

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 Fahrzeugaktuatorik 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

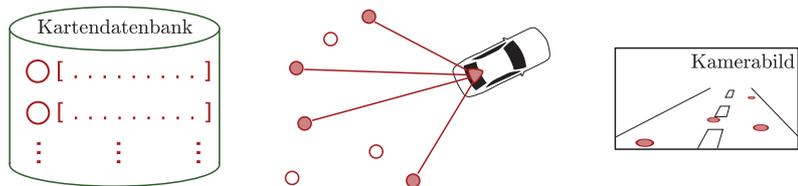


Abbildung 3: Punktmerkmalsbasierte Videolokalisierung mit rückwärtsgerichteter Kamera: 3D Landmarken werden aus einer Kartendatenbank mit dazugehörigen Deskriptoren geladen und Bildpunkten des aktuellen Kamerabildes zugeordnet. Gezeigt ist die Szene in vogelperspektive, wobei rote Kreise erfolgreich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

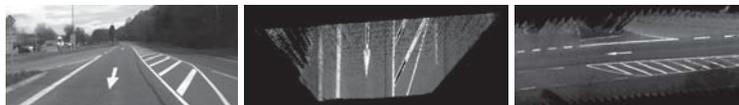


Abbildung 5: Einzelbild Stereokamera, rekonstruierte Draufsicht, Überlagerung.

des nahen Fahrzeugumfelds erzeugt. Diese 3D-Punkte werden auf die Grunde Ebene 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 erlaubte 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

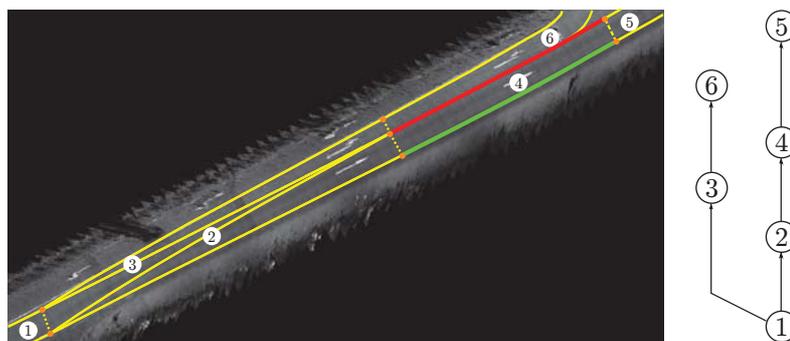


Abbildung 6: Fahrbahnrandgeometrien und resultierende Fahrstreifentopologie

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

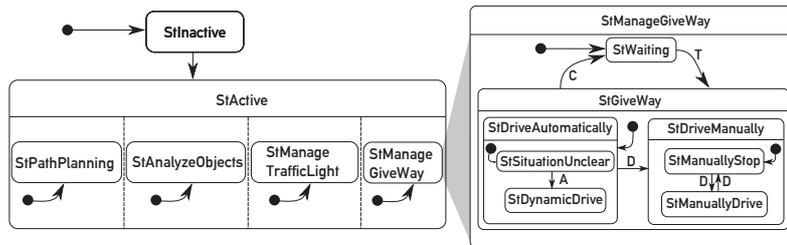


Abbildung 7: Ausschnitt des Zustandsautomaten zur Verhaltensgenerierung.

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 *kartenetrieben*.

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-

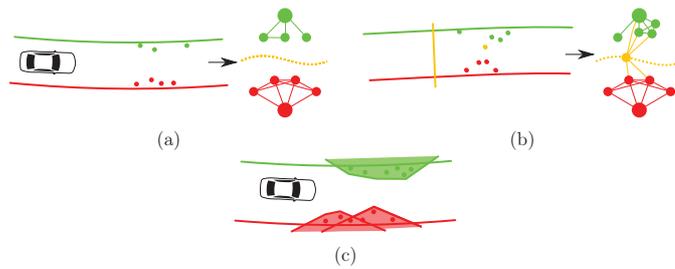


Abbildung 8: Vorverarbeitung der Hindernisdaten unter Berücksichtigung des Verlaufes des Fahrkorridors.

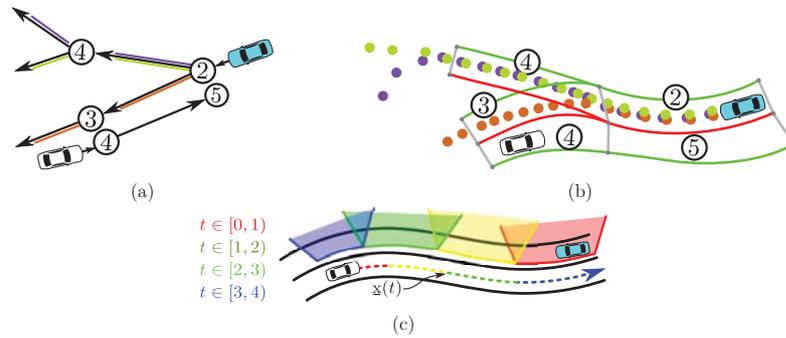


Abbildung 9: Nebenbedingungen für bewegte Objekte

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 prädiziert, 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 vorfahrtberechtigte 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 Abzisse 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 vorfahrtberechtigten Fahrzeuges (zyan), werden zwei gleichförmig beschleunigte Trajektorien angenommen, die den möglichen Beschleunigungsspielraum des vorfahrtberechtigten Fahrzeuges widerspiegeln. Die Zeiten, an denen diese Trajektorien die Halteinie 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.

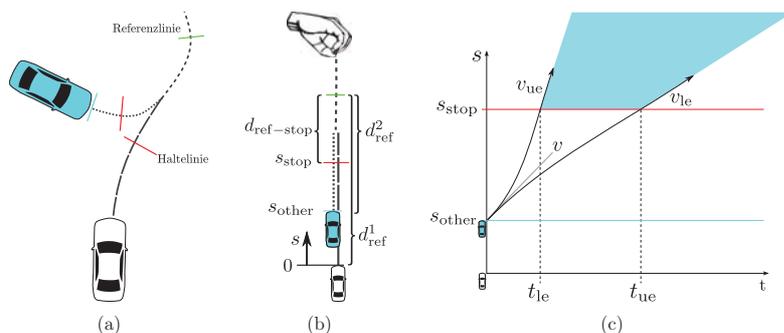


Abbildung 10: Vorfahrtssituation des Merge-Typs: (a) Szene in der Draufsicht, (b) 1-D-Abstraktion, (c) resultierendes Weg-Zeit-Hindernis (zyan).

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

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

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

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 Korridorändern verläuft. Die Funktionen d_{links} und d_{rechts} sind die vorzeichenbehafteten Abstandsfunktionen zu den Korridorändern, 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

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

erzwingt zusätzliche Glattheit 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 Fahrzeugaktuatorik 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. Hertwich. 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. Latégahn, 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. Latégahn and C. Stiller. City gps using stereo vision. In *IEEE International Conference on Vehicular Electronics and Safety*, Istanbul, Türkei, Juli 2012.
- [11] H. Latégahn 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.