

Bahnplanung für das autonome Fahrzeug ANNIEWAY

Julius Ziegler* und Benjamin Pitzer†

Zusammenfassung: Dieser Beitrag widmet sich den verschiedenen Bahnplanungsstrategien, die an Bord des autonomen Fahrzeuges ANNIEWAY beim Wettbewerb *Urban Challenge 2007* zum Einsatz kamen. Aus dem Reglement ergeben sich unmittelbar zwei unterschiedliche Szenarien für die Bahnplanung: In *strukturierter* Umgebung sind Topologie und Geometrie des Straßennetzes durch die Vorgabe von digitalem Kartenmaterial zumindest grob bekannt. *Unstrukturierte* Umgebung wird im Kartenmaterial lediglich durch einen Polygonzug (*Zone*) umrissen, der Parkplätze oder unbefestigtes Gelände enthalten kann. Innerhalb solcher Zonen muss zunächst ein Graph für die Bahnplanung errichtet werden. Team ANNIEWAY verwendet hierzu einen geometrischen Graphen, der aus einem dynamischen Fahrzeugmodell abgeleitet wird. Dies sichert die Fahrbarkeit aller generierbaren Bahnen zu und ermöglicht zusätzlich die Ableitung einer Vorsteuerung für den nachgeschalteten Regler. Der Graph wird durch einen informierten Suchalgorithmus expandiert, wobei zur Beschleunigung eine Heuristik einfließt, die sowohl die kinematischen Zwangsbedingungen des Fahrzeuges als auch die Topologie des freien Raumes berücksichtigt. Das Gesamtsystem ist in der Lage, alle praktisch auftretenden Bahnplanungsaufgaben zu lösen. Hierzu gehören präzises Einparken, Dreipunktwenden und die Navigation über längere Distanzen, in strukturierter und unstrukturierter Umgebung.

Schlüsselwörter: AnnieWAY, Urban Challenge, Bahnplanung, kollisionsfrei

1 Einleitung

Die kollisionsfreie Bahnplanung ist beim Betrieb eines autonomen Fahrzeuges von größter Wichtigkeit. Das Fahrzeug muss eine bestimmte Zielkonfiguration anfahren, dabei aber Hindernissen in seiner Umgebung ausweichen, um weder sich noch die Umgebung zu beschädigen. Die Bahnplanung ist ein gut erforschtes Teilgebiet der Robotik [6], stellt aber im Einzelfall immer noch hohe Anforderung an Modellierung und Algorithmik. Herausforderungen im vorliegenden Fall sind insbesondere die nichtholonome Kinematik des Fahrzeuges, die geforderte hohe Integration mit der vorhandenen Sensorik und Regelung sowie die Echtzeitanforderung.

Aus dem Reglement der *Urban Challenge* (UC) [2] ergeben sich zwei deutlich trennbare Anwendungsfälle, die zu zwei separaten Bahnplanungsmodulen führen und hier getrennt behandelt werden.

*Julius Ziegler ist wissenschaftlicher Mitarbeiter am Institut für Mess- und Regelungstechnik der Universität Karlsruhe, 76131 Karlsruhe (e-mail: ziegler@mrt.uka.de).

†Benjamin Pitzer promoviert am Institut für Mess- und Regelungstechnik der Universität Karlsruhe, 76131 Karlsruhe (e-mail: ben.pitzer@gmail.com)

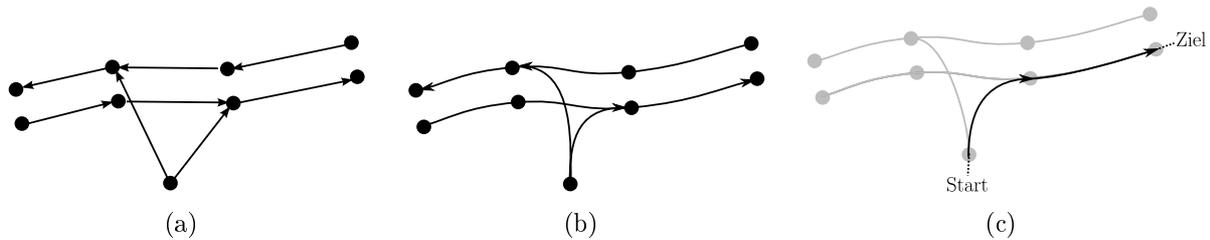


Abbildung 1: Vorverarbeitung der RNDF-Daten. (a) Direkt aus dem RNDF abgeleiteter Graph. (b) Glättung mit Splines. (c) Missionsliste. Der Krümmungsverlauf der Bahn ist stetig.

In *strukturierter* Umgebung, also in Bereichen mit einem ausgebaute Straßennetz, dass die erlaubte Bewegungsrichtung nach der Straßenverkehrsordnung vorgibt (ein Verlassen der ausgebauten Fahrwege ist nicht zulässig), kann auf digitales Kartenmaterial, das vom Veranstalter zur Verfügung gestellt wird, zurückgegriffen werden. Aus der digitalen Karte lässt sich ein geometrischer Graph ableiten, aus dem eine fahrbare Bahn durch Abtasten erzeugt werden kann. Dieser Prozess wird von einer hierarchisch organisierten Zustandsmaschine gesteuert, welche zusätzlich die Einhaltung der Straßenverkehrsregeln sicher stellt. Das für dieses Szenario entworfene *Road Planning*-Modul wird in Abschnitt 2 beschrieben.

Darüberhinaus war die freie Navigation in so genannten *Zonen* gefordert. Zonen sind in der digitalen Karte lediglich durch einen einhüllenden Polygonzug gekennzeichnet. Im Gegensatz zum Road-Planning-Modul kann der für diesen Anwendungsfall entworfene *Zone Planner* nicht auf Vorwissen aus einer digitalen Karte zurückgreifen. Sämtliches Wissen über die Umgebung muss aus der fahrzeugeigenen Sensorik abgeleitet werden. Der *Zone Planner* wird in Abschnitt 3 beschrieben.

Beiden Modulen ist das gleiche, in [14] beschriebene Reglermodul nachgeschaltet, dass einen orbitalen Trackingregler implementiert. Als Eingabe dient ihm eine Punktmenge (*Curvepoints*), die von der zu fahrenden Bahn abgetastet wurde. Diese Punkte sind zusätzlich mit Orientierung und Krümmung versehen, so dass aus ihnen ein Vorsteuerterm für den Regler abgeleitet werden kann.

Experimente und Ergebnisse werden für beide Bahnplanungsmodule gemeinsam in Abschnitt 4 präsentiert.

2 Bahnplanung in strukturierter Umgebung

Die in diesem Abschnitt beschriebene Methode wurde für den Fall entworfen, das hinreichend genaues digitales Kartenmaterial zur Verfügung steht. Dieses muss die Topologie des Straßennetzes vollständig und fehlerfrei wiedergeben, es müssen also alle Kreuzungen und Abzweigmöglichkeiten topologisch korrekt enthalten sein. Weiterhin muss die Geometrie in Form von GPS-Wegpunkten ausreichend genau vorgegeben sein.

Diese Voraussetzungen sind im Regelwerk der UC [2] durch Definition des RNDF-Dateiformats (road network definition file) formalisiert. Die RNDFs, die in der Endrunde der UC verwendet wurden, enthielten Wegpunkte in Abständen von etwa 20 m (in engen Kurven) bis 100 m (auf langen Geraden).

In einer Vorverarbeitung werden die gegebenen Wegpunkte einer Glättung durch Splines unterzogen (Abbildung 1(a)-(b)). Weiterhin werden sie fortlaufend basierend auf einer

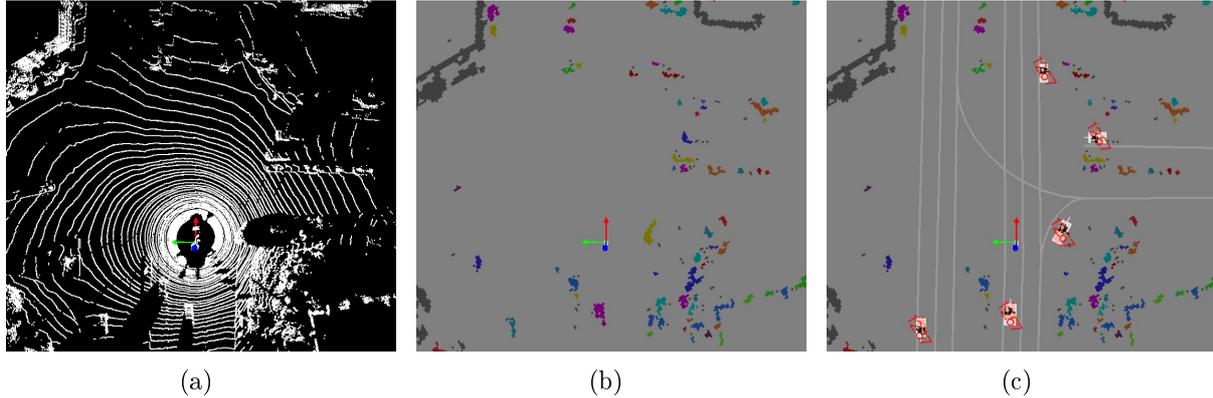


Abbildung 2: Objekterkennung und Graphrepräsentation. (a) Unverarbeitete Laserscannerdaten. (b): Clustering. Zusammengefasste Punkte sind gleich eingefärbt. (c): Erkannte Objekte werden zeitlich verfolgt und Kanten des Graphen zugeordnet, der das Straßennetz repräsentiert.

Fahrspurerkennung lokal korrigiert [8]. Nach der Vorverarbeitung ist das Straßennetz als ein *geometrischer* Graph repräsentiert: ein Knoten des Graphen entspricht einer Position im Weltkoordinatensystem, und den Kanten unterliegt ebenfalls eine geometrische Repräsentation, in unserem Fall sind dies Splines.

2.1 Bahngenerierung aus einer geometrischen Graphstruktur

Die Erfüllung einer *Mission* gemäß UC-Reglement erfordert das Anfahren bestimmter, im RNDF markierter Wegpunkte in vorgegebener Reihenfolge. Durch Graphsuche (zum Beispiel Dijkstras Algorithmus) kann eine Liste von Kanten (*Missionsliste*) gefunden werden, die nacheinander befahren werden müssen, um die Mission in kürzest möglicher Zeit zu erfüllen. Wird eine Kante, die im Graph repräsentiert wird, von der Sensorik als unbefahrbar eingestuft (Blockade), wird der Graph entsprechend modifiziert. Die Missionsliste muss dann neu berechnet werden.

Durch Verkettung aller in der Missionsliste enthaltenen Splinesegmente erhält man eine Bahn, die die Mission erfüllt (Abbildung 1(c)). Ein stetiger Krümmungsverlauf ist dabei gegeben, die Bahn ist deshalb fahrbar und eignet sich für den nachgeschalteten Regler. Sie enthält bereits sämtliche Abbiegemanöver.

Wie man sieht, ist in strukturierter Umgebung die Generierung einer fahrbaren Bahn verhältnismäßig einfach. Diese Bahn vom Regler in einer Weise abfahren zu lassen, die den Verkehrsregeln entspricht, stellt allerdings eine Herausforderung dar. Diese wurde durch Modellierung des reaktiven Fahrzeugverhaltens als ein hierarchischer Zustandsautomat angegangen. Das Vorgehen kann hier aus Platzgründen nur angerissen werden, für eine detaillierte Beschreibung sei auf [4, 5] verwiesen.

2.2 Regelgerechtes Abfahren der Bahn mit einem hierarchischen Zustandsautomaten

Um ein regelgerechtes Verhalten des Fahrzeuges zu ermöglichen, ist zunächst eine Erfassung anderer Verkehrsteilnehmer erforderlich. Weiterhin müssen diese semantisch in die graphbasierte Umfeldrepräsentation des Fahrzeuges eingebettet werden.

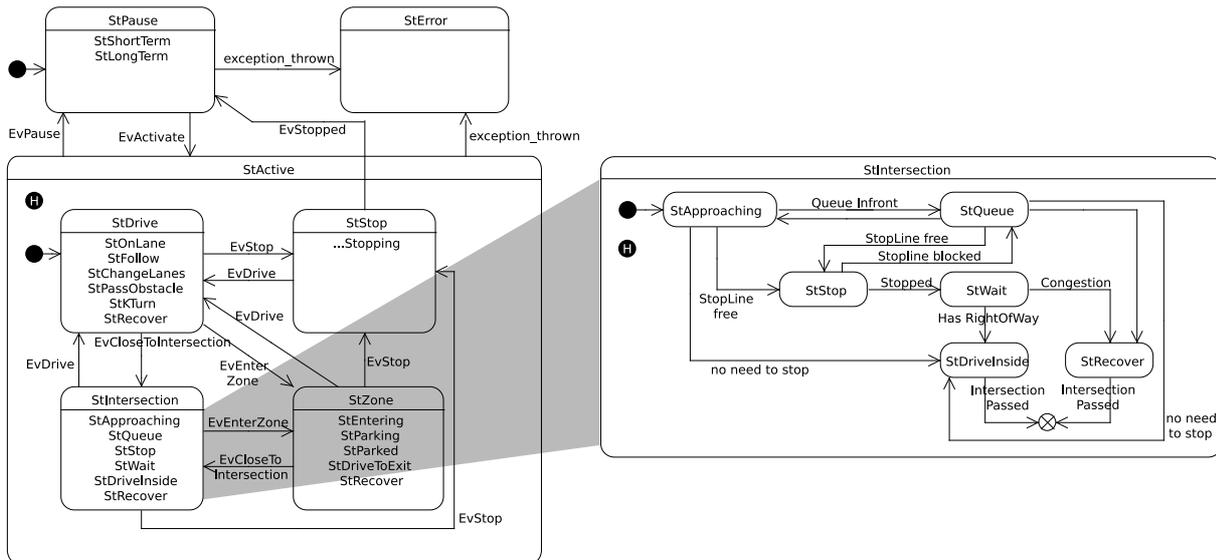


Abbildung 3: Hierarchischer Zustandsautomat. Zustandsnamen ist ein St... vorangestellt und Ereignisse beginnen mit Ev... Falls Zustände Unterzustände enthalten, kann dies in einer Kurzfassung dargestellt werden, beispielsweise enthält StDrive die Unterzustände StOnLane, StFollow und so weiter. Für StActive ist hier das detaillierte Innenleben dargestellt, und für StIntersection finden sich beide Darstellungen.

Das Objekterkennungsmodul [7] extrahiert bewegte Objekte aus den Laserscannerdaten durch Clustering und ein nachgeschaltetes Trackingverfahren (Abbildung 2). Die so erkannten Objekte werden der Graphkante zugeordnet, die ihre Position und Bewegungsrichtung am besten erklärt. Durch diese Abbildung der Objekte auf den Graphen wird semantische Information gewonnen, es kann beispielsweise leicht festgestellt werden, ob es sich bei einem Objekt um ein in der gleichen Spur vorausfahrendes Fahrzeug handelt, oder ob es sich auf einem Arm der Kreuzung befindet, die gerade angefahren wird.

Da mit der Missionsliste die zu fahrende Bahn bereits vollständig geplant ist, muss ein Planungsmodul noch entscheiden, wo auf dieser Bahn die Fahrt unterbrochen werden muss (zum Beispiel, um Vorfahrt zu gewähren). Einige weitere situationsabhängige Aktionen müssen noch durchgeführt werden, beispielsweise sind Blinker zu setzen und Geschwindigkeitsbegrenzungen einzuhalten. Dieses Verhalten wird unter Berücksichtigung der gewonnenen Information über die Umgebung mit Hilfe einer hierarchisch organisierten Zustandsmaschine erzeugt. Die Möglichkeit, Zustandsautomaten hierarchisch zu organisieren, wurde zuerst von David Harel in [5] beschrieben. Abbildung 3 zeigt den Zustandsautomaten, der bei ANNIEWAY zum Einsatz kam und illustriert das Prinzip der Hierarchisierung. Eine Beschreibung der grafischen Notation ist in [5] enthalten, und [4] beschreibt die konkrete Struktur der Zustandsmaschine aus Abbildung 3.

3 Bahnplanung in unstrukturierter Umgebung

Dieser Abschnitt gibt einen Überblick über das Verfahren, das zur Bahnplanung abseits strukturierender Verkehrswege eingesetzt wurde. Eine detailliertere Beschreibung findet sich in [15]. Die Bahnplanung wird als Problem kürzester Pfade in graphentheoretischem

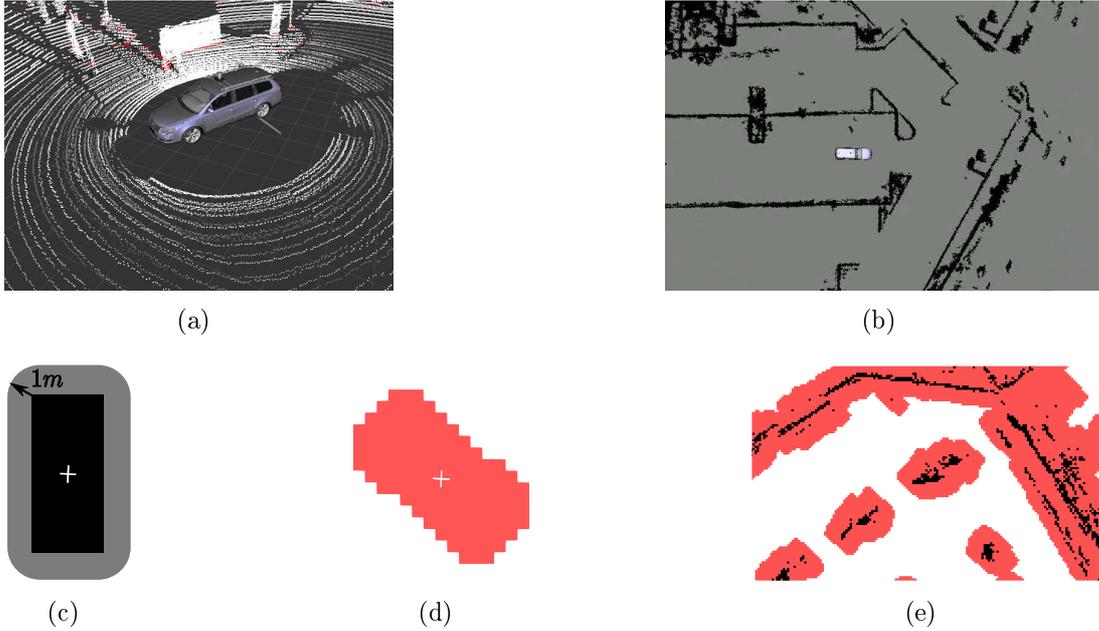


Abbildung 4: Laserscanner, Hinderniskarte und ihre Übertragung in den Konfigurationsraum. (a)-(b): Vertikale Strukturen werden erkannt und in eine diskrete, zweidimensionale Karte übertragen. (c): Die Fahrzeugform wird durch einen 1 m-Sicherheitsabstand erweitert. (d): Strukturelement für eine diskrete Fahrzeugorientierung. (e): Ergebnis der Faltung der Hinderniskarte mit dem Strukturelement. Platziert man das Strukturelement im roten Bereich, so muss es sich mit einem Hindernis überschneiden.

Kontext betrachtet. Nach Definition des Konfigurationsraums und Beschreibung der sensorischer Voraussetzung wird die Konstruktion des zu Grunde liegenden Suchgraphen beschrieben. Anschließend wird ausführlich auf Kostenfunktionen eingegangen, die zur Beschleunigung der Graphsuche eingesetzt wurden.

3.1 Konfigurationsraum

Der Bahnplanung auf einer Freifahrfläche liegt ein dreidimensionaler Konfigurationsraum \mathcal{C} zu Grunde, der von der zweidimensionalen Fahrzeugposition \mathbf{x} und der Fahrzeugorientierung ψ aufgespannt wird. Der Konfigurationsraum wird auf einen diskreten Raum beschränkt. Dieses Vorgehen lässt sich dadurch motivieren, dass sämtliche Information über die Umgebung in Form einer diskreten Karte vorliegt, die aus den Daten eines hochauflösenden Laserscanners, der auf dem Dach des Fahrzeuges angebracht ist, gewonnen wurde (Abbildung 4). Eine Selbstlokalisierung des Fahrzeuges ist durch eine Koppelnavigation aus GPS, Odometrie und Inertialsensorik gegeben, so dass eine räumliche Registrierung der Sensordaten möglich ist und durch Akkumulierung in der Karte ein vollständiges Bild der Umgebung erhalten werden kann.

Die kollisionsfreie Teilmenge von \mathcal{C} , der so genannte *freie Raum* des Fahrzeuges, kann aus dieser diskreten Hinderniskarte durch n_ψ Faltungsoperationen berechnet werden, wobei n_ψ die Anzahl diskreter Orientierungen des Fahrzeuges bezeichnet [9]. Die Hinderniskarte wird hierzu mit Strukturelementen gefaltet, die der Form des Fahrzeuges entsprechen, aber um einen Sicherheitsabstand von 1 m erweitert wurden (Abbildung 4(c)).

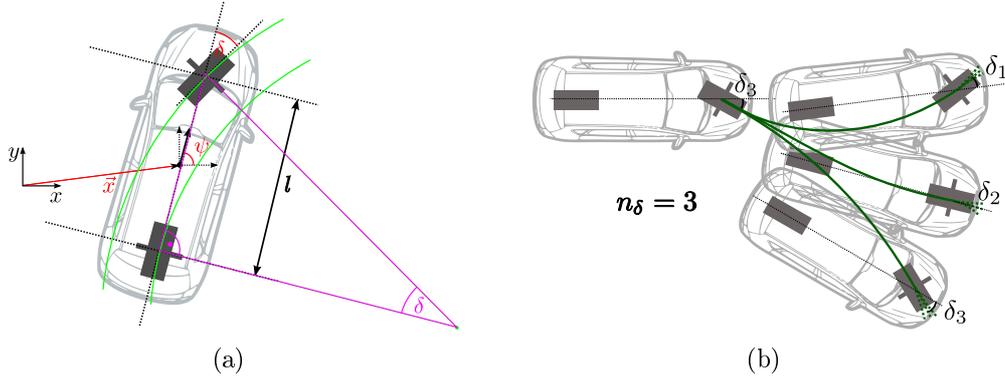


Abbildung 5: (a): Kinematisches Einspurmodell, das dem Suchgraphen zu Grunde liegt. Die kinematischen Modellgleichungen können aus den Seitenverhältnissen des rechtwinkligen Dreiecks hergeleitet werden. (b): Ausschnitt des Suchgraphen. Jeder Knoten hat n_δ Nachfolger. Sie sind durch Kurvenssegmente verbunden, die Klothoidenabschnitten ähneln.

Durch Vorberechnung des Konfigurationsraums in diskreter Form kann ein Kollisionstest für eine bestimmte Fahrzeugkonfiguration schnell und in konstanter Zeit durchgeführt werden.

3.2 Impliziter Suchgraph

Um die Fahrbarkeit der zu generierenden Bahn zu garantieren, wird ein geometrischer Graph errichtet, in dem alle Pfade fahrbar sind. Ein Knoten des Suchgraphen wird vollständig durch das Tupel $(\mathbf{x}, \psi, \delta)$ beschrieben, wobei \mathbf{x} , ψ und δ Position, Orientierung und Lenkwinkel einer Instanz des zugrunde liegenden kinematischen Einspurmodells beschreiben (Abbildung 5(a)). δ entstammt einer Menge von n_δ diskreten Lenkwinkeln, die äquidistant über den Bereich realisierbarer Lenkwinkel verteilt sind. Die Nachfolger eines Knotens werden so gewählt, dass ein verbindender Bogen die kinematischen Zwangsbedingungen des Fahrzeuges erfüllt.

Um die Nachfolger eines Knotens $v_p = (\mathbf{x}_p, \psi_p, \delta_p)$ zu erzeugen, werden die kinematischen Modellgleichungen für die Startwerte aus v_p und konstante $\dot{\delta} = \frac{\delta_p - \delta_i}{s}$ für alle $\delta_i \in D$ gelöst, wobei s die fest gewählte Bogenlänge aller die Knoten verbindenden Bögen ist. Dieses Vorgehen ist äquivalent dazu, das Einspurmodell über die Distanz s zu simulieren und dabei die Vorderräder gleichförmig von δ_p nach δ_i zu bewegen.

Für die Knotenmenge $\{(\mathbf{0}, 0, \delta_i), \delta_i \in D\}$ ergeben sich n_δ^2 Nachfolger und weitere n_δ^2 , sofern Rückwärtsbewegung erlaubt ist. Nachfolger anderer Knoten können schnell aus dieser vorberechneten Menge erzeugt werden, indem ein Element verschoben und gedreht wird. Abbildung 5(b) zeigt einen Ausschnitt des Suchgraphen.

3.3 Heuristiken

Um den Suchprozess zu führen, wurden zwei verschiedene Kostenfunktionen (*Heuristiken*) kombiniert. Die *direkte* Kostenfunktion berücksichtigt ausschließlich die kinematischen Zwangsbedingungen des Fahrzeuges. Sie ist unabhängig von der Belegung der Hinderniskarte. Wird die Distanz zum Ziel von der Notwendigkeit, Hindernisse zu umfahren, dominiert, das heist, wenn kein direkter Pfad zum Ziel existiert, unterschätzt die direkte Kostenfunk-

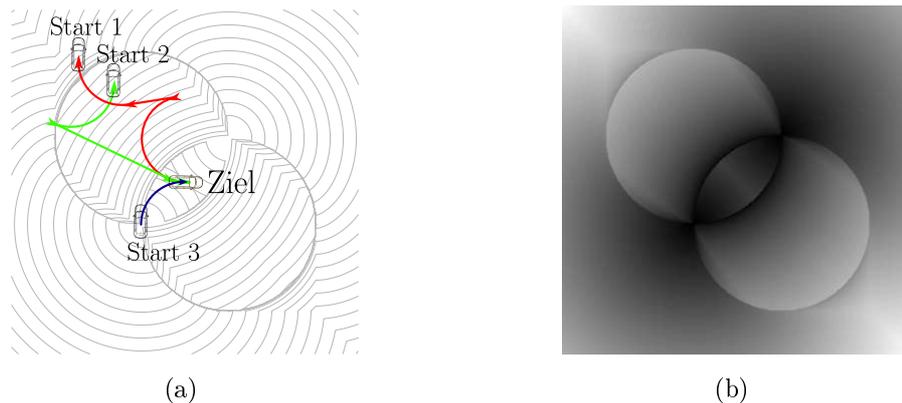


Abbildung 6: Lokale Kostenfunktion für eine Startorientierung von 90° und eine Zielorientierung von 0° . (a): Drei Beispiele für minimale RTR-Pfade. (b): Die RTR-Metrik dicht über \mathbb{R}^2 ausgewertet, Zielposition in der Mitte des Bildes, wie in (a). Hell: weite, dunkel: kurze Distanz. Dem linken Bild sind Äquidistanzlinien überlagert.

tion die Entfernung stark. Insbesondere bei der Navigation über längere Distanzen, in der Gegenwart vieler Hindernisse oder bei schwieriger, labyrinthartiger Anordnung der Hindernisse ist dies der Fall. Aus diesem Grund wird eine weitere Kostenfunktion entworfen, die vom Voronjgraphen des freien Raumes abgeleitet wird. Sie bringt Wissen über die Lage und Form der Hindernisse ein.

3.3.1 Direkte Kostenfunktion

Die so genannte *RTR-Metrik* [13] wird als eine *direkte* Kostenfunktion verwendet. RTR-Pfade (Rotation-Translation-Rotation) verbinden zwei Konfigurationen durch zwei Kreisbögen des kleinst möglichen Wenderadius und ein Geradensegment, das beide Kreisbögen berührt. Richtungswechsel sind hierbei möglich, und zwar jeweils beim Übergang von Kreis zu Geradensegment. Man kann leicht zeigen [13], dass sich jedes Paar von Konfigurationen durch eine endliche Anzahl von RTR-Pfaden verbinden lässt. Die RTR-Metrik ist die Bogenlänge des kürzesten derartigen Pfades. Obwohl RTR-Pfade weder die kürzest möglichen Pfade [11] sind, noch den geforderten stetigen Krümmungsverlauf aufweisen, werden sie für die Kostenfunktion bevorzugt, da sie einfach und schnell berechnet werden können. Abbildung 6 illustriert die RTR-Metrik, bei gegebener Startorientierung und Zielkonfiguration.

3.3.2 Voronjbasierte Kostenfunktion

Vom Voronjgraphen des freien Raumes des Fahrzeuges wird eine mächtige Kostenfunktion, die Wissen über die Lage der Hindernisse einbringt, abgeleitet.

Der Algorithmus, der verwendet wurde, um die Voronjlinien aus der binarisierten Hinderniskarte zu extrahieren, ist inspiriert von [10], allerdings wird die approximative Chamfer-Metrik zu Grunde gelegt, so dass die Voronjlinien mit einem schnellen Zweipassalgorithmus markiert werden können. Diese Methode ist von [3, 12] abgeleitet, dort wurde auf ähnliche Weise eine euklidische Distanztransformation berechnet. Wir erhalten die Voronjlinien als eine Menge von in Achternachbarschaft zusammenhängenden Pixeln, Abbildung 7(a) zeigt ein Beispiel.

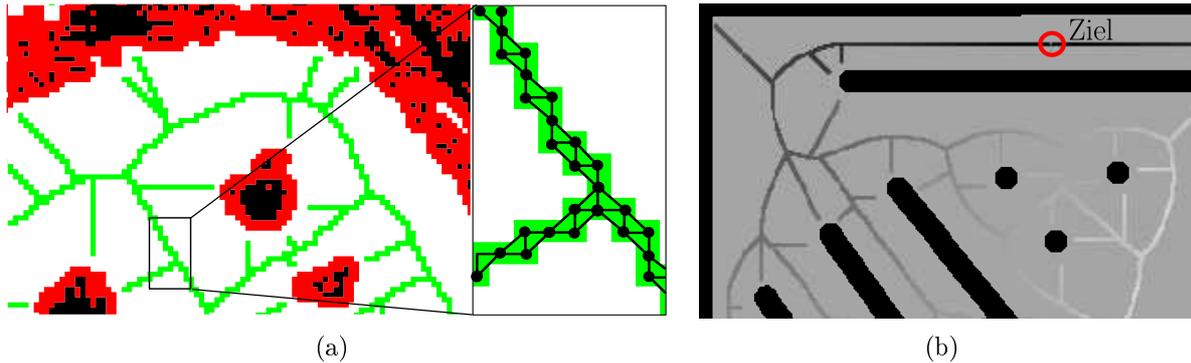


Abbildung 7: Voronojlinien. (a): Voronojlinien werden als zusammenhängende Menge von Pixeln erzeugt, die als Graph mit Achterzusammenhang betrachtet werden kann. (b): Dijkstras Algorithmus auf diesen Graphen angewendet. Hell: weite, dunkel: kurze Entfernung zum Ziel.

Auf dem Voronojgraphen wird Dijkstras Algorithmus angewendet, um die Länge des kürzesten Pfades zum Ziel für jeden Punkt auf dem Graphen zu bestimmen (siehe Abbildung 7(b)). Die Kosten für Punkte, die nicht auf dem Graphen liegen, werden bestimmt, indem diese dem jeweils nächstgelegenen Punkt auf dem Graphen zugeordnet werden.

Durch diese Kostenfunktion wird die Suche an den Voronojlinien entlang geführt. Dies hat einige Vorteile: Da der Voronojgraph die gleiche topologische Struktur wie der freie Raum aufweist, kann die Suche nicht in einer Sackgasse festlaufen. Weiterhin haben die Voronojlinien, als die Middlepunkte maximaler Inkreise, die Eigenschaft, bei gleichzeitiger Topologieerhaltung im weitest möglichen Abstand von den Hindernissen zu verlaufen. Diese Eigenschaft überträgt sich auf die geplanten Pfade und schafft so Reserven, um Fehler in Sensorik und Regelung zu kompensieren.

3.3.3 Kombination der Kostenfunktionen

Direkte und voronojbasierte Kostenfunktion können durch den Maximumsoperator kombiniert werden. Dieses Vorgehen lässt sich durch das Prinzip der Zulässigkeit, das im Kontext der A*-Suche definiert wird [6], rechtfertigen. Eine A*-Suche liefert dann den optimalen Pfad, wenn sie durch eine so genannte zulässige Heuristik gesteuert wird. Man nennt eine Heuristik zulässig, sofern sie die Kosten bis ins Ziel konsistent *unterschätzt*, also ist das Maximum zweier zulässiger Heuristiken offensichtlich immer noch zulässig.

Im vorliegenden Fall führt die Maximumsbildung dazu, dass die direkte Heuristik dann greift, wenn ein direktes Anfahren des Ziels möglich ist. Die direkte Heuristik ist dann größer als die voronojbasierte, da sie auch Kosten berücksichtigt, die daraus resultieren, das manövriert werden muss, um das Ziel in der richtigen Orientierung zu erreichen. Bei weiterer Entfernung vom Ziel ist im Normalfall die Voronojheuristik größer, da die Kosten von der Notwendigkeit dominiert werden, Hindernisse zu umfahren.

4 Experimente und Ergebnisse

Die Abbildungen 8(a)-(c) zeigen Bilder, die aus Daten erzeugt wurden, die während des Finallaufes der Urban Challenge aufgenommen wurden. Um die Speicherbandbreite des

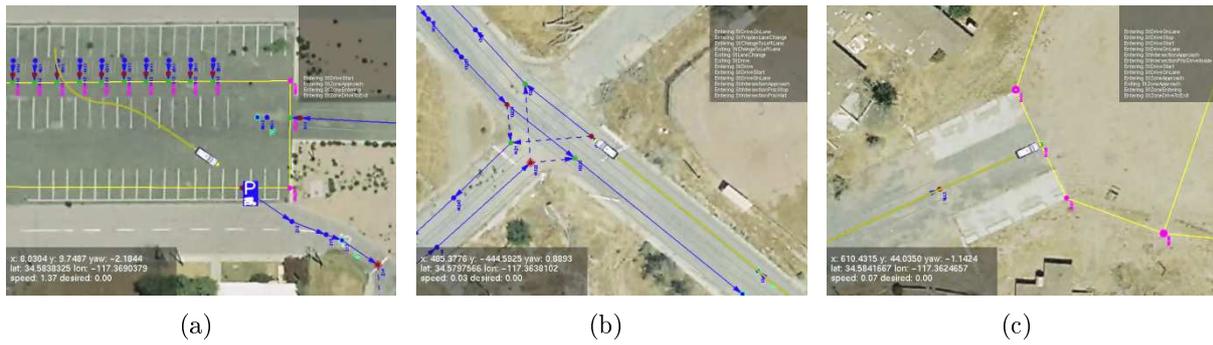


Abbildung 8: ANNIWAY im Finale der Urban Challenge. (a) Das Bahnplanungsmodul für unstrukturierte Umgebung führt ANNIWAY aus der Startzone. (b) ANNIWAY gewährt Vorfahrt an einer Abzweigung. (c) Halteposition im Finale, verursacht durch Rechnerabsturz.

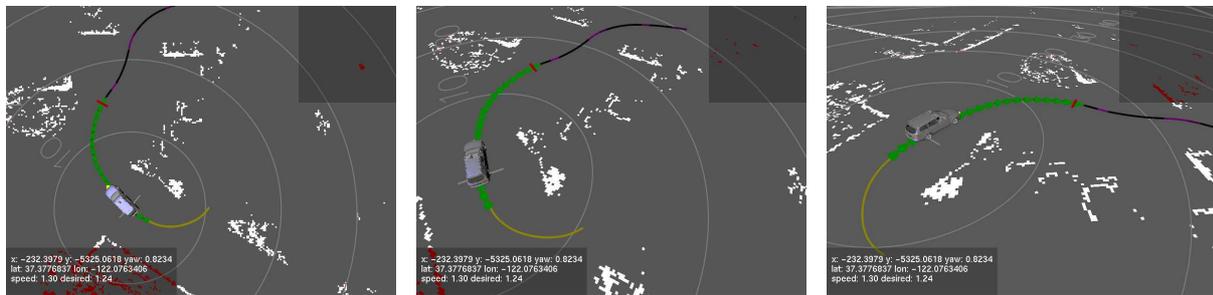


Abbildung 9: Bahnplanung in einer Zone mit Hindernissen. Hindernisse sind weiss dargestellt, die geplante Bahn schwarz und die zurückgelegte Bahn gelb. Grün dargestellt die Curvepoints, die dem Regler als Eingabe dienen.

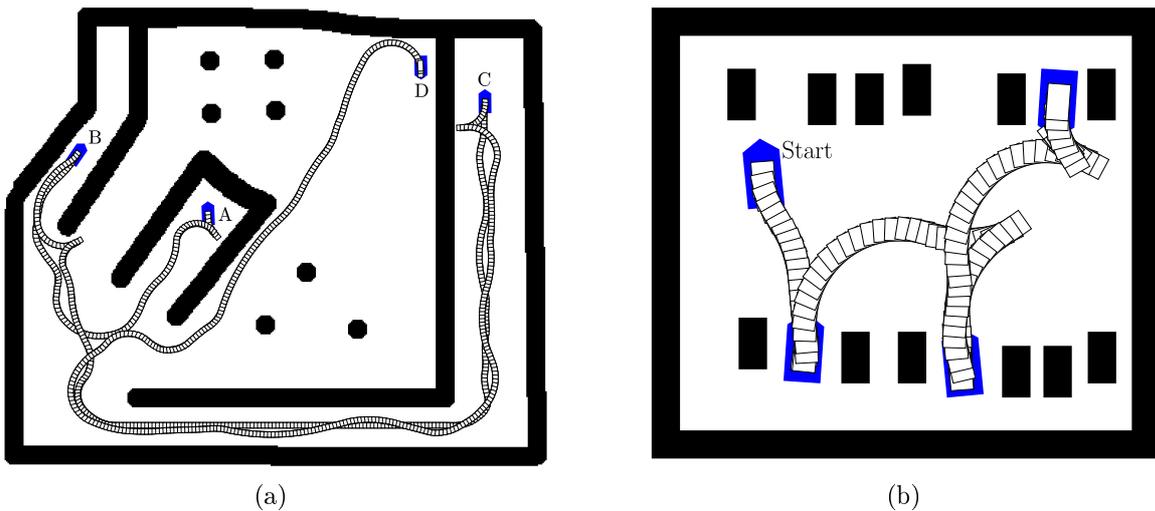


Abbildung 10: Bahnplanung auf synthetischen Hinderniskarten. (a): Navigation über längere Entfernung. Das Fahrzeug sollte nacheinander A,B,C und D in der angegebenen Orientierung anfahren. (b) Drei schwierige Einparkmanöver, die jeweils rückwärts ausgeführt wurden.

Systems nicht zu überlasten, wurden hier lediglich ANNIEWAYs Positionsdaten aufgezeichnet, durch Unterlegung von Luftbildern kann der Betrachter trotzdem einen Eindruck von der Umgebung gewinnen. In Bild 8(a) hat AnnieWAY die Startrinne verlassen und bewegt sich über eine Freifahrzone. Die Startrinne war mit Leitplanken umgeben und ließ dem Fahrzeug seitlich nur etwa einen Meter platz.

In Bild 8(b) hält AnnieWAY an einer Einmündung, um einem von links kommenden Fahrzeug (im Bild nicht sichtbar) Vorfahrt zu gewähren. An der in Bild 8(c) gezeigten Stelle blieb AnnieWAY an der Einfahrt zu einer Freifahrzone stehen, als Ursache wurde ein Rechnerabsturz diagnostiziert.

Die Anforderungen, die der Wettbewerb an das Modul zur Navigation in Zonen stellte, waren recht niedrig, es waren jeweils wenige oder keine Hindernisse zu umfahren. Abbildung 9 zeigt deshalb Versuche, die abseits des Rennens mit dem Zone Planner unternommen wurden. Die Abbildungen 10(a) und (b) zeigen anhand von synthetischen Sensordaten, das das Modul auch schwierige Einpark- und Wendemanöver ausführen kann.

5 Zusammenfassung

ANNIEWAY erreichte mit Hilfe der beschriebenen Bahnplanungsstrategien zusammen mit 11 weiteren Teilnehmern das Finale der Urban Challenge 2007. Die Funktionen der beschriebenen Module decken das Regelwerk der Urban Challenge vollständig ab. Das Fahrzeug kann selbständig Spur halten, Spurwechsel- und Parkmanöver durchführen, enge Wenden ausführen und Kreuzungssituationen handhaben. Die Verfahren zur Bahnplanung in unstrukturierter Umgebung übersteigen die Anforderungen des Wettbewerbs sogar deutlich.

6 Danksagungen

Die Arbeit in Team ANNIEWAY wird inspiriert von laufender Forschung innerhalb des transregionalen Sonderforschungsbereichs 28 *Kognitive Automobile* [1].

Literatur

- [1] *KogniMobil Homepage*. <http://www.kognimobil.org>.
- [2] *Urban Challenge Rules*. DARPA, Oktober 2007.
- [3] BORGEFORS, GUNILLA: *Distance transformations in digital images*. Computer Vision, Graphics, and Image Processing, 34(3):344–371, 1986.
- [4] GINDELE, TOBIAS, DANIEL JAGZENT, BEN PITZER und RÜDIGER DILLMANN: *Design of the planner of Team AnnieWAY's autonomous vehicle used in the DARPA Urban Challenge 2007*. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Die Niederlande, 2008.

- [5] HAREL, DAVID: *Statecharts: A Visual Formalism for Complex Systems*. Science of Computer Programming, 8(3):231–274, June 1987.
- [6] HWANG, YONG K. und NARENDRA AHUJA: *Gross motion planning - a survey*. ACM Computing Surveys, 24(3):219–291, 1992.
- [7] KAMMEL, S., J. ZIEGLER, B. PITZER, M. WERLING, T. GINDELE, D. JAGZENT, J. SCHRÖDER, M. THUY, M. GOEBL, F. v. HUNDELSHAUSEN, O. PINK, C. FRESE und C. STILLER: *Team AnnieWAY's Autonomous System*. International Journal of Field Robotics Research, 2008.
- [8] KAMMEL, SÖREN und BENJAMIN PITZER: *Lidar-based lane marker detection and mapping*. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Die Niederlande, 2008.
- [9] KAVRAKI, L.: *Computation of Configuration-Space Obstacles Using the Fast Fourier Transform*. IEEE Transactions on Robotics and Automation, 11(3):408–413, 1995.
- [10] LI, H. und A. M. VOSSEPOEL: *Generation of the euclidean skeleton from the vector distance map by a bisector decision rule*. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seite 66, Washington, DC, USA, 1998. IEEE Computer Society.
- [11] REEDS, J.A. und R.A. SHEPP: *Optimal paths for a car that goes both forward and backward*. Pacific Journal of Mathematics, 145(2):144–154, 1991.
- [12] ROSENFELD, A. und J. L. PFALTZ: *Sequential operations in digital picture processing*. The Journal of the ACM, 13:471–496, Oktober 1968.
- [13] ŠWESTKA, P. und M.H. OVERMARS: *Motion planning for car-like robots using a probabilistic learning approach*. The International Journal of Robotics Research, 16(2):119–143, April 1997.
- [14] WERLING, MORITZ und LUTZ GRÖLL: *Low-level controllers realizing high-level decisions in an autonomous vehicle*. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Die Niederlande, 2008.
- [15] ZIEGLER, JULIUS, MORITZ WERLING und JOACHIM SCHRÖDER: *Navigating car-like vehicles in unstructured environment*. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Die Niederlande, 2008.