spline. In areas where the density of measurements is low, pseudo measurements have to be generated in order to prevent ambiguous estimations. Considering the sensor model, the boundary between two boundary points has to be between



(a) Schematic Boundary Points (b) Boundary Points in Downscaled DEM

Figure 2: Exemplary Boundary Points. On the right, exemplary position of boundary points in a downscaled DEM, where blue denotes the label *background*, green the label *street*, red the label *obstacle* and yellow the label *unassigned*. The crosses represent the position of boundary points. On the left, schematic spline models, where the measured, predicted and filtered spline as well as the according boundary points of u-columns are depicted. In addition, 3 corresponding boundary points $\boldsymbol{b}_j$ are labeled.

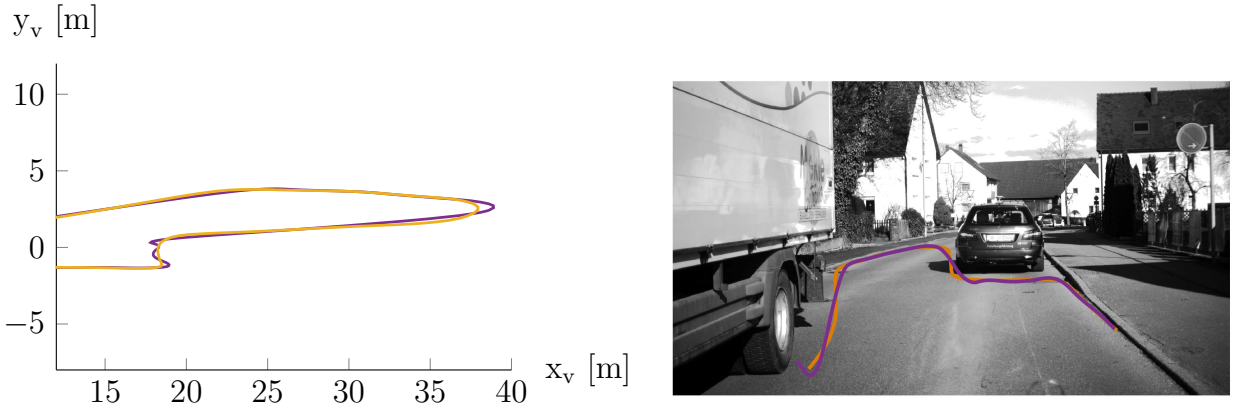
the according DEM columns. Therefore, pseudo boundary points can be estimated by linear interpolation. If the distance between two boundary points exceeds the required minimum distance, an additional boundary point $\boldsymbol{b}_j = \frac{1}{2}(\boldsymbol{b}_{j-1} + \boldsymbol{b}_{j+1})$ is generated. As a result, a smooth boundary spline can be estimated. Examples for both spline models are depicted in Figure 3.

2.3 Temporal Filtering

In order to estimate the boundary up to the margins of the DEM and to improve the robustness of the free space model, a temporal filtering based on a Extended Kalman Filter (EKF) [1] is implemented. Thereby, $\boldsymbol{C}$ is treated as the state and $\boldsymbol{B}$ contains the measurements. Therefore, the association of a predicted boundary point $\hat{\boldsymbol{b}}_j(k+1|k)$ with a measurement $\boldsymbol{b}_j(k+1)$ is done by calculating the intersections of the predicted spline with the DEM columns. As a result, each $\boldsymbol{b}_j(k+1)$ can be easily associated with a $\hat{\boldsymbol{b}}_j(k+1|k)$. Finally, the single filtered boundary points $\hat{\boldsymbol{b}}_{1,\dots,m}(k+1|k+1)$ are calculated independently and consequently the filtered control points $\hat{\boldsymbol{c}}_{1,\dots,n}(k+1|k+1)$ can be estimated. The different boundary points are depicted in Figure 2 and the according prediction of measurements and the subsequent innovation is described in the following.

Prediction

The prediction of the control points is done with the assumption of constant turn rate and velocity (CTRV) which results in a standard prediction step of a EKF. However, the transformation of the position covariances into the measurement space is approximated. This is done by treating the elements of the covariance matrices as additional dimensions of $\hat{\boldsymbol{c}}_i$ and as a result an approximation of the covariances for the position of corresponding boundary points is obtained.



(a) Top View (b) Image Plane

Figure 3: Exemplary scenes for a qualitative comparison between the proposed spline model in the xy-plane (orange) and the state of the art model with equidistant knots (purple) or respectively in the uv-plane. In the top row, the splines transformed into the image plane. In the bottom row, the splines in top view.

Innovation

Before calculating the filter gain, further adoptions are made. The model used for the prediction assumes a static boundary which, obviously, does not hold for dynamic objects. Therefore, the covariances of the boundary points are adapted according to their label. Finally, a Kalman update for the positions can be performed.

The filtering of the labels has to be done separately. For this reason, fixed weightings $w^{\boldsymbol{c}}_{\mathcal{L},j}$ and $w^{\boldsymbol{b}}_{\mathcal{L},j}$ for the previous and measured label are used with $\sum w_{\mathcal{L},j} = 1$. Consequently, the filtered label probabilities are given as

$$\hat{\boldsymbol{p}}^t_j(k+1|k+1) = w^{\boldsymbol{b}}_{\mathcal{L},j} \cdot \boldsymbol{p}^t_j(k+1) + w^{\boldsymbol{c}}_{\mathcal{L},j} \cdot \hat{\boldsymbol{p}}^t_j(k|k)\,. \tag{8}$$

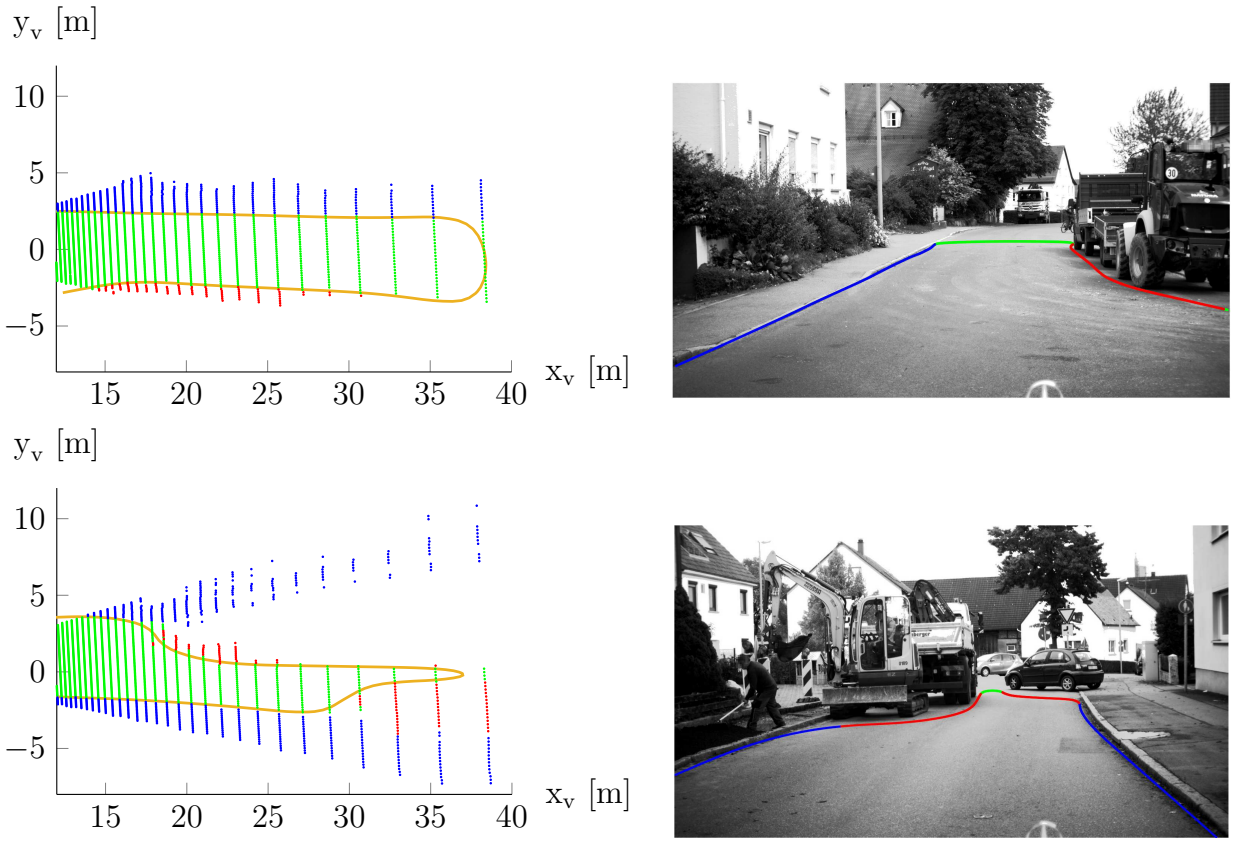
Based on that, the filtered boundary points $\hat{\boldsymbol{b}}_{1,\dots,m}(k+1|k+1)$ can be obtained by combining the corresponding position and labels. Exemplary results are shown in Figure 4. Thus, a temporal filtering for the spline is available which avoids association problems and handles semantic information.

3 Evaluation

For the evaluation, real world data is used which was acquired with an experimental vehicle of Ulm University [4] and labeled manually in order to get ground truth data. Thereby, four recorded sequences are used comprising rural roads and urban streets as well as different light conditions and complex scenarios in construction zones.

In general, the labeling in the image plane is done using MATLAB with the Annotation Tool presented in [3] and the labeling toolbox shown in [7]. As all sequences together contain more than 3000 frames, a subset of 80 frames are labeled with a high variety in the scenes. Consequently, input data and according ground truth is available to evaluate the detected free space.

Evaluating the accuracy of the spline position, boundary points of the labeled ground



(a) Top View (b) Image Plane

Figure 4: Boundary Spline Examples, where the label *background* is encoded in blue, *obstacle* in red and *street* is shown in green. On the left, the resulting free space boundary spline in vehicle coordinates, where the DEM cell centers with according labels are shown. On the right, the resulting free space boundary in the image plane, where the transition labels of the spline are shown.

truth are used and a Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{m_{gt}} \sum_{j=1}^{m_{gt}} ||\boldsymbol{b}_j - \hat{\boldsymbol{b}}(\tau_j)||} \tag{9}$$

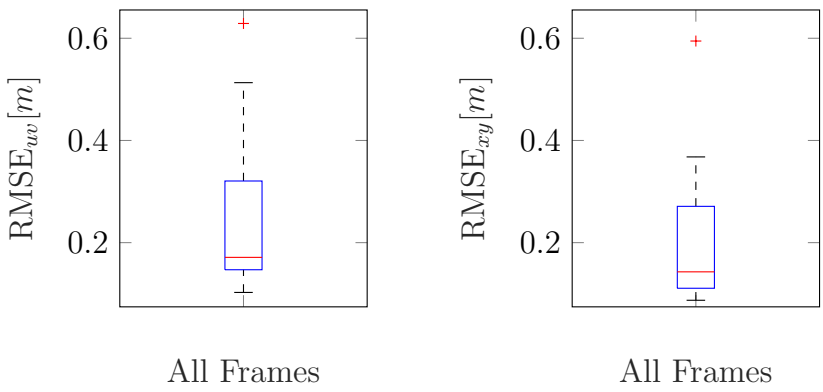
is calculated. Where $\boldsymbol{b}_j$ describes the ground truth boundary point that has the minimal Euclidean distance to the spline boundary point $\hat{\boldsymbol{b}}(\tau_j)$ and m_{gt} is the number of evaluated boundary points.

The label quality is evaluated separately by the percentage of false labels, where the labels $\boldsymbol{p}_j^t$ and $\hat{\boldsymbol{p}}^t(\tau_j)$ are compared. By that, the false label ratio is defined as

$$\text{false label ratio} = \frac{\sum_{l \in \mathcal{L}} FP}{m_{gt}}, \tag{10}$$

where FP represents the false positives for the label l.

In order to analyze the accuracy of the spline model itself, the RMSE is measured for the presented approach and the spline based free space model introduced in [9]. The



Symbol	Value
d	3
n	30
m	54
λ	0.3
k	2
range	$40m$

Figure 5: Comparison between the RMSE yielded by the proposed spline model in the xy-plane and the state of the art model with equidistant knots or respectively in the uv-plane.

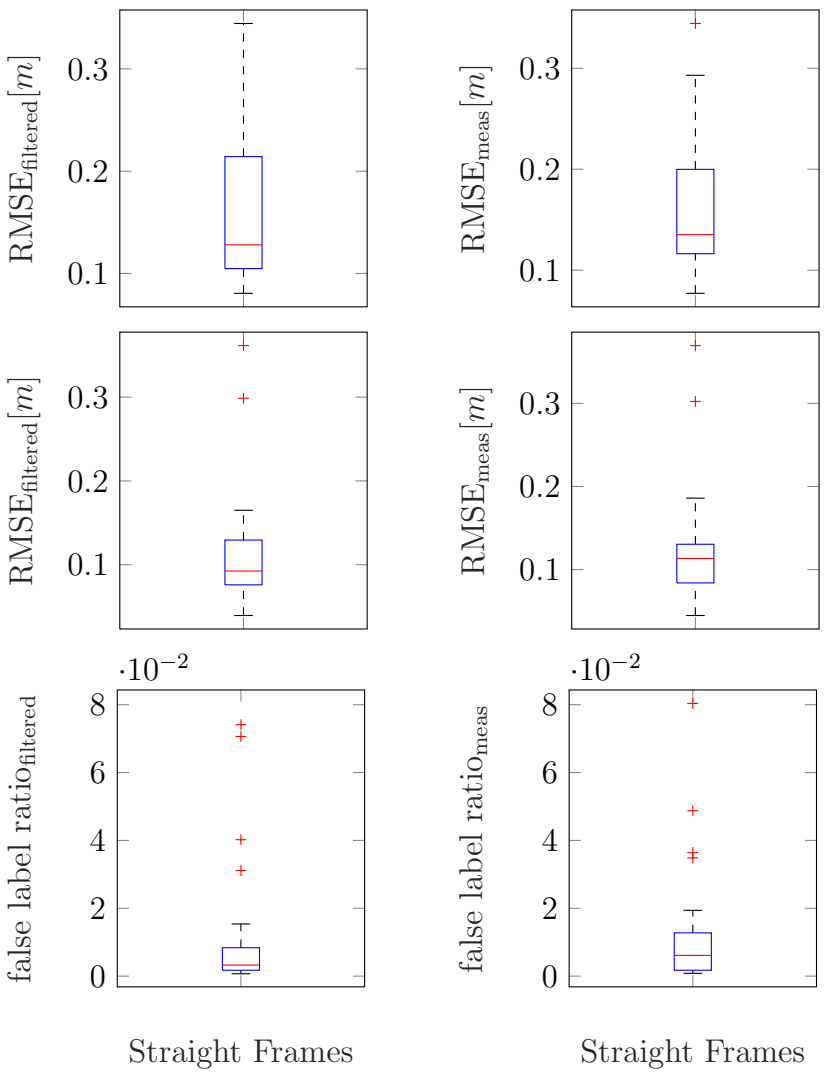
corresponding results are depicted in Figure 5 using all 80 annotated frames. Accordingly, the median is reduced from $0.17m$ to $0.14m$. Further, the standard deviation of the spline resulting from the presented model is with $0.088m$ lower than $0.095m$ in the sate of the art model. These findings are also qualitatively shown in Figure 3. Thereby, the xy-plane spline is much smoother as there is no overshooting behavior. When using camera based sensors this is often the case in transition areas between near and far objects.

Further, the temporal filtering of the spline is evaluated in Figure 6. Thereby, the advantages of the filtering can be illustrated best for straight road parts as the utilized ego motion module yields worse yaw angle estimations in curvy scenarios. This leads to errors of the spline prediction but due to sensor input and not the concept itself. The median of the label error evaluated on a range of 40 meters can be reduced from 0.61% to 0.36% by the filtering, as well as the upper quartile and whisker.

As the filtering leads to stronger smoothing at object transitions, *background* boundary parts are used to calculate the RMSE in order to keep comparability by using the same parameter set. The median of the RMSE is reduced from $0.135m$ to $0.127m$ however the upper quartile increases form $0.199m$ to $0.214m$ in the filtered result. This is due to some outlier frames in which the yaw angle estimation is still inaccurate. By regarding the spline boundary up to 25 meters and therefore reduce the influence of errors in the estimated yaw angle, it can be seen that the median with $0.099m$ and $0.113m$ as well as the upper quartile of the RMSE with $0.129m$ and $0.131m$ is lower in the filtered result. The remaining outliers occur due to systematic errors of the input data in multiple subsequent frames and therefore can not be compensated with filtering algorithms. By regarding the qualitative results[1], it can be observed that the boundary is smoothly estimated up to the DEM margins. In addition, the filtered boundary appears much more stable.

Consequently, the developed spline model successfully represents the free space boundary even better than state of the art concepts. Given a accurate yaw angle estimation, the boundary can be additionally tracked and improved by the presented temporal filtering approach.

[1] https://youtu.be/P18miHm0chE



Symbol	Value
d	3
n	30
m	54
λ	0.3
k	2
range	$40m$

Symbol	Value
d	3
n	30
m	54
λ	0.3
k	2
range	$25m$

Symbol	Value
d	3
n	30
m	54
λ	0.3
k	2
range	$40m$

Figure 6: Evaluation of the Temporal Filtering, where *background* boundaries on straight sequences are regarded. First row, evaluation of the RMSE with a range of 40 meters. Second row, evaluation of the RMSE with a range of 25 meters. Third row, evaluation of the label error ratio on a range of 40 meters.

4 Conclusion and Outlook

The main contribution of this paper is a new continuous semantic free space model with an according temporal filtering. In addition, a generic concept for the integration and fusion of semantic camera based data is proposed. As the evaluation shows, the proposed concept is capable of representing even complex scenarios and the spline model even outperforms state of the art concepts.

To further improve the results, future work includes the incorporation of lidar data, as well as an improved estimation of the yaw angle by utilizing the sequence of stereo images. In addition, it is reasonable to replace the currently used pixel classification with a state of the art concept. Furthermore, a real time implementation is aspired in order to use the free space representation for motion planning.

References

[1] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[2] Paul H. C. Eilers and Brian D. Marx. "Flexible Smoothing with B-splines and Penalties". In: *Statistical Science* 11.2 (1996), pp. 89–102.

[3] F. Korč and D. Schneider. *Annotation Tool*. Tech. rep. TR-IGG-P-2007-01. June 2007.

[4] F. Kunz et al. "Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. June 2015, pp. 666–673.

[5] F. Oniga and S. Nedevschi. "Curb detection for driving assistance systems: A cubic spline-based approach". In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. June 2011, pp. 945–950.

[6] Sebastian Ramos et al. "Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling". In: *CoRR* abs/1612.06573 (2016).

[7] Bryan C. Russell et al. "LabelMe: A Database and Web-Based Tool for Image Annotation". In: *International Journal of Computer Vision* 77.1 (May 2008), pp. 157–173.

[8] L. Schneider et al. "Semantic Stixels: Depth is not enough". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. June 2016, pp. 110–117.

[9] M. Schreier, V. Willert, and J. Adamy. "From grid maps to Parametric Free Space maps; A highly compact, generic environment representation for ADAS". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. June 2013, pp. 938–944.

[10] Matthias Schreier. *Bayesian environment representation, prediction, and criticality assessment for driver assistance systems*. Technische Universität Darmstadt, 2016.

[11] Jan Siegemund. "Street Surfaces and Boundaries from Depth Image Sequences Using Probabilistic Models". PhD thesis. Universitäts-und Landesbibliothek Bonn, 2013.

[12] J. Siegemund et al. "Curb reconstruction using Conditional Random Fields". In: *2010 IEEE Intelligent Vehicles Symposium*. June 2010, pp. 203–210.

[13] M. Thom and F. Gritschneder. "Rapid Exact Signal Scanning With Deep Convolutional Neural Networks". In: *IEEE Transactions on Signal Processing* 65.5 (Mar. 2017), pp. 1235–1250.

[14] Q. Yang et al. "Road detection by RANSAC on randomly sampled patches with slanted plane prior". In: *2016 IEEE 13th International Conference on Signal Processing (ICSP)*. Nov. 2016, pp. 929–933.

[15] J. Ziegler et al. "Making Bertha Drive; An Autonomous Journey on a Historic Route". In: *IEEE Intelligent Transportation Systems Magazine* 6.2 (Summer 2014), pp. 8–20.

Extended Object Tracking With Transparency Mitigation

Tobias Hirscher, Hanne Hettmann, Marco Heimberger, und Jens Honer *

Zusammenfassung: We propose an efficient approach to enhance the measurement likelihood of differentiable star-convex extended target models to mitigate unlikely association processes due to self-occlusion. Our approach allows us to retain an efficient computation of likelihoods and hence the correction step for extended targets while simultaneously penalizing unlikely measurement to target associations. In combination with a filter that is able to process multi-modal distributions we find that this leads to a significant increase in tracking performance, in particular during the initialization of the filter. We demonstrate an implementation based on the recently proposed Gaussian process extended target model and show how our redefined likelihood improves real-world tracking performance using automotive LIDAR data.

Schlüsselwörter: extended target, Gaussian process, transparency

1 Introduction

Robust automotive environment perception is based to a large extent on the tracking of dynamic targets. For this purpose, the recorded data from one or several sensors is processed to continuously estimate the pose and behavior of an unknown number of objects in the vehicle surrounding. Due to the high resolution of modern sensors like laser scanners a single target potentially generates more than one measurement per scan. To track such an extended target the simple point object model commonly used in literature has to be replaced by a more advanced model of the target object [1].

The widely used approach of adding a preprocessing step to reduce the complex measurements patterns to trackable features of the extended targets [2] results in using heuristics and a loss of information. The spatial distribution model avoids this problem by removing the preprocessing step and working on the full set of measurements [3]. [4] introduced the random matrices model that assumes the object extent is the primary source of measurement uncertainty. For Gaussian distributed measurements they provide a closed form expression for the target state distribution in the form of an inverse Wishart distribution coupled with a Gaussian distributed kinematic state. The restriction of negligible measurement uncertainty was later generalized by [5], with further improvements of this model done by [6] and [7]. [8] suggests rectangular and elliptical shapes by means of superposition of Gaussian likelihoods, whereas [9] describes a particle filter for tracking rectangular shapes. An approach to track general star-convex shapes was proposed by [10, 11] in the

*Valeo Schalter und Sensoren GmbH, 74321 Bietigheim-Bissingen, Germany (e-mail: {tobias.hirscher, hanne.hettmann, marco.heimberger, jens.honer}@valeo.com).

form of the random hypersurface model. [12] provide a filter to track arbitrary forms by employing local grid maps.

The technique to illustrate the boundary of extended objects described in the present paper was initially applied in the context of a tracking application in [13]. Due to its analytical properties, the Gaussian processes (GP) have often been utilized in the fields of machine learning, signal processing and statistics. They can be regarded as distribution of functions and in turn allow the representation of an object contour as star-convex shape described by a radial function. Coupled with a multivariate Gaussian to represent the distribution of the kinematic state [13] could track arbitrary star-convex targets while simultaneously estimate their shape.

The original definition of GPs is not applicable for recursive application like object tracking since the different measurements are recorded sequentially in each measurement cycle. For this reason, [14] used the recursive formalism of the GPs introduced in [13]. That this measurement model can be used within a modern multi-object tracking framework to track multiple objects with different kind of shapes was shown in [15].

A major weakness of the GP measurement model implemented in [13] and [15] is the fact that the correction step is not unique but various combinations of the objects center of rotation and its boundary contour exist in parallel. Consequently, there is a chance of associating measurements to the wrong, i.e. the occluded side of the shape, which implicitly leads to the assumption of a transparent target. In the present paper we suggest a solution for the described transparency problem and demonstrate the results with real-word data recorded with a state of the art automotive laser scanner.

The paper is structured as follows. First we are providing the main idea of Gaussian processes and how they can be fitted into tracking applications in section 2.2. Than we present how the single object is modeled and how the corresponding Gaussian process measurement model is executed in section 3. section 4 is pointing out the transparency mitigation and its use together with the Gaussian Process measurement model. Numerical results on experimental data is shown in section 5. Finally the paper is concluded in section 6.

2 Target Motion and Measurement Model

The following section imparts the basic knowledge about Gaussian processes and explains how they can be used to model extended objects in the context of tracking applications. In this paper scalars are stated with lower case letters (e.g. x), vectors with bold printed lower case letters (e.g. $\boldsymbol{x}$) and both sets and matrices with upper case letters (e.g. X).

2.1 Recursive Gaussian Process

A GP [16] is a stochastic process which can be regarded as multivariate Gaussian distribution over functions. It can be described with

$$f(\boldsymbol{u}) \sim \mathcal{GP}(x(\boldsymbol{u}), k(\boldsymbol{u}, \boldsymbol{u}')) \tag{1}$$

and uniquely defined by its mean function and its covariance function

$$x(\boldsymbol{u}) = E[f(\boldsymbol{u})] \tag{2}$$
$$k(\boldsymbol{u}, \boldsymbol{u}') = E[(f(\boldsymbol{u}) - x(\boldsymbol{u}))(f(\boldsymbol{u}') - x(\boldsymbol{u}'))]. \tag{3}$$

According to its interpretation as multivariate Gaussian distribution, the function values $f(u)$ of a GP are evaluated at the positions of the set of inputs u and follow the Gaussian distribution curve.

$$\begin{bmatrix} f(u_1) \\ \vdots \\ f(u_n) \end{bmatrix} \sim \mathcal{N}(\boldsymbol{x}, K), \text{ with } \boldsymbol{x} = \begin{bmatrix} x(u_1) \\ \vdots \\ x(u_n) \end{bmatrix} \quad K = \begin{bmatrix} k(u_1, u_1) & \dots & k(u_1, u_n) \\ \vdots & \ddots & \vdots \\ k(u_n, u_1) & \dots & k(u_n, u_n) \end{bmatrix}. \tag{4a}$$

Classical GPs are designed as batch processes, i.e. all the input data is available from the start [16]. [13] applied the recursive formulation of GPs derived in [14] to come up with the sensor data sequentially arriving in each update step. By the definition of the GP recursion as state space model it is possible to use a Kalman filter to recursively correct the GP estimation based on newly available measurements. Within this process the measurement model for the shape $f(u)$ of an extended target is defined as

$$y_t = f(u_t) + w, \quad w \sim \mathcal{N}(0, R), \tag{5}$$

w being the normal distributed zero-mean measurement noise with its covariance R. The goal of the recursion is to estimate the approximation $\boldsymbol{f}_t$ of the function $f(u)$ at a finite set of specific basis values $\boldsymbol{u}^f$. The so-called basic vectors $\boldsymbol{f}_t$ located at basic points $\boldsymbol{u}^f \widehat{=} \{u_i^f \mid i = 1, \dots, n\}$ are recursively estimated at the timesteps $t = 0, 1, \dots$. Set of measurements $\boldsymbol{z}_t \widehat{=} \{z_{t,i} \mid i = 1, \dots, m\}$ at corresponding inputs $\boldsymbol{u}_t \widehat{=} \{u_{t,i} \mid i = 1, \dots, m\}$ are observed for that. The positions and the number n of basic vectors are the same for all timesteps t. For better readability we will omit the time index t in most of the case and use it only if is needed for a better understanding. The basis vectors are a spares representation of the GP. Assuming that the measurements $\boldsymbol{z}$ and the function values $\boldsymbol{f}$ are jointly Gaussian, [16] showed that the conditional posterior density is given by

$$p(\boldsymbol{f} \mid \boldsymbol{z}_{1:t}) = \mathcal{N}(\boldsymbol{f}; \boldsymbol{x}^f, P^f) \tag{6}$$

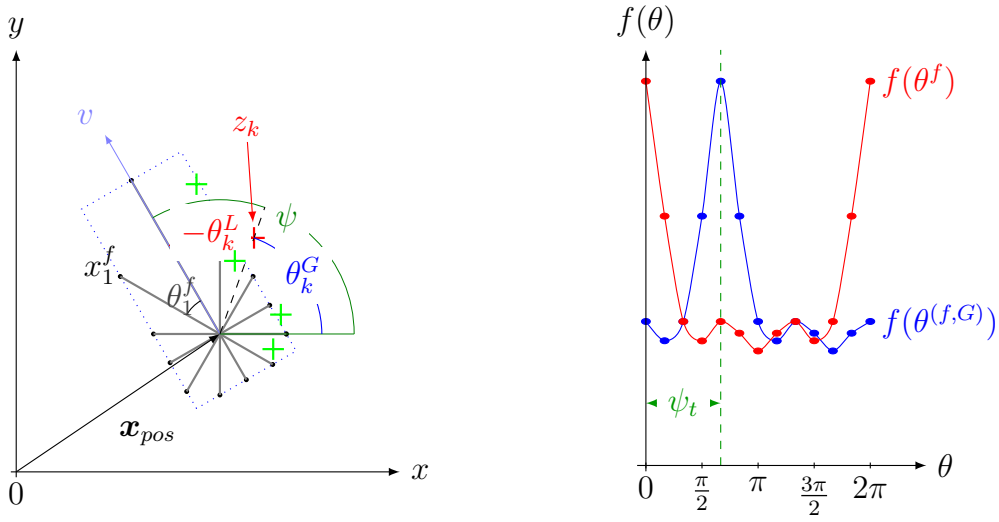
using the mean function values $\boldsymbol{x}^f = [x_1^f(u_1^f), \dots, x_n^f(u_n^f)]$ at the basis points $\boldsymbol{u}^f$ and the process noise covariance P^f. Further assuming a measurement process as defined in (5) with $w \sim \mathcal{N}(\boldsymbol{0}, R)$, the conditional measurement likelihood function [16] is given by

$$p(\boldsymbol{z} \mid \boldsymbol{f}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{z}_+, R^f). \tag{7}$$

Here the predicted measurements are $\boldsymbol{z}_+ = H^f \boldsymbol{x}^f$, the observation model and the measurement noise are

$$H^f = K_{mf} K_{ff}^{-1} \tag{8a}$$
$$R^f = K_{mm} + R - K_{mf} K_{ff}^{-1} K_{fm}. \tag{8b}$$

Here we use the short hand notation $K_{mf} = K(\boldsymbol{u}, \boldsymbol{u}^f)$, $K_{mm} = \text{diag}[K(\boldsymbol{u}, \boldsymbol{u})]$, $K_{ff} = K(\boldsymbol{u}^f, \boldsymbol{u}^f)$ and $K_{fm} = K(\boldsymbol{u}^f, \boldsymbol{u})$ for the kernels of the GP. The whole process can be described in terms of Kalman prediction and correction equations as shown in [15].



(a) (b)

Abbildung 1: Gaussian process measurement model with its describing parameters. The approximation of a vehicle with basic vectors $\boldsymbol{x}^f$ and the corresponding basic angles $\boldsymbol{\theta}^f$ is visualize in (a). The green makes are examples of observations $\boldsymbol{z}$ of a sensor while the red mark is a special marked observation showing the relation of local measurement angles $\boldsymbol{\theta}^L$ to the velocity vector v and the orientation ψ of the object model. (b) illustrates the Gaussian process modeling the vehicle in (a) as function $f(\theta)$. The red line shows the smoothed local mean function $f(\theta^f)$ of the Gaussian process whereas the blue line is the function based on the global basic angles $\theta^{(f,G)}$.

2.2 Extended Object Model

Following the implementation described in [13], GPs can be applied to model moving extended objects with a radial shape. The basis points are defined at specific angles $\boldsymbol{\theta}^f$ which span the interval $[0, 2\pi)$, whereas the values of the mean function $\boldsymbol{x}^f$ describe the radial distance at the corresponding angles as illustrated in Fig. 1.

Augmenting the kinematic object model with the state space model of the GP, it is possible to simultaneously estimate the motion and the extent of an object. The combined state vector is then given by

$$\boldsymbol{x} = [(\boldsymbol{x}^{kin})^{\mathrm{T}} \quad (\boldsymbol{x}^f)^{\mathrm{T}}]^{\mathrm{T}}, \quad \text{with} \quad \boldsymbol{x}^{kin} = [(\boldsymbol{x}_{pos})^{\mathrm{T}} \quad \psi \quad (\boldsymbol{x}_{ho})^{\mathrm{T}}]^{\mathrm{T}}. \tag{9a}$$

The motion states $\boldsymbol{x}^{kin}$ consist of the position $\boldsymbol{x}_{pos} = (x^c, y^c)$ and the target orientation ψ as well as the higher order components $\boldsymbol{x}_{ho}$ of the center of rotation (COR) and the orientation ψ. In this work we are using only the velocity v of the COR as higher order component, therefore $\boldsymbol{x}_{ho} = v$. We also assume a constant target object shape, i.e. $\boldsymbol{x}^f = \boldsymbol{x}_0^f$. This assumption significantly decreases the computational effort of the correction step since it avoids the online inversion of the GP correlation kernel. Note that discrete changes between different shapes may be introduced by the use of a mixture model in combination with state transitions. The state vector and its covariance P are recursively predicted with an unscented transform (UT) similar to the one shown in [17] and corrected with an extended Kalman filter (EKF) in analogy to [13].

2.3 Covariance Function

The choice of the GP kernel or covariance function has a strong impact on the behaviour of the target shape. [13] used a periodic covariance function given by

$$k_{per}(\theta, \theta') = \sigma_f^2 e^{-\frac{2\sin^2\left(\frac{|\theta-\theta'|}{2}\right)}{l^2}} + \sigma_r^2 \delta_{\theta\theta'}. \tag{10}$$

The parameter l denotes the length-scale which can be interpreted as the distance in the inputs required for a significant change in the outputs. The factor σ_f is the prior variance of the signal amplitude and σ_r is the variance of the Gaussian prior. Using a periodic kernel a GP can model 2π-periodic functions which allows different mean and covariance values at basis angles.

The symmetric covariance function also presented in [13] corresponds to the periodic covariance function with twice the frequency. It enables faster estimations of the object shape in case of centrically symmetric targets. Contrary to the periodic covariance function, the symmetric covariance function implies that the COR lies at the object's center. Since in the context of moving vehicles the COR is more likely to be at the center of the rear axis, [15] suggests an axis-symmetric covariance function.

Note that in the simplified case with constant target shape, i.e. $\boldsymbol{x}^f = \boldsymbol{x}_0^f$, additional symmetries within the GP kernel are not required.

3 Gaussian Process Tracking

In this section we will roughly describe the prediction step of the tracking filter and in more detail the combined measurement update of the kinematic state and the GP.

3.1 Target Prediction

The prediction of the GP state and covariance is carried out according to the equations of the GP in [15]. The prediction of the kinematic state is performed according to [17] via an UT implementation.

3.2 Measurement Correction

The a priori distribution of the state $\boldsymbol{x}$ is recursively updated with a set of measurements $Z \mathrel{\widehat{=}} \{\boldsymbol{z}_k \mid k = 1, \ldots, m\}$ at each timestep t.

In turn every measurement $\boldsymbol{z}$ has to be associated with a local angle θ relative to the COR $\boldsymbol{x}_{pos}$ and the orientation ψ of the object. These local angles are determined by

$$\boldsymbol{\theta}^L = \boldsymbol{\theta}^L(\boldsymbol{x}_{pos}, \psi) = \boldsymbol{\theta}^G(\boldsymbol{x}_{pos}) - \psi \tag{11}$$

with $\boldsymbol{\theta}^G(\boldsymbol{x}_{pos})$ the global angles of the measurements relative to the x-axis. This measurement process is depicted in Fig. 1. The local angles are used to describe the relation between the object states and the measurements

$$Y = \boldsymbol{x}_{pos} + p(\boldsymbol{x}_{pos}) f(\boldsymbol{\theta}^L) + w \tag{12}$$

with w the measurement noise as defined in (5). The radial function $f(\cdot)$ describes the extent of the objects relative to $\boldsymbol{x}_{pos}$ depending on the local angles $\boldsymbol{\theta}^L$. Last, the vector

$$\boldsymbol{p}_k(\boldsymbol{x}_{pos}) = \boldsymbol{p}(\theta_k^G(\boldsymbol{x}_{pos})) = \begin{bmatrix} \cos(\theta_k^G(\boldsymbol{x}_{pos})) \\ \sin(\theta_k^G(\boldsymbol{x}_{pos})) \end{bmatrix} = \frac{\boldsymbol{z}_k - \boldsymbol{x}_{pos}}{||\boldsymbol{z}_k - \boldsymbol{x}_{pos}||} \tag{13}$$

is the unit vector from the target center in the direction of the respective measurement. By use of the above definitions (12) may be written as

$$Y = \boldsymbol{x}_{pos} + \boldsymbol{p}(\boldsymbol{x}_{pos})[H^f(\boldsymbol{\theta}^L)\boldsymbol{x}^f + \epsilon^f] + w, \quad \epsilon^f \sim \mathcal{N}(0, R^f(\boldsymbol{\theta}^L)) \tag{14}$$

$$Y = h(\boldsymbol{x}) + e, \quad e \sim \mathcal{N}(0, R_{gp}) \tag{15}$$

with the nonlinear measurement function

$$h(\boldsymbol{x}) = \boldsymbol{x}_{pos} + \boldsymbol{p}(\boldsymbol{x}_{pos})H^f(\boldsymbol{\theta}^L)\boldsymbol{x}^f, \tag{16}$$

and the combined measurement noise

$$R_{gp} = \boldsymbol{p}(\boldsymbol{x}_{pos})R^f(\boldsymbol{\theta}^L)\boldsymbol{p}(\boldsymbol{x}_{pos})^{\mathrm{T}} + R. \tag{17}$$

Since R was already accounted in (17) it can be neglected in (8b). The combined likelihood of the kinematic state and the shape is in turn given by

$$p(Y \mid \boldsymbol{x}_+) = \mathcal{N}(Y; Z_+, R_{gp}) \quad \text{where} \quad Z_+ = \mathrm{h}(\boldsymbol{x}_+). \tag{18}$$

The m measurements Z as well as the combined measurement noise R_{gp} and the nonlinear measurement function $h(\boldsymbol{x})$ are reformulated for the purpose of the correction step as

$$\boldsymbol{z} = [\boldsymbol{z}_1^{\mathrm{T}}, \ldots, \boldsymbol{z}_m^{\mathrm{T}}]^{\mathrm{T}}, \quad R_{gp} = \mathrm{diag}[R_1, \ldots, R_m], \quad h(\boldsymbol{x}) = [h_1(\boldsymbol{x})^{\mathrm{T}}, \ldots, h_m(\boldsymbol{x})^{\mathrm{T}}]^{\mathrm{T}}. \tag{19}$$

Using this augmented formulation, the a priori distribution of the state $\boldsymbol{x}$ can be recursively updated by applying the EKF equations similar to [13] Following the well-known EKF formulation the nonlinearity of the measurement function $h(\boldsymbol{x})$ linearized and yields the Jacobian H. The assumption of a constant a priori known shape allows the analytical calculation (cp. [13]) of the gradient according to the following equations:

$$H = \frac{dh(\boldsymbol{x})}{d\boldsymbol{x}} = \frac{d}{d\boldsymbol{x}}[h_1(\boldsymbol{x})^{\mathrm{T}}, \ldots, h_n(\boldsymbol{x})^{\mathrm{T}}]^{\mathrm{T}} \tag{20}$$

$$\frac{d}{d\boldsymbol{x}}h_k(\boldsymbol{x}) = \left[\frac{dh_k(\boldsymbol{x})}{d\boldsymbol{x}_{pos}}, \frac{dh_k(\boldsymbol{x})}{d\psi}, \frac{dh_k(\boldsymbol{x})}{d\boldsymbol{x}_{ho}}, \frac{dh_k(\boldsymbol{x})}{d\boldsymbol{x}^f}\right]^{\mathrm{T}} = \left[H_{pos},\ \boldsymbol{h}_\psi,\ H_{ho},\ H_{\boldsymbol{x}^f}\right]^{\mathrm{T}} \tag{21}$$

with

$$\frac{dh_k(\boldsymbol{x})}{d\boldsymbol{x}_{pos}} = \boldsymbol{I} + \left.\frac{\partial p_k(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{x}_{pos}} H^f(\theta_k^L)\boldsymbol{x}^f + p_k(\boldsymbol{x}_{pos})\left.\frac{\partial H^f(u)}{\partial u}\right|_{u=\theta_k^L}\left.\frac{\partial \theta_k^G(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{x}_{pos}}\boldsymbol{x}^f$$

$$\frac{dh_k(\boldsymbol{x})}{d\psi} = -p_k(\boldsymbol{x}_{pos})\left.\frac{\partial H^f(u)}{\partial u}\right|_{u=\theta_k^L}\boldsymbol{x}^f \qquad \frac{dh_k(\boldsymbol{x})}{d\boldsymbol{x}_{ho}} = \boldsymbol{0} \qquad \frac{dh_k(\boldsymbol{x})}{d\boldsymbol{x}^f} = \boldsymbol{0} \tag{22}$$

Note that the dimensions of above matrices are

$$H_{pos} \in \mathbb{R}^{2\,\mathrm{x}\,2}, \quad \boldsymbol{h}_\psi \in \mathbb{R}^{2\,\mathrm{x}\,1}, \quad H_{ho} \in \boldsymbol{0} \in \mathbb{R}^{2\,\mathrm{x}\,(|\boldsymbol{x}^{kin}|-3)}, \quad H_{\boldsymbol{x}^f} \in \mathbb{R}^{2\,\mathrm{x}\,|\boldsymbol{x}^f|}. \tag{23}$$

4 Transparency Mitigation

This section describes a modified likelihood function that penalizes associations of measurement to an unobservable part of the target model and represents the main contribution. The lack of such a penalty in the original Gaussian process target model [13] may lead to the association of measurements to the target surface opposite to the sensor-facing side and hence yield to wrong association between measurement and obstacle surface which leads in this regard to a false estimate of the COR. In the following we will use the term *transparent target* for this effect. The root cause of this behavior is the efficient association scheme within the Gaussian process target model in which the relative position of the measurement with respect to the center of the star-convex shape determines where on the surface the measurement point is assumed to originate, see eq. (16).

Since we do not want to compromise the efficiency of this approach we modify the likelihood function itself rather than the association logic according to

$$g(Z|\boldsymbol{x}) = g_f(Z|\boldsymbol{x})g_p(\boldsymbol{\gamma}(Z,\boldsymbol{x})) \tag{24}$$

where $g_f(Z|\boldsymbol{x})$ is the association likelihood of the measurements Z to the track $\boldsymbol{x}$ [18], with the weighting function $g_p(\boldsymbol{\gamma}(Z,\boldsymbol{x}))$ depending on

$$\tilde{\gamma}(z,\boldsymbol{x}) = \frac{\left\langle \frac{dz}{d\theta^G}, \hat{z}(\boldsymbol{x_{pos}}) \right\rangle}{\|\frac{dz}{d\theta^G}\|\|\hat{z}(\boldsymbol{x_{pos}})\|} \tag{25}$$

the angle between the tangential vectors of the curve given by (12) and its orthogonal projection on the tangential plane spanned by the field of view of the sensor. Note that this approach may be considered as a failure recognition of the association scheme. The modification will not create a better association between measurements and the target surface like in [19] but only determine its failure and in turn penalize such an association. In turn the approach should be combined with a sophisticated multi-hypothesis tracking framework like a Bernoulli Filter.

In order to decrease the complexity of our calculation we approximate

$$\gamma(z,\boldsymbol{x}) \approx \frac{\left\langle \frac{dz}{d\theta^G}, \mathbf{n}(\boldsymbol{x}_{pos}) \right\rangle}{\|\frac{dz}{d\theta^G}\|\|\mathbf{n}(\boldsymbol{x}_{pos})\|} \tag{26}$$

in which we replaced $\hat{z}(\boldsymbol{x}_{pos})$ with $\mathbf{n}(\boldsymbol{x}_{pos})$ which is constant for a given target. This approximation is valid as long as field of view occupied by the target is small. This is usually the case for vehicles in front (or the rear) of the host vehicle and thus well justified for applications such as target tracking for adaptive cruise control. Note that in both cases the function γ only depends on the angle between the center of the target and the measurement point. The function $g_p(\boldsymbol{\gamma}(Z,\boldsymbol{x}))$ is then chosen according to the specific sensor in use and the predominant use-cases. A simple but already sufficient realization of the function $\boldsymbol{\gamma}(Z,\boldsymbol{x})$ for automotive LIDAR sensors and highway/rural road scenarios used to generate our results is illustrated in Fig. 2.

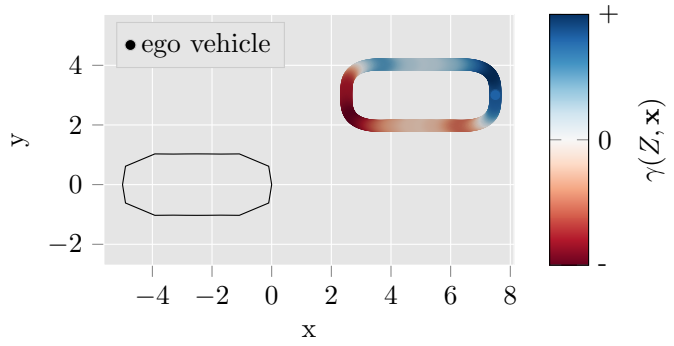


Abbildung 2: Schematic illustration of the visible part and the resulting $\boldsymbol{\gamma}(Z, \boldsymbol{x})$. The black polygon is representing the ego vehicle from which the equally separated measurements shown as colored dots in the upper right are generated on the target surface. The intensity of $\boldsymbol{\gamma}(Z, \boldsymbol{x})$ for each measurement is visualized with the colorbar on the right. The blue marks demonstrating observations which will get a penalty weight.

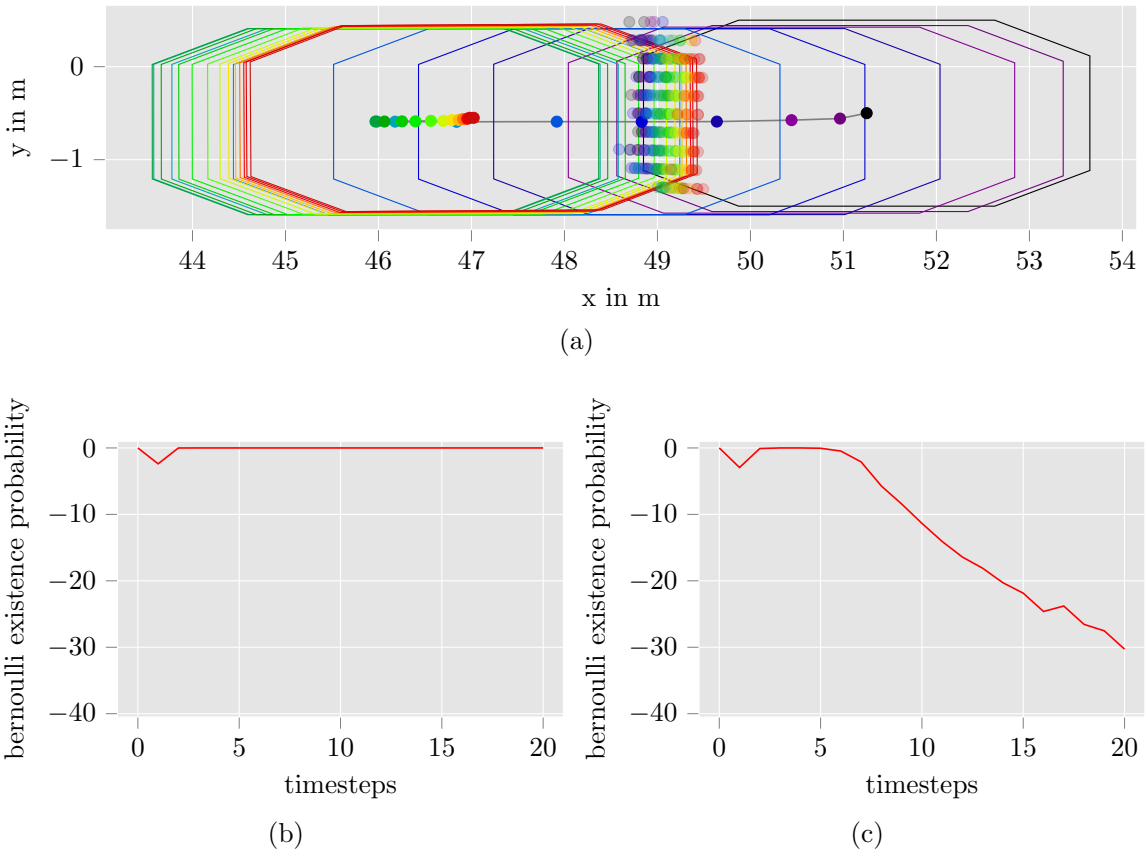


Abbildung 3: Second scenario to compare a Gaussian mixture target track with three components with and without applying the transparency weighting function $g_p(\boldsymbol{\gamma}(Z, \boldsymbol{x}))$. The figure on the top visualizes the trajectory output of the bernoulli filter. On the bottom (b) is showing the logarithmic existence probability of the target track for every timestep with executing the weighting function and (c) without it.

5 Results

The effect of the transparency mitigation function $g_p(\boldsymbol{\gamma}(Z, \boldsymbol{x}))$ we will evaluate in the fol-

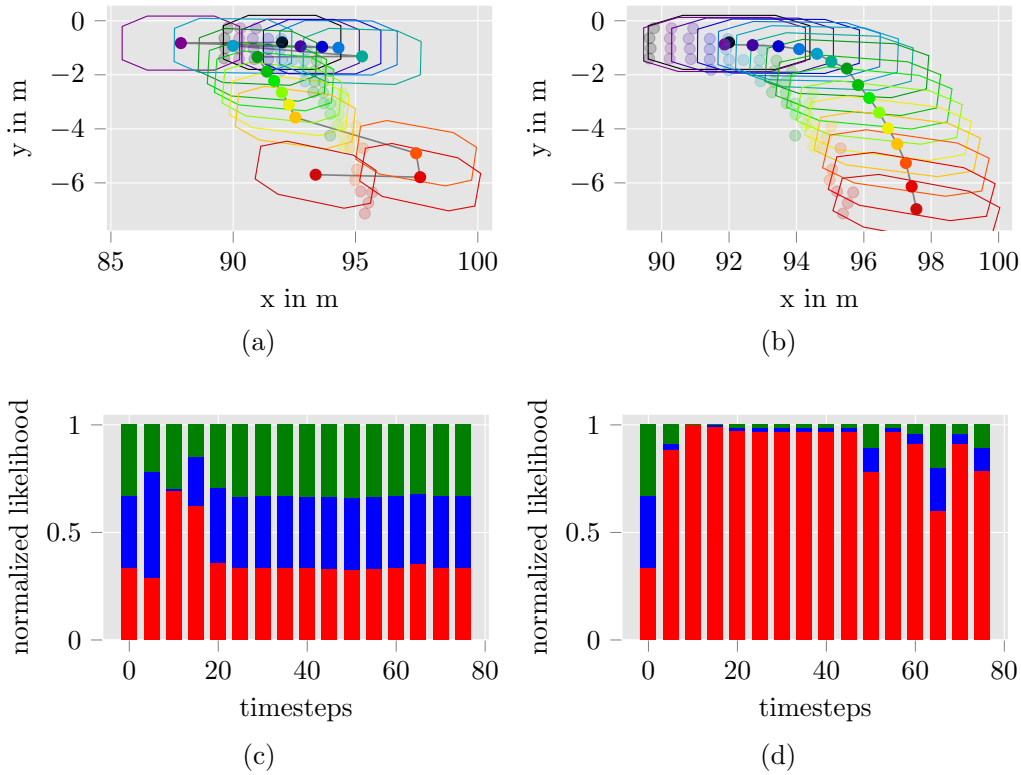


Abbildung 4: Comparison of a scenario with and without using the transparency weighting function $g_p(\boldsymbol{\gamma}(Z, \boldsymbol{x}))$. In the top left the trajectory of a Gaussian mixture target track containing three components without applying the weighting function and in (b) with use of the function. In both figures (a) and (b) the measurements are shown with colored markers and the estimated target extent with colored polygons. The different painting illustrates the diverse timesteps. Black correspond to the initial step. Figure (c) as well as (d) are describing the normalized likelihoods of the three Gaussian mixture components at each timestep of the scenarios shown in (a) and (b).

target vehicle is driving in front of the host vehicle with a similar high velocity whereas in the second scenario the target vehicle is performing a right turn at approximately the same velocity as the host vehicle. The scenarios were recorded with a Valeo Scala Generation 1 LIDAR mounted in the front bumper of the host vehicle. The measurement frequency is 25 Hz and features a range of detection of 100 m for a target with 8% reflectivity and an azimuthal resolution of 0.25° in a field of view of ±72.5°. All presented figures are visualized in the (right-handed) vehicle coordinate system with the origin located at the center of the rear axis of the host vehicle and the x-axis situated along the vehicle symmetry axis. For the target tracking framework we use a simple Bernoulli filter implementation [20]. The kinematic state of all components is modeled with (9a) and the target extent is modeled with a Gaussian Process with constant basic vectors $\boldsymbol{x}^f = \boldsymbol{x}_0^f$ and cardinality $|\boldsymbol{x}_0^f| = 20$. (see section 2.2.) The $\boldsymbol{x}_0^f$ and corresponding angles are chosen to approximate a rectangular vehicle shape with a width and length of 2.0 m and 4.8 m, respectively. The covariance if provided by the correlation function (10) with parameters $\sigma_f = 0.30$ m, $\sigma_r = 0.06/4$ m and the lengthscale $l = \pi/10$. The noises parameters of the kinematic process are set to $\sigma_v = 1.55$ m/s^2 for the velocity and $\sigma_\phi = 0.22$ rad / s^2 for the yaw

rate. Last the measurement noise variance is chosen to be $R = 0.10\boldsymbol{I}$ m^2. To highlight the competition between different components in the Bernoulli filter we do not merge similar components. Since we always consider only a single measurement cluster this modification reduces the Bernoulli filter to a parallel implementation of different trackers with joint estimation of existence probability. We are using a simple weighting function

$$g_p(\boldsymbol{\gamma}(Z, \boldsymbol{x})) = \prod_{i=1}^{|z|} f(\boldsymbol{\gamma}(\boldsymbol{z}_i, \boldsymbol{x})) \quad \text{where} \quad f(\boldsymbol{\gamma}(\boldsymbol{z}_i, \boldsymbol{x}) = e^{-\widetilde{f}(\boldsymbol{\gamma}(\boldsymbol{z}_i, \boldsymbol{x}))} \tag{27}$$

with

$$\widetilde{f}(\boldsymbol{\gamma}(\boldsymbol{z}_i, \boldsymbol{x}) = \begin{cases} \tau, & \text{if} \quad \boldsymbol{\gamma}(\boldsymbol{z}_i, \boldsymbol{x}) > 0 \\ 0, & \text{else} \end{cases}$$

to increase the effect of the transparency mitigation. In the following we will consider τ as penalty term for measurements on the transparent side of the measured object. Note that although the direction of travel of a target vehicle may be inferred from its shape with a single LIDAR measurement, the velocity and in particular its sign cannot. In turn there are two approaches to encode this ambiguity in the initialization: Either one initializes a single component with a single expected velocity and a high σ_v or one initializes the target track with several mixture components each using a different velocity. In the following we will show that in both case the transparency mitigation will improve the tracking performance.

5.1 Straight Driving Case

First we evaluate a scenario in which the target vehicle is driving in front of the host vehicle and the initial track velocity is set to $v = 0$ m/s. To compensate for the uncertainty of the initial velocity we choose a high variance $\sigma_v = 0.225$ m/s. The initial position $\boldsymbol{x}_{pos} = (51.25, -0.50)$ is chosen such that the Gaussian process contour aligns well with the measurements. Last the process noise is set to $\sigma_a = 1.55$ m/s^2. The results are shown in Fig. 3: Trajectories in Fig. 3a are constructed by extracting the component with the respective highest likelihood in every timestep. It can be clearly seen that the filter has difficulties adjusting to the correct velocity despite the relatively high process noise and needs several cycles to convergence to the correct velocity. Yet it is associating the measurement to the respective opposite side of the target model. The side that faces the sensor is not supported by measurements. If we now look at the (logarithmic) existence probability of the Bernoulli filter using the original implementation (Fig. 3b) of the likelihood does not indicate that the target is located in front of the measurements. In turn a filter will continue to track the target and most likely not recover. On the other hand our modified likelihood function (Fig. 3c) yields a steep drop in the existence probability and prevents such pathological behaviour.

5.2 Right Turn Case

In a right turn scenario we initialize the track as a multi-modal distribution with three likely but distinct velocities, namely $v_{1,2} = \pm 10$ m/s and $v_3 = 0$ m/s. The initial position

and orientation is set to $\boldsymbol{x}_{pos} = (92.0, -0.8)$ and yaw $\psi = 0$, i.e the same for all mixture components, and again chosen such that the Gaussian process contour is aligned with the measurements. Last, the process noise is set to $\sigma_a = 0.89$. Note that we are able to choose a lower variance due to the multi-modal initial state.

Fig. 4 shows the comparative output of the filter with (right) and without (left) taking into account the modified likelihood. The track trajectories in Fig. 4a and Fig. 4b are constructed in the same fashion as before (see Fig. 3a). The figures clearly show that without the modified likelihood the ambiguity between the components prevents a consistent estimation of the track: Since no component is dominant the track jumps between different components and creates the jagged trajectory. Moreover Fig. 4a shows the effect of the measurements being associated to the rear of the target, which, given the physical properties of the LIDAR, is highly unlikely. In contrast the modified likelihood prevents unlikely terms to contribute. (see Fig. 4d.) (Note that those components reappear after several steps. This is an artifact since we do not prune components which means we do not delete components with a low existence probability.

6 CONCLUSION

This paper presented an efficient approach to improve the likelihood function for extended targets to penalize unlikely association processes. We demonstrated the simple application of our approach on a Gaussian processes target model and provided examples of the effectiveness of our approach on real-world Lidar data recorded in typical driving situations.

Literatur

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software.* Wiley, 2004.

[2] C. Mertz, L. E. Navarro-Serment, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppe, C. Urmson, N. Vandapel, M. Hebert, C. Thorpe *et al.*, "Moving object detection with laser scanners," *Journal of Field Robotics*, vol. 30, pp. 17–43, 2013.

[3] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *Radar, Sonar and Navigation, IEEE Proceedings*, vol. 152, no. 5, pp. 364–371, October 2005.

[4] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, July 2008.

[5] M. Feldmann, D. Franken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409–1420, April 2011.

[6] M. Wieneke and W. Koch, "Probabilistic tracking of multiple extended targets using random matrices," in *Signal and Data Processing of Small Targets*, O. E. Drummond, Ed. International Society for Optics and Photonics, 2010.

[7] W. Koch and M. Feldmann, "Cluster tracking under kinematical constraints using random matrices," *Robotics and Autonomous Systems*, vol. 57, no. 3, pp. 296–309, 2009.

[8] K. Granström, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *in Proceedings of the 14th International Conference on Information Fusion (FUSION)*, July 2011.

[9] A. Gning, B. Ristic, L. Mihaylova, and F. Abdallah, "Introduction to box particle filtering," *IEEE Signal Processing Magazine*, 2012.

[10] M. Baum and U. D. Hanebeck, "Random hypersurface models for extended object tracking," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Dec 2009, pp. 178–183.

[11] ——, "Extended object tracking with random hypersurface models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 149–159, January 2014.

[12] M. Schütz, N. Appenrodt, J. Dickmann, and K. Dietmayer, "Simultaneous tracking and shape estimation with laser scanners," in *16th International Conference on Information Fusion (FUSION)*. IEEE, 2013, pp. 885–891.

[13] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, Aug. 2015.

[14] M. F. Huber, "Recursive gaussian process regression," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 3362–3366.

[15] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer, "Multiple extended object tracking using gaussian processes," in *19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 868–875.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[17] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153–158.

[18] R. Altendorfer and S. Wirkert, "A complete derivation of the association log-likelihood distance for multi-object tracking," *CoRR*, 2015.

[19] F. Faion, A. Zea, M. Baum, and U. D. Hanebeck, "Partial likelihood for unbiased extended object tracking," in *18th International Conference on Information Fusion (Fusion)*. IEEE, 2015, pp. 1022–1029.

[20] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, Inc., 2007.

Fusion of Camera and Lidar Data for Object Detection using Neural Networks

Enrico Schröder* Mirko Mählisch* Julien Vitay † Fred Hamker †

Zusammenfassung: We present a novel architecture for intermediate fusion of Lidar and camera data for neural network-based object detection. Key component is a transformer module which learns a transformation of feature maps from one sensor space to another. This allows large parts of the multi-modal object detection network to be trained unimodally, reducing the required amount of costly multi-modal labeled data. We show effectiveness of the transformer as well as the proposed fusion scheme.

Schlüsselwörter: Sensor Fusion, Object Detection, Convolutional Neural Networks, Lidar

1 Introduction

One of the most important tasks in automotive perception is detecting objects and obstacles such as cars or pedestrians in the vehicle's surroundings. For reasons of performance and robustness, multiple sensor modalities such as camera, Lidar and Radar sensors are used to exploit individual strenghts of each sensor type. Traditionally, discrete objects detected by each sensor are fused via some model-based Bayesian fusion framework. Recently, methods based on convolutional neural networks have been used to detect objects in raw images (e.g. [7]) and also Lidar pointclouds ([11], [8]), outperforming methods based on traditional handcrafted features. This makes it viable to explore methods which fuse sensor modalities at a lower level of abstraction directly within these object detection networks to potentially make use of additional information that has not yet been abstracted away by an underlying model. We present a general framework for fusing object detection networks for multiple sensor modalities at an intermediate stage using perspective-invariant features.

2 Related work

In contrast to traditional image classification networks, object detection networks predict position and class of *multiple objects* in the frame. Most object detection networks first feed input images through a convolutional *feature extractor* network such as ResNet [3] or VGG [9]. The resulting feature maps are then passed through a combination of convolutional

*AUDI AG, Development Sensor Fusion & Localization for Automated Driving (e-mail: {enrico.schroeder,mirko.maehlisch}@audi.de)

†TU Chemnitz, Dept. of Computer Sciences (e-mail: {julien.vitay,fred.hamker}@informatik.tu-chemnitz.de)

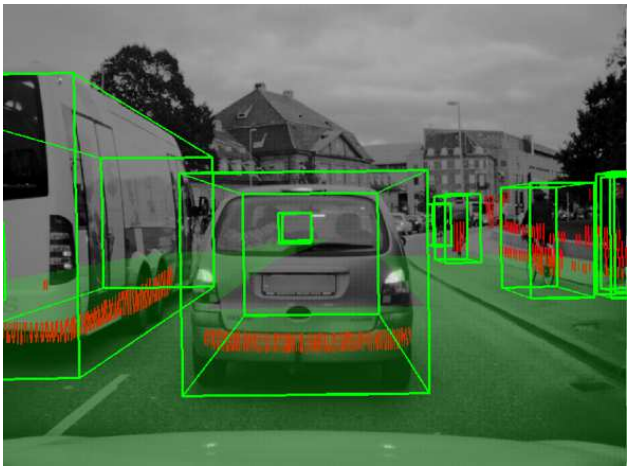


Abbildung 1: A sample frame of our labeled dataset, with Lidar data and 3D groundtruth projected into image space.

and fully-connected layers (the *detector*) to generate object bounding boxes and classes. Work on multi-modal object detection in an automative context has gained traction since introduction of the 2D birdview and 3D object detection challenges based on the KITTI dataset [2]. KITTI provides front camera images as well as Velodyne Lidar point cloud and 3D groundtruth objects. A multitude of approaches based on this dataset have been proposed. Most either use camera detections to create regions of interest in the Lidar pointcloud for final object prediction (such as [6]) or use anchor regions in 3D space and project them into camera and Lidar views to get region proposals for the different modalities. These proposals are then cropped and jointly fed to detector stages to produce final object predictions (MV3D [1] or AVOD [5]).
While many of the available approaches show good performance, they do not focus on the issues which are prevalent in series car development. Most methods train an entire monolithic multi-modal object detection network in one single step, requiring large labeled datasets containing all modalities at once. The process of collecting this data is hugely laborious and costly (and thus unfeasible in series car production, where employed sensor modalities and models change frequently). A modular approach would be desired, where perception for each modality (or sensor respectively) can be trained individually and then only be fine-tuned by a much smaller multi-modal dataset to train specific fusion layers. Additionally, having sensors perform some perception individually would allow for the entire system behavior to be much more traceable since sensors could still be analyzed separately. This is an important aspect aiding functional safety. Our approach fuses information at a single point in the object detection network by making feature map activations from different sensors compatible in locality and properties.

3 Fusion architecture

Our proposed approach focuses on fusing pre-trained object detection networks for single sensor modalities, requiring a significantly smaller amount of multi-modal labeled data compared to end-to-end object detection networks. The architecture respects that most relevant sensors produce depth information (with the exception of monocular cameras)

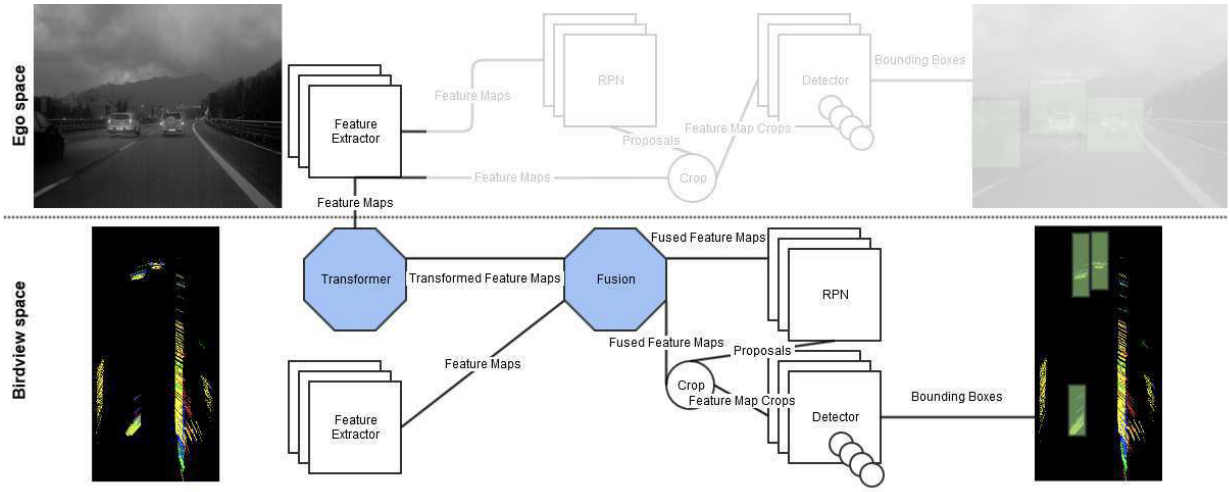


Abbildung 2: Our proposed fusion architecture. White elements denote parts of the network belonging to an arbitrary state-of-the-art object detector (Faster-RCNN [7] in our case). Blue elements denote our transformer and fusion modules mapping feature maps from ego- to birdview. This allows using camera features in addition to Lidar for predicting objects' locations and classes. The camera-only detector is used only for training the feature extractor and can be removed afterwards.

and can be processed efficiently in a 2D birdview representation. Thus, our framework mainly operates in 2D birdview and outputs 2D object bounding boxes. By projecting 3D Lidar pointclouds into 2D birdview we can leverage robust state-of-the-art 2D object detection networks for all relevant sensors.
Our low-level architecture is shown in figure 2. We employ a common two-stage object detection network (Faster-RCNN [7]) which extracts features from the input via a feature extraction network in the first stage, creates region proposals from the extracted features and then classifies and refines these region proposals in the second stage. Our proposed fusion architecture does not have any specific requirements for the object detection network and would thus work with networks other than Faster-RCNN as well. We fuse sensor modalities at a single location, after feature extraction and before region proposal generation. Object detection networks almost entirely comprise of convolutional operations in order to exploit locality in the input data for efficient processing. Main challenge for processing different sensor modalities is to create locality in the extracted features from all sensors. We try to solve this by using a transformer module which learns to translate input features from one sensor space to another and then perform object detection on combined feature maps from all sensors.

3.1 Transformer

Key component of our proposed fusion architecture is the transformer module, which is used to transform feature maps from camera space into birdview, creating locality in the feature maps needed for further processing via convolutional layers. We adopt a convolutional encoder-decoder scheme as employed in [10]. Our transformer architecture is shown in figure 3. It is not dissimilar to an autoencoder. In contrast to traditional autoencoders however, its purpose is not to recreate its original input after it has passed through a

low-capacity bottleneck layer, but to transform input data to be similar to data from another sensor domain. It represents a mapping $P : F_{ego} \rightarrow F_{birdview}$ from ego (camera) feature maps to birdview feature maps. Input to the encoder are feature maps F_{ego} from the camera space feature detector. These feature maps contain feature activations with locality w.r.t. to the input camera space. A cascade of convolutions with a stride of two downsamples these feature maps, which are then fed through two fully-connected network layers and are upsampled through transposed convolutions to generate the output feature map $F_{birdview}$.
The main idea behind the encoder-decoder architecture is that by squeezing information through a fully-connected bottleneck layer and then upsampling this information to the desired output space (birdview in our case), the transformer is forced to represent information in an perspective-invariant manner while the convolutional encoder-decoder has to learn a projection from one space to the other. The output space is enforced by letting the system predict objects in the desired output space.

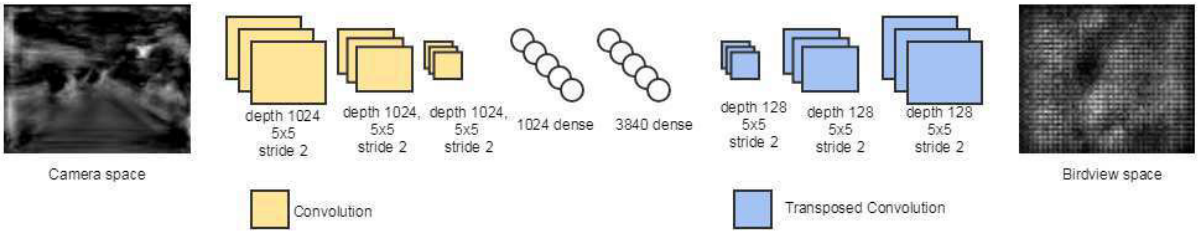


Abbildung 3: Transformer: Encoder-decoder architecture for mapping feature maps from ego space to birdview space. Rectangular elements denote convolutional operations while circle-shaped elements are fully-connected layers.

3.2 Fusion

Since the transformer module outputs transformed ego space feature maps which match the birdview Lidar feature maps in dimension and locality, any method for combining feature maps is applicable. Popular schemes include *element-wise mean* (as adopted in [5] or [1]) or *concatenation* operation (used in [1]). For our experiments we use depth-wise concatenation, meaning a depth-wise stacking of feature maps.

4 Training

For training the multi-modal object detection network we use a three step training method: First, we train **baseline single-modality networks** on camera and projected birdview Lidar images. In the second step we train the **transformer** by taking feature-maps of the previously trained camera network, feed them through the transformer and into the object detection stage of the trained Lidar network. In the third step we train the **fusion network** by taking the pretrained feature extractors of the camera and Lidar networks as well as the transformer and train a second stage object detector on the fused output featuremaps. Objective function for all training steps is the default loss function

as employed in the original Faster-RCNN paper [7], minimizing error over class and object location for each of the anchor boxes that the network predicts:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i L_{reg}(t_i, t_i^*) \tag{1}$$

L_{pred} denotes the *classification loss* while L_{reg} is the bounding box regression loss. The objective function is applied to the first stage of the network, where only objectness is predicted (binary class, object or no object) as well as for the second stage, where final multi-class classification and bounding box location refinement are performed. Note that we use the same end-to-end objective function for the different training phases of the fusion network (baseline single-modality object detector, transformer and fusion) and only freeze different parts of the network weights.

4.1 Training of transformer

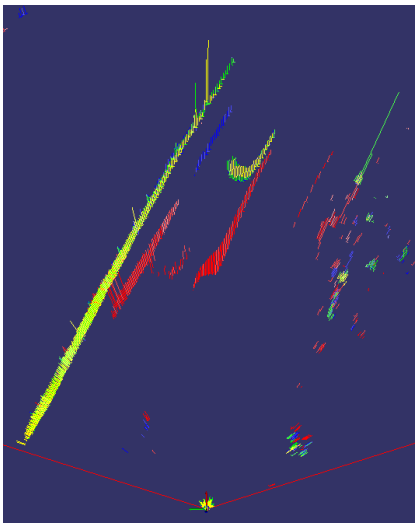
To train the perspective transformer, we take the pre-trained *feature extractor* from the camera object detection model and feed it through the transformer into the region-proposal-network (RPN) and detector network of the pre-trained birdview Lidar model. We thus use ego space camera images to predict objects in birdview space. Fixing weights of all pre-trained modules forces the transformer to learn the projection from ego features to birdview features.

4.2 Training of fusion network

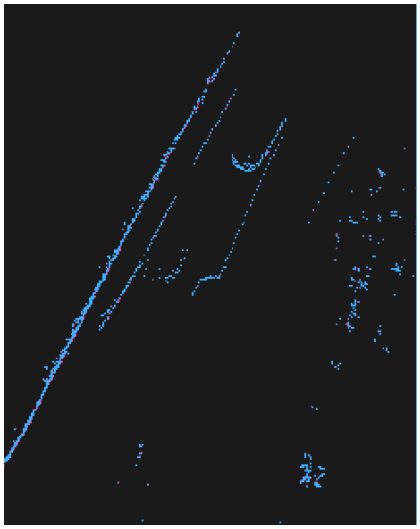
The entire fusion network is trained just like the single-modality object detector network, with the only difference being that feature maps from ego space are now transformed into birdview space by the transformer and concatenated with those of the birdview Lidar feature extractor. Objective is to predict objects in birdview space. We can employ the default Faster-RCNN training procedure as the network is otherwise unmodified. Different schemes of pre-training and weight-fixing different parts of the network are imaginable, depending on the particular use-case.

5 Experiments

We possess our own hand-labeled dataset containing approximately 16h of raw front camera images and ScaLa 4-channel Lidar pointclouds. We project 3D Lidar data into 2D birdview by employing a scheme introduced by [1]: Using two grids, we encode height and intensity of the Lidar reflection with highest azimuth angle per grid location (see figure 4). For experiments we use a subset of 1h length, sampled at 10 Hz and comprised of mixed driving (highway/urban/city) at daytime (see figure 1). Two subsequences of consecutive frames, amounting to 20% of the total length of the sequence, have been put aside for validation, the rest has been used as training data. Reported mean average precision (mAP) figures are on the validation set only. We used a modified version of the Object Detection API [4] for implementation.



(a) Original Lidar visualisation



(b) Our projection as input to the Lidar detector

(c) Camera image (cropped), input to the camera detection network

Abbildung 4: Samples of our Lidar projection. **(a)** shows the original Lidar birdview plot with length of the sticks signifying reflection intensity and color denoting one of the four channels. **(b)** shows our two channel projection, one channel encoding height and the other intensity.

5.1 Predicting objects in birdview space from camera images

We train the transformer module with input feature maps from a pre-trained camera-only feature extractor, while providing the transformer's output to a pre-trained RPN and detector module for birdview objects. Figure 5 shows input and output feature maps of the transformer after training to gain results shown in table 1. The transformer successfully learns a mapping from egoview to birdview, allowing the subsequent detector stage to predict birdview objects (and thus implicitly predicting object distance from camera images). Furthermore, the transformer outputs features so that they match the ones from the native Lidar feature extractor, since the detector module was originally trained with those and is now fixed.

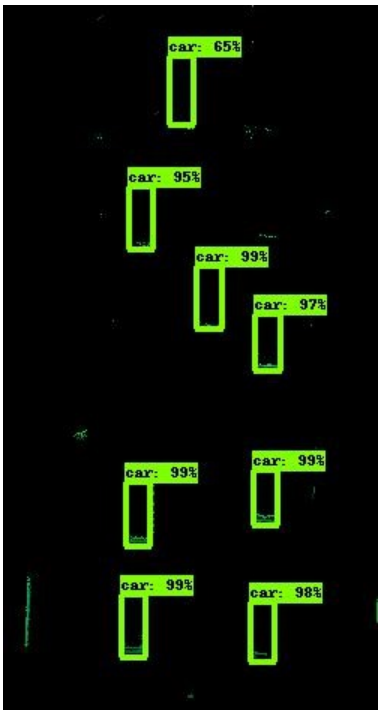
5.2 Fusion of camera and Lidar

For fusion experiments, we use the setup as depicted in figure 2 and described in section 3. Camera-space feature maps are fed to the transformer module and afterwards to the fusion module, where they are concatenated with the feature maps from the Lidar network. Weights of the feature extractors are fixed, while the transformer and detector network are being trained. Objective is to predict object's positions (i.e. bounding box coordinates)

(a) Input camera image

(b) Camera feature maps

(c) Transformed feature maps



(d) Fusion predictions

Abbildung 5: (a) shows a sample camera input image of our test set. (b) shows an exemplary plot of the first twelve (of 1024) features maps F_{ego} for a sample frame as input to the transformer. (c) shows resulting transformed feature maps $F_{birdview}$ as output by the transformer after being trained as described in section 4.1. Using $F_{birdview}$ only, we can successfully predict objects' position and class, as shown in section 5.1. Together with birdview features maps from the pre-trained Lidar object detector, we can then predict objects in birdview using both input modalities (shown in section 5.2). (d) shows predictions of object locations for the sample frame. The ego vehicle is at the bottom of the frame, facing upwards.

in birdview. To show effectiveness of our fusion scheme, we provide reference results for a Lidar-only object detector which has been trained under the same conditions (just without the camera path enabled). For results refer to table 1. We can report an increase of 0.17 and 0.07 mAP for classes *truck* and *car* for our employed dataset, showing effectiveness

of our proposed fusion scheme.

Tabelle 1: Results on our validation set. We report mean average precision (mAP) at bounding box intersection-over-union (IOU) 0.5.

	Birdview from ego camera	Lidar-only	Fusion of Lidar/camera
mAP Car	0.40	0.80	**0.87**
mAP Truck	0.54	0.63	**0.80**

6 Conclusion

We propose a novel and simple method for learning a transformation of feature maps from one sensor space to another. It allows for multi-modal object detection using state-of-the-art convolution-based object detection networks. Key advantage of the proposed method is that it allows most parts of the object detector networks to be pre-trained using only a single modality, catering to special requirements employing neural network object detectors in a series-production environment (such as multiple sensor combinations or changing sensor models for different variants of a production car). Experiments show that we can successfully predict objects in birdview space from camera images alone, showing effectiveness of the proposed transformer module, and use this transformer module to fuse camera and Lidar data for improved object detection performance.

Literatur

[1] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. *arXiv preprint arXiv:1611.07759*, 2016.

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. arXiv: 1512.03385.

[4] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv:1611.10012 [cs]*, November 2016. arXiv: 1611.10012.

[5] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d Proposal Generation and Object Detection from View Aggregation. *arXiv:1712.02294 [cs]*, December 2017. arXiv: 1712.02294.

[6] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum PointNets for 3d Object Detection from RGB-D Data. *arXiv:1711.08488 [cs]*, November 2017. arXiv: 1711.08488.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*, June 2015. arXiv: 1506.01497.

[8] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-YOLO: Real-time 3d Object Detection on Point Clouds. *arXiv:1803.06199 [cs]*, March 2018. arXiv: 1803.06199.

[9] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, September 2014. arXiv: 1409.1556.

[10] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective Transformer Nets: Learning Single-View 3d Object Reconstruction without 3d Supervision. *arXiv:1612.00814 [cs]*, December 2016. arXiv: 1612.00814.

[11] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3d Object Detection. *arXiv:1711.06396 [cs]*, November 2017. arXiv: 1711.06396.

Predicting Objects of Interest with Deep Learning

Stefan Hoermann* and Klaus Dietmayer*

Abstract: As a continuation of learned object detection in dynamic occupancy grid maps, this work aims on long-term prediction of individual objects of interest using deep learning. The algorithms focus on a complex shared space scenario including cars, bikes and pedestrians. A main advantage of the approach is that no manual labeling is required for both, object detection and prediction, due to acausal object track and shape refinement. Predicting future occupancy of single objects given the whole perceived scene, special attention is paid on the interactive influence between dynamic objects and also the peripherals.

Keywords: Automatic Labeling, Deep Learning, Dynamic Occupancy Grid Map, Long-Term Prediction, Object Detection

1 Introduction

Predictive driving behavior makes human drivers superior to today's automated vehicles. Thus, situation prediction in terms of motion or trajectory prediction is a key component to advance automated vehicles to smart systems. From the uncertain nature of prediction, it derives the objective to reduce the prediction uncertainty as much as possible, but without gaining false negative predictions. Besides kinematic limitations, interactions and context are the main sources of information to reduce the uncertainty of the evolution of a scene. Convolutional neural networks (CNNs) are known for their capability to exploit context, and as shown in the experiments of this work they are also capable to model human interactions. A dynamic occupancy grid map (DOGMa) [1] includes the entire perceivable scene in an image-like structure, naturally suggesting the usage of CNNs for situation analysis. However, the representation is model-free, what means that grid cells are considered to be independent and no hypotheses of objects, e.g. a list of vectors containing shape and pose, are provided. Although, long-term prediction can be learned model-free on cell basis [2], modeled objects are favorable for further processing like decision making. Object detection, nonetheless, is a critical task in automated driving. A misinterpretation between static background, clutter and dynamic objects can result in fatal accidents [3, 4]. A complementing use of grid maps and modeled objects seems reasonable to avoid a trade-off between false positive and false negative object hypotheses in classical early fusion stages. We argue that predicting single objects of interest in addition to the model-free prediction [2] of the entire scene can increase robustness and safety of an overall system. In this work, therefore, detected objects are predicted individually employing a CNN as motion predictor given the entire perceived DOGMa and a selection mask of single detected objects from [5]. An often claimed disadvantage

*The authors are with the Institute of Measurement, Control, and Microtechnology at Ulm University, Albert-Einstein-Allee 41, 89081 Ulm (firstname.lastname@uni-ulm.de).

of artificial neuronal networks is the high demand for training examples. Therefore, we use automatic generated labels [6]. The concept is based on offline sequence processing, in which an object trajectory is extracted forward and backward in time.

Deep learning on grid maps gains great popularity. Piewak et al. [7] employed a CNN to reduce false positive velocity estimates in a DOGMa with the goal to improve classical clustering techniques to unite cells of the same object. Learned object detection is studied with great interest in the computer vision community. In particular, the concept of anchors [8], where default boxes are classified and the offset to the true object is estimated by regression, gained great success. We adapted these techniques to object detection in grid maps [5]. The group around Posner published the Deep Tracking series [9, 10] where recurrent network structures are employed to resolve occlusions in a grid map via temporal grid cell prediction and intrinsic dynamics estimation. To clarify, the word tracking in this work does not refer to state estimation of moving objects. Deep Tracking can be seen as a learned alternative to the sequential filter employed in the DOGMa from Nuss [1]. The presented work is complementing our previous work [2], where a DOGMa is feed to a CNN used as long-term motion predictor. Instead of predicting the future occupancy of the entire scene, in the present work only selected objects are predicted.

In the remaining document, an overview of the framework is given in Section 2, features of dataset and training are given in Section 3. Experiments are carried out in Section 4 followed by conclusions.

2 System Overview

Inspired by Zhan et al. [11], considering all objects, static and dynamic, with object state $x^{(\alpha)}$ where objects are enumerated by α, the entire scene is described by $\mathcal{X} = \{x^{(1)}, x^{(2)}, ...\}$. From a planning perspective, it is favorable to get future object states $x_+^{(\alpha)}$ as predictions $p(x_+^{(\alpha)}|\mathcal{X})$. Unfortunately, the classic approach of object detection, tracking and prediction

$$\mathcal{X} \rightarrow \{p(x^{(1)}), p(x^{(2)}), ...\} \rightarrow \{p(x_+^{(1)}), p(x_+^{(2)}), ...\} \tag{1}$$

suffers already in the early stage from error prone object detections. Forward-looking driving with long-term prediction is more sensitive to false positive object detection than pure reactive driving. Also, describing the entire static and dynamic scene as object vectors can be complex without hard simplifications. An alternative to the object domain is the grid domain, where the environment is divided in a grid of cells c. For each cell the probability that the location is occupied $p(\mathrm{O}^{(c)}(\mathcal{X}))$ is estimated. Dynamic grid maps, in addition, estimate dynamic states $\mathrm{d}^{(c)}(\mathcal{X})$ without individual object hypotheses. Following this concept and feeding a DOGMa into neural networks, our system provides three outputs, as illustrated in Figure 1. The first branch transforms the grid cell representation into object domain and provides

$$\left(x^{(\alpha)}, r^{(\alpha)} = P(x^{(\alpha)}|\mathrm{O}(\mathcal{X}), \mathrm{d}(\mathcal{X}))\right) \ , \tag{2}$$

an object hypothesis with existence probability r. It was introduced in [5] and results can be seen under https://youtu.be/Rr9LOrQMgKA. The second network predicts future scene occupancy in form of a grid map

$$p(\mathrm{O}(\mathcal{X}_+)|\mathrm{O}(\mathcal{X}), \mathrm{d}(\mathcal{X})) \ . \tag{3}$$

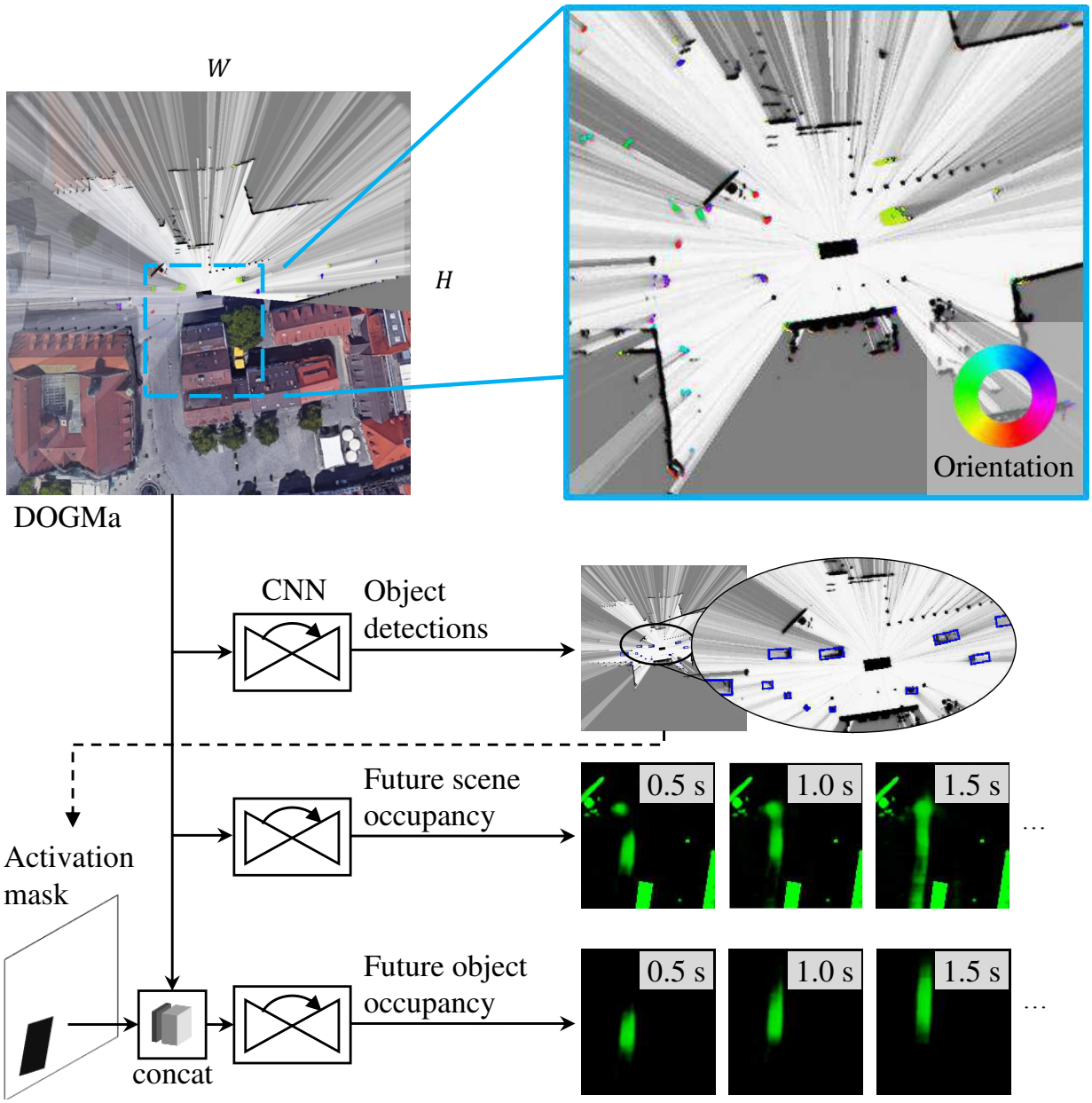


Figure 1: System overview: Dynamic occupancy grid map (DOGMa) is the input to three neural networks. The first branch detects objects in the grid fusion. The second branch provides future occupancy of the entire scene. The bottom network predicts future occupancy related to an object of interest which is selected by an additional input channel.

The network was introduced in [2] and a video showing the results can be seen under https://youtu.be/C1hhEYGs49Q. The third network is novel in this work complementing the other branches. For a selected object α, it provides predictions of future object occupancy

$$p(\mathrm{O}(x_+^{(\alpha)})|\mathrm{O}(\mathcal{X}), \mathrm{d}(\mathcal{X})) \ . \tag{4}$$

We claim that using these tools, robustness of planning with respect to false positive and false negative object detection can be improved compared to pure object based methods due to the following reasons:

- No gating/thresholding of sensor data is done since object detection is performed after sensor fusion and dynamics estimation.
- False negatives in object detection are still included in the DOGMa.
- False velocity estimates in the DOGMa are trained to be correctly predicted in scene occupancy prediction, e.g. walls with high velocity are not predicted with spatial movement.
- The influence of false positive object detections can be reduced when limiting object assessment to objects of interest.

The last bullet point might be seen controversial. However, we consider the planning task dividable in two sub tasks: reactive and strategic planning. Reactive planning, on the one hand, reacts on the environment without the intention to influence other objects and thus doesn't need object hypothesis generation, but can react on future scene occupancy from (3) to yield forward-looking driving behavior. Strategic planning, on the other hand, exploits object specific features, e.g. kinematic constraints or the capability to cooperate. To reduce the risk of considering false positive objects for strategic planning, objects of interest can be defined according to the scenario, e.g. the leading vehicle in a following scenario or oncoming vehicles in a turning scenario.

2.1 Network Structure

Except for input and output, the three networks have the same core architecture which relies on a simple convolution/deconvolution structure with skip connections similar to [12]. Input and output of all networks can be seen as a top view representation, i.e. grid map, while the spatial input dimensions are always equal to the spatial output dimensions. Pooling is used for down sampling and deconvolution for upscaling. Skip connections include a convolutional layer to reduce the channel dimension. The skip connection result is concatenated to the main tensor before upscaling via deconvolution.

2.1.1 Input and Output Data

The dynamic occupancy grid map (DOGMa) from Nuss [1] serves as input to the three CNNs. Multiple sensors are fused in a grid representation where the environment is divided in cells c. Employing a particle filter, dynamic cell states can be estimated even if velocity can not be measured directly, i.e. using lidar sensors. An example for the resulting bird's-eye image is illustrated in Figure 1. Each cell stores occupancy and

dynamics information in channels $\Omega = \left\{M_\mathrm{O}, M_\mathrm{F}, v_\mathrm{E}, v_\mathrm{N}, \sigma^2_{v_\mathrm{E}}, \sigma^2_{v_\mathrm{N}}, \sigma^2_{v_\mathrm{E},v_\mathrm{N}}\right\}$. $M_\mathrm{F} \in [0,1]$ and $M_\mathrm{O} \in [0,1]$ are the Dempster-Shafer masses for free space and occupancy, respectively. Velocity east and north components are stored in v_E and v_N, respectively. σ^2 describes the velocity variance or covariance. The occupancy probability is calculated by $p(\mathrm{O}) := 0.5 \cdot M_\mathrm{O} + 0.5 \cdot (1 - M_\mathrm{F})$. The DOGMa data is provided in $\mathbb{R}^{W \times H \times |\Omega|}$ with the spatial width W and height H. The grid is constant and the ego vehicle moves within the grid.

To select the object of interest, an additional binary channel is fed to the network with dimension $W \times H \times 1$. We call this channel the activation mask or selection mask, since grid cells covered by the object of interest are set to 1 and other cells are filled with 0.

Future scene occupancy prediction and the novel future object occupancy prediction have similar output structure. The main difference is, that object prediction contains only occupancy of a single selected object and no static/dynamic segmentation is performed. The network result $\hat{P}_\mathrm{O} \in \mathbb{R}^{W \times H \times T}$ provides occupancy maps at discrete future time steps $k \in (1, T)$. In our experiments we chose a prediction step size of 0.5 s with $t_k = t_0 + k \cdot 0.5\,\mathrm{s}$. The output interpretation of a tensor element $\hat{P}_\mathrm{O}(\mathrm{E}, \mathrm{N}, k) := \hat{P}_\mathrm{O}(c, k) := \hat{p}(\mathrm{O}_k^{(c)})$ is the occupancy probability at time step k and cell c with coordinates east E and north N. The hat symbol ˆ indicates the network output while for labels the hat is omitted.

In contrast to the result from scene occupancy prediction, where occupancy couldn't be associated to an object, the result here is completely related to the object chosen at the activation mask in the input. Normalizing the output grid maps $P_\mathrm{O}(k)$ at a single time step to the object size, i.e. number of cells, allows the probabilistic interpretation. Figure 3 illustrates an example.

3 Dataset and Training

All three tasks were trained without any manual labeling. Instead of humans annotating data, algorithms were employed to annotate over 2 h recordings. This proceeding, of course, is only successful since the annotation algorithm can use more information than the network, i.e. a whole sequence instead of only data from the past. We call this concept acausal long-term label extraction. The main benefits are as follows: Firstly, we refine estimates in the sequence. For example, velocity estimates are corrected by comparing it with actual recorded movement. Also, the estimated object shape is corrected when the object was seen from all sides during the sequence. Secondly, recorded and annotated data can be smoothed. Sequential filtering, i.e. a Kalman filter is not necessary and can be substituted by simple low pass smoothing and outlier reduction. Thirdly, we correct false positives, e.g., when an object trajectory is unreasonable, the object label is deleted. We also search for false negatives. For example, when an object is detected, it was certainly missed earlier in the sequence when it just entered the field of view. By tracking the object backwards in time, labels can be generated at these time steps. Lastly, the concept benefits from the fact that runtime is no issue, as long as the algorithm is faster than human annotations, enabling thorough calculation instead of fast approximations. A detailed explanation of the object trajectory extraction algorithm is given in [6].

Compared to human annotations, the acquired data can be considered messy. It contains approximately 5 % false labels. This has to be considered during training. Common training methods rely on perfect labels, like hard example mining where only the worst training samples of an overrepresented class are used to update the network. In our case,

however, a hard example might be a false label. Thus, we perform balancing by weighting underrepresented samples more than overrepresented examples but use a massive dataset to gain an appropriate number of underrepresented samples while keeping the number of training epochs low. Training the network to not predict corrupted labels is a matter of generalization, i.e. finding a local minimum instead of the global optimum with respect to the corrupted training data set. To gain good generalization we benefit from the possibility to produce high amounts of training data and thus, we use every example at most three times to update the network.

3.1 Spatial Balancing Loss Function

Due to the nature of temporal prediction, the result is always uncertain. Choosing an occupancy grid map representation as the network output helps to represent spatial uncertainty about the future object position. False positives, i.e. cells with $\hat{P}_{\mathrm{O}}(c) > 0$ outside the vehicle silhouette, can be interpreted as prediction uncertainty. False negatives, i.e. cells occupied by the label but with $\hat{P}_{\mathrm{O}}(c) = 0$, must be considered as prediction failure. Due to a high imbalance between occupied and free labels, we balance the loss by weighting cells with label $P_{\mathrm{O}}(c) > 0$ more with

$$L = \frac{1}{2} \sum_{k=1}^{T} \sum_{c=1}^{W \times H} \left(1 + \lambda_k P_{\mathrm{O}}(c,k)\right) \left(\hat{P}_{\mathrm{O}}(c,k) - P_{\mathrm{O}}(c,k)\right)^2 \tag{5}$$

where λ_k increases with the prediction horizon. In our experiments we chose $(\lambda_k) = (40, 50, 100, 200)$ for $(0.5\,\mathrm{s}, 1.0\,\mathrm{s}, 1.5\,\mathrm{s}, 2.0\,\mathrm{s})$, respectively.

4 Experiments

In this section we show the overall performance, as well as examples to illustrate the network capabilities for multi modal situation prediction and interaction consideration. Including GPU data transfer, the network runs with 73.5 ms per prediction on a Nvidia GTX 1080Ti. The overall prediction performance is illustrated in Figure 2. The plot compares the prediction performance of the neural network to Monte Carlo simulations. A Monte Carlo simulation forward propagates 5000 particles. The particle states were initialized with a labeled object while velocity and orientation distribution was drawn from the covariance in the occupied grid cell states. Particle acceleration a and turn rate ω are assumed to be constant over the prediction time frame but drawn from $\mathcal{N}(0, \sigma_a)$ and $\mathcal{N}(0, \sigma_\omega)$, respectively. 1000 randomly picked objects were predicted with different constellations of $(\sigma_a, \sigma_\omega)$, as well as the trained network, to exemplify the prediction performance of the CNN in terms of the relationship between false negatives and uncertainty. False positive (FP), false negative (FN) and true positive (TP) occupied cells were counted comparing any cell with predicted occupancy > 0 with labeled occupied cells. The false negative rate (FNR$=\frac{\mathrm{FN}}{\mathrm{TP+FN}}$) indicates how often the prediction fails. The uncertain area gain ($\frac{\mathrm{FP+TP}}{\mathrm{TP+FN}}$) indicates how large the predicted occupied area grows compared to the object size. It can be seen in the plot, that the network prediction fails less often and has a lower uncertainty as the first two Monte Carlo simulations. A very high process noise in the Monte Carlo run is necessary to gain similar low FNR but gaining very high uncertainty.

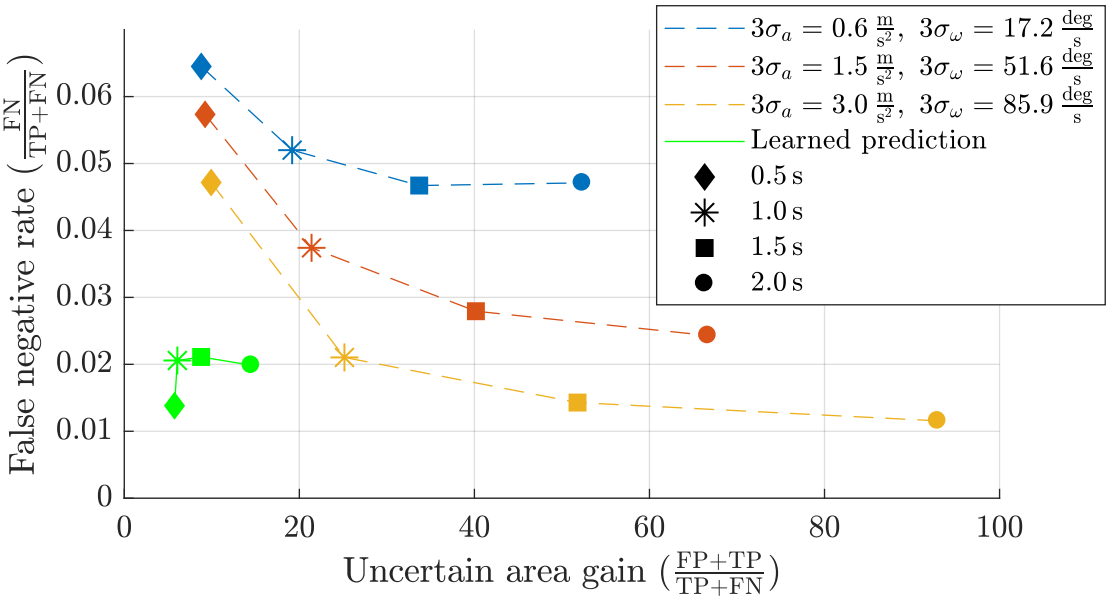


Figure 2: Comparison of learned prediction to Monte Carlo simulation. 1000 objects were predicted by the CNN and with constant velocity constant turn rate model, while 5000 particles draw process noise parameters for acceleration σ_a and turn rate σ_ω. Markers indicate prediction time steps. The plots illustrate the false negative rate, i.e. when prediction misses the true object position, over uncertainty, i.e. the area predicted as possibly occupied in ratio to occupied area. While engineered motion model prediction reaches low miss rate only with very high uncertainty, the network reaches lower miss rate with lower uncertainty in most cases.

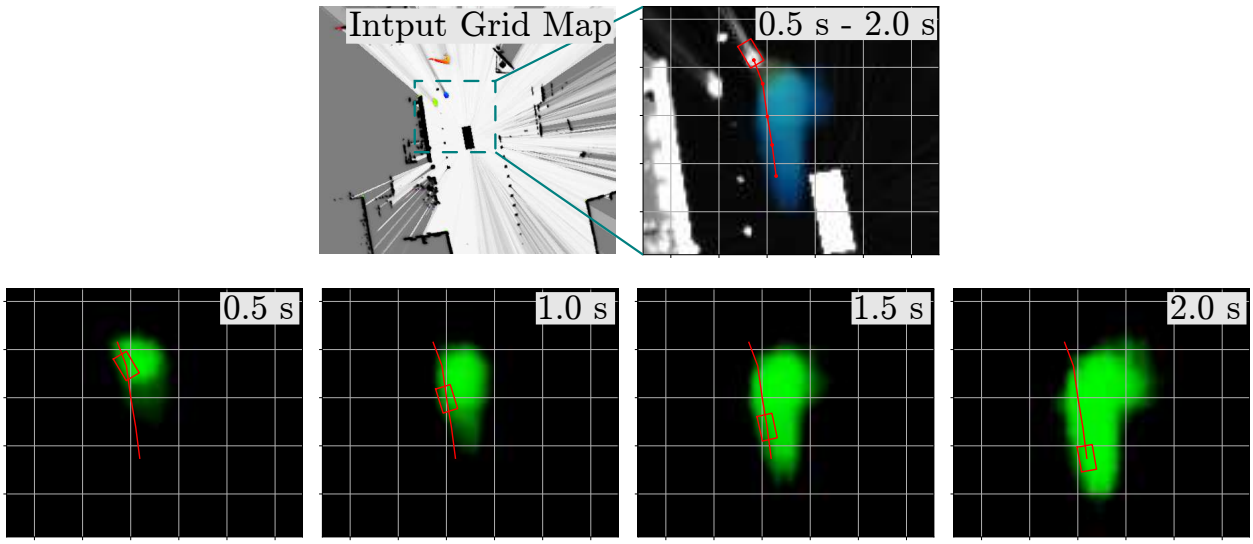


Figure 3: Example scene for object prediction: A cyclist is predicted multi modal. The complete scene is illustrated in the top left. An overview of the complete prediction result of the selected bicycle is given in the top left. Prediction results for the single time steps are shown in the bottom row. Red rectangles illustrate the true object bounding box, the red line indicates the 2 s trajectory.

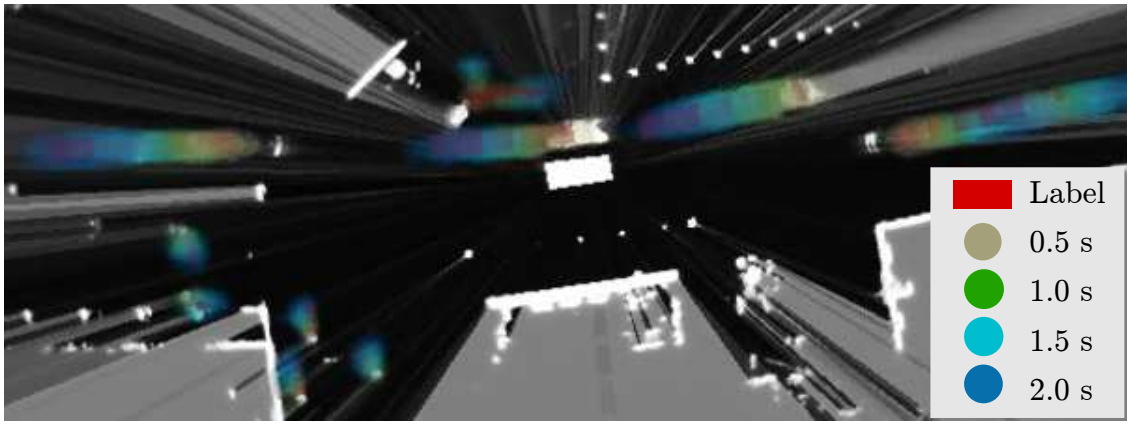


Figure 4: Example scene for object prediction: Multiple objects were predicted including pedestrians in the bottom and cars driving horizontally in center. In this image, four different prediction horizons and labels are drawn in stacked layers with transparency with the occupancy grid map of the current scene as background.

4.1 Interactions and Multi Modal Predictions

Multiple predicted objects are illustrated in Figure 4 to give an example. A single cyclist is predicted in Figure 3 showing an example for multi modal prediction capabilities. The ability to cover interactions is examined on manipulated input data, where objects are pasted close to the original trajectory of the object of interest. An example scene is illustrated in Figure 5, showing the original scene of a driving car, the same scenario with a parking car on the lane, and another example with a pedestrian approaching the lane.

5 Conclusion

To complete an existing long-term prediction framework, grid-based prediction was extended to include an individual object predictor. Learned object detection is used to create a spatial binary mask used, to select object cells in a grid map. Both object detection and object prediction was trained with automatic label generation where an engineered acausal algorithm extracts object shapes and trajectories form a sequence. Evaluation shows failure rate over uncertain area gain, a performance measure comparing predicted possible occupied area relative to actual occupied area. Experiments showed that the algorithm has less uncertainty and fails less often, in terms of false negative predictions, compared to straight forward methods. An experiment with artificially inserted traffic participants in a real scene proves intrinsically modeled interactions. In future work, generalization on other traffic scenes and prediction from a moving platform will be investigated.

References

[1] D. Nuss, *A Random Finite Set Approach for Dynamic Occupancy Grid Maps*, ser. Schriftenreihe des Instituts für Mess-, Regel- und Mikrotechnik. Universität

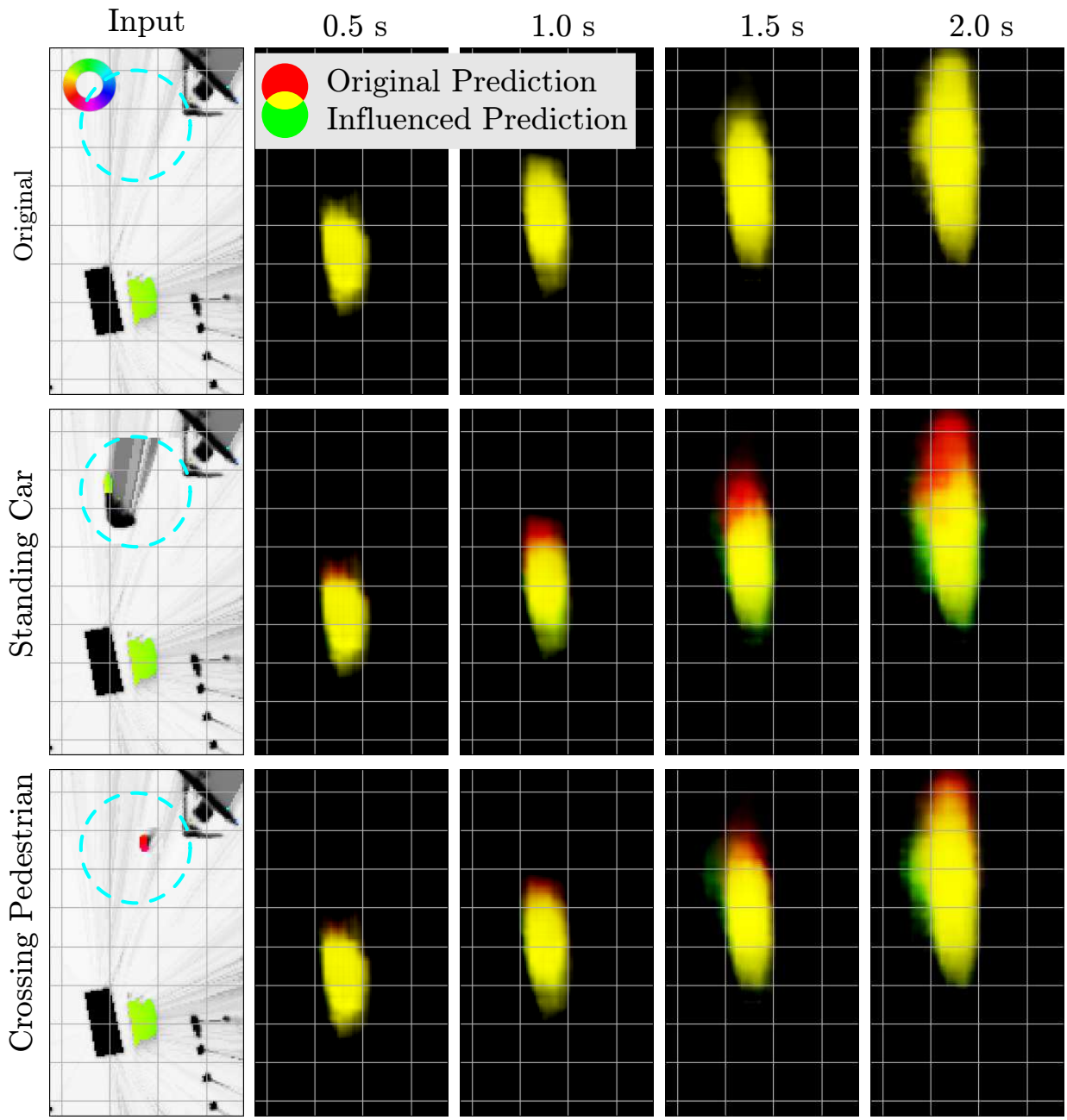


Figure 5: Influence of other vehicles on prediction: In the original scene (top row), a vehicle drives upwards on a free lane. A standing car on the lane (second row) and a pedestrian approaching the lane (third row) was pasted into the input DOGMa. The network predicts the vehicle to stop before the parking car and to slow down for the approaching pedestrian. The prediction results are illustrated in the 4 right columns as RGB images, where the original prediction is drawn in the red channel, the prediction for the manipulated input in the green channel. An overlap appears yellow.

Ulm, Institut für Mess-, Regel- und Mikrotechnik, 2017. [Online]. Available: http://dx.doi.org/10.18725/OPARU-4361

[2] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic Occupancy Grid Prediction for Urban Autonomous Driving: A Deep Learning Approach with Fully Automatic Labeling," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, to be published.

[3] Tesla driver dies in first fatal crash while using autopilot mode. Visited on 2018-07-06. [Online]. Available: https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk

[4] Technikexperte beantwortet die wichtigsten fragen zum uber-unfall. Visited on 2018-07-06. [Online]. Available: http://www.handelsblatt.com/21103570.html

[5] S. Hoermann, P. Henzler, M. Bach, and K. Dietmayer, "Object Detection on Dynamic Occupancy Grid Maps Using Deep Learning and Automatic Label Generation," in *IEEE Intelligent Vehicles Symposium Proceedings*, June 2018. [Online]. Available: https://arxiv.org/abs/1802.02202

[6] D. Stumper, F. Gies, S. Hoermann, and K. Dietmayer, "Offline Object Extraction from Dynamic Occupancy Grid Map Sequences," in *IEEE Intelligent Vehicles Symposium Proceedings*, June 2018. [Online]. Available: https://arxiv.org/abs/1804.03933

[7] F. Piewak, T. Rehfeld, M. Weber, and J. M. Zoellner, "Fully convolutional neural networks for dynamic object detection in grid maps," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 392–398.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, 2016, pp. 21–37.

[9] J. Dequaire, D. Rao, P. Ondruska, D. Z. Wang, and I. Posner, "Deep tracking on the move: Learning to track the world from a moving vehicle using recurrent neural networks," *CoRR*, vol. abs/1609.09365, 2016. [Online]. Available: http://arxiv.org/abs/1609.09365

[10] J. Dequaire, P. Ondruska, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *The International Journal of Robotics Research*, 2017. [Online]. Available: https://doi.org/10.1177/0278364917710543

[11] W. Zhan, A. Fortelle, Y. Chen, Y. Chan, and M. Tomizuka, "Probabilistic prediction from planning perspective: Problem formulation, representation simplification and evaluation metric," in *IEEE Intelligent Vehicles Symposium Proceedings*, June 2018.

[12] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1520–1528.

Differentielle Dynamische Programmierung in der Trajektorienstabilisierung

Arne-Christoph Hildebrandt, Gerald Witt, Mohit Kumar und Sven Kraus*

Zusammenfassung: Autonomes Fahren in immer komplexeren Umgebungen stellt nicht nur an die Bewegungsplanung, sondern auch an die Trajektorienstabilisierung immer höhere Anforderungen. Es geht nicht mehr nur darum eine Querablage zu einer Referenzbahn und eine longitudinale Abweichung von einer Geschwindigkeitsvorgabe zu minimieren, sondern es muss die Gesamtabweichung des Fahrzeugzustands von einer idealen Vorgabe betrachtet werden. In diesem Beitrag stellen wir ein Optimierungsverfahren, basierend auf Differentieller Dynamischer Programmierung, vor. Das Verfahren optimiert eine initiale Stellgrößenfolge, um ein exaktes Folgen einer Referenztrajektorie zu ermöglichen. Die Optimierung basiert auf einer Vorhersage der zukünftigen Regelabweichungen. Dazu können generisch verschiedene Fahrzeugmodelle verwendet werden. Anstatt das Optimierungsproblem als ein Gesamtproblem zu betrachten, teilt das vorgestellte Verfahren das Optimierungsproblem auf und löst es iterativ. Die Optimierung selbst kann jederzeit mit einer suboptimalen Lösung abgebrochen werden, was es besonders für Echtzeit-Anwendungen attraktiv macht.

Schlüsselwörter: Trajektorienoptimierung, Trajektorienstabilisierung, Fahrzeugmodellierung, Differentielle Dynamische Programmierung

1 Einleitung

Die Technik autonom fahrender Fahrzeuge wird immer greifbarer. Neben Funktionen, welche den Fahrer unterstützen, sind bereits Systeme kommerziell erhältlich, die, unter gewissen Limitierungen, das Steuern des Fahrzeugs vollständig übernehmen. Zum jetzigen Zeitpunkt werden diese Systeme vor allem für Landstraßen und Autobahnen angeboten. Vor diesem Hintergrund fängt die Forschung in Hochschulen und Industrie an, sich auf das autonome Fahren in urbanen Umgebungen zu konzentrieren.
Im Vergleich zu Schnellstraßen stellen urbane Umgebungen die Entwicklung autonom fahrender Fahrzeuge vor neue Herausforderungen. Anders als auf Schnellstraßen müssen Fahrzeuge nicht nur in der Lage sein, Fahrspuren zu wechseln oder ihnen zu folgen, sondern sie müssen auf deutlich komplexere Verkehrssituationen reagieren können.
Um auch auf komplexe Situationen noch in Echtzeit reagieren zu können, basiert die Bewegungsplanung autonomer Fahrzeuge meist auf einem hierarchischen Ansatz. Es werden mit abstrahierten Modellen des Fahrzeugs unter Berücksichtigung der Umgebung Trajektorien generiert. Diese Referenztrajektorien werden von Reglern als Vorgabe verwendet. Dabei wird häufig eine entkoppelte Regelung der Längs- und Querdynamik genutzt. [5,

*MAN Truck & Bus AG, Engineering Central Research (e-mail: arne-christoph.hildebrandt@man.eu).

2, 8, 4] Die entkoppelte Längs- und Querdynamik des Reglers vernachlässigt Kopplungseffekte der Gesamtdynamik bzw. des Gesamtfahrverhaltens des Fahrzeugs. Gerade beim urbanen Fahren ist es aufgrund von starkem Verkehrsaufkommen und schwer zu prädizierenden Situationen unabdingbar, dass das Fahrzeug nicht nur entkoppelt relativ zu einer Bahn geregelt und ein Geschwindigkeitsprofil verfolgt wird, sondern dass sich das Fahrzeug zu einem bestimmten Zeitpunkt exakt in allen seinen Zuständen auf der gewünschten Trajektorie befindet.
In diesem Beitrag versuchen wir uns einer Antwort auf folgende Fragestellung zu nähern: Wie kann, basierend auf einer ideal geplanten Referenztrajektorie, eine optimale Folge an Stellgrößen bestimmt werden, die generisch komplexe Modelle von unterschiedlichen Fahrzeugen berücksichtigt und die Abweichungen des realen Fahrzeugs zur Referenztrajektorie minimiert?
Im Folgenden werden wir ein Verfahren präsentieren, welches basierend auf [7], es ermöglicht, optimale Stellgrößen zu einer gegebenen Referenztrajektorie im Sinne einer Steuerung zu bestimmen. Das Verfahren ermöglicht, generisch Fahrzeugmodelle zu verwenden, welche Längs- und Querdyanmik berücksichtigen. Des weiteren wird das Verfahren erweitert um es als Regelung zur Trajektorienstabilisierung nutzen zu können.

2 Methodik

Wir gehen davon aus, dass als Input der hier vorgestellten Methode Trajektorien genutzt werden können, die von einer hierarchisch höheren Recheneinheit, der Bewegungsplanung [5], geplant wurden. Wir gehen davon aus, dass diese die aktuelle Verkehrssituation berücksichtigen und ein kollisionsfreies Fahren ermöglichen (vergleiche [4]). Die hier vorgestellte Methode soll die Stellgröße u, welche sich aus Lenkwinkel δ und Beschleunigung acc zusammensetzt, bestimmen, um ein möglichst exaktes Einhalten der Referenztrajektorien zu ermöglichen.

Modellierung und Kosten Die Basis unseres Algorithmus ist eine Beschreibung des Systems als dynamisches Modell

$$z_{i+1} = f(z_i, u_i) \tag{1}$$

mit dem Zustandsvektor z und der Stellgröße u. Hier ist wichtig anzumerken, dass unsere Systembeschreibung nicht auf LTI-Systeme beschränkt ist, sondern auch generisch nicht-lineare und zeit-variable Effekte abbilden kann. Die Folgen $Z = [z_0, ..., z_i]$ und $U = [u_0, ..., u_i]$ bilden die Zustands- und Stellgrößenfolgen über den Zeithorizont $t_0, ..., t_i$ ab. Die Abweichung des Fahrzeugs von den Referenztrajektorien wird über die Abweichung der Zustände zu den korrespondierenden Werten der Referenztrajektorie z_{ref} als Kosten dargestellt.
Die Kosten für Zustandsabweichungen

$$l(z) = l(z, z_{ref})$$

zusammen mit den Kosten für Stellgrößen $l(u)$ definieren die laufenden Kosten

$$l(z, u) = l(z) + l(u).$$

Über die konkrete Kostendefinition kann das Fahrzeugverhalten beeinflußt werden. Die Gesamtkosten J setzen sich aus der Summe der laufenden Kosten l und den Kosten für Abweichungen des Endzustands l_f zusammen. Der initiale Zustand z_0 ist gegeben und die Zustände $z_i \in X$ für $i > 0$ werden über die Integration von (1) über den Zeithorizont $i = 0, ..., N$ berechnet. Für die Gesamtkosten ergibt sich

$$J(z_0, U) = \sum_{i=0}^{N-1} l(z_i, u_i) + l_f(x_N). \tag{2}$$

Optimierungsproblem Das Optimierungsproblem zur Bestimmung einer optimalen Folge an Stellgrößen wird durch die Minimierung der Kostenfunktion J gelöst:

$$U^* = \arg\min_u J(z_0, U). \tag{3}$$

Das hier verwendete Verfahren zur Lösung dieses Problems basiert auf der *Dynamischen Programmierung* [1] und ihrer Erweiterung zur *Differentiellen Dynamischen Programmierung* (DDP) [7]. Der DDP Algorithmus, detailliert dargestellt in [3, 6, 7], berechnet ausgehend von einem initialem Zustand z_0 und einer nominalen Stellgrößenfolge U_0 iterativ in einem Vorwärtslauf eine nominale Zustandstrajektorie. Anschließend werden im Rückwärtslauf die Kosten (2) approximiert und lokale Verbesserungen berechnet. In der nächsten Iteration werden diese im Vorwärtslauf angebracht. Demnach wird iterativ die Stellgrößenfolge verbessert bis sie zu einem Optimum konvergiert. Algorithmus 1 fasst das Verfahren zusammen.

Algorithm 1: Zusammenfassung des Ablaufs des DDP Algorithmus.

```
N ← Länge der Zustandstrajektorie
U ← Initiale Stellgrößenfolge U_0
j ← 0
Initiale Vorärtsintegration von (1) mit U

% Optimierungsschleife
while Optimum noch nicht erreicht do
    % Gradientenberechnung
    for i=N-1:0 do
        Berechnung Kosten J_j und ihrer Ableitungen
        Berechnung lokale Verbesserungen der Stellgrößenfolge ΔU
    % Liniensuche
    while (J_j − J_{j−1}) < Δε do
        U ← U + αΔU
        Vorwärtsintegration von (1) mit U, z_0 und Berechnung J_j
        α ← α + Δα
    j ← j + 1
```

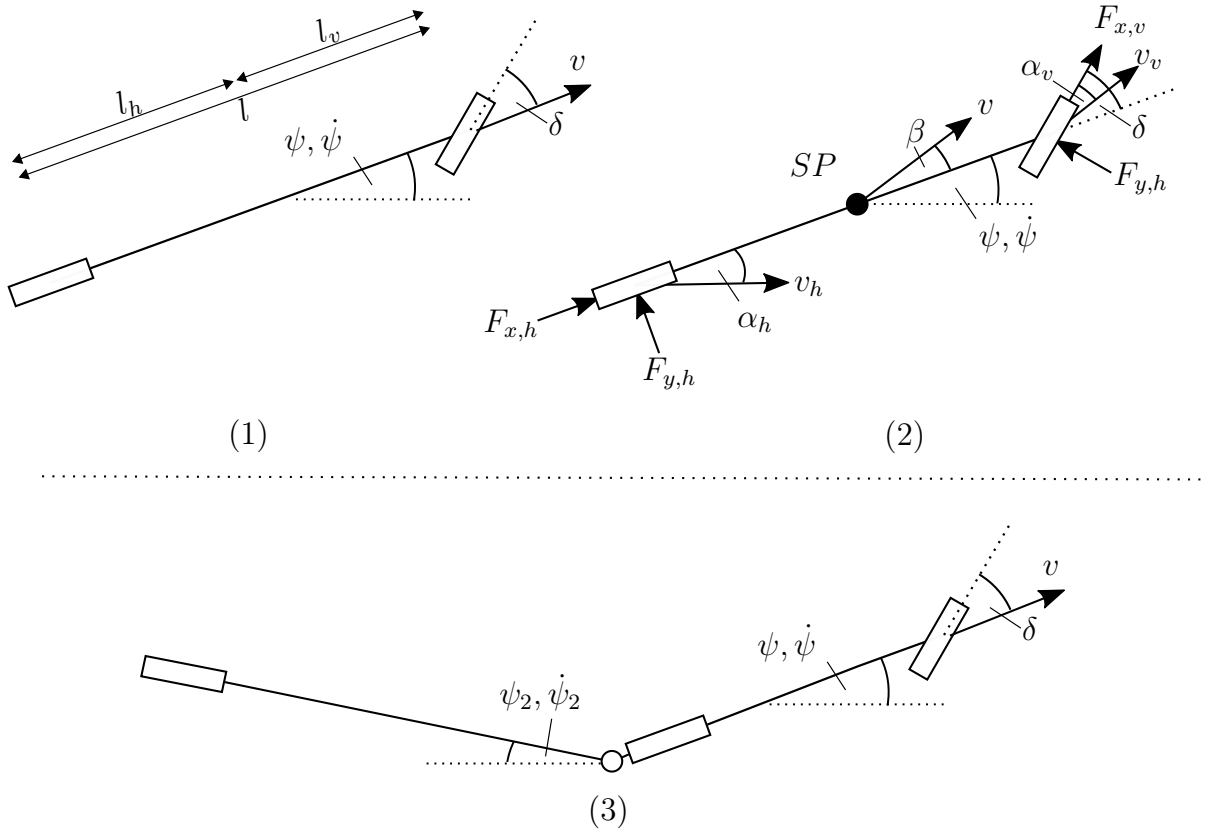
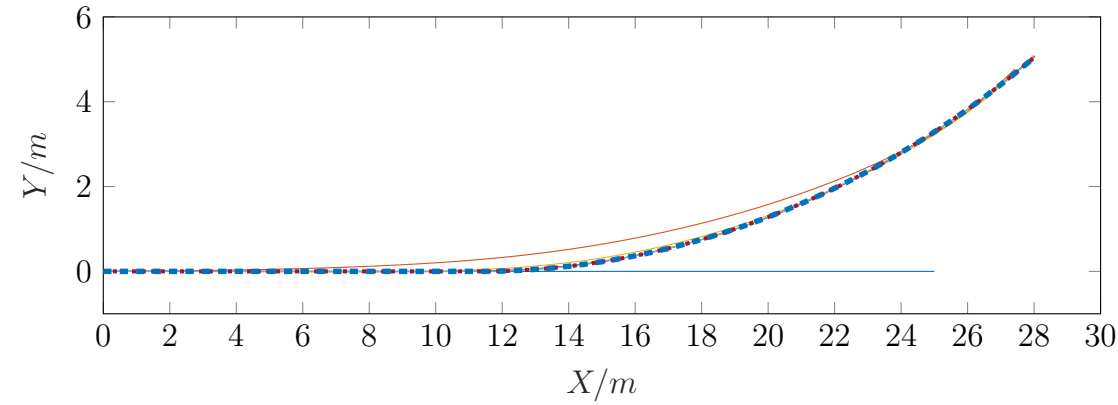


Abbildung 1: Kinematisches und dynamisches Einspurmodell, sowie das kinematische Einspurmodell erweitert um einen Anhänger. Unterschiedliche Modelle, welche hier zur Optimierung verwendet werden.

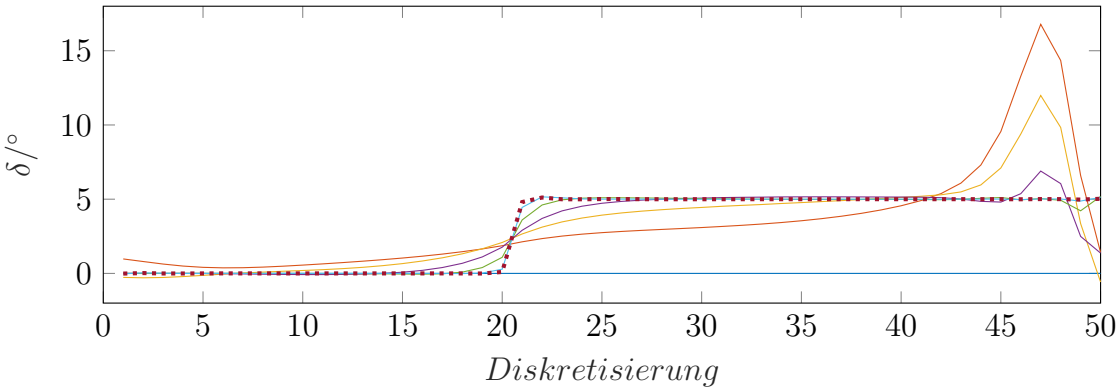
3 Anwendungen

Die großen Vorteile der vorgestellten Methode liegen darin, dass (a) beliebige Modelle in (1) verwendet werden können. (b) Zudem ermöglichen die iterativen Verbesserungen ein Abbrechen der Optimierung nach jeder Iteration. Wir haben den ursprünglich in [1] vorgestellten Algorithmus erweitert um beide Eigenschaften auszunutzen. Im Folgenden geben wir einen Einblick in unsere Erweiterungen und diskutieren mehrere Anwendungsbeispiele.

Optimierung In Abbildung 1 werden schematisch verschiedene Fahrzeugmodelle aufgezeigt, die wir für unterschiedliche Anwendungsfälle untersucht haben. In einem ersten Beispiel versuchen wir die Konvergenzeigenschaften des Optimierungsalgorithmus aufzuzeigen. In Abbildung 2 wird exemplarisch das Ergebnis des DDP Algorithmus für die Position des Fahrzeugmodells und des Lenkwinkels als eine Stellgröße der Fahrzeugbewegung dargestellt. Als Input verwenden wir eine ideale Referenztrajektorie. Der resultierende Pfad wird in Abbildung 2 (b) dargestellt. Die Geschwindigkeitverläufe haben wir aus Platzgründen vernachlässigt. Für den initialen Stellgrößenverlauf U_0 wird hier kein Vorwissen verwendet. Der Verlauf wird in Abbildung 2 dargestellt. Die Abbildung 2 zeigt sehr deutlich, dass selbst bei einer initial sehr schlechten Stellgrößenvorgabe ($U_0 = [0, ..., 0]$) sich das Fahrzeug schon nach wenigen Iterationen stark der Referenztrajektorie annähert.



(a) Zustände



(b) Lenkwinkel

Iterationen: 1, 2, 3, 4, 5, 6, 7, Referenzzustände

Abbildung 2: Exemplarischer Verlauf der Lenkwinkelstellgröße und der Position des Fahrzeugs über die Iterationen des DDP Algorithmus für ein kinematisches Einspurmodell. Als initiale Stellgröße wurde 0 für den Lenkwinkel angenommen.

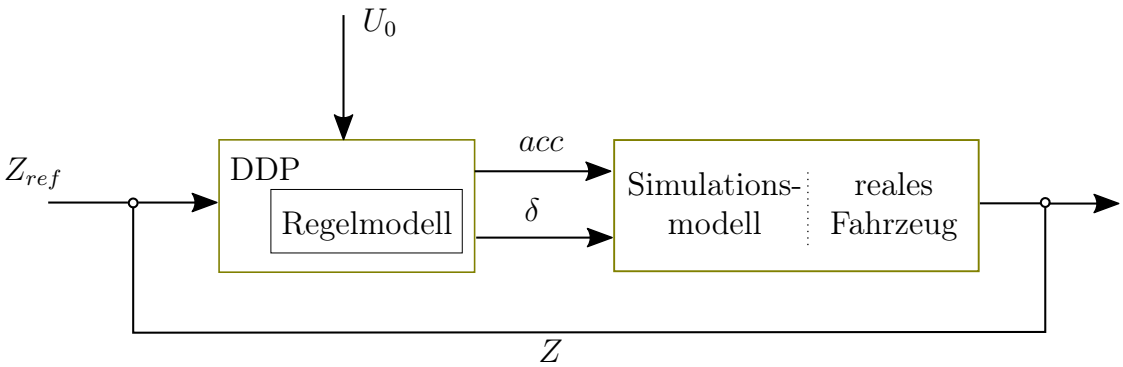
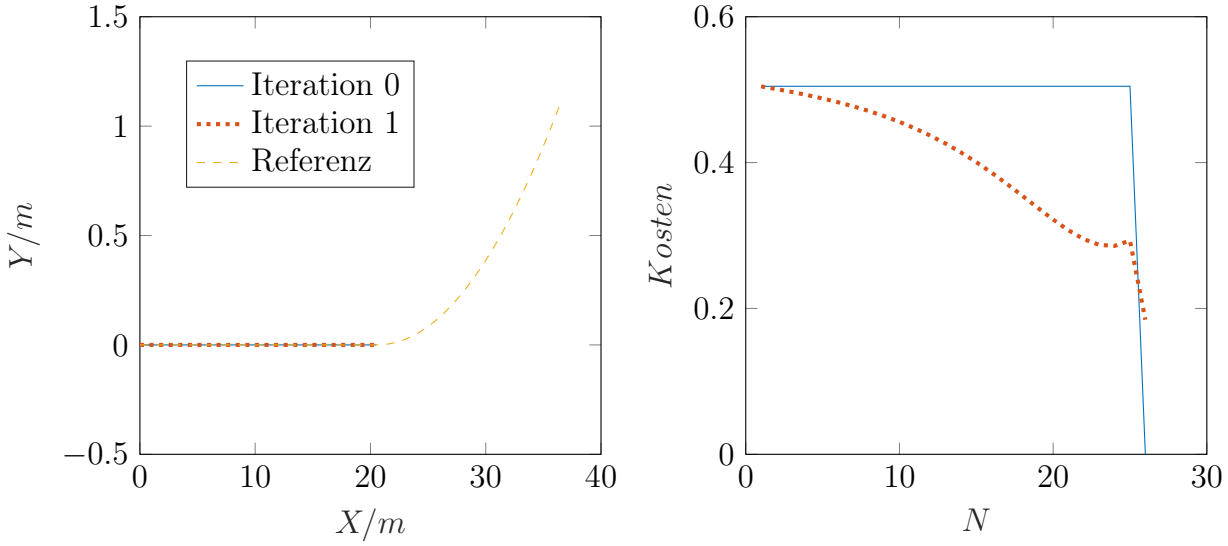


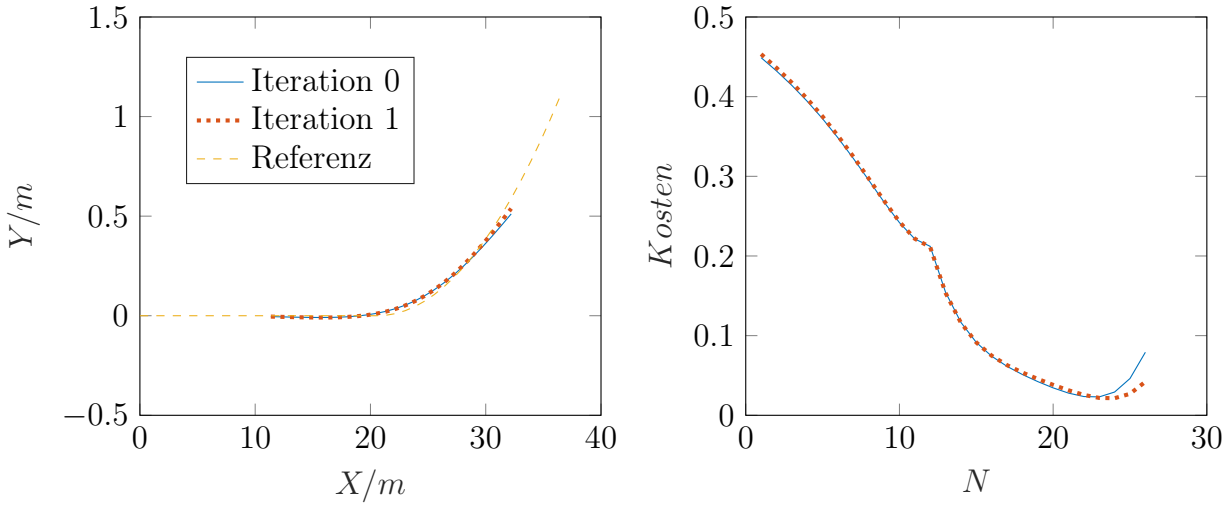
Abbildung 3: Schematische Abbildung des Aufbaus zur Untersuchung des Verfahrens und seiner Anwendung im realen Fahrzeug.

Trajektorienstabilisierung Ein weiterer Vorteil, das Optimierungsproblem nach dem Vorbild des DDP Algorithmus zu lösen, liegt darin, dass Teilprobleme rekursiv gelöst werden können und dass die Lösung des Gesamtproblems iterativ verbessert wird. Dadurch ist es möglich, einerseits viele Rechenschritte zu parallelisieren, was die Performance des Algorithmus deutlich verbessert und andererseits kann prinzipiell nach jedem Iterationsschritt die Optimierung abgebrochen werden. Damit liegt zwar keine vollständig konvergierte Lösung vor, aber eine suboptimale Lösung, welche die Initiallösung im Sinne der Kostenfunktion verbessert und in den gegebenen Zeitlimitierungen berechnet wurde. Bedingt durch diese Eigenschaften verwenden wir das vorgestellte DDP Verfahren wie folgt zur Trajektorienstabilisierung:
Zu jedem Regelzyklus wird der Algorithmus, wie in Algorithmus 1 beschrieben, aufgerufen. Der Zustand z_0 wird mit dem aktuellen Zustand des realen Fahrzeugs initialisiert. Daraufhin wird das Optimierungsproblem mit der aktuell vorliegenden Referenztrajektorie Z_{ref} als Input gelöst (vergleiche Abbildung 3). Je nach Zeithorizont der vorliegenden Referenztrajektorie kann der DDP Algorithmus als *moving horizon* Ansatz implementiert und gelöst werden. Dies bedeutet auch, dass das Optimierungsergebnis des vorherigen Regelzyklus als Initiallösung weiter verwendet werden kann, was die Konvergenz des Algorithmus nach einer Anfangsphase deutlich verbessert. Abbildung 4 und 5 zeigen die Ergebnisse für den identischen Referenzinput, wie er auch in dem vorherigen Beispiel verwendet wurde, aber mit einem dynamischen Einspurmodell als Regelmodell. Im Gegensatz zu dem vorherigen Beispiel wird nicht über den gesamten Zeithorizont optimiert, sondern der Zeithorizont im Sinne eines *moving horizon* auf $N = 25$ beschränkt. Zudem wird ein dynamisches Einspurmodell als Simulationsmodell verwendet. Um die Rechenzeit für die Anwendung im realen Fahrzeug zu reduzieren, haben wir in jedem Regelzyklus die maximale Anzahl der Iterationen auf eine beschränkt. Man kann deutlich erkennen, dass die Kostenreduzierung der Optimierung durch die Weiterverwendung der Optimierungsergebnisse mit voranschreitender Zeit deutlich abnimmt - womit sich die Lösung auch schon nach wenigen Iterationen nah an einem Optimum befindet.



(a) Trajektorien (b) Kosten

Abbildung 4: *moving horizon* Implementierung mit Zeithorizont $N = 25$. Erster Regelzyklus mit dynamischen Einspurmodell als verwendetes Regelmodel.



(a) Trajektorien (b) Kosten

Abbildung 5: *moving horizon* Implementierung mit Zeithorizont $N = 25$. 15. Regelzyklus mit dynamischen Einspurmodell als verwendetes Regelmodel.

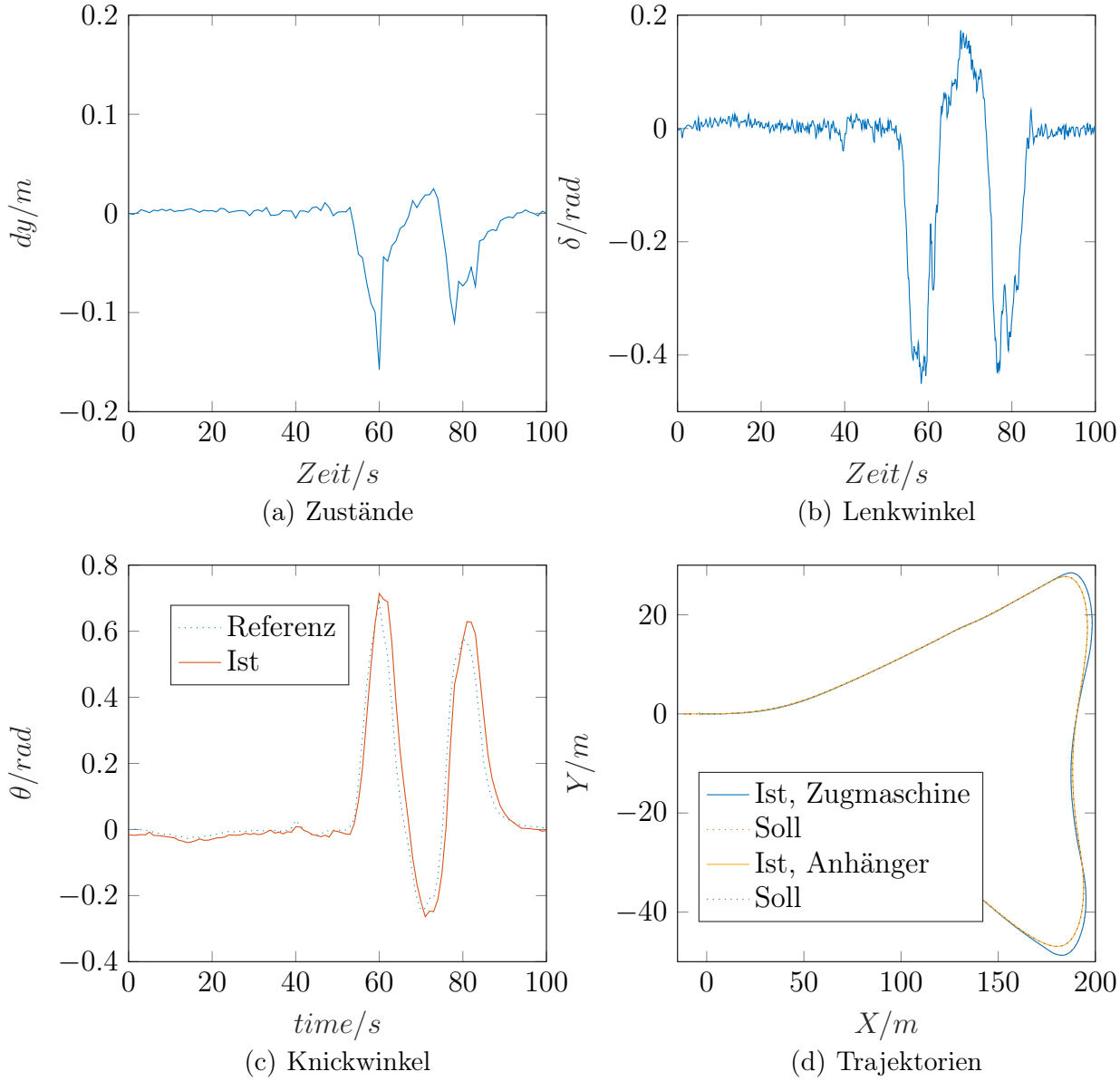


Abbildung 6: Untersuchung der *moving horizon* Implementierung für ein kinematisches Einspurmodell mit Anhänger in einer Mehrkörpersimulationsumgebung. Referenzdaten sind mit echtem Fahrzeug aufgenommen worden.

Zusätzlich haben wir das vorgestellte Verfahren in einem typischen Rangiervorgang untersucht. Der LKW fährt mit Geschwindigkeiten unter $10\frac{\mathrm{km}}{\mathrm{h}}$, in unserem Fall vorwärts, einen komplexen Pfad entlang. Der in Abbildung 6 (d) dargestellte Pfad wurde mit einem realen Fahrzeug aufgenommen. Für dieses Szenario haben wir das Regelmodell des kinematischen Einspurmodells um einen Anhänger erweitert. Im Gegensatz zu den vorherigen Untersuchungen verwenden wir zudem als Simulationsmodell ein erweitertes Mehrkörpersimulationsmodell aus einer komerziellen Software und beschränken die Stellgröße auf die Größe δ. Den Zustand Z erweitern wir um den Knickwinkel θ zwischen Zugmaschine und Anhänger. Abbildung 6 stellt neben den Referenzbahnen des Anhängers und der Zugmaschine den Regelfehler des Knickwinkels, den berechneten Lenkwinkelverlauf und die Querablage dar.

4 Zusammenfassung und Ausblick

In diesem Beitrag haben wir einen Algorithmus vorgestellt, mit dem ideale Referenztrajektorien im Sinne einer definierten Kostenfunktion optimal verfolgt werden können. Die Vorteile des vorgestellten Algorithmus liegen darin, dass (1) der Algorithmus nicht auf einer zeit-invarianten und linearisierten Modellierung des Fahrzeugs basiert. Stattdessen können generisch verschieden komplexe Modellierungsstufen genutzt werden. (2) Anders als oft in der Literatur beschrieben [2, 4], behandelt der Algorithmus Längs- und Querdynamik nicht separat, sondern kann auch Kopplungseigenschaften beider Stellgrößen berücksichtigen. (3) Zudem wird das Optimierungsproblem nicht als Gesamtproblem gelöst, sondern in Teilprobleme aufgeteilt und iterativ gelöst. Dadurch ist es gerade für Echtzeitanforderungen höchst geeignet.
Wir haben diese Eigenschaften exemplarisch in drei Anwendungen aufgezeigt. In zukünftigen Arbeiten werden wir die vorgestellte Methode in echten Fahrzeugen und auf seine Anwendbarkeit für unterschiedliche Anwendungen hin untersuchen.

Literatur

[1] Richard Bellman. *Dynamic Programming.* 1. Aufl. Princeton, NJ, USA: Princeton University Press, 1957.

[2] Benjamin Gutjahr, Lutz Gröll und Moritz Werling. "Lateral vehicle trajectory optimization using constrained linear time-varying MPC". In: *IEEE Transactions on Intelligent Transportation Systems* 18.6 (2017), S. 1586–1595.

[3] D.H. Jacobson und D.Q. Mayne. *Differential dynamic programming.* Modern analytic and computational methods in science and mathematics. American Elsevier Pub. Co., 1970.

[4] Pedro F. Lima u. a. "Spatial Model Predictive Control for Smooth and Accurate Steering of an Autonomous Truck". In: *IEEE Transactions on Intelligent Transportation Systems* 2.4 (2017), S. 238–250.

[5] Brian Paden u. a. *A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles.* arXiv:1604.07446v1, 2016.

[6] Yuval Tassa, Tom Erez und Emanuel Todorov. "Synthesis and stabilization of complex behaviors through online trajectory optimization". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2012, S. 4906–4913.

[7] Yuval Tassa, Nicolas Mansard und Emo Todorov. "Control-limited differential dynamic programming". In: *IEEE International Conference on Robotics and Automation.* 2014, S. 1168–1175.

[8] Moritz Werling. "Ein neues Konzept für die Trajektoriengenerierung und-stabilisierung in zeitkritischen Verkehrsszenarien". In: *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik* 60.1 (2012), S. 53–54.