

9. Workshop
Fahrerassistenzsysteme

FAS 2014



9. Workshop

Fahrerassistenzsysteme

FAS2014

Walting im Altmühltal, 26.-28. März 2014

Copyright Uni-DAS e.V.

Alle Rechte, auch das des auszugsweisen Nachdruckens, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany

ISBN 978-3-00-044955-0

Uni-DAS e.V.

Petersenstrasse 30

64287 Darmstadt

Telefon: 089 - 6004 4086

Telefax: 089 - 6004 2564

www.uni-das.de

Vorwort

Ausgehend vom Projekt PROMETHEUS im Jahr 1987 blicken Fahrerassistenzsysteme mittlerweile auf eine bemerkenswerte Entwicklung zurück. Deutsche und Europäische Verbundprojekte, bei denen Universitäten und Forschungseinrichtungen, Automobilhersteller und Zulieferer intensiv und vertrauensvoll zusammengearbeitet haben und zusammenarbeiten, haben zahlreiche Visionen von PROMETHEUS zu Lösungen und schließlich zu Produkten geführt. Viele der entwickelten Systeme sind mittlerweile auch in Fahrzeugen der Mittel- und Kompaktklasse angekommen.

Die nächste Herausforderung stellt zweifelsohne die Erhöhung des Automatisierungsgrades dar, die sich zunächst auf einzelne Funktionen und Manöver beschränkt und in ferner Zukunft zum vollautomatischen Fahren führen wird. Wann dieses Ziel erreicht werden kann, darüber sind sich die Experten nicht einig. Sicher ist jedoch, dass auf dem Weg dahin der Fahrer stets eine wichtige Größe im Regelkreis „Fahrer-Fahrzeug-Umwelt“ bleiben wird. Die genannten Thematiken sind Gegenstand des Workshops Fahrerassistenzsysteme. 2014 findet diese Veranstaltung zum neunten Mal statt und ist - trotz seines „ungewöhnlichen Rhythmus“ von 1,5 Jahren - mittlerweile fester Bestandteil der „FAS-Community“. Ursprünglich von M. Maurer und Ch. Stiller initiiert, ist er mittlerweile in die Verantwortlichkeit von Uni-DAS übergegangen, von dem er nun zum dritten Mal in Folge organisiert wird. Der Workshopcharakter, das heißt keine Parallelveranstaltungen, Zeit für Diskussionen innerhalb und außerhalb der wissenschaftlichen Sessions, die Arbeit in Kleingruppen an einem Thema, sowie die Beschränkung der Teilnehmerzahl hat sich über all die Jahre erhalten. Damit grenzt sich der Workshop bewusst von großen Kongressen ab.

Die einzelnen Beiträge dieses Tagungsbandes zeigen ein breites Spektrum des aktuellen Wissensstandes im Bereich der Fahrerunterstützung und des automatisierten Fahrens. Betrachtet man die Entwicklung von Fahrer-Assistenz-Systemen als langfristiges Projekt, so stellt jeder Band

der FAS-Workshop Reihe einen „Meilenstein“ dar. Es bleibt zu hoffen, dass diesem Meilenstein noch viele weitere folgen werden und am Ende das hoch- oder vollautomatisierte und vor allem unfallfreie Fahren steht.

Den Mitgliedern von Uni-DAS e.V., den Teilnehmern und ganz besonders den Sitzungsleitern möchte ich sehr herzlich für Ihre Mithilfe zum Gelingen des Workshops Fahrerassistenzsysteme im Jahr 2014 danken.

Walting, im März 2014

Berthold Färber

Inhalt

Seite

Wahrnehmung

- 1 **Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten**
R. Matthaei, B. Lichte, M. Maurer, TU Braunschweig 7
- 2 **Merkmalsbasiertes Map-Matching von radarbasierten Belegungskarten**
M. Rapp, M. Hahn, M. Muntzinger, J. Dickmann, K. Dietmayer, Universität Ulm 17
- 3 **Hocheffizientes Stereosehen durch Modellierung geneigter Ebenen und Parallelverarbeitung**
B. Ranft, T. Strauß, Forschungszentrum Informatik 27
- 4 **Registration of a Multi-Camera System with a 3D Laser Range Finder**
J. Esparza, L. Vepa, M. Helmle, B. Jähne, Robert Bosch GmbH 37

Prädiktion und Entscheidung

- 5 **Generatives Prädiktionsmodell zur frühzeitigen Spurwechselerkennung**
M. Bahram, A. Lohrer, M. Aeberhard, BMW Forschung und Technik 47
- 6 **Situation Prediction And Reaction Control (SPARC)**
M. Ruf, J. Ziehn, B. Rosenhahn, J. Beyerer, D. Willersinn, H. Gotzig, Fraunhofer Institut für Optronik 55

- 7 **Potenzial der Kopfposenerkennung zur Absichtsvorhersage von Fußgängern im urbanen Verkehr**
H. Klöden, BMW Forschung und Technik 67

Konzepte/Anwendungen

- 8 **Kartengestütztes automatisiertes Fahren auf der Berta-Benz-Route von Mannheim nach Pforzheim**
J. Ziegler, P. Bender, H. Lategahn, M. Schreiber, T. Strauß, C. Stiller, Forschungszentrum Informatik 79
- 9 **Crowdsourcing als Enabler für verbesserte Assistenzsysteme: Ein generischer Ansatz zum Erlernen von Kreuzungsparametern**
C. Ruhhammer, A. Atanasov, F. Klanner, C. Stiller, BMW AG/ KIT 95
- 10 **Fences - A Unified Architecture for Mapping Static Environment for Driver Assistance Systems**
T. Kubertschak, M. Mählich, H.-J. Wünsche, AUDI AG/UniBw 105

Aufmerksamkeit/Kontrollierbarkeit

- 11 **Kontrollierbarkeit unerwarteter Eingriffe eines Bremsassistenzsystems aus Perspektive des Folgeverkehrs**
A. Neukum, F. Naujoks, S. Kappes, T. Wey, WIVW 115
- 12 **Neuartige Warnstrategien zur Ausrichtung der Aufmerksamkeit bei Kollisionsgefahr**
L. Lorenz, T. Lindberg, K. Bengler, BMW Forschung und Technik/TU München 127

Evaluation

- 13 **Validierung von Engstellenszenarien und Querdynamik im dynamischen Fahrsimulator und Vehicle in the Loop**
F. Rüger, C. Purucker, N. Schneider, A. Neukum, B. Färber, UniBW München/WIVW 137

-
- 14 **Evaluation einer taktischen Verhaltensentscheidungsfindung für Fahrstreifenwechsel beim vollautomatisierten Fahren in Städten**
S. Ulbrich, M. Maurer, TU Braunschweig **147**
- 15 **Exposition zur Raffung der Felderprobung für Fahrerassistenzsysteme**
P. Glauner, A. Blumenstock, M. Haueis, Daimler AG **159**
- 16 **Systematische Auswertung von Testfällen für Fahrfunktionen im modularen virtuellen Testbaukasten**
F. Schuldt, B. Lichte, M. Maurer, S. Scholz, TU Braunschweig **169**

Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten

Richard Matthaei, Bernd Lichte, Markus Maurer*

Zusammenfassung: Zukünftige Fahrerassistenzsysteme stellen hohe Anforderungen an die maschinelle Umfeldwahrnehmung, insbesondere in städtischer Umgebung. Verfügbare Sensoren und Algorithmen sind derzeit noch nicht in der Lage, das städtische Umfeld in der geforderten Vollständigkeit und Robustheit in Echtzeit zu erfassen. Daher werden Kartendaten als zusätzliche Informationsquelle eingesetzt. Um die Kartendaten nutzen zu können, ist eine hoch genaue Pose (Position und Orientierung) in der Karte erforderlich. In diesem Beitrag wird ein Ansatz vorgestellt und untersucht, der die Poseschätzung lediglich durch einen Abgleich des wahrgenommenen Straßen- bzw. Fahrstreifenverlaufs mit dem Fahrstreifenverlauf aus den Kartendaten vornimmt und somit auf aufwendigere Detailkarten mit Punktlandmarken oder gitterbasierten Daten verzichtet.

Schlüsselwörter: enhanced maps, fahrstreifengenaue Karten, map-relative localization, kartenrelative Lokalisierung

1 Motivation

Mit der Entwicklung zukünftiger Fahrerassistenzsysteme (z. B. eines innerstädtischen Kreuzungsassistenten, wie er beispielsweise in Intersafe¹ und GENEVA² vorgestellt wurde) wird deutlich, wie wichtig eine umfassende Szenenrepräsentation für eine robuste Fahrentscheidung ist. Bestandteil dieser Szenenrepräsentation sind die Fahrstreifenverläufe – für die Kreuzungsassistenten vorzugsweise der gesamten Kreuzung. Da heutige Systeme nicht in der Lage sind, diese Fahrstreifeninformation vollständig zu erfassen (vgl. Bar Hillel u. a. (2012)), werden ergänzend zu weiteren Forschungsaktivitäten zur Verbesserung der maschinellen Fahrstreifendetektion Kartendaten als A-Priori-Information in das System integriert. Die Nutzung von Kartendaten eröffnet dem System zwei Vorteile: Zum einen kann die Wahrnehmung *innerhalb des Sensorsichtbereichs* unterstützt werden, zum anderen erweitern die Kartendaten aber auch den Sensorsichtbereich durch Ergänzung von Umgebungsdaten *außerhalb des Sensorsichtbereichs*. Dadurch ist es beispielsweise möglich, auch außerhalb des Erfassungsbereichs der maschinellen Fahrstreifenerkennung, detektierte Objekte (andere Verkehrsteilnehmer) einem Fahrstreifen zuzuordnen. Für diese Zuordnung ist eine hoch genaue Pose in der Karte erforderlich, die

*Institut für Regelungstechnik der TU Braunschweig, Hans-Sommer-Straße 66, 38106 Braunschweig (E-Mail: <matthaei,lichte,maurer>@ifr.ing.tu-bs.de).

¹http://cordis.europa.eu/projects/rcn/87267_en.html

²<http://www.geneva-fp7.eu>

aufgrund von Kartenfehlern und Fehlern der satellitenbasierten Ortung in der Stadt derzeit keine hohe Verfügbarkeit aufweist. Ein möglicher Ansatz ist, durch einen Abgleich der wahrgenommenen Umfeldbestandteile mit den Kartendaten eine entsprechende Ortungslösung in der Karte zu erhalten (engl. *map-relative localization*). Durch die hohe Genauigkeit der verwendeten Umfeldsensoren ist die Pose des Ego-Fahrzeugs relativ zur Umgebung präzise bekannt, wohingegen das GPS-Signal gerade im innerstädtischen Bereich nur eine rudimentäre Pose bereitstellt. Ein Abgleich der Umgebungsdaten mit den Kartendaten würde sowohl Fehler der satellitenbasierten Lokalisierung als auch geometrische Fehler in der Karte kompensieren. Die Verfügbarkeit eines solchen Ansatzes kann erhöht werden, indem nicht nur Fahrbahnmarkierungen als Informationen der stationären Umgebung genutzt werden, sondern auch die erhabene Randbebauung in den Kartenabgleich mit einbezogen wird. Die Verwendung der erhabenen Randbebauung ermöglicht somit auch den Einsatz in kleinen innerstädtischen Seitenstraßen. Darüber hinaus gehen wir nicht davon aus, dass hoch genaue Karten mit Punktlandmarken (wie Pfosten, Bäume, etc.) oder gitterbasierte Karten in absehbarer Zeit flächendeckend in Serienfahrzeugen verfügbar sein werden. Daher untersuchen wir Ansätze, die mit weniger präzisen und detaillierten Kartendaten (hier fahstreifengenaue Karten) arbeiten.

2 Herleitung der Anforderungen

Die Anforderungen an eine derartige kartenrelative Lokalisierung leiten sich aus der Applikation ab. Im Folgenden sei als Beispiel weiterhin der Kreuzungsassistent angeführt. Der Kreuzungsassistent warnt den Fahrer vor einer möglichen Kollision mit entgegenkommendem Verkehr. Dabei hilft eine korrekte Fahstreifenzuordnung des entgegenkommenden Objekts, die Fehlalarmrate zu reduzieren (Herrmann u. Schroven, 2012). Dieser Ansatz der Fahrerabsichtserkennung ist eine alternative Lösung zu C2C-Kommunikation oder kamerabasierter Blinkerdetektion. Für eine robuste Bewertung der Kritikalität stehender oder fahrender Objekte darf bei einer eigenen Geschwindigkeit von 50 km/h in einer Entfernung d von 60 m der Positionsfehler der Objekte Err_{obj} die halbe Fahstreifenbreite (ca. $1,5 \text{ m}$) nicht überschreiten. Unter Vernachlässigung des longitudinalen Fehlers relativ zur Straße in der Karte lässt sich der Zusammenhang zwischen Winkelfehler Err_{ψ} und lateralem Fehler Err_{lat} wie folgt bestimmen:

$$Err_{\psi} = \arctan\left(\frac{Err_{obj} - Err_{lat}}{d}\right)$$

Bei einem Positionsfehler von beispielsweise $0,75 \text{ m}$ des Ego-Fahrzeugs bleibt in 60 m Entfernung unter Vernachlässigung der Fehler durch die Umfeldsensoren noch ein maximal zulässiger Winkelfehler der Fahrzeugpose in der Karte von ca. $0,7^{\circ}$.

Ergänzend zu diesen rein funktionalen Anforderungen soll der Ansatz außerdem den Einsatz von seriennaher Sensorik erlauben. Das beinhaltet zum einen, dass nur ein Sensor eingesetzt wird und zum anderen dass dieser automotive-tauglich zum Beispiel in der Front des Fahrzeugs integriert ist.

3 Stand der Forschung

Die Lokalisierung in digitalen Karten mit Fahrstreifeninformation erfolgt bisher häufig rein GNSS-basiert bzw. gestützt um eine Eigenbewegungsschätzung mit Koppelnavigation (z.B. Knaup u. Homeier (2010), Wille u. a. (2010), Toledo-Moreo u. a. (2010)) und ist somit anfällig gegenüber GNSS-Empfangsproblemen und Kartenfehlern. Ein Lösungsansatz, diese Einschränkungen zu umgehen, besteht in der Nutzung von Umfeldinformationen für einen Kartenabgleich. Dieser Abgleich zwischen Kartendaten und Umfeldmerkmalen kann prinzipiell auf drei unterschiedliche Arten erfolgen:

1. Abstraktion der eingehenden Sensordaten durch Modellannahmen, bis sie das Abstraktionslevel der Kartendaten erreichen.
2. Konkretisierung der Kartendaten mit Modellannahmen, bis das Repräsentationslevel der Sensordaten erreicht ist.
3. Annäherung der Abstraktionslevel durch modellbasierte Veränderung beider Repräsentationen.

Die maximal erreichbare Genauigkeit der Lokalisierungslösung ist dabei abhängig von der Abstraktionsebene der Merkmale. Dieser Zusammenhang ist beispielsweise auch schon in Hock (1994) beschrieben. Hock unterscheidet zwischen zwei Klassen der Landmarken: Typische Landmarken der Klasse 1 seien demnach Kurvenein- und ausfahrten (also markante Merkmale des Straßenverlaufs), die lediglich der Grobpositionierung dienen. Mit Landmarken der Klasse 2 hingegen könne eine präzise Lokalisierung durchgeführt werden. Sie werden als dreidimensionale Objekte modelliert. (Hock, 1994, S. 65)

Ein Beispiel für die vollständige Abstraktion der eingehenden Sensordaten wird in Müller u. a. (2011) vorgestellt. Hier wird die Straßentopologie anhand von gitterbasierten Distanzkarten ermittelt und zur Lokalisierung mittels Partikelfilter in Kartendaten verwendet. Basierend auf den Erfahrungen aus der DARPA Urban Challenge wird in Miller u. a. (2011) ein Ansatz vorgestellt, der abstrahierte Fahrstreifenmitten und Stopplinienspositionen mit Kartendaten abgleicht. Die Fahrstreifenmitten und Stopplinienspositionen werden über eine Kamera wahrgenommen. Die kartenrelative Fahrzeugpose wird ebenfalls mittels Partikelfilter bestimmt.

Der umgekehrte Weg, die Kartendaten in Richtung der Sensordaten zu konkretisieren, wird beispielsweise in Levinson (2011, S. 42ff) und Montemerlo u. a. (2008) vorgestellt und z. B. in Wille (2012, S. 97) aufgegriffen. In diesem Ansatz wird eine Positionsschätzung der Fahrbahnmarkierung auf Basis der Kartendaten angenommen, und mit den Laser-Rohdaten abgeglichen. Da der Abgleich nur auf Höhe der Fahrzeughinterachse stattfindet, kann nur der laterale Versatz bestimmt werden. Die Fehler in longitudinaler Richtung und in der Orientierung bleiben bestehen. Außerdem müssen die wahrgenommenen Merkmale interpretiert werden. Ein weiterer Ansatz ist in Szczot u. a. (2010) vorgestellt worden. Hier werden sowohl die Modellparameter für die Konkretisierung als auch die Fahrzeugpose in einem Partikelfilter geschätzt. Ähnlich zu dem Ansatz aus Levinson (2011, S. 42ff) werden hohe Intensitäten an den Fahrbahnrandern und niedrige Intensitäten in der Fahrbahnmitte erwartet. Der Abgleich wird an mehreren Positionen entlang der Straße durchgeführt und direkt mit der gitterbasierten Repräsentation der Radarintensitäten abgeglichen. Ein sehr ähnlicher Ansatz ist in Schüle u. a. (2011) vorgestellt worden.

Ein Ansatz, in dem sowohl die eingehenden Sensordaten abstrahiert als auch die Kartendaten konkretisiert werden, wurde in Konrad u. a. (2011) veröffentlicht. Während dieses Abgleichs werden die Straßenbegrenzungen zunächst ohne Unterstützung durch Kartendaten extrahiert und dann mit den berechneten Begrenzungen aus dem Straßenverlauf abgeglichen.

Andere Ansätze, wie Lategahn u. Stiller (2012), Weiss (2011) oder Levinson (2011, S. 15ff) erfordern eine hoch detaillierte und hoch genaue Umfeldkarte mit Landmarken auf Gitter- oder Punktobjektebene. Diese Ansätze arbeiten mit Kartendaten, die auf einer anderen Merkmalebene (nach Hock (1994) Landmarken der Klasse 2) unterhalb der Fahrstreifenverläufe einzuordnen sind und sich damit von dem im Rahmen unserer Untersuchungen eingesetzten Kartenmaterial unterscheiden. Die fahrstreifengenauen Karten nehmen in der Klassifikation nach Hock (1994) eine Zwischenrolle ein. Sie lassen sich keiner der beiden dort vorgestellten Klassen eindeutig zuordnen.

Viele der während der DARPA Urban Challenge 2007 entwickelten Ansätze, die eine Lokalisierung in der zur Verfügung gestellten Karte (Route Network Definition File, RNDf) durch Umfeldmerkmale gestützt haben, lassen sich nur bedingt auf eine seriennahe Plattform übertragen. Die Randbedingungen in den DARPA-Wettbewerben waren nur bedingt vergleichbar mit denen, die durch den Serieneinsatz im realen Stadtverkehr gelten (Bar Hillel u. a., 2012). Die verwendete Sensorik sowie deren Verbauposition an der Dachreling sind nicht automotive-tauglich. Außerdem übersteigt die Anzahl der verwendeten Sensoren signifikant das zulässige Budget und das Umfeld entsprach nur begrenzt einem realen städtischen Szenario (z.B. geringere Geschwindigkeiten, signifikant mehr Platz und niedrige Bebauung, vgl. auch Wille (2012, S. 52ff)).

4 Lokalisierung in fahrstreifengenauen Karten

4.1 Extraktion der Umfeldmerkmale

Als Umfeldmerkmale kommen bei diesem Ansatz lediglich die Straßenfluchten zum Einsatz. Die Extraktion der Straßenfluchten aus Laserdaten wurde in Matthaei u. a. (2013) vorgestellt. Aus einer gitterbasierten Darstellung wird mit Hilfe eines suchenden Algorithmus abschnittsweise der Straßenverlauf extrahiert. Auf diese Weise ist es auch möglich, Straßenverläufe in der Stadt zu verfolgen, die nicht zwangsläufig dem Modell einer Klothoide folgen. Zudem sind im Vergleich zu Autobahnen oder Landstraßen enge Kurvenradien extrahierbar mit einer Vorausschau von bis zu 40 m (siehe Abbildung 1).

Die vorausschauende Extraktion des Straßenverlaufs ist dabei weitestgehend entkoppelt von der Fahrzeugeigenbewegung, so dass auch bei Fahrmanövern, bei denen das eigene Fahrzeug nicht mehr parallel zur Fahrbahn steht (wie bei Spurwechseln, Ein- oder Ausparken, Abbiegen oder Ausweichmanövern), die Richtung des eigentlichen Straßenverlaufs weiterhin korrekt extrahiert wird. Durch die Fusion von erhabener Randbebauung und Fahrbahnmarkierungen (soweit sie durch die verwendete Sensorik erfasst werden können) und dem Verzicht auf weitere Modellannahmen, bieten diese Merkmale insgesamt im urbanen Bereich eine hohe Verfügbarkeit.

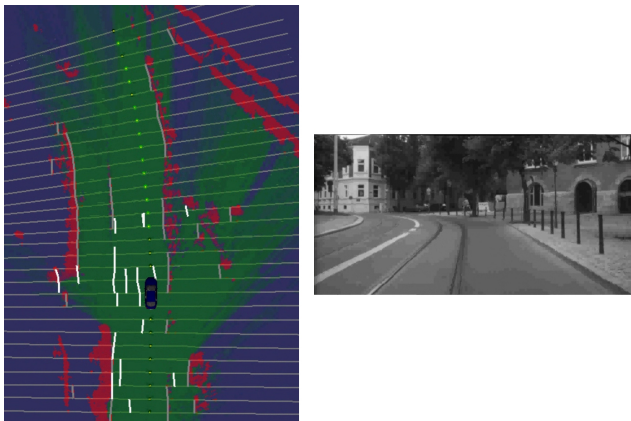


Abbildung 1: Links: Extraktion der Straßenflucht in einer Kurve mit einer Reichweite von 40 m aus Laserdaten. Graue Linien: Fluchten der erhobenen Randbebauung, weiße Linien: Fluchten von Bordsteinen und Fahrbahnmarkierungen. Im Hintergrund: Belegungsgitter. Eine detaillierte Beschreibung der Extraktion wurde in Matthaei u. a. (2013) vorgestellt. Rechts: Referenzbild aus der Kamera.

4.2 Poseschätzung

Im Bereich der GPS-basierten Lokalisierung sind Kalmanfilter weit verbreitet. Diese kommen bei Lokalisierungsaufgaben auf Basis von Umfeldmerkmalen ebenfalls zum Einsatz (vgl. Weiss (2011, S. 62)). Auch in diesem Ansatz fällt die Wahl auf eine kalmanfilterbasierte Lösung, zumal es hier primär um die Untersuchung von Merkmalen auf deren Eignung zur Lokalisierung in Karten mit Fahrstreifeninformation geht.

Durch den Abgleich der Umgebung mit den Kartendaten ist es möglich, die Orientierung Ψ_{est} des Fahrzeugs in der Karte direkt zu bestimmen. Daher bietet es sich an, auch die Filterstruktur diesen günstigen Umständen anzupassen und die Kopplung von Position und Orientierung nur in einer Richtung zuzulassen. So haben Sprünge in der Positionsbestimmung (x_{match}, y_{match}) keine Auswirkungen mehr auf die geschätzte Fahrzeugorientierung Ψ_{est} und wir können einen robusten Bewegungsvektor mit der Geschwindigkeit v und der Orientierung Ψ_{est} ermitteln. Eine mögliche Filterarchitektur ist in Abbildung 2 dargestellt.

Die Orientierung in der Karte Ψ_{est} wird mit einem Kalmanfilter geschätzt. Wir nutzen die distanznormierte Gierrate $\frac{d\Psi}{ds}$ der Fahrzeugeigenbewegung als Steuergröße des Kalmanfilters. Die gewählte Modularität der Filterarchitektur erlaubt es, die Fahrzeugeigenbewegung bereits durch Steuergrößen wie den Lenkwinkel zu stabilisieren. Die verbleibende Drift der Gierrate wird bestimmt aus der Differenz der Fahrzeugdrehung in der Karte und der Fahrzeugdrehung auf Basis der Eigenbewegungsschätzung. Die Drift wird ebenfalls über den Steuereingang in den Filterprozess integriert und steht dem Filter auch während eines GPS-Ausfalls zur Verfügung. Die Varianz var_{Ψ} der kartenrelativen

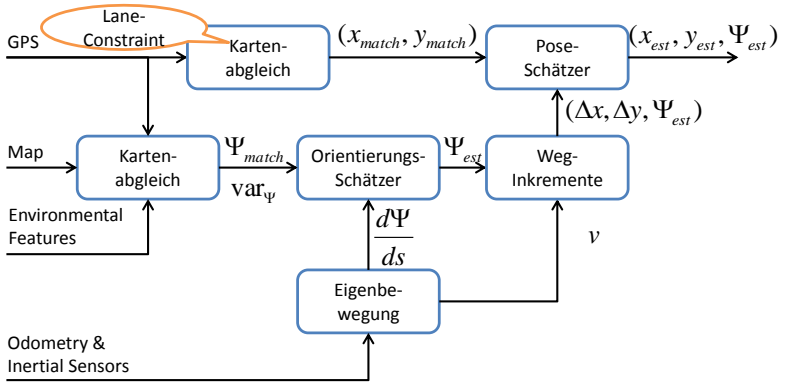


Abbildung 2: Filterarchitektur zur kartenrelativen Poseschätzung gestützt durch eine kartenrelative Orientierung Ψ_{est} . Die kartenrelative Orientierung Ψ_{match} wird durch einen Abgleich der Kartendaten mit den Umfeldmerkmalen gewonnen. Die Position (x_{match}, y_{match}) wird durch die Projektion der GPS-Position auf die Kanten der fahrstreifenengenauen Karte ermittelt („lane-constraint“).

Orientierung Ψ_{match} wird auf Basis einer statistischen Auswertung der vielen einzelnen Abgleiche zwischen den einzelnen Straßenflucht und Kartendaten bestimmt (siehe Abbildung 3).

Die gefilterte Orientierung Ψ_{est} wird dann genutzt, um in Kombination mit der Fahrzeuggeschwindigkeit die inkrementellen Wegstrecken $(\Delta x, \Delta y)$ zu bestimmen. Diese stützen ebenfalls über den Steuereingang des Filters die Position des Fahrzeugs in der Karte. Die Position (x_{match}, y_{match}) wird über eine biaskorrigierte GPS-Position ermittelt. Wir bestimmen den lateralen Bias durch einen geometrischen Map-Matching-Ansatz unter der Annahme, dass sich das Fahrzeug in der Fahrstreifenmitte befindet (engl. „lane-constraint“), ähnlich der bekannten Map-Matching-Ansätze, die annehmen, dass sich das Fahrzeug auf der Straße befindet (engl. „road-constraint“). Da wir keine sinnvollen Varianzen aus dem Matching-Prozess ableiten können, müssen feste Werte für die Varianzen experimentell ermittelt werden. Wir haben die Varianzen so gewählt, dass die Position entlang der Straße dem GPS folgt und die Position quer zur Straße hauptsächlich durch die kartenrelative Bewegung bestimmt wird.

Durch diesen Filteransatz ist ausschließlich die Fahrzeugorientierung Ψ_{est} der kartenrelativen Pose $(x_{est}, y_{est}, \Psi_{est})$ vollständig vom GPS entkoppelt. In longitudinaler Richtung ist das GPS-Signal dominant und verhindert so eine Langzeitdrift. In lateraler Richtung zur Straße hilft die Bias-Schätzung, niederfrequente Karten- und GPS-Fehler zu kompensieren. Da die Lösung insgesamt hauptsächlich dem Bewegungsvektor folgt, haben GPS-Sprünge keine direkte Auswirkung auf das Endergebnis der Pose-Schätzung.

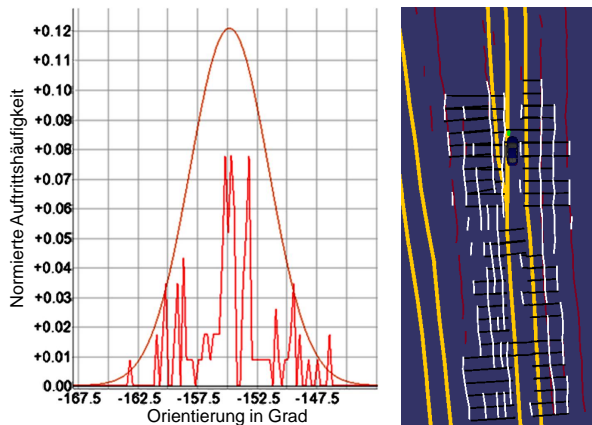


Abbildung 3: Links: Approximation der Fehler aus dem Kartenabgleich durch Normalverteilung. Rechts: Abgleich der einzelnen Fluchtelemente mit den Kartendaten. Zur besseren Visualisierung wurden die Assoziationen der stationären Randbebauung an die Kartendaten hier nicht eingezeichnet.

5 Ergebnisse

Der Versuchsträger ist mit einem seriennahen Laserscanner, einem Novatel-Empfänger zur Generierung eines seriennahen GPS-Signals sowie einem Seriensensor zur Bestimmung der Fahrzeugeigenbewegung ausgestattet. Der Laserscanner hat 4 Scan-Ebenen und einen horizontalen Öffnungswinkel von ca. 140° . Eine Scan-Ebene tastet dabei im wesentlichen die Fahrbahnoberfläche ab. Der Sensor wurde auf einer Höhe von ca. 30 cm in die Fahrzeugfront integriert. Für die Referenzmessung wurden die Inertialplattform RT3000 der Firma Oxford sowie eine von uns manuell erstellte, hoch genaue Karte mit Fahrstreifeninformation verwendet. Die Karte weist zu Testzwecken einen nahezu kontinuierlichen Kurvenverlauf auf. Die Teststrecke umfasst ca. 3 km einer städtischen Einfallstraße in Braunschweig. Dabei wurden vier unterschiedliche Auswertungen vorgenommen: 1. reines GPS-Signal, 2. reine geometrische Projektion der GPS-Position auf die Fahrstreifen, 3. GPS-Signal gestützt durch Orientierungsinformation und Bias durch Kartenabgleich und 4. GPS-Signal gestützt durch Orientierungsinformation und Map-Matching bei GPS-Ausfällen (94% der Strecke ohne GPS). Ein Vergleich mit einer Ortung, die ausschließlich auf der Schätzung der Eigenbewegung durch Fahrzeugsensoren basiert, ist aufgrund der Drift in Kombination mit den langen Ausfallzeiten nicht sinnvoll möglich. Für die Funktion sind primär Orientierung und lateraler Fehler von Bedeutung. Außerdem von Interesse ist eine Aussage über die Verfügbarkeit einer verwertbaren Objektposition in 60 m Entfernung (weniger als 1,5 m lateraler Fehler). Die Ergebnisse sind in Tabelle 1 dargestellt. Eine Veranschaulichung des Kartenabgleichs ist in Abbildung 4 gegeben.

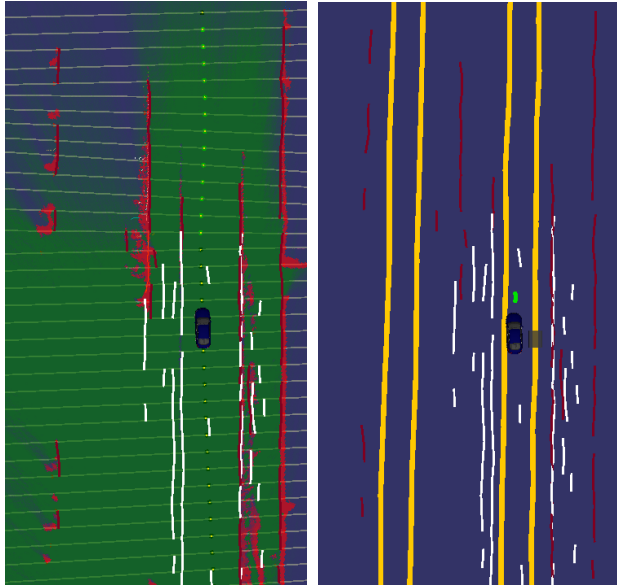


Abbildung 4: Karteneinpassung durch Stützung der Orientierung aus Umfeldmerkmalen sowie einer Bias-Korrektur aus der Kantenzuordnung. Links: Extraktion des Straßenverlaufs. Belegungsgritter im Hintergrund. Weiße Linien: Markierungen und Bordsteine, rote Linien: erhabene Randbebauung. Rechts: Einpassung in fahrstreifengenaue Karten. Grüne Punkte: Position der Referenz, graue Box: Ergebnis der Kantenzuordnung (hier fehlerhaft). Fahrzeug: geschätzte kartenrelative Pose.

	reines GPS	Fahrstreifenzuordnung ohne GPS-Ausfall	Kartenabgleich	Kartenabgleich mit GPS-Ausfall
Orientierung < 0,7°	75,7%	69,1%	86,6%	83,9%
lat. Positionenfehler < 0,75 m	8,3%	77,8%	69,5%	67,3%
Objektposition < 1,5 m in 60 m	42,8%	75,3%	83,0 %	74,5%

Tabelle 1: Ergebnisse der Referenzierung. Prozentangaben beschreiben die Verfügbarkeit anteilig an der gefahrenen Strecke. GPS-Ausfall auf 94% der Strecke.

6 Zusammenfassung und Ausblick

Die Ergebnisse mit realen Sensordaten zeigen, dass mit diesem Ansatz eine Verbesserung der Orientierung in hoch genauen Kartendaten mit Fahrstreifeninformation möglich ist. Mit der modularen Filterstruktur konnten außerdem Verbesserungen in der lateralen Positionsbestimmung erzielt werden, so dass insgesamt die Verfügbarkeit der geforderten Lokalisierungsgenauigkeit in 60 m Entfernung vor dem Fahrzeug auf der Teststrecke unter Verwendung von seriennaher Sensorik deutlich verbessert werden konnte. Weitere Messfahrten haben gezeigt, dass bei der Verwendung automatisch generierter Kartendaten die Verfügbarkeit der hoch genauen Orientierung aus dem Abgleich in etwa den Charakteristiken der Orientierung aus dem GPS gleicht. So kann mit dem vorgestellten Verfahren der laterale Fehler verringert werden und bei GPS-Ausfall weiterhin eine Stützung der Position erfolgen. Die Robustheit dieses Ansatzes liegt in den geringen Anforderungen an die Interpretationsleistung sowie der Vielzahl der Merkmale, die innerstädtisch eine hohe Verfügbarkeit aufweisen. Der Algorithmus läuft inklusive der gitterbasierten Verarbeitung in Echtzeit (Sensortakt 40 ms) auf einem aktuellen PC mit einem I5-Prozessor.

Für eine weitere Verbesserung der Position sowie einem Ausgleich von Kartenfehlern ist der Abgleich der Fahrstreifenmitten mit dem Fahrstreifenetz der Karte erforderlich. Außerdem soll ergänzend zu der hier vorgestellten modularen Filterarchitektur das Partikelfilter auf seine Eignung untersucht werden.

Literatur

- [Bar Hillel u. a. 2012] BAR HILLEL, Aharon ; LERNER, Ronen ; LEVI, Dan ; RAZ, Guy: Recent progress in road and lane detection: a survey. In: *Machine Vision and Applications* (2012), S. 1–19
- [Herrmann u. Schroven 2012] HERRMANN, Simon ; SCHROVEN, Frank: Situation analysis for driver assistance systems at urban intersections. In: *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*, 2012, S. 151–156
- [Hock 1994] HOCK, Christian J. L.: *Wissensbasierte Fahrzeugführung mit Landmarken für autonome Roboter*. München, Universität der Bundeswehr, Diss., 1994
- [Knaup u. Homeier 2010] KNAUP, Jörn ; HOMEIER, Kai: RoadGraph - Graph based environmental modelling and function independent situation analysis for driver assistance systems. In: *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. Madeira, Portugal, September 2010, S. 428–432
- [Konrad u. a. 2011] KONRAD, Marcus ; SZCZOT, Magdalena ; SCHÜLE, Florian ; DIETMAYER, Klaus: Generic grid mapping for road course estimation. In: *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011, S. 851–856
- [Lategahn u. Stiller 2012] LATEGAHN, Henning ; STILLER, Christoph: Experimente zur hochpräzisen landmarkenbasierten Eigenlokalisierung in unsicherheitsbehafteten digitalen Karten. In: *8. Workshop Fahrerassistenzsysteme*. Walting, 2012
- [Levinson 2011] LEVINSON, Jesse S.: *Automatic Laser Calibration, Mapping, and Localization for Autonomous Vehicles*, Stanford University, Diss., 2011

- [Matthaei u. a. 2013] MATTHAEI, Richard ; LICHTER, Bernd ; MAURER, Markus: Robust Grid-Based Road Detection for ADAS and Autonomous Vehicles in Urban Environments. In: *Proceedings of the 16th International Conference on Information Fusion (FUSION), 2013*. Istanbul, Turkey, Juli 2013, S. 1–7
- [Miller u. a. 2011] MILLER, Isaac ; CAMPBELL, Mark ; HUTTENLOCHER, Dan: Map-aided localization in sparse global positioning system environments using vision and particle filtering. In: *Journal of Field Robotics* 28 (2011), September, Nr. 5, S. 619–643
- [Montemerlo u. a. 2008] MONTEMERLO, Michael ; BECKER, Jan ; BHAT, Suhrid ; DAHLKAMP, Hendrik ; DOLGOV, Dmitri ; ETTINGER, Scott ; HAEHNEL, Dirk ; HILDEN, Tim ; HOFFMANN, Gabe ; HUHNEKE, Burkhard ; JOHNSTON, Doug ; KLUMPP, Stefan ; LANGER, Dirk ; LEVANDOWSKI, Anthony ; LEVINSON, Jesse ; MARCIL, Julien ; ORENSTEIN, David ; PAEFGEN, Johannes ; PENNY, Isaac ; PETROVSKAYA, Anna ; PFLUEGER, Mike ; STANEK, Ganymed ; STAVENS, David ; VOGT, Antone ; THRUN, Sebastian: Junior: The Stanford entry in the Urban Challenge. In: *Journal of Field Robotics* 25 (2008), September, Nr. 9, S. 569–597
- [Müller u. a. 2011] MÜLLER, Andre ; HIMMELSBACH, Michael ; LÜTTEL, Thorsten ; HUNDELSHAUSEN, Felix v. ; WÜNSCHE, Hans-Joachim: GIS-based topological robot localization through LIDAR crossroad detection. In: *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011, S. 2001–2008
- [Schüle u. a. 2011] SCHÜLE, Florian ; SCHWEIGER, Roland ; LÖHLEIN, Otto ; DIETMAYER, Klaus: Vehicle Positioning on a Digital Map for Road Course Prediction. Tokyo, Japan, 2011
- [Szczoł u. a. 2010] SZCZOT, Magdalena ; SERFLING, Matthias ; LÖHLEIN, Otto ; SCHÜLE, Florian ; KONRAD, Marcus ; DIETMAYER, Klaus: Global positioning using a digital map and an imaging radar sensor. In: *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, S. 406–411
- [Toledo-Moreo u. a. 2010] TOLEDO-MOREO, Rafael ; BETAÏLLE, David ; PEYRET, Francois: Lane-Level Integrity Provision for Navigation and Map Matching With GNSS, Dead Reckoning, and Enhanced Maps. In: *IEEE Transactions on Intelligent Transportation Systems* 11 (2010), März, Nr. 1, S. 100–112
- [Weiss 2011] WEISS, Thorsten-Tobias: *Hochgenaue Positionierung und Kartographie mit Laserscannern für Fahrerassistenzsysteme*, Universität Ulm. Fakultät für Ingenieurwissenschaften und Informatik, Diss., 2011
- [Wille 2012] WILLE, Jörn M.: *Manöverübergreifende autonome Fahrzeugführung in innerstädtischen Szenarien am Beispiel des Stadtpilotprojekts*. Braunschweig, TU Braunschweig, Diss., 2012
- [Wille u. a. 2010] WILLE, Jörn M. ; SAUST, Falko ; MAURER, Markus: Stadtpilot: Driving autonomously on Braunschweig's inner ring road. In: *Intelligent Vehicles Symposium (IV), 2010 IEEE*. San Diego, CA, 2010, S. 506–511

Merkmalsbasiertes Map-Matching von radarbasierten Belegungskarten

Matthias Rapp^{*}, Tilmann Giese[†], Markus Hahn[‡], Marc Muntzinger[§],
Jürgen Dickmann[¶] und Klaus Dietmayer^{||}

Zusammenfassung: Das Erstellen von Belegungskarten ist eine etablierte Methode in der mobilen Robotik, um eine Repräsentation der Umgebung mittels Entfernungssensoren zu erhalten. Dabei können mehrere Belegungskarten eines Gebietes entstehen. Eine solche Mehrfachbefahrung ist im automobilen Bereich bei einer gängigen Fahrzeugnutzung der Fall. Um die Informationen der einzelnen Belegungskarten zu vereinen, kann eine Registrierung der Belegungskarten erfolgen. Eine Möglichkeit hierzu stellt *Map-Matching* dar. In dieser Arbeit wird ein merkmalsbasierter Map-Matching Ansatz vorgestellt, um radarbasierte Belegungskarten zu registrieren.

Schlüsselwörter: Map-Matching, Occupancy-Maps, Scan-Matching, Image-Registrierung

1 Einleitung

Fahrerassistenzsysteme werden immer leistungsfähiger und haben einen weitreichenden Funktionsumfang. Dazu benötigen diese Systeme eine vollständige Erfassung des Fahrzeugumfelds. In der mobilen Robotik sind Laserscanner stark verbreitet. Im Fahrzeug findet beispielsweise aufgrund der notwendigen Robustheit das Radar seine Anwendung. Außerdem haben Radare in den letzten Jahren im Bereich der Messgenauigkeit große Fortschritte erzielt. Um mit Entfernungssensoren die Umgebung zu repräsentieren, wird aus den Messdaten eine sogenannte *Belegungskarte* (engl. *Occupancy-Map*) erstellt [1].

Viele Fahrerassistenzsysteme benötigen für eine zufriedenstellende Funktionalität eine sehr gute Repräsentation der Umgebung. Diese Belegungskarte kann erstellt werden, wenn mehrmals das Zielgebiet (aus unterschiedlichen Richtungen) befahren wird. Ein Map-Matching wird durchgeführt um die Informationen, die die Belegungskarten einzelner Fahrten beinhalten in einer Belegungskarte zusammenzufassen. Je nachdem wie eine neue Karte generiert wird, können mobile Objekte eliminiert werden. Durch Mehrfachbefahrungen kann somit auf Veränderungen der Umgebung reagiert werden, beispielsweise, wenn ein Gebäude abgerissen wurde, oder aber neu gebaut wird. Dies ist wünschenswert für die Lokalisierung beispielsweise mit einem Monte-Carlo-Ansatz [1].

^{*}DriveU, Institut für Mess-, Regel- und Mikrotechnik, Universität Ulm (matthias.rapp@driveu.de)

[†]Daimler AG, Ulm (tilmann.giese@daimler.com)

[‡]Daimler AG, Ulm (markus.hahn@daimler.com)

[§]Daimler AG, Ulm (marc.muntzinger@daimler.com)

[¶]Daimler AG, Ulm (juergen.dickmann@daimler.com)

^{||}DriveU, Institut für Mess-, Regel- und Mikrotechnik, Universität Ulm (klaus.dietmayer@uni-ulm.de)

2 Belegungskarten

Belegungskarten sind in der Robotik gängig, um die Umgebung, die mit einem Entfernungssensor erfasst wurde, abzubilden [2]. Dazu wird die Umgebung in *Zellen* diskretisiert. In diesen Zellen werden relativ zum Sensor die Messpunkte eingetragen. Unter Verwendung eines inversen Sensormodells werden neue Messpunkte entsprechend ihrer Genauigkeit in der jeweiligen Zelle akkumuliert. Dadurch entsteht eine Repräsentation der erfassten Umgebung durch Belegungswahrscheinlichkeiten. Diese Belegungskarte ist allerdings durch die Messungenauigkeit leicht „verwaschen“.

Da eine Belegungskarte als ein Graustufenbild interpretiert werden kann, liegt es nahe Methoden aus der Bildverarbeitung zu verwenden. Dabei sind die Intensitäten I eine Abbildung

$$I : \mathbb{Z}^2 \supset G \longrightarrow [0, 1] \\ (x, y) \mapsto I(x, y),$$

welche zu einem Pixel (x, y) , der in der Diskretisierung G der Umgebung für eine Zelle steht, eine Intensität $I(x, y)$ liefert.

3 Merkmals-Extraktion

Eine Möglichkeit ein Map-Matching durchzuführen besteht darin zunächst in beiden Belegungskarten Merkmale zu extrahieren. Ein Merkmal ist hierbei ein *markanter Punkt* mit einer spezifischen Beschreibung. Die Ausrichtung zweier Belegungskarten kann anhand der Assoziationen von zwei oder mehreren Merkmalen in beiden Karten erfolgen.

Um Merkmale extrahieren zu können, müssen zunächst mit einem *Detektor* markante Zellen/Punkte ausgewählt werden. Diese sind beispielsweise Punkte der Belegungskarte, die eine hohe Belegungswahrscheinlichkeit/Intensität aufweisen. Um eine Assoziation zu ermöglichen, sind diese markanten Punkte durch einen *Deskriptor* zu beschreiben. Ein Deskriptor ist dabei eine Funktion, die einem markanten Punkt einen sogenannten *Feature-Vektor* zuordnet. Der Deskriptor beschreibt die nähere Umgebung dieses markanten Punktes und bestimmt Features, die in einem Vektor angeordnet sind. Diese Vorgehensweise ist aus der Bildverarbeitung, beispielsweise bei der Erstellung von Panoramabildern, bekannt, bei der zwei Bildern mit einer gewissen Überlappung miteinander verbunden werden.

3.1 Detektor

Der erste Schritt einer Merkmals-Extraktion besteht darin markante Punkte auszuwählen. Hinter einem Detektor steht ein Verfahren, das in einer Belegungskarte/Bild eine Menge $P \subset G$ an markanten Punkten auswählt.

Um eine Zelle als markanten Punkt zu bezeichnen, liegt es nahe Elemente mit einer hohen Belegungswahrscheinlichkeit zu betrachten. Allerdings erhält man mit dieser Vorgehensweise sehr viele markante Punkte. Außerdem treten diese markanten Punkte in Belegungskarten gehäuft auf.

Ein bekanntes Verfahren zur Extraktion von markanten Punkten ist der sogenannte *Maximally Stable Extremal Region* (MSER) Algorithmus [3]. Dabei werden Bereiche

extrahiert und umrandet, bei denen die Intensitäten innerhalb des Bereichs alle größer oder kleiner den Intensitäten auf dem Rand sind. Über *Thresholding* wird eine MSER bestimmt. Ein weiterer Detektor ist *Features from Accelerated Segment Test* (FAST) [4]. Dieser *Ecken-Detektor* arbeitet auch mit Vergleichen von Intensitäten.

Der Einfluss der Wahl des Detektors auf die Bestimmung der Transformation wird aus den Experimenten in Abschnitt 5 hervorgehen.

3.2 Deskriptor

In der Literatur sind vor allem Deskriptoren verbreitet, welche auch die Skalierung des Bildes berücksichtigen, da sie vor allem für optische Bilder (z.B. Kamerabilder) konzipiert sind. Belegungskarten haben jedoch immer die gleiche Auflösung bzw. Diskretisierung der Umgebung, wodurch eine Skalierungsinvarianz nicht nötig ist und zu negativen Effekten führen kann. Außerdem verändern sich die gängigen Deskriptoren zunächst unter Rotation. Um dieses Problem zu umgehen, wird eine lokale Orientierung des markanten Punktes berechnet und der Deskriptor entlang dieser bestimmt. Auf Radardaten sind diese Orientierungen jedoch recht störanfällig, was zu Fehlern führt. In dem vorliegenden Fall ist allerdings eine Rotationinvarianz sehr wichtig, da eine Mehrfachbefahrung von jeweils unterschiedlichen Richtungen erfolgen kann.

Im Folgenden wird ein Deskriptor vorgestellt, welcher von sich aus rotationsinvariant ist. Ein Vergleich des Deskriptors mit gängigen Deskriptoren aus der Bilderverarbeitung, am Beispiel von *Fast retina keypoint* (FREAK) [5] und *Speeded-up robust features* (SURF) [6], ist in den Experimenten in Abschnitt 5 zu finden.

3.3 Radiale Transformation

In diesem Abschnitt wird der radiale Deskriptor vorgestellt. Dieser basiert auf der Idee, n Ringe um einen markanten Punkt zu betrachten, siehe Abbildung 1. Eine ähnliche Vorgehensweise ist beispielweise bereits in [7] zu finden. Der Deskriptor D ist eine Abbildung, die einem markanten Punkt $p \in P$ einen *Feature-Vektor* $f_p \in \mathbb{R}^{5n}$ zuordnet

$$\begin{aligned} D : P &\longrightarrow \mathbb{R}^{5n} \\ p &\longmapsto f_p. \end{aligned}$$

Zur Deskription von $p \in P$ wird die unmittelbare Umgebung in Form von Ringen betrachtet. Dazu bezeichnen wir mit

$$V_{p,i} := \left(I(x, y) \in \mathbb{R} : r_i < \sqrt{(p_1 - x)^2 + (p_2 - y)^2} < r_{i+1} \right)$$

den i -ten Ring um $p \in P$, $i = 1, \dots, n$. Mit $r_{i+1} - r_i$ wird der Radius der einzelnen Ringe gesteuert.

Als Features werden nun auf diesen Mengen $V_{p,i}$ die folgenden statistischen Größen herangezogen:

- Minimum $\min(V_{p,i})$
- Maximum $\max(V_{p,i})$
- Mittelwert $\overline{V_{p,i}}_{\text{arithm}}$

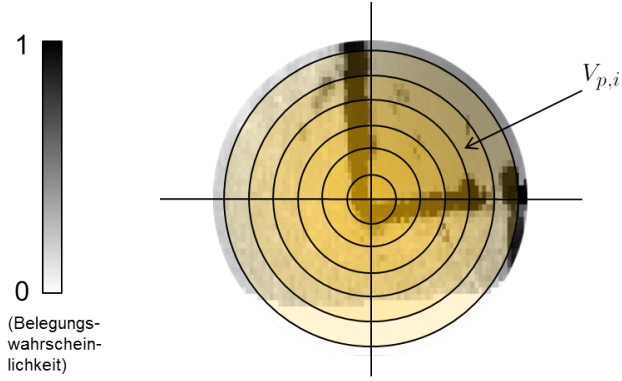


Abbildung 1: Einteilung der Umgebung eines markanten Punktes durch Ringe

- Median $\overline{V_{p,i}}_{med}$
- Standardabweichung $\sigma(V_{p,i})$

Diese Features repräsentieren die nähere Umgebung des markanten Punktes. Der Feature-Vektor $f_p \in \mathbb{R}^{5n}$ fasst diese Features nun zusammen

$$f_p(i \cdot 5 + j) := \begin{cases} \min(V_{p,i}), & j = 1 \\ \max(V_{p,i}), & j = 2 \\ \overline{V_{p,i}}_{arithm}, & j = 3 \\ \overline{V_{p,i}}_{med}, & j = 4 \\ \sigma(V_{p,i}), & j = 5 \end{cases}, \text{ wobei } i = 0, \dots, (n-1) \text{ und } j = 1, \dots, 5.$$

Dieser Deskriptor wird im Folgenden als *ASD* (Annular Statistics Descriptor) bezeichnet.

3.4 Assoziationen

Nachdem in beiden Belegungskarten die Merkmalsmengen P und Q extrahiert wurden, können die Merkmale einander zugeordnet werden. Dazu werden die Feature-Vektoren mit einem geeigneten Distanzmaß miteinander verglichen und die besten als Assoziationen gespeichert. Eine einfache Möglichkeit ist alle Paare miteinander zu vergleichen. Aus den Experimenten wurde ersichtlich, dass dieser Ansatz eine für die Anwendung akzeptable Laufzeit aufweist.

Ferner wurde zusätzlich noch ein *Cross-Check* verwendet. Dieser überprüft die Zuordnung auch in umgekehrter Richtung. Die Menge der Assoziationen \mathcal{A} wird also definiert als

$$\mathcal{A} := \left\{ (p, q) \in P \times Q : p = \arg \min_{p' \in P} \|f_{p'} - f_q\| \text{ und } q = \arg \min_{q' \in Q} \|f_p - f_{q'}\| \right\}.$$

Zum Finden von Assoziationen bei FREAK wurde die *Hamming*-Distanz zum Vergleich der Feature-Vektoren verwendet. Bei ASD hingegen wurde die $\|\cdot\|_1$ -Norm, bei SURF die $\|\cdot\|_2$ -Norm verwendet.

4 Modifizierter RANSAC

Mit den Assoziationen kann mit einem geeigneten Verfahren die Transformation (Rotation und Translation) bestimmt werden.

In [8] wird ein Verfahren beschrieben, wie dies mittels einer *Singular value decomposition* (SVD) möglich ist. Das Verfahren ist nur für eine Menge an Assoziationen geeignet, die keine Ausreißer beinhaltet, da diese ansonsten das Ergebnis stark verfälschen. Da in der Praxis viele Fehlassoziationen vorkommen, liegt für diese Problemstellung ein Algorithmus auf Basis von *Random sample consensus* (RANSAC) nahe, um eine Menge an „guten“ Assoziationen zu bestimmen. Dies ist eine gängige Vorgehensweise, um mit fehlerbehafteten Eingabedaten umzugehen.

Algorithmus 1 : Modifizierter RANSAC-Algorithmus

```

Data : Assoziationen  $\mathcal{A}$ 
Result : Transformation  $T$ 
 $i = 0$ ;
Inliers =  $\emptyset$ ;
BestInliers =  $\emptyset$ ;
while  $i < i_{max}$  do
  Inliers =  $\emptyset$ ;
  Wähle 3 zufällige Assoziationen  $A \subset \mathcal{A}$ ;
  if Assoziationen spannen annähernd deckungsgleiche Dreiecke auf then
    Schätze Transformation  $T$  mittels Assoziationen  $A$ ;
    Inliers = Inliers  $\cup$   $A$ ;
    foreach  $a \in \mathcal{A} - A$  do
      if  $a$  bestätigt  $T$  then
        Inliers = Inliers  $\cup$   $a$ ;
    if Inliers.laenge > BestInliers.laenge then
      BestInliers = Inliers;
     $i = i + 1$ ;
Schätze  $T$  mit Inliers;

```

RANSAC ist ein Verfahren, das eine Transformation trotz Ausreißer bestimmen kann. Dazu wird zunächst eine Transformation mittels SVD aufgrund einer kleinen zufälligen Teilmenge von \mathcal{A} bestimmt (hier benötigt man mindestens drei Assoziationen) und überprüft wieviele Zuordnungen in \mathcal{A} diese Transformation unterstützen. Die Assoziationen, die dies tun, werden als *Inliers* bezeichnet. Diese Vorgehensweise wird einige Male wiederholt. Speichern dieser Elemente, die die Transformation unterstützen, liefert eine Menge an *Inliers*. Es wird die Menge an *Inliers* ausgewählt, welche die meisten Elemente hat. Zum

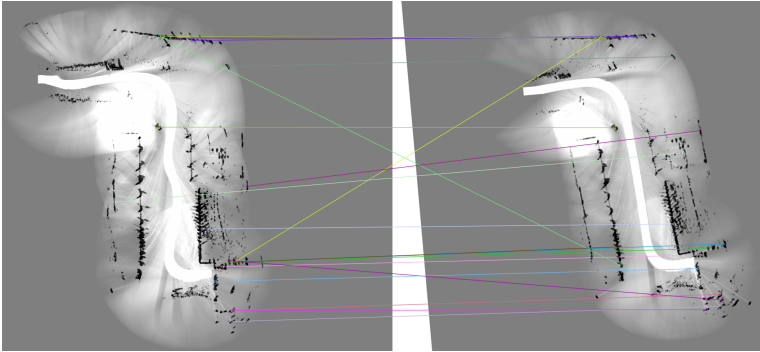


Abbildung 2: Assoziationen der Detektor-Deskriptor Kombination MSER-ASD auf den Referenzbelegungskarten (rotiert um 10°). Eingetragen sind die 20 besten Assoziationen zur besseren Darstellung.

Schluss kann man mit dieser Menge an Inliers eine Transformation berechnen, welche diese am besten bestätigen.

Da bei den Belegungskarten nur die Translation und Rotation berechnet werden muss und die Skalierung der Karten gleich ist, wird eine zusätzliche Bedingung eingefügt. Es müssen drei markante Punkte in einem Bild auf markante Punkte abgebildet werden, welche (annähernd) deckungsgleiche Dreiecke aufspannen. Die anfängliche Teilmenge wird verworfen, wenn sie eben diese Bedingung nicht in einem gewissen Toleranzbereich erfüllt. Der modifizierte RANSAC ist in Algorithmus 1 aufgeführt.

5 Experimente

Zur Bewertung der Deskriptoren werden zwei Belegungskarten der selben Umgebung, aufgenommen zu unterschiedlichen Zeiten, miteinander verglichen (siehe Abbildung 2). Der Deskriptor ASD aus Abschnitt 3.3 wurde wie folgt konfiguriert. Es wurde ein kreisförmiger Bildausschnitt mit einem Radius von 50 Pixeln um einen markanten Punkt zur Beschreibung betrachtet und dieser Kreisausschnitt in $n = 10$ Ringe aufgeteilt. Die Dicke der Ringe wurde konstant gewählt.

Die Rotation einer Belegungskarte in 1° -Schritten ermöglicht eine Untersuchung der Genauigkeit beim Schätzen des Winkels zu der zweiten Belegungskarte. Zur Rotation der Belegungskarte wurde eine affine Transformation mit einer kubischen Interpolation benutzt. Das anfängliche Ausrichten beider Belegungskarten durch *Labeling*, stellt die Referenz dar. Zur Schätzung des Winkels wurde der modifizierte RANSAC aus Abschnitt 4 nach 20 Iterationen beendet, da weitere Iterationen durch die Vorverarbeitung mit dem Dreiecksabgleich keine zusätzliche Verbesserung ergaben. Dabei wurde eine Fehlertoleranz von drei Pixel verwendet. Mehrmaliges Durchführen der oben erwähnten Transformations-schätzung kompensiert probabilistische Effekte.

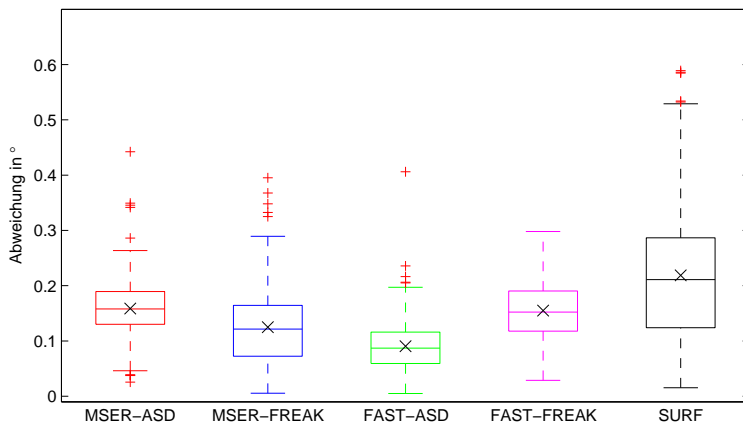


Abbildung 3: Boxplots der mittleren Abweichungen vom Rotationswinkel. Mit „×“ ist der Mittelwert eingetragen.

Ein Boxplot der durchschnittlichen Winkelabweichung ist in Abbildung 3 aufgeführt. Es wurden vier Kombinationen miteinander verglichen, je eine Kombination aus zwei Detektoren, MSER und FAST, und zwei Deskriptoren, ASD und FREAK. Zusätzlich wurde SURF zum Vergleich herangezogen, der selber auch die Auswahl der markanten Punkte vornimmt. Dabei wurde ein Hessian Threshold von 500 verwendet. Die Implementierungen der Algorithmen aus OpenCV in Version 2.4.3 stellten die Verfahren MSER, FAST, FREAK und SURF bereit. Wie aus den Abweichungen in Abbildung 3 hervor geht, war diese anfängliche Ausrichtung sehr gut, da bei den mittleren Abweichungen der geschätzten Winkel von den tatsächlichen kein erheblicher Offset zu beobachten ist.

Es ist ersichtlich, dass alle Kombinationen mit Abweichungen von unter 0.6° sehr gute Ergebnisse liefern. Obwohl SURF das verbreitetste Verfahren in der Bildverarbeitung ist, scheidet es bei diesem Vergleich am schlechtesten ab. Die mittlere Abweichung der Winkelschätzung ist in Tabelle 1 aufgeführt. Daraus lässt sich ablesen, dass der Deskriptor ASD sich trotz seines anschaulichen Ansatzes durchaus mit einem etablierten Verfahren messen kann. Es ist zu beobachten, dass die Güte der Winkelschätzung sehr von der Wahl des Detektors beeinflusst wird. Um nun die Güte der Assoziationen zu bestimmen, wird der Quotient der Anzahl der Inliers zur Anzahl aller Assoziationen berechnet. Dieser Quotient repräsentiert welcher Anteil aller Assoziationen mit einer gewissen Toleranz die berechnete Transformation unterstützen. Der Anteil der Inliers der Assoziationen ist in Tabelle 1 für die einzelnen Detektor-Deskriptor Kombinationen aufgeführt. In Abbildung 4 ist der Anteil der Inliers über den Rotationswinkel aufgetragen. Interessant ist dabei, dass eine gewisse Winkelabhängigkeit bei SURF zu beobachten ist. Daher ist anzunehmen, dass SURF auf diesen Belegungskarten nicht gänzlich rotationsinvariant arbeitet. Die anderen Verfahren weisen ein konstantes Verhalten auf. Dabei ist zu sehen, dass die Kombination von FAST und FREAK den größten Anteil richtiger Assoziationen hatte. Die Kombination

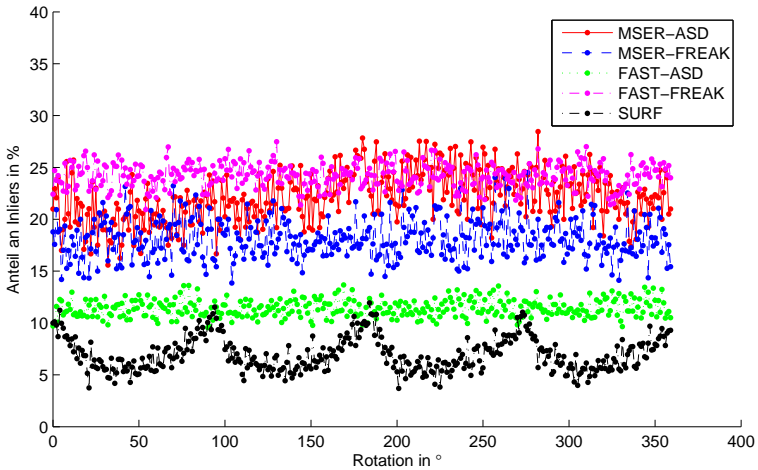


Abbildung 4: Quotient der Inliers und Assoziationen aufgetragen in 1° Schritten.

Tabelle 1: Mittlerer Fehler und Anteil an Inliers der einzelnen Kombinationen

	MSER-ASD	MSER-FREAK	FAST-ASD	FAST-FREAK	SURF
Mittl. Fehler	0.16°	0.13°	0.09°	0.16°	0.22°
Mittl. Anteil an Inliers	22.1%	18.1%	11.4%	24.2%	7.0%

von MSER und ASD erreicht mit einer kleineren Feature-Vektorlänge nahezu die gleiche Performance.

Der Vorteil des modifizierten RANSAC aus Abschnitt 4 ist in Abbildung 5 zu sehen. Dabei wurde der RANSAC ohne den Dreiecksabgleich mit der Kombination MSER-ASD verwendet. Aufgeführt ist die maximale Abweichung bei allen Winkelschätzungen pro Rotationswinkel. Die maximale Abweichung der Winkelschätzung mit dem modifizierten RANSAC betrug 0.49°. Ohne diese Modifikation hingegen beträgt die maximale Abweichung 0.65°. Es ist zu beobachten, dass ohne Dreiecksabgleich mehr Ausreißer nach oben, in Form von Abweichungen über 0.4° auftreten. Zur Vergleichbarkeit wurde der modifizierte RANSAC mit 20 Iterationen ausgeführt, der nicht modifizierte RANSAC hingegen mit 500 Iterationen. Die Anzahl der Iterationen wurde so gewählt, dass beide RANSAC-Verfahren eine in etwa gleiche Laufzeit aufweisen. Daraus ergibt sich, dass bei möglicher Echtzeitanforderung diese Modifikation einen Vorteil bezüglich der Robustheit liefert.

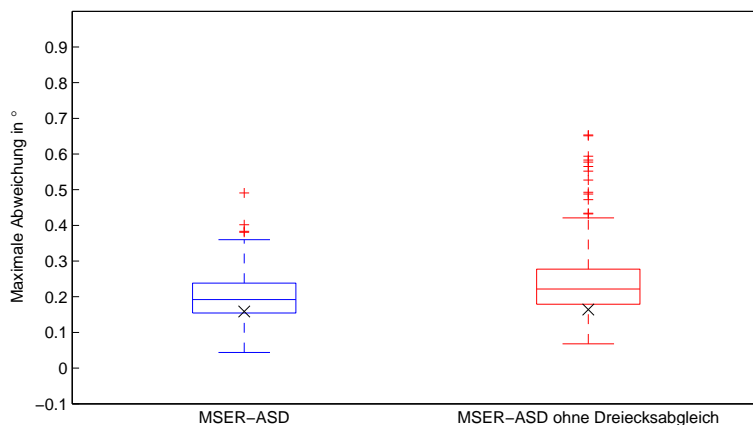


Abbildung 5: Vergleich des modifizierten RANSAC mit RANSAC ohne Dreiecksabgleich. Gezeigt ist ein Boxplot der maximalen Abweichungen.

6 Zusammenfassung

In dieser Arbeit wurde ein merkmalsbasiertes Map-Matching Verfahren vorgestellt. Dieses erweist sich als geeignet für radarbasierten Belegungskarten. Dazu werden zunächst markante Punkte mit einem etablierten Algorithmus aus der Bildverarbeitung extrahiert. Die Arbeit stellt einen neuen Ansatz vor, diese extrahierten Punkte zu beschreiben. Durch einen modifizierten RANSAC-Algorithmus werden dann die Belegungskarten aufeinander registriert. In Experimenten konnte gezeigt werden, dass trotz des speziellen Ansatzes die Performance verbreiteter Deskriptoren (am Beispiel von FREAK und SURF) bei dem vorliegenden Fall von Belegungskarten erreicht wurde. Durch dieses Registrieren der Belegungskarten kann durch Mehrfachbefahrung die Umgebung sehr gut repräsentiert werden und sogar auf Veränderungen der Umgebung reagiert werden.

Literatur

- [1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, 2005.
- [2] Hans Moravec and Alberto Elfes, “High resolution maps from wide angle sonar,” in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. IEEE, 1985, vol. 2, pp. 116–121.
- [3] Jiri Matas, Ondrej Chum, Martin Urban, and Tomáš Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

- [4] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *Computer Vision-ECCV 2006*, pp. 430–443. Springer, 2006.
- [5] Alexandre Alahi, Raphael Ortiz, and Pierre Vanderghelynst, "Freak: Fast retina key-point," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 510–517.
- [6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [7] A. Chalechale, A. Mertins, and G. Naghdy, "Edge image description using angular radial partitioning," *Vision, Image and Signal Processing, IEE Proceedings -*, vol. 151, no. 2, pp. 93–101, 2004.
- [8] K Somani Arun, Thomas S Huang, and Steven D Blostein, "Least-squares fitting of two 3-D point sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 5, pp. 698–700, 1987.

Hocheffizientes Stereosehen durch Modellierung geneigter Ebenen und Parallelverarbeitung

Benjamin Ranft* und Tobias Strauß†

Zusammenfassung: Stereokameras sind leistungsfähige Sensoren für Fahrerassistenzsysteme. Moderne Algorithmen zur Disparitätsschätzung erzielen eine hohe Ergebnisqualität, benötigen jedoch lange Verarbeitungszeiten. Dagegen sind fensterbasierte lokale Verfahren sehr effizient implementierbar, aber fehleranfälliger. Letzteres gilt besonders bei örtlichen Änderungen der Disparität, etwa auf zur Bildebene geneigten Flächen. Im Folgenden präsentieren wir ein Verfahren, welches diese Änderungen für beliebig im Raum orientierte Ebenenscharen kompensieren kann. Hierfür ist keine Modifikation der eigentlichen Disparitätsschätzung, sondern nur eine Transformation je eines Eingangsbildes sowie der Zwischenergebnisse erforderlich. Bei Kombination mit einer Stereo-Implementierung, welche aktuelle Mehrkern- und Grafikprozessoren effizient nutzt, konnte ein echtzeitfähiger Durchsatz erhalten, die Qualität aber deutlich gesteigert werden.

Schlüsselwörter: Echtzeit, Parallelverarbeitung, Stereosehen, Umfelderkennung

1 Einleitung

Unsere in [1] beschriebene Disparitätsschätzung erreicht mit Standardprozessoren in Zwischen Latenzzeiten unter 12 ms auf 0.5 Mpx großen Bildern. Allerdings trifft sie – wie die meisten fensterbasierten Verfahren – die Modellannahme rechtwinklig zur optischen Achse ausgerichteter Ebenen. Im Einsatzbereich von Fahrerassistenzsystemen ist diese Annahme jedoch nicht nur auf der Fahrbahn, sondern auch für Objekte am Straßenrand häufig unzutreffend. Dies kann zu ungenauen, lücken- und fehlerhaften Ergebnissen führen, wie Abb. 1 exemplarisch zeigt. Das hier vorgestellte Verfahren ermöglicht eine deutliche Reduktion dieses Effekts und liefert als Nebenprodukt eine Klassifikation der Fahrbahn. Es ist nicht auf die Spezialfälle waag- oder senkrechter Flächen bzgl. der Kamera beschränkt und eignet sich daher etwa auch für die signifikanten Rollwinkel von Einspurfahrzeugen.

Der hierfür entwickelte Ansatz bildet mit Kap. 4 den Schwerpunkt dieses Beitrags. Zuvor stellen wir zur Verdeutlichung der Neuerungen in Kap. 2 verwandte Arbeiten vor und fassen in Kap. 3 unsere aktuellste Basis-Implementierung für maschinelles Stereosehen zusammen. Letztere beinhaltet u. a. eine Variante der Census-Transformation, welche als Vorverarbeitungsschritt die Robustheit des Verfahrens erhöht und sehr effizient berechnen und vergleichbar ist. Kap. 5 präsentiert die Ergebnisse einer quantitativen Auswertung der Qualität und Geschwindigkeit des vorgestellten Verfahrens, während Kap. 6 eine Zusammenfassung dieser Veröffentlichung sowie einen Ausblick auf zukünftige Arbeiten bietet.

*Benjamin Ranft ist Wissenschaftlicher Mitarbeiter am FZI Forschungszentrum Informatik, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe (e-mail: ranft@fzi.de).

†Tobias Strauß ist Wissenschaftlicher Mitarbeiter am Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie (KIT), Engler-Bunte-Ring 21, 76131 Karlsruhe (e-mail: strauss@kit.edu).

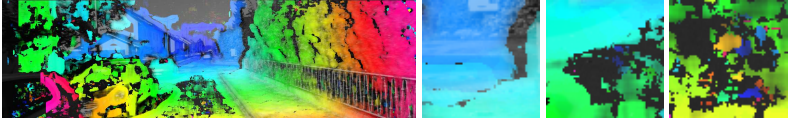


Abbildung 1: Durch ein typisches lokales Schätzverfahren gewonnene Disparitätsschätzungen, hervorgehobene ungenaue (links), lückenhafte (mitte) und fehlerhafte Ergebnisse (rechts)

2 Verwandte Arbeiten

Zur Steigerung der Qualität einer Disparitätsschätzung nutzt etwa [2] die Census-Transformation, welche jedes Pixel durch eine Bitfolge beschreibt: Je ein Bit codiert, ob es heller als ein bestimmtes Nachbarpixel ist. Mit zwei Bits lässt sich ergänzend ein Vergleich mit dem Mittelwert aller betrachteten Nachbarn [3] oder – zwecks Robustheit gegen Bildrauschen – eine Einteilung in deutlich dunklere, ähnlich helle und deutlich hellere Nachbarn [4] darstellen. Die Unähnlichkeit zweier Bildpunkte kann durch deren *Hamming*-Distanz quantifiziert werden, d. h. der Anzahl verschiedener Bits in den Bitfolgen beider Pixel.

Vielfältige Ansätze widmen sich geneigt betrachteten Ebenen, v. a. jedoch dem Spezialfall einer Grundebene bei vernachlässigbarem Rollwinkel: Nur falls diese durch eine statische horizontale Scherung des rechten Kamerabildes nicht auf das linke abgebildet wird, führt [5] eine aufwändigere Disparitätsschätzung durch. [6] wendet ein lokales Schätzverfahren jeweils auf originale und gescherte Bilder an. Eine Fusion beider Disparitätskarten erzielt verbesserte Ergebnisse auf der Fahrbahn und eine Gesamtlaufzeit von 430 ms. Analog führt auch unser Verfahren Disparitätsschätzungen auf transformierten Eingangsbildern durch, kann allerdings eine Obermenge an Ebenen behandeln und Hypothesen anhand der aktuellen Szene auswählen. Eine dynamische Anpassung der Scherung ermöglicht es [7], ohne die explizite Berechnung von Disparitäten, Hindernisse zu detektieren. Im Unterschied zu den bisher Genannten behandelt [8] senkrechte Flächen wie etwa Wände, indem einzelne Zeilensegmente nicht nur unter mehreren Disparitäten, sondern auch nach unterschiedlichen horizontalen Streckungen oder Stauchungen verglichen werden.

Für das nicht nur im Bereich der Fahrerassistenz bekannte Semi-Global Matching beschreibt [9] eine Erweiterung zur Behandlung zuvor klassifizierter waag- bzw. senkrechter Flächen: Von deren Disparitätsgradienten verursachte Kosten werden bei den betroffenen Aggregationspfaden geringer gewichtet, was die Ergebnisqualität, aber auch die Laufzeit steigert. Das in [10] vorgestellte Verfahren könnte beliebige Ebenen ideal abbilden, nimmt in der Praxis jedoch eine sog. “manhattan world” an: Als Hypothesen werden drei Stapel von Ebenen mit orthogonalen Normalenvektoren untersucht. Bei einer Auflösung von 512×384 px ist es auf einem leistungsstarken Grafikprozessor echtzeitfähig. Dagegen ist eine dynamische Berücksichtigung vieler beliebig orientierter Ebenen komplexeren Verfahren vorbehalten, etwa auf Basis von Superpixeln und Markow-Netzwerken. Deren Effizienz konnte [11] deutlich steigern, benötigt aber dennoch wenige Minuten pro Bildpaar.

Durch Ergänzung einer Disparitätsschätzung mit einer separaten [12] oder integrierten [13] Bestimmung von Optischem Fluss können nicht nur 3D-Punkte, sondern auch deren Geschwindigkeitsvektoren rekonstruiert werden – eine wertvolle Information für Fahrerassistenzsysteme. Die im Folgenden vorgestellte Implementierung ist grundsätzlich auch zur Flusschätzung geeignet, jedoch wurde dieser Einsatzzweck nicht weiter verfolgt.

3 Basis-Implementierung

Als Grundlage für die detaillierte Beschreibung der Disparitätsschätzung auf geeigneten Ebenen möchten wir dafür relevante Eigenschaften und noch unveröffentlichte Weiterentwicklungen unseres Basis-Verfahrens zusammenfassen. In [1] präsentierten wir für Mehrkern-, Embedded- und Grafikprozessoren ausgelegte Implementierungen. Darauf aufbauend zeigt [14] eine automatische Zusammenarbeit von CPUs und GPUs. Zentrum des Verfahrens ist ein klassisches Block-Matching, welches die Summe der absoluten Differenzen (SAD) zweier Fenster, unabhängig von deren Größe, mit je zwei Additionen und Subtraktionen berechnet. Zusätzlich können durch Nutzung der Vektor-Instruktionen moderner CPUs inzwischen pro Kern die SADs für 16 Disparitäten gleichzeitig bestimmt werden¹.

Abb. 2 zeigt die gesamte Verarbeitungskette mit dem Block-Matching vor- und nachgelagerten Schritten. Eine Census-Transformation ersetzt dabei den zuvor genutzten Hochpassfilter. Grundsätzlich ist eine solche Vorverarbeitung hilfreich, indem sie Beleuchtungsunterschiede zwischen linkem und rechtem Grauwertbild kompensiert und die Ähnlichkeit von deren Pixeln für den Matching-Schritt schnell und robust messbar macht. Gegenüber dem vorherigen Algorithmus kann die Census-Transformation nicht nur additive, sondern auch multiplikative Unterschiede ausgleichen. Unsere Variante kombiniert je zwei Bits für insgesamt acht Nachbarn in einer Entfernung von $s = 4$ Pixeln zu einer Folge von 16 Bits:

$$C(u, v) = \bigotimes_{i,j=-1}^{i,j=1} \begin{cases} 00 & \text{für } I(u+si, v+sj) \leq I(u, v) - t \\ 01 & \text{für } I(u, v) - t < I(u+si, v+sj) \leq I(u, v) + t \\ 11 & \text{für } I(u, v) + t < I(u+si, v+sj) \end{cases} \quad (1)$$

Der Helligkeitsunterschied $t = 2$ spannt ein Intervall von Grauwerten auf, die dem Mittelpunkt ähnlich sind, um den Einfluss von Sensorrauschen zu reduzieren. Die Codierung ist so gewählt, dass die *Hamming*-Distanz zwischen dieser Kategorie und einem deutlich dunkleren oder helleren Nachbarn 1, die der letzteren untereinander jedoch 2 beträgt. Abb. 3 zeigt eine farbliche Visualisierung dieser Census-Transformation, da die Bitfolgen selbst bildlich nicht sinnvoll darstellbar sind. Unsere Implementierung dieses Vorverarbeitungsschritts benötigt ca. 70% weniger Rechenzeit als der bisherige Hochpassfilter.

Die Census-Bitfolge eines Pixels wird nur einmal berechnet, aber beim Matching für jede Disparität mit der eines anderen verglichen. Damit kommt einer effizienten Bestimmung der *Hamming*-Distanz von je 16 Bits eine essentielle Bedeutung zu. Hierbei wird zuerst per exklusives Oder eine Folge von 0 für identische und 1 für verschiedene Bits generiert; letztere sind nun zu zählen. Einige Prozessoren bieten Instruktionen zur direkten

¹Kompatibilität zu früheren Prozessoren wird hierbei durch die *C++ vector class library* gewährleistet, welche als Open-Source-Software unter <http://agner.org/optimize/#vectorclass> erhältlich ist.

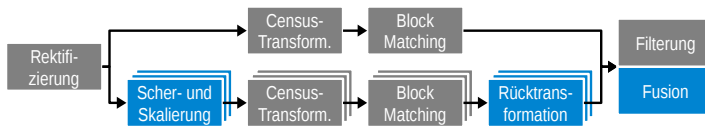


Abbildung 2: Schätzung von klassischer Disparität (oberer Pfad) und mehreren Pseudo-Disparitäten auf geeigneten Ebenenscharen (untere Pfade): ebenenspezifische (blau) und übrige Verarbeitungsschritte (grau)

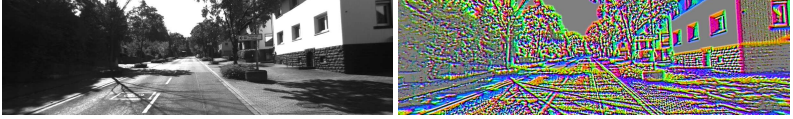


Abbildung 3: Visualisierung unserer Variante der Census-Transformation und originales Grauwertbild: Der Blau-/Grün-/Rotkanal codiert ein relativ zu oben/unten-links/unten-rechts gelegenen Nachbarpixeln gleichhelles oder (bei voller Sättigung) helleres Zentrum.

Berechnung dieses sog. “population count”. Alternativ erreicht eine Lookup-Tabelle mit $2^{16} = 65536$ Einträgen beim Vergleich einzelner Pixel eine ähnliche Geschwindigkeit. Da unsere Implementierung jedoch wie oben beschrieben 16 Disparitäten bzw. Pixel gleichzeitig untersucht, sind parallel ebenso viele *Hamming*-Distanzen zu bestimmen. Abb. 4 zeigt einen Algorithmus, der sich zu diesem Zweck auf tatsächlich verfügbare Vektor-Instruktionen beschränkt und damit den 6-fachen Durchsatz der vorigen Varianten erzielt. Er basiert auf dem Ansatz von [2], welcher hier die auf 1 gesetzten Bits in Gruppen von 2, 4, 8 und schließlich 16 Bits bestimmt. Zur Geschwindigkeitssteigerung ersetzen wir die ersten beiden Stufen durch eine Lookup-Tabelle innerhalb eines Prozessorregisters, welche direkt die Anzahl gesetzter Bits jeder 4er-Gruppe liefert.

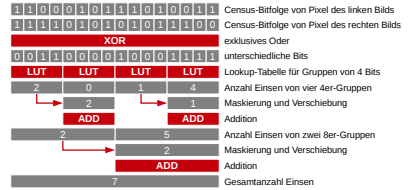


Abbildung 4: Exemplarische Berechnung einer *Hamming*-Distanz, parallel auf 15 weitere Pixel-Paare anwendbar (nicht dargestellt)

4 Disparitätsschätzung auf geneigten Ebenen

Wie eingangs angesprochen wird die Disparitätsschätzung auf allen nicht fronto-parallelen Flächen dadurch erschwert, dass deren Entfernung und Disparität über das beim Block-Matching verwendete Fenster hinweg nicht konstant ist. Allerdings wird jede ebene Fläche in Weltkoordinaten X , Y und Z auch auf eine Ebene im Parameterraum von Bildspalte u , -zeile v und Disparität d abgebildet; entsprechend sind ihre Gradienten dd/du und dd/dv konstant. Somit kann eine geeignete horizontale Scherung und Streckung bzw. Stauchung

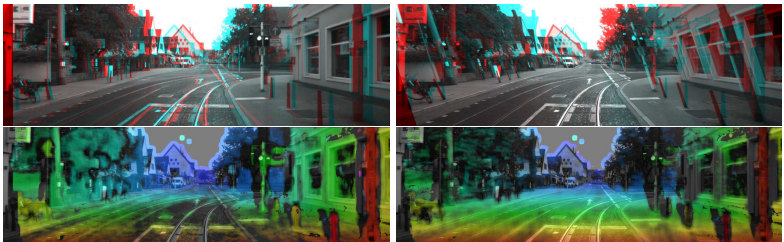


Abbildung 5: überlagerte Eingangsbilder und Disparitätskarten für klassisches Stereo (links) und für durch Scherung des rechten Bildes modellierte horizontale Ebenen (rechts)

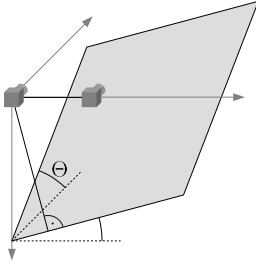


Abbildung 6: Stereokamerasystem mit Basisbreite B , unter den Winkeln Φ und Θ geneigt betrachtete Ebene im Abstand D

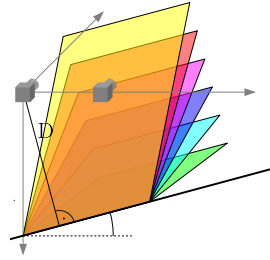


Abbildung 7: bei konstanter Scherung und Skalierung für verschiedene d_t untersuchte Ebenen mit gemeinsamer Schnittgerade

eines oder beider Stereobilder die gewünschte konstante Pseudo-Disparität d_t zwischen diesen herstellen. Als Vorgriff veranschaulicht Abb. 5, wie nach einer strikten Filterung anhand der Summe von *Hamming*-Distanzen, d. h. der Ähnlichkeit der Matching-Fenster, je nach Transformation überwiegend Disparitäten auf fronto-parallelen bzw. waagrechten Flächen verbleiben. Da das linke Bild klassischerweise als Referenz dient, wenden wir eine solche Transformation ausschließlich auf das rechte an: Dessen Inhalt wird abhängig vom Schergradienten g und Skalierungsfaktor s horizontal von Spalte u_r nach u_t verschoben. Vertikal findet keine Verschiebung statt, so dass mit der Lage korrespondierender Punkte auf der gleichen Bildzeile eine zentrale Eigenschaft rektifizierter Bilder erhalten bleibt.

$$u_t = s(u_r + gv) \implies d_t = u_t - u_r = sd + (1 - s)u_r - gv \tag{2}$$

Wir verwenden für den Versatz d_t zwischen originalem linken und transformiertem rechten Bild den Begriff Pseudo-Disparität, da diese zwar mit dem gleichen Verfahren geschätzt wird, geometrisch jedoch anders zu interpretieren ist als eine klassische Disparität. Die folgende Herleitung von g und s beschreibt u. a. diese Interpretation:

Abb. 6 zeigt eine Stereokamera mit Basisbreite B , welche unter dem Rollwinkel Φ und dem Winkel Θ zur optischen Achse eine Ebene betrachtet. Der Abstand D zwischen dieser und dem optischen Zentrum der linken Kamera wird auf der XY - bzw. Bildebene angegeben. Für jeden Punkt (X, Y, Z) auf der Ebene gilt:

$$D = X \sin(\Phi) + Y \cos(\Phi) + Z \tan(\Theta) \tag{3}$$

Einsetzen von $X = Z(u_t - u_0)/f$, $Y = Z(v - v_0)/f$ sowie $Z = Bf/d$ ergibt mit Brennweite f und Hauptpunkt (u_0, v_0) der Kamera einen Zusammenhang zur Disparität d :

$$d = \frac{B}{D}((u_t - u_0) \sin(\Phi) + (v - v_0) \cos(\Phi) + f \tan(\Theta)) \tag{4}$$

Damit kann aus den partiellen Ableitungen von Gl. 2 nach Bildzeile und -spalte dasjenige Tupel von Schergradient g und Skalierungsfaktor s bestimmt werden, welches die Ebene unter der gewünschten konstanten Pseudo-Disparität abbildet:

$$\frac{\partial d_t}{\partial u_t} = s \left(\frac{B}{D} \cos(\Phi) - g \right) \stackrel{!}{=} 0 \implies g = \frac{B}{D} \cos(\Phi) \tag{5}$$

$$\frac{\partial d_t}{\partial v} = 1 + s \left(\frac{B}{D} \sin(\Phi) - 1 \right) \stackrel{!}{=} 0 \implies s = \frac{D}{D - B \sin(\Phi)} \quad (6)$$

Für die Scherung und Streckung unerheblich ist dagegen der Nick- bzw. Gierwinkel² Θ zwischen Ebene und optischer Achse. Stattdessen bildet ein Intervall von Pseudo-Disparitäten d_t den in Abb. 7 skizzierten ‐Fächer‐ aus einer Schar von unter verschiedenen Winkeln Θ betrachteten Ebenenhypothesen, deren gemeinsame Schnittgerade in der XY -Ebene liegt:

$$d_t = \left(f \tan(\Theta) - u_0 \sin(\Phi) - v_0 \cos(\Phi) \right) B / (D - B \sin(\Phi)) \quad (7)$$

Wie die Verarbeitungskette in Abb. 2 bereits andeutete, können für jedes Stereo-Bildpaar mehrere Pseudo-Disparitätsschätzungen für unterschiedlich orientierte Ebenenscharen durchgeführt werden. Zur Festlegung von deren Anzahl und jeweiliger Parametrierung (g, s) schlagen wir zwei Vorgehensweisen vor:

1. Geometrisch: Im Bereich der Fahrerassistenz sind häufig Informationen verfügbar, welche auch ohne genaue Kenntnis der Umgebung geeignete Parametrierungen ermöglichen. Zur Behandlung der Fahrbahn können etwa deren Abstand D und Rollwinkel Φ zur Stereokamera aus Messungen des aktuellen Federwegs geschätzt werden. Die Fahrzeugposition in einer Karte wäre ein Indiz, ob seitlich mit Gebäudefassaden oder Fahrzeugen auf benachbarten Fahrspuren zu rechnen und somit eine hierfür angepasste Schätzung von Pseudo-Disparitäten überhaupt erforderlich ist.
2. Automatisch: Ohne ergänzende Informationen können an die aktuelle Szene angepasste Tupel (g, s) auch aus einem bereits vorhandenen Disparitätsbild gewonnen werden – etwa dem vorherigen einer Sequenz oder einer rein fronto-parallel Schätzung aus dem aktuellen Bildpaar. Entsprechende Tupel treten als lokale Maxima eines 2D-Histogramms von dessen Gradienten dd/du und dd/dv auf. Um in Nachkommastellen genaue Parameter zu erhalten, müssen ganzzahlig quantisierte Disparitätsbilder herunterskaliert und die Gradienten über mehrere Pixel hinweg gemessen werden. Ersteres trägt auch zu der mit 0,1 ms geringen Rechenzeit dieser Methode bei, welche damit deutlich unter der eines Ebenenfits mittels RANSAC liegt. Abb. 8 zeigt Exemplare solcher Histogramme und zugehörige Pseudo-Disparitätskarten.

²Die treffendere Bezeichnung hängt vom Rollwinkel Φ zwischen Ebene und Stereokamera ab.

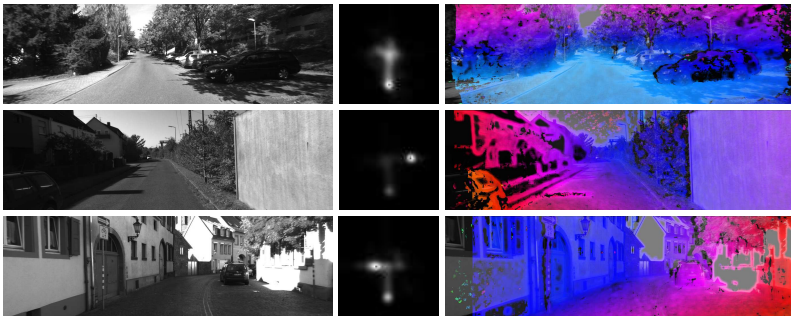


Abbildung 8: Gradientenhistogramme einer klassischen Disparitätsschätzung, Ergebnisse der Pseudo-Disparitätsschätzung für das im Histogramm jeweils markierte Maximum

Nach der Scherung, Streckung und Census-Transformation des rechten Kamerabildes können die zugehörigen Pseudo-Disparitäten, wie eingangs beschrieben, durch einen beliebigen Algorithmus geschätzt werden. Hierbei ist zu berücksichtigen, dass bildweit konstante Grenzen der Disparität $[d_{min}, d_{max}]$ gemäß Gl. 2 auf ein bildkoordinaten-abhängiges Intervall von d_t abgebildet werden. Unser Schätzverfahren [1] gewinnt seine Geschwindigkeit jedoch u. a. dadurch, dass je ein Intervall auf eine rechteckige Bildpartition angewendet wird; dieses muss entsprechend die Vereinigungsmenge der Pseudo-Disparitäts-Intervalle aller enthaltenen Pixel umfassen. Daher dauert die Schätzung von Pseudo-Disparitäten auf geneigten Ebenen etwa 10% länger als diejenige klassischer Disparitäten auf fronto-parallelen Flächen. Unsere GPU-Implementierung bearbeitet unabhängig 64×32 px große Kacheln, aus deren Lage im linken Bild $[u_{l,min}, u_{l,max}] \times [v_{min}, v_{max}]$ die minimale – und analog die maximale – dort zu untersuchende Pseudo-Disparität leicht zu berechnen ist:

$$d_{t,min} = s d_{min} + (1-s) \begin{cases} u_{l,min} & \text{für } s < 1 \\ u_{l,max} & \text{für } s \geq 1 \end{cases} - s g \begin{cases} v_{min} & \text{für } g < 0 \\ v_{max} & \text{für } g \geq 0 \end{cases} \quad (8)$$

Im Gegensatz dazu weist unsere Implementierung für Mehrkern-CPUs jedem Kern einen horizontalen Streifen des zu schätzenden Disparitätenbildes zu, welcher sich über dessen gesamte Breite erstreckt. Um unnötige Berechnungen zu vermeiden, beginnt allerdings für die jeweils nächsten 16 zu untersuchenden Pseudo-Disparitäten $[d_{t,min}, d_{t,max}]$ der Bereich relevanter Spalten im linken Bild erst bei $u_{l,min}$ und endet analog bereits bei $u_{l,max}$:

$$u_{l,min} = \frac{1}{1-s} \begin{pmatrix} d_{t,min} - s d_{max} & \text{für } s < 1 \\ d_{t,max} - s d_{min} & \text{für } s \geq 1 \end{pmatrix} - s g \begin{cases} v_{min} & \text{für } (1-s)g < 0 \\ v_{max} & \text{für } (1-s)g \geq 0 \end{cases} \quad (9)$$

Im Anschluss an jede (Pseudo-)Disparitätsschätzung führen wir die bereits in [1] und [14] beschriebene Filterung der Ergebnisse durch: Für jedes Pixel wird sowohl die niedrigste gefundene Summe von *Hamming*-Distanzen als auch die Konsistenz der Disparitäten bzgl. linkem und rechtem Bild ausgewertet und in ein Gütemaß $\in [0, 256]$ überführt. Hierfür müssen vorab die Ergebnisse bzgl. jedes gescherten und gestreckten Eingangsbilds rücktransformiert und Pseudo-Disparitäten in klassische Disparitäten umgerechnet werden:

$$u_r = \frac{u_t}{s} - gv \quad d = \frac{d_t}{s} + \left(1 - \frac{1}{s}\right) u_l + gv = d_t (s-1) u_r + sgv \quad (10)$$

Bei der finalen Fusion der Disparitätskarten aus einer klassischen und mehreren Schätzungen für verschieden geneigte Ebenenscharen hat sich der Ansatz von [6] bewährt: Pixelweise unabhängig wird diejenige Disparität mit der geringsten zugehörigen Summe von *Hamming*-Distanzen gewählt. Allerdings werden dabei keine Disparitäts-Kandidaten berücksichtigt, bei denen der entsprechende Links/Rechts-Check (s. o.) gescheitert ist.

Bei diesem letzten Verarbeitungsschritt bietet sich zudem die eingangs genannte Möglichkeit, innerhalb der betrachteten Szene eine Klassifikation in verschiedenen orientierte Ebenentypen vorzunehmen: Liegen etwa für eine Transformation ($g > 0.2, s \approx 1$) – typische Werte für die Bodenebene bei moderatem Rollwinkel – in einem Bildbereich deutlich bessere Gütemaße als für weitere Transformationen und fronto-paralleles Stereo vor, so ist dies ein starkes Indiz für die Fahrbahn. Vor allem letztere ist mit dieser Technik gut in den grünen Bereichen von Abb. 9 zu erkennen. Die Unterscheidung vertikaler Strukturen in fronto-parallele (rot) und tangential betrachtete (blau) gestaltet sich dagegen schwieriger, wie das auf diesen vorherrschende Magenta zeigt. Mangels geeigneter Referenzdaten können wir diese Methode, ebenso wie die dargestellte Nutzung einer Stereokamera auf einem Motorrad, in der folgenden Auswertung jedoch leider nicht detaillierter betrachten.

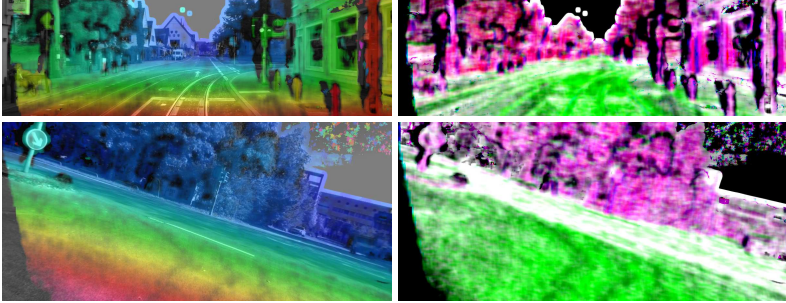


Abbildung 9: fusionierte Disparitätskarten (links) und Gütemaße der genutzten Ebenenmodelle (rechts) für PKW (oben) und Motorrad (unten): Gütemaße fronto-paralleler sowie geneigt betrachteter waag-/senkrechte Ebenen jeweils auf Rot-/Grün-/Blaukanal

5 Experimentelle Auswertung

Das folgenden Kapitel umfasst eine quantitative Untersuchung des vorgestellten Verfahrens in Bezug auf zwei Kriterien: Rechenzeit und Ergebnisqualität. Zur Interpretation von ersterer möchten wir kurz das Testsystem vorstellen, welches mit 19 x 19 cm Grundfläche deutlich kompakter und sparsamer ist als der in [1] und [14] genutzte Fahrzeugrechner im Workstation-Format. Es verfügt über je eine CPU und GPU:

- Der Prozessor *Intel Core i7-4771* bietet vier Kerne mit je 3.5 GHz und unterstützt den Vektor-Instruktionssatz *AVX2*, mittels dessen unsere Implementierung pro Kern 16 statt 8 Pixel parallel auswertet. Seine Leistungsaufnahme liegt maximal bei 84 W.
- Die Grafikkarte vom Typ *GeForce GTX 760* betreibt 1152 Kerne mit 1 GHz und benötigt hierfür bis zu 170 W. Trotz des Fehlens von Vektor-Instruktionen verarbeitet unsere Implementierung gleichzeitig 2 Pixel pro Kern [14].

Als Eingangs-Datensatz verwenden wir die *KITTI Vision Benchmark Suite* [15], deren bereits rektifizierte Stereo-Bildpaare jeweils knapp 0.5 Mpx groß sind und mit einem PKW-Versuchsträger in überwiegend städtischer Umgebung aufgenommen wurden. Daher liegt der Rollwinkel Φ der Kamera zur Fahrbahn fast immer bei etwa 0, derjenige zur Randbebauung bei $\pm\pi/2$. Die Fähigkeit zur Modellierung anderer Ebenen konnte leider entsprechend selten zum Tragen kommen. Dennoch erwies sich die in Kap. 4 beschriebene histogramm-basierte Adaption der untersuchten Transformationen an die aktuelle Szene als hilfreich: Dem klassischen fronto-parallelen Modell wurden so im Durchschnitt 1.43 weitere Ebenenscharen hinzugefügt, unter diesen in 96% der Fälle die Fahrbahn. Die Obergrenze von drei zusätzlichen Modellen wurde für 7.8% der Bilder ausgeschöpft.

Anhand der o. g. Zahlen ist einleuchtend, dass das vorgestellte Verfahren die Gesamt-Laufzeit relativ zur klassischen Disparitätsschätzung etwa verdreifacht, wie die letzte Spalte von Tab. 1 zeigt. Auch die detaillierte Aufstellung der Zeiten pro Verarbeitungsschritt in Abb. 10 belegt dies. Ein Echtzeitbetrieb kann bei einem Durchsatz von 31 (CPU) bzw. 44 Hz (GPU) jedoch nach wie vor gewährleistet werden. Bezogen auf deren Leistungsaufnahmen ergibt sich eine Energieeffizienz von 0.37 bzw. 0.26 Hz/W, im Vergleich zu für dieselbe Auflösung geschätzten 2.9 Hz/W einer modernen FPGA-Implementierung [16].

Methode	Out-Noc	Out-All	Avg-Noc	Avg-All	Dichte	Laufzeit
Hochpass/fronto-par.	14.27 %	15.57 %	2.5 px	2.8 px	77 %	12.0 / 8.7 ms
Census/fronto-par.	8.36 %	9.72 %	1.8 px	2.1 px	84 %	11.8 / 8.5 ms
Hochpass/geneigt	12.62 %	13.97 %	2.1 px	2.5 px	81 %	32.9 / 23.0 ms
Census/geneigt	7.77 %	9.13 %	1.6 px	2.0 px	87 %	32.5 / 22.6 ms

Tabelle 1: Bewertung jeder Kombination aus Hochpass-/Census-Vorverarbeitung und rein fronto-paralleler/geneigte Ebenen modellierender Disparitätsschätzung gemäß [15]: Anteil der Disparitäten mit einem Fehler von mindestens 3 px (“Out”) und durchschnittlicher Disparitätsfehler (“Avg”), jeweils in für linke und rechte Kamera sichtbaren Bereichen (“Noc”) sowie überall (“All”), Laufzeiten pro Bildpaar auf CPU/GPU

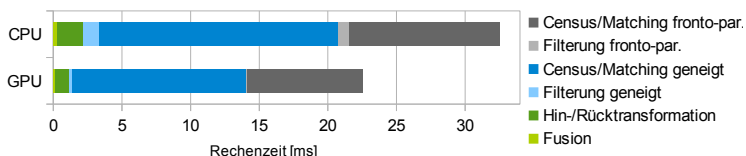


Abbildung 10: Rechenzeiten der einzelnen Schritte der vorgestellten Disparitätsschätzung: Mit ca. 0.05 ms/Bild ist die Census-Transformation nicht als separater Schritt erkennbar.

Die Bewertung der Qualität geschätzter Disparitäten basiert auf einem Vergleich mit per Laserscanner gewonnenen Referenzergebnissen. Tab. 1 fasst entsprechende Kennwerte für alle beschriebenen Varianten unserer Implementierung zusammen. Auf der Rangliste der *KITTI Vision Benchmark Suite* liegt das vorgestellte Verfahren im Mittelfeld, bietet allerdings die mit Abstand kürzeste Laufzeit. Für einen detaillierten und auch in Zukunft aktuellen Vergleich möchten wir jedoch auf die Liste im Internet³ verweisen.

6 Zusammenfassung und Ausblick

Wir haben Weiterentwicklungen eines lokalen Verfahrens zur Disparitätsschätzung präsentiert, welches u. a. durch Parallelverarbeitung echtzeitfähig ist. Eine effizient berechnen- und auswertbare Vorverarbeitung erzielte zugleich eine erste Steigerung der Ergebnisqualität. Der Kernaspekt dieses Beitrags liegt jedoch bei den Algorithmen zur korrekten Behandlung nicht nur fronto-parallel betrachteter, sondern beliebig im Raum orientierter Ebenen. Im Bereich der Fahrerassistenz ist diese Fähigkeit nicht nur, aber insbesondere für die signifikanten Rollwinkel von Einspurfahrzeugen relevant. Für die Rechenzeit unserer Implementierung bedeutet jede zusätzlich untersuchte Ebenenschar zwar eine Vervielfachung, dennoch liegt diese in einem günstigen Verhältnis zur Ergebnisqualität.

Zukünftig werden wir die Verarbeitungsschritte Filterung und Fusion weiterentwickeln: An die Stelle der Fuzzy-Logik zur Bestimmung der Gütemaße und der pixelweisen Auswahl einer Ebenenhypothese soll ein probabilistisches Modell treten. Idealerweise ermöglicht dieses einen Verzicht auf die Konsistenz-Überprüfung von linker und rechter Disparitätskarte, so dass die zeitaufwändige Schätzung von letzterer nicht mehr notwendig ist. Unabhängig davon werden wir die Strategien aus [14] zur Adaption der Parallelverarbeitung an konkurrierende Prozesse auf der hier vorgestellten komplexeren Disparitätsschätzung und gleichzeitig laufenden Fahrerassistenz-Anwendungen evaluieren.

³http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo

Literatur

- [1] B. Ranft, T. Schönwald und B. Kitt, *Parallel Matching-based Estimation – a Case Study on Three Different Hardware Architectures*, IEEE Intelligent Vehicles Symposium 2011.
- [2] C. Zinner, M. Humenberger, K. Ambrosch und W. Kubinger, *An Optimized Software-Based Implementation of a Census-Based Stereo Matching Algorithm*, International Symposium on Visual Computing 2008.
- [3] L. Ma, J. Li und H. Zhang, *A Modified Census Transform Based on the Neighborhood Information for Stereo Matching Algorithm*, IEEE International Conference on Image and Graphics 2013.
- [4] F. Stein, *Efficient Computation of Optical Flow Using the Census Transform*, DAGM Pattern Recognition Symposium 2004.
- [5] P. Burt, L. Wixson und G. Salgian, *Electronically Directed “Focal” Stereo*, IEEE International Conference on Computer Vision 1995.
- [6] N. Einecke und J. Eggert, *Stereo Image Warping for Improved Depth Estimation of Road Surfaces*, IEEE Intelligent Vehicles Symposium 2013.
- [7] K. H. Won und S. K. Jung, *Ground Plane Stereo for Obstacle Detection*, International Conference on Image and Vision Computing New Zealand 2011.
- [8] A. S. Ogale und Y. Aloimonos, *Stereo Correspondence with slanted surfaces: critical implications of horizontal slant*, IEEE Conference on Computer Vision and Pattern Recognition 2004.
- [9] R. Spangenberg, T. Langner und R. Rojas, *Weighted Semi-Global Matching and Center-Symmetric Census Transform for Robust Driver Assistance*, International Conference on Computer Analysis of Images and Patterns 2013.
- [10] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang und M. Pollefeys, *Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions*, IEEE Conference on Computer Vision and Pattern Recognition 2007.
- [11] K. Yamaguchi, T. Hazan, D. McAllester und R. Urtasun, *Continuous Markov Random Fields for Robust Stereo Estimation*, European Conf. on Computer Vision 2012.
- [12] C. Rabe, T. Müller, A. Wedel und U. Franke, *Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-time*, European Conference on Computer Vision 2010.
- [13] J. Cech, J. Sanchez-Riera und R. Horaud, *Scene Flow Estimation by Growing Correspondence Seeds*, IEEE Conf. on Computer Vision and Pattern Recognition 2011.
- [14] B. Ranft und O. Denninger, *Run-time Adaptation to Heterogeneous Processing Units for Real-time Stereo Vision*, IEEE International Conference on Embedded Software and Systems 2012.
- [15] A. Geiger, P. Lenz und R. Urtasun, *Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite*, IEEE Conference on Computer Vision and Pattern Recognition 2012.
- [16] H. Sahlbach, R. Ernst, S. Wonneberger und T. Graf, *Exploration of FPGA-based Dense Block Matching for Motion Estimation and Stereo Vision on a Single Chip*, IEEE Intelligent Vehicles Symposium 2011.

Registration of a Multi-Camera System with a 3D Laser Range Finder

Jose Esparza^{*†}, Leo Vepa^{*†}, Michael Helmle^{*} und Bernd Jähne[†]

Abstract:

This paper presents a novel procedure for the registration of a 3D Laser Range Finder to a multi-camera system. In order to obtain reference depth information for evaluation and benchmarking of vision-based driver assistance functionalities based on a multi-camera system, our goal is to find the pose of the Laser Range Finder which provides a reprojection error minimized over all available cameras.

Keywords: Lidar, Calibration, Multi-Camera System, Fish-eye

1 Introduction

In the context of driving assistance, the gathering of information about the surrounding of a vehicle is the first step towards functionalities that can support a driver while maneuvering. Automotive systems employ different kinds of sensors as listed in [1] to perceive this information. In particular, cameras are of special interest, since they enable numerous driver assistance functionalities due to their versatile capabilities, as described in [2]. Recently, multi-camera systems of the kind described in [3] are used in low-speed driver assistance to provide a 360° imagery of the vehicle surrounding. Computer vision algorithms can also be applied to all available cameras of these systems to obtain information about the vehicle's environment.

In order to bring vision-based driver assistance systems to the market, the developed image processing algorithms have to fulfill high requirements regarding functional safety and legal constraints [4] and need to be evaluated before putting them into a real product. Lately, 3D Laser Range Finders (LRF) were introduced as a possibility to obtain ground truth measurements in real world scenarios for testing and evaluating computer vision algorithms [5, 6]. In order to use the LRF as reference for these algorithms applied to multi-camera systems, a registration of the LRF with all the cameras in the system is needed.

Previous work in this field coped with the registration of 2D LRFs using planar calibration patterns to perspective [7] or catadiotric cameras [8]. These methods were extended to work with 3D LRFs [9, 10, 11]. In [12], multiple cameras were registered to a 3D LRF by estimating each camera position individually. In the work of [13], a calibration toolbox

^{*}Robert Bosch GmbH (e-mail: Firstname.Lastname@de.bosch.com).

[†]Heidelberg Collaboratory for Image Processing (HCI), Heidelberg University
(e-mail: Firstname.Lastname@ivr.uni-heidelberg.de).

was presented that provides registration of a 3D LRF with a stereo camera system in the 3D domain by matching automatically detected calibration target planes.

The presented methods provide only a locally best solution since they are limited to determining the pose of an LRF relative to a single camera or a stereo system. Hence these approaches show a visible mismatch when evaluated on the other cameras of the system (See Fig.5c). In contrast to this, we introduce a new registration method of a 3D LRF with a multi-camera system. Instead of using a single camera, a set of cameras is considered which are previously registered and referenced with respect to a global coordinate system. The LRF is then registered by minimizing the reprojection error of the LRF measurements over all cameras of the system. We demonstrate that our global method provides a more stable global pose estimation of the LRF.

This work is structured as follows. Section 2 provides a review of the preliminary work required for the proposed method. In Section 3 our registration algorithm is presented and Section 4 provides an evaluation of the proposed method, drawing a comparison to existing approaches. Section 5 presents a summary and an outline to future work.

2 Preliminaries

2.1 Cameras

Multi-camera systems employed for driving assistance commonly make use of fish-eye optics, providing horizontal field of views (FoV) of more than of 180° . Such optics enable monitoring of the whole surrounding of a vehicle with four cameras only, in a configuration as described in [3]. Several models have been proposed in literature to describe distortions due to the fish-eye optics [14, 15]. The so-called intrinsic calibration is an important characteristic of the camera since it permits the mapping of pixels to view rays from the camera center. We use the model presented in [14] for its simplicity and small overall reprojection error.

Previous registration of the multi-camera with respect to a global coordinate system is assumed. For this purpose, several methods like [16] have been proposed in literature. In our case, this registration is done by means of special calibration targets as well as additional cameras and bundle adjustment. As reference for the vehicle's coordinate system, DIN70000 convention is considered, which establishes the origin to be in the middle of the rear axis, at ground level.

2.2 3D Laser Range Finder

In recent years, 3D LRFs gained a lot of popularity within the automotive industry, especially in the context of autonomous driving [17]. They enable to monitor the vehicle surrounding with high precision 3D measurements. Distances to reflecting surfaces are measured and these measurements are usually converted to 3D coordinates, as seen from the virtual center of the LRF.

Since manual inspection of 3D data is complex and tedious, a 2D depth map based on range information is generated in order to ease the task. For rotating LRFs, a set of key directions can be defined, and the closest measurements to these be taken. This generates a full 2D matrix that encodes depth which can be rastered as an image. This process

resembles that of a cylindric warping of an image sequence acquired by a camera panning 360°. An example of the resulting view is shown in Fig.1. Although this preparation step is used for simplicity of data inspection, the original 3D measurements are to be considered during the whole calibration process.



Figure 1: Example of rasterized depth map, as seen from the 3D LRF virtual center. In red: Target used for manual selection of keypoints.

2.3 Selection of Corresponding Measurements

The proposed method relies on manual selection of corresponding points visible across different sensors, which is a critical part during the registration process. In order to ease the correspondence search, a calibration target has been designed. This target is such that its colour properties make it distinguishable on camera images and, simultaneously, its geometry is clearly recognizable in the range data from the LRF, as highlighted in Fig.1.

Before registration, keypoints are manually labelled, each label consisting of the pixel position of the target's center in the camera image and the corresponding 3D coordinate measured by the LRF.

3 Registration of 3D Laser Range Finder

The extrinsic calibration is based on the minimization of the global reprojection error of key geometrical points onto the different camera images. Keypoints are selected by means of the proposed target in a way that the whole surrounding of the vehicle is covered. Each keypoint shall be selected such that it is present on the LRF data and on, at least, one camera image. Also all cameras should contribute with keypoints, since the method is aimed at a global solution that minimizes the reprojection error over all cameras.

For each keypoint, we consider a measurement \mathbf{S} :

$$\mathbf{S} = \{\mathbf{U}^k, \mathbf{P}\} \quad (1)$$

with k being the camera index, $\mathbf{P} = (X, Y, Z)^T$ the measured 3D position of the keypoint with respect to the LRF centre and \mathbf{U}^k a 3D-vector defined as the direction on which the keypoint lies, as observed from the virtual center of camera k . \mathbf{U}^k can be calculated from the pixel coordinates $\mathbf{u}^k = (u, v)^k$ corresponding to the keypoint in camera k , given that the cameras are intrinsically calibrated. This is done according to the model proposed in [14].

Let us now assume a set of parameters $\theta = \{X_L, Y_L, Z_L, \alpha_L, \gamma_L, \beta_L\}$ that define the pose of the LRF with respect to the vehicle's coordinate system. Based on this set of parameters, a 4×4 translation and rotation matrix $M_{D70K, LRF}(\theta)$ can be defined, such

that a point \mathbf{P} observed from the LRF origin, can be described as another point $\mathbf{Q}(\theta)$, with respect to the vehicle’s origin, by means of Eq.2.

$$\begin{pmatrix} \mathbf{Q}(\theta) \\ 1 \end{pmatrix} = M_{D70K, LRF}(\theta) \cdot \begin{pmatrix} \mathbf{P} \\ 1 \end{pmatrix} \quad (2)$$

Similarly, for any given point \mathbf{Q} represented in vehicle coordinates, a transformation $M_{k, D70K}$ to local camera frame coordinates, namely \mathbf{V}^k , can be applied by means of Eq.3. For every camera k , the transformation matrix $M_{k, D70K}$ is known, given the previous registration of the multi-camera system with respect to the global origin of the vehicle.

$$\begin{pmatrix} \mathbf{V}^k \\ 1 \end{pmatrix} = M_{k, D70K} \cdot \begin{pmatrix} \mathbf{Q} \\ 1 \end{pmatrix} \quad (3)$$

By combining (2) and (3), it is now possible to describe any point \mathbf{P} observed by the LRF with respect to the coordinate system of any camera k as function of the parameter set θ , as in Eq.4

$$\begin{pmatrix} \mathbf{V}^k(\theta) \\ 1 \end{pmatrix} = M_{k, D70K} \cdot M_{D70K, LRF}(\theta) \cdot \begin{pmatrix} \mathbf{P} \\ 1 \end{pmatrix} \quad (4)$$

For each keypoint \mathbf{S}_i and for a given LRF pose estimate θ , an error measure $e_i(\theta)$ can be defined, as the angular distance between \mathbf{V}_i^k , and \mathbf{U}_i seen from camera k ’s center. Considering $\hat{\mathbf{V}}_i^k$ and $\hat{\mathbf{U}}_i$ as the normalized \mathbf{V}_i^k and \mathbf{U}_i respectively, $e_i(\theta)$ can be obtained by means of Eq.5.

$$e_i(\theta) = \arccos\left(\hat{\mathbf{V}}_i^k(\theta)^T \cdot \hat{\mathbf{U}}_i\right) \quad (5)$$

In our setup, as in the work of [10], the angular error is more suitable for characterising the quality of the pose estimation than the pixel error. This is due to the non uniform pixel density over the whole FoV on cameras with fish-eye optics.

Through the use of non-linear search algorithms, a set of parameters $\tilde{\theta}$ can be obtained, by minimizing the cost function over all reference keypoints, as proposed in Eq.6

$$\tilde{\theta} = \arg \min_{\theta} \frac{1}{2} \sum_i e_i^2(\theta) = \arg \min_{\theta} \frac{1}{2} \sum_i \arccos^2\left(\hat{\mathbf{V}}_i^k(\theta)^T \cdot \hat{\mathbf{U}}_i\right) \quad (6)$$

such that the overall reprojection error is minimum over all cameras on the system.

4 Results

For evaluating the proposed method, a vehicle equipped with a Velodyne HDL-64E S2 LIDAR and a multi-camera system in a common surround-view configuration [3] was used. The Velodyne LIDAR provides a full 360° field of view on the horizontal direction and a field of view of 26.8° on the vertical direction, distributed over 64 independent laser beams. This accounts for a vertical angular resolution equivalent to approximately 0.43° per laser beam. As described in Section 2.2, a rasterization of the depthmap is generated for the sake of data inspection. For this, horizontal angular steps of 0.5° are considered.

Our multi-camera system consists of four fish-eye cameras with a horizontal FoV of 185° and a resolution of 1280 × 960 pixels. The cameras are mounted on the front and

rear ends of the vehicle, as well as on the left and right external mirrors. A set of 240 measurements (60 measurements per camera) was manually labelled, each comprising the camera index, the 3D position of the target measured from the LRF center and its corresponding 2D pixel coordinates in the camera image, as defined in Eq.1. For the measurements, the positions of the target were chosen such that different ranges are covered, while posing a uniform distribution over the FoVs of all cameras.

In order to evaluate the proposed method, focus has been set on two aspects. First, evaluation of the robustness of the presented approach over multiple runs with different measurements. Second, evaluation of the reprojection error of the 3D LRF measurements into the image plane of the cameras and comparison the results to a single-camera registration method, as proposed in [10].

4.1 Evaluation of the Robustness of the Pose Estimation

The robustness of the presented approach is evaluated by examining the variation of the estimated pose parameters over multiple repetitions. In particular, we aim at evaluating the stability of the parameters, depending on the amount of measurements used for the pose estimation. Hence the number of keypoints used for the registration of the LRF was varied between 1 and 10 per camera (between 4 and 40 in total). This process was repeated 100 times. Fig.2 shows the distribution of the six pose parameters over different iterations. Careful analysis of these results shows that the estimation of the parameters becomes more stable when the amount of reference keypoints is increased. However, the variance of all parameters does not decrease significantly after a certain point. We conclude that with approximately 5 measurements per camera (20 measurements for the whole system) a stable pose can be estimated. A good selection of keypoints may further reduce the number of measurements needed. Table 1 shows the average estimated position and orientation of the LRF considering 5 reference points per camera.

\bar{X} [m]	σ_X [m]	\bar{Y} [m]	σ_Y [m]	\bar{Z} [m]	σ_Z [m]
0.894	0.010	-0.012	0.009	1.993	0.009
$\bar{\alpha}$ [deg]	σ_α [deg]	$\bar{\gamma}$ [deg]	σ_γ [deg]	$\bar{\beta}$ [deg]	σ_β [deg]
0.141	0.069	0.981	0.135	0.592	0.129

Table 1: Averaged parameter estimation of the 3D LRF pose over 100 repetitions, considering 5 reference points per camera.

4.2 Evaluation of the Reprojection Error

The reprojection error represents a good quality measure to describe the accuracy of the estimated parameters that define the pose of the LRF with respect to the global coordinate system [18]. It can be expressed in terms of pixel error or angular error. Although the angular error may be a more correct metric, given the non-uniform distribution of pixels over the whole field of view of our cameras, the pixel error is also analyzed, since it is an important figure for data association and in data fusion systems [10]. A test set was defined, consisting of 40 measurements uniformly distributed over all cameras in the

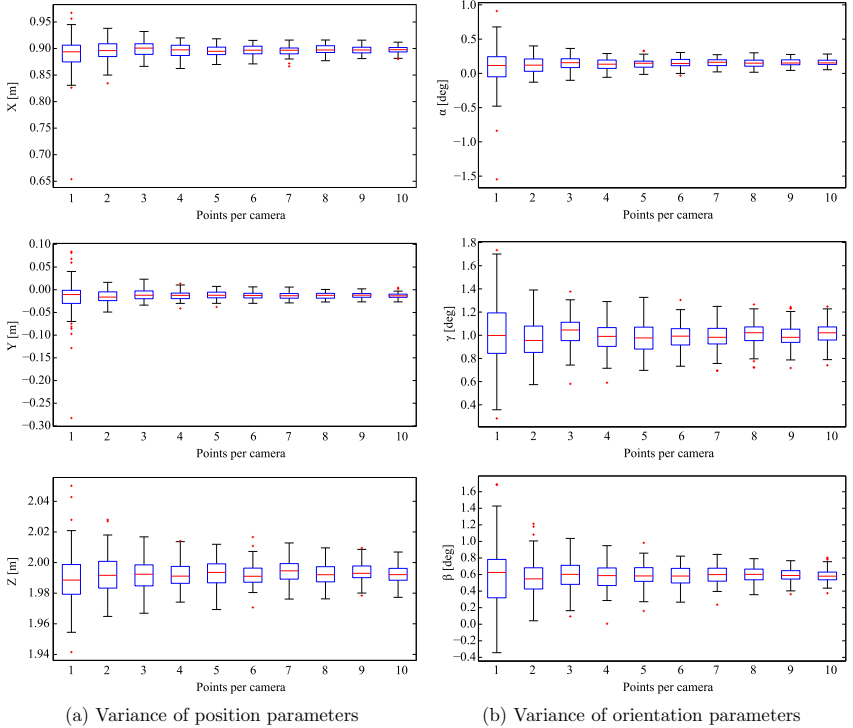


Figure 2: Variance of position (a) and orientation (b) parameters of the 3D LRF registration based on the number of measurements per camera.

system. The evaluation was performed by reprojecting the 3D LRF measurements of the test set into the associated camera image using the estimated pose as in Eq.4. The results were compared with manually labelled reference data.

Based on the results presented in Sec.4.1, five measurements for each of the four cameras (20 measurements in total) were considered for the registration of the LRF to the multi-camera system. Table 2 shows the average errors obtained. Given the angular resolutions of the LRF data which was considered for the experiments (namely 0.43° and 0.5° for the vertical and horizontal directions) the resulting 0.624° error can be considered a good result.

As reference, we compared our results with a single-camera registration as proposed in [10]. We registered the LRF to every camera of our system individually using 20 measurements for the pose estimation. Evaluation is in every case performed over all cameras of the system. For all the experiments, the registration process was repeated

\bar{e} [deg]	σ_e [deg]	\bar{e} [pel]	σ_e [pel]
0.624	0.285	4.649	2.308

Table 2: Reprojection error after registration of the 3D LRF with the multi-camera system.

100 times. In Fig.3, the mean of the reprojection error is shown for each combination of datasets used for pose estimation and evaluation. As expected, it can be observed that

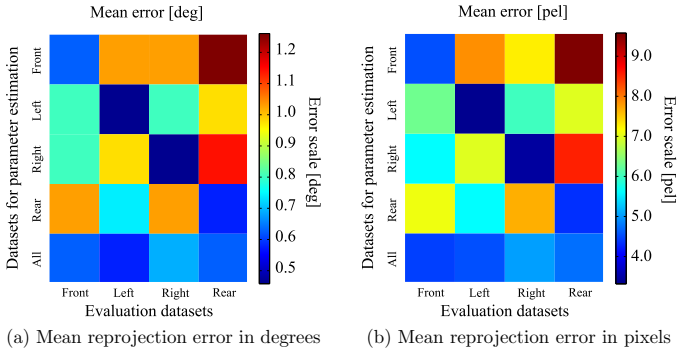


Figure 3: Comparison of the mean reprojection error in degrees (a) and in pixels (b) of different single and a multi-camera LRF registration. The error is depicted for every camera individually.

the cases where registration and evaluation are conducted with data from the same single camera produce the smallest error. One can also notice that the error increases strongly if the estimated pose is evaluated on any other cameras of the system. Evaluation on the cases where all cameras are used for the registration, on the other hand, produces a much smaller reprojection error over all cameras in the system. This global best is only slightly outperformed by the cases where both pose estimation and evaluation are performed with data of the same camera individually.

Although the average reprojection error is a good indicator of for the accuracy of the registration, it is also interesting to look into the maximum reprojection error found for each case. This approximates the worst-case scenario and defines an upper bound limit to the magnitude of the errors which are to be expected. Results are shown in Fig.4.

A graphical comparison of the results obtained after registration of the LRF globally with the multi-camera system, and with a single camera is presented in Fig.5. The improvement of our global multi-camera approach with respect to a single-camera registration is clearly visible by comparison of Fig.5c and Fig.5d.

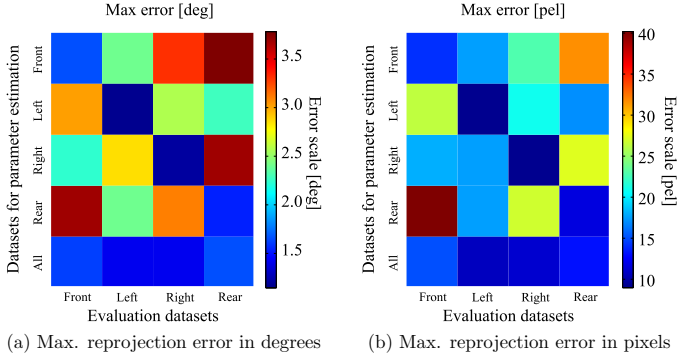


Figure 4: Comparison of the maximal reprojection error in degrees (a) and in pixels (b) of different single and a multi-camera LRF registration. The error is depicted for every camera individually.

5 Summary and Outline

We presented a new approach for global registration of a 3D LRF to a multi-camera system. Our proposed approach shows improved results compared to a state of the art LRF registration with respect to a single camera, if evaluated for the whole system. Future work will include the development of a fully autonomous keypoint selection based on automatic target detection and natural scenes.

References

- [1] William Fleming. New automotive sensors - A review. *Sensors Journal, IEEE*, 8(11):1900–1921, 2008.
- [2] Uwe Franke, David Pfeiffer, Clemens Rabe, Carsten Knoeppel, Markus Enzweiler, Fridtjof Stein, and Ralf G. Herrtwich. Making bertha see. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2013.
- [3] Paul Furgale et al. Toward Automated Driving in Cities using Close-to-Market Sensors, an Overview of the V-Charge Project. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 809–816, Gold Coast, Australia, 23–26 June 2013.
- [4] Fridtjof Stein. The challenge of putting vision algorithms into a car. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 89–94. IEEE, 2012.
- [5] Sandino Morales and Reinhard Klette. Ground truth evaluation of stereo algorithms for real world applications. In *Computer Vision-ACCV 2010 Workshops*, pages 152–162. Springer, 2011.

- [6] 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.
- [7] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2301–2306. IEEE, 2004.
- [8] C. Mei and Patrick Rives. Calibration between a central catadioptric camera and a laser range finder for robotic applications. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 532–537, 2006.
- [9] Ranjith Unnikrishnan and Martial Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. Technical Report CMU-RI-TR-05-09, Robotics Institute, Pittsburgh, PA, July 2005.
- [10] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4164–4169, 2007.
- [11] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Extrinsic calibration of a 3D laser scanner and an omnidirectional camera. In *7th IFAC Symposium on Intelligent Autonomous Vehicles*, volume 7, 2010.
- [12] M. Haselich, R. Bing, and D. Paulus. Calibration of multiple cameras to a 3D laser range finder. In *Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on*, pages 25–28, 2012.
- [13] Andreas Geiger, Frank Moosmann, Omer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3936–3943. IEEE, 2012.
- [14] C. Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3945–3950, 2007.
- [15] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5695–5701, 2006.
- [16] Lionel Heng, Bo Li, and Marc Pollefeys. Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.
- [17] Frank Moosmann and Christoph Stiller. Velodyne slam. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 393–398. IEEE, 2011.
- [18] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

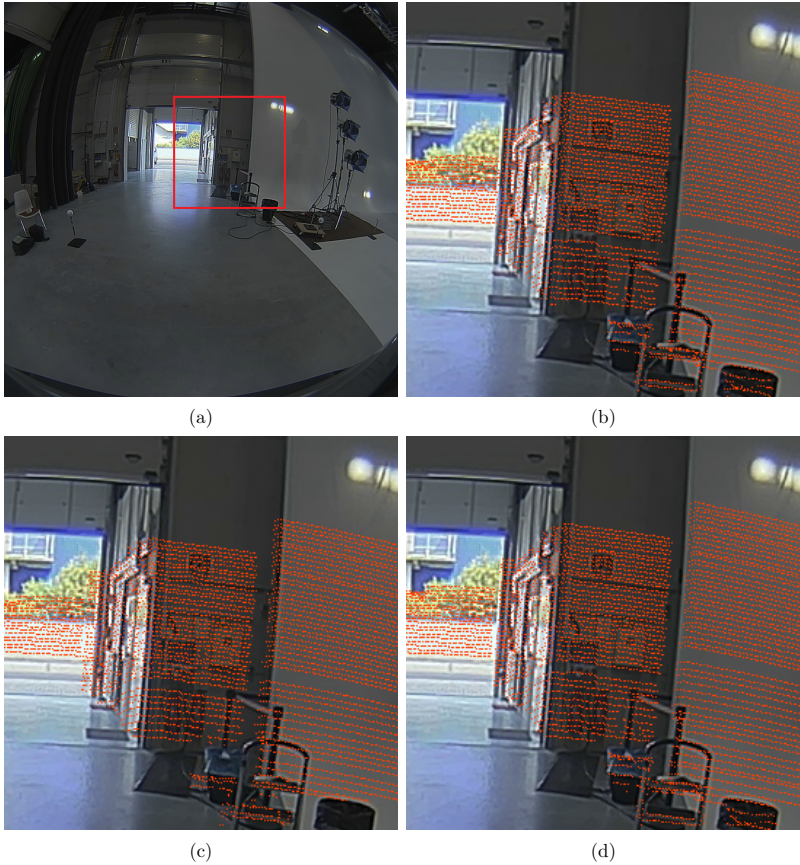


Figure 5: Reprojection of LRF points using different input for registration. The quality of the registration can be clearly observed by comparing the edges of the walls to the reprojected points. a) Original image. In red: area selected for visualization. b) Registration using rear camera only - LRF points projected onto rear camera image. c) Registration using front camera only - LRF points projected onto rear camera image. d) Registration using all cameras - LRF points projected onto rear camera image

Generatives Prädiktionsmodell zur frühzeitigen Spurwechselerkennung

Mohammad Bahram* und Artur Lohrer† und Michael Aeberhard‡

Zusammenfassung: In diesem Beitrag wird eine modellbasierte Prädiktion für das Spurwechselmanöver anderer Verkehrsteilnehmer auf der Autobahn vorgestellt. Dies stellt eine wichtige Grundvoraussetzung für eine vorausschauende Fahrstrategie des hochautomatisierten Fahrens dar. Der vorgestellte Ansatz unterteilt das Prädiktionsproblem in Subprobleme und vereinfacht dadurch die Modellierung. Die native Erweiterbarkeit und Transparenz sind weitere Vorteile des Ansatzes, womit er sich von rein diskriminativen Klassifizierungsverfahren abhebt. Die Funktionalität ist anhand eines Versuchsträgers im realen Verkehr erfolgreich getestet und evaluiert worden.

Schlüsselwörter: Hochautomatisiertes Fahren, Manöverprädiktion, generative Klassifikation, maschinelles Lernen

1 Einleitung

Das hochautomatisierte Fahren (HAF) auf der Autobahn ist ein wichtiger Grundstein der zukünftigen Mobilität. Durch den Einsatz eines solchen Assistenzsystems kann die Anzahl der Verkehrsunfälle auf der Autobahn, die vorwiegend durch fahrlässige Überholmanöver verursacht sind [1], signifikant reduziert werden. Ein weiteres Einsatzszenario des Systems ist die vollständige Übernahme der Fahraufgabe, falls eine schwerwiegende gesundheitliche Beeinträchtigung des Fahrers (z. B. Herzinfarkt) erkannt wurde [2, 3]. Die kontinuierliche Zunahme des Automatisierungsgrads kann zudem zur Verbesserung des Verkehrsflusses und Reduzierung des Verbrauchs führen [4].

Die Fahrstrategie stellt bei der hochautomatisierten Fahrfunktion die intelligente Verhaltenssteuerung des kognitiven Fahrzeugs dar. Innerhalb der Fahrstrategie wird anhand verschiedener Informationen, wie der Fahrer-Intention, Verkehrssituation sowie back-end Information, ein high-level Entscheidungsfindungsprozess durchgeführt. Die Fahrstrategie sollte neben reaktivem Verhalten ebenfalls eine vorausschauende Planung (d.h. ein deliberatives Verhalten) aufweisen. Auf diese Weise wird eine frühzeitige Reaktion auf kritische Ereignisse innerhalb des Umfelds (z. B. knapp einscherende Fahrzeuge) ermöglicht. Für die vorausschauende Entscheidungsfindung der Fahrstrategie ist eine robuste und recheneffiziente Manöverprädiktion der anderen Verkehrsteilnehmer erforderlich.

Ein solches Prädiktionsmodell kann ebenfalls unabhängig in anderen Assistenzsystemen integriert werden. Beispielsweise kann durch die Erweiterung der adaptiven Ge-

*arbeitet bei BMW Forschung & Technik in München, (mohammad.bahram@bmw.de).

†schrieb seine Abschlussarbeit bei BMW Forschung & Technik in München, (artur.lohrer@tum.de).

‡arbeitet bei BMW Forschung & Technik in München, (michael.aeberhard@bmw.de).

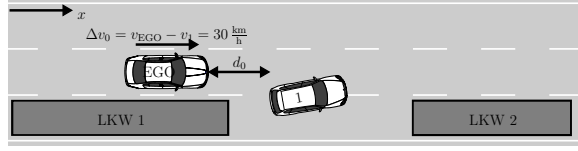


Abbildung 1: Beispiel einer kritischen Einscherer-Situation.

schwindigkeitsregelung (ACC), um das entwickelte Prädiktionsmodell, riskante Manöver des Vorderfahrzeugs frühzeitig erkannt und darauf in geeigneter Weise reagiert werden. Die Relevanz der Manöverprädiktion kann anhand des folgenden Szenarios verdeutlicht werden (siehe Abbildung 1). Es wird angenommen, dass das Ego-Fahrzeug bei der aktiven Geschwindigkeitsregelung mit einem maximalen konstanten Bremsmoment von $a_{\text{Brems,max}} = -4 \text{ m/s}^2$ verzögern kann. Der Abstand zwischen Ego-Fahrzeug und Fahrzeug 1 zum Zeitpunkt der Spurüberschreitung ist im Folgenden als d_0 definiert. Der Abstand, bei dem das Bremsmanöver eingeleitet wird, ist als d_{Brems} gekennzeichnet. Es wird zudem die konservative Annahme getroffen, dass das Fahrzeug 1 das Überholmanöver mit einer konstanten Geschwindigkeit fortsetzt. Der relative Abstand Δd kann durch das Weg-Zeit-Gesetz der gleichmäßig beschleunigten Bewegung zu jedem Zeitpunkt bestimmt werden. Die Kollisionsfreiheit ist theoretisch garantiert falls:

$$\Delta d = x_t^{\text{EGO}} - x_t^1 = \frac{1}{2} a_{\text{Brems,max}} t^2 + \Delta v_0 t - d_{\text{Brems}} \leq 0 \quad (1)$$

Löst man die Gleichung nach t auf, erhält man die Bedingung der kollisionsfreiheit für den Abstand d_{Brems} :

$$t_{1,2} = -\frac{\Delta v_0}{a_{\text{Brems,max}}} \pm \sqrt{\frac{\Delta v_0^2}{a_{\text{Brems,max}}^2} + \frac{2d_{\text{Brems}}}{a_{\text{Brems,max}}}} \implies d_{\text{Brems}} \geq \left| \frac{\Delta v_0^2}{2a_{\text{Brems,max}}} \right| \quad (2)$$

Im reaktiven Fall wird erst dann auf das einscherende Fahrzeug geregelt, falls die Mitte des Fahrzeugs 1 (modelliert durch ein Rechteck) die Fahrbahnmarkierung überquert, also $d_0 = d_{\text{Brems}}$. Setzt man nun die Werte aus dem Beispielszenario in der Gleichung 2 ein, sollte der Anfangsabstand demnach mindestens $d_0 = d_{\text{Brems}} \geq 8.7 \text{ m}$ betragen. Dasselbe Szenario wird nun unter der Voraussetzung betrachtet, dass das Prädiktionsmodell mit einem Prädiktionshorizont von t_{predict} eine robuste Vorhersage über das künftige Manöver des Fahrzeugs 1 treffen kann. Das Ego-Fahrzeug kann demnach bereits um t_{predict} vorher mit dem Bremsen anfangen. Daher gilt

$$d_{\text{Brems}} = d_0 + \Delta v_0 t_{\text{predict}} \quad (3)$$

Setzt man einen konservativen Prädiktionshorizont von einer Sekunde voraus ($t_{\text{predict}} = 1 \text{ s}$), erhält man $d_0 \geq 0.4 \text{ m}$. Somit wird der Risikobereich um 8.3 m verringert. Dieses Beispiel verdeutlicht die Sicherheit, die durch die Anwendung des Prädiktionsmodells gewonnen werden kann.

2 Problemstellung

Im Autobahnscenario können für einen kleinen Prädiktionshorizont die Geschwindigkeiten in longitudinaler Richtung annähernd als konstant angenommen werden. Diese Annahme gilt allerdings nicht in lateraler Richtung, da das Spurwechselmanöver einen nicht linearen Verlauf aufweist. Daher wird ein separates Prädiktionsmodell benötigt, welches die Wahrscheinlichkeit eines Spurwechsels $P(c_i|\mathbf{f})$ bestimmt. Gesucht ist:

$$P(c_i|\mathbf{f}) = g_i(\mathbf{f}) \quad (4)$$

Mit

$$\begin{aligned} c_i &\in C = \{\text{Spurwechsel links, Spurhalten, Spurwechsel rechts}\} \\ \mathbf{f} &\in \mathbb{R}^n \hat{=} \text{Merkmalsraum} \\ g_i(\mathbf{f}) &= \text{Klassifikationsfunktion der } i\text{-ten Klasse} \end{aligned}$$

Bevor das entwickelte Prädiktionsmodell im nächsten Abschnitt detailliert erläutert wird, stellen wir im Folgenden geeignete Anforderungen an das Prädiktionsmodell. Anschließend wird den Stand der Technik diskutiert.

2.1 Anforderungsanalyse

Das Prädiktionsmodell sollte folgende Anforderungen erfüllen:

- Probabilistische Repräsentation: Die Manöverprädiktion ist stets mit Unsicherheiten behaftet. Diese sollte deshalb in Form von Wahrscheinlichkeiten modelliert werden.
- Erweiterbarkeit: Das Prädiktionsmodell sollte um neue Merkmale nativ erweiterbar sein. Diese Tatsache garantiert die künftige Erweiterbarkeit.
- Ressourcenschonende Berechnung: Da die Prädiktion für möglichst viele Teilnehmer zu jedem Zeitschritt erfolgen soll, ist eine minimale Rechenkomplexität erforderlich.
- Transparenz: Ein Prädiktionsmodell mit höherem Erklärungsgehalt besitzt eine bessere Nachvollziehbarkeit und Absicherung.

2.2 Stand der Technik

Die Prädiktionsmodelle für Spurwechsel können grundsätzlich in die zwei Kategorien intentionbasierte und trajektorienbasierte Prädiktion eingeteilt werden. Bei der ersten Kategorie wird versucht, die Intention (bzw. den Wunsch) der anderen Fahrer für verschiedene Szenarien anhand von Verhaltensmodelle vorherzusagen. Die Komplexität steigt abhängig vom Modell zum Teil erheblich. Außerdem sind die Verhaltensmodelle meist schwer evaluierbar. Für ein aktuelles Beispiel eines solchen Prädiktionsmodells wird auf [5] verwiesen.

Bei der trajektorienbasierten Prädiktion wird aufgrund der gemessenen Sensordaten das künftige Manöver präzisiert. Ein aktuelles Beispiel (für die Manöverprädiktion des Eigenfahrzeugs) ist in [6] beschrieben. Die Autoren benutzen eine Kombination aus support vector machines und Bayes-Filter. Ein Vorteil dieses Ansatzes ist, dass hier keine

Modellannahmen benötigt werden. Die Klassifikation ist allerdings aufgrund des diskriminativen Verfahrens wenig transparent. Weiterhin sind diskriminative Methoden eher als Black-box-Modelle zu verstehen, die weniger Generalität (Erweiterung) zulassen [7].

Im nächsten Abschnitt wird ein generatives Prädiktionsmodell auf Basis von Trajektorie vorgestellt und um ein effizientes Intentionsmodell (Risikobetrachtung) erweitert, welches ohne die oben genannte Nachteile auskommt.

3 Generatives Prädiktionsmodell

3.1 Ansatz

Eine direkte Modellierung der Spurwechselwahrscheinlichkeit erweist sich als schwierig. Aus diesem Grund wird das Problem weiter unterteilt und zunächst die Verteilungen der Merkmale innerhalb der einzelnen Klassen betrachtet. Dabei kommt der Satz von Bayes zum Einsatz [8]:

$$P(c_i|\mathbf{f}) = \frac{P(\mathbf{f}|c_i)P(c_i)}{\sum_j P(\mathbf{f}|c_j)P(c_j)} = g_i(\mathbf{f}) \quad (5)$$

Neben der Vereinfachung der Modellierung bietet dieser Ansatz eine native Erweiterbarkeit, einfaches Training und Integration von Modellwissen über die einzelnen Klassen. Dieses Vorgehen wird als Diskriminanzanalyse bzw. im Englischen als generative classifier bezeichnet.

3.2 Merkmalsraum und Modellierung

Als Merkmale werden zunächst die gemessenen Trajektorien der einzelnen Fahrzeuge betrachtet. Weitere Einflüsse wie Interaktion und Intention werden vorerst vernachlässigt. Dadurch reduziert sich der Merkmalsraum auf physikalisch messbare Größen. Innerhalb der Trajektorien sind vor allem die Querablage d_y und Quergeschwindigkeit \dot{d}_y der einzelnen Fahrzeuge relativ zu deren Fahrstreifenmitte (Frenet-Koordinatensystem) relevant. Hierdurch wird die richtige Prädiktion auch in Kurven garantiert. Die Querbeschleunigung wird zugunsten der Robustheit vernachlässigt. Der Merkmalsvektor ist zunächst definiert als:

$$\mathbf{f} \in \mathbb{R}^{n=2} = \begin{pmatrix} d_y \\ \dot{d}_y \end{pmatrix} \quad (6)$$

Die Merkmale werden innerhalb der einzelnen Klassen als normalverteilt und mit unterschiedlichen Kovarianzen modelliert. Weiterhin wird angenommen, dass die Querablage und Quergeschwindigkeit bedingt stochastisch unabhängig voneinander sind (Naive Bayes Ansatz). Dies trifft streng genommen nur auf ein freies Teilchen zu. Die Näherung erweist sich aber innerhalb der einzelnen Klassen als hinreichend genau und vereinfacht das Training des Modells.

3.3 Training

Das Trainieren des Klassifizierungsalgorithmus wird durch die Unterteilung in Subprobleme vereinfacht. Es wird konkret für jede Klasse die n -dimensionale Normalverteilung

der Merkmale gelernt. Hierzu wurden 13 Trainingsszenen mit den Trajektorien der erfassten Fremdfahrzeuge auf der Autobahn aufgenommen. Die Szenen haben insgesamt eine Gesamtdauer von etwa 45 Minuten. Dabei wurden 25 Spurwechsel von verschiedenen Fremdfahrzeugen erfasst. Eine wesentliche Herausforderung beim Trainieren des Prädiktionsmodells stellt die Zuordnung (engl. labelling) der Trajektorienabschnitte zu den einzelnen Klassen dar. Es wurde hierfür ein Algorithmus entwickelt, welcher automatisch diese Zuordnung durchführt. Eine gute Separierbarkeit zwischen den Klassen im Trainingsset wurde dadurch ermöglicht, dass die Trajektorienpunkte eines Spurwechselmanövers weiter in drei verschiedenen Abschnitten unterteilt sind. Der Abschnitt zwei (engl. Region of Interest) stellt sich als das signifikante Segment des Spurwechselmanövers dar (siehe Abbildung 2). Erst in diesem Unterbereich des Zustandsraums ist eine klare Differenzierung zum Spurhalte-Manöver möglich. Aus diesem Grund wird zum Trainieren der Spurwechsel-Klassen nur diesen Abschnitt verwendet. Die Abbildung 3 zeigt das reduzierte Trainingsset mit der 95% Konfidenzintervalle der gelernten Normalverteilungen. Die Ausreißer können sowohl durch nicht modellierte Manöver wie z. B. doppelter Spurwechsel als auch durch Messrauschen verursacht werden.

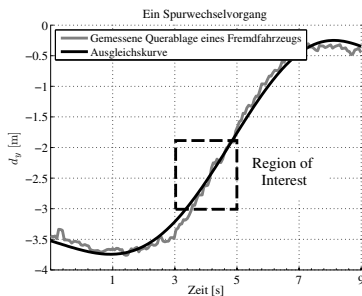


Abbildung 2: Eine Spurwechsel-Trajektorie und das Region of Interest.

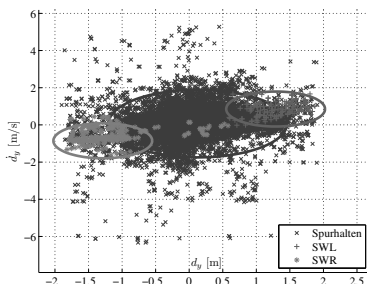


Abbildung 3: Reduzierte Trainingsdaten und gelernte Klassen der einzelnen Fahrmanöver.

3.4 Erweiterbarkeit

Der vorgestellte Ansatz bietet ein geeignetes Framework für Erweiterungen. Hierfür können weitere unabhängige Merkmale \mathbf{u} wie Blinker und Interaktionsmodelle integriert werden:

$$P(c_i|\mathbf{f}, \mathbf{u}) = \frac{P(d_y|c_i) P(\dot{d}_y|c_i) P(\mathbf{u}|c_i) P(c_i)}{\sum_j P(d_y|c_j) P(\dot{d}_y|c_j) P(\mathbf{u}|c_j) P(c_j)} \quad (7)$$

Exemplarisch wurde das Prädiktionsmodell um eine Risikobetrachtung $P(\mathbf{u} = r|c_i)$ erweitert, wobei das Kollisionsrisiko r aus TTC (engl. Time To Collision) über eine stückweise lineare Sättigungsfunktion wie folgt modelliert wird:

$$r_{\text{ttc}} = \begin{cases} 1, & \text{if } t_{\text{ttc}} \leq t_{\text{ttc},\text{min}} \\ 0, & \text{if } t_{\text{ttc}} \geq t_{\text{ttc},\text{max}} \\ 1 - \frac{t_{\text{ttc}} - t_{\text{ttc},\text{min}}}{t_{\text{ttc},\text{max}} - t_{\text{ttc},\text{min}}} & \end{cases} \quad (8)$$

Andere komplexe Risikomodelle wie in [5] können hier ebenfalls eingesetzt werden.

4 Testaufbau

Das Prädiktionsmodell wurde anhand aufgenommener Daten aus dem realen Verkehr trainiert und anschließend sowohl in einem Fahrsimulator als auch auf der Autobahn in Echtzeit getestet. Abbildung 4 zeigt ein aufgenommenes Autobahnzenario. Die Spurwechselprädiktion wird hier für das rot markierte Fahrzeug im Abstand von ungefähr 80 Meter für drei Zeitpunkte demonstriert. Die Plots zeigen die gemessenen Merkmale während des Spurwechsels. Eine Einscherer-Szene aus dem Fahrsimulator wird in Abbildung 5 gezeigt. Der Prädiktionshorizont (Intervall zwischen $P(c_i|\mathbf{f}, \mathbf{u}) \geq 0.5$ und Überquerung der Fahrzeugmitte über die Spurmarkierung) beträgt etwa 1.3 s für das reale Fahrzeug und 0.8 s für den Spurwechsel im Fahrsimulator. Der Unterschied ist damit begründet, dass Spurwechsel im Simulator schneller vollzogen werden als im realen Verkehr. Obwohl der Klassifikator nur anhand von Echtdaten trainiert wurde, ist er ebenfalls in der Lage, die Spurwechsel im Simulator korrekt vorherzusagen. Diese Tatsache belegt die gute Generalisierung des modellbasierten Ansatzes.

5 Evaluierung

Zur Evaluierung des Prädiktionsmodells wird die Genauigkeit (engl. precision) und Trefferquote (engl. recall) für einen separaten Satz von Testdaten berechnet. Die Genauigkeit gibt den Quotienten aus den korrekt vorhergesagten Spurwechseln und allen vorhergesagten Spurwechseln an. Die Trefferquote ist der Quotient aus den korrekt vorhergesagten Spurwechseln und allen tatsächlich stattgefundenen Spurwechseln. Weiterhin wird der durchschnittliche Prädiktionshorizont, der ebenfalls eine wichtige Größe zur Bewertung des Prädiktionsmodells ist, angegeben. Dabei wird die Zeit zwischen dem Erreichen einer Spurwechselwahrscheinlichkeit von 50% und dem Überschreiten der Fahrbahnmarkierung durch den Fahrzeugmittelpunkt gemessen. Tabelle 5 zeigt die Werte für verschiedene Testdaten.

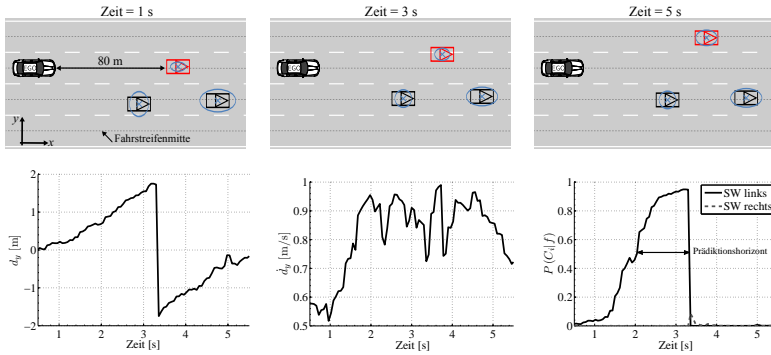


Abbildung 4: Spurwechselprediction während der Fahrt auf der Autobahn. Trotz weiter Entfernung und relativ hohen Unsicherheiten (blaue Ellipsen) funktioniert die Prädiktion robust.

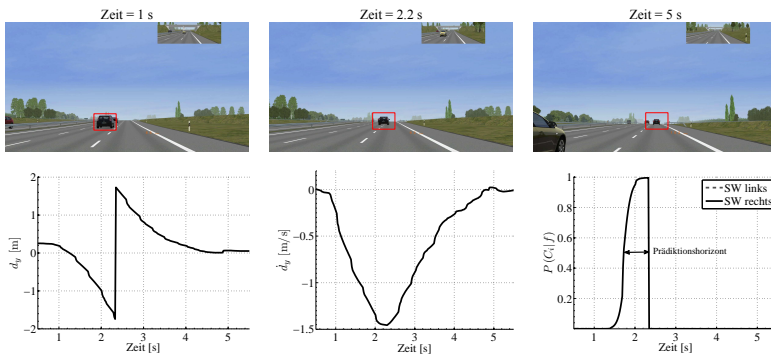


Abbildung 5: Spurwechselprediction während der Fahrt im Fahrsimulator.

	Genauigkeit	Trefferquote	Prädiktionshorizont
Simulator	100%	100%	0.8 s
Reale Fahrt	30%	100%	1.1 s

Tabelle 1: Evaluierung anhand von aufgenommenen Testdaten.

6 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Modell für die Spurwechselprediction auf der Autobahn vorgestellt, welches eine vorausschauende Fahrstrategie ermöglicht. Das generative Wahrscheinlichkeitsmodell erfüllt die Anforderungen eines transparenten, erweiterbaren und recheneffizienten Ansatzes. Die Funktionalität wurde sowohl im Fahrsimulator als auch in

Echtzeit auf der Autobahn getestet.

In Zukunft werden weitere Merkmale wie Blinker in dem Prädiktionsmodell integriert. Außerdem sollte zukünftig die Korrelation der Merkmale berücksichtigt und die Ausreißer unterdrückt werden. Als Weiteres sollte eine zeitliche Analyse in dem Modell integriert werden.

Literatur

- [1] D. Clarck, P. J. Ward, and J. Jones, "Overtaking road-accidents: differences in manoeuvre as a function of driver age," *Accident Analysis and Prevention*, vol. 30, pp. 455–467, 1998.
- [2] N. Kaempchen, M. Aeberhard, P. Waldmann, M. Ardel, and S. Rauch, "Der bmw nothalteassistent: Hochautomatisiertes fahren für mehr sicherheit im straßenverkehr," *Elektronik Automotive*, vol. 8/9, pp. 26–29, September 2011.
- [3] S. Rauch, M. Aeberhard, M. Ardel, and N. Kämpchen, "Autonomes fahren auf der autobahn - eine potenzialstudie für die zukünftige fahrerassistenzsysteme," in *Proc. of the 5th Tagung Fahrerassistenz*, 2012.
- [4] M. Benmimoun, A. Pütz, A. Zlocki, and L. Eckstein, "Effects of acc and few on speed, fuel consumption, and driving safety," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp. 1–6, 2012.
- [5] A. Lawitzky, D. Althoff, C. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive scene prediction for automotive applications," in *IEEE Intelligent Vehicles Symposium*, 2013.
- [6] P. Kumar, M. Perrollaz, S. Lefèvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *IEEE Intelligent Vehicles Symposium*, 2013.
- [7] T. Jebara, *Machine Learning: Discriminative and Generative*. Springer, 2004.
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

Situation Prediction And Reaction Control (SPARC)

(Patent Pending)

M. Ruf^{*†} J. R. Ziehn^{*‡§}

B. Rosenhahn[‡] J. Beyerer^{*†} D. Willersinn^{*} H. Gotzig[¶]

Abstract:

Approaches to automated driving typically hand over vehicle control to specialized modules for intersection handling, parking, obstacle avoidance etc., depending on the perceived traffic situation. This paper proposes a continuous-state alternative that allows to take all modeled goals and influences into account simultaneously, similarly to how a human driver would behave. A dynamic map of the environment is analyzed in real time for traffic rules and obstacles. The behavior of dynamic objects is predicted into the near future. This information is used to generate a 3D penalty map over space and time. An optimal trajectory is found based on these penalties as well as on penalties for internal control parameters. This holistic approach considers all relevant goals as well as the dynamic limits of the ego vehicle simultaneously when planning the trajectory, and requires no sharp state transitions during operation.

Keywords: Fully automated driving, prediction, situation analysis, trajectory planning

1 Overview

The concept presented here is based on [14] and is limited to the situation interpretation and maneuver planning / execution tasks of autonomous driving. It relies on processed sensor information: A dynamic map with objects detected around the ego vehicle, along with stochastic measures of uncertainty for all observations, is assumed to exist.

This paper introduces a two-step concept for autonomous driving, *situation prediction* and *reaction control*, short SPARC. Given the information in the dynamic map, the SPARC concept performs the following steps (cf. Fig. 1):

1. Use navigation instructions and static map to place the next waypoints for the ego vehicle (cf. Fig. 2(a, bottom)).

^{*}Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), 76131 Karlsruhe, {miriam.ruf, jens.ziehn, dieter.willersinn, juergen.beyerer}@iosb.fraunhofer.de

[†]Vision and Fusion Laboratory (IES), Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, miriam.ruf@kit.edu

[‡]{rosenhahn, ziehn}@tnt.uni-hannover.de, Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover, 30167 Hanover

[§]Corresponding author

[¶]heinrich.gotzig@valeo.com, Valeo Schalter und Sensoren GmbH

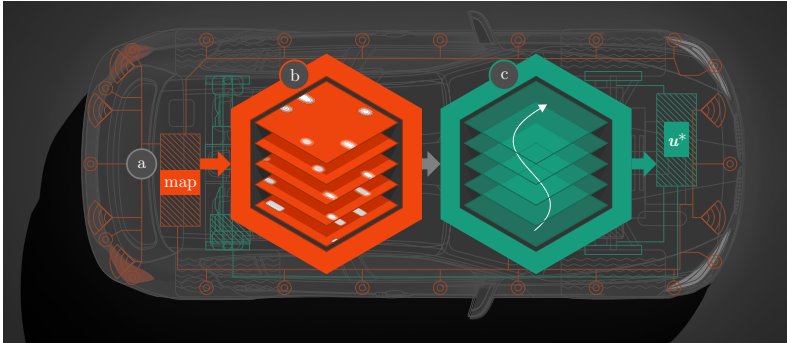


Figure 1: Outline of the proposed SPARC concept. The ego vehicle (a) sends a map of processed sensor information to the *situation prediction* block (b) that analyzes the scene and produces the holistic representation \mathbf{H} . The resulting *penalties* are passed to the *reaction control* block (c), which finds an optimal trajectory ξ^* and the corresponding control parameters \mathbf{u}^* . These are eventually fed back to the ego vehicle (a).

2. Derive information about static traffic rules (e.g. speed limits; not presented here) and assign *penalties* for their violation (cf. Fig. 2(a, top)).
3. Compute current (t_0) occupancy probabilities for spatial locations and assign penalties for their traversal (cf. Fig. 2(a, middle)).
4. Predict occupancies for $\{t_1, \dots, t_{\max}\}$ and assign penalties (cf. Fig. 2(b)).
5. Find an optimal trajectory that approximately connects the waypoints based on the hard constraint of physical feasibility and the soft constraint of minimizing the sum of the following penalties (cf. Fig. 2(c)):
 - collisions (higher penalties: more severe collisions or more likely collisions)
 - traffic rule violations (higher penalties: more severe violations)
 - dynamics (higher penalties: uneconomic or uncomfortable driving)
 - deviations (higher penalties: wide offset from waypoints)
6. Estimate control commands until $t_{\max} \approx 5$ sec, but recompute the trajectory much more frequently (≈ 10 Hz) and only pass the very next set of control commands to the ego vehicle. Thus, the planning up to t_{\max} is not intended to just bridge the gap between updates but to prevent decisions that will lead to ill states in the (relatively) far future. Furthermore, planning ahead can serve to bring the ego vehicle into a safe state upon total or partial sensor failure given the best last estimate.

The SPARC concept proposes a scalar or vector field over space and time, $\mathbf{H}(x, y, t)$, to store known and predicted information about the environment (the *holistic representation*,

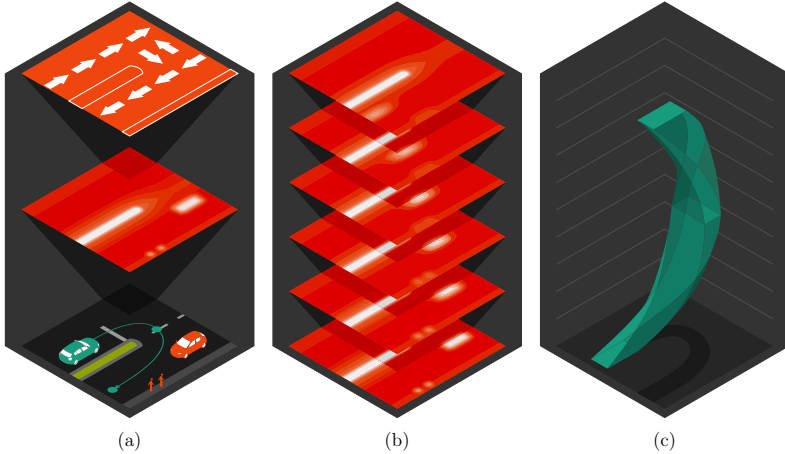


Figure 2: Elements of trajectory planning. (a) Bottom: Dynamic and static map of the scene, showing the ego vehicle (green) along with another car and two pedestrians (all red). The planned path ξ is indicated along with two waypoints. Middle: \mathbf{H} at current collision risks, taking into account uncertainties in measurements. Top: $\mathbf{H}(x, y, t_{\text{now}})$ at traffic rules (admissible flow directions). (b) $\mathbf{H}(x, y, t_{\text{now}} \dots t_{\text{max}})$ at prediction, showing the future development of the scene with growing uncertainties. (c) Planned trajectory ξ with ego vehicle footprint Φ wrapped around it to produce $\Phi \circ \xi$. The penalties contained within the green volume determine the expected detriment of the passage.

cf. Fig. 2). Given a functional that evaluates the total penalty $\mathcal{P}[\xi | \mathbf{H}]$ of a trajectory ξ given \mathbf{H} , the goal of finding the *optimal* trajectory ξ^* can be defined formally as

$$\xi^* = \arg \min_{\xi} \mathcal{P}[\xi | \mathbf{H}] \quad (1)$$

Practically ξ^* has to be approximated using numerical techniques, and the computations of \mathcal{P} in general and \mathbf{H} in particular have to trade off between accuracy and computation time. Approaches to both based on the current progress of development are presented here.

2 State of the Art and Motivation

The most well-known approaches to autonomous driving—prominently the award winners of the DARPA Urban Challenge 2007—feature a considerable level of discretization. Examples include the use of a limited set of velocity profiles for the ego vehicle [10], a limited set of states [8, 4] and the use of dedicated systems for particular driving tasks [10, 8, 13, 1] that hand over responsibility among themselves based on a finite situation classification [13, 8], as well as a separation between path planning and vehicle control

[10, 1, 7, 2]. Another common case of discretization is to limit the behavior of the ego vehicle to a finite set of path candidates [10, 8, 7, 12].

The aim of the SPARC concept is to propose an alternative that requires no explicit classification of the current driving state, and no discrete state transitions for the ego vehicle. The intended benefit of this approach is to take all goals into account simultaneously, from basic navigation over traffic rules and collision avoidance down to currently desirable control parameters. The number of influences is intentionally kept flexible by reducing every contributing aspect to a unified *penalty* model, that is conciliated by the common language of stochastic modeling. The internal software state of the ego vehicle can in theory be kept constant between regular travel, unusual situations and even emergencies, and therefore no high-level understanding of the situation is required—neither are explicit state transitions that would have to be triggered by a decision module. The purpose of this paper is to shed light on the possibilities and feasibility of this alternative rather than to conclude its superiority over discretized approaches.

3 Penalties in a Holistic Model

The SPARC concept uses a holistic *penalty* concept that aims to reconcile all factors that are meant to influence the behavior of the ego vehicle. These factors involve legal constraints, safety, comfort, efficiency and ecology. Some of these are subject to uncertainty while others apply deterministically. All soft constraints are reduced to such penalties, while hard constraints are reserved exclusively for physically impossible states. Penalties are chosen such that they are real numbers, where higher penalties denote less desirable states, and such that their sum or integral is a meaningful quantity as well.

Penalties can generally be divided into two distinct classes of certainty (cf. Tab. 1): Those involving uncertainty (which will be discussed in Section 3.1) require a measure of probability, which can be approximated objectively by a prediction process. Those that apply deterministically (such as speed limits or the loss of comfort due to a sharp braking maneuver) can be evaluated directly. Both however require a definition of how undesirable a certain state is. This definition effectively determines how the ego vehicle will trade off between (e.g.) comfort and collision avoidance. Such priorities are implicitly inherent to any human driver, but they lie in everybody’s personal responsibility. Defining a general set of rules is not the scope of this paper—the ideas will be outlined on an (obviously deficient) exemplary definition.

Penalties can further be divided into distinct classes of influence (cf. Tab. 1): Those affected by the current situation and its possible development (as represented in \mathbf{H}) are called *outer penalties*. We further distinguish between *primary outer penalties* $\mathcal{P}_{\text{out}}^I[\xi | \mathbf{H}]$, which are based on collision risks, and *secondary outer penalties* $\mathcal{P}_{\text{out}}^{II}[\xi | \mathbf{H}]$, which represent localizable traffic rules, such as speed limits, or non-vital interactions, such as potholes or speed bumps. These penalties are often also uncertain because they rely on sensor measurements (as considered in step 3 of the SPARC concept) and possibly the prediction (step 4). Penalties that only relate to state transitions of the ego vehicle itself are called *inner penalties* $\mathcal{P}_{\text{in}}[\xi]$. Examples for these would be uncomfortably strong lateral or longitudinal accelerations or high fuel consumption. Minimization towards the optimal trajectory ξ^* will need to take into account all of these terms, although their influence in the trajectory calculation should clearly vary.

		deterministic	probabilistic
inner		fuel consumption	loss of friction
outer	primary	collision (e.g. trees)	collision (e.g. cars)
	secondary	traffic rule violation	collision (e.g. wild animals)

Table 1: Types of penalties along with an example for each combination.

$$\mathcal{P}[\boldsymbol{\xi} | \mathbf{H}] = \mathcal{P}_{\text{in}}[\boldsymbol{\xi}] + \mathcal{P}_{\text{out}}^{\text{I}}[\boldsymbol{\xi} | \mathbf{H}] + \mathcal{P}_{\text{out}}^{\text{II}}[\boldsymbol{\xi} | \mathbf{H}] \quad (2)$$

The *holistic representation* \mathbf{H} is, in its most simple form, a mapping from space and time to scalar-valued, positive penalties

$$\mathbf{H} : X \times Y \times T \rightarrow \mathbb{R}_{\geq 0} \quad (3)$$

where X and Y are bounded intervals of 2D space around the ego vehicle (which is located at $\mathbf{x}_0 = [x_0, y_0]^T$), and T is an interval from t_0 (“now”) up to several seconds into the future. The value $\mathbf{H}(x, y, t)$ denotes the penalty of traversing the location $[x, y]^T = \mathbf{x}$ at time t .

3.1 Expected Values for Penalties

To compute the penalty for a possible collision, expected values can be used. The penalty for traversing a point \mathbf{x} where an obstacle is certain to be found depends only on the severity of the collision. Defining a severity penalty \mathcal{P}_{sev} is an ethical, legal, economical and physical question and, for this very reason, far beyond the scope of this paper. To convey an intuition, the severity penalty is set to the kinetic energy of the impact for the examples in this paper; clearly this definition hardly satisfies the practical requirements and should not be taken as a proposal. The kinetic energy W_{kin} that one object of mass m and velocity \mathbf{v} exerts on the ego vehicle at time t_{coll} is

$$W_{\text{kin}} = m \|\dot{\boldsymbol{\xi}}(t_{\text{coll}}) - \mathbf{v}_{\text{obj}}\|_2^2 =: \mathcal{P}_{\text{sev}}. \quad (4)$$

While its exclusive use has obvious shortcomings, having the kinetic energy contribute to the penalty is justified as the kinetic energy of the impact can be turned into deformation energy acting on the participants, and is thus a measure of damage inflicted. A more sophisticated discussion on the effect of kinetic energy on traffic accidents can be found in [9].

If the obstacle is not known to overlap with $\boldsymbol{\xi}(t_{\text{coll}})$ with certainty, then its occupancy probability at this place and time must influence the traversal penalty at $\boldsymbol{\xi}(t_{\text{coll}})$. The *expected penalty* for this particular collision c with probability $p(c)$ is $p(c) \cdot \mathcal{P}_{\text{sev}}(c)$. Here $p(c)$ is equal to the probability of the obstacle occupying $\boldsymbol{\xi}(t_{\text{coll}})$, since the traversal of the ego vehicle is the conditional premise (i.e. taken to be certain) for this evaluation. $\mathcal{P}_{\text{sev}}(c)$ needs to be approximated from the ego vehicle trajectory and the predicted target trajectory (as well as its mass). If a set of several collisions $C(\boldsymbol{\xi}(t_{\text{coll}}))$ may occur at $\boldsymbol{\xi}(t_{\text{coll}})$, these potential collisions can (as a gross simplification) be regarded as independent. Through this, the total expected penalty of traversing $\boldsymbol{\xi}(t_{\text{coll}})$ can be described as

$$E[\mathcal{P}_{\text{sev}} | \boldsymbol{\xi}(t_{\text{coll}})] = \sum_{c \in C(\boldsymbol{\xi}(t_{\text{coll}}))} p(c) \cdot \mathcal{P}_{\text{sev}}(c). \quad (5)$$

Even though the assumption of stochastic independence involves the possibility of two obstacles occupying the same position, it greatly simplifies the computation, because $E[\mathcal{P}_{\text{sev}} | \boldsymbol{\xi}(t_{\text{coll}})]$ can be computed as an independent sum over all observed obstacles and their trajectories.

3.2 Prediction of Occupancy Probabilities

The prediction of the traffic development in the near future is step 4 in the proposed SPARC concept. To determine the probability $p(c)$ of a collision at a given location (or equivalently the probability of an obstacle o occupying a position \boldsymbol{x} at time t , $p_o(\boldsymbol{x}, t)$), requires a prediction process. Presented here is the prediction process for observed cars. The prediction assumes the cars to follow legal tracks (as opposed to driving in arbitrary, illegal patterns), but can represent uncertainty about speed changes and forks in their tracks, as well as inter-vehicle dependencies concerning their behavior (both not presented here).

3.2.1 Speed changes and the Kumaraswamy distribution

To efficiently represent arbitrarily fine speed changes, speed distributions are used. Given a statistical distribution of speeds e.g. on a particular street or in inner-city scenarios in general, a car can be modeled to pick a speed at random from this distribution. The resulting distribution of uncertainties in speeds can be integrated into a distribution of uncertainties in positions. While the model corresponds to the vehicle picking the new speed at t_0 and keeping it constant from then on, the uncertainty in its initial choice propagates through time and thus resembles the (more accurate but also much more costly) use of a stochastic differential equation (cf. Fig. 3 (a)).

The model for speed distributions proposed in this paper is the Kumaraswamy distribution (introduced in [6]), whose probability density function (PDF) p_K and cumulative distribution function (CDF) P_K are given by

$$p_K(x; a, b) = abx^{a-1}(1-x)^{b-1} \quad (6)$$

$$P_K(x; a, b) = 1 - (1-x^a)^b \quad (7)$$

It is similar to the Beta distribution (see [3, p. 137]) in that it has two shape parameters, a and b (which are useful to represent varying speed distributions), and is defined on a double-bounded interval (which is useful to represent physical limits of speeds and to narrow down the prediction range). The usual domain for the Kumaraswamy distribution as given above is $[0, 1]$, but the distribution functions can easily be scaled to represent a suitable range of speeds.

Given a cumulative speed distribution $P(v_{\text{upper}})$ which represents the probability of the speed lying between $-\infty$ and v_{upper} , the cumulative position distribution (for a position s_{upper} along a given track) at time t can be computed by

$$P_t(s_{\text{upper}}) = P\left(\frac{s_{\text{upper}}}{t}\right). \quad (8)$$

The resulting density profile of positions is shown in Fig. 3, along with a possible interpretation of these densities as a trajectory density. Therefore speed changes are accounted

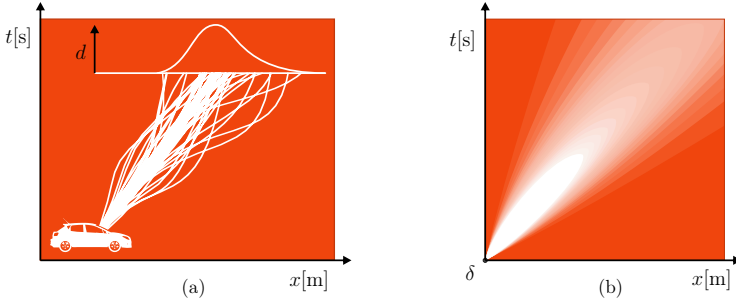


Figure 3: (a) Stochastic differential equation showing several possible trajectories for a car. (b) Approximation of the positional probability density by the Kumaraswamy distribution.

for in the prediction, even though an explicit stochastic differential equation is not set up. The occupancy probabilities along the track can thus be computed from a series of multiplications and sums given integral shape parameters to the Kumaraswamy distribution.

Setting the shape parameters depending on the state of the vehicles is beyond the scope of this paper; however, Fig. 5 shows the exemplary fitting of the Kumaraswamy distribution to German highway statistics, as a demonstration of its applicability. The data was taken from [5], as few applicable statistics are publicly available. In this publication the data was gathered by the German Federal Highway Research Institute (BAST) on German motorways (Autobahnen) in 1995, where no speed limits were given. Therefore the data is grouped by the expected speed at the respective locations of measurement (the three groups are shown in Fig. 5). Each cumulative distribution of measurements $M(v)$ was measured for $v \in V := \{80 \text{ km h}^{-1}, 90 \text{ km h}^{-1}, \dots, 180 \text{ km h}^{-1}\}$. Using linear least squares, a Kumaraswamy distribution CDF (with free parameters a and b and a domain of $[0 \text{ km h}^{-1}, 180 \text{ km h}^{-1}]$) was fitted to each of the three groups to satisfy:

$$P_K^* = \arg \min_{P_K} \sum_{v \in V} |P_K(v) - M(v)|^2 \quad (9)$$

The results are shown in Fig. 5 to demonstrate the applicability of the Kumaraswamy distribution for representing various speed distributions at a low computational effort.

4 Reaction Control

The 5th step of the proposed SPARC concept is the reaction control (RC) (cf. Fig. 1), which generates the trajectory based on the input of the SP block according to Eq. (1). The optimization is constrained by physical limits of the ego vehicle, but no further constraints are imposed. In particular it is not advised to introduce hard constraints to avoid collisions since this would limit the options in trading off between unavoidable collisions in extreme situations.

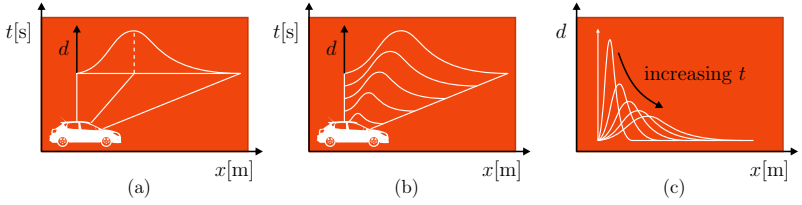


Figure 4: Speed distributions: (a) Connection between minimum speed, maximum speed and expected speed. (b) Family of probability distributions (density not to scale) by time. (c) Family of probability distributions (here: density to scale) converging to δ for $t \rightarrow 0$ (the leftmost “function” represented by an upward-pointing arrow).

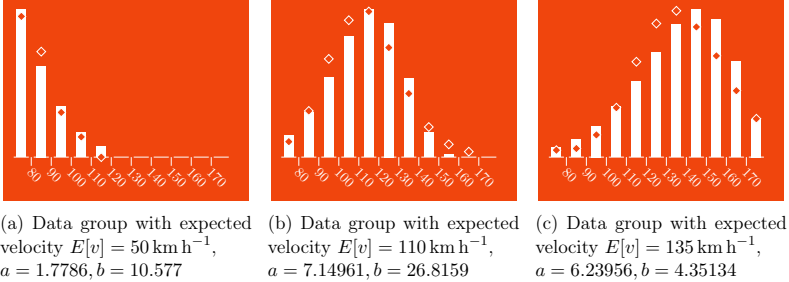


Figure 5: Kumaraswamy distribution PDFs (bars) mapped to actual speed distributions (ticks) measured in [5] by linear least squares over the corresponding CDFs.

The results presented in Section 5 were produced using a variational approach. In this, the Euler–Lagrange equation (see [11, p. 33], [14, p. 48])

$$\frac{\partial \mathcal{P}[\xi | \mathbf{H}]}{\partial \xi} - \frac{d}{dt} \frac{\partial \mathcal{P}[\xi | \mathbf{H}]}{\partial \dot{\xi}} + \left(\frac{d}{dt} \right)^2 \frac{\partial \mathcal{P}[\xi | \mathbf{H}]}{\partial \ddot{\xi}} \equiv \mathbf{0} \quad (10)$$

is used for a gradient descent from an initial $\xi^{(0)}$ which satisfies the following properties:

- The spatial components follow the path $\xi^{(0)}$ which avoids permanent obstacles (such as walls or road limits) but intentionally does not take dynamic or unexpected obstacles (such as the shopping cart in Sec. 5 / Fig. 7) into consideration. It thus represents the characteristics of an offline map.
- The initial speed profile is the linear speed interpolation between the current speed $\dot{\xi}(t_b)$ and target speed $\dot{\xi}(t_e)$.

5 Practical Results

The methods presented here were evaluated in CarMaker, a commercial simulation software by IPG for driver assistance systems and vehicle dynamics. The trajectory optimization was performed in MATLAB. The dynamic traffic situations were modeled based on video footage, the static traffic situations were composed manually following requirements by Valeo.

Figure 6 shows the dynamic case of an intersection example. The ego vehicle is entering an intersection to take a left turn. Two cars (c_1 and c_2) are approaching the intersection on the opposite lane. They have the right of way, but the ego vehicle can decide to pass before or after them, or even in between, given that the spacing between c_1 and c_2 is adequately wide. The methods proposed herein allow to gauge the risks involved in such a maneuver and execute it if they are considered sufficiently low.

The uncertainty of c_1 and c_2 spreads visibly along the track in a “cloud” shape, due to uncertainties in future speed. As there is only one legal lane for both cars (their current lane), certainty about their lateral position is high. c_2 has a right turn indicator set and is thus predicted to turn right, as indicated by its cloud. c_1 is predicted to pass straight through the intersection. If the planning space for the ego vehicle is limited to its legal path, only the progress parameter s along the path needs to be optimized. The 3D planning problem ($x \times y \times t$) turns into a 2D planning problem ($s \times t$), indicated in Fig. 6(b) and used in Figs. 6(c–e). The relevant extracts of \mathbf{H} (cut out along the given path) are shown (along with arrows denoting the most likely paths for c_1 and c_2 up to the point of intercept).

Figures 6(c–e) show the effect of spacing between c_1 and c_2 : If the spacing is wide enough (Fig. 6(c)) the ego vehicle plans to pass between them. If the spacing is too narrow, the best solution is to wait until both cars have passed (Fig. 6(d)). The solutions are compared in Fig. 6(e). Everything shown in Fig. 6 represents a single time step in the real world. The very next control commands to follow the calculated trajectory are executed and the process steps are repeated (one quarter of a second later in our simulation). In the examples presented here, the ego vehicle passed collision-free through the oncoming traffic.

To contrast the dynamic case, Fig. 7 shows a static example typical of a simple valet parking application: A static obstacle (here: a shopping cart) needs to be avoided. For this, the trajectory is replanned along the two spatial coordinates. As the example is static, prediction is not a relevant influence. The spatial planning is constrained by the minimum turn radius of the ego vehicle. The simulation results in a collision-free passage around the shopping cart. If the walls (represented by traffic cones) are too narrow for the ego vehicle to drive around the shopping cart, the ego vehicle stops before it.

6 Conclusions and Outlook

The approach presented in this paper is currently in an early simulation stage, since several process steps need to be triggered manually. Therefore the amount of simulation examples is very limited. However these examples do suggest that it is possible to use the given methods for planning trajectories in autonomous driving which take the environment, traffic rules and vehicle dynamics into account at the same time. Work towards a more

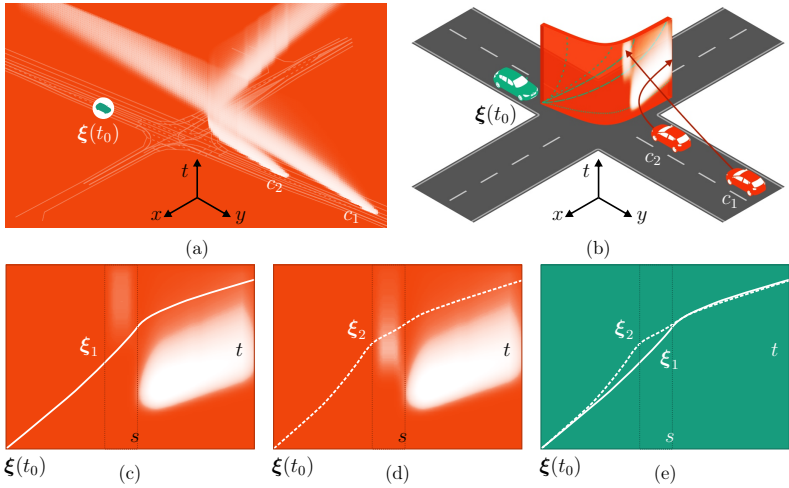


Figure 6: Example of the SPARC concept applied to an intersection, where the ego vehicle (green) located at $\xi(t_0)$ wants to take a left turn through oncoming traffic (red), consisting of two cars c_1 and c_2 . (a) Resulting energy cube (isometric perspective; vertical axis: t). (b) Schematic view (not to scale with (a)), cut-out of \mathbf{H} along a given path and arrows (dark red) indicating most likely trajectories of c_1 and c_2 . (c) Cut-out of \mathbf{H} for wide gap between c_1 and c_2 . (d) The same for a narrow gap. (e) Comparison of trajectories obtained in (c) and (d). In (c–e) the s boundaries of the opposite lane are indicated by the dashed vertical lines, to highlight the transition interval.

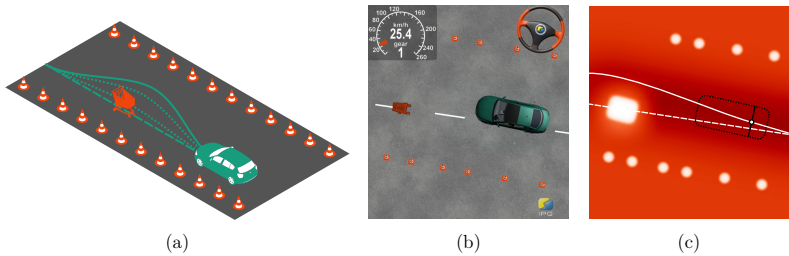


Figure 7: Example of the SPARC concept applied to a valet parking example, where the ego vehicle (green) needs to avoid a shopping cart (red). (a) Schematic view (not to scale) of original path (dashed), intermediate iteration results (dotted) and optimal trajectory (solid). (b) Situation as simulated in CarMaker. (c) Situation with initial trajectory (dashed) and optimal trajectory (solid) as obtained in MATLAB. The ego vehicle and its rear axis are shown in black.

extensive automation is underway that will allow for a more systematic evaluation in various traffic situations. The same applies to parameter estimation for the Kumaraswamy distribution.

The methods were chosen with a real-time implementation in mind. In particular, methods were preferred that confine themselves to multiplication and addition, as featured in dedicated hardware such as multiply-accumulate blocks, as well as methods that exhibit a high degree of parallelity to be exploited in hardware implementations, while retaining flexibility and expressiveness. As currently little optimization was performed, reliable figures on the computational effort cannot be given yet and remain to be produced. In particular, a hardware implementation has to be specified.

The concept remains valid in theory for any level of detail in a dynamic map—however, its usefulness is questionable with highly uncertain data (e.g. just point clouds of obstacles, no interpretation) or highly certain data (mostly fully automated vehicles communicating state and intentions via Car2Car). A more sophisticated assessment of applicability has yet to be developed.

Further techniques, such as Non-Linear Model Predictive Control (NMPC), as well as graphical models for prediction and trajectory planning, are currently under evaluation but lie beyond the scope of this paper. All approaches have parameters that depend on statistical data and/or factual data such as vehicle limits and traffic rules. Determining methods to set these parameters adequately will be mandatory to apply the techniques to real-world situations.

Acknowledgements

This work was partially supported by Valeo Schalter und Sensoren GmbH within the V50 project, and by the Fraunhofer-Gesellschaft along with the state of Baden-Württemberg within their joint innovation cluster REM 2030.

References

- [1] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, S. Cacciola, P. Currier, A. Dalton, J. Farmer, J. Hurdus, S. Kimmel, P. King, A. Taylor, D. V. Covern, and M. Webster. Odin: Team VictorTango's entry in the DARPA Urban Challenge. *Journal of Field Robotics*, 25(8):467–492, 2008.
- [2] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derenick, J. Spletzer, and B. Satterfield. Little Ben: The Ben Franklin Racing Team's Entry in the 2007 DARPA Urban Challenge, 2008.
- [3] M. G. Bulmer. *Principles of Statistics*. Dover Publications, 2nd edition, 1979.
- [4] Y.-L. Chen, V. Sundareswaran, C. Anderson, A. Broggi, P. Grisleri, P. Porta, P. Zani, and J. Beck. TerraMax: Team Oshkosh Urban Robot. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 595–622. Springer Berlin Heidelberg, 2009.

- [5] G. Kellerman. Geschwindigkeitsverhalten im Autobahnnetz 1992. *Straße + Autobahn*, 05 1995.
- [6] P. Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2):79-88, March 1980.
- [7] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. S. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams. A Perception Driven Autonomous Urban Robot. *International Journal of Field Robotics*, 25(10):727-774, 2008.
- [8] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandoski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The Stanford Entry in the Urban Challenge. *Journal of Field Robotics*, 2008.
- [9] A. Sobhani, W. Young, D. Logan, and S. Bahrololoom. A kinetic energy model of two-vehicle crash injury severity. *Accident Analysis & Prevention*, 43(3):741 - 754, 2011.
- [10] C. Urmson, J. Anhalt, H. Bae, J. A. D. Bagnell, C. R. Baker, R. E. Bittner, T. Brown, M. N. Clark, M. Darms, D. Demitrish, J. M. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. M. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y.-W. Seo, S. Singh, J. M. Snider, J. C. Struble, A. T. Stentz, M. Taylor, W. R. L. Whittaker, Z. Wolkowicki, W. Zhang, and J. Ziglar. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8):425-466, June 2008.
- [11] B. Van Brunt. *The Calculus of Variations*. Springer, 2010.
- [12] M. Wang, T. Ganjineh, and R. Rojas. Action annotated trajectory generation for autonomous maneuvers on structured road networks. In *5th International Conference on Automation, Robotics and Applications (ICARA)*, 2011, pages 67-72, 2011.
- [13] J. M. Wille, F. Saust, and M. Maurer. Segmentübergreifende Bahnplanung mittels eines analytischen Optimierungsverfahrens für die autonome Fahrzeugführung auf dem Braunschweiger Stadtring. In *6. Workshop Fahrerassistenzsysteme - FAS 2009*, 2009.
- [14] J. R. Ziehn. Energy-based collision avoidance for autonomous vehicles. Master's thesis, Leibniz Universität Hannover, Germany, October 16th 2012.

Potenzial der Kopfposenerkennung zur Absichtsvorhersage von Fußgängern im urbanen Verkehr

Horst Kloeden* Nina Brouwer[†] Stefan Ries[‡] Ralph H. Rasshofer[§]

Zusammenfassung: Diese Arbeit analysiert das Potenzial der Kopfposenerkennung zur Vorhersage der Bewegungsintention eines Fußgängers. Ausgehend von Unfallstatistiken werden Szenarien abgeleitet, in denen die Kopfpose einen frühzeitigen Indikator für die Änderung des Bewegungszustandes beschreibt. Dabei wird exemplarisch gezeigt, dass die Kopfposition eine statistische Differenzierung zwischen querenden und nicht querenden Fußgängern erlaubt. Neben den statistischen Betrachtungen wird ein Modell zur Intentionserkennung mit Hilfe eines Mixture of Gaussians Hidden Markov Modells (MoG-HMM) vorgestellt und anhand von Beispielszenarien analysiert. Die Ergebnisse werden hinsichtlich des Potenzials zur Bewegungsvorhersage bewertet und dienen der Anforderungsanalyse an die Kopfposenerkennung auf Basis eines hochauflösenden Kamerasystems.

Schlüsselwörter: Fußgängerschutz, Intentionserkennung, Kopfposition, MoG-HMM

1 Einleitung

In zukünftigen Fußgängerschutzsystemen wird die Umsetzung eines früheren Warnzeitpunktes oder die Realisierung einer früheren Aufmerksamkeitssteuerung des Fahrers angestrebt. Hierfür ist zum einen eine genaue Erfassung des Fahrzeugumfelds erforderlich, zum anderen ist auf Basis dieser Informationen eine Kritikalität möglicher Konflikte zu bestimmen. Die ermittelte Kritikalität dient zur Auslösung von Warnungen oder unterschwelligen Maßnahmen zur Aufmerksamkeitssteuerung des Fahrers.

Die Genauigkeit der bestimmten Kollisionswahrscheinlichkeit wird dabei von der Präzision der Perzeption und von der Prädiktionsgüte des Modells bestimmt. Je größer der Prädiktionszeitraum gewählt wird, desto geringer ist der Einfluss der momentanen Position und Geschwindigkeit eines Fußgängers, die mit Hilfe der fahrzeuglokalen Sensoren bestimmt werden. Für die Realisierung einer früheren Prädiktion ist daher die Prädiktionsgüte des eingesetzten Fußgängermodells als limitierender Faktor zu betrachten.

Die bisher für den Fußgängerschutz entwickelten und eingesetzten Modelle lassen sich nach [1] in die Kategorien „ad-hoc“ (binäre Bewertung der Kollisionswahrscheinlichkeit,

*Horst Kloeden: BMW Group Forschung und Technik, horst.kloeden@bmw.de.

[†]Nina Brouwer: BMW Group Forschung und Technik, nina.brouwer@bmw.de

[‡]Stefan Ries: BMW Group Forschung und Technik, stefan.ries@bmw.de

[§]Ralph H. Rasshofer: BMW Group Forschung und Technik, ralph.rasshofer@bmw.de

TTC-Ansätze), geometrisch [2], stochastisch [6] und physikalisch unterscheiden. Grundlage der Modelle ist typischerweise die Nutzung von physikalischen Zusammenhängen der momentanen und zukünftigen kinematischen Größen, wie Position, Geschwindigkeit oder Beschleunigung [3]. Ebenfalls sind Modelle bekannt, die die physikalischen Zusammenhänge mit physiologischen Grenzen kombinieren [5, 1, 4].

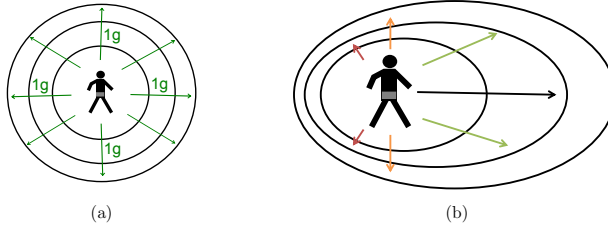


Abbildung 1: Physikalisches Fußgängermodell mit einer angenommen maximalen Beschleunigung von „1g“ in alle Richtungen (a) und das physiologische Bewegungsmodell von Scherf (b)

Die typische Prädiktionsdauer der angewendeten Modelle liegt in einer Größenordnung von 0,5 bis 1,5 Sekunden. Für eine Prädiktionsdauer von über 2 Sekunden ist dagegen keine ausreichende Unterscheidung von kritischen und unkritischen Fußgängern möglich. Damit eignen sich die Bewegungsmodelle nicht zur Umsetzung einer frühzeitigen Warnung oder einer unterschweligen Aufmerksamkeitssteuerung des Fahrers.

Der Mensch ist hingegen auch über einen längeren Zeithorizont in der Lage, kritische Fußgänger zu identifizieren und ihren Bewegungsspielraum realistisch einzuschätzen. Hierfür wird durch den Mensch nicht nur die Position und Geschwindigkeit einer Person betrachtet, sondern zusätzlich deren Bewegung analysiert und in Kontext der umgebenden Infrastruktur und anderen Objekten gesetzt. Dabei wird durch den Mensch unter anderem der Verlauf der Kopfpositionen betrachtet und daraus eine mögliche Verhaltensabsicht antizipiert. Diese Tatsache verdeutlicht das Potenzial zur weiteren Optimierung zukünftiger Fußgängerschutzsysteme.

In dieser Veröffentlichung wird das Potenzial der Kopfposenerkennung als Zusatzindikator für Fußgängerschutzsysteme analysiert. Hierfür wird zunächst ein hochauflösendes Bildverarbeitungssystem vorgestellt, das die Kopfposenerkennung von Fußgängern im innerstädtischen Bereich ermöglicht und damit die Grundlage für die Nutzung des Indikators bildet. Anschließend erfolgt der Nachweis des Potenzials der Kopfposendetektion anhand von Unfallstatistiken in Kombination mit einer Probandenstudie, in der die Zusammenhänge zwischen Kopfposenbewegung, Position, Geschwindigkeit und Bewegungsintention experimentell nachgewiesen werden.

Im Gegensatz zu kinematischen Größen, wie Position oder Geschwindigkeit, gibt es jedoch keine physikalische Interpretation der Kopfposenbewegung bezüglich der Aufenthaltswahrscheinlichkeit eines Fußgängers. Daher wird dieser Zusammenhang mit Hilfe eines Markov Modells beschrieben, anhand von realen Beispielszenarien bewertet und Anforderungen an das Perzeptionssystem abgeleitet.

2 Kopfposenerkennung

Die Kopf- oder Gesichtserkennung ist insbesondere in der Sicherheitstechnik bereits weit verbreitet. Grundlage bildet typischerweise ein Klassifikator zur Kopfdetektion, z.B. auf Basis einer *Support Vector Machine* (SVM) mit *Histogram of Oriented Gradients* (HOG) Merkmalen. Mit Hilfe des gleichen Prinzips wird es ermöglicht einen nachgelagerten Klassifikator auf die Erkennung der Kopfpose zu trainieren. Der genutzte Klassifikator nutzt hierfür vier verschiedene Grundausrichtungen und interpoliert die Ausrichtung auf Basis der Klassenwahrscheinlichkeiten. Abbildung 2 zeigt das Ergebnis eines solchen Klassifikators.

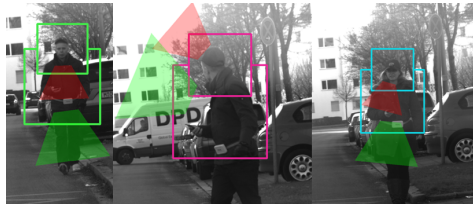


Abbildung 2: Fußgängerdetektion mit erkannter Ausrichtung des Kopfes und des Oberkörpers.

Die Kopfposenerkennung stellt dabei auch neue Anforderungen an die Sensorik. Ist für die Klassifikation der Kopfrichtung z.B. eine minimale Kopfgröße von 25 px erforderlich und hat die Kamera einen Öffnungswinkel von 40° , so ist eine minimale Auflösung von 12 Mpx erforderlich, um eine Reichweite von 40 m zu erreichen. Diese Entfernung entspricht einem „4-Sekunden-Abstand“ bei einer Geschwindigkeit von 36 km/h in einem Wohngebiet.

3 Analyse des Nutzenpotenzials der Kopfposenerkennung

Die Betrachtung aktueller Unfalldaten zeigt, dass die Mehrheit von 74,3 % aller Fußgängerunfälle durch das Überqueren der Straße bei gleichzeitiger Geradeausfahrt des Fahrzeugs entstehen.

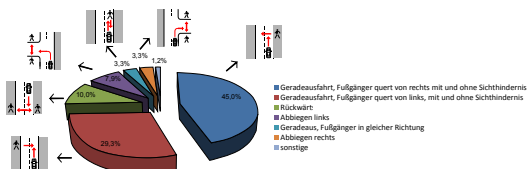


Abbildung 3: Typische Unfallsituationen aus GIDAS ([7])

Abbiegeunfälle, die häufig auf Grund der späten Sichtbarkeit des Fußgängers entstehen, machen dagegen nur einen geringeren Anteil von 11,2% (3). Die Ergebnisse stimmen mit denen des Statistischen Bundesamts überein, die außerdem die überwiegende Relevanz von Innenstadtszenarien zeigen (vgl. Abbildung 4a). Aus der GIDAS¹-Datenbank lässt sich ebenfalls ableiten, dass 88% aller Unfälle eine Ausgangsgeschwindigkeit des Fahrzeugs von weniger als 55 km/h aufweisen (vgl. Abbildung 4b). Zukünftige Fußgängerschutzsysteme mit einem hohen Nutzenpotenzial müssen daher besonders diese Situationen adressieren. Bei einer typischen Geschwindigkeit von 50 km/h und einer Verzögerung von $9,81 \text{ m/s}^2$ ergibt sich ein Bremsweg von 14 m und eine Bremszeit von 2,4 s (inkl. Reaktionszeit von 1 s). Diese Prädiktionszeit kann von bisherigen Fußgängermodellen, selbst bei frühzeitiger Sichtbarkeit des Fußgängers, nicht erreicht werden. Die Erweiterung des Vorhersagehorizonts bedarf daher der Beschreibung einer Person als willensgesteuertes Objekt, das aufgrund seiner Verhaltensabsicht ein mittel- oder längerfristiges Ziel verfolgt. Die Herausforderung besteht dabei in der Erkennung der aktuellen Verhaltensabsicht.

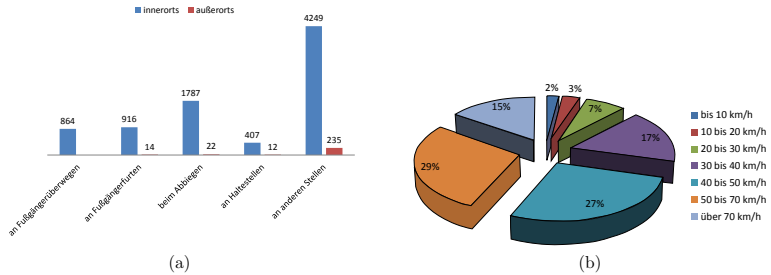


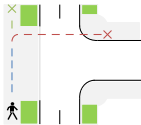
Abbildung 4: Prozentualer Anteil der Fahrzeuggeschwindigkeiten aus GIDAS (ca. 5 Sekunden vor der Kollision) (a) und Fehlverhalten des Fahrers, welches zu Fußgängerunfällen führt [8] (b).

Eine Studie von Färber [9] zeigt beispielsweise, dass der Mensch nicht allein anhand der Trajektorie die Querungsabsicht eines Fußgängers erkennen kann, sondern weitere Informationen über Kopfposition- oder Beinposition benötigt. Auch bei einer von Färber in der gleichen Studie vorgestellten Untersuchung, woran die Teilnehmer die Intention des Fußgängers erkennen, zeigt sich, dass die Kopfposition in 28% der Fälle der ausschlaggebende Indikator war. In 10% der Fälle spielte die Beinbewegung eine Rolle. Weitere Studien beschäftigen sich mit dem typischen Verhalten von Fußgängern an Fußgängerüberwegen und dem Sicherungsverhalten vor dem Queren. [10] nennt dabei Zeiten von drei Sekunden zwischen dem Beginn des Sicherungsverhaltens und dem Betreten der Fahrbahn. [11] zeigt weiterhin, dass beim Queren am Fußgängerüberweg nur circa 1% der Personen weder ein Abbremsen noch ein Sicherungsverhalten zeigen. Wird von diesen Ergebnissen ausgehend angenommen, dass eine Querungsabsicht mit einem Sicherungsverhalten einhergeht, kann dies als Grundlage für Untersuchungen zur Intentionserkennung verwendet werden.

Auf Grundlage der Unfallstatistik und den Erkenntnissen über die menschliche Per-

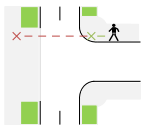
¹German In Depth Accident Study

zeption wurde eine Auswahl von zwei relevanten Unfallszenarien getroffen. In beiden Szenarien ermöglicht die Betrachtung der kinematischen Größen erst einen sehr späten Warnzeitpunkt eines Fußgängerschutzsystems. Gleichzeitig besteht eine hohe Gefahr der Falschauslösung.



(a) Szenario 1

Szenario 1: Der Fußgänger bewegt sich parallel zur Straße und ändert seine Richtung erst bei der nächstgelegenen Möglichkeit die Straße zu überqueren.



(b) Szenario 2

Szenario 2: Der Fußgänger ist im Begriff die Straße zu überqueren und bewegt sich auf die Fahrbahn zu. Lediglich das Sicherungsverhalten lässt auf ein mögliches Stehenbleiben schließen.

Abbildung 5: Beschreibung der Querungsszenarien

3.1 Datenerhebung

Die Untersuchung des Zusammenhangs der Fußgängerintention und der Kopfpose erfolgt mit Hilfe von experimentell erhobenen Daten auf einer zentimetergenau kartierten Kreuzung. Die Daten dienen der Analyse der Kopfposition verschiedener Probanden in den abgeleiteten Szenarien. Weiterhin wird untersucht, ob die Kopfposition in diesen Szenarien als frühzeitiger Indikator zur Bestimmung der Fußgängerintention geeignet ist und noch vor der Veränderung des kinematischen Bewegungszustands eine Aussage über das zukünftige Verhalten erlaubt.

Die Generierung der Testdaten erfolgte jeweils durch einen Referenzsensor (IMU¹) am Kopf und am Torso des Probanden sowie durch einen A-GPS-Empfänger. Weiterhin verfügte der Proband über eine Eingabeeinheit zur Signalisierung seiner Querungsabsicht, die nachfolgend als Ground Truth verwendet wird.

3.2 Statistische Datenauswertung

Zunächst werden die Daten aus Szenario 1 im Hinblick auf den Unterschied zwischen Geradeausgehen ohne Queren und Abbiegen mit Queren untersucht. Im letzten Fall zeigt sich dabei eine Erhöhung der Kopfpositionsänderungen (vgl. Abbildung 7). Eine biserial Korrelationsanalyse² und ein t-Test³ (Zweistichprobentest, rechtsseitig) zeigen, dass dieser

¹Inertial Measurement Unit der Firma Ximu

²Zusammenhang zwischen einer binären und einer kontinuierlichen Variablen

³Statistischer Test zweier Stichproben auf Gleichheit der Mittelwerte

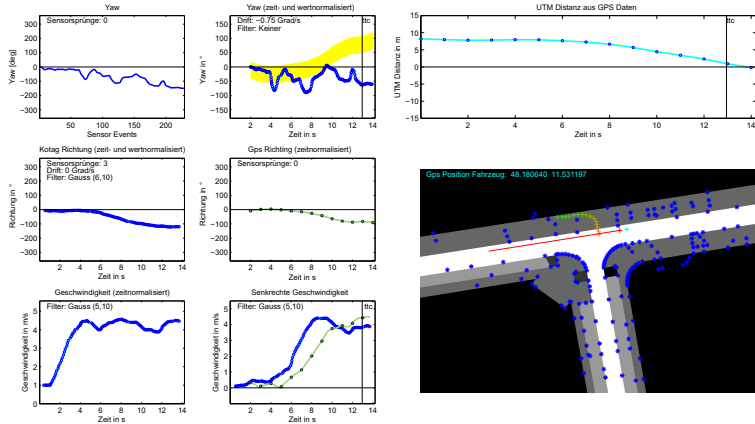


Abbildung 6: Zeigt die aufgezeichneten Daten der IMU-Sensoren und der GPS Daten. Der gelbe Streifen kennzeichnet den Sichtbereich des Fahrzeugs aus der Perspektive des Fußgängers. Liegt die blaue Kurve im mittleren oberen Plot im Bereich des gelben Sichtstreifens, blickt der Fußgänger in Richtung des Fahrzeugs.

Unterschied der Varianzen bei der ausgewählten Stichprobe ($n=17$) statistisch nachweisbar ist ($R = 0,5233$; $H = 1$; $P = 0,0127$; $Ci = 2,3609$ Inf). Um zu überprüfen, ab wann ein Unterschied feststellbar ist, wurde eine weitere Korrelationsanalyse durchgeführt. Dazu wurden die Kopfpositionsdaten bis zum Zeitpunkt, an dem die Bewegungsrichtung des Fußgängers um 20° von der ursprünglichen Richtung abweicht, betrachtet. Dieser Punkt wird als Abbiegepunkt festgelegt, da ab diesem Zeitpunkt auch eine Quergeschwindigkeit des Fußgängers in Richtung Fahrbahn vorhanden ist ($R = 0,4552$; $H = 1$; $P = 0,0287$; $Ci = 0,8832$ Inf). Von dem Abbiegepunkt ausgehend wird die Betrachtungszeit schrittweise verkürzt. Die Korrelationsanalyse zeigt, dass bis zu 60 Sensorevents (durchschnittlich 3,8 Sekunden) vor dem Abbiegepunkt weiterhin ein statistischer Unterschied besteht ($R = 0,4224$; $H = 1$; $P = 0,0403$; $Ci = 0,1039$ Inf).

Mit Hilfe der Kopfpositionsänderung wurde aus den Versuchsergebnissen der Zeitpunkt bestimmt, an dem die Kopfbewegung auf den Beginn des Sicherungsverhalten deutet. Ebenfalls wurde die TTC zu diesem Zeitpunkt ermittelt. Als Kollisionspunkt wurde dabei die Mitte der Fahrbahn angenommen. Durch die Festlegung des Kollisionspunktes wurde die Zeit zwischen der ersten Kopfdrehung und der Kollision bestimmt (vgl. Tabelle 1).

Neben einer frühzeitigen Erkennung der Querungsabsicht, stellt sich außerdem die Frage, woran ein rechtzeitiges Stehenbleiben des Fußgängers erkannt werden kann. Die Ergebnisse der Experimente zu Szenario 2 zeigen, dass ein Fußgänger bei der Erkennung eines potenziell gefährlichen Fahrzeugs sein Sicherungsverhalten unterbricht und das Fahrzeug fixiert (vgl. Abbildung 8).

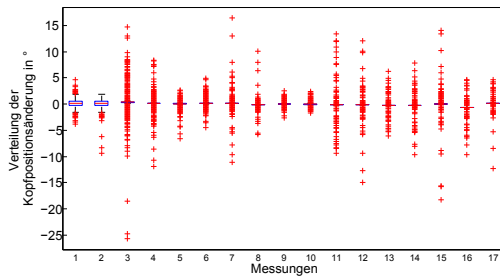


Abbildung 7: Boxplots der Kopfbewegungen zeigen die größere Streuung der Kopfbewegung bei den Szenarien mit Queren. Bei den Messungen 1-2, 5-6, und 9-10 handelt es sich um Szenarien ohne Queren.

Größe	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Zeit in s	8,559	4,718	3,295	4,380	7,304	5,593	2,052	2,877	5,811	4,220
Weg in m	35,07	30,65	25,20	15,68	13,65	13,09	12,80	13,41	11,05	10,48
TTC in s	8,062	6,238	3,092	4,466	11,70	4,493	1,863	2,597	6,182	4,563

Tabelle 1: Zeit zwischen erster Kopfbewegung und Kollision und TTC bei erster Kopfbewegung

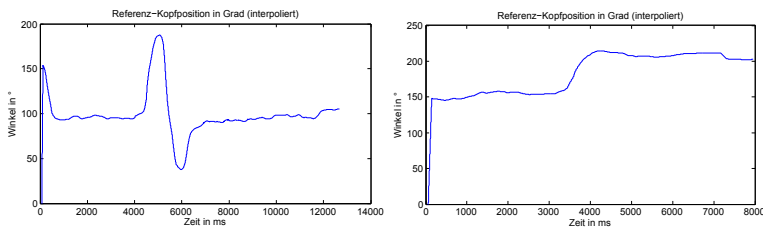


Abbildung 8: Kopfposition des Fußgängers beim Queren (links) und beim Anhalten am Straßenrand (rechts)

Fazit Für Szenario 1 konnte ein statistischer Unterschied aufgezeigt werden. Durch das Sicherungsverhalten verändert sich der Verlauf der Kopfpositionen vor Querungsbeginn. Dies zeigt eine grundsätzliche Eignung des Merkmals Kopfposition zur Intentionserkennung. Der Zeitraum zwischen der ersten Kopfbewegung und dem Kollisionszeitpunkt variiert stark. Er ist in allen Fällen jedoch größer als die Prädiktionzeit bisheriger Fußgängermodelle. Eine Kopfpositionserkennung besitzt also dann ein Potenzial zur früheren Erkennung der Querungsabsicht, wenn das Sicherungsverhalten möglichst frühzeitig stattfindet. Des Weiteren lässt sich bei Szenario 2 ein Unterschied im Sicherungsverhalten feststellen, wenn der Fußgänger ein potenziell kritisches Fahrzeug bemerkt.

4 Intentionserkennung bei Fußgängern

4.1 Absichtsmodellierung

Ziel der Intentionserkennung bei Fußgängern ist die Erkennung der Absicht die Straße vor einem sich nähernden Fahrzeug zu überqueren. Bisherige Modelle zur Intentionserkennung verfolgen dabei unter anderem Ansätze zum Trajektorien-Matching [12], basierend auf visuellen Umrissmerkmalen oder Conditional Random Field Modellen [13].

Im Folgenden wird ein Modell vorgestellt, das die Querungsabsicht speziell aufgrund der Kopfpose und des gezeigten Umschauverhalten des Fußgängers ermittelt. Dazu wird das statistische *Mixture of Gaussians Hidden Markov Modell* (MoG-HMM) verwendet. Der bestimmende Merkmalsvektor des Modells besteht aus dem Blickwinkel des Fußgängers auf das Fahrzeug, der ermittelten Quergeschwindigkeit und der Distanz zum Fahrerschlauch. Betrachtet werden die zwei Teilszenarien aus Abbildung 5, der Fußgängerklassifizierung, mit folgenden Annahmen:

1. Langzeit-Prognose (Szenario 1): Der Fußgänger taucht frühzeitig im Sichtfeld der Kamera auf. Prädiert wird die Absicht die Straße demnächst zu queren. Diese ist dabei maßgeblich von einer frühzeitig intensivierten Verkehrsbeobachtung geprägt.
2. Kurzzeit-Prognose (Szenario 2): Der Fußgänger tritt erst in das Sichtfeld der Sensoren, nachdem der sich bereits auf die Fahrbahn zu bewegt. Der Überquerungsvorgang hat bereits begonnen. Der Fußgänger entscheidet nach Erblicken des sich nähernden Fahrzeugs, ob er weiter läuft oder stehen bleibt.

Der Prädiktionszeitraum erstreckt sich dabei von der sicheren Erstklassifizierung bis hin zur Auflösung der Gefahrensituation. Je nach Situation und Verhalten des Fußgängers können die beiden Klassifikationsszenarien getrennt auftreten oder sequenziell hintereinander ablaufen.

4.2 MoG-HMM Klassifizierung

Das MoG-HMM wird durch das Quintupel (N, M, π, A, B) beschrieben. Dabei ist π die Wahrscheinlichkeitsverteilung des Anfangszustands, $A = a_{ij}$ die Übergangswahrscheinlichkeiten der Zustände und $B = b_i(k)$ die Wahrscheinlichkeit der Beobachtung b_i , des Merkmals k , im Zustand a_i . N ist die Anzahl der möglichen Zustände. M ist die Menge der Beobachtungen. Das Hidden Markov Modell wurde mit Hilfe des *Baum-Welch-Algorithmus* trainiert.

Als Klassifikations-, oder Auswertungs-Problem, wird bei Markov Modellen die Berechnung der Wahrscheinlichkeit $P(O|\lambda)$ bezeichnet. Diese gibt an, dass die beobachtete Merkmalssequenz O von dem vorliegenden Modell ($\lambda = \pi, A, B$) erzeugt wurde.

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (1)$$

Dabei ist α die *Forward-Variable*, welche die Wahrscheinlichkeit repräsentiert, die Merkmalssequenz $O = o_1 o_2 \dots o_t$ zum Zeitpunkt t ($1 < t \leq T$) beobachtet zu haben.

Hidden Markov Modelle besitzen häufig ein diskretes Alphabet an Merkmalsymbolen. Eine Quantifizierung der Sensordaten zur Abbildung auf ein eindimensionales Signal führt

allerdings zu Informationsverlusten. Ein Mixture of Gaussians Hidden Markov Model nähert hingegen die kontinuierliche Verteilung der Merkmals-Auftrittswahrscheinlichkeit mittels Gausskurven an.

$$b_j(O) = \sum_{m=1}^M c_{jm} \eta[O, \mu_{jm}, U_{jm}] \quad (2)$$

Dabei beschreibt η die Gauss-Verteilung für das m -te Merkmal im Zustand j . μ ist der Mittelwertsvektor und U_{jm} die zugehörige Kovarianzmatrix. $c_{jm} = P(M_m = m | a_t = j)$ bezeichnet die Gewichtung der m -ten Mix-Komponente in diesem Zustand.

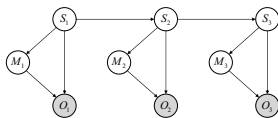


Abbildung 9: Zustandssequenz eines MoG-HMM

Für die Klassifizierung werden die in der Datenerhebung aufgezeichneten Datensätze mit einem festen Zeitintervall abgetastet. Im Hinblick auf eine größere Verarbeitungszeit der Bilderkennung wurde dieses Intervall auf 100 ms festgelegt.

4.3 Ergebnisse

Die Abbildungen 10 und 11 zeigen die Klassifikation auf Basis der Referenzdaten für eine Auswahl an Testsequenzen der beiden Szenarien 5a und 5b. Grundlage ist die Historie der Daten bis zum Zeitpunkt vor Betreten des Fahrschlauchs.

Die Güte der Langzeitprädiktion hängt, wie auch in Abschnitt 3.2 gezeigt, stark vom gezeigten Sicherungsverhalten ab. Je früher und deutlicher dieses ausgeprägt ist, desto besser kann der querende Fußgänger von einem vorbeigehenden Fußgänger unterschieden werden.

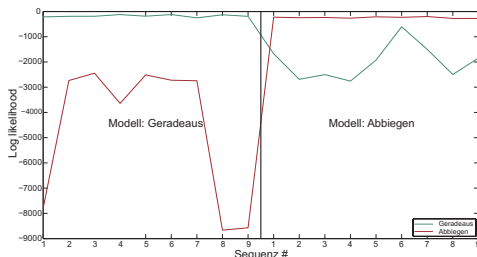


Abbildung 10: Testsequenz *Geradeaus* (grün) und *Abbiegen* (rot) der Langzeit-Prädiktion (Szenario 1), kreuzvalidiert in den jeweilig trainierten Modellen

Die Klassifikation ist dann korrekt, wenn der berechnete *Log-likelihood*-Wert der Datensequenz, in dem für dieses Szenario trainierten Modell, am höchsten ist.

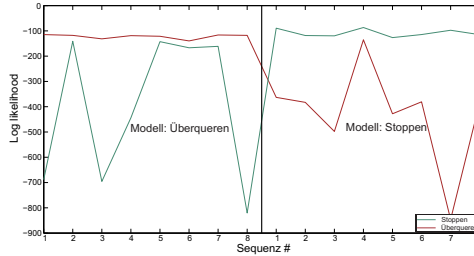


Abbildung 11: Testsequenz *Stehenbleiben* (grün) und *Überqueren* (rot) der Kurzzeit-Prädiktion (Szenario 2), kreuzvalidiert in den jeweilig trainierten Modellen

Im Gegensatz zu Szenario 1 zeigen sich die charakteristischen Verhaltensweisen erst später vor der Kollision, was sich in einer schwächeren Unterscheidbarkeit zwischen beiden Absichten, Überqueren und Stehenbleiben, widerspiegelt.

Mit der Sensorgenauigkeit des Referenzsystems lässt sich der Fußgänger in beiden Szenarien zu 100% korrekt klassifizieren. Zusätzliches, normalverteiltes Sensorrauschen der Kopfposition führt zu einer Verringerung der Detektionsrate (Tabelle 2).

Rauschverteilung	$\mu = 0, \sigma = 0^\circ$	$\mu = 0, \sigma = 15^\circ$	$\mu = 0, \sigma = 20^\circ$	$\mu = 0, \sigma = 40^\circ$
Korrekt klassifiziert	100%	87,5%	81,3%	75,0%

Tabelle 2: Rückgang korrekt klassifizierter Sequenzen bei steigendem additivem Rauschen. Normalverteilung mit μ = Mittelwert und σ = Standardabweichung.

Ebenfalls wurde der Einfluss der Länge der Datenhistorie untersucht. Unter der Voraussetzung eines ausgeprägten Sicherungsverhaltens, erreicht der Klassifikator ebenfalls eine 100%-Aussage bevor eine Unterscheidung durch kinematische Größen möglich ist.

5 Fazit

Die Auswertung einer Probandenstudie und weiteren Unfallstatistiken zeigt einen Zusammenhang zwischen der Kopfbewegung und der Querungsabsicht eines Fußgängers. In Abhängigkeit des Szenarios beginnt dieses charakteristische Verhalten bereits früher als die Veränderung kinematischer Größen, die momentan die einzige Grundlage aktueller Fußgängerschutzsysteme bilden. Um dieses Potenzial für ein Assistenzsystem nutzbar zu machen, wurde ein MoG-HMM-Klassifikator entwickelt, der die Kopfposenbewegung mit kinematischen Größen verknüpft und eine Aussage über die Querungsabsicht eines Fußgängers trifft. Zur Ermittlung der Kopfposition wurde ein Kamerasystem vorgeschlagen, das aufgrund der hohen Auflösung einen Kopf in mindestens 40 m Entfernung erkennt und dessen Kopfposition klassifiziert. Die Güte der Kopfpositionbestimmung beeinflusst dabei die Rate der Falschklassifikationen des MoG-HMM. Abhängig von der darauf aufbauenden Assistenzfunktion lassen sich hieraus die Anforderungen an das Bilderkennungs-system ableiten.

Literatur

- [1] N. Tiemann, *Ein Beitrag zur Situationsanalyse im vorausschauenden Fußgängerschutz*. Dissertation. Universität Duisburg-Essen, 2012.
- [2] C. Rodgers, D. Greenlee, R. Blomberg, *System and method for providing pedestrian alerts*. U.S. Patent Nr. 7,095,336, 2006.
- [3] K. Fuerstenberg, *Pedestrian protection using laserscanners*. Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, September 13-16, 2005
- [4] O. Scherf, S. Zecha, *Verfahren zum Bestimmen eines wahrscheinlichen Bewegungsbereichs eines Lebewesens*. Nr. 102007037610, 2009.
- [5] M. Meinecke, M. Roehder und N. Nguyen, *Motion model estimation for pedestrians in street-crossing scenarios*. Proceedings of the International Workshop on Intelligent Transportation, 2010.
- [6] G. de Nicolao, A. Ferrara und L. Giacomini, *Onboard sensor-based collision risk assessment to improve pedestrians' safety*. Vehicular Technology, IEEE Transactions, 56. Jg., Nr. 5, S. 2405-2413, 2007.
- [7] Advanced Forward-Looking Safety Systems (vFSS) - eVALUE http://www.evaluate-project.eu/pdf/evaluate-20101124-wp5-final_event_presentation_vfss.pdf
- [8] Destatis, *Verkehr Verkehrsunfälle Fachserie 8 Reihe 7 - September 2013*. Wiesbaden, Germany: Statistisches Bundesamt, 2013.
- [9] S. Schmidt und B. Faerber, *Pedestrian at the kerb- Recognising the action intentions of humans*. Transportation Research Part F: Traffic Psychology and Behaviour, 12, p.300-310, 2009 .
- [10] J.G. Schoon *Pedestrian observation-reaction times: concepts and pilot study*. 35th Annual Meeting of the Universities' Transport Study Group, Loughborough University, Loughborough, UK, 2003.
- [11] T. Schweizer, C. Thomas und P. Regli, *Verhalten am Fussgängerstreifen*. Zürich, Schweiz: Fussverkehr Schweiz, 2009.
- [12] C. G. Keller und C. Hermes und D. M. Gavrila *Will the Pedestrian Cross? Probabilistic Path Prediction Based on Learned Motion Features* Springer-Verlag Berlin Heidelberg, 2011
- [13] F Flohr und und D. M. Gavrila *PedCut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues* Environment Perception Department, Daimler R&D, Ulm, Germany

Kartengestütztes automatisiertes Fahren auf der Bertha-Benz-Route von Mannheim nach Pforzheim

Julius Ziegler* Philipp Bender[†] Henning Lategahn[‡]
Markus Schreiber[§] Tobias Strauß[¶] Thao Dang^{||}
Christoph Stiller^{**}

Zusammenfassung: Im September 2013 fuhr die Mercedes-Benz-S-Klasse S 500 INTELLIGENT DRIVE der Daimler AG vollautomatisch von Mannheim nach Pforzheim. Sie folgte dabei der 100 km langen Route, die Bertha Benz 125 Jahre zuvor für die erste automobilen Fernfahrt wählte. Das Fahrzeug war ausschließlich mit video- und radarbasierter Sensorik ausgestattet. Eine besondere Herausforderung lag in der Streckenführung, die durch eine hohe Variabilität der Verkehrsräume, und damit auch der möglichen Verkehrsszenarien gekennzeichnet ist. Entlang der Route liegen die Großstädte Mannheim, Heidelberg und Pforzheim sowie 23 kleinere Ortschaften. Auf Überlandstrecken wird bis 100 km/h gefahren. Insgesamt sind 18 Kreisverkehre und 155 Ampelkreuzungen zu überfahren. Dieser Beitrag umfasst den Teil des Systems, der, basierend auf einem gegebenen Lagebild der Fahrzeugsensorik, eine Solltrajektorie plant, die dann von einem Fahrzeugregler in Lenk- und Beschleunigungsvorgaben für die Fahrzeugaktorik umgesetzt wird. Hierbei kam eine detaillierte digitale Straßenkarte zum Einsatz. Die Struktur dieser Karte, der Prozess ihrer Erstellung und ihre Verwendung in den für die automatische Fahrfunktion entwickelten Teilsystemen werden in diesem Artikel beschrieben.

Schlüsselwörter: autonomes Fahren, Trajektorienplanung, videobasierte Lokalisierung

1 Einleitung

Autonome oder teilautonome Fahrfunktionen bergen das Potential, Verkehrsunfälle zu vermeiden, die durch menschliche Fehler verursacht werden. Dass autonomes Fahren im öffentlichen Straßenverkehr möglich ist, wurde im September 2013 demonstriert. Eine zum autonomen Fahren ausgestattete Mercedes-Benz S-Klasse fuhr entlang der Bertha Benz Memorial Route (Abbildung 1) autonom von Mannheim nach Pforzheim.

*FZI Forschungszentrum Informatik, Karlsruhe,

[†]pbender@fzi.de, FZI Forschungszentrum Informatik, Karlsruhe

[‡]henning.lategahn@kit.edu, Institut für Mess- und Regelungstechnik (MRT), Karlsruher Institut für Technologie (KIT), Karlsruhe

[§]schreiber@fzi.de, FZI Forschungszentrum Informatik, Karlsruhe

[¶]strauss@kit.edu, Institut für Mess- und Regelungstechnik (MRT), Karlsruher Institut für Technologie (KIT), Karlsruhe

^{||}thao.dang@daimler.com, Daimler AG, Research and Development, Sindelfingen

^{**}stiller@kit.edu, Institut für Mess- und Regelungstechnik (MRT), Karlsruher Institut für Technologie (KIT), Karlsruhe



Abbildung 1: Übersicht über die Ortschaften entlang der historischen Route.

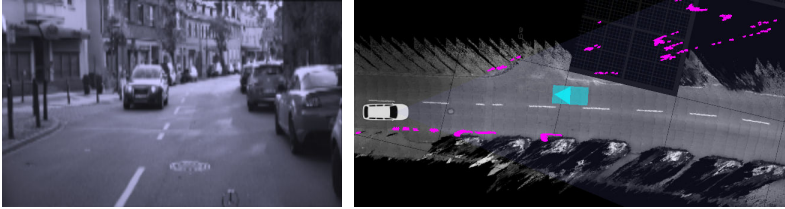


Abbildung 2: Umfeldrepräsentation: Statische Hindernisse und Objekte.

Der vorliegende Artikel beschreibt die Strategien zur Bewegungsplanung und Fahrzeugregelung, die für dieses Projekt entwickelt wurden. Von größter Wichtigkeit war hierbei die Bereitstellung einer detaillierten digitalen Straßenkarte. In Abschnitt 2 werden wir zunächst einige Voraussetzungen aufzuführen, die für das kartenbasierte Fahren erforderlich sind. Hierzu gehört an erster Stelle eine genaue Lokalisierung relativ zur Straßenkarte. Es werden videobasierte Verfahren aufgeführt, die ohne Satellitenortung auskommen. Anschließend gehen wir kurz auf die Schnittstelle zur Fahrzeugsensorik ein. Abschnitt 3 dokumentiert die digitale Straßenkarte und den Prozess ihrer Erstellung. In Abschnitt 4 wird erläutert, wie Kartendaten und Daten der Fahrzeugsensorik in die Bewegungsplanung eingehen, und wie die geplante Trajektorie durch einen Regler in Stellbefehle für die Fahrzeugaktuatorik umgesetzt wird.

2 Voraussetzungen

Sensorik In diesem Abschnitt werden kurz die Sensordaten beschrieben, die für die Bewegungsplanung zur Verfügung stehen. Im Projekt kamen ausschließlich seriennahe Kamera- und Radarsysteme zum Einsatz. Wir möchten hier hauptsächlich die Repräsentation, in der die Sensordaten in die Bewegungsplanung eingehen, beschreiben. Die Beschreibung von Sensorhardware ist nicht Teil dieser Arbeit. Dasselbe gilt für die Bild- und Signalverarbeitung, sowie die dieser nachgeschalteten Fusions- und Validierungsprozesse. Für eine ausführliche Darstellung der verwendeten Bildverarbeitungsalgorithmen verweisen wir auf [4].

Das sensorische Lagebild setzt sich zusammen aus einer Abtastung statischer Hindernisse und einer Liste von Objekten. Unter Hindernissen verstehen wir ausschließlich *statische* Objekte im Verkehrsraum. In diesem Projekt wurden sie durch Stereobildverarbeitung detektiert. Das Kamerabild wird hierbei in Spalten zerlegt, die einige Pixel breit sind. Pro Spalte wird dann die Entfernung zum jeweils nächstgelegenen Hindernis bes-

timmt. Die Hindernisse werden also an ungefähr äquidistanten Winkelstellen abgetastet. Ein einzelner Abtastpunkt wird im folgenden auch als *Stixel* bezeichnet. Die Stixel werden über einen kurzen Zeitraum in einem weltfesten Koordinatensystem akkumuliert. In Abbildung 2 sind Stixel rechts magentafarben dargestellt.

Objekte werden durch Fusion von Stereobild- und Radardatenverarbeitung erzeugt. Während die statische Hinderniskontur lediglich den Fahrkorridor des Fahrzeuges geometrisch einschränkt, spielen Objekte eine entscheidende Rolle auch auf Ebene der Verhaltensentscheidung, beispielsweise für die Bewertung von Vorfahrtsituationen. In Abbildung 2 wird das entgegenkommende Fahrzeug als Objekt repräsentiert und ist hier als zyanfarbenes Viereck eingezeichnet. In Fusions- und Validierungsstufen wird sichergestellt, dass die beiden Repräsentationen konsistent sind und dass Redundanzen aufgelöst wurden: im Bild ist zu sehen, dass Abtastpunkte der Freiraumkontur auf dem Objekt fehlen.

Die im Bild zu sehende Bodentextur wurde bei einer Kartierungsfahrt mit einer nach hinten gerichteten Stereokamera erstellt. Die so gewonnenen Daten dienen sowohl als Grundlage zur Erzeugung der digitalen Straßenkarte als auch zur Erzeugung einer Referenzkarte zur videobasierten Lokalisierung.

Lokalisierung über Punktmerkmale Nachfolgend beschreiben wir ein videobasiertes System, welches Punktmerkmale im Kamerabild erkennt, mit Hilfe derer es sich relativ zu einer zuvor berechneten Merkmalskarte lokalisiert. Initial wird die gesamte Strecke mit einem Stereokamerasystem vermessen. Die gesammelten Bilddaten werden nachfolgend in einer Offline-Berechnung in eine digitale Merkmalskarte überführt. Der Kartierungsprozess erfolgt vollautomatisch und erzeugt eine Menge dreidimensionaler Punktmerkmale. Zuerst wird die gefahrene Trajektorie des Fahrzeugs ermittelt und die zugehörige Kamerapose bestimmt. Die Bildmerkmale werden danach über die gesamte Sequenz assoziiert, sodass aus der Beobachtung eines Punktes aus unterschiedlichen Posen dessen 3D-Position durch nichtlineare Ausgleichsrechnung geschätzt werden kann. Diese natürlichen Landmarken werden zusammen mit Bilddeskriptoren in einer Kartendatenbank gespeichert.

Die Lokalisierung wird mit einer monoskopischen Kamera durchgeführt. Landmarken in der direkten Fahrzeugumgebung werden aus der Karte geladen und Bildpunkten des aktuellen Kamerabildes zugeordnet. Aus dieser Zuordnung lässt sich über eine robuste Ausgleichsrechnung die Pose der Kamera relativ zur vorher berechneten Karte bestimmen. Abbildung 3 veranschaulicht schematisch das beschriebene Verfahren. Das Verfahren wird detailliert in [10, 11, 9] beschrieben.

Lokalisierung über Fahrspurmarkierungen Insbesondere in außerstädtischer Umgebung hat sich gezeigt, dass oft keine ausreichende Anzahl Punktmerkmale zugeordnet werden kann. In diesem Abschnitt wird eine Lokalisierung relativ zu *Fahrspurmarkierungen* beschrieben, die die punktmerkmalbasierte Lokalisierung ergänzt. Dazu wird in einem ersten Schritt eine detaillierte Karte mit allen Fahrbahnmarkierungen erstellt. Hierzu werden die Werkzeuge verwendet, die später in Abschnitt 3 beschrieben werden. Neben Seitenmarkierungen (durchgezogen und gestrichelt) und Haltelinien sind auch Bordsteine und Tram-Schienen hinterlegt. In der Online-Lokalisierung wird die Karte der Umgebung geladen und in das Kamerabild projiziert. Im Bereich der in der Karte hinterlegten Markierungen wird nun im Bild nach tatsächlich vorhandenen Fahrbahnmarkierungen gesucht. Die Assoziation zwischen gefundenen Punkten auf den Fahrbahnmarkierungen

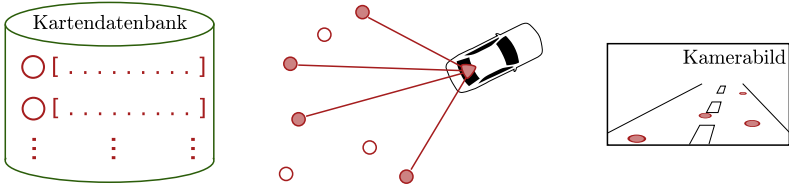


Abbildung 3: Punktmerkmalsbasierte Videolokalisierung mit rückwärtsgerichteter Kamera: 3D Landmarken werden aus einer Kartendatenbank mit dazugehörigen Deskriptoren geladen und Bildpunkten des aktuellen Kamerabildes zugeordnet. Gezeigt ist die Szene in vogelperspektive, wobei rote Kreise erfolgreich assoziierte Landmarken darstellen.



Abbildung 4: Markierungsbasierte Lokalisierung: Markierungskarte, Projektion ins Bild, Assoziation von Kartendaten und Messungen.

und der Karte erfolgt über eine Nächste-Nachbar-Zuordnung zu der abgetasteten Karte (Abbildung 4). Über ein Kalmanfilter werden diese Residuen über die Zeit minimiert. Falls statt einer Fahrbahnrandmarkierung nur ein Bordstein vorhanden ist, wird das Verfahren durch einen Bordsteinklassifikator aus [3] gestützt. Das gesamte Verfahren ist im Detail in [13] beschrieben.

3 Digitale Straßenkarte

Von zentraler Bedeutung für die Bewegungsplanung ist eine detaillierte *digitale Straßenkarte*. In ihr wird offline all das abgelegt, was das Fahrzeug nicht oder nicht zuverlässig genug zur Fahrtzeit über seine Sensoren erfassen kann. Dies betrifft zum Beispiel den Verlauf von Fahrstreifen, insbesondere innerhalb von Kreuzungen. Die Datengrundlage für die manuelle Erstellung der in dieser Karte enthaltenen Geometrien sind virtuelle Luftbilder, die mittels Stereorekonstruktion von Punkten der Fahrbahnebene erzeugt werden: aus jedem Stereobildpaar wird ein Disparitätsbild berechnet und daraus eine 3D-Rekonstruktion

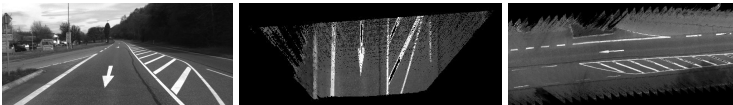


Abbildung 5: Einzelbild Stereokamera, rekonstruierte Draufsicht, Überlagerung.

des nahen Fahrzeugumfelds erzeugt. Diese 3D-Punkte werden auf die Grundebene projiziert und mittels der aufgezeichneten Fahrzeugtrajektorie einander überlagert (Abbildung 5). Zu sehen sind diese Bilder als Hintergrund der Abbildungen 2 und 6. Die hierfür verwendeten Stereodaten sind die selben, die für die Erzeugung der Referenzkarte für die punktmerkmalsbasierte Lokalisierung verwendet werden (Abschnitt 2). Dies garantiert, dass die videobasierte Lokalisierung relativ zur Straßenkarte erfolgt.

Zur Repräsentation und Erstellung der Karte benutzen wir Formate und Werkzeuge des OpenStreetMap-Projektes (OSM) [12]. OSM definiert ein XML-basiertes Dateiformat zum Austausch von Kartendaten. Außerdem wird eine Serverarchitektur [1] bereitgestellt, auf der Kartendaten zentral verwaltet werden können. Diese Architektur ermöglicht es, dass mehrere Benutzer an verschiedenen Orten parallel Teile der Karte bearbeiten. Änderungssätze werden vom Server automatisch zusammengeführt. Zur manuellen Annotierung der Kartendaten verwenden wir den OSM-Editor JOSM.

In den folgenden Abschnitten beschreiben wir den strukturellen Aufbau der Karte. Abschließend wird erläutert, wie zur Laufzeit effizient auf die umfangreichen Kartendaten zugegriffen werden kann.

3.1 Aufbau der Karte

Knoten, Polygonzüge und Relationen Die Grundelemente, aus denen die Straßenkarte aufgebaut ist, sind *Knoten*, *Polygonzüge* und *Relationen*. Knoten und Polygonzüge entsprechen in OSM den Primitiven *nodes* und *ways*. Jedes dieser Grundelemente wird zur Erstellungszeit automatisch mit einer eindeutigen Stammmnummer (*ID*) versehen. Ein Knoten ist durch seine geographische Position (Länge und Breite) vollständig beschrieben. Ein Polygonzug wird durch eine Liste definiert, die die IDs der Knoten enthält, die der Polygonzug interpolieren soll. Eine Relation enthält eine beliebige Anzahl IDs. Jedem dieser Kindelemente ist innerhalb der Relation eine Rolle zugewiesen. Alle Grundelemente können mit *Attributen* versehen werden, so das beispielsweise einem Fahrstreifensegment eine Geschwindigkeitsbegrenzung zugeordnet werden kann. Insbesondere das flexible Konzept der Relationen und Attribute erlaubt es, OSM an die Anforderungen des Projektes anzupassen.

Fahrstreifensegmente und Fahrstreifentopologie Die Straßenkarte besteht aus *Fahrstreifensegmenten*. Unter einem Fahrstreifensegment ist ein in der Regel kurzer Fahrstreifenabschnitt zu verstehen, der – außer im Spezialfall eines lateralen Spurwechsels – nur über je einen Ein- und Ausgang betreten und wieder verlassen werden kann. Ein solches Segment wird über Polygonzüge, die den jeweils linken und rechten Rand bilden, beschrieben. Im OSM-Formalismus wird ein Fahrstreifensegment als eine Relation repräsentiert, die zwei Polygonzüge enthält. Innerhalb dieser Relation erhält jeder Polygonzug die Rolle *left* (links) oder *right* (rechts). Über die Identität der Anfangs- und Endknoten der Fahrstreifensegmentränder lassen sich die einzelnen Fahrstreifensegmente untereinander in eine Nachfolgebeziehung setzen, wodurch ein routingfähiger Graph entsteht. Exemplarisch ist dies in Abbildung 6 dargestellt: zu sehen sind sechs Fahrstreifensegmente (hier mit 1 bis 6 bezeichnet). Das Segment 4 ist hervorgehoben: der rechte Rand ist grün, der linke rot markiert. An den orangenen Punkten sind Fahrstreifensegmente miteinander verbunden. Rechts als Graph dargestellt ist die Topologie zu sehen, die sich aus diesen

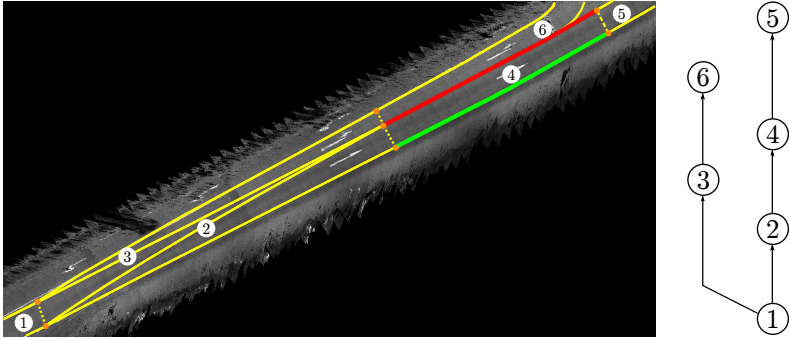


Abbildung 6: Fahrbahnrandgeometrien und resultierende Fahrstreifenentopologie

Fahrstreifensegmenten ableiten lässt. Zur Kartierung der Bertha-Benz-Route wurden etwa viertausend Fahrstreifensegmente manuell annotiert.

Vorfahrtsituationen (Abbiegen, Einfädeln) und Ampeln Später in Abschnitt 4.1 zeigen wir, dass jeder Vorfahrtsituation als Abfolge von zwei Grundmaneuvern ausgedrückt werden kann, *merge* und *yield*. Beide werden in der Karte durch den gleichen Relationstyp beschrieben. Eine Vorfahrtsrelation enthält eine Haltelinie und eine Referenzlinie (jeweils ein Polygonzug mit zwei Punkten), sowie einen Verweis auf das Fahrstreifensegment, dem Vorrang zu gewähren ist oder mit dem zusammengeführt wird. Der Relation werden noch zwei Punkte hinzugefügt, die beim Überfahren Ereignisse auslösen, die den Zustandsautomaten zur Verhaltensgenerierung (Abschnitt 4.1) steuern. Diese Punkte heißen **trigger** und **complete**. Ampelgesteuerte Kreuzungsüberfahrten werden ähnlich in der Karte repräsentiert.

3.2 Zugriff auf die Kartendaten zur Laufzeit

Die Karte mit allen Geometrien und Relationen wird in einem R-Baum (engl. *R-Tree*) abgelegt[6]. Ein R-Baum ist eine hierarchische Datenstruktur, die es erlaubt, räumliche Anfragen (engl. *spatial queries*) effizient zu beantworten. Es ist so möglich, effizient auf die Elemente der Karte zuzugreifen, die sich in einem definierten Umkreis um die aktuelle Fahrzeugposition befinden. In den meisten Fällen sind Teile der Karte, die weit (>200 Meter) vom Fahrzeug entfernt sind, für das Fahrzeugverhalten irrelevant. Nachgeschaltete Module operieren stets nur auf dieser lokalen Teilmenge der Kartendaten. Die Verfahren werden dadurch beschleunigt und ihre Laufzeit wird unabhängig vom Umfang der Gesamtkarte.

4 Bewegungsplanung

Für die Bewegungsplanung sind drei Aufgaben zu unterscheiden. Die Verhaltensentscheidung legt Nebenbedingungen fest, denen die nachgeschaltete Trajektorienplanung unterworfen wird. Die Trajektorienplanung erzeugt dann eine Trajektorie, die diese Nebenbedingungen einhält und ein gewähltes Gütemaß optimiert. Die wiederum nachgeschaltete Regelung stabilisiert das Fahrzeug auf dieser Solltrajektorie. Auf diese drei Teilaufgaben wird in den folgenden Unterabschnitten eingegangen.

4.1 Verhaltensgenerierung

Das Modul *Verhaltensgenerierung* ist der Trajektorienplanung vorgeschaltet. Die Trajektorienplanung ist als differentialgeometrisches Optimierungsproblem formuliert, die Trajektorie soll also ein Zielfunktional minimieren, das hauptsächlich nach fahrdynamischen Kriterien ausgewählt wurde, so enthält es zum Beispiel Strafterme für Ruck und Beschleunigung. Für dieses Optimierungsproblem erzeugt nun die Verhaltensgenerierung im Vorfeld Nebenbedingungen (engl. *Constraints*), die dann von der Trajektorienplanung berücksichtigt werden. Auf die eigentliche Trajektorienplanung gehen wir im folgenden Abschnitt 4.2 noch detailliert ein, zunächst wird nun beschrieben, welche Nebenbedingungen es gibt und wie sie situationsabhängig erzeugt werden.

Einige der Nebenbedingungen werden aus der Karte abgeleitet, das ist in erster Linie der Verlauf der Fahrkorridors, aber auch die Lage von Haltelinien. Haltelinien werden von der Verhaltensgenerierung je nach Verkehrssituation aktiv oder inaktiv geschaltet. Andere Nebenbedingungen entstehen durch die Auswertung der in Abschnitt 2 beschriebenen Sensordaten. Im Falle des Einfädels in fließenden Verkehr - das ist beispielsweise bei der Einfahrt in einen Kreisverkehr erforderlich - werden spezielle Weg-Zeit-Bedingungen für die Trajektorienplanung erzeugt. Die einzelnen Nebenbedingungen werden in den folgenden Abschnitten beschrieben.

Zustandsautomat Die Verhaltensgenerierung ist als *hierarchischer, nebenläufiger Zustandsautomat* definiert. Die hierzu verwendete Notation wurde von David Harel eingeführt[7] und wird deshalb auch als *Harel Statecharts* bezeichnet. Die Notation erlaubt die übersichtliche und durchschaubare Modellierung reaktiver Systeme. Die einfache Notation endlicher Automaten wird hierbei um *nebenläufige* Zustände erweitert, außerdem können die Zustände *hierarchisiert* werden, wodurch eine schrittweise Verfeinerung (top-down) des Entwurfs möglich wird. In einer *Zustandshierarchie* befindet sich das System zusätzlich in allen Elternzuständen eines aktiven Zustandes. Zustände können nebenläufig sein, dann befindet sich das System gleichzeitig in allen parallelen Unterzuständen. Auf Ereignisse reagieren aktive Zustände mit Zustandsübergängen. Mit dem Betreten und Verlassen eines Zustandes kann Programmcode verknüpft werden. Abbildung 7 auf der nächsten Seite zeigt einen Ausschnitt des verwendeten Automaten. Zustandsnamen beginnen mit **St**. Der linke Teil der Abbildung zeigt das Konzept der Nebenläufigkeit. Rechts dargestellt ist die Detailansicht mit Unterzuständen des Zustandes **StManageGiveWay**, der sich normalerweise in einem Wartemodus befindet. Ist eine Vorfahrtssituation mit dem aktuellen Fahrkorridor assoziiert und ist deren Auslösepunkt überfahren (**trigger**, Ereignis T), wird der Zustand **StGiveWay** aktiv. In diesem Zustand gibt es zwei Unterzustände: standardmäßig geht das System in den Zustand **StDriveAutomatically**, der für eine

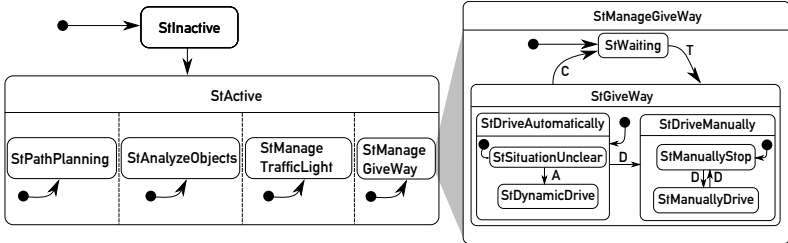


Abbildung 7: Ausschnitt des Zustandsautomaten zur Verhaltensgenerierung.

automatische Durchfahrt der Situation sorgt, sowie ein vom Fahrer aktivierbarer Zweig (*StDriveManually*, Ereignis *D*, *Driver*), der ein von der Sensorik unabhängiges Testen der Planung und Regelung ermöglicht oder bei unzureichender Sensorinformation im Ausnahmefall aktiviert werden kann. Im Automatikmodus geht das System zuerst in den Zustand *StSituationUnclear*, der der eingeschränkten Sensorreichweite Rechnung trägt und mit Hilfe geeigneter Nebenbedingungen dafür sorgt, dass das Fahrzeug an einem definierten Punkt zum Stillstand kommt. Dieser Zustand wird jedoch verlassen, sobald die Sensorik den relevanten Bereich zuverlässig abdeckt. Dies lässt sich über einen zu unterschreitenden Abstand zur Haltelinie für jede Kreuzung einstellen (Ereignis *A*). Das System geht dann in den Zustand *StDynamicDrive* über, in dem abhängig von der Verkehrssituation Nebenbedingungen für die Trajektorienplanung generiert werden. Ist ein weiterer Punkt am Ende der Kreuzung überfahren (*complete*, Ereignis *C*), wird der Wartezustand wieder aktiv. Diese Art von Zustandsübergängen bezeichnen wir als *kartengetrieben*.

Fahrkorridor Als *Fahrkorridor* wird die Abfolge der Fahrstreifensegmente bezeichnet, die die Route zum Ziel bilden, oder gegebenenfalls ein lokaler Ausschnitt davon, wobei ausreichend Vor- und Rückschau für die Trajektorienplanung garantiert wird. Zur Ermittlung des Fahrkorridors wird zunächst die Fahrzeugpose genau einem Fahrstreifensegment aus der Karte zugeordnet, wobei die Abstände zu den Segmenträndern und die Winkelabweichung der Fahrzeugpose zum Segmentverlauf berücksichtigt werden. Von diesem initialen Segment aus wird in der Fahrstreifentopologie (Abschnitt 3.1) durch Graphsuche der kürzeste Weg zum Zielsegment ermittelt. Eine Nebenbedingung der Trajektorienplanung ist, dass sich die Trajektorie innerhalb des Fahrkorridors befindet.

Hindernisse Statische Hindernisse werden wie in Abschnitt 2 beschrieben durch eine Menge von Stixeln repräsentiert. Für alle Stixel, die innerhalb des Fahrkorridors liegen, wird entschieden, ob das Fahrzeug sie links oder rechts passieren soll. Hierzu wird ein optimaler Graphschnitt durchgeführt. Der Aufbau des Graphen ist in Abbildung 8a und 8b schematisch dargestellt: Jedem einzelnen Stixel - im linken Beispiel 7, im rechten 9 - entspricht ein Knoten im Graph. Die beiden größer dargestellten Knoten repräsentieren den linken und rechten Rand des Fahrkorridors. Zwei Knoten werden verbunden, wenn zwischen den entsprechenden Stixeln, bzw. zwischen entsprechendem Stixel und Korridorrand, eine Durchfahrt geometrisch nicht kollisionsfrei möglich ist. Es wird nun ein mini-

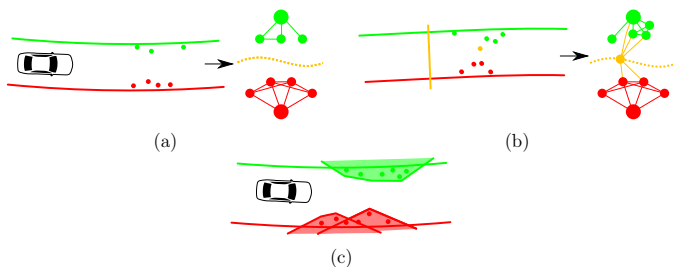


Abbildung 8: Vorverarbeitung der Hindernisdaten unter Berücksichtigung des Verlaufes des Fahrkorridors.

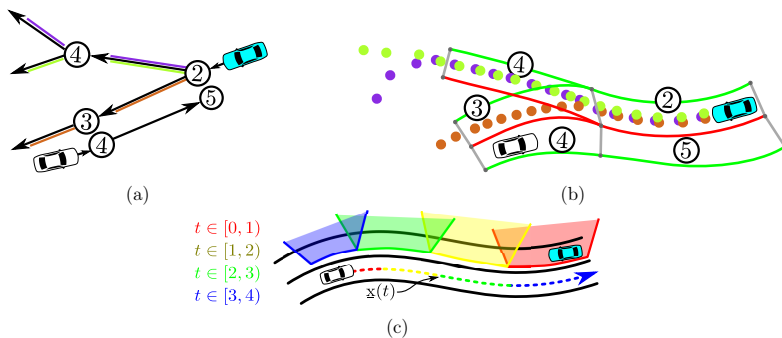


Abbildung 9: Nebenbedingungen für bewegte Objekte

maler Schnitt (engl. *minimum vertex cut*)[2] durch den Graphen berechnet, das bedeutet, es wird die kleinst-mögliche Menge Knoten aus dem Graphen entfernt, sodass linker und rechter Fahrbahnrand nicht mehr zusammen hängen. Für den Fall in Abbildung 8a ist diese Menge leer, da der Graph initial nicht zusammenhängend war. In Abbildung 8b kann der Graph durch Entfernung von einem Knoten (orange) geschnitten werden. Wenn die Entfernung mindestens eines Knotens erforderlich ist, so bedeutet dies immer, dass an dieser Stelle ein Durchfahren nicht möglich ist. Dann wird an einer geeigneten Stelle eine Haltelinie erzeugt (im Bild orange).

Wenn alle Stixel auf diese Weise dem linken oder rechten Fahrstreifenrand zugeordnet wurden, werden sie wie in Abbildung 8c durch konvexe Polygone eingehüllt. Dies hat folgende praktische Gründe: Erstens kann auf diese Weise ein Sicherheitsabstand vorgesehen werden. Zweitens ist es zur Formulierung einer Nebenbedingung für die Trajektorienplanung erforderlich, eine differenzierbare Abstandsfunktion zu den Hindernissen anzugeben.

Objekte Jedes von der Sensorik bereitgestellte Objekt (Abschnitt 2) wird räumlich einem oder mehreren Fahrstreifensegmenten zugeordnet. Von jedem dieser möglichen

Fahrstreifensegmente aus wird in der Fahrstreifentopologie (Abschnitt 3.1) eine Suche nach dem kürzesten Pfad expandiert. So erhält man alle Fahrkorridore, die das Fahrzeug innerhalb eines bestimmten Zeithorizonts (<20 Sekunden) befahren kann. In Abbildung 9a wurden für ein entgegenkommendes Objekt drei mögliche Pfade identifiziert. Für jeden dieser Fahrkorridore wird eine Trajektorie für das Objekt präzisiert, wobei angenommen wird, dass das Fahrzeug dem Fahrstreifen folgt und den Abstand zum rechten Rand beibehält. Abbildung 9b zeigt, wie für das entgegenkommende Fahrzeug drei Trajektorienhypothesen erzeugt wurden. Ähnlich wie für die statischen Hindernisse werden für jede Trajektorie nun Polygone identifiziert, die aber, da das Objekt in Bewegung ist, nur für einen bestimmten Zeitraum aktiv sind. In Abbildung 9c sind diese Polygone schematisch veranschaulicht. Jede Farbe entspricht einem Zeitintervall. Es werden nun unterschiedliche Zeitbereiche der eigenen Trajektorie (gestrichelt) durch eine neue polygonale Nebenbedingung eingeschränkt. Im Bild würden sich die Fahrzeuge voraussichtlich im gelben Bereich begegnen. Die im Betrieb tatsächlich verwendeten Zeitintervalle sind 0,3 Sekunden lang.

Vorfahrt gewähren und Einfädeln in fließenden Verkehr Es gibt zwei Klassen von Abbiege- und Einfädelvorgängen: Solche, bei denen vorfahrtsberechtigten Fahrstreifen den Eigenen kreuzen (*yield*-Typ), und solche, bei denen Fahrstreifensegmente zusammenlaufen (*merge*-Typ). Biegt man beispielsweise an einer T-Kreuzung, vom einmündenden Kreuzungsarm kommend und nicht vorfahrtsberechtigt, nach links ab, so wird der eigene Fahrweg vom von links kommenden Verkehr gekreuzt (*yield*). Anschließend läuft der eigene Fahrstreifen mit dem des von rechts kommenden Verkehrs zusammen (*merge*). Durch die nebenläufige Struktur des Zustandsautomaten können beide Teilprobleme separat, aber gleichzeitig behandelt werden. Eine *merge*-Maneuvers leitet auch die Einfahrt in einen Kreisverkehr ein.

Wir betrachten im folgenden Objekte in der Weg-Zeit-Ebene, es wird also auf der Abszisse die Zeit abgetragen, und auf der Ordinate der zurückgelegte Weg. Objekte, die nach dem *yield*-Typ behandelt werden müssen, bilden in der Weg-Zeit-Ebene rechteckige Hindernisse. Der *merge*-Typ erfordert trapezförmigen Weg-Zeit-Hindernisse, deren Herleitung wir in in Abbildung 10 auf der nächsten Seite exemplarisch illustrieren. Die Referenzlinie (grün) entstammt der Kartenrelation (Abschnitt 3.1), sie dient dazu, über die Hilfsgrößen $d_{\text{ref-stop}}$, d_{ref}^2 und d_{ref}^1 (Abbildung 10) alle beteiligten Fahrzeuge in ein gemeinsames, eindimensionales Koordinatensystem zu referenzieren (Abbildung 10b). Dessen Ursprung ist der vordere Stoßfänger des eigenen Fahrzeuges (weiß). Ab der Stelle s_{stop} (rot) beginnt der Bereich, ab dem in dieser 1D-Abstraktion nicht mehr zwei Fahrzeuge nebeneinander fahren können. Ausgehend von der gemessenen Fahrzeuggeschwindigkeit v des vorfahrtsberechtigten Fahrzeuges (zyan), werden zwei gleichförmig beschleunigte Trajektorien angenommen, die den möglichen Beschleunigungsspielraum des vorfahrtsberechtigten Fahrzeuges widerspiegeln. Die Zeiten, an denen diese Trajektorien die Haltelinie schneiden (t_{ue} , *upper estimate*, t_{le} , *lower estimate*) bilden zusammen mit den Geschwindigkeiten an dieser Stelle (v_{ue} , v_{le}) das trapezförmige Weg-Zeit-Hindernis (Abbildung 10c). Das Trapez ist nach oben unbeschränkt. Die so erzeugten Trapeze bilden eine Nebenbedingung für die Trajektorienplanung.

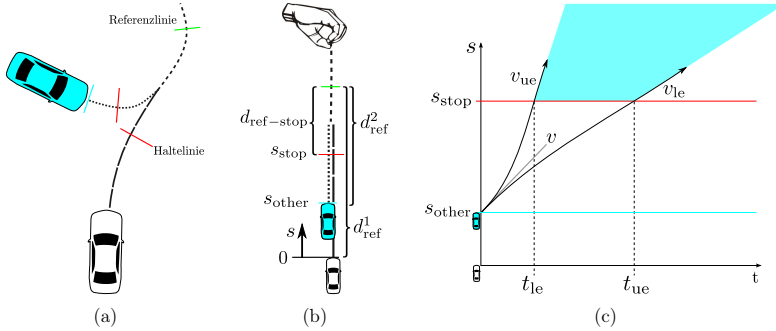


Abbildung 10: Vorfahrtssituation des *Merge*-Typs: (a) Szene in der Draufsicht, (b) 1-D-Abstraktion, (c) resultierendes Weg-Zeit-Hindernis (zyan).

4.2 Trajektorienplanung

Die Trajektorienplanung ermittelt unter Berücksichtigung der von der Verhaltensentscheidung aufgestellten Nebenbedingungen eine Optimaltrajektorie $\mathbf{x}(t) = (x(t), y(t))^T$. Als Referenz für den Regler (Abschnitt 4.3) werden der Tangentenwinkel $\theta(t)$ und die Krümmung $\kappa(t)$ der Trajektorie benötigt. Diese differentialgeometrischen Größen sind definiert als

$$\theta(t) = \arctan \frac{\dot{y}(t)}{\dot{x}(t)}$$

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\sqrt[3]{\dot{x}^2 + \dot{y}^2}}$$

Die Optimaltrajektorie minimiert das Integral

$$J[\mathbf{x}] = \int_{t_0}^{t_0+T} j_{\text{ablage}}(\mathbf{x}(t)) + j_{\text{geschw}}(\mathbf{x}(t)) + j_{\text{beschl}}(\mathbf{x}(t)) + j_{\text{ruck}}(\mathbf{x}(t)) + j_{\text{gierrate}}(\mathbf{x}(t)) dt.$$

Wir erläutern nun die einzelnen Summanden, über die integriert wird. Alle Summanden sind mit einer Gewichtung w_{\dots} versehen.

Der Term

$$j_{\text{ablage}}(\mathbf{x}(t)) = w_{\text{ablage}} \left| \frac{1}{2} (d_{\text{links}}(\mathbf{x}(t)) + d_{\text{rechts}}(\mathbf{x}(t))) \right|^2$$

bewirkt, dass die Trajektorie möglichst in der Mitte zwischen den beiden Korridorrrändern verläuft. Die Funktionen d_{links} und d_{rechts} sind die vorzeichenbehafteten Abstandsfunktionen zu den Korridorrrändern, der Abstand ist also positiv für alle Punkte links des Randes, und negativ für alle Punkte rechts davon. Der Term

$$j_{\text{geschw}}(\mathbf{x}(t)) = w_{\text{geschw}} |\mathbf{v}_{\text{soll}}(\mathbf{x}(t)) - \dot{\mathbf{x}}(t)|^2$$

enthält den quadratischen Fehler der vektoriellen Geschwindigkeit der Trajektorie im Vergleich zu einem Referenzgeschwindigkeitsvektor \mathbf{v}_{soll} . Der Vektor \mathbf{v}_{soll} wird in seinem Betrag v_{soll} von der Verhaltensentscheidung bestimmt und entspricht einer der Karte entnommenen Höchstgeschwindigkeit. Die Richtung des Referenzgeschwindigkeitsvektors ist orthogonal zu den Gradienten der Korridor-Abstandsfunktionen, die Sollfahrtrichtung ist also parallel zu den Korridorrändern. Es gilt

$$\mathbf{v}_{\text{soll}}(\mathbf{x}(t)) = v_{\text{soll}} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{1}{2} (\nabla d_{\text{links}}(\mathbf{x}(t)) + \nabla d_{\text{rechts}}(\mathbf{x}(t))).$$

Die beiden bis jetzt beschriebenen Terme geben das Wunschverhalten der Trajektorie vor: Sie soll in der Mitte des Korridors verlaufen und Fortschritt entlang des Korridors erzielen. Sie wirken damit den folgenden Termen entgegen, die fahrdynamisch motiviert sind und eine Glättung der Trajektorie bewirken. Der Term

$$j_{\text{beschl}}(\mathbf{x}(t)) = w_{\text{beschl}} |\ddot{\mathbf{x}}|(t)^2$$

unterdrückt starke Beschleunigungen in Quer- und Längsrichtung, und damit die Kräfte, die auf die Passagiere einwirken. Der Ruckterm

$$j_{\text{ruck}}(\mathbf{x}(t)) = w_{\text{ruck}} |\ddot{\mathbf{x}}|(t)^2$$

erzwingt zusätzliche Glätte der Trajektorie, indem schnelle Änderungen der Beschleunigung unterdrückt werden. Die Unterdrückung von Beschleunigung und Ruck kann noch nicht verhindern, das beim Fahren entlang der Trajektorie schnelle Richtungsänderungen erfolgen. Hierzu wird ein Term in den Integranden eingeführt, der hohe Drehraten dämpft:

$$j_{\text{gierrate}}(\mathbf{x}(t)) = w_{\text{gierrate}} \dot{\theta}(t)^2.$$

Neben dem beschriebenen Optimalitätskriterium müssen bei der Berechnung der Trajektorie noch Nebenbedingungen eingehalten werden. Dies sind einerseits solche, die Kollisionsfreiheit und das Verbleiben im Fahrkorridor zusichern. Sie wurden im vorangehenden Abschnitt beschrieben, und wir bezeichnen sie als *externe* Nebenbedingungen. Darüber hinaus gibt es Nebenbedingungen, die sich aus der Fahrzeugkinematik und -dynamik ergeben. Wir nennen sie *interne* Nebenbedingungen. Sie betreffen die Krümmung und Beschleunigung der Trajektorie, die durch κ_{max} und a_{max} limitiert werden:

$$|\kappa(t)| < \kappa_{\text{max}}$$

$$\|\ddot{\mathbf{x}}(t)\| < a_{\text{max}}.$$

Zur Lösung wird dieses Variationsproblem nach der Methode finiter Differenzen in ein Extremwertproblem überführt. Dies muss unter Berücksichtigung der beschriebenen Nebenbedingungen gelöst werden. Sowohl die Zielfunktion als auch die Nebenbedingungen stellen nichtlineare Gleichungen und Ungleichungen dar. Als Optimierungsverfahren kommt deshalb die Methode des sequentiellen quadratischen Programmierens (SQP) [8] zum Einsatz.

4.3 Trajektorienregelung

Der Fahrzeugregler setzt die geplante Trajektorie in Stellbefehle für die Fahrzeugaktoriik um. Durch Rückführung der Posenschätzung aus der Lokalisierung führt er das Fahrzeug auf der Solltrajektorie entlang. In diesem Abschnitt wird nur auf den verwendeten Querregler eingegangen, zu Längsregelung wurde der in [5] beschriebene modellprädiktive Regler verwendet.

Die Querregelung ist folgendermaßen kaskadiert: Zunächst wird aus Gierratenvorsteuerung und Ablage-Regelung die Sollgierrate ermittelt. Dann wird diese Sollgierrate mit einem modellbasierten Lenkwinkelregler eingeregelt.

Die Gierratenvorsteuerung ermittelt an einem Vorausschaupunkt die erforderliche Gierrate, um der geplanten Trajektorie zu folgen, während die Ablage-Regelung eine Querabweichung zu dieser ausregelt. Aus Gründen der Stabilität der Ablage-Regelung wird nicht direkt auf die Ablage geregelt. Stattdessen wird basierend auf der Ablage ein Sollwert für den Kurswinkel relativ zur Trajektorie bestimmt, über den das Annäherungsverhalten an die Trajektorie festgelegt wird. Mit einem P-Regler für den relativen Kurswinkel wird dann die zu überlagernde Gierrate bestimmt. Aus diesem wird dann mit einem P-Glied die zu überlagernde Gierrate berechnet.

Im modellbasierten Lenkwinkelregler wird anschließend der für die Umsetzung der Sollgierrate erforderliche Lenkwinkel berechnet, und zwar über den Zusammenhang, der sich für das dynamische Einspurmodell im Fall stationärer Kreisfahrt ergibt. Um stationäre Genauigkeit des geschlossenen Regelkreises zu erhalten, wird der Modellzusammenhang durch Beobachtung von Lenkwinkel und Gierrate nachgeführt. Hierzu wird ein zusätzlicher I-Anteil in Form eines Lenkwinkel-Offsets eingeführt.

5 Diskussion

Mit dem vorgestellten System gelang es, die anspruchsvolle Route in sechs Teilabschnitten ohne Fahrereingriff zu durchfahren. Dabei wurden unter anderem Kreuzungen unter Beachtung der Vorfahrtsregeln überfahren, Kreisverkehre durchfahren, in fließenden Verkehr eingefädelt und Entgegenkommer auch an Engstellen richtig vom Fahrzeug behandelt. Wir glauben, dass das Gesamtsystem einen Fortschritt gegenüber dem bisherigen Stand der Technik darstellt, sowohl im Bezug auf die Variabilität der bewältigten Verkehrssituationen, als auch auf den sensorischen Aufwand, der niedriger ist als bei sonst vergleichbaren Projekten.

Die Kombination aus einer detaillierten Straßenkarte und einer kartenrelativen, videobasierten Lokalisierung hat sich bewährt, da einige statische Eigenschaften des Verkehrsraumes zur Zeit noch nicht gut genug sensorisch erfasst werden können. Dies betrifft einerseits geometrische Informationen, zum Beispiel die Lage von Fahrstreifen in Bereichen, in denen diese nicht gut sichtbar markiert sind, zum Beispiel innerhalb von Kreuzungen. Noch schwieriger sensorisch zu interpretieren sind Relationen zwischen diesen geometrischen Elementen, zum Beispiel, welcher Fahrstreifen einem anderen gegenüber vorfahrtsberechtigt ist, oder welche Ampelanlagen einem Fahrstreifen zugeordnet sind.

Die Verwendung einer solchen Karte birgt natürlich auch Nachteile. So kann, wenn nur lang genug Zeiträume betrachtet werden, streng genommen kein Teil des Verkehrsraumes als wirklich statisch und unveränderbar angenommen werden. Aber auch in kurzen

Zeitspannen kann das System durch vorübergehende Veränderungen im Verkehrsraum kompromittiert werden, zum Beispiel durch Baustellen und Umleitungen.

Wir streben deshalb für die Zukunft eine Verbesserung des Systems in zweierlei Hinsicht an: Einerseits soll die Sensorik leistungsfähiger werden, wodurch weniger Informationen in der Karte abgelegt werden müssen. Andererseits sollen die Verfahren zur Kartierung so erweitert werden, dass eine Anpassung der Karte im Fahrbetrieb durch das Fahrzeug selbst vorgenommen wird. Hierbei ist auch denkbar, durch neue Methoden zur Kommunikation und Datenhaltung die Sensordaten vieler Fahrzeuge zu einer Gesamtkarte zusammenzuführen.

Literatur

- [1] The rails port - OpenStreetMap wiki.
- [2] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. In *Combinatorial Optimization*, pages 31–33, 2001.
- [3] MarkusENZweiler, Pierre Greiner, Carsten Knoppel, and Uwe Franke. Towards multi-cue urban curb recognition. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 902–907. IEEE, 2013.
- [4] U. Franke, D. Pfeiffer, C. Rabe, C. Knoepfel, M. Enzweiler, F. Stein, and R. G. Herrtwich. Making Bertha see.
- [5] Andreas Geiger, Frank Moosmann, Martin Lauer, Benjamin Ranft, Holger Rapp, and Julius Ziegler. *Team AnnieWAY's Entry to GCDC (2011)*.
- [6] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2):47–57, June 1984.
- [7] David Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, June 1987.
- [8] A. K. Kaw and E. E. Kalu. *Numerical Methods with Applications*. <http://www.autarkaw.com>, second edition, 2010.
- [9] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller. Urban localization with camera and inertial measurement unit. In *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australien, 2013.
- [10] H. Lategahn and C. Stiller. City gps using stereo vision. In *IEEE International Conference on Vehicular Electronics and Safety*, Istanbul, Türkei, Juli 2012.
- [11] H. Lategahn and C. Stiller. Experimente zur hochpräzisen landmarkenbasierten Eigenlokalisierung in unsicherheitsbehafteten digitalen Karten. In *FAS Workshop*, Walting, Altmühltal, 2012. UNiDAS e.V.
- [12] Frederik Ramm, Jochen Topf, and Steve Chilton. *OpenStreetMap*. UIT Cambridge, 2007.

-
- [13] Markus Schreiber, Carsten Knoppel, and Uwe Franke. Laneloc: Lane marking based localization using highly accurate maps. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 449–454. IEEE, 2013.

Crowdsourcing als Enabler für verbesserte Assistenzsysteme: Ein generischer Ansatz zum Erlernen von Kreuzungsparametern

Christian Ruhhammer* , Atanasko Atanasov* , Felix Klanner*
und Christoph Stiller†

Zusammenfassung: Eine Reihe prototypisch entwickelter Fahrerassistenzsysteme benötigen detaillierte Kartendaten wie zum Beispiel Kreuzungsparameter, um die Sicherheit und Effizienz im Straßenverkehr signifikant zu erhöhen. Aktuelle Straßenkarten enthalten jedoch nicht ausreichend viele Daten. Im vorliegenden Manuskript wird ein generischer Ansatz zum Erlernen von Kreuzungsparametern vorgestellt. Dieser basiert auf der Auswertung von Daten, die mit einer Fahrzeugflotte gesammelt wurden. In Summe stehen für die Entwicklung 141.173 reale Kreuzungsüberfahrten zur Verfügung. Als Anwendungsbeispiel wird mit der entwickelten Methode die Haltelinienposition und tageszeitliche Veränderung des Verkehrsflusses an 23 Kreuzungsarmen gelernt. Die Methode wird anhand realer Messdaten evaluiert.

Schlüsselwörter: Backend, Crowdsourcing, Digitale Karte, Trajektorien

1 Von Crowdsourcing profitierende Fahrerassistenzsysteme

Crowdsourcing bezeichnet in diesem Manuskript die Auslagerung der Datenaquise und -vorverarbeitung auf eine Vielzahl von Fahrzeugen. Erfasste Daten sind beispielsweise die GPS-Position des Fahrzeugs und ein globaler Zeitstempel. Diese Daten werden über das Internet an eine zentrale, im Folgenden als Backend bezeichnete Recheneinheit übertragen.

Bereits heute sind eine Reihe von Assistenzsystemen verfügbar, die auf Crowdsourcing und eine Datenübertragung zum Backend via Mobilfunk zurückgreifen. Beispiele dafür sind die Darstellung des aktuellen Verkehrsflusses im Fahrzeug (zum Beispiel *BMW Real Time Traffic Information*, *Audi Verkehrsinformationen online*), im Internetbrowser (zum Beispiel *Google Maps Traffic*) oder über Smartphone Apps (zum Beispiel *INRIX Traffic*). Ein weiterer Onlinedienst der aktuellen Fahrzeuggenerationen ist das Streaming von Musik. Die Anwendung internetbasierter Anwendungen im Fahrzeug fokussiert sich heutzutage auf die Bereiche Navigation, lokale Gefahrenstellen und Entertainment.

*Christian Ruhhammer ist Doktorand, Atanasko Atanasov Masterand und Felix Klanner Projektleiter bei der BMW Forschung und Technik GmbH, Hanauer Str. 46, 80992 München (E-Mail: {christian.ruhhammer, atanasko.atanasov, felix.klanner}@bmw.de).

†Christoph Stiller ist Leiter des Instituts für Mess- und Regelungstechnik am Karlsruher Institut für Technologie, Engler-Bunte-Ring 21, 76131 Karlsruhe (E-Mail: stiller@kit.edu).

Zukünftig werden immer mehr Fahrzeuge über das Internet mit Backendsystemen verbunden sein. Dadurch ergeben sich neben den bisher verfügbaren Diensten eine Vielzahl weiterer Möglichkeiten, neue Assistenzsysteme zu entwerfen und vorhandene Systeme zu verbessern. Eine Schlüsselrolle für die Weiterentwicklung von Fahrerassistenzsystemen hat hierbei die Erhöhung der Detailtiefe von digitalen Karten.

Im Rahmen aktueller Forschungsarbeiten werden fehlende Details in digitalen Karten durch manuelles Vermessen und Kartieren mit speziellen Fahrzeugen erzeugt, was einen hohen Aufwand generiert. Eine effizientere Methode zur Ermittlung der benötigten, detaillierteren Information ist Crowdsourcing. Dabei erzeugen eine Vielzahl von Fahrzeugen mit einfacher Sensorik Daten, aus denen durch intelligente Aggregation gewünschte Detailinformationen über das Fahrzeugumfeld extrahiert werden. Diese Informationen werden den einzelnen Fahrzeugen wieder zur Verfügung gestellt und für Fahrerassistenzsysteme bis hin zum hochautomatisierten Fahren verwendet. Ein möglicher Weg, die Daten zu sammeln und die extrahierten Informationen zu verbreiten, ist das Internet.

Die Idee, aus einer Vielzahl aufgezeichneter GPS-Trajektorien Straßenmodelle zu extrahieren, wurde bereits in früheren Arbeiten aufgegriffen [1] [2]. Ziel der Arbeiten war es, Spurinformatoren aus aufgezeichneten Positionsdaten zu extrahieren. Des Weiteren wurde in [3] vorgestellt, wie anhand von Kreuzungsüberfahrten erkannt werden kann, ob an der Kreuzung eine Lichtsignalanlage existiert. Weitergehende Informationen über Lichtsignalanlagen wurden in [4] und [5] anhand von Geschwindigkeitsprofilen extrahiert, um effizientere Fahrstrategien zu ermöglichen. Die bisherigen Arbeiten auf dem Gebiet des Crowdsourcings beziehen sich hauptsächlich auf die geometrische Beschreibung des Straßenverlaufs sowie auf die Informationsextraktion in Bezug auf Lichtsignalanlagen.

Mögliche Anwendungen von detaillierteren Kreuzungsparametern für Fahrerassistenzsysteme werden in Abschnitt 2 näher betrachtet. Die in aktuellen Forschungsarbeiten vorgestellten Ansätze fokussieren sich außerdem auf die Bestimmung von nur einem Kreuzungsparameter. Daher wird in Abschnitt 3 ein generischer Ansatz zum gleichzeitigen Erlernen von mehreren Kreuzungsparametern vorgestellt. Das Verfahren wird für die Ermittlung der Haltelinienposition und den von der Tageszeit abhängigen Verkehrsfluss an Kreuzungsarmen mit Lichtsignalanlagen angewendet und evaluiert. Abschließend werden die gewonnenen Erkenntnisse in Abschnitt 4 zusammengefasst und noch offene Forschungsfelder adressiert.

2 Kreuzungsparameter für die Anwendung in Fahrerassistenzsystemen

Eine Reihe von Assistenzsystemen lassen sich mit weiteren Details in Navigationskarten verbessern. Folgende Systeme profitieren von Kartenattributen wie der Vorfahrtsregel an Kreuzungen oder der Position von Haltelinien:

- A-priori-Wissen für kamerabasierte Assistenzsysteme wie der Vorfahrtsassistent,
- Anpassung der Betriebsstrategie der Motor-Start-Stopp-Automatik,
- Eingangsgröße für Fahrerabsichtserkennung [6] zur Kollisionsvermeidung,
- Haltelinien als Landmarken zur besseren Längspositionierung für die Navigation.

Kamerabasierte Systeme mit einer lokalen Haltelinienenerkennung [7] profitieren von der Information über die absolute Position von Haltelinien. Damit kann abhängig von der Position des Fahrzeugs plausibilisiert werden, ob das aktuelle Sichtfeld der Kamera eine Haltelinie beinhaltet. Zudem gibt es die Möglichkeit, den Suchbereich im Bild abhängig vom geschätzten Abstand zur Haltelinie einzuschränken. Durch dieses Vorgehen lassen sich Fehl- und Falschklassifikationen erkennen.

Die Verbesserung der Betriebsstrategie der Motor-Start-Stopp-Automatik ist eine weitere potentielle Anwendung von Kreuzungsparametern. Das Abstellen des Motors steigert die Effizienz erst nach einer gewissen Standzeit. Aus diesem Grund ist es wichtig, bereits zu Beginn der Standzeit eine Schätzung über die Dauer des Anhaltevorgangs zu treffen. Dazu muss der Grund des Anhaltevorgangs interpretiert werden, wozu weitere Informationen über die Umgebung des Fahrzeugs hilfreich sind. An einer Kreuzungseinfahrt mit Stopp-Schild ist ein Anhaltevorgang im Mittel von kürzerer Dauer als bei Kreuzungen mit Lichtsignalanlage. Außerdem deutet bei Kreuzungen mit Lichtsignalanlage ein Stopp nach einer Haltelinie auf ein Anhalten kürzerer Dauer hin.

Es wird davon ausgegangen, dass die genannten Assistenzsysteme eine Haltelinienge- nauigkeit von 3 bis 5 Metern benötigen.

<i>Kategorie</i>	<i>Anforderung</i>	<i>Anwendung</i>
1	± 3 m	Vorfahrtsassistent
2	± 5 m	Motor-Start-Stopp Automatik

3 Extraktion von Kreuzungsparametern durch iterative Simulation und Analyse von GPS-Trajektorien

Als Grundlage für die Entwicklung der folgenden Methoden dienen aufgezeichnete Daten von Fahrzeugflotten. Neben einer eigenen Flotte wurden aufgezeichnete Daten aus dem nationalen Forschungsprojekt sim^{TD} [8] verwendet. Aus diesen Daten werden zusammenhängende Geschwindigkeits- und Positionswerte von einzelnen Kreuzungsüberfahrten extrahiert. Damit stehen insgesamt 141.173 Kreuzungsüberfahrten zur Verfügung.

3.1 Parameterschätzung durch iterative Simulation

Eine Möglichkeit der Extraktion von Kreuzungsparametern aus GPS-Trajektorien ist die Analyse von beobachtbaren Größen in den Daten. Ein Beispiel für eine derartige Größe ist die Verteilung der Haltepositionen an einem Kreuzungseingang. Diese Verteilung ist von mehreren Eigenschaften der Kreuzung abhängig:

- der Anzahl und Konnektivität der Fahrspuren, sprich der Kreuzungstopologie,
- die Wahrscheinlichkeiten für Abbiege- und Geradeausfahrmanöver,
- die Position der Haltelinie,
- der absolute Verkehrsfluss an den Kreuzungseingängen.

Eine einfache Kreuzungstopologie und -geometrie wird aus dem frei verfügbaren Kartenmaterial von OpenStreetMap (OSM) [9] automatisiert extrahiert. Weitere Parameter, wie die Manöverwahrscheinlichkeiten, werden direkt aus den Überfahrten bestimmt. Die Position der Haltelinie oder der absolute Verkehrsfluss an den Kreuzungseingängen können weder direkt beobachtet werden, noch liegen externe Quellen für diese Information vor.

Ein Ansatz zur Ermittlung von derartigen Parametern einer Kreuzung ist ein maschinelles Lernverfahren, das sämtliche Abhängigkeiten zu den jeweiligen Parametern berücksichtigt. Aufgrund der hohen Zahl unterschiedlicher Eigenschaften und individuellen Besonderheiten von Kreuzungen benötigt man dazu eine noch größere Menge an Trainingsdaten, was eine Umsetzung in der Praxis erschwert.

3.2 Iterative Simulation

In dieser Arbeit wird ein Verfahren vorgestellt, welches mittels iterativer Simulation des Verkehrs an einer Kreuzung deren Parameter schätzt. Abbildung 1 zeigt den Aufbau der Methode. Gewisse Eingangsparameter für die Simulation werden als a-priori-Wissen direkt aus einer Menge an Kreuzungsüberfahrten, dem Realdatensatz, extrahiert. Zudem dient eine OSM-Karte als Basis für die Geometrie und Topologie der Kreuzung. Für weitere Parameter, die nicht direkt beobachtet werden können, wird ein diskreter Wertebereich definiert. Diese Parameter werden nach jedem Simulationsschritt variiert und für jede mögliche Kombination wird eine Simulation der Kreuzung durchgeführt.

Nach jeder Iteration werden aus der simulierten Menge von Kreuzungsüberfahrten Merkmale beobachtet, welche direkt von den gesuchten Parametern beeinflusst werden. Ein Beispiel dafür zeigt das rote Histogramm in Abbildung 1. Diese Merkmale werden parallel dazu einmalig auch aus den realen Verkehrsdaten extrahiert, entsprechend dem blauen Histogramm in Abbildung 1. Das Ergebnis aus der Simulation wird für jeden Parametersatz mit den realen Daten verglichen. Die prädizierten Parameter ergeben sich aus der besten Übereinstimmung der extrahierten Merkmale zwischen Simulation und realen Daten.

3.3 Kreuzungsmodell

Als Basis für die Verkehrssimulation einer Kreuzung wird der mikroskopische Verkehrssimulator *SUMO* [10] verwendet. Als Eingang in die Simulation wird eine Menge an Parametern als Modell M zur Beschreibung einer Kreuzung verwendet. Ein Teil des Modells besteht aus einem Ausschnitt OM der OpenStreetMap und repräsentiert in einfacher Darstellungsform die Geometrie und die Topologie der Kreuzung. Der relative Verkehrsfluss und die Manöverwahrscheinlichkeit für Pfade durch die Kreuzung werden in Abhängigkeit von der Tageszeit direkt geschätzt. Voraussetzung für diese Schätzungen ist, dass die verwendeten Daten von einer repräsentativen Stichprobe aller Fahrzeuge, die die Kreuzung überfahren, stammen.

Zur Ermittlung der relativen Verteilung des Verkehrsflusses der einzelnen Kreuzungseingänge zueinander wird eine Kreuzung mit einer Menge E an Eingängen und einer Menge A an Ausgängen betrachtet. In einem Zeitintervall $T = [\tau_1; \tau_2]$ berechnet sich der relative Verkehrsfluss ρ_e eines Kreuzungseingangs $e \in E$ über die Anzahl der Fahrzeuge

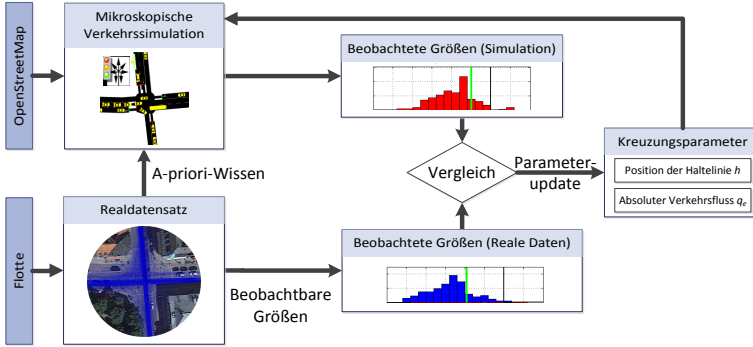


Abbildung 1: Iterative Simulation zur Schätzung von Kreuzungsparametern basierend auf einer Vielzahl an Kreuzungsüberfahrten.

$N_x(t \in T)$ aller Kreuzungseingänge $x \in E$ zu

$$\rho_e = \frac{N_e(t \in T)}{\sum_{x \in E} N_x(t \in T)}$$

Neben der relativen Verteilung des Verkehrsflusses wird auch die Manöverwahrscheinlichkeit für einzelne Pfade durch eine Kreuzung direkt aus einer Vielzahl von Überfahrten ermittelt. Ein Manöver über eine Kreuzung ist definiert durch einen Pfad von einem Eingang $e \in E$ zu einem Ausgang $a \in A$. Die Wahrscheinlichkeit für ein Manöver $P(m_{ea})$ berechnet sich anhand der Anzahl der Fahrzeuge N_{ey} auf allen Pfaden mit gleichem Eingang e :

$$P(m_{ea}) = \frac{N_{ea}(t \in T)}{\sum_{y \in A} N_{ey}(t \in T)}$$

Die direkt geschätzten Parameter ergeben eine Menge

$$K = \{OM, F, P(M)\}, \text{ mit } F = \{\rho_e | e \in E\} \text{ und } M = \{m_{ea} | e \in E, a \in A\}$$

von konstanten Kreuzungsparametern.

Weitere Kreuzungsparameter, wie die Positionen h von Haltelinien, oder der absolute Verkehrsfluss q_e an jedem Kreuzungseingang $e \in E$, können nicht direkt aus den Daten extrahiert werden, weshalb diese mit dem beschriebenen Verfahren geschätzt werden.

Der absolute Verkehrsfluss an den Kreuzungseingängen lässt sich jedoch mit den relativen Verkehrsflüssen und einem unbekanntem eindimensionalen Faktor c_q berechnen:

$$q_e = c_q \cdot \rho_e$$

Um den absoluten Verkehrsfluss an den Kreuzungseingängen zu schätzen muss daher nur der eindimensionale Faktor c_q geschätzt werden. Für die zu schätzenden Parameter werden diskrete Wertebereiche W festgelegt. Dadurch ergibt sich für diese variablen

Parameter ein Raum

$$PR = \{(h, c_q) | h \in W_h, c_q \in W_{c_q}\}.$$

Ein Kreuzungsmodell M ergibt sich aus der Kombination der konstanten Parameter K und eines variablen Parametersatzes V :

$$M = (K, V), \text{ mit } V \in PR.$$

Insgesamt existieren

$$|PR| = |W_h| \cdot |W_{c_q}|$$

unterschiedliche Kreuzungsmodelle.

3.4 Merkmalsextraktion und -vergleich

Für jedes einzelne Kreuzungsmodell wird eine Simulation durchgeführt. Je nach Menge der verfügbaren Überfahrten, wird ein entsprechender Zeitraum für die Simulation gewählt. Für die vorliegenden Daten wurden die konstanten Parameter für einen durchschnittlichen Wochentag mit acht Zeitslots unterschiedlicher Parametrisierung extrahiert. Über einen Zeitraum von drei Stunden stehen mit dem vorliegenden Datensatz genügend Daten zur Verfügung um die Parameter zu schätzen. Entsprechend dieser Parameter wird jeweils im selben Zeitraum eine Kreuzung simuliert.

Während jeder Simulation werden von zufällig gewählten Fahrzeugen Merkmale extrahiert. Im konkreten Fall zur Schätzung der Haltelinienposition und des Verkehrsflusses, dient die Dichtfunktion der Haltepositionen von Fahrzeugen entlang der Fahrtrichtung als Merkmal. Dazu werden sämtliche Haltepositionen im Kreuzungsbereich auf die vorliegende OSM-Karte abgebildet. Ebenso wird der Mittelpunkt der Kreuzung als Referenz-

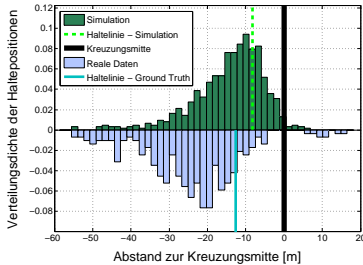


Abbildung 2: Vergleich der angenäherten Dichtfunktionen der Haltepositionen aus einem Simulationslauf (oberes Histogramm), mit der Dichtfunktion aus dem realen Datensatz (unteres Histogramm).

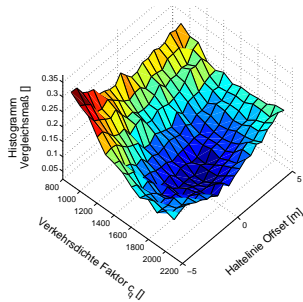


Abbildung 3: Ergebnisse des Histogrammvergleichs für jeden einzelnen Parametersatz, beziehungsweise jeden einzelnen Simulationslauf.

punkt auf die Karte abgebildet. Die Dichtefunktion der Haltepositionen wird dadurch eindimensional durch ein Histogramm angenähert.

Für sämtliche Simulationen, sowie für die Daten aus den realen Kreuzungsüberfahrten, wird für jeden Kreuzungseingang ein Histogramm für die Verteilung der Haltepositionen erstellt. Als Beispiel zeigt Abbildung 2 eine simulierte Verteilung der Haltepositionen. Die zugehörige gestrichelte vertikale Linie entspricht der gesetzten Haltelinienposition in dieser Simulation. Im Vergleich dazu repräsentiert das negative Histogramm die Verteilung der Haltepositionen in den realen Daten. Die türkise, vertikale Linie entspricht dem wahren Wert der Haltelinienposition. Die Simulation unterschiedlicher Parametersätze bewirkt eine Veränderung des Histogramms.

Die simulierten Histogramme werden jeweils mit dem Histogramm aus den realen Überfahrten verglichen. Anhand statistischer Methoden zum Vergleich von Histogrammen, wird nach der Simulation für jedes Kreuzungsmodell, ein Vergleichsmaß berechnet. Dieser Vorgang entspricht bei Betrachtung eines Kreuzungseingangs und genau einem Kreuzungsmodell der Abbildung

$$\text{compare} : M \rightarrow \mathbb{R}.$$

Für jedes Kreuzungsmodell mit einer bestimmten Belegung der zu schätzenden Parameter ergibt sich ein reeller Wert des Vergleichsmaßes. Je niedriger dieser Wert ist, umso besser stimmen die Verteilungen überein. Der translatorische Unterschied der beiden Histogramme in Abbildung 2 ist deutlich ersichtlich. Dies ist gleichbedeutend mit einer Abweichung der Haltelinienpositionen.

Abbildung 3 zeigt ein Ergebnis des Histogrammvergleichs für einen Kreuzungseingang und für jeden einzelnen simulierten Parametersatz (in diesem Fall ist der Parameterraum zweidimensional). Für jeden einzelnen Punkt auf der Oberfläche wird ein Vergleich gemäß Abbildung 2 durchgeführt. Für das beste Vergleichsergebnis, also dem minimalen Wert auf der z -Achse im Bild, ergibt sich das zugehörige Kreuzungsmodell m_{min} mit den entsprechenden Parametern zu

$$\exists m_{min} : \forall m' \in M : \text{compare}(m_{min}) \leq \text{compare}(m').$$

3.5 GPS-Messrauschen

Die realen Messdaten entsprechen rohen GPS-Messungen, welche mit einem starken Messrauschen beaufschlagt sind. Um vergleichbare Verteilungen für die Haltepositionen der Fahrzeuge zwischen Simulation und Realität zu erhalten, müssen auch die simulierten Daten verrauscht werden. Im Rahmen dieser Arbeit wird das Rauschen als normalverteilt modelliert.

Aufgrund der unterschiedlichen Abhängigkeiten des GPS-Messrauschens, wird für jeden Kreuzungseingang separat eine geeignete Standardabweichung $\sigma_{GPS} \in W_\sigma$ ermittelt. Dazu wird die Standardabweichung im Rahmen dieser Arbeit als weiterer Parameter für die Minimierung der Vergleichskosten der Histogramme verwendet gemäß

$$\text{compareSig} : M, \sigma_{GPS} \rightarrow \mathbb{R}.$$

Durch Variation der Standardabweichung erhält man nach der Minimierung der Vergleichskosten, basierend auf den einzelnen Simulationsergebnissen, unterschiedlich ideale Kreuzungsparameter. Zur Darstellung der Abhängigkeit der Schätzung vom GPS-Messrauschen, zeigt Abbildung 4 exemplarisch für eine Schätzung zum einen den wahren

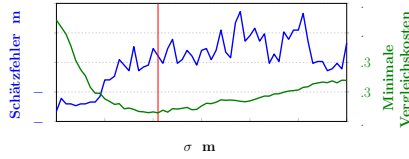


Abbildung 4: Abhängigkeit der GPS-Standardabweichung zum Schätzfehler der Haltelinie, sowie zu den absoluten minimalen Vergleichskosten zwischen Simulation und realem Datensatz für einen exemplarischen Kreuzungsarm.

Schätzfehler, sowie das jeweilige minimale Vergleichsmaß in Abhängigkeit von σ_{GPS} . Daran ist ersichtlich, dass eine Minimierung des Vergleichsmaßes unter Berücksichtigung des weiteren Parameters σ_{GPS} eine Verbesserung der Schätzung zur Folge hat. Es existiert damit eine Kombination aus einem Kreuzungsmodell m_{res} und einer Standardabweichung σ_{res} für welche der beste Wert des Vergleichsmaßes erzielt wird,

$$\exists m_{\text{res}} \sigma_{\text{res}} : \forall m' \in M \forall \sigma' \in W_{\sigma} : \text{compareSig}(m_{\text{res}}, \sigma_{\text{res}}) \leq \text{compareSig}(m', \sigma').$$

Diese Parameterkombination entspricht dem Schätzungsergebnis für einen Kreuzungseingang.

Neben der Methode, zunächst für sämtliche diskreten Parameterkombinationen eine Simulation durchzuführen, kann optional auch ein Optimierungsverfahren angewendet werden, um die benötigte Zeit zu verringern. Da es sich um ein nichtlineares Optimierungsproblem handelt, müssen Methoden verwendet werden, welche keine Berechnung des Gradienten der Vergleichsfunktion benötigen. Ein Beispiel für ein geeignetes Verfahren ist der Genetische Algorithmus [11].

3.6 Evaluierung und Diskussion

Zur Evaluierung der Haltelinienpositionsprädiktion mittels iterativer Simulation wurden Daten von insgesamt sechs Kreuzungen mit Lichtsignalanlagen gesammelt und 23 Kreuzungseingänge ausgewertet. Von diesen Eingängen wurden jeweils die realen Positionen der Haltelinien als Referenz hochgenau vermessen. Aufgrund des nichtdeterministischen Verhaltens des modulierten GPS-Rauschens, wurden die geschätzten Positionen jeweils 100-mal berechnet und anschließend gemittelt. Abbildung 5 zeigt die Ergebnisse der Evaluierung in Abhängigkeit von unterschiedlichen, aus der Statistik bekannten Distanzmaßen zwischen Histogrammen. Die Auswertung zeigt, dass sich mehrere Maße für eine Schätzung mit einem Interquartilsabstand von ± 1 m eignen. Aufgrund der begrenzten Genauigkeit der verwendeten Lokalisierungslösung ist dieses Ergebnis durchaus plausibel. Der konstante positive Offsetfehler ergibt sich daraus, dass die meisten Fahrzeuge nicht direkt auf der Haltelinie halten, sondern vor der Linie. Diese Abweichung könnte auf Basis einer Menge von Haltelinien geschätzt und für weitere Kreuzungen korrigiert werden.

Die Ergebnisse zeigen, dass die Anforderungen aus Abschnitt 2 unter Anwendung des *Match Distance* Vergleichsmaßes bei den vorliegenden Testdaten wie folgt erfüllt werden:

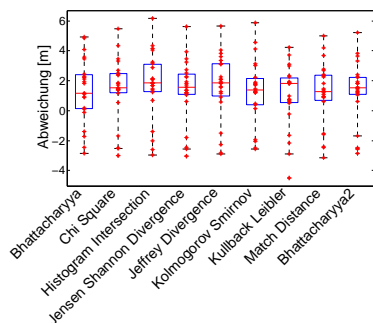


Abbildung 5: Evaluierungsergebnisse von 23 Haltelinien-schätzungen (Basis: 141.173 Überfahrten) unter Verwendung unterschiedlicher Histogramm-Distanzmaße.

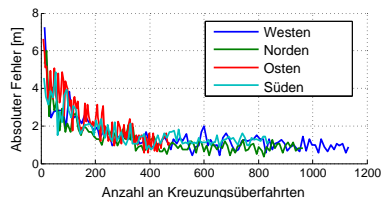


Abbildung 6: Absoluter Fehler von vier Haltelinienposition (W, N, O, S) einer exemplarischen Kreuzung, abhängig von der Anzahl der verwendeten Überfahrten.

Kategorie	Anforderung	Erfüllungsgrad
1	± 3 m	83 %
2	± 5 m	100 %

Neben der maximal erreichbaren Genauigkeit mit dem vorliegenden Datensatz wurde außerdem eine Untersuchung für die Abhängigkeit des absoluten Fehlers von der Anzahl der verwendeten Kreuzungsüberfahrten durchgeführt. Das Ergebnis dieser Evaluierung ist in Abbildung 6 exemplarisch für eine Kreuzung mit vier Kreuzungseingängen dargestellt. Die Analyse zeigt, dass die Verwendung von mehr als ca. 400 Überfahrten keine weitere Verbesserung der absoluten Genauigkeit bringt.

4 Zusammenfassung und Ausblick

Der vorgestellte Ansatz zeigt, dass es möglich ist, auf Basis einer Vielzahl von Kreuzungsüberfahrten Parameter einer Kreuzung zu schätzen. Datenbasis für die vorliegende Arbeit sind 141.173 Überfahrten. Aus diesen Daten wurden als konkrete Anwendung der Methode, Haltelinien für 23 Kreuzungseingänge extrahiert.

Kreuzungsparameter ermöglichen die Realisierung neuartiger und die Verbesserung bereits vorhandener Fahrerassistenzsysteme. Weiterer Forschungsbedarf besteht allerdings darin, exakte Geometrien und Topologien von Kreuzungen zu ermitteln. Konkret beeinflusst beispielsweise die Anzahl an Abbiegespuren die Verteilung der Haltepositionen. OpenStreetMap enthält zwar eine Angabe über die Anzahl an Spuren, allerdings ist diese nicht verlässlich und im Kreuzungsbereich nicht detailliert genug. Auf dem Gebiet der Konfidenzschätzung sind weitere Tätigkeiten nötig, da für die meisten Fahrerassistenzsysteme das Wissen über die Verlässlichkeit der Eingangsinformation essentiell ist.

Literatur

- [1] S. Rogers, P. Langley, C. Wilson, and P. Alto, "Mining GPS Data to Augment Road Models," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999.
- [2] S. Rogers, "Creating and Evaluating Highly Accurate Maps with Probe Vehicles," in *IEEE Intelligent Transportation Systems*, 2000.
- [3] S. Hu, L. Su, H. Liu, H. Wang, and T. Abdelzaher, "SmartRoad : A Crowd-Sourced Traffic Regulator Detection and Identification System," in *Proceedings of the 12th international conference on Information processing in sensor networks*, 2013.
- [4] M. Kerper, C. Wewetzer, and M. Mauve, "Analyzing vehicle traces to find and exploit correlated traffic lights for efficient driving," *IEEE Intelligent Vehicles Symposium*, June 2012.
- [5] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve, "Learning Traffic Light Phase Schedules from Velocity Profiles in the Cloud," *International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, May 2012.
- [6] M. Liebner, C. Ruhhammer, F. Klanner, and C. Stiller, "Generic Driver Intent Inference based on Parametric Models," in *IEEE Intelligent Transportation Systems (ITSC 2013)*, 2013.
- [7] T. Marita, M. Negru, R. Danescu, and S. Nedevschi, "Stop-line detection and localization method for intersection scenarios," in *IEEE 7th International Conference on Intelligent Computer Communication and Processing*, pp. 293–298, IEEE, Aug. 2011.
- [8] C. Weiß, "V2X communication in Europe - From research projects towards standardization and field testing of vehicle communication technology," *Computer Networks*, vol. 55, pp. 3103–3119, Oct. 2011.
- [9] M. M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, pp. 12–18, Oct. 2008.
- [10] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.
- [11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, Apr. 2002.

Fences – A Unified Architecture for Mapping Static Environments for Driver Assistance Systems

Tim Kubertschak*, Mirko Maehlich* und Hans-Joachim Wuensche†

Zusammenfassung: Fusing several sensors to create a view of the static environment is tedious and laborious. Every participating sensor needs to be modeled very accurately to derive robust maps that are useful for driver assistance systems. This paper proposes therefore a new architecture to ease setting up map-building by unifying crucial parts of the process.

Schlüsselwörter: Driving Assistance Systems, Multisensor Data Fusion, Static Environment, Unified Architecture

1 Introduction

Perception of the static environment of one's own vehicle is becoming more and more important for driving assistance systems. Especially for achieving fully autonomous driving an exact knowledge of the location of buildings, pavements, parked cars and infrastructural elements like traffic lights and signs is necessary. Robust maps of obstacles and free space are required for aspects like driving path detection [1], emergency braking [2] and parking assistants [3].

In order to derive reliable and robust models of the static environment, data of several sensors need to be fused. For example, the approaches for driving path detection, emergency braking and parking mentioned above require scanning radar, scanning lidar and ultrasonic sensors. However, integrating several sensors is very tedious in current map-building processes. Every sensor needs a special conversion into a mapable representation. Simultaneously, almost every driving assistance function needs special information from the mapped surroundings.

To ease the mapping process, the *Fences*-architecture for mapping static environment is proposed. It overcomes the difficulties by unifying the whole process of sensor data fusion and information extraction.

The paper is organized as follows: In section 2, related work of other authors is presented. Section 3 describes the conventional architecture along with its difficulties. The proposed *Fences*-architecture is presented in section 4 and an application of it in section 5. Finally, some concluding remarks and future work is given in section 6.

*AUDI AG (I/EE-31), 85045 Ingolstadt (e-mail: tim.kubertschak@audi.de).

†Universität der Bundeswehr, Technik Autonomer Systeme (TAS), 85577 Neubiberg.

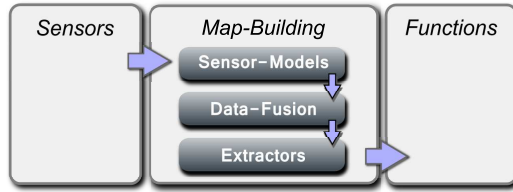


Figure 1: Conventional architecture for mapping static environment.

2 Related Work

First steps towards unified architectures for mapping have already been made unintentionally during the 1980s by researchers who used geometric approaches for map-building. For example, Chatila and Laumond [4] require some kind of generic polygonal representation to assemble their map and thus fixing the interface between sensors and the map. Coué et al. [5] created generic representations for general multi-sensor data fusion. They defined three types of models to group sensors based on their physical properties.

Dietmayer et al. [6] recognized the need of unified architectures to create efficient fusion systems. Although primarily meant for moving objects, this statement applies to static environment as well. Based on this assertion, Grewe et al. [7] developed a generic model for static environment that uses runtime-length encoded grid-maps to exchange occupancy-information between sensors, map building and driving assistance functions.

3 Conventional Map-Building Architecture

The classic process for mapping static environments as originally developed by Moravec and Elfes [8] and later used by several authors is depicted in figure 1. The architecture consists of three main components: sensors, map-building and driving assistance functions. Each of the three components are performing several tasks in order to establish a fusion-supported driving assistance system.

Sensors usually measure information about obstacles and objects that reside within their field of view as well as locations that may be assumed to be free. The measurements are provided in many different styles: multiple echoes from ultrasonic sensors, a distance for every "beam" of a ranging lidar and locations for every moving and non-moving object from ranging radar sensors. The information from all sensors need to be transformed into a representation that can be used for building maps.

The transformation is done using a sensor-model, a crucial part of the map-building component. The sensor-model is specific to each kind of sensor and even specific to sensors of the same kind from different manufacturers. It uses all information available about a sensor to transform the measurements. Afterwards, the transformed information is used to fuse it with other sensor readings in order to build a view of the static environment. The procedure for fusing data may be arbitrary; occupancy grids [8], topological maps [4] as well as other approaches may be used.

The current map of the surroundings must be further transformed, to be useful for

driving assistance functions. Driving path detection requires information about different kinds of road boundaries ([1]). Emergency braking needs to know where there are drivable free spaces in the environment and obstacles that are crossing the trajectory of the vehicle ([2]). Finally, parking assistants need the location of nearby parking lots and a drivable trajectory to get there ([3]).

However, integrating several sensors for fusing them is very tedious and currently needs to be done in the map-building component since every sensor supplies its measurements in its very own format. Thus, there must be an appropriate sensor-model for every sensor. In order to make the model as accurate as possible, very detailed knowledge about the sensor and its physical characteristics is required. Depending on the physical carrier, every sensor differs in its detection properties and field of view. The internal signal processing further alters the properties of the sensor. All this information must be gathered to build the sensor-model. But manufacturers usually don't provide all required information and the model is in general a rough approximation of the sensor's real behavior without spending too much time modeling the sensor.

Providing information about the environment for driving assistance functions is similarly demanding. Although all functions essentially need the locations of obstacles and free spaces, every function wants to have different parts of it. The map-building component has to provide relevant extractors and therefore needs function-related information.

In summary, the map-building component builds a view of the environment but needs to do a lot of work to transform available information. This transformation is usually crude, since very specific knowledge is required that is often not available.

4 Unified Map-Building Architecture

4.1 Unified Architecture

To overcome the disadvantages of conventional map-building described in the last section, a unified architecture termed *Fences* is proposed. This unification takes place at the interface between the sensors and the map-building component as well as the interface between the map-building component and the driving assistance functions that builds upon the perceived environment. A unification between all components of static environment-fusion is possible, because all participants deal with the same kind of data: obstacles and free spaces.

For example ultrasonic sensors and ranging lidars detect objects that are inside of their fields of view and are compatible with the characteristics of their physical carrier. Furthermore, the region between these sensor's origin and the obstacles may be assumed to be free to some degree of certainty. Even parts of the sensor's field of view without valid measurements may be assumed to be free, although with a much lesser certainty since there may be residing objects with unfavorable physical characteristics or in fact "negative obstacles" like ditches or holes in the ground.

When considering driver assistance functions in detail, they basically just need to know where obstacles are located and where not. Taking the examples from the last section, the actual required information is the following: Parking assistants need free spaces that look like parking lots and that may be surrounded by obstacles. The trajectory to the designated parking lots is a tube-like region that is fully drivable but may lead through

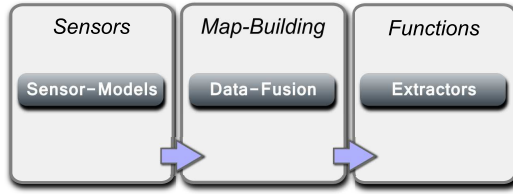


Figure 2: Unified architecture (*Fences*) for mapping static environment.

openings in walls or narrow streets. Emergency braking systems require regions that are free of obstacles and the type of boundaries that limits those regions to choose the correct evasion strategy. At last, driving path detection requires obstacles along an expected trajectory that may serve as road boundaries.

The proposed *Fences*-architecture is depicted in figure 2. Like the conventional architecture, it is composed of three parts: sensors, map-building and driving assistance function. Contrary to the conventional architecture, the location of the processing tasks have changed. The construction of the sensor-model and transformation of measurements has moved to the sensor-component, sensor data fusion is still performed in the map-building component and the extraction of function-relevant information has moved to the driving assistance function-component.

The motivation for this rearrangement is manifold. First of all, by moving the creation of the sensor-model to the sensor, the difficulty of gathering all required information to accurately model the sensor is overcome. Since the sensor-model is built by the sensor’s manufacturer, even publicly not available information can be incorporated. The manufacturer knows best which signal processing is applied that has ramifications on the sensor’s field of view, the kind of objects that can be sensed and the degree of certainty for the object’s existence and non-existence respectively. Another advantage of moving the sensor-model is the use of homogeneous interfaces among all sensors participating in sensor data fusion. Thus integrating additional sensors is becoming much more simple.

Relocating the function-specific extractors from map-building to driver assistance function has similar advantages. The developers of those functions usually know best what details in the environment are required in order to reliably implement certain functionality. Since the extractors have moved to the function-component, homogeneous interfaces for all driver assistance functions may be used as well and thus adding further driver assistance functions is again simple.

4.2 Homogeneous Interface

Besides the simplifications of integrating additional functionality and sensors, the major benefit of the proposed architecture is the use of the same homogeneous interface as input and output of the map-building component and thus unifying the whole mapping process. In this work, a topological representation for obstacles and free spaces is proposed as interface. This representation is extended with a set of attributes that further define properties of the environment. An example of a specific instance of the proposed interface

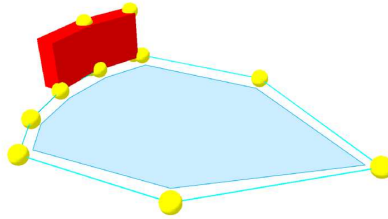


Figure 3: Topological representation of obstacles (red) and free space (blue).

is shown in figure 3.

Since the interface is a topological representation, it essentially describes relations between sites. Each entity in the environment is composed of several nodes that stand for significant features. If the features are somehow connected, the corresponding nodes are connected as well. The nodes are called *vertex* or *vertices* hereafter. All vertices of an entity are grouped. These groups form polygons and polylines, depending on which entity they represent. Polygons are used for entities whose border is fully recognized, polylines for entities where only parts of it can be seen. Polygons and polylines are called *contours*. The entire interface shares some commonalities with real fences. Each contour is basically a fence with poles that are irregularly distributed. Thus the interface is called *Fences* and the proposed architecture is entitled after it. The composition of the environment by multiple poles has also been proposed by Badino et al. [9]. However, their representation uses equally distributed poles and may not be arbitrarily attributed.

Figure 3 shows two contours built from a single sensor reading. The red one is an obstacle that is represented by a polyline since only parts of the object can be seen. The blue contour is a polygon that encloses free space. The size of the polygon is limited in parts by the sensor's field of view, in parts by the object itself. The polygon as well as the polyline consist of several vertices, that are shown as yellow spheres.

To fully represent spatial sensor readings and environments, the current interface-definition needs to be extended. Therefore a set of attributes is attached to contours and vertices. These attributes may be arbitrary, depending on the sensor set used as well as on the driver assistance functionality that should be implemented. However, a basic set of attributes is required for almost all instances of the *Fences*-architecture. These attributes are positions, positional uncertainties and existential uncertainties for all vertices and different contour-types for each contour. Some of the attributes apply to the connections between neighboring vertices, i.e. the line-segments, since they do not apply to a single vertex very well.

The position turns the topological representation into a geometrical one which is crucial for spatial structures. For positioning the vertices, several types of coordinates may be used. Depending on the particular application, Cartesian coordinates in two and three dimensions and polar coordinates may be used. It is even possible to use quite unconventional coordinates for special applications, like building a map from vertical mounted lidars.

The need for positional and existential uncertainties originate from the well known

fact that the nature of every sensor-reading is fuzzy. All measured positions are correct only up to a certain small error. In addition, the existence and non-existence of objects cannot be determined with full certainty. Depending on the sensor's physical carrier and the properties and condition of an object, free spaces and obstacles may not be detected with full certainty. Since each sensor is suffering from these issues, they apply as well to the fused environment and thus are fundamental to an effective representation of the environment.

An efficient representation requires some mechanism to describe different kinds of contours. This is achieved by adding a type-attribute to each contour. Three types are useful but others might be required as well: polylines, polygons with free space inside and polygons with free space outside. The first two types have already been discussed above. The hole-type or free space outside is necessary for objects or unknown regions that are fully enclosed by free space. Take for example an advertising pillar. It may be completely contained in the current view of the environment and emerges as a hole in the free space. These holes can only be adequately represented by using this contour-type.

Some of the attributes are also shown in figure 3. All vertices are obviously having a position attached to it. Furthermore, a positional uncertainty is attached to each obstacle which is illustrated by the thickness of each line segments. Contour-types are illustrated in this figure as well. The polyline-type is used for the red obstacle and the free space inside-type for the blue polygon.

4.3 Benefits of *Fences*

Using the architecture and the interface that has been described in the last two sections yields several advantages. The main advantage has in parts already been mentioned: A full abstraction of every component of the map-building architecture and thus offering a plug and play ability. The generic *Fences*-interface permits a very simple process of integrating additional sensors and driving assistance functions. They merely need to implement the same *Fences*-instance and can easily incorporate their unique abilities in map-generation or use the view of the vehicles environment to provide advanced functionality. There is even another possibility of abstraction that isn't as obvious as the others. By using the *Fences*-architecture, the strategy for building the map can be changed indiscernibly. It does not matter whether a Bayesian approach like [8], a topological approach like [4] and other approaches are used. Thus it is possible to use different mapping strategies for different system configurations without changing the interfaces.

Another advantage of the proposed architecture is its ability to scale to different economic circumstances. Due to the same interface between all components, the same tools can be used for debugging and diagnostic purposes. No additional costs arise for tools for specific interfaces that are valuable for single components only. If the architecture is deployed in low-cost systems with very simple driver assistance functions, it is possible to rationalize the map-building component from the whole system. No changes are necessary, since sensor and function are using the same interface for exchanging data. This ability demonstrates the robustness of the proposed architecture.

An interesting feature that may be beneficial in certain situations is the possibility to adapt the accuracy of the representation to different requirements without changing the interface or any software component that implements the interface. Thus it is possible to seamlessly reduce the amount of data that needs to be transferred between the three

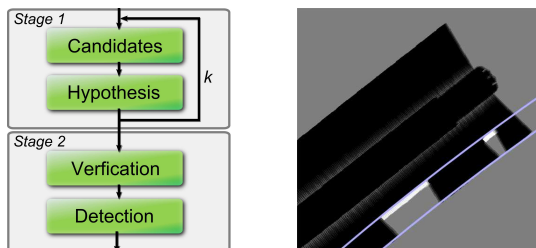


Figure 4: The scheme on the left shows the sequence diagram of the algorithm of [3]. The map on the right is a result of the readings of sonar sensors integrated over time while the vehicle moved in a right-upward way. The blue line are circles with large radii that illustrate different levels of road boundaries.

components. This adaption is done during a simplification-process that increases positional uncertainty but reduces the number of vertices of each contour. This feature of the *Fences*-architecture makes it very different to the generic environment model proposed by Grewe et al. [7]. Since they propose a run-length encoded occupancy grid representation, the amount of data compression without changing the interface is not as high as the one proposed here. Especially maps created by fusing several sensors over some period of time are probably difficult to compress in the representation if [7], since the maps are usually quite complex.

5 Application to Parking Assistant

To provide some evidence of the applicability of the proposed *Fences*-architecture, this section presents an example of a driving assistance function based on this architecture. An important part of an electronic parking assistant is the detection of parking lots in the surrounding of the vehicle, in order to have a destination for a semi-automated parking process.

In [3], Kubertschak and Mählich presented a robust algorithm for detecting parking lots on the roadside in Bayesian occupancy grids built from ultrasonic sensor readings. The algorithm is summarized in the following paragraph. Details about the generation of the Bayesian map can be found in the original publication. The sequence-diagram is shown on the left of figure 4.

The algorithm detects parking lots in two stages. In the first stage, the roadside is detected by regression using parallel circles as models. The motivation for this approach can be seen on the right of figure 4. The roadside of arbitrary streets usually follow a clothoid model and can be approximated by circles quite accurately. Especially in situations with parked cars, the road boundary consists of several levels of obstacles and edges between free and unknown regions. Those levels are found in the first stage of the algorithm. Each level is modeled by a circle with the additional constraint that the circles must be parallel. The parameters of all circles are estimated from a set of candidate points that consist of occupied cells and cells at free space edges. The estimation is done on a

probabilistic basis using the RANSAC algorithm.

The second stage of the algorithm uses the estimated model to detect potential parking lots. They are identified along the circle that is closest to the vehicle by examining transitions between free and occupied cells in the map. A parking lot is located between transitions from occupied to free cells and free to occupied cells, with a given distance between each transition.

Adapting this algorithm to *Fences* is fairly simple. There are only two steps that must be performed: define an appropriate instance of the interface and change some parts of the algorithm. The *Fences*-instance uses the following attributes:

- *Contour*: polyline-type, polygon with free space inside-type
- *Vertex*: position, positional uncertainty, existential uncertainty, obstacle-edge, field of view-edge

The different edge-types of the vertex-attributes are of special interest. They are used to distinguish between different kinds of connections between vertices. The connection might be either of type obstacle or of type field of view, i.e. an edge between free and occupied cells or an edge between free and unknown cells.

Changing the algorithm requires three parts to be examined. First of all, the process described in [3] for building the occupancy grid from ultrasonic sensor readings must be adapted to the *Fences*-architecture. The sonar sensor-model must be converted and the map-building component needs to be rebuilt to let it understand the *Fences*-interface. However, both steps are not further detailed here. The sensor-model that is used to create *Fences* from single measurements is shown in figure 3. Each contour is rasterized using computer vision algorithms and fused with the approach of [8]. Further details are omitted due to space limitations and it is assumed, that an accurate *Fences*-representation of the environment of the vehicle exists.

The second part that needs to be changed is the extraction of candidate points from the supplied model of the surroundings. The *Fences*-representation turns this extraction into an easy task. While the algorithm had to examine the neighborhood of every cell of the map to find all candidate points with the conventional architecture, the *Fences*-based algorithm merely has to check every vertex of each contour. The provided classification of each vertex is sufficient to obtain the required candidate points. These points induce satisfactory models for the second stage of the algorithm.

Lastly, the part of the algorithm that locates the parking lots on the closest circle must be adapted to *Fences*. The original algorithm needs to examine the cells of the map to calculate the gradient. The proposed interface simplifies this process, by again examining the classes of the vertices. A transition as described above emerges if two connected vertices have different classes attached to them. The additional constraints of a parking lot can then be checked easily.

An example of a successfully detected parking lot with the adapted algorithm is depicted in figure 5. The input to the algorithm is presented in the left part of the image. It shows an instance of the *Fences*-interface with attributes as described above. The orange circles represent the vertices, the red lines obstacles and the bluish lines limitations of free space through a limited field of view of the participating sensors. The fused map that is built from the sensor readings is shown lightly shaded to attain a better understanding of the abstract topological representation. Each vertex is attributed with a two-dimensional Cartesian coordinate that is used to position the vertex in the world, a dedicated class and

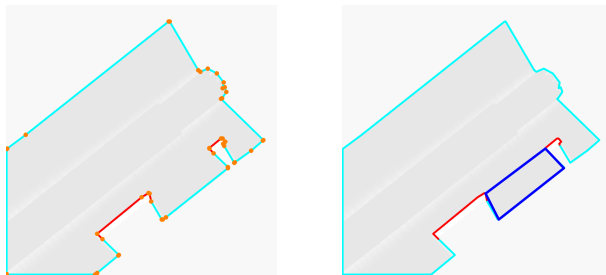


Figure 5: Application of the proposed architecture to parking lot detection: The input to the algorithm is shown on the left side, the detected parking lot in the right side.

its corresponding existential and positional uncertainties. The right side of figure 5 shows a detected parking lot. The results are comparable to those presented in the original paper. The parking lot and its bounds are reliably detected.

6 Conclusions

This paper proposed *Fences*, a new architecture for building maps of the static environment from sensor readings. It rearranges the location of several parts of the conventional architecture to achieve unification with homogeneous and generic interfaces. The main properties and benefits of this new architecture are:

- Exact sensor-models and function-specific information by moving those parts to their manufacturers and developers.
- Easy integration of additional sensors and driving assistance functionality due to a homogeneous interface (plug and play).
- Full abstraction of components which enable unnoticed replacement and removal of certain parts and even full components.
- Seamless adaption of the *Fences*-representation to different requirements, particularly bandwidth-related issues in distributed settings.
- Topological representation of free space and obstacles with an attached set of attributes that further describes the environment.

Since the intention of this article is to present the motivation and concepts behind the proposed *Fences*-architecture only a subset of the described properties and benefits could be evaluated. An example has been shown, how existing algorithms may be adapted to the new architecture without spending too much effort adapting it. However, to gain full confidence in the applicability of the presented architecture and its features several aspects need to be evaluated in more detail. Especially the changes that need to be done in each of the tree components need special attention.

Future work on the architecture, its homogeneous interface and the topological representation will concentrate on these aspects. A detailed examination of the topological representation is necessary in order to prove its ability to provide a full abstraction of

sensors and functions and prove its ability to efficiently describe sensor readings and structures of the static environment with full accuracy and every detail. Furthermore, its applicability to different driver assistance functions, the seamless replacement of the mapping strategy and the adaption to specific accuracy- and size-requirements needs to be evaluated.

References

- [1] Michael Darms, Matthias Komar, and Stefan Lueke. Map based road boundary estimation. In *Proceedings of 2010 IEEE Intelligent Vehicles Symposium*, pages 609 – 614, 2010.
- [2] Michael Reichel, Mohamed Essayed Bouzouraa, Andreas Siegel, Karl-Heinz Siedersberger, and Markus Maurer. Erweiterte Umfelderkennung und Nutzung einer Ausweichanalyse als Grundlage einer aktiven Gefahrenbremsung. In *Proceedings of 2010 Symposium on Automatisierungssysteme, Assisstenzsysteme und eingebettete Systeme für Transportmittel*, 2010.
- [3] Tim Kubertschak and Mirko Mählich. Extraktion von Parklücken auf probabilistischen Ultraschallkarten. In Fernando Puente León and Michael Heizmann, editors, *Forum Bildverarbeitung 2012*, pages 303 – 314. KIT Scientific Publishing, 2012.
- [4] Raja Chatila and Jean-Paul Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings of 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 138 – 145, 1985.
- [5] Christophe Coué, Thierry Fraichard, Pierre Bessière, and Emmanuel Mazer. Using bayesian programming for multi-sensor data fusion in automotive applications. In *Proceedings of 2003 IEEE International Conference on Robotics and Automation*, volume 2, pages 2104 – 2109, 2003.
- [6] Klaus Dietmayer, Alexander Kirchner, and Nico Kämpchen. Fusionsarchitekturen zur Umfeldwahrnehmung für zukünftige Fahrerassistenzsysteme. In Markus Maurer and Christoph Stiller, editors, *Fahrerassistenzsysteme mit maschineller Wahrnehmung*, pages 59 – 88. Springer-Verlag, 2005.
- [7] Ralph Grewe, Andree Hohm, Stefan Hegemann, Stefan Lueke, and Hermann Winner. Towards a generic and efficient environment model for adas. In *Proceedings of 2012 IEEE Intelligent Vehicles Symposium*, pages 316 – 321, 2012.
- [8] Hans P. Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Proceedings of 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 116 – 121, 1985.
- [9] Hernán Badino, Uwe Franke, and David Pfeiffer. The stixel world - a compact medium level representation of the 3d-world. In Joachim Denzler, Gunther Notni, and Herbert Süße, editors, *Proceedings of 31st DAGM Symposium on Pattern Recognition*, number 5748 in Lecture Notes in Computer Science, pages 51 – 60. Springer-Verlag, 2009.

Kontrollierbarkeit unerwarteter Eingriffe eines Bremsassistentensystems aus Perspektive des Folgeverkehrs

Alexandra Neukum^{*}, Frederik Naujoks[†], Stefan Kappes[‡]
und Torsten Wey[§]

Zusammenfassung: Ziel der Studie war die Abschätzung der Kontrollierbarkeit unerwarteter Eingriffe eines Notbremsassistentensystems aus Perspektive des Folgeverkehrs. Im Fokus der Studie stand das Verhalten von Fahrern, die mit einer plötzlichen und nicht vorhersagbaren Bremsung eines vorausfahrenden Fahrzeugs konfrontiert wurden. Die der Bewertung zugrunde gelegten Folgeabstände sollten hierbei 90% der im Realverkehr beobachtbaren geschwindigkeitsabhängigen Sekundenabstände abdecken. Betrachtet wurden drei Verzögerungsprofile mit Maximalverzögerungen von 0,3, 0,6 und 0,9g (Geschwindigkeitsabbau von ca. 20, 30 bzw. 40 km/h innerhalb von ca. 2,2s). In einer Studienreihe konnte mit einer Kombination aus Realfahrzeug- und nachgelagerten Simulatoruntersuchungen die Kontrollierbarkeit der Bremsingriffe innerhalb der untersuchten Bereiche des Folgeabstands sowie für verschiedene Geschwindigkeitsbereiche in einfachen und komplexen Verkehrsszenarien nachgewiesen werden.

Schlüsselwörter: Bremsassistentz, Controllability, unerwartete Bremsingriffe.

1 Einleitung

Zielsetzung der Arbeiten war es, eine Abschätzung der Kontrollierbarkeit für den Fall einer unerwarteten Aktivierung eines Notbremsassistentensystems auf der Basis empirischer Daten vorzunehmen. Im Fokus stand dabei das Verhalten unmittelbar nachfolgender Verkehrsteilnehmer, die mit einer plötzlichen und nicht vorhersagbaren Bremsung eines Vorausfahrenden konfrontiert wurden (vgl. hierzu auch [1]).

Die Untersuchung setzte sich auch mehreren Teilstudien zusammen: Im Hauptteil der Studie wurden die Fahrerreaktionen bei Bremsingriffen unter realen Verkehrsbedingungen bei einfachen Folgefahrten in verschiedenen Geschwindigkeitsbereichen experimentell untersucht. Aufbauend auf der Realfahrzeugstudie wurde mit dem Fokus

^{*} Dipl.-Psych. Alexandra Neukum, Würzburger Institut für Verkehrswissenschaften, neukum@wivw.de

[†] Dipl.-Psych. Frederik Naujoks, Würzburger Institut für Verkehrswissenschaften, naujoks@wivw.de

[‡] Dipl.-Ing. Stefan Kappes, Ford Werke GmbH, skappes@ford.de

[§] Dr.-Ing. Torsten Wey, Ford Werke GmbH, twey@ford.de

auf methodischen Aspekten der Kontrollierbarkeitsprüfung eine experimentelle Untersuchung im dynamischen Simulator des WIVW durchgeführt (Studie II). Zur Validierung der im Simulator erfassten Daten wurden die bereits im Realverkehr realisierten Geschwindigkeitsstufen und Bremsprofile betrachtet. Die Prüfungsumgebung Fahrsimulation wurde außerdem genutzt, um die Übertragbarkeit der in den beiden ersten Studienteilen gewonnenen Ergebnisse auf komplexere Verkehrssituationen abzuschätzen (Studie III). Die Fahrerreaktionen wurden hierbei in unterschiedlichen urbanen Szenarien untersucht, in denen die Aufmerksamkeit der Fahrer nicht allein auf das Vorderfahrzeug gerichtet war.

2 Prüfsituation und Bewertungskriterien

2.1 Prüfsituation

Die Prüfsituation bestand in Folgefahrten, in denen das vorausfahrende Fahrzeug plötzlich und ohne Vorankündigung mit einem der definierten Bremsprofile verzögerte. Vereinfacht, d.h. unter Annahme konstanter Verzögerungen, kann die Beherrschbarkeit fehlerhafter Bremsengriffe hierbei angenommen werden, wenn der Folgeabstand zum Vorausfahrenden beim Zeitpunkt des Bremsbeginns größer ist als die Fahrerreaktionszeit und der Fahrer im Folgefahrzeug mindestens so stark bremsst wie der Vorausfahrende ([2], siehe Formel 1):

$$(v_{\text{leading}} * THW) + \frac{v_{\text{leading}}^2}{ax_{\text{leading}}} > (v_{\text{follow}} * Brt) + \frac{v_{\text{follow}}^2}{ax_{\text{follow}}}$$

Formel 1: Bedingungen kontrollierbarer Bremsengriffe (THW = Zeitabstand, Brt = Reaktionszeit).

Aus dieser Betrachtung wird deutlich, dass dem Folgeabstand bei Bremsbeginn eine entscheidende Rolle für die Beherrschbarkeit zukommt. Die der Kontrollierbarkeitsbewertung zugrunde gelegten Folgeabstände bei Bremsengriff des Vorderfahrzeugs orientierten sich deshalb an umfassenden Felddaten: In der Untersuchung abgedeckt werden sollten 90% der im Realverkehr beobachtbaren Sekundenabstände. Die zur Beurteilung herangezogenen Perzentile sind hierbei geschwindigkeitsabhängig: 1.3s bei 30 km/h, 1.0s bei 50 km/h und 0.7s bei 100 km/h. Betrachtet wurden drei Verzögerungsprofile mit nach einer Dauer von ca. 0.5s auftretenden Maximalverzögerungen von 0.3, 0.6 und 0.9g. Durch die ca. 2.2s andauernden Bremsengriffe wurde jeweils ein Geschwindigkeitsabbau von 20, 30 bzw. 40 km/h erzielt.

2.2 Messgrößen und Bewertungskriterien

Zur Beschreibung des Fahrerverhaltens und der resultierenden Fahrzeugreaktionen wurden verschiedene objektive und subjektive Kenngrößen ermittelt (z.B. Maximalverzögerung, Bremsreaktionszeit etc.). Für die Kontrollierbarkeitsuntersuchung ist hierbei besonders die Kritikalität der Bremssituationen von Bedeutung:

- **Extended Time-to-Collision (ETTC)**: Die Eingangsdaten hierfür stammen aus dem Radarsensor des Folgefahrzeugs. Die ETTC gibt an, wie viel Zeit bis zur Kollision zwischen zwei Fahrzeugen bleibt, gegeben dem aktuellen Abstand (in Metern), der momentanen Relativgeschwindigkeit und der momentanen Relativverzögerungen [3]. Laut einschlägigen Forschungsarbeiten kann das Unterschreiten einer TTC von $\leq 1s$ als kritisch bewertet werden [4;5].
- Neben objektiven Kriterien wurden **Fahrerurteile** anhand der Skala zur Bewertung der Kritikalität von Fahr- und Verkehrssituationen erhoben [6]. Der wesentliche Vorteil dieser Kategorienskala besteht in der klaren Definition einer subjektiven Toleranzgrenze. Diese ist durch die Benennung von Kategorien und die Instruktion eindeutig definiert, so dass klar hervorgeht, in welche Kategorienbereiche Situationen eingeordnet werden sollen, die vom Fahrer als sicherheitsrelevant (Urteil ‚gefährlich‘ mit >6 Skalenpunkten) erlebt werden.

Zur **Bewertung der Beherrschbarkeit** wurde ein mehrdimensionaler Kriterienatz herangezogen: Zentrales Kriterium ist die Notwendigkeit eines Bremsingriffs des Versuchsleiters mittels der Beifahrer-Pedalerie als Indikator einer drohenden Kollision. Des Weiteren wird geprüft, ob Fahrerurteile im „gefährlich“-Bereich der Skala abgegeben werden und/oder ETTC-Werte unterhalb von 1s vorliegen. Die Kombination dieser beiden Parameter ist notwendig, da nicht auszuschließen ist, dass in Einzelfällen sehr kritische subjektive Bewertungen vorliegen können, obwohl die Situation objektiv gut beherrschbar war (und umgekehrt). In diesen Fällen wurden zur Kritikalitätsbewertung detaillierte Einzelfallanalysen vorgenommen.

3 Studie I: Folgefahrten im realen Straßenverkehr

3.1 Methodik

Als Versuchsträger wurden zwei Ford Focus (Modelljahr 2012) eingesetzt. Um die Sicherheit der Probanden während des Versuchs zu gewährleisten, wurde im Folgefahrzeug zusätzlich eine Beifahrer-Pedalerie eingebaut, über die der Versuchsleiter ggf. in die Fahrzeugführung eingreifen konnte. Die Stichprobe setzte sich aus 21 männlichen und 24 weiblichen Fahrern zusammen. Keiner der Fahrer hatte bislang an einer Untersuchung mit ähnlichen Fragestellungen teilgenommen. Das mittlere Alter der Fahrer betrug 40.3 Jahre ($sd=11.7$, $min=24$, $max=65$). Realisiert wurde ein vollständig abhängiger Versuchsplan mit den Faktoren „Geschwindigkeit“ (30, 50 und 100 km/h), „Abstand bei Bremsbeginn“ (dicht vs. normal) und „Bremsprofil“ (0.3, 0.6 und 0.9g). Um Reihenfolgeeffekte zu kontrollieren, wurden die Geschwindigkeitsbedingungen permutiert. Das Erstereignis bestand immer in der Auslösung eines 0.3g- oder 0.6g-Bremsprofils mit der Instruktion, einen „dichten“ Abstand zu fahren. Die Probanden wurden per Zufall auf die Bedingungen von Geschwindigkeit und Bremsprofil verteilt. Die nachfolgenden Bremsereignisse wurden in sechs verschiedenen Zufallsreihenfolgen absolviert. Das Erstereignis wurde nach dem ersten Geschwindigkeitsblock wiederholt, um Übungeffekte auf Bremsreaktion und Situationskritikalität abschätzen zu können.

Bei Ankunft an der Versuchsstrecke wurden die Probanden lediglich darüber informiert, dass es sich um eine Untersuchung zur Bewertung von Folgeabständen handele. Anschließend sollten sich die Fahrer mit dem Versuchsfahrzeug und dessen Beschleunigungs- und Bremsverhalten vertraut machen. Im anschließenden Hauptteil der Untersuchung wurden die Fahrer instruiert, dem Vorausfahrenden bei Einhaltung der jeweils geforderten Geschwindigkeit mit einem „normalen“ oder „dichten“ Abstand zu folgen. Der Versuchsleiter im Vorderfahrzeug löste jeweils auf ein Kommando des Versuchsleiters im Probandenfahrzeug die Falschbremsungen aus. Die Auslösung erfolgte – mit variablen Zeitspannen – nachdem die Fahrer den geforderten Folgeabstand („normal“ oder „dicht“) eingeregelt hatten. Nach jedem Bremsmanöver wurden die Fahrer zur erlebten Situationskritikalität befragt.

3.2 Ergebnisse

Alle in der Realfahrzeugstudie vorgenommenen Falschbremseingriffe erwiesen sich als beherrschbar. Innerhalb der definierten Folgeabstandsbereiche – und darüber hinaus auch bei z.T. aufgetretenen geringeren Folgeabständen – kam es nicht zu Unfällen oder Eingriffen des Versuchsleiters über die Beifahrer-Pedalerie. Alle Fahrer waren in der Lage, die Fehlersituation selbständig zu bewältigen.

Erstkonfrontation. Von besonderer Bedeutung ist die erste Konfrontation der unvorbereiteten Probanden: Diese folgten dem Vorderfahrzeug mit der Instruktion, einen dichten Abstand zu fahren und mussten auf Verzögerungen von 0.3 bzw. 0.6g reagieren. Wie aus Abbildung 2 links hervorgeht, lagen die beobachteten Folgeabstände im Mittel bei einer Sekunde und unterschieden sich nicht zwischen den Geschwindigkeitsbedingungen. Verzögerungen von 0.3g wurden subjektiv – und zwar geschwindigkeitsunabhängig – im Mittel als „harmlos“, Eingriffe von 0.6g als „unangenehm“, aber in der Regel nicht als „gefährlich“ bewertet (Abbildung 1 rechts). Vereinzelt abgegebene Beurteilungen im „gefährlich“-Bereich der Skala lassen sich durch objektive Kennwerte nicht untermauern ($ETTC_{min} > 1s$, Abbildung 2 links).

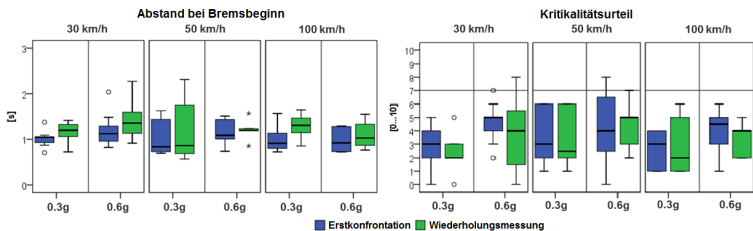


Abbildung 1: Zeitabstand bei Bremsbeginn (links) und Kritikalitätsurteil (rechts) bei Erstkonfrontation und Wiederholungsmessung.

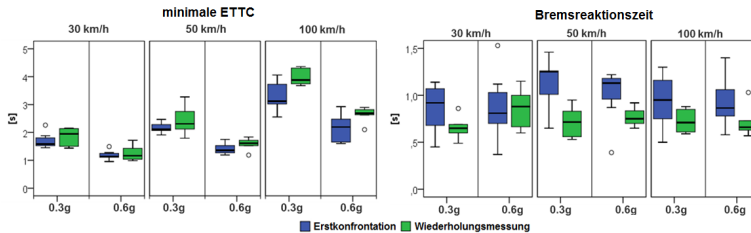


Abbildung 2: Minimale ETTC (links) und Bremsreaktionszeit (rechts) bei Erstkonfrontation und Wiederholungsmessung.

Nach der ersten Konfrontation mit einer Falschbremsung traten erwartungsgemäß Veränderungen im Fahrerverhalten auf, die sich über eine **direkte Messwiederholung** abschätzen lassen: Beobachtet wurde eine Erhöhung der Folgeabstände ($\Delta THW=0.15s$, Abbildung 1 links) sowie eine Verkürzung der Bremsreaktionszeiten ($\Delta RT=0.20s$, Abbildung 2 rechts) unabhängig von der Ausgangsgeschwindigkeit und der Bremsstärke des Vorderfahrzeugs. Infolge dieser Verhaltensanpassungen resultierten bei wiederholter Konfrontation höhere $ETTC_{min}$ -Werte (Abbildung 2 links). Unterschiede in den subjektiven Bewertungen lagen nicht vor.

Im **weiteren Untersuchungsverlauf** wurden alle drei Bremsprofile permutiert dargeboten. Insgesamt zeigten sich in den betrachteten subjektiven und objektiven Kenngrößen ($ETTC_{min}$, Maximalverzögerung der Fahrer, Bremsreaktionszeiten) neben dem Einfluss der Stärke des Falscheingriffs, Abhängigkeiten von der Fahrgeschwindigkeit sowie der Höhe der bei Beginn des Falschbremseneingriffs vorliegenden Folgeabstände. Abbildung 3 und Abbildung 4 stellen die minimale ETTC und die Maximalverzögerungen in den verschiedenen Geschwindigkeitsbereichen dar.

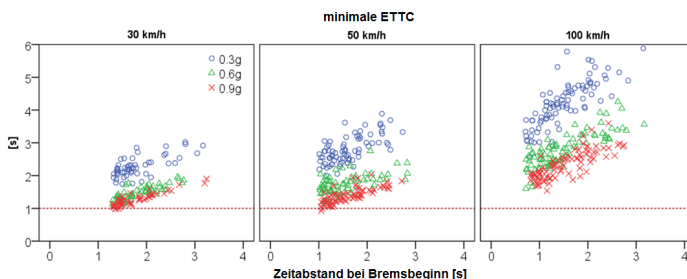


Abbildung 3: Minimale ETTC im definierten Folgeabstandsreich. Rote Linie: Potenziell kritischer $ETTC_{min}$ -Bereich von 1s.

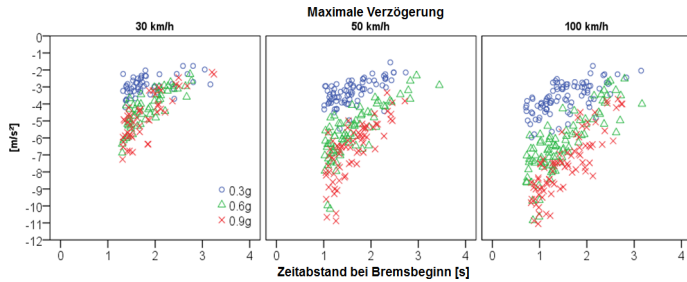


Abbildung 4: .Maximalverzögerungen im definierten Folgeabstandsbereich.

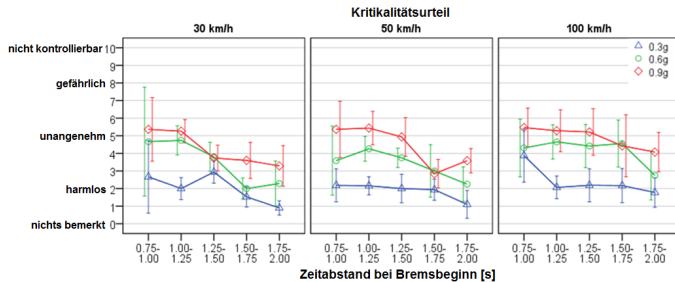


Abbildung 5: .Mittelwerte und Konfidenzintervalle der Kritikalitätsurteile.

Bei $v=100$ km/h resultierten durchgängig minimale ETTC-Werte im Bereich deutlich oberhalb einer Sekunde. Diese als nicht sicherheitsrelevant einzustufenden Werte werden z.T. durch sehr starke Verzögerungen der Fahrer erreicht. Im Bereich niedrigerer Geschwindigkeiten ($v=30$ km/h und $v=50$ km/h) wurden geringere $ETTC_{min}$ -Werte beobachtet, die aber ebenfalls den Bereich um 1s nicht unterschreiten. Die von den Fahrern aufgebrachten Maximalverzögerungen fallen hier – insbesondere bei $v=30$ km/h – deutlich geringer aus. Für die in Einzelfällen abgegebenen Urteile im „gefährlich“-Bereich der Skala (siehe Abbildung 5) lässt sich objektiv keine Gefährdung nachweisen (d.h. deutliche Unterschreitungen der $ETTC_{min}$ von 1s lagen nicht vor).

4 Studie II: Folgefahrten im Fahr Simulator

4.1 Methodik

Aufbauend auf der Realfahrzeugstudie wurde eine Untersuchung im dynamischen Fahr Simulator des WIVW durchgeführt, um abzuschätzen, inwieweit in den beiden

Prüfungsumgebungen vergleichbare Ergebnisse gewonnen werden.** Die Probanden wurden mit denselben Prüfsituationen konfrontiert, die auch im Realfahrzeug absolviert wurden. Vergleichbar zur Realfahrzeugstudie wurde ein vollständig abhängiges Versuchsdesign gewählt. Am Simulatorversuch nahmen N=21 Fahrer teil. Alle Fahrer hatten bereits an der Realfahrzeugstudie teilgenommen. Die Stichprobe setzte sich aus 8 männlichen und 13 weiblichen Fahrern zusammen. Das mittlere Alter der Fahrer betrug 41.6 Jahre (sd=12.3, min=25, max=61).

Wie in der Realstudie wurden die Fahrer darüber informiert, dass es sich um eine Untersuchung zur Bewertung verschiedener Folgeabstände handele. Die Fahrer wurden instruiert, dem jeweils Vorausfahrenden zu folgen und dabei einen eingeblendeten farbigen Balken auf der Fahrbahn zu beachten, der den einzuhaltenden Folgeabstand visualisierte. Die verschiedenen Streckenteile wurden nach einer Eingewöhnungsfahrt in permuierter Reihenfolge durchfahren. Das Vorderfahrzeug bremste mit den im Realfahrzeug gemessenen Bremsprofilen in unregelmäßigen Abständen während der Folgefahrt. Folgeabstand und Geschwindigkeit wurde hierbei zwischen den Bremssituationen variiert.

4.2 Ergebnisse

Im Fahr Simulator finden sich – wie aus Abbildung 6 (rechts) hervorgeht – fast durchgängig längere **Bremsreaktionszeiten**. Die Unterschiede sind bei höheren Geschwindigkeiten (100km/h) und starken Verzögerungen des Vorderfahrzeugs am deutlichsten ausgeprägt.

Abbildung 6 (links) stellt die minimalen **ETTC-Werte** aus Simulator- und Realfahrten getrennt nach Geschwindigkeitsbedingung und Bremsprofilstärke gegenüber. Es zeigt sich prinzipiell eine sehr gute Übereinstimmung. Auch in der Fahrsimulation findet sich die Abhängigkeit der $ETTC_{min}$ -Werte von Folgeabstand und Geschwindigkeit sowie der Stärke des Bremsengriffs. Aus Abbildung 6 (links) wird aber auch ersichtlich, dass durchaus systematische Abweichungen zwischen den beiden Prüfungsumgebungen vorliegen: Im Simulator fallen die $ETTC_{min}$ -Werte insbesondere bei niedrigen Geschwindigkeiten, geringen Folgeabständen und starken Bremsengriffen kritischer aus als im Realfahrzeug. So liegen die Differenzen beim Bremsprofil von 0.9g und geringen Sekundenabständen – betrachtet werden Sekundenabstände im Intervall 0.75-1.5s – im Mittel bei 340ms.

Mitbedingt durch die späteren Reaktionen werden im Simulator in der Mehrzahl der Bedingungen stärkere **Verzögerungen** aufgebracht (siehe Abbildung 7 links). Ausnahmen sind die Fahrten mit stärkeren Bremsprofilen bei hoher Fahrgeschwindigkeit (100km/h).

Insgesamt zeigt sich hinsichtlich der Geschwindigkeit ein systematischer Unterschied zwischen Simulator- und Realfahrt: Während die Fahrer im Realfahrzeug mit steigender Geschwindigkeit stärker bremsen, findet sich in der Simulatorstudie ein entgegengesetzter Zusammenhang zwischen Geschwindigkeit und Maximalverzögerung.

** Generell ist bei Untersuchungen in der Fahrsimulation nicht von absoluter Validität der Ergebnisse auszugehen, vielmehr muss ermittelt werden, ob und in welcher Weise systematische Abweichungen gegenüber der realen Prüfungsumgebung vorliegen.

Konsistent zu den Befunden aus den objektiven Daten fallen die **Fahrerurteile** in der Fahrsimulation kritischer aus als im Realfahrzeug (siehe Abbildung 7 rechts). Dies betrifft vor allem Fahrten mit geringen Folgeabständen zum Zeitpunkt des Bremsingriffs.

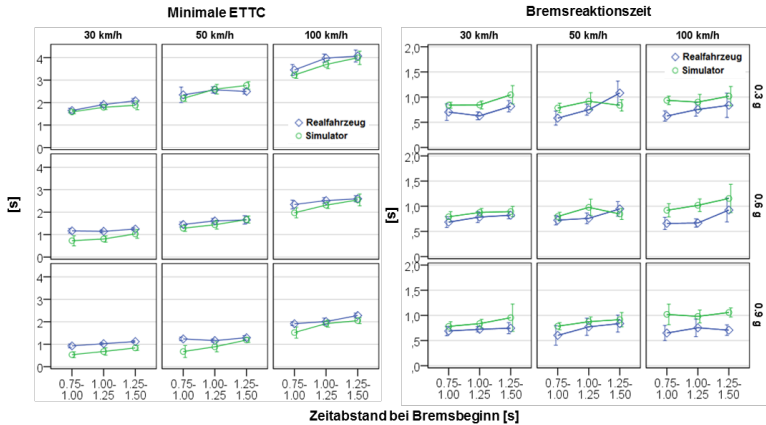


Abbildung 6: Minimale ETTC (links) und Bremsreaktionszeit (rechts).

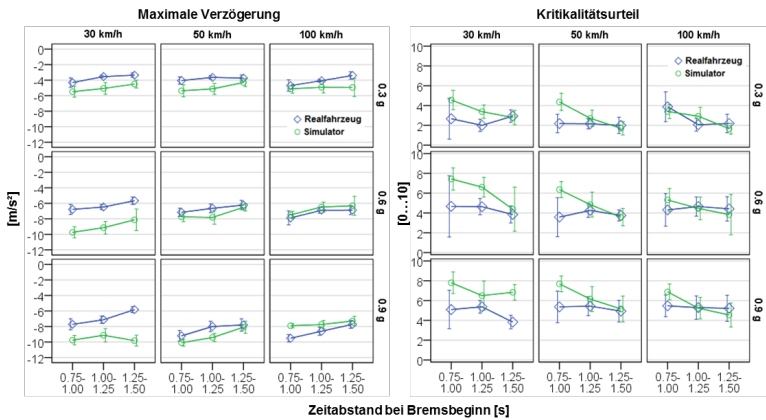


Abbildung 7: Maximale Verzögerung (links) und Kritikalitätsurteil (rechts).

5 Studie III: Folgefahrten in komplexeren Verkehrssituationen im Fahrsimulator

Aufgrund der in Studie I und II gewählten einfachen Folgefahrtsituationen, in denen die Aufmerksamkeit der Fahrer ausschließlich auf das Vorderfahrzeug ausgerichtet war, ist generell eine Unterschätzung der Bremsreaktionszeiten und damit der Kritikalität der Bremsereignisse zu erwarten. Im dritten Studienteil erfolgte deshalb eine Prüfung der Fahrerreaktionen in komplexeren urbanen Szenarien, die durch höhere visuelle Anforderungen gekennzeichnet waren. Ausgewählt wurden hierfür urbane Szenarien im Geschwindigkeitsbereich von 50km/h und das 0.6g-Bremsprofil. Erstellt wurden zwölf Varianten des ursprünglichen Folgefahrtszenarios. Bei der Gestaltung der Prüfsituationen wurde auf das SEEV-Modell [7] der Aufmerksamkeit zurückgegriffen. Die Szenarienvarianten beinhalteten plötzliche Bremsereignisse des Vorderfahrzeugs in Situationen, in denen a) die Aufmerksamkeitssteilung zwischen verschiedenen Aufmerksamkeitsquellen geteilt werden muss (sog. „Effort“, z.B. Links- und Rechtsabbiegen); b) die Bremsleuchten des Vorderfahrzeugs nicht die einzigen aufmerksamkeitsbindenden Umgebungsreize darstellen (sog. „Saliency“, z.B. Einsatzfahrzeug mit aktiviertem Blaulicht am Straßenrand); und c) Bremsereignisse sehr unwahrscheinlich sind (sog. „Expectancy“, z.B. das Überqueren einer ampelgeregelten Kreuzung).

5.1 Versuchsplan und Stichprobe

Alle Prüfsituationen wurden in einem vollständig abhängigen Versuchsplan absolviert. Um Reihenfolgeeffekte zu kontrollieren, wurden sechs verschiedene Versuchsabfolgen erstellt. Angestrebt war ein Folgeabstand bei Bremsbeginn von ca. einer Sekunde mittels der Abstandsanzeige aus Studie 2. Es nahmen N=20 Fahrer teil, von denen N=7 bereits an der Realfahrzeugstudie teilgenommen hatten. Die Stichprobe setzte sich aus 11 männlichen und 9 weiblichen Fahrern zusammen. Das mittlere Alter der Fahrer betrug 44.1 Jahre (sd=14.2, min=26, max=64).

5.2 Ergebnisse

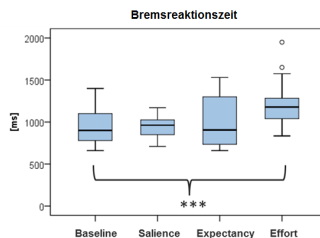


Abbildung 8: Bremsreaktionszeit, *** = $p < .001$.

Gegenüber der einfachen Folgefahrt zeigte sich eine signifikante Erhöhung der Bremsreaktionszeiten nur in den Szenarien der Kategorie „Effort“ (siehe Abbildung 8). Die Reaktionszeitdifferenz zur Baseline betrug im Mittel ca. 260ms ($KI_{95\%}$ [128; 385]). Nimmt man die obere Grenze des 95%-Vertrauensintervalls an, ist in komplexeren Verkehrsszenarien von einer **Verlängerung der Reaktionszeit um 400ms** auszugehen.

6 Zusammenfassende Bewertung

Zur Abschätzung der Kontrollierbarkeit unerwarteter Eingriffe eines Bremsassistenzsystems wurde eine Kombination unterschiedlicher Prüfumgebungen eingesetzt. Aufgrund der sehr zufriedenstellenden Übereinstimmung zwischen den Ergebnissen der Testumgebungen erweist sich die Fahrsimulation insgesamt als geeignetes Medium zur Untersuchung der Auswirkungen von Falschbremseingriffen. Zu beachten ist dabei, dass Einschätzungen der Beherrschbarkeit auf der Basis der Untersuchung in der Fahrsimulation konservativer ausfallen als im Realfahrzeug. Systematische Abweichungen lagen insbesondere vor in den beobachteten Bremsreaktionszeiten, die im Simulator durchgängig höher ausfallen. Relevant für die Kontrollierbarkeitsbewertung sind weiterhin vor allem die geringer ausfallenden $ETTC_{\min}$ -Werte: Deutliche Differenzen wurden insbesondere bei niedrigen Geschwindigkeiten, geringen Folgeabständen und starken Bremsingriffen beobachtet. So liegen die Differenzen beim Bremsprofil von 0.9g und geringen Sekundenabständen (0.75-1.5s) im Mittel bei 340ms. Analog hierzu werden die Bremsingriffe in der Fahrsimulation kritischer bewertet.

Für die Einschätzung der Kontrollierbarkeit werden die Ergebnisse der durchgeführten Realfahrzeugstudie als zentrale Basis herangezogen. Bei der Interpretation der erhobenen Daten müssen jedoch zwei Effekte Berücksichtigung finden: Erwartungsgemäß finden sich Übungseffekte, d.h. Reaktionszeitdifferenzen von 0.2s, die auf das abhängige Versuchsdesign zurückzuführen sind. Weiterhin lassen sich im Realverkehr lediglich einfache Folgefahrten realisieren, die mit niedrigen Aufmerksamkeitsanforderungen einhergehen. Eine Abschätzung der Reaktionszeitdifferenzen gegenüber komplexeren Verkehrssituationen wurde aus den Untersuchungen in der Fahrsimulation gewonnen. Ermittelt wurde eine Verlängerung der Reaktionszeiten um weitere 0.4s.

Auf der Basis dieser Ergebnisse werden die $ETTC_{\min}$ -Werte der Realfahrzeuguntersuchung für die abschließende Kontrollierbarkeitsbewertung reevaluiert. Die $ETTC_{\min}$ -Werte werden hierbei um die jeweiligen Reaktionszeitdifferenzen adjustiert. Hierbei wird davon ausgegangen, dass (1) der Effekt von Übung und Komplexität der Verkehrssituation additiv wirkt und (2) spätere Bremsreaktionen nicht mit stärkeren Verzögerungen kompensiert werden, was eine konservative Annahme darstellt. Abbildung 9 stellt die adjustierten $ETTC_{\min}$ -Werte aus Studienteil 1, basierend auf einer Reduktion der Werte um 0.6s ($\Delta_{\text{Übung}} \approx 0.2s$, $\Delta_{\text{Komplexität}} \approx 0.4s$). Aus der Abbildung wird deutlich, dass keiner der Werte unter 0s fällt, nachdem diese für Übungs- und Aufmerksamkeits-effekte adjustiert wurden. Auf Basis dieses Ergebnisses kann demnach von einer Kontrollierbarkeit nicht-intendierter Bremsingriffe innerhalb der untersuchten Bereiche des Folgeabstands ausgegangen werden.

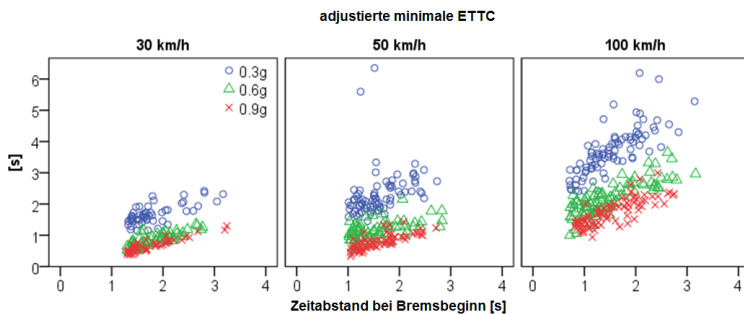


Abbildung 9: Adjustierte $ETTC_{\min}$ -Werte innerhalb der definierten Bereiche des Folgeabstands.

7 Literatur

- [1] Fach, M., Baumann, F., Breuer, J. & May, A. J. (2010). Bewertung der Beherrschbarkeit von Aktiven Sicherheits- und Fahrerassistenzsystemen an den Funktionsgrenzen. VDI-Berichte, 2104, 425-435.
- [2] Faber, F., Jonkers, E., Aust, M. L., Benmimoun, M., Regan, M., Jamson, S. & Dobberstein, J. (2011). Analysis methods for user related aspects and impact assessment on traffic safety, traffic efficiency and environment. euroFOT Deliverable D6.2 Aachen: euroFOT Consortium.
- [3] Winner, H. (2012). Frontalkollisionsschutzsysteme. In H. Winner, S. Hakuli & G. Wolf (Eds.), Handbuch Fahrerassistenzsysteme (pp. 522-542). Wiesbaden: Vieweg & Teubner.
- [4] Hayward, J. C. (1972). Near miss determination through use of a scale of danger. The Pennsylvania State University, Pennsylvania.
- [5] van der Horst, A. R. A. (1990). A time-based analysis of road user behaviour in normal and critical encounters. Dissertation, Technische Universiteit, Delft.
- [6] Neukum, A., Lübbecke, T., Krüger, H.-P., Mayser, C., & Steinle, J. (2008). ACC-Stop&Go: Fahrerverhalten an funktionalen Systemgrenzen. In M. Maurer & C. Stiller (Eds.), 5. Workshop Fahrerassistenzsysteme - FAS 2008 (pp. 141-150). Karlsruhe: fmrt.
- [7] Wickens & McCarley. (2008). *Applied Attention Theory*. Boca Raton: CRCPress.

Neuartige Warnstrategien zur Ausrichtung der Aufmerksamkeit bei Kollisionsgefahr

Lutz Lorenz^{*}, Thomas Lindberg^{**} und Klaus Bengler[†]

Zusammenfassung: Dieser Beitrag untersucht das Potential von fahrerabhängigen – also an den Fahrerzustand angepassten – Warnstrategien. Im statischen Fahrsimulator der BMW Forschung und Technik GmbH wurde hierzu eine Studie mit N = 109 Probanden durchgeführt. Drei neuartige Warnkonzepte wurden gegenüber einer Baseline (Serienstand) auf Wirksamkeit untersucht. Es wird eine deutlich höhere Wirksamkeit von fahrerabhängigen Warnstrategien bei Kollisionsgefahr nachgewiesen. Darüber hinaus wird erstmals ein neuartiges Warnkonzept in Form einer stroboskopischen Bewegungsanimation vorgestellt.

Schlüsselwörter: Auffahrwarnung, Fahrerabhängige Warnstrategien, Mensch-Maschine-Interaktion (MMI).

1 Einleitung

Die bisherige Forschung hat gezeigt, dass die Fahrerleistung durch das Bearbeiten einer Nebenaufgabe, z. B. der Bedienung von Fahrerinformationssystemen, beeinträchtigt werden kann [1]. Daher muss gewährleistet sein, dass der Fahrer auch bei einer Blickabwendung ins Central Information Display (CID) kritische Verkehrssituationen rechtzeitig erkennt und ein entsprechendes Fahrmanöver zur Unfallvermeidung einleiten kann. Der Fahrer wird hierbei durch Fahrerassistenzsysteme wie z. B. der Kollisionswarnung unterstützt. Bei Unterschreitung einer bestimmten Time to Collision (TTC) [2] wird unabhängig vom visuellen Aufmerksamkeitsfokus des Fahrers ein Warnsymbol in Kombi Instrument und Head Up Display (HUD) ausgegeben.

Für den Fall einer Blickabwendung ins CID wird jedoch angenommen, dass eine fahrerabhängige – also an den visuellen Aufmerksamkeitsfokus angepasste – Warnstrategie zu einer schnelleren Ausrichtung der Aufmerksamkeit und somit zu einer kürzeren Bremsreaktionszeit führt. Die in diesem Beitrag vorgestellten Warnkonzepte werden somit nur bei einem Blick in das CID ausgelöst. Sie sind also als eine zusätzliche Unterstützung für den Fahrer bei einer Blickabwendung vom Fahrgeschehen zu verstehen und nicht als alleinstehende Warnkonzepte.

^{*} Dipl. -Ing. Lutz Lorenz, BMW Forschung und Technik GmbH (Lutz.Lorenz@bmw.de).

^{**} Dr. -Ing. Thomas Lindberg, BMW AG (Thomas.Lindberg@bmw.de).

[†] Univ.-Prof. Dr. phil. Klaus Bengler, Technische Universität München (Bengler@tum.de).

2 Neuartige Warnstrategien zur Aufmerksamkeitslenkung

2.1 Grundlagen zu fahrerabhängigen Warnstrategien

Kollisionswarnsysteme aktueller Fahrzeuggenerationen nutzen häufig mehrstufige Warnstrategien. Je nach verbleibender TTC zum Vorderfahrzeug wird zum Beispiel zwischen einer Vorwarnung (TTC < 2,8 s) und einer Akutwarnung unterschieden (TTC < 1,6 s). Bei der Vorwarnung wird in Kombiinstrument und HUD ein rotes Warnsymbol (Fahrzeugsilhouette) ausgegeben, wobei die Akutwarnung durch ein zusätzliches Blinken der Anzeige und eine akustische Ausgabe verstärkt wird.

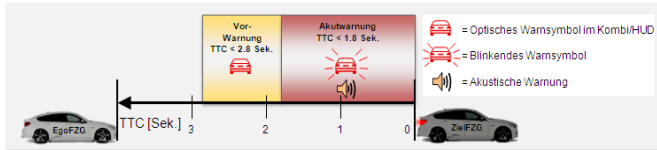


Abbildung 1: Eindimensionale Warnstrategie (ohne Fahrdynamische Eingriffe, z. B. Anbremsen)

Die bisherige Warnstrategie orientiert sich hauptsächlich an potentiellen Gefahren im Fahrzeugumfeld. Um den Fahrer jedoch schneller auf eine bevorstehende Gefahr hinzuweisen, sollte zusätzlich der Fahrerzustand – in diesem Fall die Blickrichtung – in die Warnstrategie mit einbezogen werden. Mit einer solchen fahrerabhängigen Warnstrategie wird dem Fahrer je nach Blickrichtung eine maßgeschneiderte Warnung präsentiert, wodurch die Wirksamkeit des Assistenzsystems erhöht werden kann. So ergibt sich aus der eindimensionalen Warnstrategie (siehe Abb. 1) durch hinzufügen der Dimension „Grad der Blickabwendung“ eine Warnmatrix (siehe Abb. 2).

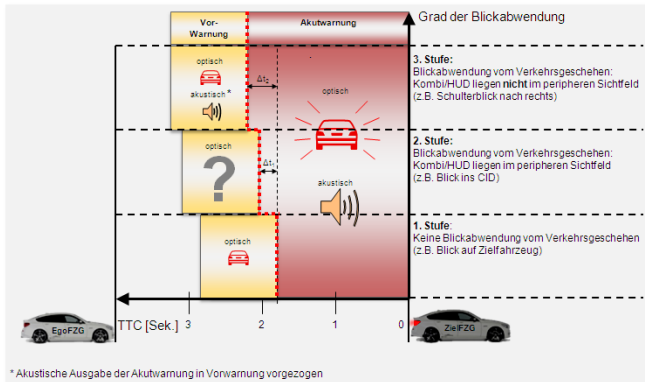


Abbildung 2: Zweidimensionale Warnstrategien (Warnmatrix)

Der Grad der Blickabwendung ist in der Warnmatrix in drei Stufen auf der Ordinate angegeben. In der ersten Stufe ist der Blick des Fahrers auf das Verkehrsgeschehen gerichtet, wodurch eine Anpassung der Warnstrategie nicht notwendig ist (vgl. Abb. 1). In Stufe 2 ist der Aufmerksamkeitsfokus vom Verkehrsgeschehen soweit abgewendet (z. B. Blick ins CID), dass Anzeigen im Kombi und HUD noch peripher wahrgenommen werden können. Befindet sich der Aufmerksamkeitsfokus komplett aus dem Wahrnehmungsbereich aller Anzeigeelemente (z. B. bei einem Schulterblick), wird Stufe 3 der Blickabwendung erreicht. Aus diesen drei Stufen und den zwei zeitlich eskalierten Warnstufen (Vor- und Akutwarnung) lassen sich nun die Gestaltungsmöglichkeiten für fahrerabhängige Warnstrategien ableiten.

Im Falle einer Akutwarnung ist eine Kollisionsgefahr so hoch, dass ein aufdringlicher Warnnton zur Ausrichtung der Aufmerksamkeit unverzichtbar ist. Somit bleibt die Akutwarnung über alle Stufen der Blickabwendung von der Ausgabemodalität her gleich (optisch & akustisch). Lediglich die Warnzeitpunkte können mit zunehmender Blickabwendung vom Verkehrsgeschehen nach vorne verschoben werden (dargestellt durch die Parameter Δt_1 und Δt_2) [3].

Die Vorwarnung hingegen besitzt aufgrund der geringeren Kritikalität einen eher informierenden als warnenden Charakter. Akustische Warngaben sind in dieser Phase der Unfallentstehung zu aufdringlich und somit zu vermeiden [4]. Als Ausnahme gilt die dritte Stufe der Blickabwendung. In diesem Fall sollte bereits bei der Vorwarnung akustisch gewarnt werden, da optische Ausgaben im Kombi und HUD nicht wahrgenommen werden können.

Zentraler Untersuchungsgegenstand in diesem Beitrag ist die Gestaltung der Vorwarnung in der zweiten Stufe der Blickabwendung (Fragezeichen in Abb. 2). Ein häufiger Anwendungsfall hierfür ist die Bearbeitung einer Nebenaufgabe z. B. die Eingabe eines Navigationsziels. Hierbei ist der Blick zeitweise auf das CID in der Mitte der I-Tafel gerichtet, wodurch der Fahrer in Gefahrensituationen das Warnsymbol in Kombiinstrument und HUD sehr spät oder gar nicht mehr wahrnimmt und erst durch die Akutwarnung auf die Gefahr aufmerksam wird.

Daher stellt sich die Frage, wie eine optische Vorwarnung angepasst werden muss, damit die Aufmerksamkeit schnell vom Anzeigeort der Nebenaufgabe (CID) auf die Gefahr gerichtet wird. Zwei Grundprinzipien zur Ausrichtung der visuellen Aufmerksamkeit werden in diesem Beitrag angewendet:

- Ein optischer Reiz aus Richtung der Gefahr führt zu einer Aufmerksamkeitsausrichtung (Periphere Blickführung, Kap. 2.1).
- Ein bewegter optischer Reiz, der im Aufmerksamkeitsfokus beginnt und sich in die Richtung der Gefahr bewegt, führt zu einer Aufmerksamkeitsausrichtung (Animierte Blickführung, Kap. 2.2).

2.2 Warnstrategie „Periphere Blickführung“

Größen- und Intensitätsänderungen optischer Reize im peripheren Sichtfeld sind stark aufmerksamkeitsbindend [5]. Demzufolge wurde als erstes Warnkonzept die Variante „HUD“ umgesetzt, bei der die gesamte HUD-Fläche (ca. 21 x 9 cm, Breite x Höhe) mit

einer Blinkfrequenz von 3,33 Hz (On-Off-Ratio: 1:1) aufleuchtet. Bei jedem Blinkzyklus entsteht ein rotes Rechteck im Ursprung und vergrößert sich auf die gesamte HUD-Fläche. Dieser optische Reiz im peripheren Sichtfeld soll den Blick schnell auf die Straße lenken und eine schnellere Bremsreaktion verursachen.

2.3 Warnstrategie „Animierte Blickführung“

Bei dem zweiten Warnkonzept verjüngt sich das angezeigte CID Bild zu einem roten Punkt, der sich innerhalb des Displays in Richtung Frontscheibe bewegt, und über ein LED-Lauflicht bis zur Scheibenwurzel fortgeführt wird (Animationsdauer ca. 300 ms). Das LED-Lauflicht besteht aus 5 LEDs, die in einem Abstand von 8 cm in die I-Tafel integriert sind und nacheinander für 40 ms aufleuchten, jedoch erst nachdem die CID Animation beendet ist. Der Fahrer erhält so den Eindruck eines bewegten optischen Reizes, der durchgehend vom CID bis hin zur Fahrbahn verläuft. Hierdurch soll der Blick an die Bewegung gebunden werden, und somit die Aufmerksamkeit schnell auf die bevorstehende Gefahr ausgerichtet werden (Warnvariante „LED“).

Diese Warnstrategie beruht auf dem Prinzip der Scheinbewegung. Das Auge nimmt ab einer bestimmten Geschwindigkeit (30 – 60 ms) aufeinanderfolgende Bilder nicht mehr einzeln sondern als Schein- bzw. Teilbewegung wahr. Die durch Einzelbilder erzeugte Positionsänderung wird als durchgehende Bewegung wahrgenommen (stroboskopische Bewegung) [6]. Dieses Phänomen wird hier erstmals zur Ausrichtung der Aufmerksamkeit in einem Fahrkontext untersucht.

Um den spezifischen Nutzen der LEDs zu zeigen, wurde zusätzlich eine dritte Warnstrategie getestet, die nur aus der Animation im CID besteht (Warnvariante „CID“).



Abbildung 3 : Untersuchte Warnstrategien

Als Baseline diente die Vorwarnung der aktuellen BMW 7er Serie (siehe Abb. 1). Alle hier untersuchten Warnausgaben wurden bei einer TTC von ca. 2,8 Sekunden ausgelöst. Der Vergleich zwischen Baseline und den drei neuartigen Warnkonzepten soll die Wirksamkeit und den Mehrwert von fahrerabhängigen Warnstrategien zeigen.

Es ist anzunehmen, dass die Varianten *LED* und *HUD* aufgrund ihrer höheren optischen Präsenz schnellere Reaktionszeiten als die Baseline hervorrufen. Bei Variante *CID* wird dagegen angenommen, dass sich die Reaktionszeiten verlängern. Aufgrund der Entfernung zwischen CID und Fahrbahn reicht der animierte Reiz im CID alleine nicht aus, um den Blick schnell genug bis zur Straße zu führen.

3 Methodik

3.1 Versuchsdesign

Im statischen Fahrsimulator fuhren $N = 109$ Probanden auf einer Autobahn mit drei Fahrstreifen, je nach Anweisung entweder 60 oder 80 km/h. Zu Beginn sollten sich die Fahrer in einer zehn Minuten langen Eingewöhnungsfahrt mit der Fahrzeugführung im Fahrsimulator und der Nebenaufgabe vertraut machen. Als Nebenaufgabe wurde eine Navigationseingabe gewählt (basierend auf dem iDrive Bedienkonzept der aktuellen BMW 7er Serie). Während der Fahrt sollten die Versuchspersonen die Namen deutscher Städte über das sogenannte Spellermenü (siehe Abb. 4) in die Navigation eingeben (insgesamt fünf Städte). In der eigentlichen Versuchsfahrt (ca. 15 Minuten) erlebte jeder Proband nur eine der vier Warnvarianten in einer Prüfsituation, die nach einem vordefinierten Ablauf konzipiert war (siehe Kapitel 3.2). Die Probanden wurden nicht über das Ziel des Versuches informiert.

3.2 Prüfsituation – Ablauf & Ausschlusskriterien

In der Prüfsituation wurde der Fahrer angewiesen konstant 60 km/h zu fahren. Ein vorausfahrendes Fahrzeug – das Zielfahrzeug – regelte daraufhin seine Geschwindigkeit automatisch so, dass die Zeitlücke zwischen Egofahrzeug (Proband) und Zielfahrzeug ca. 1,6 Sekunden (ca. 26,6 Meter) betrug. Der Versuchsleiter wies den Fahrer an, einen Städtenamen im Navigationsmenü einzugeben, und beobachtete online seine Blickrichtung mit Hilfe des Eyetrackers Dikablis der Firma Ergoneers. Sobald der Fahrer ins CID blickte, löste der Versuchsleiter per Knopfdruck eine Notbremsung des Zielfahrzeuges aus ($a_{\text{ZielfZg}} = -6,5 \text{ m/s}^2$). Da das Zielfahrzeug unmittelbar vor der Auslöseschwelle der Vorwarnung vorausfuhr, löste unmittelbar nach der Bremsung das Kollisionswarnsystem die entsprechende Warnvariante aus. Alle Warntöne wurden bei einer TTC von ca. 2,8 Sekunden ausgegeben. Da sich dieser Beitrag mit dem Vorwarnbereich – also rein optischen Warnstrategien (siehe Fragezeichen in Abb. 2) – beschäftigt, wurde auf akustische Warntöne verzichtet. Das Streckendesign in der Prüfsituation war so konzipiert, dass ein Ausweichen nicht möglich war (aufgrund der Fahrbahnbegrenzung links und eines Fahrzeuges rechts neben dem Egofahrzeug).

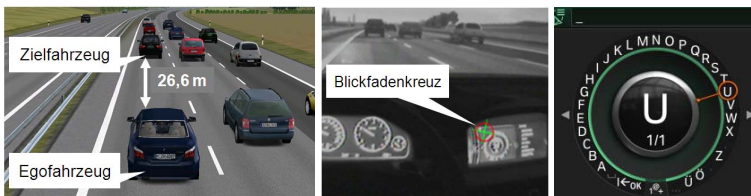


Abbildung 4: Streckendesign in der ersten Prüfsituation (links), Dikablis Onlinebild mit Blickfadenkreuz, das die aktuelle Blickrichtung anzeigt (mitte), Spellermenü zur Navigationseingabe (rechts). Die Probanden haben durch Drehen und Drücken des iDrive Controllers jeden Buchstaben einzeln im Speller eingegeben

Probanden, die in der Bremssituation einen größeren Abstand als 28 Meter zum Zielfahrzeug aufwiesen oder langsamer als 50 km/h fuhren, wurden nicht in die Auswertung mit einbezogen (insgesamt acht Probanden). Ein weiteres Ausschlusskriterium war, wenn Fahrer die Warnung nicht im visuell abgewendeten Zustand erlebt haben (insgesamt 19 Probanden). Das heißt der Fahrer hat bereits auf die Straße geblickt bevor die Warnung ausgelöst wurde. Drei Probanden mussten wegen Übelkeit den Versuch noch vor der Prüfsituation abbrechen.

3.3 Stichprobe

Aufgrund aller in Kapitel 3.2 genannten Ausschlusskriterien wurden $N = 79$ Probanden in die Auswertung aufgenommen, darunter waren 19 Frauen und 60 Männer im Alter von 19 bis 57 Jahren ($M = 32,9$ Jahre ; $SD = 9,6$ Jahre). Diese Stichprobe wurde in vier Versuchsgruppen unterteilt, wobei innerhalb einer Gruppe jeweils nur eine der vier Bedingungen getestet wurde (Between Subject Design). Mögliche Störvariablen wie Geschlecht, Alter und iDrive-Erfahrung wurden über alle Gruppen hinweg möglichst gleich gehalten. So bestand jede Versuchsgruppe zu ca. 40 % aus Fahrern einer jüngeren (19 - 24 Jahre), zu ca. 40 % aus Fahrern einer mittelalten (25 - 49 Jahre) und zu ca. 20 % aus Fahrern einer älteren (50 - 57 Jahre) Altersgruppe. Der Frauenanteil pro Gruppe lag bei ca. 20 % und knapp die Hälfte der Probanden einer Gruppe hatte iDrive Erfahrung.

3.4 Auswertung

Die Warnkonzepte wurden auf den drei Datenebenen Blick-, Fahrverhalten und subjektive Bewertung miteinander verglichen. Das Blickverhalten wurde mit Hilfe der Blickreaktionszeit und der Blickstrategie bewertet. Bei der Blickreaktionszeit handelt es sich um den Zeitraum vom Beginn der Warnausgabe bis der Blick das CID verlassen hat. Sie beschreibt somit, wie lange es dauert, den Fahrer durch die Warnausgabe visuell aus der Nebenaufgabe zu lösen. Darüber hinaus wurde durch Analyse der Blickstrategie untersucht, wie sich die Blickwege vom CID bis zum Zielfahrzeug unterscheiden. Das Fahrverhalten wurde durch die Variable Bremsreaktionszeit – also der Zeitdauer von der Warnausgabe bis zur Berührung des Bremspedals – beschrieben. Zur subjektiven Bewertung wurden die Fahrer unmittelbar nach der Warnausgabe gefragt, was sie wahrgenommen haben.

4 Ergebnisse

4.1 Blickverhalten

Sowohl die *HUD*- ($M = 0,41$ s ; $SD = 0,12$ s) als auch *LED*-Variante ($M = 0,47$ s ; $SD = 0,18$ s) besitzen mit ca. 0,4 Sekunden die kürzeste Blickreaktionszeit (siehe Abb. 5). Am längsten benötigten die Fahrer bei Variante *CID* ($M = 0,83$ s ; $SD = 0,52$ s), um ihren Blick aus dem Display zu lösen (siehe Abb. 5).

Mit Hilfe eines Kruskal-Wallis Test konnte ein signifikanter Unterschied zwischen den Mittelwerten der Blickreaktionszeit nachgewiesen werden ($H(3) = 9,996$, $p = .02^*$). Zum Vergleich der fahrerabhängigen Warnstrategien mit der Baseline ($M = 0,68$ s ; $SD = 0,39$ s) wurden Mann-Whitney U-Tests durchgeführt. So zeigte sich, dass die Blickreaktionszeiten von Variante *HUD* ($U = 139$, $p = .008^*$, 1-seitig) und Variante *LED* ($U = 162$, $p = .025^*$, 1-seitig) um ca. 200 ms kürzer als bei der Baseline waren (Vergleiche mit einer Baseline erfordern keine Alpha-Kumulierung, siehe Argumentation von Keppel [7]). Bei Variante *CID* verschlechterte sich die Blickreaktionszeit hingegen um ca. 150 ms, wobei dieser Unterschied nicht signifikant wurde.

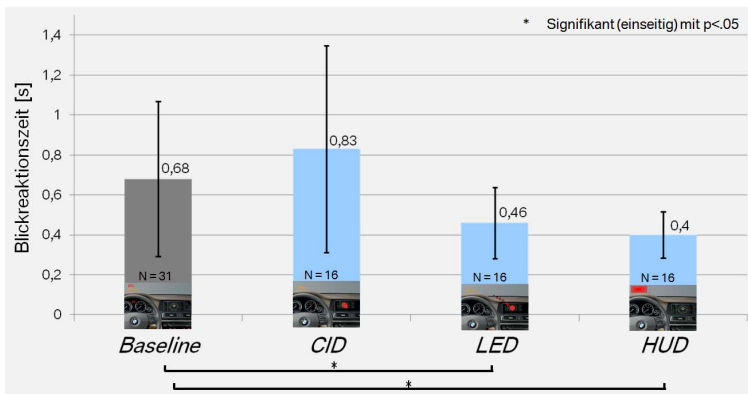


Abbildung 5 : Mittelwerte-Vergleich der Blickreaktionszeit

Die Analyse der Blickstrategie ergab, dass bei der Variante *LED* die Blicke aller Probanden dem Lauflicht bis zur Scheibenwurzel folgten, wonach 80 % der Fahrer anschließend direkt auf das Zielfahrzeug blickten. 20 % machten nach der Warmausgabe einen kurzen Kontrollblick zurück auf das CID. Bei der Variante *CID* waren es hingegen 38 %. Bei Variante *HUD* blickten 11 % der Fahrer in einer Sakkade vom CID direkt auf das Zielfahrzeug. Alle übrigen Probanden der *HUD*-Variante machten mindestens einen kurzen Kontrollblick in das aufflackernde HUD. 54 % der Fahrer aus der Baseline-Gruppe

blickten direkt vom CID auf das Zielfahrzeug, wogegen 18 % einen kurzen Kontrollblick zurück auf die Warnanzeige im HUD machten nachdem sie zuvor auf das Zielfahrzeug geblickt haben. Die restlichen 28 % der Fahrer aus der Baseline-Gruppe blickten zuerst in das Kombiinstrument und anschließend auf das Zielfahrzeug.

4.2 Fahrverhalten

Abbildung 6 zeigt, dass die Fahrer bei der Variante *HUD* mit einer mittleren Bremsreaktionszeit von 1,59 Sekunden (SD = 0,24 s) am schnellsten reagierten. Etwas langsamer reagierten die Fahrer bei der Variante *LED* (M = 1,66 s ; SD = 0,15 s), wogegen die Variante *CID* (M = 1,89 s ; SD = 0,25 s) die längste Bremsreaktion verursachte.

Mittels Varianzanalyse konnte gezeigt werden, dass sich die Mittelwerte der Bremsreaktionszeiten signifikant unterscheiden ($F[3,75] = 4,50$; $p < .01^*$). Ein Vergleich der fahrerabhängigen Warnstrategien mit der Baseline (M = 1,75 s ; SD = 0,27 s) zeigt, dass die Variante *CID* zu einer längeren bzw. die Varianten *LED* und *HUD* zu einer kürzeren Bremsreaktionszeit führen. Um dies statistisch zu prüfen, wurde jede Variante mittels eines einseitigen t-Tests mit der Baseline verglichen. So unterscheiden sich Variante *CID* mit $p = .044^*$ und Variante *HUD* mit $p = .032^*$ signifikant von der Baseline (auch hier: Vergleiche mit einer Baseline erfordern keine Alpha-Kumulierung, siehe Argumentation von Keppel [7]). Im Vergleich zur Baseline führte die *LED*-Variante tendenziell zu kürzeren Bremsreaktionszeiten, wobei der Unterschied der Mittelwerte nicht signifikant wurde.

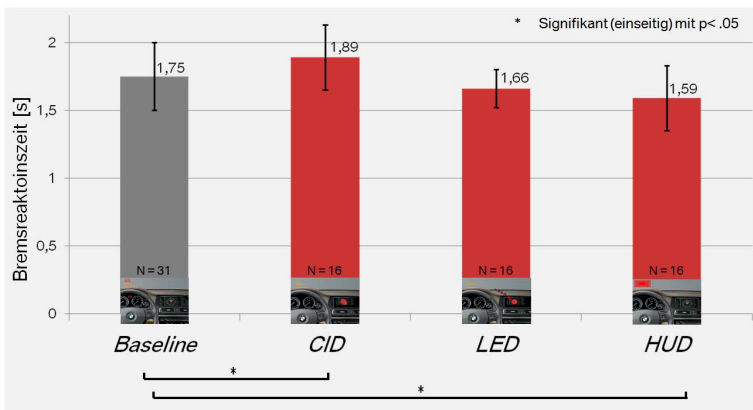


Abbildung 6 : Mittelwert-Vergleich der Bremsreaktionszeit

5 Subjektive Bewertung

Unmittelbar nach der Prüfsituation gaben 45 % der Probanden, die die Baseline erlebt hatten, an, sie hätten keine Warnung gesehen („das Vorderfahrzeug hat gebremst“). Bei der Variante *CID* verstanden 69 % die Warnung nicht als solche, und antworteten z. B. „das Navi ist abgestürzt“. Im Gegensatz dazu verstanden alle Fahrer die *LED*-Variante als Warnung und gaben zu 87 % an „ein roter Strich lenkt meinen Blick auf die Straße“, obwohl sie dies meist nicht mit den verbauten LEDs in Verbindung brachten. Alle Probanden der *HUD*-Variante verstanden diese auch als „Abstandswarnung“.

6 Diskussion

Die Verkürzung der Reaktionszeiten der Varianten *LED* und *HUD* gegenüber der Baseline zeigen, dass eine fahrerabhängige Warnstrategie – also die Anpassung von Warnausgaben an die aktuelle Blickrichtung – großes Potential haben kann. Die visuelle Aufmerksamkeit wird bei beiden Varianten schnell auf die Straße gerichtet, wodurch die Fahrer früher bremsen. Mittels der Variante *LED* kann hier erstmals die Wirksamkeit einer Scheinbewegung durch eine stroboskopische Bewegungsanimation im Fahrkontext gezeigt werden. Der Blick folgte bei allen Probanden in einer fließenden Bewegung dem Lauflicht zeitlich versetzt hinterher und endete auf dem Zielfahrzeug. Aufgrund einer sehr kurzen Reizverarbeitungszeit ($< 0,4$ s) kann davon ausgegangen werden, dass die visuelle Aufmerksamkeit schnell und automatisch auf die Gefahr gerichtet wird.

Variante *CID* hingegen hat einen negativen Effekt auf das Blick- und Fahrverhalten. Eine geführte Blickbewegung zur Straße, ausgelöst alleine durch die Animation im *CID*, hatte keine Wirksamkeit. Die Entfernung zwischen *CID* und Scheibenwurzel ist zu groß, dass der visuelle Aufmerksamkeitsfokus ohne weiteren Reiz unmittelbar auf die Fahrbahn gerichtet wird. So lässt die doppelt so hohe Blickreaktionszeit auf eine längere Reizverarbeitung durch Interpretationsprozesse schließen. Diese These wird durch häufige Fehlinterpretationen („Navi ist abgestürzt“) gestützt. Auch die hohe Streuung der Mittelwerte der Blickreaktionszeit zeigt ($SD = 0,25$ s, siehe Abb. 5), dass sich die Fahrer sehr unterschiedlich verhalten, da sie die Warnausgabe mehr interpretieren müssen und dadurch die individuelle Varianz stärker einfließt.

Obwohl das *LED*-Lauflicht meist nur unbewusst wahrgenommen wurde, konnte ihr Nutzen aufgrund der signifikanten Reaktionszeitunterschiede zu Variante *CID* deutlich nachgewiesen werden ($p = .02^*$).

Trotz kürzester Blickreaktionszeit zeigte sich bei der *HUD*-Variante ein Nachteil im Bezug auf die Blickstrategien. Die Intensität des optischen Reizes war so hoch, dass viele Fahrer erst ins *HUD* blickten und dann auf das Zielfahrzeug (bzw. einen Kontrollblick zurück ins *HUD* machten). Dagegen folgte bei der *LED*-Variante der Blick automatisch dem Lauflicht bis zur Scheibenwurzel und sprang anschließend direkt auf das Zielfahrzeug.

7 Ausblick

In diesem Beitrag wurde die Wirksamkeit von fahrerabhängigen Warnstrategien gezeigt, unter der Voraussetzung, dass die Anpassung der Warnung an den visuellen Aufmerksamkeitsfokus immer gerechtfertigt ist. Eine weitere interessante Fragestellung ergibt sich nun, wie sich eine unangebrachte Anpassung der Warnung – also eine Art fahrerabhängiger Falsch-Alarm – auf das Fahrerverhalten auswirkt.

Darüber hinaus lässt sich aus dem Prinzip der animierten Blickführung ein erweiterter Lösungsraum für fahrerabhängige Warnstrategien ableiten. Darin sind eine Vielzahl an Gestaltungsmöglichkeiten (Animationsgeschwindigkeit, Animationsinhalt, usw.) enthalten, die zur Optimierung der Wirksamkeit von Fahrerassistenzsystemen beitragen können. Ein Beispiel hierfür ist, den animierten Reiz im CID durch Semantik anzureichern (z. B. wegfliegendes Warnsymbol statt roter Punkt). Aus dem dadurch erhöhten Warncharakter könnte eine verbesserte Akzeptanz resultieren. Auch andere Fahrerassistenzsysteme wie z. B. der Spurwechselwarner könnten sich solcher Konzepte bedienen, um die Sicherheit im Straßenverkehr weiter zu erhöhen.

Literatur

- [1] M. Bayly, K. Young, and M. Regan, "Sources of Distraction inside the Vehicle and Their Effects on Driving Performance" *Driver distraction: theory, effects, and mitigation* / edited by M. Regan, J. Lee, K. Young, Taylor & Francis Group, LLC, Boca Raton, S. 191-213, 2009.
- [2] M. Kopf, "Ein Beitrag zur modellbasierten, adaptiven Fahrerunterstützung für das Fahren auf deutschen Autobahnen" *Fortschritt-Berichte VDI Reihe 12 Nr. 203* Düsseldorf: VDI-Verlag, S. 166-172, 1994.
- [3] R. Lermer, "Konzeption und Bewertung eines fahrsituations- und fahrerleistungsadaptiven Warn- und Informationsmanagers" *dissertation.de* - Verlag im Internet GmbH, Berlin, 1. Auflage, S. 117-130, 2010.
- [4] J. Lee, J. Hoffman, and E. Hayes, "Collision Warning Design to Mitigate Driver Distraction" *Tagungsbeitrag der Computer Human Interaction (CHI)*. Wien, S. 65-72, 2004.
- [5] S. Yantis and J. Jonides (1984), zitiert nach: U. Ansorge, "Die Rolle von Absichten bei der automatischen Verarbeitung visuell-räumlicher Reizinformation" *Psychologische Rundschau*, 57(1) Hogrefe Verlag. Göttingen, S. 2-12, 2006.
- [6] E. B. Goldstein, M. Ritter (Hrsg.). "Wahrnehmungspsychologie" Spektrum Akademischer Verlag. Heidelberg, Berlin, 2. dt. Auflage, S. 280-288, 2002.
- [7] G. Keppel, zitiert nach: R. Vilimek, "Design and analysis: A researcher's handbook" (3rd ed.), Englewood Cliffs, NJ: Prentice Hall, 1991.

Validierung von Engstellenszenarien und Querdynamik im dynamischen Fahrsimulator und Vehicle in the Loop

Fabian Ruger^{*}, Christian Purucker[†], Norbert Schneider[‡],
Alexandra Neukum[§] und Berthold Farber^{**}

Zusammenfassung: Die Weiterentwicklung von Fahrerassistenzsystemen fur den urbanen Raum ist derzeit Schwerpunkt nationaler und internationaler Forschungsbemuhungen. Bereits wahrend des Entwicklungsprozesses solcher Systeme, die beispielsweise kurzzeitig in die Fahrzeugquerfuhrung eingreifen sollen, ist der Nachweis der Wirksamkeit und der Kontrollierbarkeit durch den Fahrer von groer Bedeutung. Gleichwohl es zur Evaluation verschiedentliche methodische Zugange gibt, fehlen bisweilen klare Richtlinien zur Gestaltung derartiger Untersuchungen. Dies betrifft beispielsweise die Auswahl geeigneter Prufszszenarien, Pass-Fail-Kriterien in Prufanordnungen oder die Auswahl der Prufumgebung. Um Erkenntnisse uber die Validitat von Prufumgebungen fur Querfuhrungs- und Engstellenszenarien im stadtischen Raum zu gewinnen, werden in diesem Beitrag identische Fahrmanover in drei Prufumgebungen, einem dynamischen Fahrsimulator, einem Realfahrzeug auf einer Teststrecke und dem Vehicle in the Loop (VIL) - einem Hybrid aus Realfahrzeug und Fahrsimulation - untersucht. Es zeigt sich, dass Fahrer die Szenarien im dynamischen Simulator und VIL tendenziell kritischer beurteilen und die Geschwindigkeit entsprechend bei der Einfahrt in die Engstelle verlangsamen. Weiterhin ist in den Simulationsumgebungen ein systematischer Rechtsversatz von der Ideallinie zu beobachten, der sich mit abnehmender Fahrbahnbreite erhohet. Sowohl fur das VIL, als auch den dynamischen Simulator konnen bei subjektiver Bewertung, Geschwindigkeitswahl und Spurversatz relative Validitat bescheinigt werden. Die beobachteten Unterschiede zum Realfahrzeug mussen in Kontrollierbarkeitsstudien Berucksichtigung finden.

Schlusselworter: Fahrsimulation, Kontrollierbarkeit, Validierung, Vehicle in the Loop

^{*} Fabian Ruger ist wissenschaftlicher Mitarbeiter am Institut fur Arbeitswissenschaft, Universitat der Bundeswehr Munchen (fabian.rueger@unibw.de).

[†] Christian Purucker ist postgraduierter wissenschaftlicher Mitarbeiter am Interdisziplinaren Zentrum fur Verkehrswissenschaften (IZVW), Universitat Wurzburg (christian.purucker@uni-wuerzburg.de).

[‡] Norbert Schneider ist wissenschaftlicher Mitarbeiter am Interdisziplinaren Zentrum fur Verkehrswissenschaften (IZVW), Universitat Wurzburg (norbert.schneider@psychologie.uni-wuerzburg.de).

[§] Alexandra Neukum ist Leiterin des Wurzburger Instituts fur Verkehrswissenschaften (WIVW) und wissenschaftliche Mitarbeiterin am Interdisziplinaren Zentrum fur Verkehrswissenschaften (IZVW), Universitat Wurzburg (neukum@psychologie.uni-wuerzburg.de).

^{**} Berthold Farber ist Universitatsprofessor und Leiter des Instituts fur Arbeitswissenschaft der Universitat der Bundeswehr Munchen (berthold.farber@unibw.de).

1 Einleitung

Im Rahmen der Forschungsinitiative UR:BAN (Urbaner Raum: Benutzergerechte Assistenzsysteme und Netzmanagement) [1] werden verschiedene Fahrerassistenzfunktionen für den städtischen Raum entwickelt. Die untersuchten Funktionen umfassen beispielsweise kombinierte Brems- und Ausweichassistenzen für kritische Verkehrssituationen, die potenziell dynamische Eingriffe in Längs- und Querverführung beinhalten. Bereits in frühen Entwicklungsphasen muss für diese Systeme die Kontrollierbarkeit im Sinne der ISO 26262 nachgewiesen werden [2]. In diesem Kontext existieren bereits Leitlinien wie der RESPONSE Code of Practice [3], oder das European Statement of Principles on Human Machine Interface [4]. Aussagen bezüglich der anzuwendenden Evaluationsmethoden bleiben jedoch bisweilen unscharf oder uneinheitlich. Aus diesem Grund werden im Rahmen von UR:BAN in der Projektsäule Mensch im Verkehr (MV) im Teilprojekt Kontrollierbarkeit (KON) [5] Prüfmethoden weiterentwickelt und evaluiert, die zum Nachweis der Beherrschbarkeit der Systeme durch den Fahrer herangezogen werden können.

Dazu werden unter anderem Vorschläge für geeignete Zugänge zur Evaluation erarbeitet, z.B. anhand von Checklistenverfahren oder empirischen Untersuchungen. Letztere bedürfen zudem eines für das zu prüfende System angepassten Designs mit geeigneten Stichproben, Prüfscenarien und Pass-Fail-Kriterien. Eine zentrale Frage ist, in welcher Prüfumgebung ein entsprechendes FAS valide bezüglich der Kontrollierbarkeit untersucht werden kann. Dazu gibt der vorliegende Beitrag wesentliche Hinweise.

2 Prüfumgebungen für Kontrollierbarkeitsfragen

Die Wahl der Prüfumgebung, in der Kontrollierbarkeitsuntersuchungen vorgenommen werden sollen, ist ein zentraler und zugleich schwieriger Aspekt der Evaluationsmethodik. Grundsätzlich steht für derartige Untersuchungen ein breites Repertoire an Versuchsumgebungen zur Verfügung, wie beispielsweise in [6] für Frontalkollisionsgegenmaßnahmen überblicksartig dargestellt. Speziell Sicherheitsfunktionen können im realen Straßenverkehr aber oft nicht gefahrlos überprüft werden, weshalb die Untersuchung auf Teststrecken nahe liegt. Für die neuen urbanen Assistenzfunktionen, die zum Beispiel in Szenarien auf Kreuzungen mit baulichen Besonderheiten oder mit komplexem Umgebungsverkehr greifen sollen, sind Untersuchungen auf der Teststrecke nur begrenzt oder mit enormem technischem und finanziellem Aufwand realisierbar und bei Sicherheitsfunktionen zudem oft mit Risiken verbunden. In Fahrsimulatoren sind diese Beschränkungen zunächst nicht gegeben. Allerdings existieren hier möglicherweise grundsätzliche Einschränkungen bezüglich der Darstellung realer Gegebenheiten, beispielsweise bei der Bewegungsrückmeldung oder der visuellen Darstellung (vgl. [7], [8]).

Ein Ansatz, der zwischen Simulation und Realfahrzeug einzuordnen ist, ist das Vehicle in the Loop (VIL). Bei diesem System fährt der Fahrer im Realfahrzeug auf einer Teststrecke und wird über ein Head-Mounted-Display in eine virtuelle Welt eingebunden. Der Fahrer sieht also eine vollständige virtuelle Welt, spürt aber die Dynamik des Realfahrzeuges (vgl. [9]). Sowohl bei Fahrsimulatoren als auch beim VIL muss jedoch stets die Frage der Validität bzw. der Übertragbarkeit der gewonnenen Erkenntnisse auf die Realität beantwortet werden.

Die Durchführung von allgemeinen Validierungsstudien von Fahrsimulatoren kann auf eine längere Tradition zurückblicken. Für viele Fragestellungen ist der Nachweis zumindest relativer Validität für Simulationsumgebungen in verschiedenlichen Zusammenhängen erbracht (vgl. [10]). Im Hinblick auf die konkreten Fragen, welche sich aus dem Kontext des urbanen Fahrens ergeben, finden sich aber bislang nur wenige Erkenntnisse. So beziehen sich die Studien vorrangig auf

Autobahn- oder Landstraßenszenarien und zielen nur selten auf typische Spezifika des Stadtverkehrs ab. Auch gibt es nur wenige Validierungsstudien, die schwerpunktmäßig die Interaktion mit anderen Verkehrsteilnehmern, etwa in Ausweichsituationen im Stadtverkehr, betrachten.

Das VIL wurde zwar ebenfalls schon Untersuchungen zur Validität unterzogen (u.a. [11], [12], [13]), die aber nicht dem Komplex „Kontrollierbarkeit“ zugeordnet werden können, so dass auch aus diesem Grund weiterer Validierungsbedarf besteht.

3 Fragestellung

Die vorliegende Studie untersucht anhand dreier Prüfumgebungen, in einem dynamischen Fahrsimulator, einem Realfahrzeug und einem VIL das Verhalten und die Wahrnehmung von Fahrern bei der Durchfahrt einer schematisierten Engstellensituation, wie sie analog auch im Stadtverkehr vorgefunden werden kann. Sie ist damit Teil eines generischen Untersuchungsansatzes, der im Rahmen des Projekts UR:BAN MV KON im Hinblick auf die Entwicklung von Prüfmethoden zur Evaluation von urbanen Fahrerassistenzsystemen durchgeführt wird. Hierbei werden zunächst Komponenten von Fahrmanövern und Szenarien (wie etwa das Verhalten in Querregelungssituationen) separat betrachtet, um basale Erkenntnisse in einfachen Grundlagensituationen zu erhalten.

Ein Augenmerk liegt hierbei auf der Wahrnehmung der Situation durch den Fahrer. Da die Fahraufgabe stark von visuellen Informationen abhängt, können Unterschiede im Visualisierungssystem, wie die Größe des Sichtfeldes oder der Displayauflösung, zu unterschiedlichem Empfinden von Geschwindigkeit oder Kritikalität führen [14]. Auch die Güte von vestibulären Rückmeldungen beeinflusst das Situationsempfinden und hilft dem Fahrer bei der Kontrolle des Fahrzeugs [15]. Eine unterschiedliche Wahrnehmung identischer Situationen in verschiedenen Prüfumgebungen kann somit das Fahrverhalten maßgeblich beeinflussen, was wiederum zu abweichenden Ergebnissen führen kann.

Ziel der vorliegenden Studie ist es deshalb, Grundlageninformationen darüber zu liefern, wie weit sich die Prüfumgebungen im Hinblick auf Fahrerwahrnehmung und -verhalten bei der Durchfahrt von Engstellen ähneln, um in einem späteren Schritt zu untersuchen, inwiefern sich Ergebnisse zwischen den einzelnen Prüfverfahren durch mögliche Transferfunktionen übertragen lassen.

4 Versuchsdesign

Für den Versuch wurde ein between-within-subjects-Design gewählt, mit je 30 Versuchspersonen pro Prüfumgebung. Um den Einfluss von Störgrößen möglichst gering zu halten, wurden Szenario, Versuchsplan und Instruktionen über alle Prüfumgebungen hinweg identisch gehalten. Jeder Proband wurde einer Prüfumgebung zugeteilt und durchfuhr in dieser alle Ausprägungen des Szenarios in pseudorandomisierter Reihenfolge. Objektive Messdaten wurden während der Fahrt aufgezeichnet, im Nachgang waren jeweils subjektive Bewertungen abzugeben.

4.1 Prüfumgebungen

Insgesamt kamen im Versuch drei unterschiedliche Prüfumgebungen zum Einsatz.

Der dynamische Fahrsimulator des IZVW basiert auf einem voll instrumentierten BMW 520i-Mockup, das mit einer Stewart-Plattform mit sechs Freiheitsgraden und einem frontalen Sichtfeld von 180° ausgestattet ist. Als Software wird die Fahrsimulationssoftware SILAB der WIVW GmbH verwendet.

Auf der Teststrecke dient ein Audi A6 3.0 TDI Avant als Versuchsträger, der mit entsprechender Messtechnik zur Erfassung von Fahrzeugsignalen und zur exakten Positionsbestimmung mit einer iTrace RT-F200 und DGPS ausgerüstet ist.

In dasselbe Fahrzeug ist auch das VIL eingebaut, das (abweichend zur in [11] beschriebenen Konfiguration) den Fahrer mit dem Head Mounted Display (HMD) nVIS ST50 in die virtuelle Welt einbindet. Ebenfalls wurde als Weiterentwicklung zum technischen Stand in [11] die Headtracking-Latenz durch Sensorfusion von Inertial- und optischen Messverfahren wesentlich verringert. Die Verwendete Simulationssoftware ist Virtual Test Drive der Firma Vires [16].

4.2 Szenario

Als Versuchsumgebung wurde in den Simulatoren die Teststrecke der UniBw in Aussehen und Geometrie nachgebildet. Um eine maßstabsgetreue Umsetzung des Szenarios in den beiden Simulationsumgebungen zu gewährleisten, wurde zur Konstruktion auf Luftansichten und Lagepläne zurückgegriffen. Die beteiligten Fahrzeuge wurden in Größe und Farbe ebenfalls an den Realversuch angepasst.

Auf einer geraden Strecke mit zwei Fahrstreifen und Standstreifen führen die Probanden zunächst einem Vorderfahrzeug mit 50 km/h und einem Abstand von 1,5 s hinterher. Fünf Sekunden vor einer Engstelle blinkte das Vorderfahrzeug rechts und wechselte auf den Standstreifen. Der Fahrer hatte nun freie Sicht auf eine Engstelle, die durch Leitbaken markiert wurde, welche symmetrisch rechts und links der Fahrstreifenmarkierung aufgestellt waren. Das bedeutet, der Fahrer musste einen Querversatz von einer halben Fahrstreifenbreite herstellen, um die Engstelle mittig zu durchfahren (vgl. Abbildung 1).

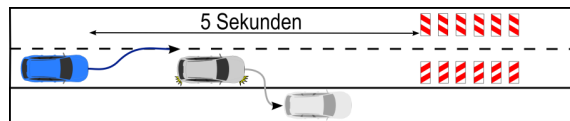


Abbildung 1: Engstellenszenario

Die rot-weiß gestreiften Leitbaken hatten eine Höhe von einem Meter und eine Breite von 25 cm und waren für den Teststreckenversuch aus Styropor hergestellt. Die Engstelle hatte insgesamt eine Länge von 15 m, die Breite variierte zwischen 2,25 m, 2,75 m, 3,25 m und 3,75 m. Die engste Gassenbreite orientiert sich dabei an Ergebnissen aus [17], wo in einem ähnlichen Versuch festgestellt wurde, dass eine Engstelle mit dieser Breite, die allerdings durch Ballooncars begrenzt war, die unterste Grenze von noch durchfahrenen Engstellen darstellt.

4.3 Versuchsdurchführung

Dem Fahrer wurde der Ablauf der Situation vor der Fahrt erklärt und Hilfestellungen zum Einhalten des geforderten Abstandes zum Vorderfahrzeug gegeben. Er bekam die Anweisung, möglichst mit der Ausgangsgeschwindigkeit von 50 km/h die Gasse zu durchfahren, bei Bedarf aber die Geschwindigkeit anzupassen oder vor der Gasse stehen zu bleiben. Die vier verschiedenen Gassenbreiten wurden systematisch variiert und nacheinander durchfahren. Es wurden Fahrtrajektorie, Lenktätigkeit wie auch Geschwindigkeit aufgezeichnet. Zudem wurden nach jeder Engstellendurchfahrt subjektive Einschätzungen anhand der Skala zur Bewertung der Kritikalität von Fahr- und Verkehrssituationen erhoben [18], die in Abbildung 2 zu sehen ist.

nicht kontrollierbar	10
gefährlich	9
	8
	7
unangenehm	6
	5
	4
harmlos	3
	2
	1
nichts bemerkt	0

Abbildung 2: Skala zur Bewertung der Kritikalität von Fahr- und Verkehrssituationen [18]

4.4 Stichprobe

In allen Prüfungsgebungen wurden Stichprobengrößen von $n = 30$ Versuchspersonen erhoben (je zur Hälfte Männer und Frauen). Wie in Tabelle 1 zu sehen, wurde zudem eine vergleichbare Altersverteilung erreicht.

Dynamischer Simulator	Teststrecke	Vehicle in the Loop
Ø 29.5 (+/-8.5) Jahre	Ø 29.2 (+/-10.9) Jahre	Ø 30.4 (+/-10.7) Jahre

Tabelle 1: Altersverteilung der Stichprobe

5 Ergebnisse

Bei der Datenauswertung durch statistische Tests wurden zunächst einfaktorielle Varianzanalysen gerechnet, um zu überprüfen ob generell zwischen den drei Prüfungsgebungen Unterschiede vorliegen (Signifikanzniveau $\alpha = 0.05$). Wenn ja, wurden im weiteren Verlauf Einzelvergleiche gerechnet, die in den Unterkapiteln aufgeführt sind.

Die Analysen ergaben, dass bei der Einfahrgeschwindigkeit signifikante Unterschiede nur bei der engsten Gassenbreite von 2.25 m vorliegen (siehe Tabelle 2). Bei der Trajektorie hingegen gibt es (mit Ausnahme der breitesten Gasse) bei allen Gassenbreiten signifikante Unterschiede. Bei der subjektiven Beurteilung ergab das nonparametrische Äquivalent zur einfaktoriellen Varianzanalyse (Kruskall-Wallis-Test) signifikante Unterschiede bei den Gassenbreiten 2.25 m und 2.75 m, bei den breiteren Gassen waren keine signifikanten Unterschiede zu finden (vgl. Tabelle 2).

Gassenbreite	2.25 m	2.75 m	3.25 m	3.75 m
Subjektive Beurteilung	$H(2) = 6.815$, $p = .033$	$H(2) = 9.835$, $p = .007$	$H(2) = 1.890$, $p = .389$	$H(2) = 1.802$, $p = .406$
Einfahrgeschwindigkeit	$F(2,86) = 9.187$, $p < .001$	$F(2,90) = 2.148$, $p = .123$	$F(2,91) = 2.759$, $p = .069$	$F(2,92) = 0.775$, $p = .624$
Trajektorie	$F(2,82) = 8.347$, $p = .001$	$F(2,90) = 14.005$, $p < .001$	$F(2,90) = 7.812$, $p = .001$	$F(2,92) = 2.939$, $p = .058$

Tabelle 2: Ergebnisse: Kruskal Wallis-Test (subjektive Beurteilung) und einfaktorielle ANOVAs

Für die im weiteren Verlauf vorgestellten Einzelvergleiche zwischen den einzelnen Prüfungsumgebungen wurde eine Bonferroni-Korrektur des Signifikanzniveaus auf $\alpha = 0.017$ vorgenommen.

5.1 Subjektive Beurteilung

Die subjektive Bewertung der Probanden anhand der Kritikalitätsskala ergibt für die beiden Simulationsumgebungen im Vergleich zur Fahrt im Realfahrzeug, eine tendenziell kritischere Einschätzung der Engstellsituation. Eine Ausnahme stellt die engste Gasse (2.25 m) dar, die im VIL etwas weniger kritisch eingestuft wird. Die Ergebnisse sind in Abbildung 3 aufgetragen, wobei die dicken schwarzen Balken die Mediane und die Punkte einzelne Ausreißer markieren.

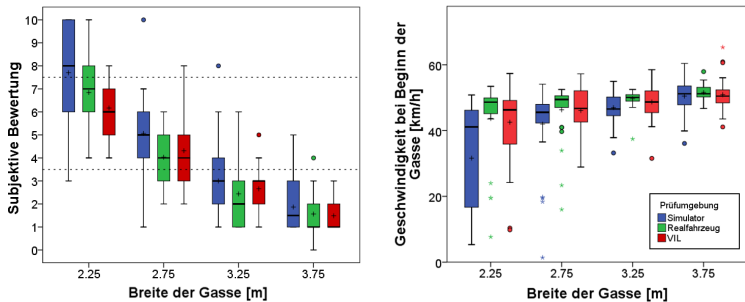


Abbildung 3: Subjektive Bewertung der Engstellsituationen und Geschwindigkeit bei Gasseneinfahrt

Bei den Paarvergleichen der subjektiven Bewertungen wurden Mann-Whitney-Tests gerechnet. Sie ergaben keine signifikanten Unterschiede mit Ausnahme der Gassenbreite 2.25 m, wo ein solcher zwischen VIL und Simulator beobachtet wird ($U = 276.500$, $z = -2.518$, $p = .012$) und der Gassenbreite 2.75 m, wo es einen signifikanten Unterschied zwischen Simulator und Realfahrt gibt ($U = 236.000$, $z = -2.835$, $p = .005$).

5.2 Einfahrtsgeschwindigkeit

Bei der Einfahrt in die Engstellen mit geringer Breite zeigt sich im VIL und im dynamischen Simulator eine geringfügig niedrigere Geschwindigkeit als auf der Teststrecke, wie man auf dem rechten Boxplot in Abbildung 3 erkennen kann.

Gassenbreite	2.25 m		2.75 m		3.25 m		3.75 m	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Simulator	31.62	(16.29)	41.96	(11.86)	46.99	(4.99)	50.47	(5.70)
Realfahrzeug	43.70	(12.11)	46.33	(8.40)	49.69	(2.72)	51.59	(2.18)
VIL	42.53	(11.51)	46.11	(7.04)	48.70	(5.39)	50.92	(4.78)

Tabelle 3: Einfahrtsgeschwindigkeit bei Engstellenbeginn in km/h (Mittelwert und Standardabweichung)

Dabei unterscheiden sich Simulator und VIL in der engsten Gassenbreite signifikant, ebenso Simulator und Realfahrzeug bei den Gassenbreiten 2.25 m und 3.25 m (vgl. Tabelle 4). Bei den Gassenbreiten 2.75 m und 3.75 m sind zwischen Simulator und Realfahrzeug keine signifikanten Unterschiede vorhanden. Dies gilt auch generell zwischen VIL und Realfahrzeug. Die entsprechenden Mittelwerte der einzelnen Ausprägungen sind in Tabelle 3 hinterlegt.

Gassenbreite	2.25 m	2.75 m	3.25 m	3.75 m
Simulator Realfahrzeug	$t(49) = -3.00,$ $p = .004$	$t(58) = -1.65,$ $p = .105$	$t(57) = -2.61,$ $p = .012$	$t(58) = -1.01,$ $p = .320$
Simulator VIL	$t(62) = -3.57,$ $p = .001$	$t(61) = -1.71,$ $p = .093$	$t(62) = -1.34,$ $p = .185$	$t(63) = -.35,$ $p = .729$
VIL Realfahrzeug	$t(57) = -.38,$ $p = .707$	$t(61) = -.112,$ $p = .911$	$t(62) = -.90,$ $p = .373$	$t(63) = -.71,$ $p = .483$

Tabelle 4: Ergebnisse der Paarvergleiche für die Einfahrtsgeschwindigkeit in die Engstelle

Diese Beobachtung deckt sich auch mit den subjektiven Urteilen und stellt eine Anpassung der Geschwindigkeit an die empfundene Kritikalität dar.

5.3 Trajektorie

Bei der Wahl der Trajektorie ist in beiden Simulationsumgebungen bei den engen Gassenbreiten von 2.25 m und 2.75 m ein leichter, systematischer Rechtsversatz im Vergleich zum Realfahrzeug zu beobachten, der sich mit zunehmender Enge der Gasse deutlicher manifestiert (siehe Abbildung 4).

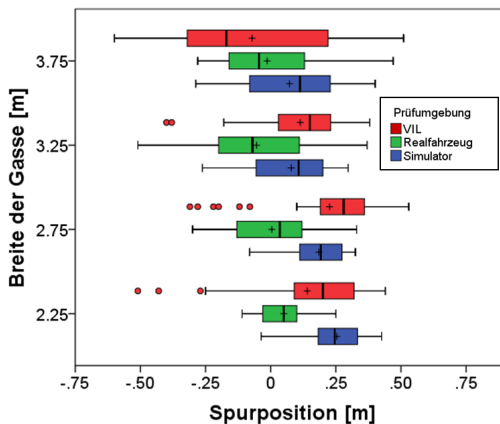


Abbildung 4: Maximale Abweichung von der Ideallinie (hier bei Spurposition: 0 m)

Dabei unterscheiden sich Simulator und Realfahrzeug, außer bei einer Gassenbreite von 3.75 m, signifikant, ebenso VIL und Realfahrzeug, außer bei den Gassenbreiten 2.25 m und 3.75 m. Allerdings muss beachtet werden, dass bei 2.25 m die Werte im VIL extrem streuen, weswegen

trotz Abweichung im Mittelwert (vgl. Tabelle 5) der Unterschied nicht signifikant wird, wie aus Tabelle 6 entnommen werden kann.

Gassenbreite	2.25 m		2.75 m		3.25 m		3.75 m	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Simulator	0.25	(0.11)	0.18	(0.10)	0.08	(0.16)	0.07	(0.03)
Realfahrzeug	0.05	(0.09)	0.00	(0.16)	0.06	(0.20)	0.01	(0.17)
VIL	0.14	(0.25)	0.23	(0.23)	0.11	(0.19)	0.07	(0.31)

Tabelle 5: Maximale Abweichung von der Ideallinie in m (Mittelwert und Standardabweichung)

Gassenbreite	2.25 m	2.75 m	3.25 m	3.75 m
Simulator	$t(49) = 7.10,$	$t(58) = 5.20,$	$t(57) = 3.01,$	$t(58) = 1.83,$
Realfahrzeug	$p = .000$	$p = .000$	$p = .004$	$p = .072$
Simulator	$t(58) = 2.13,$	$t(61) = -0.90,$	$t(62) = -0.78,$	$t(63) = 2.20,$
VIL	$p = .038$	$p = .370$	$p = .441$	$p = .032$
VIL	$t(57) = 1.71,$	$t(61) = 4.39,$	$t(61) = 3.59,$	$t(63) = -0.90,$
Realfahrzeug	$p = .092$	$p = .000$	$p = .001$	$p = .370$

Tabelle 6: Ergebnisse der Paarvergleiche für Abweichung von der Ideallinie

6 Fazit

Gemäß den Definitionen in [19], [20] und [21] wird für das VIL und den dynamischen Simulator relative Validität bei subjektiver Bewertung, Geschwindigkeitswahl und Spurversatz erreicht. Gleichwohl ist einschränkend anzumerken, dass die beobachteten systematischen Verzerrungen in Wahrnehmung und Verhalten in Engstellungssituationen bei der Interpretation von Resultaten aus diesen Prüfungsumgebungen berücksichtigt werden müssen.

Die teilweise kritischere Einschätzung von Engstellungssituationen im dynamischen Simulator und VIL ist ein Phänomen, das auch in weiteren Szenarien beobachtet wurde, die im Rahmen dieser Validierungsstudie durchgeführt wurden. Grundsätzlich ist dadurch die Empfindung der Kritikalität bei gleicher Parametrierung von Kontrollierbarkeitsszenarien in den beiden Simulationsumgebungen höher, wodurch Ergebnisse konservativer als auf der Teststrecke zu bewerten sind.

Die Betrachtung der Geschwindigkeit zeigt grundsätzlich, dass die Fahrer bei Einfahrt in engere Gassenbreiten ihre Geschwindigkeit verringern – im Simulator aber deutlicher als im Realfahrzeug. Dies deckt sich weitgehend mit Beobachtungen, wonach Geschwindigkeitsanpassungen bedingt durch bauliche Maßnahmen im Simulator als gleich [21] oder stärker [22] gegenüber dem Realfahrzeug beschrieben werden.

Ursachen für den Rechtsversatz in den Simulationsumgebungen sowie die stärkere Geschwindigkeitsreduktion sind möglicherweise in der erhöhten Schwierigkeit, die Eigenposition im Fahrsimulator valide bestimmen zu können, begründet. So fehlt zum derzeitigen Entwicklungsstand im VIL eine Ansicht des Cockpits und der Motorhaube. Im dynamischen Simulator besteht ein räumlicher Abstand zwischen Fahrer und Projektionsfläche, wodurch der Abstand zwischen eigenem Fahrzeug und den Hindernissen nur erschwert anhand visueller Reize eingestellt werden kann (vgl. [14]).

Die Erkenntnisse dieser Studie liefern für künftige Kontrollierbarkeitsuntersuchungen zunächst zwei Aussagen. Zum einen ist darauf zu achten, dass gefahrene Trajektorien bei Engstellenmanövern zwischen Realfahrzeug und Simulator einen systematischen Versatz aufweisen können, der jeweils spezifisch bestimmt werden sollte, um Ergebnisse richtig zu interpretieren.

Zum anderen werden identische Szenarien in Simulationsumgebungen tendenziell kritischer eingeschätzt als im Realfahrzeug. Das bedeutet, Ergebnisse aus Simulatoren fallen konservativer aus und werden im Realfahrzeug weniger gefährlich erlebt.

Aufbauend auf diesen Beobachtungen sind im VIL Vergleiche mit Video-See-Through- Fahrten geplant. Zur genaueren Bestimmung des Einflusses von Beschleunigungsreizen sollen Untersuchungen derselben Szenarien in einem statischen Simulator stattfinden.

Aus diesen Resultaten werden im weiteren Projektverlauf die Übertragbarkeit von Ergebnissen aus unterschiedlichen Prüfungsumgebungen beleuchtet und gegebenenfalls Transferfunktionen errechnet werden.

Anmerkung

Dieser Beitrag entstand im Rahmen von UR:BAN (Urbaner Raum: Benutzergerechte Assistenzsysteme und Netzmanagement). Das UR:BAN-Projekt wird gefördert vom Bundesministerium für Wirtschaft und Energie aufgrund eines Beschlusses des Deutschen Bundestages.

Literatur

- [1] C. Lehsing, K. Bengler, F. Busch und T. Schendzielorz, „UR:BAN – the German Research Initiative for User Centered Driver Assistance,“ in *mobil TUM*, München, 2013.
- [2] ISO 26262, *Road vehicles - Functional Safety*, International Organization for Standardization, 2011.
- [3] RESPONSE Consortium, „Code of practice for the design and evaluation of ADAS,“ RESPONSE 3: a PReVENT Project, 2006.
- [4] Commission of the European Union, „Commission recommendation on safe and efficient in-vehicle information and communication systems: update of the European Statement of Principles on human machine interface,“ *Official Journal of the European Union*, Nr. L 32, pp. 200-241, 2006.
- [5] D. Mannstetten, K. Bengler, F. Busch, B. Färber, C. Lehsing, A. Neukum, I. Petermann-Stock und T. Schendzielorz, „UR:BAN MV” – a German project focusing on human factors to increase traffic safety in urban areas,“ in *Proceedings of the 20th ITS*, Tokyo, 14.-18. Oktober 2013.
- [6] N. Fecher, F. Regh, S. Habenicht, J. Hoffmann und H. Winner, „Test- und Bewertungsmethoden für Sicherheitssysteme der Bahnführungsebene,“ *Automatisierungstechnik* 56, pp. 592-600, 11 2008.
- [7] E. Blana, „Driving Simulator Validation Studies: A Literature Review,“ Working Paper, Leeds: Institute of Transport Studies, 1996.
- [8] J. Breuer, „Bewertungsverfahren von Fahrerassistenzsystemen,“ in *H. Winner, S. Hakuli & G. Wolf (Hrsg.), Handbuch Fahrerassistenzsysteme*, Wiesbaden, Vieweg + Teubner, 2012, pp. 55-68.
- [9] I. Karl, G. Berg, F. Rügner und B. Färber, „Driving Behavior and Simulator Sickness While Driving the Vehicle in the Loop: Validation of longitudinal driving behavior,“ *IEEE Intelligent Transportation Systems Magazine*, pp. 42-57, 23 Januar 2013.
- [10] N. Mullen, J. Charlton, A. Devlin und M. Bedard, „Simulator Validity: Behaviors Observed on the Simulator and on the Road,“ in *D.L. Fisher, M. Rizzo, J.K. Caird & J.D. Lee (Hrsg.), Handbook of Driving Simulation For Engineering, Medicine and Psychology*, Boca Raton, CRC Press, Taylor & Francis Group, 2010, pp. 13-11 - 13-17.

- [11] G. Berg, I. Karl und B. Färber, „Vehicle in the Loop - Validierung de virtuellen Welt,“ *Der Fahrer im 21. Jahrhundert: Fahrer, Fahrerunterstützung und Bedienbarkeit. VDI-Berichte 2134*, pp. 143-154, 2011.
- [12] M. Sieber, G. Berg, I. Karl, K.-H. Siedersberger, A. Siegel und B. Färber, „Validation of Driving Behavior in the Vehicle in the Loop: Steering Responses in Critical Situations,“ in *Proceedings of the 16th ITSC*, The Hague, Oktober 2013.
- [13] T. Bock, M. Maurer und G. Färber, „Validation of the vehicle in the loop (VIL) - A milestone for the simulation of driver assistance systems,“ in *Proceedings of the 2007 IEEE Intelligent Vehicle Symposium*, Istanbul, Turkey, 2007.
- [14] G. Anderson, „Sensory and Perceptual Factors in the Design of Driving Simulation Displays,“ in *D.L. Fisher, M. Rizzo, J.K. Caird & J.D. Lee (Hrsg.), Handbook of Driving Simulation For Engineering, Medicine and Psychology*, Boca Raton, CRC Press, Taylor & Francis Group, 2011, pp. 8-1 - 8-10.
- [15] J. Greenberg und M. Blommer, „Physical Fidelity of Driving Simulators,“ in *D.L. Fisher, M. Rizzo, J.K. Caird & J.D. Lee (Hrsg.), Handbook of Driving Simulation For Engineering, Medicine and Psychology*, Boca Raton, CRC Press, Taylor & Francis Group, 2011, pp. 7-1 - 7-24.
- [16] K. v. Neumann-Cosel, M. Dupuis und C. Weiss, „Virtual test drive - provision of a consistent tool-set for [d,h,s,v]-in-the-loop,“ in *Proceedings of the Driving Simulation Conference*, Monaco, 2009.
- [17] M. Reichel, *Situationsanalyse für fortschrittliche Fahrerassistenzsysteme*, Dissertation, TU Braunschweig, 2013.
- [18] A. Neukum, T. Lübbeke, H.-P. Krüger, C. Mayser und J. Steinle, „ACC-Stop&Go: Fahrerverhalten an funktionalen Systemgrenzen,“ in *5. Workshop Fahrerassistenzsysteme - FAS 2008*, Karlsruhe, M. Maurer & C. Stillner, 2008, pp. 141-150.
- [19] G. Blaauw, „Driving experience and task demands in simulator and instrumented car: A validation study,“ *Human Factors*, vol. 24, no. 4, pp. 473-486, 1982.
- [20] J. Törnros, „Driving behaviour in a real and a simulated road tunnel - a validation study,“ *Accident Analysis & Prevention*, Nr. 30(4), pp. 497-503, 1998.
- [21] S. Godley, T. Triggs und B. Fildes, „Driving simulator validation for speed research,“ *Accident Analysis & Prevention*, Nr. 34, pp. 589-600, 2002.
- [22] J. Riemersma, A. van der Horst und W. Hoekstra, „driving simulator in evaluation speed-reduction measures,“ *Traffic Engineering and Control*, Nr. 31, pp. 416-420, 1990.

Evaluation einer taktischen Verhaltensentscheidungsfindung für Fahrstreifenwechsel beim vollautomatisierten Fahren in Städten

Simon Ulbrich* und Markus Maurer†

Zusammenfassung: Im Rahmen des Stadtpilot-Projekts wurden bereits einige Aspekte des vollautomatisierten Fahrens in urbanen Räumen erfolgreich demonstriert. Diese Veröffentlichung adressiert speziell die Erprobung und Evaluation der Algorithmen der Autoren zur taktischen Verhaltensentscheidungsfindung für Fahrstreifenwechsel beim Fahren in urbanen Räumen. Um die Unsicherheit der Umfeldfassung adäquat bei der Entscheidungsfindung berücksichtigen zu können, wird ein partiell beobachtbarer Markov-Entscheidungsprozess in einem zweistufigen Verfahren aus Vorverarbeitung und Entscheidungsfindung eingesetzt. Es werden die Ergebnisse einer Evaluation auf dem Testgelände wie auch im realen Straßenverkehr präsentiert und Metriken zur Bewertung eingeführt und angewandt.

Schlüsselwörter: Fahrstreifenwechsel, Entscheidungsfindung, taktisch, POMDP

1 Einleitung

Innerhalb der letzten zwanzig Jahre und insbesondere nach den DARPA Grand Challenges hat sich das vollautomatisierte Fahren zum Forschungsgegenstand vieler Teams weltweit entwickelt. Nach der Teilnahme des Teams der TU Braunschweig an der DARPA Urban Challenge in 2007 [1], hat sich das Team im Rahmen des Stadtpilot-Projekts das vollautomatisierte Fahren in urbanen Räumen als neuen Fokus gesucht. Ziel des Stadtpilot-Projekts ist das vollautomatisierte Fahren auf Braunschweigs Innenstadtring. Erste Erfolge wurden hierbei bereits erreicht und der Öffentlichkeit demonstriert [2]. In einer vorausgegangenen Veröffentlichung [3] wurde bereits das Grundkonzept zur probabilistischen Entscheidungsfindung mittels POMDPs gezeigt. Im Rahmen dieses Papers soll insbesondere auf die Evaluation und Erprobung dieser Ansätze eingegangen werden.

Das Handhaben von Unsicherheiten in der Wahrnehmung von Umfeldobjekten und der darauf aufbauenden Entscheidungsfindung ist eine der zentralen Herausforderungen für das vollautomatisierte Fahren im öffentlichen Straßenverkehr. Eine typische Form der Repräsentation dieser Unsicherheiten sind z.B. Kovarianzmatrizen aus einer Kalmanfilter-basierten Objektverfolgung. Zusätzlich bestehen Unsicherheiten in der Ego-Lokalisierung,

*Simon Ulbrich ist wissenschaftlicher Mitarbeiter am Institut für Regelungstechnik der TU Braunschweig, Hans-Sommer-Str. 66, 38106 Braunschweig (e-mail: ulbrich@ifr.ing.tu-bs.de).

†Markus Maurer ist Leiter des Instituts für Regelungstechnik der TU Braunschweig, Hans-Sommer-Str. 66, 38106 Braunschweig (e-mail: maurer@ifr.ing.tu-bs.de).

der Fahrstreifenerkennung, oder der Objekt-zu-Fahrstreifenzuordnung. Speziell in urbanen Räumen kommt hierzu noch die Unsicherheit über die Intentionen und Verhaltensweisen der anderen Verkehrsteilnehmer hinzu. Gerade in der Stadt bestehen oft eine Vielzahl an Handlungsalternativen und Verkehrsteilnehmer überdenken ihr eigenes Handeln häufig, um unerwartete Änderungen der Situation adäquat zu berücksichtigen. Mittels eines probabilistischen, vorausplanenden Entscheidungsverfahrens ist es möglich harte Entscheidungen erst möglichst spät in der Datenverarbeitungskette zu treffen. In dieser Veröffentlichung präsentieren die Autoren Metriken zur Evaluation einer solchen Entscheidungsfindung.

Diese Veröffentlichung gliedert sich wie folgt: Zunächst wird der Aspekt der taktischen Verhaltensentscheidungsfindung von übrigen Aufgaben abgegrenzt, Anforderungen an selbige definiert und in anderen Veröffentlichungen genutzte Evaluationsmetriken präsentiert. Nach einer kurzen Wiederholung der Theorie von Markov-Entscheidungsprozessen wird der von den Autoren gewählte zweistufige Ansatz skizziert. Besonderer Fokus wird in dieser Veröffentlichung auf die Evaluation der Ansätze in Kapitel 4 gelegt. Kapitel 6 rundet diese Veröffentlichung ab und zeigt zukünftige Forschungsrichtungen auf.

2 Hintergrund

2.1 Taktische Verhaltensentscheidungsfindung als Teil der gesamten Fahraufgabe

Nach Donges [4] und Maurer [5] lässt sich die Fahraufgabe in verschiedene Hierarchieebenen gliedern. Donges führte eine Klassifikation in Navigationsaufgaben (strategische Ebene), Führungsaufgaben (taktische Ebene) und Stabilisierungsaufgaben (Operationelle Ebene) ein. Abbildung 1 veranschaulicht das Zusammenspiel der unterschiedlichen Ebenen der Entscheidungsfindung für einen Fahrstreifenwechsel. Alle diese Ebenen sind zur erfolgreichen Ausführung eines Fahrstreifenwechsels notwendig. In dieser Veröffentlichung liegt der Fokus auf der taktischen Ebene. In Ulbrich et al. [3] werden auch die Aufgaben auf den übrigen Ebenen diskutiert.

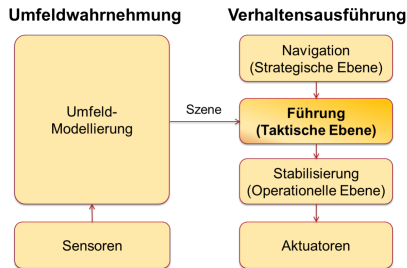


Abbildung 1: Taktische Verhaltensentscheidungsfindung als Teilproblem der gesamten Fahraufgabe

2.2 Evaluation taktischer Verhaltensentscheidungsfindungen für Fahrstreifenwechsel

Abbildung 2 zeigt eine typische Entscheidungssituation für Fahrstreifenwechsel in städtischen Umgebungen. Ob ein Fahrstreifenwechsel *möglich* ist, ergibt sich aus den relativen Distanzen, Geschwindigkeiten und Beschleunigungen der umgebenden Verkehrsteilnehmer. Ob ein Fahrstreifenwechsel *vorteilhaft* ist, ist abhängig vom Straßennetzwerk und dem Verhalten der übrigen Verkehrsteilnehmer. Eine Verhaltensentscheidungsfindung für Fahrstreifenwechsel sollte möglichst schnell, konsistent, vorausschauend und vorhersagbar sein. Eine ausführlichere Diskussion dieser Anforderungen findet sich in Ulbrich et al. [3].

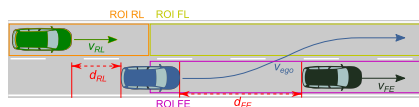


Abbildung 2: Typisches Szenario eines Fahrstreifenwechsels mit zwei dynamischen Objekten und drei Regions of Interest (ROI) hinten links (RL), vorne links (FL) und vorne im Ego-Fahrstreifen (FE)

In der vorangegangenen Veröffentlichung [3] haben die Autoren bereits ein umfassendes Literaturreview zu den von anderen Gruppen eingesetzten Verfahren zur (taktischen) Entscheidungsfindung gemacht. Daher wird an dieser Stelle auf eine ausführliche Diskussion verzichtet. Taktische Verhaltensentscheidungen zu Fahrstreifenwechseln wurden bereits in einigen anderen Veröffentlichungen evaluiert. So evaluierten Naranjo et al. [7] das Überholen eines stehenden und eines fahrenden Fremdfahrzeugs. In beiden Fällen wurde jedoch einzig ein zu überholendes Fahrzeug ohne umgebenden Fremdverkehr betrachtet. Schubert et al. [8] berichten von einer Evaluation ihres Ansatzes zur Fahrstreifenwechselentscheidung im Versuchsträger. Bei ihren Ansätzen werden Objekte in allen Regionen um das Fahrzeug berücksichtigt, jedoch wird in der Evaluation nur eine exemplarische Szene präsentiert. Metriken zum Messen und Vergleichen der Entscheidungsperformanz werden nicht gegeben. Wei et al. [9] evaluieren ihren Entscheidungsfindungsalgorithmus für Fahrstreifenwechsel auf einer simulierten Teststrecke. Sie messen die Zeit bis zum Erreichen des Fahrziels und zählen die Anzahl an durchgeführten Fahrstreifenwechsel. Eine niedrigere Anzahl ist für Wei et al. ein Maß für eine höhere Entscheidungskonsistenz. Brechtel et al. [10] evaluieren ihren Ansatz zum Fahrstreifenwechsel-Entscheidung ebenfalls in einer Simulation. Sie fokussieren ihre Evaluation auf die Präsentation der sich ergebenden Aktionen in einer mehrdimensionalen Zustandsraumdarstellung der Szene. Spezielle Bewertungsmetriken werden nicht eingeführt oder verwendet.

2.3 MDPs und POMDPs

Markov-Entscheidungsprobleme (engl. Markov decision processes, MDPs) sind ein allgemeines Framework zur Modellierung von Entscheidungs- und Planungsproblemen. Das Ausführen einer Folge von Aktionen $u \in U$ für ein System im Systemzustand $x \in X$ wird als Ausführen einer Politik $\pi : x \rightarrow u$ bezeichnet. Das Ziel eines solchen Planungsproblems

ist das Finden einer nach einem gegebenen Kostenfunktional optimalen Folge von Aktionen π^* , welche die erwartete Belohnung R_T über einen Planungshorizont T maximieren, indem die Belohnungen r_τ in jedem Zeitschritt τ mit γ diskontiert werden:

$$R_T = E\left[\sum_{\tau=0}^T \gamma^\tau * r_\tau\right] \quad (1)$$

Für reale Anwendungen ist der wahre Systemzustand jedoch häufig nicht komplett beobachtbar. Partiiell beobachtbare Markov-Entscheidungsprozesse greifen die Idee hinter einem Verborgenen Markov-Modell (engl. Hidden Markov Model) auf und repräsentieren die Unsicherheit über einen wahren Systemzustand in Form eines Zustandsglaubens bzw. Zustandsvertrauens (engl. belief) $bel(x_t)$ an einen Systemzustand x_t zum Zeitpunkt t .

POMDPs werden aufgrund ihrer Komplexität nur selten in Anwendungsfällen eingesetzt, bei denen eine schnelle Berechenbarkeit notwendig ist. Für die Anwendbarkeit von POMDPs auf reale Entscheidungsprobleme wurde eine Vielzahl von Approximationsverfahren entwickelt.

3 Umsetzung

In Abbildung 3 wird der in Quelle [3] im Detail diskutierte zweistufige Entscheidungsalgorithmus skizziert. Die in der Umfeldwahrnehmung erkannte Szene wird um die in der strategischen Ebene geplanten Missionsziele des Fahrzeugs zur aktuellen Situation erweitert. Um die Komplexität der hoch-dimensionalen Situation für den Entscheidungsprozess zu vereinfachen wird ein Signalverarbeitungsnetzwerk vor den eigentlichen Entscheidungsprozess vorgeschaltet. Dieses bewertet die Situation bezüglich der Frage, ob ein Fahrstreifenwechsel möglich ist und ob selbiger vorteilhaft ist. Auf diesem nun vereinfachten, weiter abstrahierten Zustandsraum wird ein Markov-Entscheidungsprozess gerechnet. Dieser wählt dann immer wieder, für jede neue Situation in der derzeitigen Implementierung alle 100 ms eine neue, nach einem Kostenfunktional optimale Aktion aus. Diese wird dann an die Trajektorienplanung übergeben, um eine konkrete Trajektorie für das gewünschte Manöver zu berechnen.

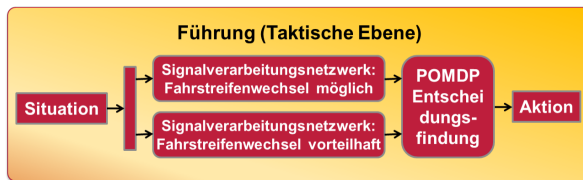


Abbildung 3: Genereller Ablauf des zweistufigen Entscheidungsprozesses: Ein Signalverarbeitungsnetzwerk vereinfacht die Komplexität des Zustandsraumes, ein POMDP-Entscheidungsprozess wählt die nach dem Kostenfunktional optimalen Aktion aus

Als Markov-Entscheidungsprozess wird eine Modellierung mit partieller Beobachtbarkeit der Zustandsgrößen (POMDP) gewählt. Um selbige hinreichend schnell (z.B.

$\leq 100\text{ ms}$ Zykluszeit) zu berechnen, ist es notwendig Approximationsverfahren einzusetzen. Im Rahmen dieser Veröffentlichung wird eine Variante des von Paquet et al. [12] vorgestellten „Real-Time Belief Space Search“-Ansatzes eingesetzt. Analog zu Ross et al. [13] wird eine obere und untere Grenze für die Wert-Funktion $V^*(x)$ bestimmt und die Suche einer möglichst optimalen Folge von Aktionen im Belief-Baum der aus den Aktionen folgenden, zukünftigen Beliefs per Branch-and-Bound-Suche (Verzweigen-und-Begrenzen-Suche) effizienter gestaltet. Details des gewählten Ansatzes und deren konkrete Umsetzung für das Fahrstreifenwechsel-Entscheidungsproblem können Quelle [3] entnommen werden.

4 Evaluation

Die Evaluation der Verhaltensentscheidungsfindungs-Algorithmen für Fahrstreifenwechsel gliedert sich in drei Aspekte: Zunächst wird per Simulation die korrekte Funktionsweise der Algorithmen überprüft. Dann werden besonders kritische Situationen auf einem Testgelände speziell nachgestellt und analysiert. Zuletzt werden die Algorithmen beim automatisierten Fahren in der Stadt selbst evaluiert.

4.1 Evaluation in der Simulation

Als erster Schritt wird die generelle Funktionsfähigkeit der Algorithmen in Virtual Test Drive (VTD)¹ simuliert. Hierfür wurde eine hoch genaue Straßenkarte in eine OpenDrive-kompatible Straßenkarte für VTD umgewandelt und aus VTD eine Pose des Ego-Fahrzeugs in die Stadtpilot-Algorithmen gegeben. Ebenso werden in VTD simulierte Umfeldobjekte mit virtuellen Sensoren ausgelesen und als Sensorobjekte in die Stadtpilot-Algorithmen gegeben. Steuergrößen, wie der Lenkwinkel oder die Längsbeschleunigung werden genutzt, um mit einem einfachen Fahrzeugmodell einen neuen Ort zu berechnen, an dem das Ego-Fahrzeug im nächsten Zeitschritt im VTD gesetzt werden soll. Aus Platzgründen wird hier auf eine ausführlichere Diskussion verzichtet. Sie wird jedoch in einer späteren Veröffentlichung nachgeholt werden.

4.2 Ausführen eines Fahrstreifenwechsels



Abbildung 4: Videostream der Rück- (links) und Frontkamera (rechts) beim Ausführen eines Fahrstreifenwechsels mit einem Verkehrsteilnehmer auf dem Nachbarfahrstreifen

Abbildung 4 zeigt das Durchführen eines Fahrstreifenwechsels gemäß Abbildung 2. Das Ego-Fahrzeug fährt auf dem rechten, ein Fremdfahrzeug fährt auf dem linken Fahr-

¹www.vires.com

streifen hinter dem Ego-Fahrzeug auf einer zweispurigen Straße. Das Fremdfahrzeug auf dem rechten Nachbarfahrstreifen fährt langsamer als das Ego-Fahrzeug, daher vergrößert sich der Abstand zu dem Fremdfahrzeug auf dem Nachbarfahrstreifen. Abbildung 5 zeigt das Votum des Signalverarbeitungsnetzwerks, den internen POMDP-Belief, die POMDP-Aktionsentscheidung und das Votum eines menschlichen Experten, ob ein Fahrstreifenwechsel zum aktuellen Zeitpunkt möglich ist. Das Votum des Signalverarbeitungsnetzwerks und des Menschen, dass ein Fahrstreifenwechsel möglich sei, steigt ab $t = 31.4\text{ s}$. Ebenso steigt der Belief (Zustandstrauen) des POMDPs und bei $t = 31.9\text{ s}$ ist das Vertrauen des POMDP hoch genug, als dass die Belohnung einen Fahrstreifenwechsel durchzuführen die Bestrafung durch mögliche Restkollisionswahrscheinlichkeiten überwiegt. Die Trajektorienplanung wird angetriggert eine Fahrstreifenwechseltrajektorie auf den Nachbarfahrstreifen zu planen. Zum Zeitpunkt $t = 35.8\text{ s}$ überquert die Fahrzeugmitte des Ego-Fahrzeugs die Mittellinie zwischen den beiden Fahrstreifen. Der Ego-Fahrstreifen ist nun der linke Fahrstreifen, der Nachbarfahrstreifen ist rechts. In den letzten beiden Bildern der Frontkamera in Abbildung 4 ist ein auf dem linken Fahrstreifen stehendes Fremdfahrzeug zu erkennen (das schwarze Fahrzeug „FE“ in Abbildung 2). Dieses hat das Ego-Fahrzeug ursprünglich zu dem gerade durchgeführten Fahrstreifenwechsel motiviert. Nach dem Durchführen des Fahrstreifenwechsels wird nun dieses ehemalige „FE“-Fahrzeug zu einem auf dem Nachbarfahrstreifen stehendem Fremdfahrzeug, auf welches das Ego-Fahrzeug schnell aufschließt. Daher sinkt sowohl das „Fahrstreifenwechsel-möglich“-Votum des menschlichen Experten, wie auch des Signalverarbeitungsnetzwerks und des POMDPs ab $t = 39\text{ s}$ wieder ab. Die Ausreißer in der Kurve des Signalverarbeitungsnetzwerks entstehen durch Abrisse/Fehlziele des Objekttrackings im verwendeten Lidar-Sensorsetup.

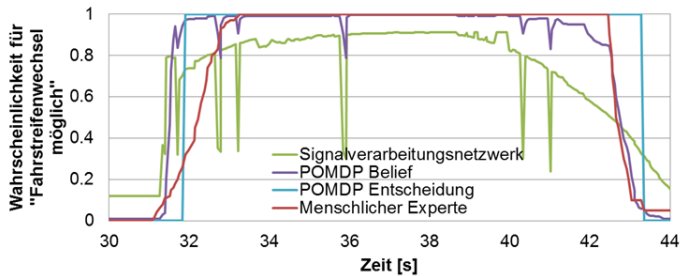


Abbildung 5: Zeitlicher Verlauf für ein „Fahrstreifenwechsel möglich“-Votum aus einem Signalverarbeitungsnetzwerk, der POMDP-Belief, die gewählte POMDP-Aktion und als Referenz das Votum eines menschlichen Experten

4.3 Abbrechen eines Fahrstreifenwechsels

Abbildung 6 zeigt das Abbrechen bzw. Überdenken eines Fahrstreifenwechsels gemäß Abbildung 2. Das Ego-Fahrzeug fährt auf dem rechten, ein Fremdfahrzeug fährt auf dem linken Fahrstreifen hinter dem Ego-Fahrzeug. Zunächst ist eine ausreichend große Lücke



Abbildung 6: Videostream der Rück- (links) und Frontkamera (rechts) beim Abbrechen eines Fahrstreifenwechsels mit einem Verkehrsteilnehmer auf dem Nachbarfahrstreifen

vorhanden, so dass gemäß Abbildung 7 ein Fahrstreifenwechsel gemäß des Votums eines menschlichen Experten wie auch dem Votum des POMDPs und dem Signalverarbeitungsnetzwerk möglich ist. Um das Abbrechen eines Fahrstreifenwechsels untersuchen zu können, wird dieser in der taktischen Ebene entschieden, jedoch vom Normalbetrieb abweichend nicht an die Trajektorienplanung übergeben, so dass kein Querversatz zum Fahrstreifen aufgebaut wird. Bei Zeitschritt $t = 44\text{ s}$ beschleunigt das Fahrzeug auf dem Nachbarfahrstreifen, so dass ein Fahrstreifenwechsel nicht mehr gefahrlos möglich ist. Das Votum, des menschlichen Experten, dass ein Fahrstreifenwechsel noch möglich ist, sinkt. Ebenso das des POMDPs. Bei $t = 45.3\text{ s}$ entscheidet der POMDP den Fahrstreifenwechsel abzubauen. Bei $t = 46.2\text{ s}$ wird das Ego-Fahrzeug von dem Fremdfahrzeug überholt. Der Peak in den Kurven in Abbildung 7 resultiert aus einem kurzen Verlieren des Objekts im Objekttracking beim Überholvorgang. Ab $t = 48.5\text{ s}$ ist das Fremdfahrzeug weit genug entfernt, so dass ein Fahrstreifenwechsel wieder eingeleitet werden kann.

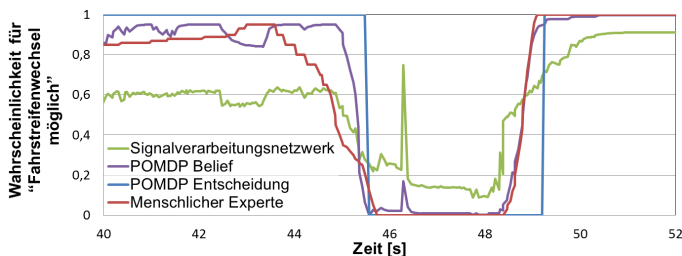


Abbildung 7: Zeitlicher Verlauf für ein „Fahrstreifenwechsel möglich“-Votum aus einem Signalverarbeitungsnetzwerk, der POMDP-Belief, die gewählte POMDP-Aktion und als Referenz das Votum eines menschlichen Experten

5 Evaluation in Straßenverkehr

Nachdem die korrekte Funktionsweise der Algorithmen in einigen ausgewählten Szenarien gezeigt werden konnte, wurde der komplette Freigabeprozess durchlaufen, um die Algorithmen auch im realen Straßenverkehr einzusetzen. Verglichen mit dem Fahren im

eigenen Fahrstreifen sind Fahrstreifenwechsel deutlich risikobehaftetere Manöver, wenn diese vom automatisiert fahrenden Fahrzeug komplett selbst entschieden werden. Denn, um eine angeplante Lücke tatsächlich zu treffen, steht häufig nur wenig Zeit zum Übersteuern/Abbrechen eines Fahrstreifenwechsels zur Verfügung, sobald dieser einmal begonnen wurde. Daher werden derzeit die Fahrstreifenwechsel dem Sicherheitsfahrer per Sprachausgabe angekündigt und es wird mit der Durchführung der Fahrstreifenwechsel abgewartet, bis der Sicherheitsfahrer diese bestätigt hat.



Abbildung 8: Zur Evaluation genutzter Streckenabschnitt in Braunschweigs Innenstadt

Für diese Veröffentlichung bestehen nicht die Ressourcen zur Durchführung einer umfassenden Feldstudie zur Akzeptanz des Fahrgefühls bei Fahrstreifenwechseln. Daher wird diese Evaluation auf technische Metriken der Zustandsschätzung beschränkt, anstatt auch die adäquate Aktionsauswahl und das Fahrgefühl mit zu evaluieren. Die Evaluation wurde somit in der Art durchgeführt, als dass die Zustandsschätzung - ob ein Fahrstreifenwechsel möglich ist - betrachtet wird, nicht jedoch der Fahrstreifenwechsel wirklich ausgeführt wird. Dazu wurden die Sensorrohdaten wie auch die Ergebnisse des Entscheidungsalgorithmus aufgezeichnet. Als Grundwahrheit (engl. Ground Truth), ob ein Fahrstreifenwechsel in einer gegebenen Situation möglich ist, wurden die Aufzeichnungen manuell durch Beurteilen der Situation auf einer Skala zwischen 0 und 1 im Videostream gelabelt.

Tabelle 1: Evaluation der absoluten Entscheidungsgüte gegenüber der angenommenen Grundwahrheit eines Menschen als Situationsbewerter durch Pearson Korrelation und normalisierte Transinformation

Methode	Pearson Korrelation	normierte Transinformation
Signalverarbeitungsnetzwerk alleine	0.876	0.38
Gefiltertes Signalverarbeitungsnetzwerk	0.884	0.39
POMDP	0.889	0.48

Zur Evaluation wurde der in Abbildung 8 gezeigte, bereits in Saust et al. [2] vorgestellte Abschnitt des Braunschweiger Stadtrings genutzt. Es wurde angenommen, dass ein Fahrstreifenwechsel zu jeder Zeit erwünscht/notwendig ist. Situationen in denen ein Fahrstreifenwechsel nicht sinnvoll wäre, z.B. während des 180°-Wendemanövers wurden ausgeschlossen. Insgesamt wurde eine Sequenz von 300 Sekunden Fahrt im Innenstadverkehr zur Evaluation genutzt.

In einer vorhergehenden Veröffentlichung [3] wurde bereits eine Entscheidungsgüteevaluation mittels Receiver-Operator-Kurven (ROC), linearer Pearson-Korrelationen und normalisierter Transinformation präsentiert. Nachdem ROC-Kurven jedoch von binären

Entscheidungssituationen ausgehen und Fahrstreifenwechselentscheidungen zumeist eher graduelle Entscheidungen sind und zugleich sich eine Einschränkung auf lineare Korrelationen aus der Problemstellung nicht motivieren lässt, halten die Autoren eine Evaluation mittels der auf Shannon basierenden normalisierten Transinformation zwischen der Zustandsschätzung der Algorithmen X und den gelabelten Grundwahrheitsdaten Y für am sinnvollsten. In der vorausgegangenen Veröffentlichung [3] konnte nach Tabelle 1 gezeigt werden, dass der POMDP-basierte Ansatz deutlich besser mit dem Grundwahrheitsvotum übereinstimmt, als wenn nur das Signalverarbeitungsnetzwerk alleine eingesetzt wird. Gegenüber dem per Tiefpassfilter geglättetem Ergebnis der Entscheidungsfindung zeigt der Pearson-Koeffizient eine kaum bessere Entscheidungsqualität des POMDPs, die normalisierte Transinformation deutet sogar auf eine gewisse Verbesserung der Entscheidungsgüte.

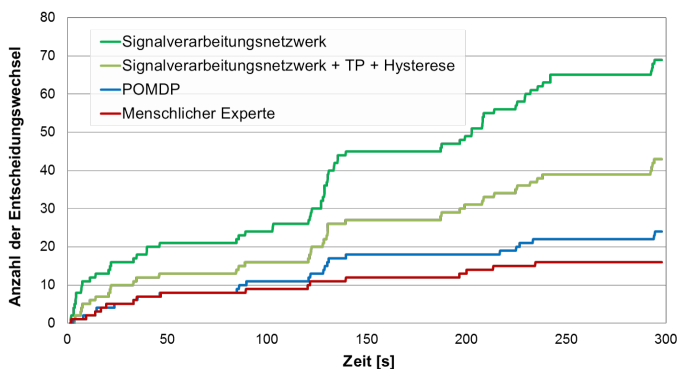


Abbildung 9: Anzahl der Entscheidungsänderungen über die Zeit vom Signalverarbeitungsnetzwerk, dem gefilterten und per Hysterese konsistenter gemachten Ergebnisses des Signalverarbeitungsnetzwerks und der POMDP-basierten Entscheidungsfindung gegenüber der eines Menschen

Bereits in Abschnitt 2.2 wurde die Entscheidungskonsistenz als wichtiges Kriterium genannt. Abbildung 9 zeigt die Anzahl der Entscheidungsänderungen über die Zeit beim Fahren im urbanen Straßenverkehr. Es werden die Anzahl der Entscheidungswechsel des Signalverarbeitungsnetzwerks alleine, des gefilterten und per Hysterese konsistenter gemachten Ergebnisses des Signalverarbeitungsnetzwerks und die Ergebnisse des POMDP-Ansatzes mit der Entscheidungskonsistenz eines Menschen verglichen. Abbildung 9 zeigt, dass die Verkehrssituationen 16 Wechsel des Votums, ob gerade noch ein Fahrstreifenwechsel durchgeführt werden kann, notwendig machte. Obwohl das Signalverarbeitungsnetzwerk alleine zwar bereits eine akzeptable absolute Entscheidungsgüte zeigte, hat es dennoch viermal so viele Entscheidungswechsel gegenüber einem Menschen benötigt. Wird das Ergebnis des Signalverarbeitungsnetzwerks per Tiefpassfilter und Hysterese geglättet, so steigt die Konsistenz der Fahrstreifenwechselentscheidungen. Insgesamt hat der POMDP-

basierte Ansatz eine dem Menschen am nächsten kommende Konsistenz der Entscheidung zeigen können, bei gleichzeitig hoher absoluter Entscheidungsgüte.

6 Zusammenfassung

Diese Veröffentlichung präsentiert eine ausführliche Evaluation der Algorithmen der Autoren zur Verhaltensentscheidungsfindung für Fahrstreifenwechsel. Hierfür wurden die verschiedenen Stufen der Evaluation vorgestellt. Nach einer simulativen Evaluation wurden ausgewählte Situationen mit realen Fahrzeugen auf einem Testgelände nachgestellt. Es konnte die korrekte Funktionsweise der Algorithmen sowohl beim Durchführen als auch beim Revidieren einer Fahrstreifenwechselentscheidung gezeigt werden. Daran anschließend wurde eine Evaluation der Ergebnisse im realen, urbanen Straßenverkehr gezeigt. Hierbei konnte eine deutlich höhere Konsistenz der Entscheidungen bei einem gleichbleibend hohen bzw. sogar leicht besserem Grad an Entscheidungsgüte gezeigt werden. Die gewonnene Konsistenz wird mit dem Nachteil einer deutlich höheren Rechen-, Modellierungs- und Parametrisierungskomplexität des POMDP-basierten Ansatzes gegenüber einem einfachen, nicht vorausplanenden, reaktiven Ansatzes erkauft. Im Rahmen der Evaluation konnte jedoch gezeigt werden, dass durch Approximationen und das zweistufige Vorgehen eine für den Praxiseinsatz geeignete Lösung gefunden werden kann.

Im Rahmen dieser Veröffentlichung wurde kein spezieller Schwerpunkt auf Driver-in-the-loop Szenarien gelegt. Zum Transfer der Ansätze in den Bereich der Fahrerassistenzsysteme ist die Mensch-Maschine-Interaktion jedoch besonders essenziell. Die hier erprobten, vorausplanenden Ansätze lassen sich jedoch besonders gut in diese Richtung erweitern, indem zusätzliche Zustände wie „Fahrstreifenwechsel vorbereiten“ oder „Fahrstreifenwechsel an Fahrer vorschlagen“ eingeführt werden. Insbesondere für die Mensch-Maschine-Interaktion ist auch bei der Information des Fahrers die mit den hier gezeigten Algorithmen gewonnene Entscheidungskonsistenz von zentraler Bedeutung.

Danksagung

Die Autoren danken dem *Stadtпилot* Team, namentlich Dr. Bernd Lichte, Tobias Nothdurft, Falko Saust, Andreas Reschka, Sebastian Ohl, Fabian Schuldt, Jaebum Choi, Jens Rieken, Richard Matthaei und Sven Böhme für Ihre Unterstützung beim Verbessern von Hard- und Software des Forschungsfahrzeugs „Leonie“ und für das Investieren von Tagen und Wochen auf dem Testgelände und im öffentlichen Straßenverkehr. Richard Matthaei sei für die konstruktiven Diskussionen zur Fahrzeugarchitektur in Kapitel 2.1 gedankt; Fabian Schuldt für einen maßgeblichen Teil der Virtual-Test-Drive-Implementierung.

Literatur

- [1] Rauskolb, F. W., Berger, K., Lipski, C. et al.: „Caroline: An autonomously driving vehicle for urban environments,“ in: *Journal of Field Robotics*, vol. 25 (9), S. 674-724, 2008.

- [2] Saust, F., Wille, J.M., Lichte, B., Maurer, M.: „Autonomous Vehicle Guidance on Braunschweig’s inner ring road within the Stadtpilot Project,“ in: *Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, S. 169-174, Juni 2011.
- [3] Ulbrich, S., Maurer, M.: „Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving,“ in: *International IEEE Conference on Intelligent Transportation Systems*, The Hague, Niederlande, S. 2063-2070, Okt. 2013.
- [4] Donges, E.: „A Conceptual Framework for Active Safety in Road Traffic,“ in: *Vehicle System Dynamics*, vol. 32, no. 2-3, S. 113-128, Aug. 1999.
- [5] Maurer, M.: „Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen“, Dissertation, Universität der Bundeswehr München, München, Deutschland, 2000.
- [6] Pellkofer, M.: „Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung,“ Dissertation, Universität der Bundeswehr München, München, Deutschland, 2003.
- [7] Naranjo, J. E., Gonzalez, C., Garcia, R., de Pedro, T.: „Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver,“ in: *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, S. 438-450, 2008.
- [8] Schubert, R., Schulze, K., Wanielik, G.: „Situation Assessment for Automatic Lane-Change Maneuvers,“ in: *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, S. 607-616, 2010.
- [9] Wei, J., Dolan J. M., Litkouhi, B.: „A Prediction- and cost function-based algorithm for robust autonomous freeway driving,“ in: *IEEE Intelligent Vehicles Symposium*, San Diego, USA, S. 512-517, 2010.
- [10] Brechtel, S., Gindele, T., Dillmann, R.: „Probabilistic MDP-behavior planning for cars,“ in: *International IEEE Conference on Intelligent Transportation Systems*, Washington, USA, S. 1537-1542, 2011.
- [11] Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics*, The MIT Press, Cambridge, 2005.
- [12] Paquet, S., Tobin, L., Chaib-draa, B., „An online POMDP algorithm for complex multiagent environments,“ in: *Proceedings of The fourth International Joint Conference on Autonomous Agents and Multi Agent Systems*, Utrecht, Niederlande, S. 970-977, 2005.
- [13] Ross, S., Pineau, J. Paquet, S. Chaib-draa, B.: „Online planning algorithms for POMDPs,“ in: *Journal of Artificial Intelligence Research*, vol. 32, S. 663-704, 2008.
- [14] Witten, I. H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, Amsterdam, S. 291, 2005.
- [15] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P.: *Numerical Recipes in C - The Art of Scientific Computing*, 2nd edition, Cambridge University Press, Cambridge, 1992.

Exposition zur Raffung der Felderprobung für Fahrerassistenzsysteme

Langfassung

Philipp Glauner*, Axel Blumenstock† und Dr. Martin Haueis‡

Zusammenfassung: Zur Erlangung der erforderlichen Systemreife von Fahrerassistenzsystemen für die Serienfreigabe werden, neben Tests im Fahrsimulator und auf Teststrecken, zahlreiche Kilometer im realen Straßenverkehr in so genannten Feldtests benötigt. Um die dafür notwendige Fahrleistung zu reduzieren, wird eine ereignisbasierte Bewertung von Fahrumfängen vorgeschlagen. Durch eine Korrelation von Ereignissen und Testdimensionen lässt sich die Felderprobung auf relevante Bereiche fokussieren und effizienter gestalten. Anhand der Exposition kann jedes Segment aus einer digitalen Karte entsprechend dem Auftreten relevanter Ereignisse bewertet werden. Diese bewerteten Segmente können schließlich für einen Routenplaner genutzt werden, welcher neue Strecken im Sinne einer effizienten Erprobung generiert. Der Mehrwert wird schließlich anhand realer Fahrdaten validiert.

Schlüsselwörter: Systemreife, Testdimensionen, Korrelation, Erprobungsplanung, Routenplanung

1 Einleitung

In modernen Kraftfahrzeugen steigt die Vielfalt und Komplexität angebotener Fahrerassistenzsysteme, besonders mit Blick auf die Einführung autonomer Teilfunktionen, rapide, und damit auch der Testbedarf für die Fahrzeughersteller. Der Nachweis des gewünschten Systemverhaltens eines Fahrerassistenzsystems erfordert, neben reproduzierbaren Tests im Fahrsimulator und auf Teststrecken, zahlreiche Kilometer in sogenannten Feldtests im realen Straßenverkehr. Hierbei wird die Alltagstauglichkeit auch unter komplexen und unerwarteten Bedingungen nachgewiesen. Aktuelle Abschätzungen des Absicherungsaufwandes zukünftiger autonomer Funktionen zum Nachweis von Unfallraten gegenüber dem Stand der Technik, gehen dabei in Bereiche von mehreren Millionen Kilometern [6], ¹. Der theoretische Raum aller Testszenarien für ein Fahrerassistenzsystem besteht aus unzähligen Dimensionen und ist zu groß, um in der Praxis vollständig abgedeckt zu werden. In der heutigen Praxis werden die Testbereiche vor allem basierend auf Expertenwissen reduziert, indem qualitative Vorgaben für Teststrecken festgelegt werden. Für die Erstellung von Routen mit hohem Erprobungsbeitrag, wird eine Fokussierung auf relevante

*Doktorand, RD/FFS, Daimler AG, 71034 Böblingen (philipp.glauner@daimler.com).

†Entwickler, RD/FFS, Daimler AG, 71034 Böblingen (axel.blumenstock@daimler.com).

‡Teamleiter, RD/FFU, Daimler AG, 71034 Böblingen (martin.haueis@daimler.com).

¹<http://www.faz.net/-gqi-7j28i>

Bereiche benötigt, um die spezifischen Anforderungen der Fahrerassistenzsysteme mit der Felderprobung koppeln zu können. Als Voraussetzung für analytische Arbeiten wurde eine Datenbank geschaffen, welche in geeigneter Struktur und Dichte alle notwendigen Messdaten enthält. Der iterative Gesamtprozess wird in Abbildung 1 dargestellt.

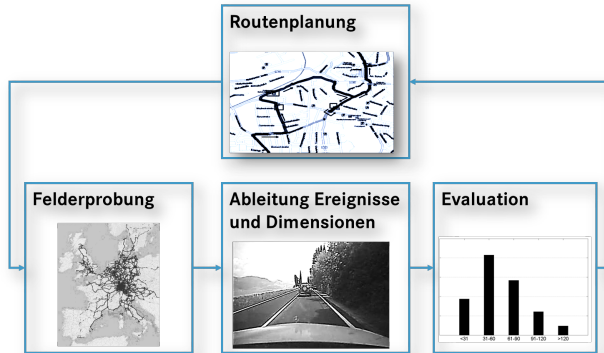


Abbildung 1: Iterativer Prozess der Felderprobung.

In der *Felderprobung* werden die Fahrzeugmessdaten in zahlreichen Testfahrten mit unterschiedlichen Fahrern gesammelt und auf Datenservern gespeichert. Dabei werden diese in verschiedene Datenformate konvertiert und mit weiteren Datenquellen wie z.B. digitalen Karten verknüpft und in einer aggregierten Datentabelle zur Verfügung gestellt. Als zweiter Schritt dient die *Definition und Ableitung relevanter Testereignisse- und Dimensionen* als Basis für eine Fokussierung auf Fahrabschnitte mit hohem Erprobungsnutzen. Anhand dieser Ereignisse und Dimensionen folgt dann die *Evaluation*, in der Fahrabschnitte anhand der Expositionskenzahl bewertet werden. Die *Routenplanung* nutzt die bewerteten Straßenabschnitte und zusätzliche Planungsvorgaben, um neue Strecken zu generieren. Der Beitrag dieses ereignisorientierten Ansatzes wird schließlich anhand realer Fahrdaten validiert.

2 Related Work

Hintergründe zum Nutzen und zur Anwendung von Feldtests findet man im Projekt „Field Operational Test Support Action“ (FESTA) [2], FOT-NET². Im Bericht wird beschrieben, wie anhand von Feldtests der Nachweis der Wirksamkeit des Systems im realen Straßenverkehr gezeigt werden kann. Außerdem werden Hinweise zur Planung, Umsetzung und Auswertung von Feldtests gegeben. Die vorgestellten Methoden zur Datenanalyse liefern eine Beschreibung von Ereignissen, jedoch keine verwertbaren Ansätze zur Bewertung dieser im Sinne einer ereignisorientierten Felderprobung.

²www.fot-net.eu

Im Projekt PReVENT³ werden mehrere Aspekte von präventiven Sicherheitssystemen und deren Anwendung im Verkehr behandelt. Dazu gehört das Teilprojekt RESPONSE 3 [3] und MAPS & ADAS. RESPONSE 3 dient als Entwicklungsdokumentation für die Einführung von ADAS im Hinblick auf eine valide Umsetzung von neuen Entwicklungen bis hin zum Produktionsstart. Es werden Anleitungen zur Identifikation und Bewertung von Risiken abhängig der Frequenz und des Schweregrades möglicher Ereignisse gegeben. Außerdem wird gezeigt, wie der Entwicklungsprozess eines Systems, der in eine Design- und Produktionsphase unterteilt ist, in verschiedene Elemente abhängig von den Sicherheitsanforderungen und der Komplexität des Systems aufgeteilt werden kann.

Die Bewertung von Ereignissen bezüglich der Kritikalität orientiert sich dabei an Bewertungsgrößen wie dem ASIL (Automotive Safety Integrity Level) gemäß ISO 26262, in der Ereignisse gemäß des Schweregrades, Auftretenshäufigkeit und Beeinflussbarkeit bewertet werden. Wie diese Aspekte mit Umgebungsbedingungen korreliert werden, um relevante Testbereiche ableiten zu können, ist nicht erläutert.

Welche Möglichkeiten bestehen, um ein Fahrerassistenzsystem sowohl anhand eines Fahr-simulators als auch in realen Verkehrssituationen zu bewerten, wird in Kapitel 6 im Handbuch Fahrerassistenzsysteme [4] beschrieben. Es werden Anforderungen für einen Bewertungsprozess beschrieben sowie ein Ausblick mittels beispielhafter Anwendungen für ausgewählte Systeme gegeben. Die Überprüfung zielt dabei stark auf die Akzeptanz und Nutzbarkeit des Systems. Mögliche Unzulänglichkeiten des Systems auf Grund unvorhergesehener Fahrumstände wird nicht näher betrachtet.

Eine Bewertung von Feldteststudien hinsichtlich des Fahrverhaltens und Fahrmanövern ist in [5] beschrieben. Es werden verschiedene Fahrmanöver definiert und deren Häufigkeit und Kritikalität bewertet wie z.B. Anfahrvorgänge pro Stunde. Eine Datenmatrix, ähnlich dem Vorgehen bei epidemiologischen Studien, wird benutzt um die Anzahl von Ereignissen abhängig der Exposition auszuwerten. Durch die vorab Definition einiger weniger Fahrmanöver ist eine Abdeckung relevanter Testereignisse und vor allem deren Ursache zur Bestimmung relevanter Testdimensionen nicht gewährleistet.

3 Ereignisbasierte Erprobung und Potentiale

Das Ziel von Feldtests ist der Nachweis der Realitätstauglichkeit des Systems. Dabei sollen mögliche funktionale Unzulänglichkeiten des Systems aufgedeckt und behoben werden. Es ist zu beachten, dass Systeme von vornherein nur für bestimmte Situationen konzipiert wurden oder nur in bestimmten Geschwindigkeitsbereichen aktiv sind. Das motiviert den Ansatz, sich auf bestimmte Bereiche bei realen Straßenfahrten zu konzentrieren. Heutige Tests basieren hauptsächlich auf Vorgaben bezüglich der Laufleistung, die wiederum von der Komplexität oder Funktionserweiterung des Systems abhängen. Abbildung 2 stellt einen Vergleich der bisherigen Vorgehensweise einer möglichen Zukünftigen entgegen.

Für die heutige Laufleistungs-basierte Erprobung werden anhand von Erfahrungswerten und Expertenschätzungen Zielvorgaben gemacht, die bis zur Markteinführung erreicht werden sollen. Der Fortschritt der Erprobung kann dabei stets mittels Vergleich des Soll- und Ist-Wertes ermittelt werden, währenddessen die nötigen Verbesserungen am System vorgenommen werden. Bezüglich der Teststrecken existieren einige qualitative Vorga-

³<http://prevent.ertico.webhouse.net>

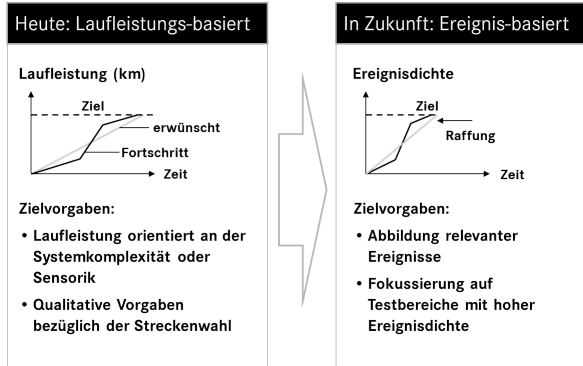


Abbildung 2: Vergleich der heutigen Erprobung mit einer möglichen Zukünftigen.

ben zur Streckenauswahl, wie Überland- oder Autobahnfahrten. Der zukünftige Ereignis-basierte Ansatz orientiert sich dagegen an Ereignissen die einen starken Nutzen im Sinne der Erprobung liefern. Diese können zunächst einmal grob in drei Kategorien unterteilt werden:

- Systemreaktionen (z.B. (nicht-) plausible Systemeingriffe)
- Fahrer initiierte Ereignisse (z.B. Überholvorgänge, Bremsengriffe)
- Umgebungsbedingungen(z.B. Kreuzungsdurchfahrten, Tunnelleinfahrten)

Entsprechend dem zu testenden System, ist eine entsprechende Auswahl und vorab Definition von Ereignissen notwendig, die dann mit den Umgebungssituation korreliert werden. Der vorgeschlagene Prozess zielt hier auf eine Raffung bezüglich der benötigten Kilometer für die gleiche Anzahl von Ereignissen ab. Um den Nutzen für die Erprobung fassen zu können werden Kennzahlen benötigt. Die aufgeworfene Frage dabei ist: Was ist der Nutzen von 100 km oder 1 h in der Erprobung für das abzusichernde System?

4 Exposition

Das Wort Exposition wird oft in Zusammenhang mit epidemiologischen Studien verwendet und beschreibt die Aussetzung eines Objektes oder Menschen gegenüber gewissen Einflüssen oder Gefahren aus der Umwelt. In Bezug auf ein Fahrerassistenzsystem interessieren die Einflüsse oder Situationen, auf die das System überhaupt reagiert und besonders wo es ein unerwünschtes Verhalten aufweist. Eine Strecke die das System stark exponiert, also gegenüber vielen Einflüssen aussetzt, ist damit im Sinne der Erprobung wertvoll. Die aufgeworfene Frage bezüglich dem Nutzen eine Strecke umfasst im Sinne der Exposition nun zwei relevante Punkte: wie häufig tritt ein testrelevantes Ereignis auf und wie relevant im Sinne der Erprobung ist das Ereignis? Daraus abgeleitet lässt sich die Exposition bestimmen mit

$$Exp(s) = \sum_e Crit(e) \cdot P(e, s), \tag{1}$$

mit der Kritikalität $Crit(e)$ eines Ereignisses e der Auftretenswahrscheinlichkeit des Ereignisses $P(e, s)$ und des befahrenen Straßenabschnittes s . Um die Auftretenswahrscheinlichkeit zu berechnen können in einem iterativen Prozess die bestehenden Messdaten verwendet werden. Liegen noch keine Daten vor, kann eine Schätzung auf Basis von Expertenwissen angewandt werden. Ein Maß für die Häufigkeit eines Ereignisses ist z.B. die Warnrate, also Systemwarnungen pro Stunde SW/h . Summiert über alle Streckenabschnitte s lässt sich schließlich die Exposition Exp für eine zusammenhängend Route als Teststrecke bestimmen. Die Bewertung der Kritikalität von Ereignissen lässt sich an bestehenden Entwicklungsdokumenten wie dem Response Code of Practice [3] orientieren. Die darin beschriebene Gefahren und Risiko Analyse nutzt unter anderem den ASIL gemäß ISO 26262 [1] für eine Einstufung möglicher Gefahrenpotentiale. Aus der Beherrschbarkeit und Schwere möglicher Situationen laut ASIL lässt sich die Kritikalität $Crit(e)$ von Ereignissen ableiten. Die Auftretenswahrscheinlichkeit $P(e, s)$ wird in Gleichung (1) direkt anhand bestehender Fahrdaten berechnet.

5 Bestimmung relevanter Testdimensionen

Der Erprobungsraum setzt sich aus allen theoretischen Test-Dimensionen wie Fahrumgebung, Fahrweise und Systemparametern zusammen. Dabei stehen sowohl hunderte von Messkanälen aus Fahrzeugdaten als auch Infrastrukturdaten aus digitalen Karten zur Verfügung. In der Felderprobung spielen sowohl die Erkennung von Falscheingriffen und deren Behebung als auch Häufigkeitsraten von gewünschten Eingriffen eine Rolle. Generell sind Situationen von Interesse, die das System herausfordern und zu Systemreaktionen führen. Aus Entwicklungsanleitungen wie dem „Response Code of Practice“ [3] lassen sich Strukturen zur Beschreibung relevanter Verkehrssituationen ableiten. Abbildung 3 zeigt einen Ausschnitt daraus.

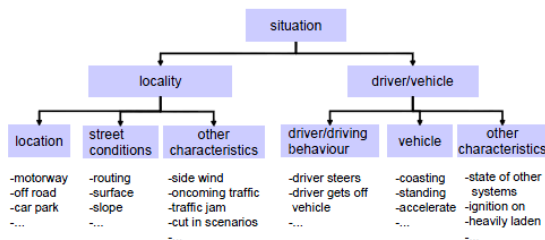


Abbildung 3: Situationsbaum zur Ableitung relevanter Verkehrssituationen.

Die Auswahl von Dimensionen lässt sich dabei an Situationen bezüglich Umgebungsbedingungen wie Infrastruktur oder Straßenbeschaffenheit, oder auf der anderen Seite Ei-

genschaften die dem Fahrer/Fahrzeug anhaften, orientieren. In dieser Arbeit beschränken wir uns auf Dimensionen die sich der Infrastruktur zuordnen lassen und aus digitalen Karten entnommen werden können. Die Verknüpfung von Kartendaten und Fahrzeugdaten erfolgt über das GPS-Signal mittels eines sogenannten „Map-Matching“ Prozesses. Dabei wird jedem Straßensegment (Link) eine eindeutige Kennzahl (ID) zugeordnet. Anhand dieser Link-ID lassen sich die entsprechenden Kartenattribute aus digitalen Navigationskarten nachschlagen.

Für eine Optimierung des Absicherungsaufwandes wird ähnlich dem Vorgehen bei der statistischen Versuchsplanung „Design of Experiment“ (DoE) [8] eine Fokussierung auf relevante Testbereiche und -Dimensionen vorgeschlagen. Eine Erprobung nach der „Brute-Force-Methode“, die sämtliche Kombinationen von Umständen umfasst, ist schon alleine aus wirtschaftlichen Gesichtspunkten nicht realisierbar. In Abbildung 4 ist dieser Gedanke graphisch dargestellt.

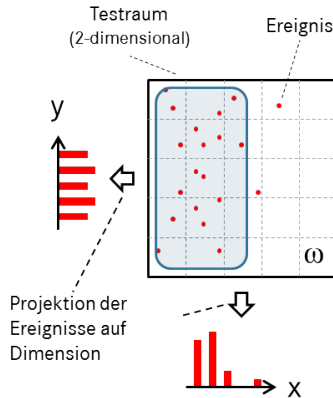


Abbildung 4: Ereignishäufigkeit im Erprobungsraum und Projektion auf geeignete Testdimensionen.

Dabei wird der Erprobungsraum beispielhaft in zwei Dimensionen aufgeteilt. Die darin verteilten Ereignisse zeigen, dass entlang der y -Achse die Ereignisse mehr oder weniger gleich verteilt auftreten, was auf eine weniger geeignete Dimension zur Fokussierung schließen lässt. Im Gegensatz dazu kommen entlang der x -Achse die Ereignisse stark verteilt vor und beschränken sich auf gewisse Bereiche innerhalb der Dimensionen. Für eine automatische Bestimmung relevanter Dimensionen wird das normierte Kontingenzmaß „Cramér's V “ [11] berechnet, welches die Stärke des Zusammenhangs zwischen erprobungsrelevanten Ereignissen und Dimensionen liefert. Für die Festlegung der Bereiche innerhalb der Dimensionen müssen gewisse Annahmen getroffen werden. Je nach Dimension macht eine Festlegung von bis zu sechs oder sieben Intervallen Sinn. Da die Korrelation für eine gute Vergleichbarkeit unabhängig von der Anzahl der Bereiche und der Menge der Daten sein soll, wird das normierte Kontingenzmaß Cramér's V [11] verwendet, welches berechnet

wird mit

$$V(E, D) = \sqrt{\frac{\chi^2(E, D)}{n \cdot (\min[r, c] - 1)}}. \quad (2)$$

Dabei gilt $V \in [0, 1]$, wobei 0 keine und 1 maximale Korrelation bedeutet. χ^2 ist die Chi-Quadrat-Verteilung [11], r die Anzahl von Zeilen, c die Anzahl von Spalten, n die Anzahl an Messwerten, E das untersuchte Ereignis und D die Testdimension. Durch ein Ranking des Zusammenhangsmaßes für alle verfügbaren Dimensionen, lässt sich schließlich eine Auswahl von Dimensionen treffen, die für eine Fokussierung geeignet sind.

6 Routenplanung und Validierung

Für eine Generierung von optimalen Routen im Sinne einer effizienten Felderprobung werden zur Planung alle Straßenlinks mit der Exposition als Zielgröße in einer Datenbank abgelegt. Jeder Link stellt damit eine gewichtete Kante in einem Graph dar. Der Quotient aus Nutzen (Exposition) und Kosten (Fahrdauer, -Länge) gilt es zu maximieren. Da dies keine monotone Zielfunktion darstellt, scheitern klassische Verfahren wie Dijkstra oder Bellmann-Ford. Außerdem sollen Vorgaben zur Dauer und Startpunkt gleich Zielpunkt gemacht werden können. Das daher angewandte genetische Verfahren findet aus einer Population von Routen durch Kreuzung und Mutation eine erweiterte Population. Daraus werden dann Abschnitte ausgewählt und der Vorgang solange wiederholt, bis es zu einer Konvergenz kommt. Der genetische Planungsalgorithmus der als Zielgröße das zu maximierende Kosten/Nutzen Verhältnis hat, generiert Routen mit einer hohen Ereignisdichte. Abbildung 5 zeigt eine geplante Route mit Start und Ziel vor dem Werk in Sindelfingen, einer vorgegeben Länge von 80 km und als Zielfunktion das zu maximierende Nutzen/Kosten Verhältnis aus Streckenlänge und Exposition. Dabei sind wiederholt befahrene Links auszuschließen.



Abbildung 5: Generierte Route mit hoher Exposition zur Validierung.

Der prognostizierte Wert dieser Route könnte durch einen Vergleich mit den real vorkommenden Ereignissen auf den bereits in der Erprobung befahrenen Fahrabschnitten nachgewiesen werden. Der real gemessene Wert für die Warnrate auf dieser Route liegt dabei bei $1,94 \text{ SW/h}$ und damit über dem fast fünffachen des durchschnittlichen Wertes von $0,41 \text{ SW/h}$ aus allen Fahrdaten. Neben der Bewertung einzelner Routen, die einer gewissen Streuung unterliegen, ist eine Validierung anhand größerer Datensätze von Interesse. Bei der Bewertung von Straßenlinks anhand der Exposition bleibt die Frage nach der Substanz dieser Prognose offen. Da jedoch sehr viele Messdaten aus Erprobungen vorhanden sind, lässt sich die Prognose mit den tatsächlich in den Messdaten auftretenden Ereignissen verknüpfen und somit validieren. Anhand der Kartenattribute wird jedem Link ein Expositionswert zugewiesen. Damit lässt sich für jeden in den Erprobungsdaten vorkommenden Link der Expositionswert mit den tatsächlich aufgetretenen Ereignissen vergleichen. Der hier verwendete Datensatz umfasst mehrere hunderttausend Kilometer an Messdaten und über dreitausend testrelevante Ereignisse. Für die Validierung wird der Datensatz dabei in zwei gleichgroße, disjunkte Datenmengen aufgeteilt, die Trainingsmenge α und die Testmenge β . Aus der Trainingsmenge α wird die Prognose für die Exposition anhand der Kartenattribute erstellt. Über die Verknüpfung anhand der Kartenattribute können alle Links in der Testmenge β bewertet werden. Schließlich kann die Prognose mit der Evaluation verglichen werden, wie Abbildung 6 zeigt.

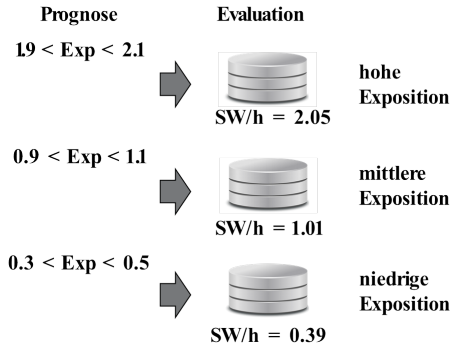


Abbildung 6: Validierung der Prognose anhand realer Fahrdaten in drei Bereichen.

Zur Validierung werden drei Bereiche verglichen. Im Bereich mit hoher Exposition werden alle Straßenabschnitte in der Testmenge β ausgewählt, die laut Prognose eine Exposition zwischen 1.9 und 2.1 aufweisen und mit der tatsächlichen Anzahl von Systemeingriffen pro Stunde SW/h verglichen. Die Evaluation ergibt einen tatsächlichen Wert von 2.05 SW/h , was einer sehr guten Prognose entspricht. Das gleiche Vorgehen ist in der Abbildung für eine mittlere und niedrige Exposition gezeigt. Insgesamt lässt sich somit auch anhand großer Datenmengen der Nutzen des Vorgehens zeigen sowie eine verlässliche Aussage der Prognose nachweisen.

7 Zusammenfassung und Ausblick

Auf Grund der steigenden Zahl und Vielfalt von Fahrerassistenzsystemen liegt der Bedarf auf der Hand, die Erprobung möglichst effizient und zielgerichtet zu gestalten. Dies umfasst zum einen Messwerkzeuge, die Auskunft über den Erkenntnisgewinn von Fahrleistungen geben, zum anderen einen Prozess, der ausgehend von dem zu testenden System relevante Dimensionen für die Planung von Straßenfahrten ableiten kann. Mittels ausgewählter Dimensionen lassen sich Fahrabschnitte anhand quantitativer Kennzahlen wie Exposition bewerten und für eine Erprobungsplanung nutzen. Die generierten Routen liefern eine hohe Dichte von relevanten Ereignissen, was anhand vorhandener Fahrdaten validiert werden konnte. Auf einer neu generierten Teststrecke konnte dabei ein Raffungsfaktor von über vier erreicht werden. Neben quantitativen Bewertungskennzahlen besteht der Wunsch nach Testabschlusskriterien, welche in weiteren Arbeiten untersucht werden können.

Literatur

- [1] ISO 26262: Road Vehicles - Functional Safety. International Organization for Standardization, 2012.
- [2] FESTA Consortium. Field operational test support Action (FESTA) Handbook, 2008.
- [3] ERTICO-ITS Europe: RESPONSE3, Code of Practice for the Design and Evaluation of ADAS. Information Society 2009, V5.0.
- [4] J. Breuer: Bewertungsverfahren von Fahrerassistenzsystemen. Handbuch Fahrerassistenzsysteme, Vieweg+Teubner 2009.
- [5] V. Hargutt, N. Schebitz, D. Scheunert, A. Seeck: Naturalistic Driving Studies in Germany: analysis methods and current results of a pilot study. Technical Congress, VDA, p. 297-311.
- [6] H. Winner, A. Weitzel: Quo vadis, FAS? Handbuch Fahrerassistenzsysteme, Vieweg+Teubner, Wiesbaden, 2012, p. 658-667.
- [7] S. Tattersall, U. Petersen, J. Breuer: Ein Messdatenmanagementsystem für die Feldabsicherung von neuen Fahrerassistenzsystemen. 28. VDI/VW Gemeinschaftstagung Fahrerassistenz und Integrierte Sicherheit, 2012.
- [8] F.-L. Krause, Chr. Kind, C. Biantoro: The Application of a Statistical Design of Experiment for Quantitative Analysis and Optimisation of Development Processes. The Future of Product Development. Springer Berlin Heidelberg, 2007, p. 493-502.
- [9] P. Glauner, A. Blumenstock, M. Haucis, U. Petersen, S. Tattersall: Ansätze zur effizienteren Optimierung von Assistenzsystemen. Kolloquium Mensch und Fahrzeug, Darmstadt 2013.
- [10] R. Bonita, R. Beaglehole, T. Kjellstrom: Basic epidemiology, 2nd edition, WHO 2006.

- [11] W. Kohn: Statistik, Datenanalysis und Wahrscheinlichkeitsrechnung. Springer, 2005, p. 113-120.

Systematische Auswertung von Testfällen für Fahrfunktionen im modularen virtuellen Testbaukasten

Fabian Scholdt*, Bernd Lichte†, Markus Maurer‡ und Stephan Scholz§

Zusammenfassung: In modernen Kraftfahrzeugen findet eine immer stärkere Demokratisierung der Fahrerassistenzsysteme in der mittleren und unteren Fahrzeugklasse statt. Für einen geeigneten Serieneinsatz benötigen die Systeme eine hohe Testabdeckung und einen hohen Reifegrad. Eine Möglichkeit diese hohe Testabdeckung zu generieren, stellt der in diesem Beitrag vorgestellte modulare virtuelle Testbaukasten dar. Dieser verfolgt das Ziel, die Anzahl der benötigten Testfälle für eine gleichbleibende Testabdeckung durch geeignete Variationen von Szenarien zu minimieren. Im Beitrag werden der modulare virtuelle Testbaukasten und verschiedene Auswertungsverfahren für kombinatorische Testfälle vorgestellt. Des Weiteren werden die Anwendung des Testbaukastens sowie die Testfallauswertung anhand der Fallstudie eines Baustellenassistenten skizziert.

Schlüsselwörter: modularer virtueller Testbaukasten, kombinatorisches Testen, Design of Experiments, Testauswertung

1 Einleitung

1.1 Motivation

Viele Fahrerassistenzsysteme haben in den letzten Jahren Serienreife erlangt und sind mittlerweile fester Bestandteil der oberen und mittleren Fahrzeugklasse [20]. Als Beispielsystem ist das ACC (Adaptive Cruise Control) zu nennen, welches seit der Einführung einige Entwicklungsstufen durchlaufen hat (Full Speed Range ACC). Ein weiteres, in der Serie etabliertes, System ist der sogenannte Spurhalteassistent (lane keeping assist system), der den Fahrer bei der Querführung unterstützt. Insgesamt ist zu beobachten, dass die entwickelten Fahrfunktionen in steigendem Umfang in die Längs- und Querführung des Fahrzeugs eingreifen. Aktuelle Forschungsprojekte wie UR:BAN [15] zeigen, dass die Fahrerassistenzsysteme in Zukunft nicht nur auf Autobahnen und Schnellstraßen, sondern auch gezielt für städtische Szenarien entwickelt werden. Diese Szenarien stellen erhöhte Anforderungen an die Fahrfunktion, welche sich durch dichten Verkehr, unstrukturierte

*Fabian Scholdt ist wissenschaftlicher Mitarbeiter am Institut für Regelungstechnik der TU Braunschweig, Hans-Sommer-Straße 66, 38106 Braunschweig (e-mail: scholdt@ifr.ing.tu-bs.de).

†Bernd Lichte ist Akademischer Rat am Institut für Regelungstechnik der TU Braunschweig, Hans-Sommer-Straße 66, 38106 Braunschweig (e-mail: lichte@ifr.ing.tu-bs.de).

‡Markus Maurer ist Professor am Institut für Regelungstechnik der TU Braunschweig, Hans-Sommer-Straße 66, 38106 Braunschweig (e-mail: maurer@ifr.ing.tu-bs.de).

§Stephan Scholz ist Projektleiter im Unternehmen Volkswagen AG, Berliner Ring 2, 38440 Wolfsburg (e-mail: stephan.scholz1@volkswagen.de).

Randbebauungen sowie komplexe Verkehrssituationen insbesondere in Kreuzungsbereichen auszeichnen.

Ein sicherer Einsatz solcher Systeme im öffentlichen Straßenverkehr kann nur gewährleistet werden, wenn vor einer Serieneinführung ein technisch erforderlicher Reifegrad und eine ausreichende Testtiefe nachgewiesen wurde. Die Testtiefe ist hierbei üblicherweise direkt mit einer entsprechend hohen Anzahl von Einzeltests verbunden. Die Anzahl der nötigen Einzeltests steigt wiederum mit der Komplexität der Anwendungsszenarien. Welchen Umfang eine Absicherung eines auf der Autobahn automatisiert fahrenden Fahrzeugs annehmen kann, hat H.Winner skizziert [20]. Dieser schätzt den Testaufwand für die Absicherung einer Funktionsversion auf ca. 50 Millionen Testkilometer, was Kosten in Höhe von 50 Millionen Euro verursachen würde. Bei jeder Änderung der Funktionsversion (Softwarestand) müssten diese Tests für eine Absicherung zumindest teilweise wiederholt werden. Die Abschätzung zeigt, wie aufwändig die Generierung einer ausreichenden Testtiefe für die funktionale Absicherung ist. Für Systeme, die auf innerstädtische Szenarien fokussieren, wird die Absicherung wesentlich aufwendiger, da die Zahl der Einflussparameter sowie deren Variationen im Vergleich zu Autobahnen oder Schnellstraßen und der damit verbundene Testaufwand immens ansteigt. Abbildung 1 stellt gemäß der nach Gasser [4] eingeführten Stufen der automatisierten Fahrfunktionen eine vereinfachte Abschätzung des damit verbundenen Testaufwands dar.

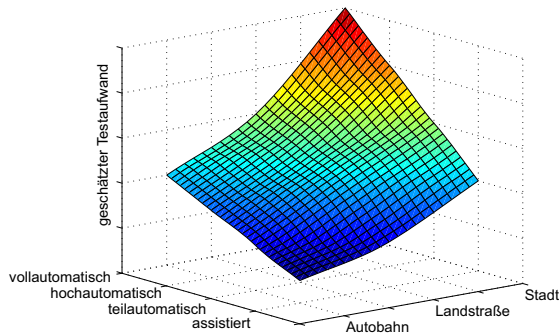


Abbildung 1: Vereinfachte Schätzung des Testaufwands auf verschiedenen Automatisierungsstufen

1.2 Bisherige Arbeiten

Die Absicherung von automatisierten Fahrfunktionen stellt eine große Herausforderung dar. Aktuelle Systeme werden über eine fehlerfreie Funktion über eine bestimmte Anzahl von Testkilometern abgesichert [20]. Ein Beispiel für den Umfang der Testkilometer wurde in der Einleitung gezeigt. Es müssen somit geeignete Methoden gefunden werden, um die Systeme nachhaltig mit einem umsetzbaren und vertretbaren Umfang abzusichern.

Alternative Ansätze zeigen, dass die Simulation einen entscheidenden Teil zur Optimierung der Absicherung beitragen kann. Der Beitrag von ERTICO-ITS Europe [14] stellt dar, in welchen Phasen der Entwicklung einer Funktion die Simulation als Testwerkzeug

verwendet werden kann. Nentwig et. al. [10, 11, 12] zeigen, dass die Ergebnisse von Tests in einer Software-in-the-Loop Simulation und Realtests vergleichbar sind und die Simulation für den Test von Fahrfunktionen verwendet werden kann. Im Beitrag von Tideman [19] wird der Test eines automatisch abblendenden Lichtsystems in der Simulation gezeigt.

Der Beitrag [13] skizziert, wie die Simulation bei dem Test von kamerabasierten Fahrfunktionen unterstützen kann. Dies wird anhand eines Spurhalteassistenten demonstriert. Es werden einige exemplarische Tests mit fünf Einflussparametern durchgeführt. Jedoch werden keine kombinatorischen Verfahren zur Generierung einer definierten Testabdeckung verwendet. Des Weiteren wird keine systematische Testauswertung durchgeführt.

Ein alternativer Ansatz für die Verwendung der Simulation beim Test von Fahrfunktionen wird in den Beiträgen [2, 5] beschrieben. Beim Vehicle-Hardware-in-the-Loop (V-E-H-I-L) befindet sich das zu testende Fahrzeug auf einem Rollenprüfstand. Die Zustandsdaten, wie zum Beispiel der Lenkwinkel und die Geschwindigkeit des Fahrzeugs, werden der Simulation übergeben. Die anderen Verkehrsteilnehmer der Simulation werden durch fahrende Plattformen realisiert, die sich relativ zum Fahrzeug auf dem Prüfstand bewegen. Beim Vehicle-in-the-Loop (V-I-L) bewegt sich das Fahrzeug auf einem Prüfgelände. Die Bewegungen werden simultan in die Simulation übergeben. Die anderen Verkehrsteilnehmer werden dem Fahrer durch eine Visualisierungsbrille dargestellt. Die beiden Ansätzen zeigen, wie die Simulation für realitätsnähere Tests verwendet werden kann. Aus diesem Grund wird in diesem Beitrag ein V-I-L ähnlicher Ansatz für die Testdurchführung verwendet.

Die Methoden des „Design of Experiments“ und der Taguchis Qualitätsverlustfunktionen werden in vielen Bereichen zur Optimierung von Prozessen verwendet. Der Beitrag von Khorramshahgol [6] zeigt, wie sich die Qualitätsverlustfunktionen im Bereich der Qualitätssicherung anwenden lassen. Es werden Experimente auf verschiedenen Faktorstufen durchgeführt und optimale Faktorstufen durch Qualitätsverlustfunktionen identifiziert. Bei den Tests von Fahrfunktionen sind ähnliche Problemstellungen vorhanden. Es müssen ebenfalls die Einflüsse von verschiedenen Parametern bestimmt werden. Aus diesem Grund wurden das Vorgehen und die Bewertungsmethoden in diesem Beitrag durch die Qualitätsverlustfunktionen und das Design of Experiments motiviert.

1.3 Aufbau des Beitrags

Im folgenden Abschnitt wird der Aufbau des modularen virtuellen Testbaukastens skizziert. Dabei werden die Funktionen der einzelnen Ebenen vorgestellt. Im dritten Abschnitt dieses Beitrags wird speziell auf die Testfallauswertung im modularen virtuellen Testbaukasten eingegangen. Es werden verschiedene Bewertungsverfahren aufgezeigt. Im vierten Abschnitt wird die Fallstudie eines Baustellenassistenten vorgestellt und erste Ergebnisse der Testauswertung skizziert. Bei dem System handelt es sich um ein Assistenzsystem, welches dem Fahrer bei der Durchfahrt von Baustellen assistiert. Abschließend werden die Ergebnisse zusammengefasst und ein Ausblick über weitere Forschungsarbeiten gegeben.

2 Modularer Testbaukasten

In diesem Abschnitt wird der Ansatz des modularen virtuellen Testbaukastens vorgestellt. Der Testbaukasten verfolgt das Ziel, die nötigen Tests für die Absicherung von

Fahrerassistenzsystemen bei einer gleichbleibenden Testabdeckung in Grenzen zu halten. Das Konzept, welches im Beitrag [16] vorgestellt wurde, zeigt einen Ansatz zur effizienten systematischen Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen. Dieser Ansatz wird aufgegriffen und zu einem modularen Testbaukasten für das Testen von Fahrfunktionen in Simulationen weiterentwickelt. In welchem Rahmen die Tests in der Simulation für eine Absicherung von Fahrfunktionen eingesetzt werden können, zeigt der Beitrag [1]. Der Testbaukasten realisiert den Anspruch, die Absicherung einer Vielzahl von Fahrfunktionen, d.h. vom assistierten bis zum vollautomatisierten Fahren, abzudecken. Abbildung 2 zeigt das Konzept des modularen Testbaukastens.

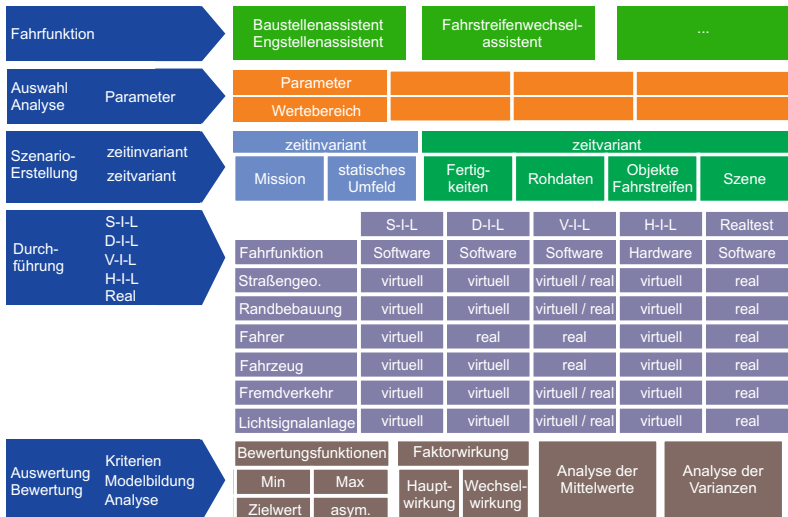


Abbildung 2: Konzept des modularer Testbaukastens. Abkürzungen: S-I-L: Software in the loop, D-I-L: Driver in the loop, V-I-L: Vehicle in the Loop, H-I-L: Hardware in the loop

Auf der ersten Ebene wird die zu testende Fahrfunktion festgelegt. Darauf folgt eine Auswahl und Analyse von relevanten Einflussparametern, die für die Fahrfunktion von Bedeutung sind. Es werden geeignete Wertebereiche der Parameter identifiziert und Diskretisierungsstufen für stetige Größen festgelegt. Auf der dritten Ebene werden die Szenarien für den Test der Fahrfunktion generiert. Dabei werden sowohl zeitinvariante als auch zeitvariante Größen definiert. Bei den zeitinvarianten Größen handelt es sich beispielsweise um das statische Umfeld und die Mission der Fahrfunktion. Die zeitvarianten Größen können Fertigkeiten, Sensorrohdaten, Objekt- und Fahrstreifenhypothesen oder aber komplette Szenen sein. Eine Szene wird hierbei als eine räumliche und zeitliche Anordnung von statischen und dynamischen Objekten aus der Sicht eines Betrachters interpretiert [9]. Abbildung 3 zeigt einige statische und dynamische Bestandteile einer Szene. Das Szenario wird durch kombinatorische Verfahren parametrisiert. Dabei werden

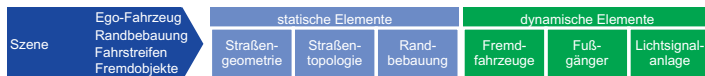


Abbildung 3: Bestandteile einer Szene nach Maurer [9]

die Stufen der einzelnen Einflussparameter so variiert, dass beispielsweise jedes Paar von Stufen der Einflussparameter in mindestens einem Testfall vorhanden ist [8]. Der modulare Testbaukasten sieht eine Durchführung der Testfälle auf verschiedenen Testebenen vor. Es wird unterschieden zwischen Tests auf den Ebenen Software-in-the-Loop (S-I-L), Driver-in-the-Loop (D-I-L), Vehicle-in-the-Loop (V-I-L), Hardware-in-the-Loop (H-I-L) und Realtests. Auf jeder Testebene werden verschiedene Simulationen durchgeführt. Eine Übersicht, welche Elemente auf den verschiedenen Ebenen simuliert werden, ist in Abbildung 2 dargestellt. Je nach Fahrfunktion und Test ist eine geeignete Testebene für die Testdurchführung auszuwählen. Abschließend werden die Testfälle anhand von eindeutigen Bewertungskriterien systematisch ausgewertet. Ziel ist die Analyse und Bewertung der Ergebnisse in Bezug zu der Parametrisierung der verschiedenen Testfälle. Dabei werden Haupt- und Wechselwirkungen der einzelnen Faktoren identifiziert und verschiedene Funktionen zur Bewertung verwendet. Eine Beschreibung einiger Elemente der systematischen Testauswertung des modularen virtuellen Testbaukastens wird im nächsten Abschnitt vorgestellt.

3 Systematische Testfallauswertung

Die systematische Testauswertung wird anhand von verschiedenen Bewertungskriterien für die Fahrfunktionen durchgeführt. Ziel der Auswertung ist es, die Effekte, d.h. die mittlere Veränderung der Qualitätsmerkmale in Folge von gezielten Stufenvariationen, zu bestimmen. Dabei wird zwischen Haupt- und Wechseleffekten unterschieden. Um die Wirkungen der einzelnen Stufen der Einflussparameter zu identifizieren, kann eine Analyse der Mittelwerte (ANOM) durchgeführt werden [7, 17]. Zur besseren Vergleichbarkeit wird eine Normierung der unterschiedlichen Kriterien mit Hilfe von sogenannten Qualitätsverlustfunktionen durchgeführt. Die für die Fallstudie verwendeten Verfahren Qualitätsverlustfunktion und ANOM werden in den folgenden Abschnitten vorgestellt.

3.1 Qualitätsverlustfunktionen

Die Qualitätsverlustfunktionen haben das Ziel, die unterschiedlichen Bewertungskriterien zur besseren Vergleichbarkeit zu normieren. Die Funktion beschreibt den Qualitätsverlust, welchen die Fahrfunktion verzeichnet, sobald der gemessene Wert y vom Sollwert m abweicht. Der maximal erlaubte Qualitätsverlust wird als A_0 bezeichnet und wird bei $m \pm \Delta_0$ erreicht. Sobald der Wert A_0 überschritten ist, wird der Test als nicht bestanden bewertet. Es wird zwischen Zielwert-, Maximierungs- und Minimierungsfunktion sowie asymmetrischer Verlustfunktion unterschieden [18]. Für jedes Kriterium wird eine geeignete Verlustfunktion, die maximal erlaubte Abweichung vom Sollwert und deren maximaler Qualitätsverlust gewählt.

Tabelle 1: Qualitätsverlustfunktionen

	Zielwert-Qualitätsmerkmal	Minimierungsmerkmal
Formel	$L(y) = k(y - m)^2$ $= \frac{A_0}{\Delta_0^2}(y - m)^2$	$L(y) = ky^2$ $= \frac{A_0}{\Delta_0^2}y^2$
	Asymmetrische Verlustfunktion	Maximierungsmerkmal
Formel	$L(y) = \begin{cases} \frac{A_0}{\Delta_0^2}(y - m)^2, & \text{für } y > m \\ \frac{A_1}{\Delta_1^2}(y - m)^2, & \text{für } y \leq m \end{cases}$	$L(y) = \frac{k}{y^2}$ $= \frac{A_0 \Delta_0^2}{y^2}$

3.2 Mittelwertanalyse (ANOM)

Die Analyse der Mittelwerte dient zur qualitativen Abschätzung der Haupteffekte von den unterschiedlichen Faktorstufen der Einflussparameter auf die verschiedenen Bewertungskriterien. Zur Bestimmung der Haupteffekte wird zunächst für jedes Bewertungskriterium der Mittelwert \bar{Y}_{ij} für jede Stufe j eines Faktors i über die Messwerte y_{ijk} gebildet [7]. Weiterhin wird der Mittelwert des Bewertungskriteriums über alle Testfälle gebildet:

$$\bar{Y}_{ij} = \frac{\sum_{k=1}^{n_j} y_{ijk}}{n_j} \quad \forall \quad j = 1 \dots J \quad \bar{Y}_i = \frac{\sum_{j=1}^J \bar{Y}_{ij}}{J}.$$

Die Hauptwirkung einer Faktorstufe \bar{Y}_{ij} wird durch die Abweichung vom Gesamtmittelwert \bar{Y}_i geschätzt.

4 Fallstudie eines Baustellenassistenten

Die Anwendung des Testbaukastens und speziell die Testfallauswertung wird im Folgenden anhand der Fahrfunktion eines prototypisch umgesetzten Forschungsstand eines Baustellenassistenten in einer statischen Fallstudie exemplarisch mit einer Auswahl von Einflussparametern demonstriert. Statisch bedeutet in diesem Fall, dass keine Fremdfahrzeuge in den Szenarien auftreten. Das System hat die Aufgabe, dem Fahrer bei der Querführung während der Durchfahrt von Baustellen zu assistieren. Es müssen verschiedene Fahrstreifenverschwenkungen durchfahren werden, die durch eine definierte Länge und Fahrstreifenablage beschrieben werden und durch eine Randbebauung begrenzt sind. Das Verschwenkungsmaß ergibt sich aus dem Verhältnis von Verschwenkungsablage zu Verschwenkungslänge. Tabelle 2 fasst die Einflussparameter zusammen, die während der Analyse identifiziert wurden. Als Stufen wurden geeignete Werte aus den geltenden Normen gewählt [3]. Für eine vollständige Testabdeckung müssten sämtliche Kombinationen der einzelnen Stufen der verschiedenen Einflussparameter getestet werden, was zu 2160 Testfällen führen würde. Zur Reduzierung der Testfälle und für die Erstellung des Testplans wurde in dieser Fallstudie der kombinatorische Ansatz IPO-G aus dem Beitrag von Lei et. al. [8] verwendet. Da nur noch Paarkombinationen der Parameterstufen getestet werden, konnte die Anzahl der Tests auf 36 Fälle reduziert werden. Tabelle 3 zeigt einen Ausschnitt des reduzierten Testplans mit Paarkombinationen der Faktorstufen. Für die

Tabelle 2: Übersicht der Einflussparameter eines statischen Baustellenszenarios

Einflussparameter	Stufen					
	3.5	7.0	10.5	14.0		
Verschwenkungsablage [m]	3.5	7.0	10.5	14.0		
Verschwenkungsmaß	1 : 5	1 : 7	1 : 10	1 : 15	1 : 20	1 : 25
Randbebauung	Leitbake	Leitkegel	Leitwand			
Abstand der Randbebauung [m]	5.0	7.0	9.0	11.0	13.0	
Behelfsfahrstreifenbreite [m]	2.5	2.75	3.0	3.25	3.5	3.75

Durchführung der Tests wurde ein V-I-L ähnlicher Ansatz gewählt [2]. Es wurde mit einem realen Fahrzeug durch virtuelle Baustellen gefahren. Die Elemente der Baustelle wurden dem Fahrer auf einem Bildschirm angezeigt. Die Informationen über die virtuelle Randbebauung wurden durch Sensormodelle an die Fahrfunktion übergeben. Zur Vergleichbarkeit der einzelnen Testfälle wurden die Baustellen mit einer konstanten Geschwindigkeit von $60 \frac{km}{h}$ durchfahren. Der Fahrer hatte während der Durchfahrt die Hände nicht am Lenkrad, sodass nur die Fahrfunktion einen Einfluss auf die Querregelung des Fahrzeugs hatte. Tabelle 4 zeigt die gewählten Bewertungskriterien und deren Qualitätsverlustfunktion für die Fallstudie. Ziel der Auswertung ist es, die Faktorstufen zu identifizieren, die einen signifikanten Einfluss auf die Qualitätsmerkmale haben.

Tabelle 3: Reduzierter Testplan, *-don't care Werte

Testfall	Verschwenkungsmaß	Verschwenkungsablage	Art der Randbebauung	Abstand der Randbebauung	Fahrstreifenbreite
1	1:5	3.5	Leitbake	5	2.5
2	1:5	7	Leitwand	7	2.75
3	1:5	10.5	Leitkegel	9	3.0
...
36	*	1:25	*	*	3.75

Tabelle 4: Bewertungskriterien für die Auswertung der Tests. Abkürzungen: B: Breite, Fs: Fahrstreifen, Fzg: Fahrzeug

Bewertungskriterium	Zielwert m	erlaubte Abweichung Δ_0	Qualitätsverlust A_0	Qualitätsverlustfunktion
Randbebauung min. Abstand [m]	–	0.03	100	max
Fahrstreifen max. Ablage [m]	0	$(B_{Fs} - B_{Fz})/2$	100	Zielwert
Varianz der Ablage [m ²]	0	1.0	100	min
Querbeschleunigung maximal [m/s ²]	0	3	50	Zielwert

Mit Hilfe der Qualitätsverlustfunktionen wird der jeweilige Qualitätsverlust für die einzelnen Bewertungskriterien berechnet. Die Ergebnisse für das Bewertungskriterium minimaler Abstand ist in Abbildung 4 dargestellt. Es wurde der mittlere Qualitätsverlust der einzelnen Stufen der Einflussfaktoren Verschwenkungsmaß und Fahrstreifenbreite aufgetragen. Abbildung 5 zeigt die Qualitätsverluste, welche durch die maximale Ablage im eigenen Fahrstreifen hervorgerufen wird. Diese wurden ebenfalls mit dem mittleren Qualitätsverlust auf den Stufen der Bewertungskriterien Verschwenkungsablage und Abstand der Randbebauung bewertet. Abbildung 4 zeigt, dass es eine hohe negative Korrelati-

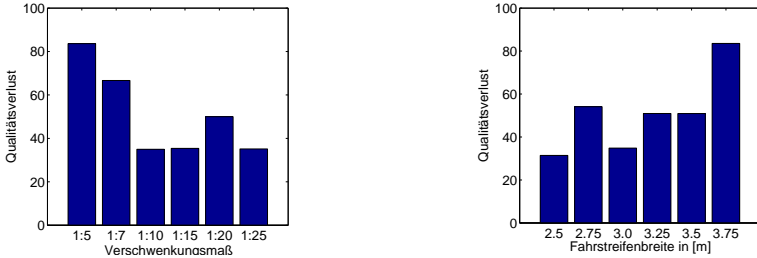


Abbildung 4: Qualitätsverluste durch den minimalen Abstand zur Randbebauung

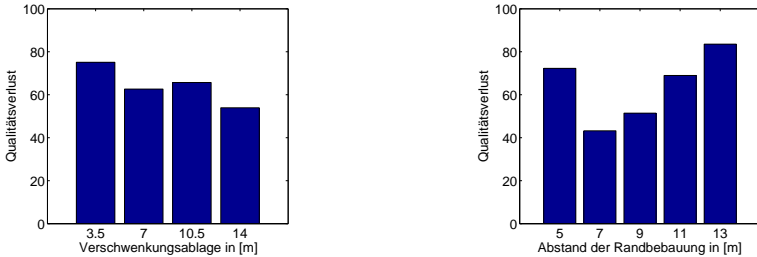


Abbildung 5: Qualitätsverluste durch die maximale Fahrstreifenablage

on zwischen dem Verschwenkungsmaß und dem Qualitätsverlust durch den minimalen Abstand zur Randbebauung gibt. Mit einem fallenden Verschwenkungsmaß steigt der Qualitätsverlust und somit fällt die Performance der Fahrfunktion. Des Weiteren zeigt Abbildung 4, dass mit einer steigenden Fahrstreifenbreite der Qualitätsverlust ebenfalls steigt und somit eine positive Korrelation vorhanden ist, die allerdings nicht so stark ausgeprägt ist, wie die Korrelation des Verschwenkungsmaß. Abbildung 5 zeigt, dass die Verschwenkungsablage und der Qualitätsverlust durch die maximale Ablage im eigenen Fahrstreifen ebenfalls korrelieren. Eine steigende Verschwenkungsablage führt zu einem fallenden Qualitätsverlust. Somit hat die Fahrfunktion bei der Faktorstufe $3.5m$ den größten Qualitätsverlust. Weiterhin zeigt sich eine positive Korrelation zwischen dem Abstand der Randbebauung und dem Qualitätsverlust durch die maximale Fahrstreifenablage. Auffällig ist die Stufe $5m$ mit einem hohen Qualitätsverlust, die nicht zur Korrelation passt. Bei der Analyse der Testfälle zeigt sich, dass von den 7 Versuchen, die auf dieser

Stufe durchgeführt wurden, 3 Versuche mit einer Verschwenkungsablage von 3.5m durchgeführt wurden und somit der hohe Qualitätsverlust auf diesen Faktor zurückzuführen ist.

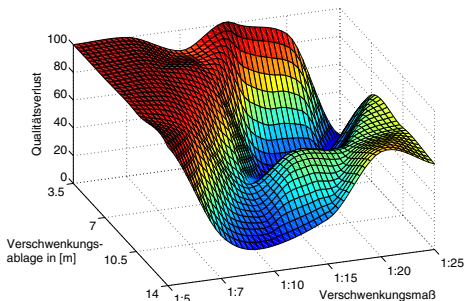


Abbildung 6: maximaler Qualitätsverlust durch den minimalen Abstand zur Randbebauung und der Ablage im eigenen Fahrstreifen anhand der Einflussfaktoren Verschwenkungsablage und Verschwenkungsmaß

Abbildung 6 zeigt den maximalen Qualitätsverlust der Bewertungskriterien minimaler Abstand zur Randbebauung und Verschwenkungsmaß. Dieser wird anhand der Einflussfaktoren Verschwenkungsablage und Verschwenkungsmaß aufgetragen. Das Ergebnis der Auswertung zeigt, dass ein geringes Verschwenkungsmaß und eine geringe Verschwenkungsablage zu einem maximalen Qualitätsverlust führen. Weiterhin kann gezeigt werden, dass die beiden Einflussfaktoren einen verstärkenden Einfluss haben, da der Qualitätsverlust nicht linear ansteigt über die verschiedenen Faktorstufen. Abbildungen 7 und 8

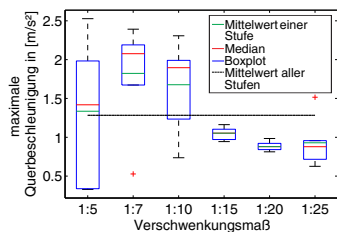


Abbildung 7: Maximale Querbeschleunigungen aufgetragen auf das Verschwenkungsmaß

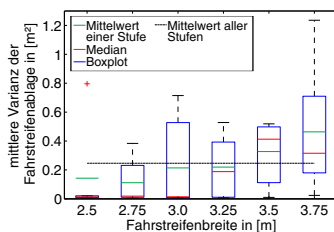


Abbildung 8: Mittlere Varianz der Ablage aufgetragen auf die Fahrstreifenbreite

zeigen die Ergebnisse der Analyse der Mittelwerte. Es werden die Effekte des Verschwenkungsmaß auf die maximal gemessene Querbeschleunigung bewertet. Die Messwerte zeigen, dass mit steigendem Verschwenkungsmaß die maximale Querbeschleunigung sinkt. Auffällig ist, dass der Wert bei der 1 : 5 Verschwenkungen zunächst gering ist. Dies ist

darin begründet, dass in den Testfällen auf dieser Faktorstufe die Fahrfunktion nicht in die Verschwenkung eingelenkt hat und somit keine Querbeschleunigung aufgebaut wurde. Weiterhin wird die mittlere Varianz der Ablage im eigenen Fahrstreifen anhand der Fahrstreifenbreite bewertet. Es zeigt sich, dass die Varianz der Ablage über die steigende Fahrstreifenbreite stetig zunimmt, was an einer nicht realisierten Fahrstreifenmittelführung liegt.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde das Konzept eines modularen virtuellen Testbaukastens vorgestellt. Ziel des Baukastens ist die systematische Szenarioerstellung, Testung und Bewertung von Fahrfunktionen. Bei der Szenarioerstellung wurden kombinatorische Verfahren verwendet, um die Anzahl der Testfälle gezielt zu reduzieren. Weiterhin wurden Ansätze zur systematischen Testfallauswertung vorgestellt. Dabei wurde das Konzept der Analyse der Mittelwerte sowie der Qualitätsverlustfunktionen vorgestellt. Die Anwendung des modularen Testbaukastens und der systematischen Testfallauswertung wurde anhand der Fallstudie eines Baustellenassistenten erläutert. Trotz der reduzierten Anzahl von Testfällen konnten durch die Qualitätsverlustfunktionen Systemgrenzen identifiziert werden. Durch die Analyse der Mittelwerte wurden die Effekte der Faktorstufen für einige Bewertungskriterien gezeigt. Für zukünftige Forschungsarbeiten werden weitere Fallstudien mit Variationen der Parameterstufen durchgeführt. Dabei werden auch dynamische Objekte in die Szenarien integriert. Es soll unter anderem evaluiert werden, wie das Verhalten der dynamischen Objekte mit Hilfe des modularen Testbaukastens bewertet werden kann.

Literatur

- [1] A. Bartels, *High Automated Driving Functions: Future challenges and research needs from the perspective of Volkswagen group research*, TRB workshop on road vehicle automation, Stanford, Juli 2013
- [2] T. Bock, *Vehicle in the loop : Test- und Simulationsumgebung für Fahrerassistenzsysteme*, Cuvillier Verlag, Göttingen, 2008
- [3] Bundesministerium für Verkehr, *Richtlinien für die Sicherung von Arbeitsstellen an Straßen (RSA)*, Kirschbaum, Bonn, 2009
- [4] T. M. Gasser, *Rechtsfolgen zunehmender Fahrzeugautomatisierung: gemeinsamer Schlussbericht der Projektgruppe* in *Berichte der Bundesanstalt für Straßenwesen: Fahrzeugtechnik*, 2012, Wirtschaftsverlag NW
- [5] F. Hendriks, R. Pelders, M. Tideman, *Future Testing of Active Safety Systems* in *SAE International Journal of Passenger Cars- Electronic and Electrical Systems*, 3(2), 170-175, 2010
- [6] R. Khorramshahgol, G. R. Djavanshir, *The Application of Analytic Hierarchy Process to Determine Proportionality Constant of the Taguchi Quality Loss Function* in *Transactions on Engineering Management*, 55(2), 340-348, 2008

- [7] B. Klein, *Versuchsplanung - DoE: Einführung in die Taguchi/Shainin-Methodik*, Oldenbourg, München, 2007
- [8] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, J. Lawrence, *IPOG/IPOG-D: efficient test generation for multi-way combinatorial testing* in Journal of Software Testing, Verification and Reliability, 18(3), 125-148, 2008
- [9] M. Maurer, *Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen*, Dissertation, Universität der Bundeswehr München, München, 2000.
- [10] M. Nentwig, M. Stamminger, *A Method for the Reproduction of Vehicle Test Drives for the Simulation Based Evaluation of Image Processing Algorithms* in Proceedings of International Conference on Intelligent Transportation Systems (ITSC), 1307-1312, Madeira Island, Portugal, 2010
- [11] M. Nentwig, M. Stamminger, *Hardware-in-the-loop testing of computer vision based driver assistance systems* in Proceedings of Intelligent Vehicles Symposium (IV), 339-344, Baden-Baden, 2011
- [12] M. Nentwig, M. Maximilian, M. Stamminger, *Concerning the Applicability of Computer Graphics for the Evaluation of Image Processing Algorithms* in Proceedings of International Conference on Vehicular Electronics and Safety (ICVES), 205-210, Istanbul, Türkei, 2012
- [13] K.v. Neumann-Cosel, E. Roth, D. Lehmann, J. Speth, A. Knoll, *Testing of Image Processing Algorithms on Synthetic Data* in Proceedings of International Conference on Software Engineering Advances (ICSEA), 169-172, Porto, Portugal, 2009
- [14] ERTICO-ITS Europe, *Response 3: code of practice for design and evaluation of adas*, Information Society, V5.0, 2009
- [15] W. Scholl, *UR:BAN - Homepage*, abgerufen 05.10.2013
- [16] F. Schuldt, F. Saust, B. Lichte, M. Maurer, S. Scholz, *Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen*, in AAET 2013 - Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel, Braunschweig, 2013
- [17] K. Siebertz, D.v. Bebber, T. Hochkirchen, *Statistische Versuchsplanung*, Springer Verlag, Heidelberg, 2010
- [18] G. Taguchi, S. Chowdhury, Y. Wu, *Taguchi's quality engineering handbook, Chapter 7: Introduction to the Quality Loss Function*, John Wiley, Michigan, 2005
- [19] M. Tideman, S. J. Janssen, *A Simulation Environment for Developing Intelligent Headlight Systems* in Proceedings of Intelligent Vehicles Symposium (IV), 225-231, San Diego, CA, USA, 2010
- [20] H. Winner, S. Hakuli, G. Wolf, *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort, Kapitel 44: Quo vadis, FAS?*, Wiesbaden, Vieweg + Teubner, 2009



Herausgeber:

Berthold Färber

(Organisation)

Klaus Dietmayer

Klaus Bengler

Markus Maurer

Christoph Stiller

Hermann Winner

ISBN 978-3-00-044955-0