

Vollautomatisiertes Valet Parking: Funktions- und Planungsarchitektur

Martin Friedl* Adrian Hupka† Georg Tanzmeister‡

Zusammenfassung: Das vollautomatisierte BMW Valet Parking System navigiert das Fahrzeug ohne Fahrer durch ein Parkhaus und führt selbständig Ein- sowie Ausparkmanöver durch. Mit einer Smartwatch lässt sich die auf der Consumer Electronics Show 2015 in Las Vegas gezeigte Funktion aktivieren, woraufhin das Fahrzeug selbständig durch das Parkhaus fährt und den Fahrer abholt. Eine besondere Herausforderung lag in der Entwicklung einer hierarchischen Planungsarchitektur, die das Fahrzeug robust, sicher und komfortabel durch beliebig strukturierte Umgebungen navigieren sowie komplexe Ein- und Ausparkmanöver ausführen kann. Dieser Beitrag stellt das Gesamtsystem vor und setzt den Fokus auf die entwickelte Planungsarchitektur.

Schlüsselwörter: Autonome Fahrzeuge, Kostenkarten, Pfadplanung.

1 Einleitung

Parkvorgänge zählen zu den Fahrsituationen, die von vielen Personen als herausfordernd und unangenehm angesehen werden, was entsprechend seitens der Automobilhersteller zu einem großen Angebot an Parkassistenzsystemen geführt hat. Bekannte Systeme fokussieren hierbei auf das Ausführen des eigentlichen Parkvorgangs in Längs- und Querparklücken, wobei in neuen Ausprägungen der Fahrer das Fahrzeug sogar verlassen und den Parkvorgang von außen überwachen kann. Eine nächste Entwicklungsstufe stellt das vollautomatisierte BMW Valet Parking System dar, welches den gesamten Parkvorgang in Parkhäusern abdeckt. Dieser Beitrag gibt einen Überblick über das Gesamtsystem und beschreibt detailliert die entwickelte generische Planungsarchitektur.

2 Funktionsarchitektur

Dieser Abschnitt gibt einen Überblick über die Funktionsarchitektur bevor in den folgenden Abschnitten auf die einzelnen entwickelten Ebenen der Planungshierarchie genauer eingegangen wird.

Abbildung 1 zeigt die verschiedenen Teilfunktionalitäten und deren Zusammenspiel. Ein zentrales Element des Systems bildet das Umfeldmodell (oberer Block), dessen Datenbasis sowohl a-priori Wissen als auch sensorbasierte Informationen umfasst. Die spezifisch

*BMW Group Forschung und Technik, Hanauer Str. 46, 80992 München, Martin.MF.Friedl@bmw.de.

†BMW Group, Max-Diamand-Straße 25, 80937 München, Adrian.Hupka@bmw.de.

‡BMW Group Forschung und Technik, Hanauer Str. 46, 80992 München, Georg.Tanzmeister@bmw.de.

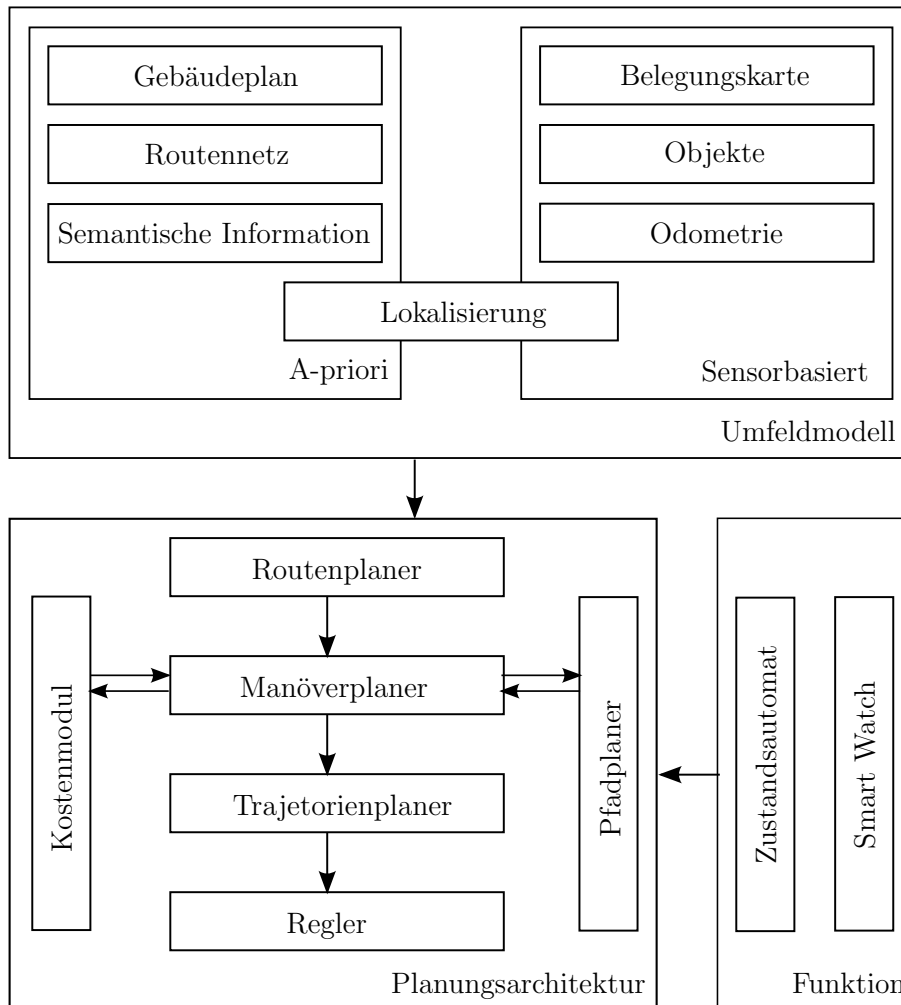


Abbildung 1: Funktionsüberblick.

entwickelte Kartendarstellung enthält neben den Referenzelementen des Parkhauses auch ein Routennetz und semantische Informationen, wie zum Beispiel Parkplätze und Fußwege. Auf Basis von Sensordaten aus Laserscannern und Ultraschallsensoren sowie eines zentralen Odometriemoduls werden Belegungskarten und Objektlisten erstellt. Eine Aktualisierung dieser Informationen erfolgt getriggert durch neue Messdaten der Sensoren.

Die Verknüpfung zwischen den beiden Elementen des Umfeldmodells erfolgt durch die Lokalisierung. Der eingesetzte Positionsschätzer ist in Form eines Kalman-Filter umgesetzt und nutzt eine effiziente Grid Matching Implementierung für das Messupdate, um ohne GPS Information auszukommen, welche in Parkhäusern häufig nicht verfügbar ist.

Die Funktion (Block rechts unten) wird durch eine Smartwatch getriggert, welche über einen Server mit dem Fahrzeug kommuniziert. Der Funktionsablauf ist als Zustandsautomat implementiert; neben Initialisierungszuständen umfasst der Zustandsautomat auf oberster Ebene zwei Fahrzustände für die Suche nach einem Parkplatz und für die Fahrt zum Ausgang des Gebäudes.

Die Planungsarchitektur (Block links unten) baut auf den Informationen des Umfeldmodells auf und wird durch den Funktionsblock angesteuert. In folgendem Kapitel erfolgt eine Beschreibung der Bestandteile der Planungsarchitektur.

3 Planungsarchitektur

Zuerst wird in diesem Abschnitt kurz auf bestehende Ansätze eingegangen (Abschnitt 3.1) und die entwickelte Planungshierarchie (Abschnitt 3.2) vorgestellt. Danach werden die einzelne Planungsebenen sowie das Kostenmodul (Abschnitte 3.3–3.6) näher erläutert.

3.1 Stand der Technik

Die eingesetzten Planungsarchitekturen für automatisierte Fahrfunktionen lassen sich in zwei Klassen einteilen: Planungsarchitekturen für strukturierte und unstrukturierte Umgebungen. Planungsarchitekturen für strukturierte Umgebungen basieren oftmals auf einer Karte, aus der die Verläufe der Fahrstreifenmitten extrahiert werden können. Ist eine solche Referenzbahn gegeben und geht man von keinen großen Abweichungen von dieser aus, lässt sich die Planung auf eine lokale Optimierung einer Bahn bzw. Trajektorie unter Berücksichtigung von Informationen des Umfeldmodells reduzieren [1]. Planungsarchitekturen für unstrukturierte Umgebung kommen dann zum Einsatz, wenn sich das Problem nicht als lokale Optimierung einer Referenzbahn beschreiben lässt. Typische Anwendungsfälle sind Parkflächen oder blockierte Straßenverläufe. Typischerweise kommt eine Graphensuche für die Planung zum Einsatz, womit eine globale Optimierung im Zustandsraum erreicht werden kann [3]. Besonders deutlich wird die Aufteilung der beiden Ansätze bei Betrachtung der Planungsarchitekturen der Teilnehmer der DARPA Urban Challenge, da hier sowohl die Fahrt in strukturierter Umgebung als auch auf unstrukturierten Parkflächen zu bewerkstelligen war. Die demonstrierten Lösungen zeichnen sich dadurch aus, dass beide Typen von Planungsarchitekturen implementiert sind und situationsadaptiv umgeschaltet wird [9, 12].

3.2 Planungshierarchie

Die Anforderung an die Entwicklung des BMW Valet Parking Systems war einerseits, dass mehrzügige Manöver in komplexer, unstrukturierter Umgebung, die stark von einer Referenzbahn abweichen, möglich sein müssen und andererseits sollten alle geltenden Verkehrsregeln berücksichtigt werden, wie z.B. das Fahren auf der richtigen Straßenseite. Das Ziel einer Fahrt liegt zudem häufig in einem Bereich der Umgebung, über den zum Startzeitpunkt keine Informationen mithilfe der Sensorik gewonnen werden können. Das Ausplanen eines kompletten Pfades zum Ziel nur auf Basis von a-priori verfügbaren Informationen (abgespeicherte Kartendaten) ist nicht sinnvoll, weil von einer abweichenden Umgebung ausgegangen werden muss, sobald der Zielpunkt im Sichtbereich der Sensorik ist. Das Anwenden einer Graphensuche für das Finden eines Pfades von Start- zu Zielpunkt ist daher nicht ausreichend für eine hochwertige autonome Fahrweise. Die Graphensuche zum Ausplanen von Pfaden gliedert sich als ein Instrument der Fahrstrategie in eine Planungsarchitektur ein.

Der Begriff Fahrstrategie beschreibt die Herangehensweise ein gegebenes übergeordnetes Ziel unter Berücksichtigung von Randbedingungen und auftretenden Ereignissen im Verlauf der Fahrt zu erreichen. Eine Fahrstrategie setzt sich aus einer Menge von abstrakt formulierten Regeln zusammen. Diese reichen von sehr allgemeinen Bestimmungen (Rechtsfahrgebot) bis hin zu Regeln, die speziell das Fahren in Parkhäusern betreffen

können. Die Umsetzung einer Fahrstrategie erfolgt durch eine geeignete Planungsarchitektur. Für ein komfortables autonomes Fahren ist ein umfangreiches und vielfältiges Regelset notwendig, was die Komplexität einer solchen Planungsarchitektur erhöht. Um dem zu begegnen und Flexibilität für veränderte Fahrstrategien zu gewährleisten, wurde die Planungsarchitektur hierarchisch aufgebaut. Die Aufgabe des Erreichens eines Fahrtziels wird in Teilaufgaben zerlegt, die anschließend von den einzelnen Modulen getrennt bearbeitet werden. Abbildung 1 enthält eine Darstellung der hierarchischen Planungsarchitektur und wird nachfolgend erläutert. Ist das autonom fahrende Fahrzeug Teilnehmer im Straßenverkehr, bewegt es sich in einer strukturierten Umgebung. Dabei existiert zu einem gewissen Grad Vorwissen, das bei der Wahl der Fahrstrategie berücksichtigt wird. Dazu gehören hochaufgelöste Belegungskarten und Informationen über Infrastrukturelemente (u. a. Fahrstreifen, Kreuzungspunkte, Fußwege, Verkehrssignale). Ausgehend von einem gegebenen Routennetz bestimmt der *Routenplaner* (Abschnitt 3.3) eine passenden Route zum Ziel. Ein Routenabschnitt ist eine zusammenfassende Beschreibung aller Strukturelemente eines Streckenabschnitts und eignet sich daher nicht als Eingang für die Längs- und Querführung. Die Fahrstrategie führt das autonom fahrende Fahrzeug im Routennetz entlang von Fahrstreifenmitten zum Zielpunkt. Durch Hinzunahme der über die Umfeldsensorik gewonnenen Informationen werden während der Fahrt fortwährend durch den *Manöverplaner* (Abschnitt 3.4) einzelne Manöver zum Bewältigen von Teilabschnitten der Route geplant. Dabei müssen auftretende Ereignisse berücksichtigt und die in Fahrstreifenmitten liegenden Sollziele gegebenenfalls korrigiert werden. Ausgehend von Start- und Zielpose sowie einer Kostenkarte, die in einem *Kostenmodul* (Abschnitt 3.5) erstellt und in den Konfigurationsraum transformiert wird, bestimmt das Modul des *Pfadplaners* (Abschnitt 3.6) einen fahrbaren Weg. Das Pfadplanungsmodul ist nicht als eigenständige Hierarchiestufe eingegliedert, weil es als Instrument zum Ausplanen von Fahrmanövern dient und damit als Teil des Manöverplaners betrachtet werden kann. Eine lokale Optimierung unter Berücksichtigung von Optimierungskriterien höherer Ordnung erfolgt im *Trajektorienplaner* [4]. Da das Trajektorienplanungs-Modul mit einer deutlich höheren Frequenz als Manöver- und Pfadplaner arbeitet, kann schnell auf plötzlich auftretende Hindernisse reagiert werden. Ein *Trajektorienfolgeregler* ermittelt schließlich die Stellgrößen für die Aktuatorik.

3.3 Routenplaner

Für den ersten, groben Planungsschritt wird aus der Menge aller Infrastrukturelemente ein Routennetz generiert. Dieses Netz setzt sich aus Knoten und Kanten zusammen, wobei Kanten Fahrstreifen entsprechen und Knoten Punkte markieren, an denen Fahrstreifen ineinander führen. Den Kanten sind Kosten zugeordnet, anhand derer die optimale Route bestimmt wird. Der Planungsvorgang wird durch die Wahl eines Zielpunktes ausgelöst. Für den im Rahmen der Arbeit behandelten Anwendungsfall sind Ziele Parkflächen oder frei liegende Punkte der Ebene. Der Zielpunkt selbst ist nicht Teil des Routennetzes und muss zu Beginn einem Fahrstreifen zugeordnet werden. Parkflächen werden im Vorfeld explizit Fahrstreifen zugeordnet, da sie ggf. von verschiedenen Fahrstreifen aus angefahren werden können. Ist der Ziel-Fahrstreifen bestimmt, wird die aktuelle Fahrzeugpose in das Routennetz projiziert und anhand der Fahrzeugorientierung einem Fahrstreifen zugeordnet. Ausgehend von Start- und Ziel sowie dem Routennetz mit gewichteten Kanten bestimmt ein A*-Algorithmus die optimale Route.

3.4 Manöverplaner

Die Komponente des Manöverplaners ist das zentrale Element der Planungshierarchie. Um eine qualitativ hochwertige autonome Fahrweise zu erreichen, ist es notwendig ein Höchstmaß an Struktur in der Umwelt des Fahrzeugs zu erkennen, diese entsprechend zu interpretieren und im Planungsprozess zu berücksichtigen. Ein Manöver bezeichnet einen mit einer gewissen Intention geplanten Pfad zwischen zwei gewählten Abschnittszielen der Route, wie zum Beispiel Anhalte- oder Einparkmanöver. Um von einem Abschnittsziel zum folgenden zu kommen, werden mithilfe des Pfadplaners verschiedene Manöver ausgeplant. Die Bestimmung von Abschnittszielen ist Ausgangspunkt für eine darauf aufbauende Fahrstrategie. Diese beinhaltet die Intelligenz zum Aneinanderreihen von Manövern zum Anfahren des Ziels.

Die Planungsroutine des Manöverplaners orientiert sich entlang der Fahrstreifenmitten in den Korridoren des Parkhauses und weicht nur dann davon ab, wenn ein Ereignis oder Hindernis es erfordert. Sämtliche relevante Elemente der Fahrzeugumgebung werden auf die im Routenplaner geplante Route abgebildet. Dadurch entsteht eine eindimensionale Abbildung der Umgebungsstruktur entlang der Route, als *Leading Route* bezeichnet. Auf Basis der Leading Route und Kostenkarten aus verarbeiteten Sensordaten werden fortwährend Abschnittsziele bestimmt, die sich an der Umgebungsstruktur und nicht an einer festen Sampling-Distanz orientieren.

In jedem Zyklus entscheidet der Manöverplaner neu, welches Abschnittsziel als nächstes angesteuert werden soll. Anhand der Leading Route werden fortwährend Abschnittsziel-Alternativen bestimmt. Soll bspw. im nächsten Zyklus die Überquerung eines Kreuzungsbereichs ausgeplant werden, müssen sowohl für den Fall von Gegenverkehr Abschnittsziele bestimmt werden, die ein als unkomfortabel empfundenen Anhalten inmitten der Kreuzung verhindern als auch Abschnittsziele so gewählt werden, dass die Kreuzung in einem Zug überquert werden kann. Für den Fall eines plötzlich auftauchenden Hindernisses muss außerdem stets ein valides Abschnittsziel vorgehalten werden, das es erlaubt, das bewegte Fahrzeug in einen sicheren Zustand zu bringen. Je nachdem, wie ein Abschnittsziel der Leading Route klassifiziert ist, wird das Fahrverhalten an den vorliegenden Streckenabschnitt angepasst. Ein dementsprechend ausgeplantes Manöver wird an den Trajektorienplaner weitergereicht [4].

3.5 Kostenmodul

Üblicherweise reicht es nicht, lediglich kollisionsfreie Pfade bzw. Trajektorien zu generieren, sondern auch solche, die zusätzlich hinsichtlich bestimmter Kostenfunktionen optimal sind. Kosten lassen sich in extrinsische und intrinsische Kosten aufteilen. Intrinsische Kosten bewerten u.a. Länge, Krümmung, Fahrtrichtung und Richtungsänderungen eines Pfades. Sie sind unabhängig von der aktuellen Umgebung und lassen sich oftmals vorberechnen. Extrinsische Kosten hingegen berücksichtigen Umgebungsinformationen sowie die Fahrzeugausdehnung im jeweiligen Zustand eines Pfades. Um einen Pfad oder ein Pfadsegment auf Kollision zu prüfen, muss die gesamte Fahrzeugmaske in jedem Zustand des Pfades auf Kollision mit allen Objekten geprüft werden. Die Objekte in der Umgebung des Fahrzeug, d.h. im Arbeitsraum (work space), können aber auch in den Konfigurationsraum (configuration space) zu Konfigurationsraumobjekten (configuration space obstacles) transformiert werden, so dass die einzelnen Konfigurationen direkt

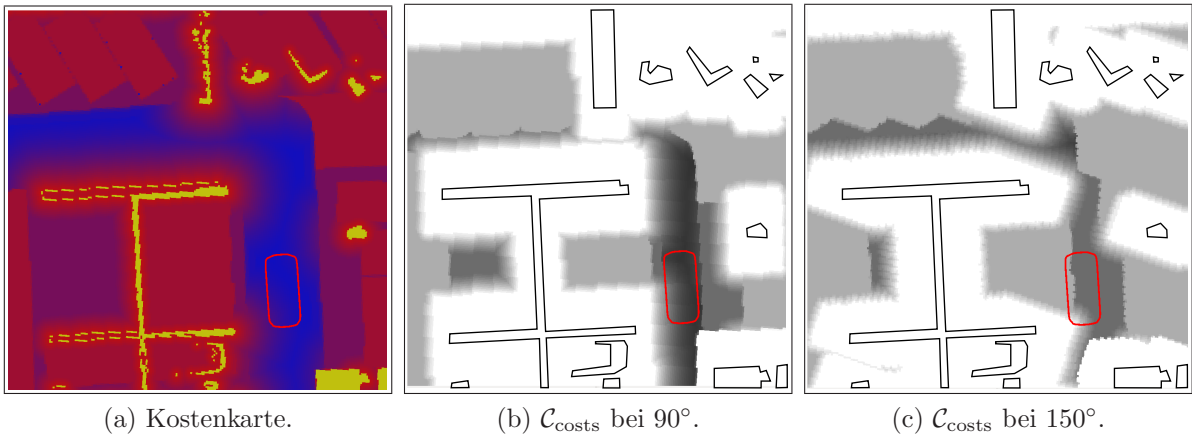


Abbildung 2: Fusionierte Kostenkarte um das Ego-Fahrzeug (rote Kontur). Der Fahrspurverlauf der geplanten Route ist mit einer Kostensenke modelliert. In b) und c) sind zwei Schichten der $\mathcal{C}_{\text{costs}}$ mit verschiedenen Orientierungen dargestellt.

geprüft werden können [7].

Extrinsische Kosten sind oftmals in einer Kostenkarte gegeben. Durch Fusionierung von aktuellen Sensordaten, a-priori verfügbaren Umgebungsinformationen und klassifizierten Ereignissen aus der Fahrzeugumgebung wird eine mehrschichtige Kostenkarte (Layered Costmap) erstellt (siehe Abbildung 2a). Hierbei wird entlang des Verlaufs der aktuellen Route aus dem Routenplaner ein Korridor als Kostensenke modelliert. Ähnlich zu *Konfigurationsraumobjekten (configuration space obstacles)*

$$\mathcal{C}_{\text{obs}} = \{q \in \mathcal{C} \mid S_q \cap \mathcal{O} \neq \emptyset\}, \quad (1)$$

welche die Menge aller Konfigurationen $q = (x, y, \theta)$ aus Position (x, y) und Orientierung θ des Konfigurationsraums \mathcal{C} darstellt, in denen die Fahrzeugmaske S_q eine nichtleere Schnittmenge mit den Objekten \mathcal{O} hat, lassen sich auch Kosten, wie in [11] gezeigt, vom Arbeitsraum in den Konfigurationsraum transformieren. Die in [11] eingeführten *Konfigurationsraumkosten (configuration space costs)*

$$\mathcal{C}_{\text{costs}} : \begin{cases} \mathcal{C} \rightarrow \mathbb{R}_0^+ \\ q = (j, \theta) \mapsto \max(M(s) + S_{j,\theta}(s) \mid s \in S_{j,\theta}) \end{cases} \quad (2)$$

der Kostenkarte M werden unter Verwendung des Maximumoperators mit der Maske $S_q = S_{j,\theta}$ einer Orientierung θ und eines Positionsindex j berechnet. Die $\mathcal{C}_{\text{costs}}$ von zwei fixen Fahrzeugorientierungen sind in Abbildung 2b und 2c gezeigt. Eine solche Schicht entspricht einer Grauwertdilatation, bekannt aus der Bildverarbeitung. Die $\mathcal{C}_{\text{costs}}$ dienen dem Pfadplaner, der wie folgt beschrieben wird, als Berechnungsgrundlage für Kosten und Kollisionen.

3.6 Pfadplaner

Das Modul des Pfadplaners ist dafür zuständig, Pfade für den Übergang von Start- zu Zielzuständen zu berechnen.

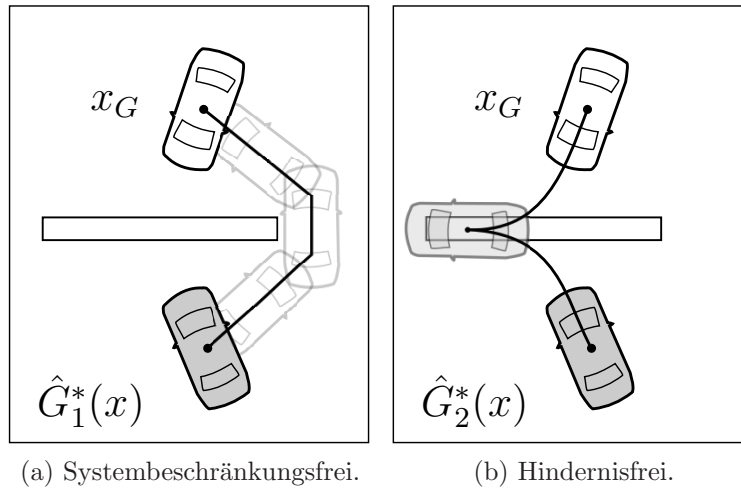


Abbildung 3: Eingesetzte Heuristiken in der A* Suche.

Für den entwickelten Pfadplaner wurde auf verschiedene bestehende Methoden zurückgegriffen, die sich in der Praxis bewährt haben. Der Zustandsraum ist in einem Zustandsgitter (state lattice) diskretisiert (vgl. [10]), wobei ein Zustand im Gitter die Dimensionen Position (x, y) und Orientierung θ umfasst. Das Zustandsgitter bildet also einen Graph, dessen Knoten die erreichbaren Konfigurationen darstellen und dessen Kanten aus Bewegungsprimitiven gebildet sind, die benachbarte Zustände durch fahrbare Segmente verbinden. Die Bewegungsprimitive werden unter Berücksichtigung der nicht-holonomen Eigenschaften des Fahrzeugs als Bézierkurven 3. Grades modelliert und vorberechnet (vgl. [5, 6, 13]), so dass der aus der Kombination der Bewegungsprimitive entstandene Pfad in Position und Orientierung stetig ist.

Bei dem Pfadplanungsalgorithmus handelt es sich um einen Anytime Repairing A*-Algorithmus (ARA*), welcher in [8] vorgestellt wurde. Dieser ermöglicht das Finden einer suboptimalen Lösung innerhalb einer kurzen Zeitspanne, indem der Suche nur schwache Grenzen gesetzt werden. Ist, nachdem eine erste Lösung gefunden wurde, noch eine Restzeit vorhanden, wird der Algorithmus schrittweise stärker begrenzt, so dass nachfolgend gefundene Lösungen einen niedrigeren Grad an Suboptimalität besitzen. Bei ausreichend großer Zeitspanne findet ARA* auch die optimale Lösung für das Suchproblem. ARA* unterscheidet sich von anderen Anytime-Planern dahingehend, dass er für die stärker begrenzten Folge-Iterationsschritte den Graphen nicht erneut durchläuft, sondern den vergangenen Suchverlauf nach Inkonsistenzen durchsucht und diesen schrittweise verbessert.

Der Planer setzt zudem auf einen Multi-Heuristik-Ansatz, wie er in [2] untersucht wurde. Zum einen wird eine Heuristik eingesetzt, welche die nicht-holonomen Eigenschaften des Fahrzeugs ignoriert, die Hindernisse jedoch berücksichtigt (Abbildung 3a). Auf Basis einer binären Belegungskarte wird hierfür mit Hilfe eines Dijkstra-Algorithmus die kürzeste Zellfolge von Start- zu Zielpunkt bestimmt. Außerdem wird eine Heuristik eingesetzt, welche die Restkosten bis zum Ziel dadurch unterschätzt, dass die Hindernisse im Umfeld des Fahrzeugs vernachlässigt werden (Abbildung 3b). Da für die letztere Heuristik keine Umgebungsinformationen erforderlich sind, wird diese vorberechnet und zur Laufzeit als Lookup-Tabelle vorgehalten.

Da immer die Heuristik $\hat{G}(x)$, welche die Restkosten zum Zielzustand am wenigsten

unterschätzt, das Potential besitzt die Suche in größtem Maße zu beschleunigen, werden alle verfügbaren zulässigen Heuristiken wie folgt kombiniert:

$$\hat{G}(x) = \max\left(\hat{G}_1(x), \hat{G}_2(x), \dots, \hat{G}_n(x)\right). \quad (3)$$

Der gekapselte Einsatz des Pfadplaners innerhalb des Manöverplaners führt zu einem globalen Optimierungsverhalten unter Berücksichtigung strukturierender Umgebungsinformationen wie in folgenden Ergebnissen ersichtlich wird.

4 Ergebnisse

Abbildung 4 zeigt verschiedene Verarbeitungsschritte in einer exemplarischen Situation. Innerhalb des Routennetzes (magenta) wird eine Route (orange) vom Eigenfahrzeug (rote Kontur) zu einem Zielparkplatz (grün) geplant (Abbildung 4a). Neben dem Routennetz sind Referenzelemente (Wände und Säulen) und Parkplätze visualisiert. Das Lokalisierungsmodul führt ein Matching der Messwerte mit Referenzelementen der Karte durch (Abbildung 4b). Neben statischen Elementen (Wände, Säulen) werden auch quasistatische Informationen in Form von Parkflächen berücksichtigt. Das Matching erfolgt in einer lokalen Umgebung (blaues Quadrat). Die Matching-Güte ist farbcodiert; es zeigt sich ein schmaler Peak im Bereich der Hinterachse des Fahrzeugs. Die statischen sensorbasierten Umgebungsinformationen werden in einem Occupancy Grid (Abbildung 4c) repräsentiert. Neben den sensorbasierten Informationen wird in der Kostenkarte (Abbildung 4d) das Routennetz, die eigene Route und Parkflächen berücksichtigt. In Bereichen, die für das Fahrzeug nicht erfassbar sind (graue = unknown Bereiche im Grid) werden Informationen aus der a-priori Karte genutzt um Belegung abzuleiten.

Die Situation in Abbildung 4 ist dadurch charakterisiert, dass das Routennetz großflächig blockiert ist; eine lokale Optimierung auf Basis der Route wird hier keine Lösung finden. Der in diesem Beitrag vorgestellte Planer ermittelt in diesem Fall einen Weg, der das Hindernis neben einer Säule über Parkflächen großflächig umfährt. In bestimmten Fällen kann solch eine Umfahrung auch ein mehrzügiges Fahrmanöver erfordern. Nach der Umfahrung des Hindernisses orientiert sich der Pfadplaner am Routennetz, befolgt das Rechtsfahrgebot, vermeidet die Überquerung von Parkfläche und hält Abstand zu Hindernissen.

5 Zusammenfassung

Mit dem vorgestellten System gelang die Durchführung autonomer Parkvorgänge in unterschiedlichen Parkhäusern. Bei der CES 2015 wurde das System der Öffentlichkeit unter dem Begriff „BMW Remote Valet Parking Assistant“ vorgestellt und konnte intensiv getestet werden. Hierfür wurde eine Planungsarchitektur eingesetzt, die sowohl die Fahrt durch das Parkhaus wie auch den Parkvorgang selbst adressiert. Die Architektur ist durch einen hierarchischen Ansatz charakterisiert, der es erlaubt komplexe mehrzügige Manöver mittels Pfad-/Manöver-Planer zu planen und zugleich durch die Krümmungsstetigkeit der ermittelten Trajektorien ein komfortables Verhalten zeigt.

Zum Erreichen einer hochwertigen Fahrweise, die in hohem Maße der menschlichen Erwartung entspricht, sind strukturierende Umgebungsinformationen (Richtungsfahrbahn,

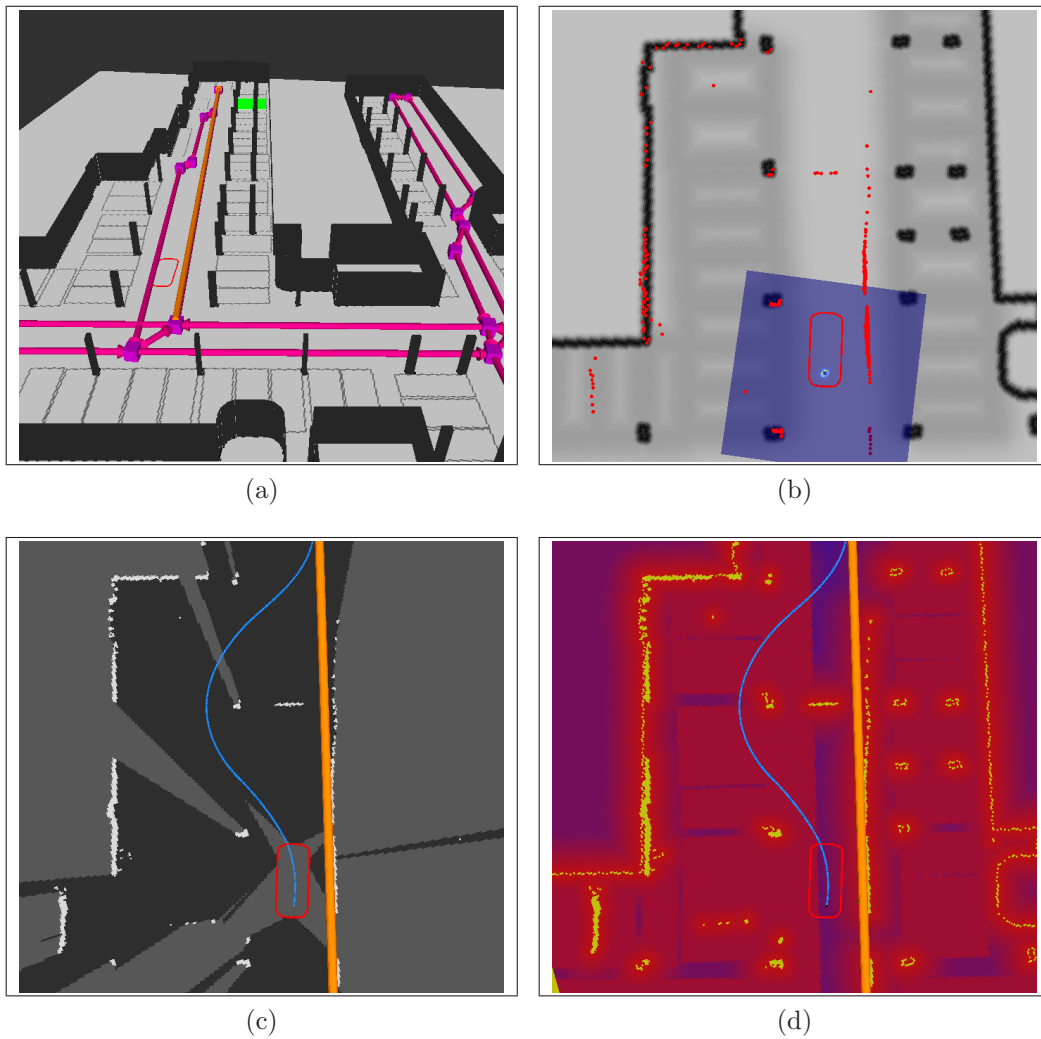


Abbildung 4: (a) A-priori Information mit Routennetz und geplanter Route. (b) Lokalisierung. (c) Geplanter Pfad ausgehend von geplanter Route. (d) Kostenkarte.

Kreuzungen, Parkflächen, Fußwege,...) essentiell. Diese Informationen werden aus a-priori Karteninformationen und sensorbasiertem Umfeldmodell gewonnen. Der Manöverplaner übersetzt die Informationen in Abschnittsziele und Kostenkarten, so dass diese durch den Pfadplaner berücksichtigt werden. Zur effizienten Kostenermittlung in der Pfadplanung werden die Kosten im Konfigurationsraum repräsentiert.

Literatur

- [1] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen. Experience, results and lessons learned from automated driving on germany's highways. *IEEE Intelligent Transportation Systems Magazine*, 7(1):42–57, Spring 2015.
- [2] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev. Multi-heuristic A*. In *Proc. of Robotics: Science and Systems*, 2014.
- [3] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *International Journal of Robotics Research*, 29(5):485–501, Apr. 2010.
- [4] B. Gutjahr and M. Werling. Optimale Fahrzeugquerführung mittels linearer, zeitvarianter MPC. In *Workshop Fahrerassistenzsysteme*, Walting im Altmühltal, 2015.
- [5] S. Landrault, P. Lucidarme, and N. Delanoue. Path planning optimization based on Bézier curves through open-doors way point. In *Proc. of the International Conference on Informatics in Control, Automation and Robotics*, pages 156–161, 2013.
- [6] B. Lau, C. Sprunk, and W. Burgard. Kinodynamic motion planning for mobile robots using splines. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2427–2433, 2009.
- [7] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, UK, 2006.
- [8] M. Likhachev, G. J. Gordon, and S. Thrun. ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems*, 2003.
- [9] M. Montemerlo et al. Junior: The Stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, Sept. 2008.
- [10] M. Pivtoraiko, R. A. Knepper, and A. Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, Mar. 2009.
- [11] G. Tanzmeister, M. Friedl, D. Wollherr, and M. Buss. Efficient evaluation of collisions and costs on grid maps for autonomous vehicle motion planning. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2249–2260, Oct. 2014.
- [12] C. Urmson et al. Tartan racing: A multi-modal approach to the DARPA urban challenge. Technical report, Carnegie Mellon University, Robotics Institute, Apr. 2007.
- [13] F. Zhou, B. Song, and G. Tian. Bézier curve based smooth path planning for mobile robot. *Journal of Information and Computational Science*, 8(12):2441–2450, Dec. 2011.