



Bachelorarbeit am Fachbereich Mathematik und Informatik  
in der Arbeitsgruppe Intelligente Systeme und Robotik

# Fahrspurmodellierung mit Punktvalidierung bei autonomen Modellfahrzeugen

Till-Julius Krüger  
Matrikelnummer: 4671123  
[till.krueger@fu-berlin.de](mailto:till.krueger@fu-berlin.de)

Gutachter:  
Prof. Dr. Raúl Rojas  
Prof. Dr. Daniel Göhring

28. September 2015

---

### **Eidesstattliche Erklärung**

Ich versichere hiermit an Eides statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, 28.September 2015

---

Till-Julius Krüger

---

## Zusammenfassung

Autonomes Fahren ist in der heutigen Zeit ein hochaktuelles Thema. Die TU Braunschweig veranstaltet jedes Jahr den Carolo-Cup, ein Wettbewerb bei dem Studenten autonome Modellfahrzeuge im Maßstab 1 : 10 bauen und programmieren. Um an den verschiedenen Disziplinen des Carolo-Cup teilnehmen zu können, muss als Grundfunktionalität die Fahrspur vom Fahrzeug gehalten werden können. Diese Aufgabe besteht aus drei Teilaufgaben: Als Erstes ist die Erkennung und Modellierung von Linien durchzuführen. Danach erfolgt die Unterscheidung zwischen Fahrbahnmarkierungen, Fehlern und anderen Straßenmarkierungen. Im Anschluss daran muss ein Pfad bestimmt werden, der vom Fahrzeug abgefahren wird. Im Rahmen dieser Arbeit wurde eine Fahrspurerkennung mit SequentialRANSAC und deren Bewertung durch Punktvalidierung implementiert.

---

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Carolo-Cup . . . . .	1
1.3 Berlin United Racing Team . . . . .	2
1.4 Problemstellung und Zielsetzung . . . . .	3
<b>2 Grundlagen</b>	<b>5</b>
2.1 Fahrstrecke im Carolo-Cup . . . . .	5
2.2 Kantenerkennung . . . . .	6
2.3 Newtonpolynom . . . . .	8
2.4 RANSAC . . . . .	9
2.5 Lösungsansätze . . . . .	10
2.5.1 Fahrspurerkennung . . . . .	11
2.5.2 Fahrspurbewertung . . . . .	12
2.6 Wahl des Ansatzes . . . . .	13
<b>3 Fahrspurmodellierung mit Punktvalidierung</b>	<b>15</b>
3.1 Fahrspurerkennung mit SequentialRANSAC . . . . .	15
3.2 Punktvalidierung . . . . .	17
3.2.1 Mathematische Grundlagen . . . . .	19
3.2.2 Abstandsbestimmung . . . . .	20
3.3 Positionstracking . . . . .	21
3.4 Laufzeitverbesserung . . . . .	23
3.5 Fahrspurverfolgung . . . . .	24
<b>4 Evaluation</b>	<b>26</b>
4.1 Bewertungssystem . . . . .	26
4.2 Vorbetrachtung . . . . .	27
4.3 Experiment 1: Funktionalität mit optimalem Fahrpfad . . . . .	27
4.4 Experiment 2: Funktionalität mit fehlerhaftem Fahrverhalten . . . . .	29
4.5 Experiment 3: Funktionalität im Ausnahmefall . . . . .	30
4.6 Experiment 4: Überholmanöver . . . . .	32
4.7 Diskussion . . . . .	33
<b>5 Zusammenfassung und Ausblick</b>	<b>35</b>

## 1 Einleitung

Selbstständige Fahrzeuge ohne Fahrer sind in der heutigen Zeit keine futuristische Vision mehr. Autonomes Fahren ist ein hochaktuelles Thema. In der Forschung und Entwicklung von autonomen Fahrzeugen ist ein Wettstreit entbrannt, in den nicht nur Automobilunternehmen wie Audi oder BMW eingetreten sind[1], sondern auch Unternehmen wie Google[2] und Universitäten wie die Freie Universität Berlin[3].

Die Herausforderungen an die Fahrzeuge sind enorm: Einhaltung der Fahrspur, Lokalisierung zur Umgebung, Erkennung von Hindernissen und Vermeidung von Unfällen sind nur einige auftretende Schwierigkeiten im Straßenverkehr.

Um Studenten an diese Problematiken heranzuführen, wird der Carolo-Cup an der Technischen Universität Braunschweig veranstaltet. Dort werden anhand von selbst gebauten Modellfahrzeugen studentische Lösungsansätze der Probleme entwickelt. In verschiedenen Disziplinen werden die Ansätze auf einer straßenähnlichen Strecke erprobt.

### 1.1 Motivation

Das Einhalten einer Fahrspur ist eine unerlässliche Aufgabe eines intelligenten Fahrzeugs[4]. Man kann diese Aufgabe als Basis des gesamten autonomen Fahrens verstehen. Die meisten komplexeren Aufgaben könnten ohne eine funktionierende Fahrspurhaltung nicht gelöst werden. Für das studentische Team der Freien Universität Berlin wurde für die Verwendung im Carolo-Cup bereits ein Ansatz entwickelt[5]. Der Ansatz selbst und die damit verbundenen Probleme sind in Abschnitt 2.4 beschrieben.

In dieser Arbeit sollen nun die Schwächen des vorherigen Ansatzes verringert und die Funktionalität verbessert werden.

### 1.2 Carolo-Cup

Der Carolo-Cup ist ein Wettbewerb für autonomes Fahren an der TU Braunschweig, der seit 2008 jährlich stattfindet. Die Aufgabe an die ausschließlich studentischen Teams besteht in einem Auftrag eines fiktiven Automobilherstellers. Dieser Auftrag beinhaltet die Erstellung eines autonomen Modellfahrzeugs im Maßstab von 1 : 10. Die Modellfahrzeuge werden am Tag des Wettkampfs nicht nur hinsichtlich ihrer Funktionalität und der damit verbundenen Straßentauglichkeit beurteilt. Hinzu kommen noch die Faktoren

der Energieeffizienz und Kostenminimierung, um so auch die Wirtschaftlichkeit in die Bewertung einfließen zu lassen.

Generell gilt es während des Wettbewerbs verschiedene Disziplinen zu bestehen. Die auf den hier gewählten Ansatz bezogenen Disziplinen werden im Folgenden kurz erläutert[6]:

- **Paralleles Einparken:** Fahren auf einer Straße bis zum Erkennen einer zur Straße parallelen, geeigneten Parklücke und anschließendes Einparken
- **Rundstrecke ohne Hindernisse:** Fahren auf einer Straße in einem Zeitfenster ohne Beachtung von Vorfahrtsregelungen. Teile der Fahrbahnmarkierungen können verdeckt sein. Es gibt keine Hindernisse. Die zurückgelegte Strecke wird durch Strafen bei Fehlverhalten verringert.
- **Rundstrecke mit Hindernissen:** Ähnlich wie bei Rundstrecke ohne Hindernisse, allerdings muss nun die Vorfahrtsregelung beachtet werden. Zusätzlich werden auf und neben der Strecke Hindernisse platziert, die umfahren werden müssen.

Bei diesen Aufgaben ist die Einhaltung der Fahrspur von großer Bedeutung, um die Hauptaufgaben der jeweiligen Disziplinen meistern zu können.

### 1.3 Berlin United Racing Team

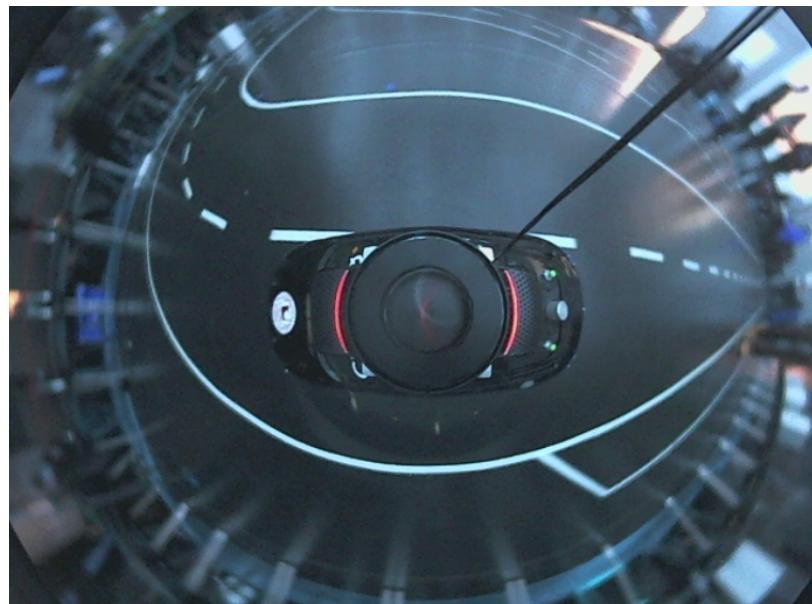


**Abb. 1:** Modellfahrzeug Abel des Berlin United Racing Teams. Foto: Severin Junker

Die Freie Universität Berlin nimmt seit 2012 mit dem Berlin United Racing Team am Carolo-Cup teil. Zum gegenwärtigen Zeitpunkt wird das Fahrzeug *Abel* verwendet, das in Abbildung 1 zu sehen ist. Abel war ursprünglich ein Modellbausatz, zu dem noch

benötigte Komponenten wie die Kamera und der ARM-Computer *OdroidX2* hinzugefügt wurden.

Der Hauptsensor zur Wahrnehmung der Umwelt ist eine Kamera mit einem Parabolspiegel. Diese Kamera hat eine omnidirektionale Sicht, es ist daher möglich durch den Spiegel den gesamten Bereich in einem Radius von etwa 1.5 Metern um das Fahrzeug einzusehen. In Abbildung 2 ist ein typisches Bild aus Sicht des Fahrzeugs zu sehen.



**Abb. 2:** Omnidirektionales Kamerabild. Das Fahrzeug befindet sich in der rechten Fahrspur und hat den Parkstreifen gerade passiert

Die Aufnahme muss zunächst an der oberen Bildkante gespiegelt werden, damit das Bild der Wirklichkeit entspricht. Auffällig ist bei dieser Art von Kamera, dass die Verzerrung der Perspektive vom Zentrum des Bildes zum Rand stark zunimmt.

#### 1.4 Problemstellung und Zielsetzung

Das Halten einer Fahrspur<sup>1</sup> kann in drei Teilaufgaben eingeteilt werden. Als Erstes müssen Fahrbahnmarkierungen<sup>2</sup> erkannt und modelliert werden. Danach ist allerdings nicht bekannt, ob es sich bei den gefundenen Modellen wirklich um Fahrspurmarkierungen

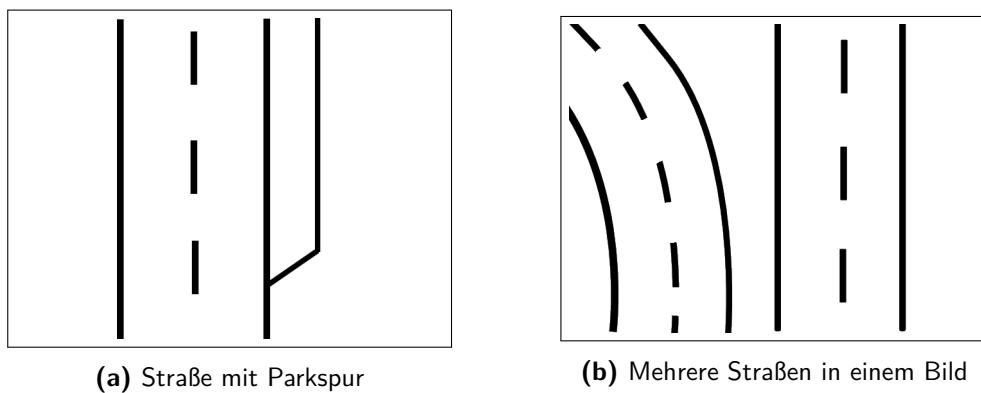
---

<sup>1</sup>Eine Fahrspur besteht hierbei aus begrenzenden Fahrspurmarkierungen und der Strecke dazwischen.

<sup>2</sup>Fahrbahnmarkierung wird als äquivalenter Begriff für Fahrspurmarkierung verwendet. Eine Straße hat immer drei Fahrspurmarkierungen, die zu zwei Fahrspuren gehören.

handelt. Daher muss in der Teilaufgabe der Validierung zwischen realen Fahrbahnmarkierungen, Parkspuren und falschen Modellen differenziert werden. Im letzten Schritt muss ein Pfad zur Fahrspurhaltung bestimmt werden. In dieser Arbeit wird ein Ansatz zur Erkennung und Validierung von Fahrspurmarkierungsmodellen entwickelt.

Dabei stellt die Funktionalität sowohl im allgemeinen Straßenverkehr, als auch in Ausnahmesituationen die entscheidende Anforderung dar. Das Fahrzeug sollte die Fahrspur so selten wie möglich verlassen. Dies gilt insbesondere auch in Situationen, in denen mehrere Markierungen zu sehen sind (siehe Abbildung 3). Auch beim Eintreten von Ausnahmesituationen, wie dem Fehlen von Fahrbahnmarkierungen, sollte eine Einhaltung der Fahrspuren gewährleistet sein. Im Falle des Verlassens einer Fahrspur und auch der Straße sollte zudem eine Rückkehr geplant werden. Da bei der Erfüllung dieser Aufgabe nur ein geringes Maß an Rechenkapazität zur Verfügung steht, sollte die Anwendung in Echtzeit auf dem Fahrzeug ausführbar sein.



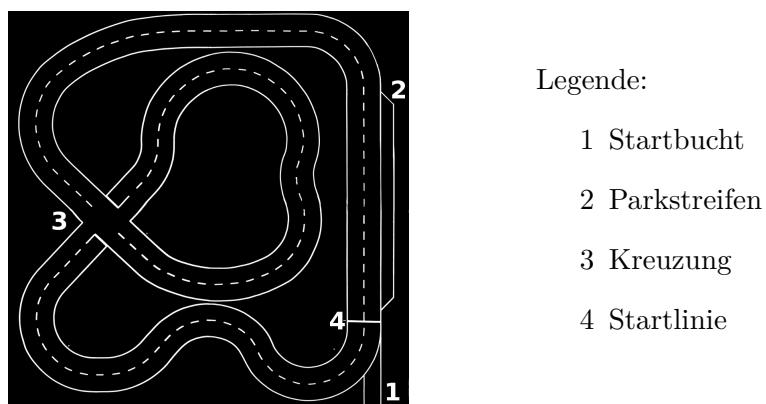
**Abb. 3:** Beispielhafte Streckenabschnitte mit zusätzlichen Markierungen

## 2 Grundlagen

In diesem Kapitel werden einige Grundlagen vorgestellt, die im Verlauf der Arbeit benötigt werden. Dabei wird zunächst auf die Besonderheiten der Fahrstrecke im Carolo-Cup eingegangen, die die physikalische Grundlage darstellt. Dies wird für die Validierung benötigt. Im Anschluss daran wird die Ausgangssituation auf der Seite des Algorithmus in Form der Kantenerkennung vorgestellt. Da zur Modellierung der Straßenmarkierungen Polynome verwendet werden sollen, erfolgt eine kurze Einführung in Newtonpolynome und Interpolation. Danach wird ein wichtiger Algorithmus zur Modellauswahl aus der Computer Vision vorgestellt: *RANdom SAmple Consensus* (RANSAC). Zusätzlich wird auf die vorangegangene Arbeit eingegangen und deren Probleme erläutert. Als Letztes werden Lösungsansätze diskutiert und ein Ansatz ausgewählt.

### 2.1 Fahrstrecke im Carolo-Cup

Auf der Fahrstrecke des Carolo-Cup's wird die Funktionalität des Fahrzeugs getestet. In ihrem Verlauf finden sich Geraden, Kurven und Kreuzungen, auf denen das Fahrzeug bestehen muss. Jedes Jahr kann sich der Straßenverlauf ändern. Abbildung 4 zeigt einen möglichen Straßenverlauf aus der Vogelperspektive.



**Abb. 4:** Karte der Fahrstrecke vom Carolo-Cup im Jahr 2014/2015

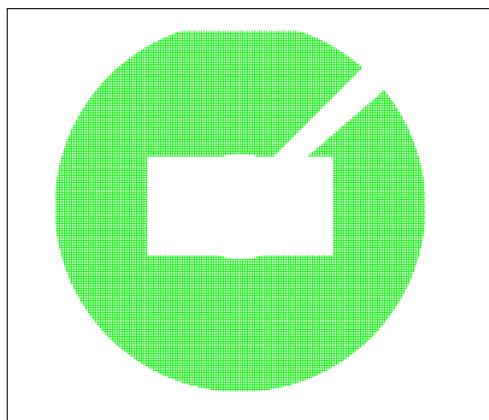
Die Fahrstrecke ist einer Straßenumgebung nachempfunden. Um den Teilnehmern die Erkennung zu erleichtern, ist die gesamte Strecke im Vergleich zu einer realen Straße stark vereinfacht. Vorkommende Farben der Straßenumgebung sind Schwarz und Weiß. Die Fahrstrecke ist eben und frei von Verunreinigungen.

Eine weitere Besonderheit im Carolo-Cup ist, dass sämtliche Fahrbahnmarkierungen eine festgelegte Länge und Breite haben. Gleiches gilt auch für den Abstand von Fahrspuren zueinander, sowie dem Abstand der Parkspur zur rechten Fahrbahnmarkierung. Dies ist eine sehr wichtige Information für die in Abschnitt 3 auf Seite 15 vorgestellte Punktvalidierung. Abbildung 23 auf Seite 40 im Anhang veranschaulicht einen Straßenabschnitt mit den Maßen für die Fahrbahnmarkierungen.

## 2.2 Kantenerkennung

Um Fahrspuren erkennen zu können, müssen zunächst Punkte der Fahrbahnmarkierungen aus dem Kamerabild extrahiert werden. Diese Punkte dienen als Ausgangslage für die Modellierung der Fahrspur.

Das Berlin United Racing Team verwendet eine Lösung, die auf der Erkennung von Kanten im Bild basiert[7][8, S.37ff]. Dazu werden zunächst horizontale und vertikale Scanlines über das Bild gelegt. Abbildung 5 zeigt, wie die Scanlines ein Netz auf dem Bild schaffen.

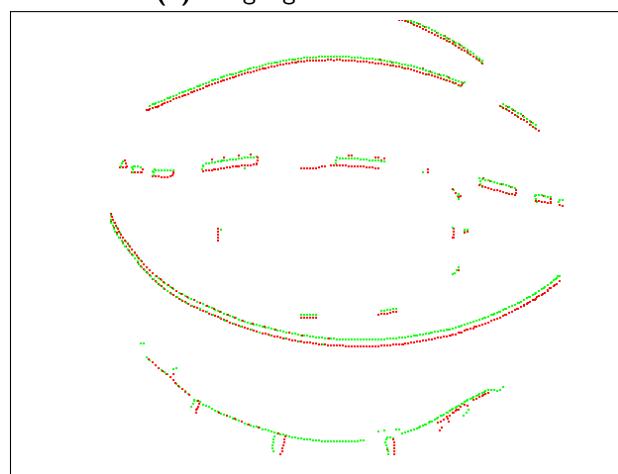


**Abb. 5:** Scanlines zur Kantenerkennung

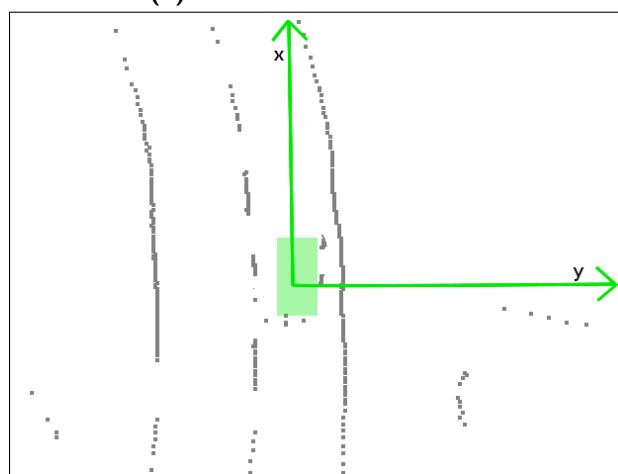
Auf dem Gitter wird nun entlang der Scanlines der Sobel-Operator[9] angewandt. Dadurch werden die Kanten der Weißflächen im Bild herausgearbeitet. Abbildung 6 auf der nächsten Seite zeigt ein Beispiel für dieses Verfahren. Wie dort in Abbildung b zu sehen ist, wird dieser Schritt auf dem Originalbild der Kamera ausgeführt.



(a) Ausgangsbild der Kamera



(b) Gefundene Kanten im Bild



- (c) Extrahierte und projizierte Punkte der Fahrbahnmarkierungen in relativem Koordinatensystem.  
Grüne Elemente dienen nur zur Veranschaulichung

**Abb. 6:** Demonstration des Verfahrens zur Kantenerkennung

Um eine weitere Verarbeitung zu vereinfachen, werden die Punkte in ein relatives Koordinatensystem überführt. Dieses Koordinatensystem ist immer relativ zu einem gewissen Objekt ausgerichtet, in diesem Fall zum Fahrzeug. Dadurch befindet sich das Fahrzeug immer im Ursprung des Koordinatensystems. Außerdem zeigt die Vorderseite des Fahrzeugs in Richtung der  $x$ -Achse. Das Koordinatensystem ist hierbei gedreht, sodass die  $x$ -Achse auf der Vertikalen liegt. Dies wird für die Erstellung der Polynome benötigt.

Als Nächstes werden mit Hilfe einer Entfernungsabschätzung die Mittelpunkte der Fahrbahnmarkierungen ermittelt. Die Mittelpunkte der Straßenmarkierungen werden projiziert und transformiert, um dadurch die Wirklichkeit zu modellieren. Ein Beispiel dafür ist in Abbildung 6c auf der vorherigen Seite zu sehen. Das Fahrzeug und die Achsen des Koordinatensystems sind zur Verdeutlichung in Grün dargestellt. Zusätzlich veranschaulicht die Abbildung auch die Nachteile der Nutzung eines relativen Koordinatensystems, da die gefundenen Fahrspurmarkierungen zum Rand des Bildes stark ausdünnen. Dies ist der Entfernung der Verzerrung des Bildes geschuldet. Als Folge dessen sind die Entfernung im relativen Koordinatensystem allerdings wie in der Realität. Das ist ein großer Vorteil, da nun Abstände zwischen Straßenmarkierungen bestimmt werden können.

### 2.3 Newtonpolynom

Die Beschreibung der Fahrbahnmarkierungen soll durch Polynome erfolgen. Für die Modellierung sollen Polynome zweiten Grades verwendet werden. Der Grund dafür ist, dass die Krümmung der Kurven nicht sehr stark ist[10].

Um ein Polynom zu erstellen, kann eine Interpolation aus Punkten vorgenommen werden. Dabei werden für ein Polynom  $n$ -ten Grades  $n+1$  Wertepaare benötigt. Diese Wertepaare müssen paarweise verschieden sein und werden Stützstellen genannt.

In dieser Arbeit wird zur Interpolation das Newton-Verfahren verwendet, dass ein Polynom mit Hilfe der Newton-Basisfunktion berechnet und die Form

$$\begin{aligned} P(x) &= c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \cdots + c_n(x-x_0) \dots (x-x_{n-1}) \\ &= \sum_{i=0}^n c_i \cdot N_i(x) \end{aligned}$$

besitzt, wobei  $c_i$  für errechnete Koeffizienten und  $x_i$  für Stützstellen stehen[11].

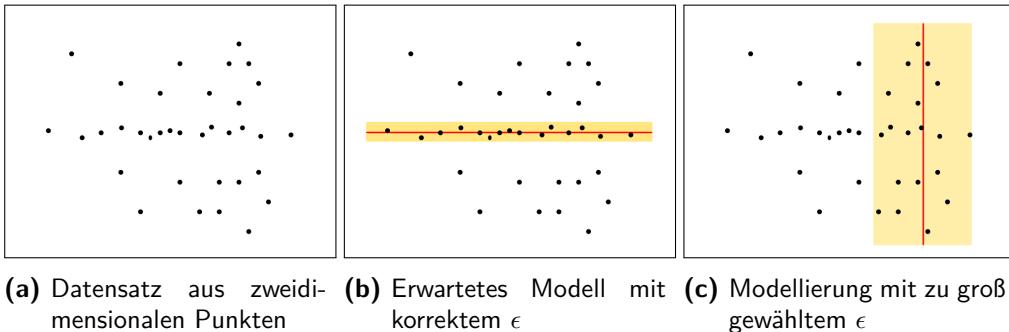
## 2.4 Ransac

Bisher wurde die Ausgangsbasis durch die Kantenerkennung und die Interpolation von Polynomen mit dem Newton-Verfahren erklärt. In diesem Abschnitt wird nun die Verbindung zwischen den beiden Verfahren hergestellt. Da die Straßenmarkierungspunkte bekannt sind und eine Möglichkeit zur Modellierung vorhanden ist, muss nun geklärt werden, wie die Daten der Punkte in Instanzen des Modells überführt werden können. Dabei sollte die Methode robust gegen Ausreißer sein, um falsche Modelle und damit verbundene Fehlinterpretationen der Daten vermeiden zu können.

Ein bekanntes Mittel aus der Computer Vision, das in vielen Bereichen zum Einsatz kommt, ist der RANSAC-Algorithmus. Er basiert auf dem iterativen Bestimmen der Modellinstanz aus zufälligen Teilmengen der Daten mit anschließender Bewertung. Dieser Algorithmus wird im Folgenden vorgestellt. Pseudocode ist in Algorithmus 2 auf Seite 40 im Anhang zu finden.

RANSAC wurde erstmals von Fischler und Bolles[12] vorgestellt, um das *Location Determination Problem* zu lösen. Dabei soll aus einer Reihe von Fotografien eines Objektes die Position des Fotografen erkannt werden. Allerdings lässt sich der RANSAC-Algorithmus sehr universell anwenden[13][14] und kann daher auch für das Auswählen der Polynome, die die Fahrbahnmarkierung beschreiben, genutzt werden.

RANSAC besteht im Wesentlichen aus zwei Schritten, die solange ausgeführt werden, bis entweder eine vorgegebene Anzahl an Iterationen  $N$  erfolgt ist, oder bis eine zufriedenstellende Lösung gefunden wurde. Der erste Schritt besteht aus dem Erstellen der Instanz  $M^*$  des Modells  $M$ . Dies geschieht durch die zufällige Auswahl von  $n$  Datenpunkten aus der Datenmenge  $D$ , aus denen die Instanz erstellt wird. Dabei ist  $n$  gleich der minimalen Anzahl an Punkten, die für die Erstellung von  $M^*$  benötigt werden. Im zweiten Schritt wird  $M^*$  bewertet. Für die Bewertung wird eine Abstandsfunktion verwendet, die für alle Punkte  $P_i$  mit  $i = 1, \dots, |D|$  aus  $D$  überprüft, ob sie durch  $M^*$  abgedeckt werden. Dabei erfolgt die Überprüfung, indem die Funktion feststellt, ob  $P_i$  in einem maximalen Abstand  $\epsilon$  zu  $M^*$  liegt.



**Abb. 7:** Auswirkungen von Schwellwert  $\epsilon$  für die Modellbewertung mit RANSAC. Die rote Linie stellt das Modell dar, während der helle Bereich um die Linie  $\epsilon$  darstellt.

Dadurch wird  $D$  in zwei Mengen aufgeteilt: Auf der einen Seite finden sich Punkte, die in einem Abstand kleiner  $\epsilon$  zu  $M^*$  liegen und  $M^*$  dadurch unterstützen. Diese Punkte  $I$  werden auch *Inlier* genannt. Das Gegenstück sind Punkte aus  $D$  mit einem Abstand größer oder gleich  $\epsilon$ . Da diese Punkte außerhalb des Einflussbereichs von  $M^*$  liegen, werden sie *Outlier* genannt. Eine zufriedenstellende Lösung wird definiert durch einen minimalen, festgelegten Prozentsatz  $\mathcal{T}$  an Inlier im Verhältnis zu  $D$ . Ist  $M_I^*$  größer oder gleich  $\mathcal{T}$ , terminiert der Algorithmus mit  $M^*$ .

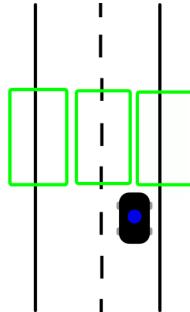
Der Parameter  $\epsilon$  spielt für den Algorithmus eine wichtige Rolle. Wenn dieser Schwellwert zu klein oder zu groß gewählt wird, kann der Algorithmus nicht die optimale Instanz finden, da die Inlier das Modell inkorrekt repräsentieren. Ein Beispiel für einen zu großen Wert ist in Abbildung 7 zu sehen.

## 2.5 Lösungsansätze

Bisher wurde im Berlin United Racing Team die Arbeit von Conrad Läßig verwendet. Läßig stellt in [5] seinen Ansatz zur Spurerkennung mit RANSAC vor. Dort nimmt er die Voraussetzung an, stets die richtige Fahrspur zu befahren. Dadurch wird die Modellierung der Fahrbahnmarkierungen auf bestimmte Bereiche relativ zum Fahrzeug begrenzt, die in Abb. 8 auf der nächsten Seite zu sehen sind. In den Teilbereichen kann dann RANSAC ausgeführt werden. Wenn Fahrspuren im vorherigen Bild gefunden wurden, verengen sich die Teilbereiche um die Fahrspurmarkierungen, um so effizienter suchen zu können.

Diese Voraussetzung stellt jedoch bei Fehlverhalten wie dem kurzzeitigen Verlassen der Fahrspur ein Problem dar, da die Straßenmarkierungen in Regionen relativ zum Fahrzeug erwartet werden und bei Verlust nicht mehr gefunden werden können. Dadurch

kann das Fahrzeug nicht ohne externe Hilfe den Weg in die richtige Fahrspur zurückfinden. Auch ein Überholmanöver zum Passieren von Hindernissen kann nicht dynamisch erreicht werden, da beim Wechsel der Fahrspur die Straßenmarkierungen durch mehrere Suchregionen führen.



**Abb. 8:** Partitionierung der Suchregionen(grün) zur Untersuchung mit RANSAC

Um diese Fehler zu vermeiden, soll die Fahrspurerkennung nicht auf kleine Teilbereiche beschränkt werden, sondern auf der gesamten Fläche vor dem Fahrzeug stattfinden. Dadurch müssen nun allerdings mehrere Instanzen eines Modells gefunden werden. Anschließend ist eine Differenzierung zwischen Fahrbahnmarkierungen und anderen Linien wie Parkspuren notwendig, um die Straßenumgebung korrekt modellieren zu können.

### 2.5.1 Fahrspurerkennung

Die einfachste Lösung zum Finden von mehreren Instanzen eines Modells ist SequentialRANSAC. Dabei wird RANSAC ausgeführt und anschließend die Inlier der gefundenen Instanz aus der Datenmenge entfernt, bis die Datenmenge stark genug reduziert wurde[15][16]. Das Verfahren wird in [17] jedoch als nicht optimal bezeichnet.

Als Verbesserung wird dort das Verfahren MultiRANSAC vorgestellt, das ebenfalls RANSAC verwendet. Es können sowohl mehrere Instanzen eines Modells, als auch von verschiedenen Modellen gefunden werden. Der Hauptunterschied liegt im Aufbau der Modellinstanzen. Anstatt eine Instanz nach der anderen zu erstellen, werden hier alle Instanzen aller Modelle gleichzeitig ermittelt. Dadurch lässt sich dieser Algorithmus leicht parallelisieren. In der qualitativen Ausführung zeigt MultiRANSAC gegenüber SequentialRANSAC mit der Zunahme von Hintergrundrauschen eine wesentlich bessere Modellierung. Allerdings werden auch deutlich mehr Iterationen benötigt und die Anzahl der zu erstellenden

Instanzen muss vorher bekannt sein[18]. Auch bei sich schneidenden Modellinstanzen findet der Algorithmus keine optimale Lösung[19].

Ein weiterer Ansatz zum Erkennen von Modellinstanzen ist die *Randomized Hough Transformation*[20]. Hier wird ein zusätzlicher Parameterraum zum Erkennen der Instanz angelegt. In den Punkten des Parameterraums werden die Parameter von Modellen zusammen mit einer Bewertung für die Häufigkeit ihres Auftretens gespeichert. Iterativ werden zufällige Pixel aus dem Bild entfernt und einem Punkt im Parameterraum zugewiesen. Dabei wird die Bewertung stets verändert. Als Letztes werden aus dem Parameterraum die Maxima extrahiert, die dann die Modelle darstellen. Dieses Verfahren ist ein weit verbreiteter Ansatz, bei dem die geringe Recheneffizienz[19] nicht auf Echtzeitanwendungen ausgelegt ist.

Interessant ist auch die Methode *J-Linkage*[19]. Bei diesem Verfahren wird jeder Punkt der Datenmenge durch eine charakteristische Funktion repräsentiert. Für diese Funktion wird zuerst eine große Anzahl an zufälligen Instanzen aus der Datenmenge erstellt. Anschließend wird für jeden Punkt bestimmt, wie oft er einen Inlier in allen Modellen darstellt. Danach werden die Punkte anhand des Ergebnisses der charakteristischen Funktion mit agglomerativem Clustering gruppiert. Die Ergebnisse sind die Inlier aller Instanzen, aus denen diese dann gebildet werden. J-Linkage zeigt sehr gute Resultate, ist aber laut [21] nicht für Echtzeitanwendungen geeignet.

### 2.5.2 Fahrspurbewertung

Für die Validierung der gefundenen Polynome und Klassifizierung der Position der Fahrspurmarkierung konnten nur wenige Ansätze gefunden werden. Meist wird vorausgesetzt, dass nur eine Straße und alle Fahrspuren im Kamerabild zu sehen sind. Daher beschäftigen sich die meisten Verfahren für PKWs mit der Unterscheidung zwischen Straße und Randbebauung[22][23].

Eine sehr einfache Möglichkeit ist auch, die gefundenen Straßenmarkierungen nicht zu bewerten. Dadurch entsteht die Vorbedingung, dass sich das Fahrzeug bereits in der richtigen Fahrspur befindet[10]. Der Fokus liegt dann nur auf der Begrenzung der derzeitigen Fahrspur. Es kann allerdings nicht auf Ausnahmesituationen reagiert werden, ebenso können keine Aussagen über die aktuelle Position des Fahrzeugs getroffen werden.

In [24] wird eine Klassifizierung der gefundenen Straßenmarkierungen mit Hilfe der

Fourier-Analyse vorgenommen. Allerdings wird dort nur in unterbrochene, durchgezogene oder geschnittene Markierungen unterteilt.

Eine Möglichkeit stellt auch das Fitten von Fahrspurshablonen auf die Straße dar[25]. Dabei werden im Vorfeld Schablonen der Straße mit allen möglichen Szenarien erstellt und dann während des Fahrens auf die Fahrspurmarkierungen gelegt. Dadurch können die Fahrspuren, wenn sie vorhanden sind, immer in ein Straßenmodell überführt werden. Allerdings ist hier die Performance mit 10 Bildern pro Sekunde nicht ausreichend.

## 2.6 Wahl des Ansatzes

In Anbetracht der Anforderungen an diese Arbeit sind die Lösungsansätze kritisch zu betrachten. Dabei sollte das einfachste Modell gewählt werden, das die notwendige Funktionalität bietet:

*Entia non sunt multiplicanda praeter necessitatem.*

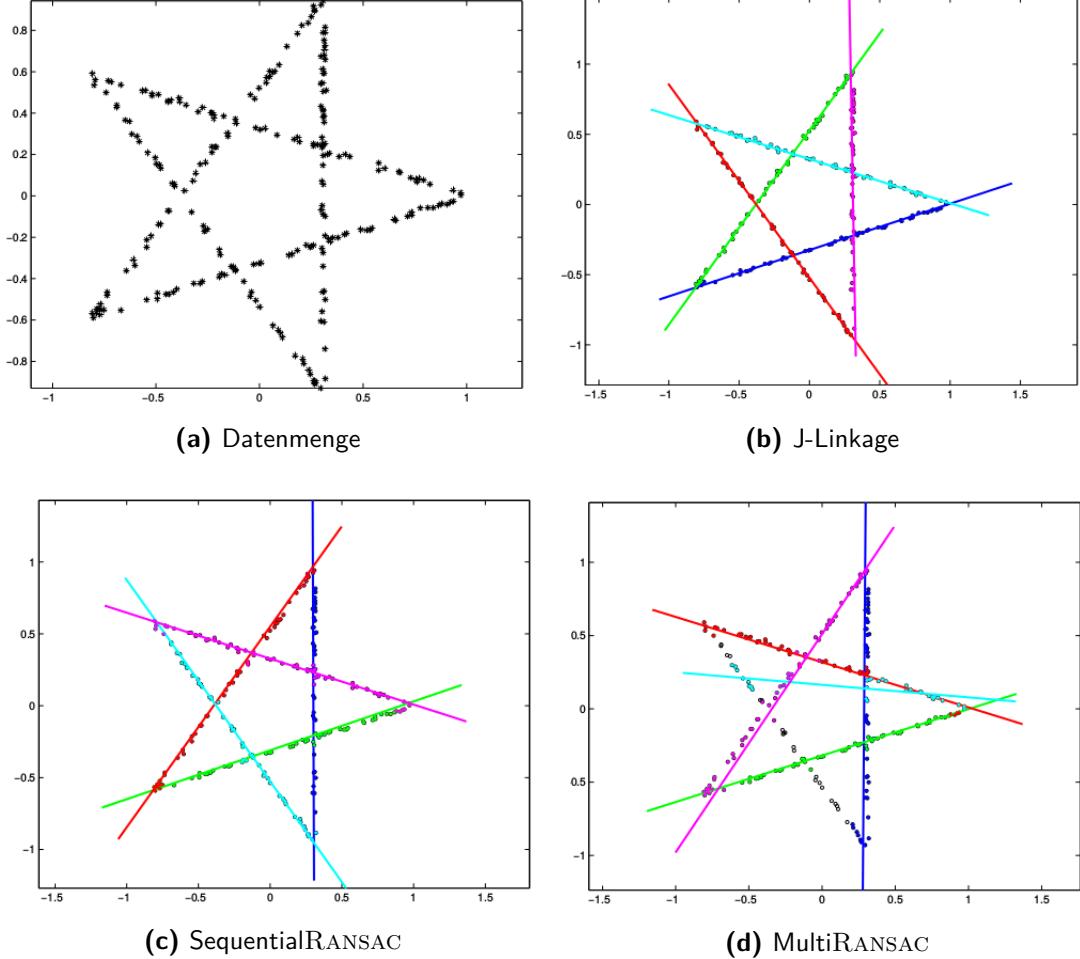
Occam's Razor[26]

Auf Seiten der Fahrspurmodellierung kann die Randomized Hough Transformation ausgeschlossen werden, da sie der Echtzeitanforderung nicht gewachsen wäre. In Abbildung 9 auf der nächsten Seite wird eine Übersicht der Qualität der anderen Lösungsansätze dargestellt. Dort ist zu sehen, dass bei einer Datenmenge mit wenig Ausreißern J-Linkage und SequentialRANSAC die besten Ergebnisse liefern. J-Linkage ist laut [21] allerdings ebenfalls nicht für Echtzeitanwendungen geeignet, jedoch wird dort auch eine Verbesserung des Verfahrens zur Nutzung in Echtzeit vorstellt. Diese Optimierung hat jedoch einen Qualitätsverlust zur Folge, der den Vorsprung des Verfahrens bei verrauschten Datenmengen annulliert.

Übrig bleibt der direkte Vergleich zwischen SequentialRANSAC und MultiRANSAC. Der größte Nachteil von SequentialRANSAC ist die Beeinträchtigung der Qualität der Resultate beim Auftreten von viel Hintergrundrauschen. Da die Anzahl der fehlerhaften Punkte durch die Kantenerkennung (siehe Abschnitt 2.2 auf Seite 6) jedoch sehr gering ist, hat dieses Argument kaum Bedeutung.

Auf der anderen Seite müssen bei MultiRANSAC die Anzahl der zu modellierenden Instanzen bei der Ausführung als Parameter übergeben werden. Da die Anzahl der Straßenmarkierungen jedoch stark schwankt, wie beispielsweise in Abbildung 3 auf Seite 4 gezeigt,

kann diese Voraussetzung nicht ohne weiteres Wissen erfüllt werden. Auch ist es fraglich, ob der Vorgang der Parallelisierung bei MultiRANSAC überhaupt rentabel ist.



**Abb. 9:** Ergebnisse von Algorithmen zur Bestimmung von mehreren Instanzen eines Modells[19]. In der Datenmenge ist nur sehr wenig Rauschen vorhanden.

Aus diesen Gründen wird in dieser Arbeit zur Erkennung der Straßenmarkierung SequentialRANSAC verwendet. Für die Validierung und Lokalisierung soll ein Verfahren verwendet werden, das die Abstände zwischen den Straßenmarkierungen misst und anhand dessen eine Validierung vornimmt. Die Messung erfolgt in verschiedenen Punkten, um eine bessere Aussage treffen zu können. Daher nennt sich diese Methode *Punktvalidierung*.

### 3 Fahrspurmodellierung mit Punktvalidierung

Nachdem nun verschiedene Lösungsansätze betrachtet wurden, erfolgt die genaue Erklärung des gewählten Ansatzes. Für das Modell, das die Fahrbahnmarkierungen der Fahrspuren repräsentieren soll, werden Polynome zweiten Grades verwendet. Diese Polynome werden durch Newtoninterpolation aus Datenpunkten erstellt, die mit der Kantenerkennung des Berlin United Racing Teams gefunden wurden.

In diesem Abschnitt wird zunächst SequentialRANSAC genauer vorgestellt. Die Ergebnisse des Algorithmus sollen dann mit dem Verfahren der Punktvalidierung bewertet werden. Dazu erfolgt zunächst die Beschreibung der Idee und anschließend die Erklärung der mathematischen Umsetzung. Danach wird das Verfahren zur Erkennung der Straßenmarkierungen erweitert, um die Resultate und die Laufzeit zu verbessern. Um alle Anforderungen zu erfüllen, muss auch die Spur gehalten werden können, wenn Straßenmarkierungen fehlen oder verdeckt sind. Im Abschnitt *Positionstracking* wird dazu ein einfacher Ansatz erklärt. Als Letztes wird noch die Möglichkeit für einen Fahrspurwechsel betrachtet, die sich mit diesem Ansatz bietet.

#### 3.1 Fahrspurerkennung mit SequentialRansac

SequentialRANSAC ermöglicht das Finden von mehreren Modellinstanzen in einem Bild. Im Falle des Kamerabildes des Modelfahrzeugs sind dies alle Straßenmarkierungen. Hierbei werden jedoch nur Kandidaten für Fahrbahnmarkierungen gefunden. Daher können auch andere Linien wie Parkspuren, Streckenbegrenzungen oder schlicht falsche Modellinstanzen darin enthalten sein. Als positives Resultat ist die erste Zeile von Abbildung 10 auf Seite 17 zu sehen. Es wurden alle sichtbaren Straßenmarkierungen gefunden, sowohl für die gerade befahrene Straße, als auch die Straße zur Rechten des Fahrzeugs. Ein Beispiel für eine falsche Modellinstanz ist in der zweiten Zeile zu finden. Dort wurde die rechte Fahrspur der befahrenen Straße nicht optimal modelliert. Zudem ist rechts neben der Fahrspur eine nicht zugehörige Linie erkannt worden.

Der Algorithmus wendet RANSAC auf eine Datenmenge  $D$  an. Dabei werden die Inlier des  $j$ -te Modells  $M_j^*I$  aus  $D$  entfernt, wenn  $M_j^*$  gefunden werden konnte. Dies geschieht, bis entweder  $m$  Polynome gefunden wurden oder die Kardinalität von  $D$  unter einen Schwellwert  $\delta$  fällt oder keine gültigen Modellinstanz  $M_j^*$  mehr gefunden werden kann. Pseudocode befindet sich in Algorithmus 1 auf der nächsten Seite.

Da der Algorithmus RANSAC verwendet, hat er zusätzlich zu den gerade genannten Parametern noch die in Abschnitt 2.4 auf Seite 9 erklärten Parameter:  $\epsilon$  als maximaler Abstand zur Modellinstanz,  $n$  als Anzahl der Punkte, die zum Erstellen der Modellinstanz verwendet werden,  $N$  als Anzahl der Iterationen und  $\mathcal{T}$  als Abbruchkriterium.

---

**Algorithmus 1 :** SequentialRANSAC

---

**Data :** Datenmenge  $D$ , bestehend aus Punkten

**Result :** Instanzen  $M_j^*$  des Modell  $M$  mit  $j = 1 \dots m$ , falls gefunden

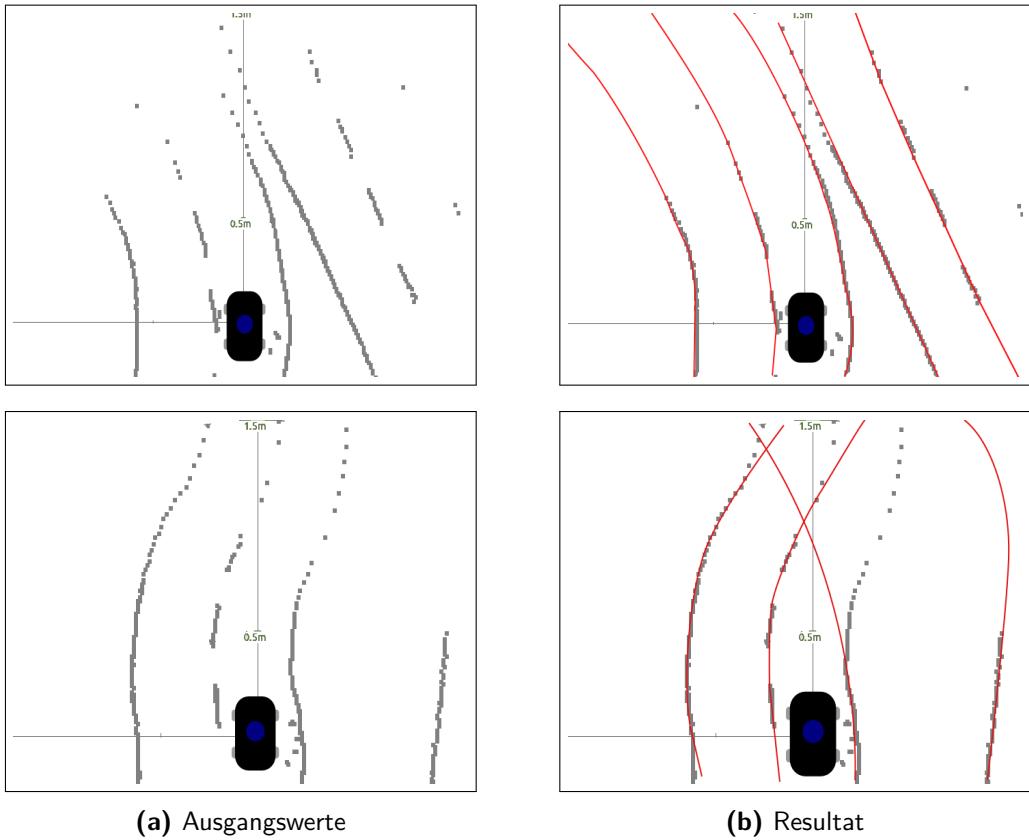
```

while  $j < m$  do
     $M_j^* \leftarrow \text{RANSAC}(\epsilon, \mathcal{T}, n, N)$ 
    if  $M_j^*$  ist ungültig then
         $\downarrow \text{return}$ 
    if  $|D| < \delta$  then
         $\downarrow \text{return}$ 
     $\text{removeInlier}(M_j^* I, D)$ 
     $j \leftarrow j + 1$ 
```

---

Durch die Verwendung der Polynome zweiten Grades ist  $n = 3$  immer gegeben.  $\epsilon$  kann zwar mathematisch berechnet werden[16], ist in diesem Ansatz jedoch wie  $\mathcal{T}$  und  $N$  durch Testen bestimmt worden.  $\delta$  wird auf 6 fest gewählt, da dies die maximale Anzahl an Straßenmarkierungen ist, die durch die Limitierung des Carolo-Cup's auftreten können. Die Datenmenge  $D$  beinhaltet alle Punkte vor dem Fahrzeug.

Um einen Punkt als Inlier zu identifizieren, muss für den RANSAC-Algorithmus noch eine Abstandsfunktion (Metrik) gewählt werden. Durch die Abstandsfunktion wird die Entfernung jedes Punktes aus der Datenmenge zum Polynom bestimmt. In diesem Ansatz wird der horizontale Abstand eines Punktes zum Polynom gewählt. Diese Berechnung hat im Vergleich zur euklidischen Distanz einen Laufzeitvorteil, da nicht quadriert werden muss. Außerdem ist zu erwarten, dass die Aussagekraft sehr ähnlich ist. Das liegt daran, dass die Polynome Straßenmarkierungen darstellen sollten, deren Krümmung nicht sehr stark ist.



**Abb. 10:** Ergebnisse von SequentialRANSAC (Rot) auf den Punkten der Kantenerkennung.  
Das Fahrzeug dient nur zur Veranschaulichung

### 3.2 Punktvalidierung

In diesem Abschnitt soll die Bewertung der erkannten Polynome vorgenommen werden. Dabei soll zunächst geprüft werden, ob es sich bei dem Polynom um eine Straßenmarkierung handelt. Wenn dies der Fall ist, soll die Zugehörigkeit bestimmt werden. Das bedeutet, dass eine Aussage über ihre Position im Straßenkontext getroffen wird.

Ein Polynom kann entweder eine rechte, linke oder mittlere Fahrbahnmarkierung darstellen, da im Carolo-Cup Straßen immer aus einer Spur pro Fahrtrichtung bestehen. Eine Straße wird immer in beide Richtungen befahren. Dieser Sachverhalt ist in Abbildung 11 auf der nächsten Seite veranschaulicht.

Da die Straßen des Carolo-Cup's einer strikten Norm unterliegen, haben alle Straßenmarkierungen einen erwarteten Abstand zueinander. Dieses Wissen kann genutzt werden, indem die Polynome paarweise durch ihren Abstand zueinander validiert werden.



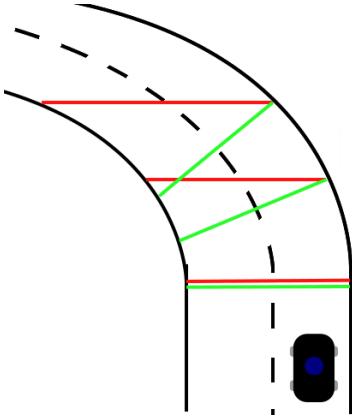
**Abb. 11:** Darstellung des Straßenverständnisses mit Positionen der Fahrbahnmarkierungen.  
Das Fahrzeug befindet sich in der rechten Fahrspur.

Da dies in mehreren Punkten des Polynoms getan werden sollte, nennt sich das Verfahren Punktvalidierung.

Mit Hilfe des Verfahrens soll nicht nur zwischen Fahrbahnmarkierung und anderen Straßenmarkierungen unterschieden werden. Zusätzlich werden die Fahrbahnmarkierungen in Straßenmodelle überführt, um dadurch auch mehrere Straßen erkennen zu können. Diese Straßenmodelle enthalten die Fahrbahnmarkierungen und ihre Positionierung. Die Position kann dabei, wenn mehr als eine Markierung in einem Modell enthalten ist, aus dem kombinierten Wissen des Abstandes der Markierungen zueinander und ihrer Abfolge auf der  $y$  – Achse erschlossen werden. Sollte kein Straßenmodell erstellt werden können, kann ohne weiteres Wissen (siehe Abschnitt 3.3 auf Seite 21) nur eine bedingte Aussage getroffen werden.

In diesem Fall wird das Polynom mit der höchsten Anzahl an Inlier als rechte Fahrspur deklariert, um so hoffentlich wieder auf eine Straße zu treffen. Derzeit wird bei mehreren erkannten Straßen schlicht das Straßenmodell zur Befahrung gewählt, dessen rechte Fahrspurmitte sich näher am Fahrzeug befindet.

Der Abstand zwischen den Polynomen ist von größter Bedeutung, da er direkt die Validierung beeinflusst. Daher muss er so genau wie möglich bestimmt werden. Es reicht nicht aus, lediglich die horizontalen Abstände zu bestimmen. Würde das Fahrzeug in eine Kurve fahren, wäre der Abstand sonst in der Krümmung viel höher als auf dem geraden Teil der Kurve. Dieses Phänomen ist in Rot in Abbildung 12 auf der nächsten Seite dargestellt. Stattdessen muss der Abstand anhand des Normalenvektors berechnet werden. Dies wird in den nachfolgenden Abschnitten erklärt.



**Abb. 12:** Zwei Methoden zur Abstandsbestimmung zwischen der linken und der rechten Fahrbahnmarkierung. In Rot ist der horizontale Abstand zu sehen, während Grün den Abstand entlang des Normalenvektors zeigt.

### 3.2.1 Mathematische Grundlagen

Die Idee dieser Abstandsbestimmung stammt aus dem Robocup. Dort wurden Linienfragmente durch einen Punkt mit seinem Normalenvektor zur Linie repräsentiert[27]. Straßenmarkierungen werden allerdings durch Polynome dargestellt, daher lässt sich zwar nicht die Repräsentation nutzen, dafür aber der Normalenvektor des Polynoms in einem bestimmten Punkt zur Bestimmung des Abstandes.

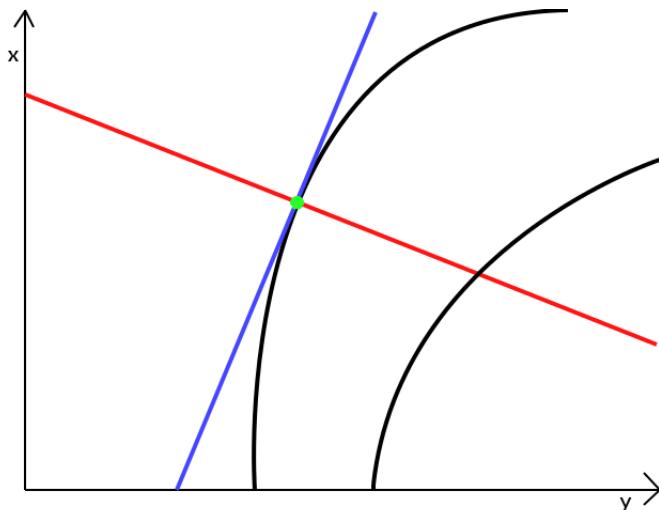
Eine Tangente  $t$  eines Polynoms  $P$  stellt die Steigung des Polynoms im Punkt  $a := (a_x, P(a_x))$  dar. Zur Bestimmung von  $P(a_x)$  muss  $P$  in  $a_x$  differenzierbar sein[28].

$$\begin{aligned} t(x) &= P'(a_x)(x - a_x) + P(a_x) \\ &= m_t x + n \end{aligned} \tag{1}$$

Hierbei entspricht  $m_t$  der Steigung von  $P$  im Punkt  $a$

Der Normalenvektor  $N$  einer Geraden  $g$  liegt orthogonal zu dieser Geraden. Soll der Normalenvektor durch einen Punkt  $a = (a_x, P(a_x))$  von  $G$  verlaufen, so kann er bestimmt werden durch:

$$N(x) = P(a_x) - \frac{1}{m_t}(x - a_x) \tag{2}$$



**Abb. 13:** Veranschaulichung des Normalenvektors (Rot) bestimmt durch die Tangente (Blau) in einem Punkt (Grün) auf einem Polynom (Schwarz, links)

Schnittpunkte von zwei Funktionen  $F(x)$  und  $G(x)$  können durch die Gleichsetzung der beiden Funktionen berechnet werden.

### 3.2.2 Abstandsbestimmung

Die drei Punkte  $a_i$  mit  $i = 1, 2, 3$  zur Bestimmung des Abstands befinden sich 10, 40 und 70 Zentimeter vor dem Fahrzeug. Für jeden Punkt wird nun eine Tangente auf dem aktuellen Polynom im Punkt  $a_i$  berechnet. Von dieser Tangente wird wiederum die Normale von  $a_i$  bestimmt. Anschließend werden die Normale und ein anderes Polynom  $R$  gleichgesetzt, um den Schnittpunkt zur Abstandsbestimmung zu berechnen. Dies ist auch in Abbildung 13 zu sehen. Dort wird  $P$  als das linke Polynom und  $R$  als das rechte Polynom dargestellt.

$$\begin{aligned} R(x) &= c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) \\ &= P(a_x) - \frac{1}{m_t}(x - a_x) \end{aligned} \tag{3}$$

Gleichung 3 lässt sich überführen in die Form:

$$ax^2 + bx + c = 0 \quad (4)$$

Dies ist wiederum mit der Mitternachtsformel aufzulösen[29]:

$$s_{x_{1,2}} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (5)$$

Hierbei werden allerdings nur die entstandenen Schnittpunkte  $s$  verwendet, die einen maximalen Abstand von 2 Metern zum Fahrzeug haben. Falls mehrere Schnittpunkte dieses Kriterium erfüllen, wird der Punkt verwendet, dessen euklidische Distanz zum Fahrzeug am Geringsten ist. Anschließend muss nur noch überprüft werden, ob die euklidische Distanz  $d_i$  zwischen allen Punkten  $a_i$  und den dazugehörigen Punkten  $s_i$  übereinstimmt.

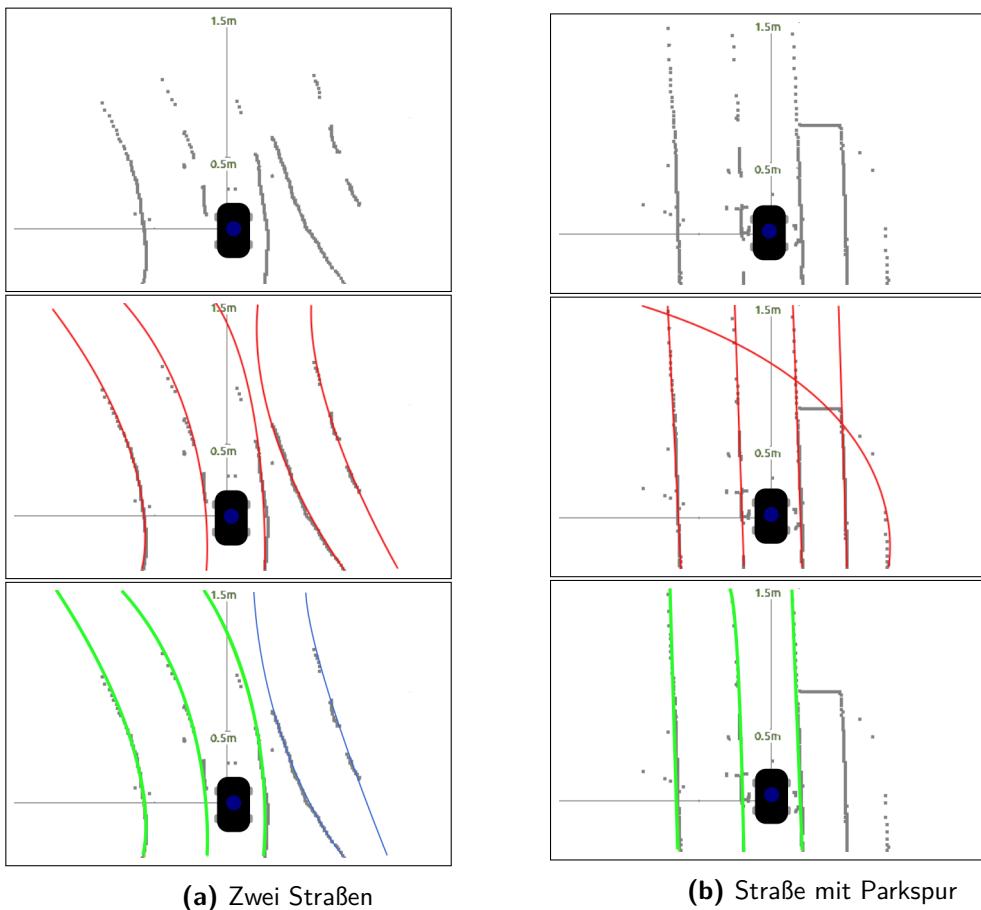
Haben alle Abstände  $d_i$  einen Wert, der innerhalb eines Schwellwerts der Norm aus dem Regelwerk des Carolo-Cup's entspricht, sind sowohl  $P$  als auch  $R$  valide Fahrbahnmarkierungen einer Straße. Im Anschluss an die Validierung werden alle gültigen Fahrspurpaare vereinigt. Dadurch werden nun die Fahrspurmarkierungen in Straßenmodellen dargestellt, wie in Abbildung 14 auf der nächsten Seite zu sehen ist.

### 3.3 Positionstracking

Es kann vorkommen, dass nicht alle Fahrbahnmarkierungen erkannt werden können. Wenn dies der Fall ist, kann die Straße nicht vollständig modelliert werden. Dieser Umstand kann aus zwei Gründen auftreten:

Der erste Grund ist das Modellieren durch Polynome, die nicht repräsentativ für Fahrbahnmarkierungen sind, wie es in Abbildung 10 auf Seite 17 mit der rechten Fahrbahnbegrenzung der zweiten Zeile der Fall ist. Diese Polynome werden bei der Validierung entfernt. Der andere Grund ist das Fehlen der Fahrspurmarkierungen im Bild. Dies ist im Carolo-Cup beispielsweise eine übliche Herausforderung, um die Robustheit des Ansatzes zu testen.

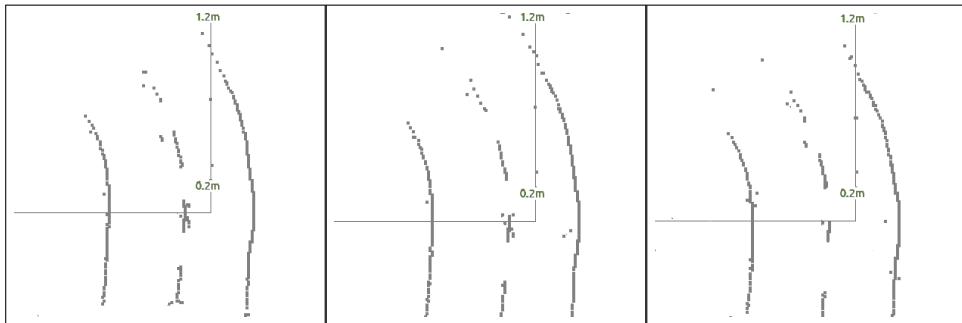
Existiert im Bild nur eine oder zwei Fahrbahnmarkierung der Straße, kann mit dem bisherigen Aufbau zur Straßenerkennung nur eine sehr schlechte Aussage über die Position der Fahrbahnmarkierung getroffen werden.



**Abb. 14:** Ergebnis des Verfahrens. Die erste Zeile zeigt die Ausgabe der Kantenerkennung. Die zweite Zeile zeigt die gefundenen Kurven im Bild mit SequentialRANSAC. Die dritte Zeile zeigt die Straßenmodelle durch die Punktvalidierung, wobei das zur Befahrung ausgewählte Modell in Grün dargestellt wird

Eine Lösung für dieses Problem ist die Verfolgung der Positionen der Fahrbahnmarkierungen. Dies lässt sich dadurch realisieren, dass die Straßenmodelle des letzten Bildes gespeichert und das Wissen über die Positionen im aktuellen Bild genutzt wird. Da zwischen zwei Bildern der Unterschied der Fahrbahnmarkierungen nur sehr begrenzt ausfällt, ist anzunehmen, dass sich die repräsentierenden Polynome ähneln (siehe Abb. 15 auf der nächsten Seite).

Daher kann aus der Position einer Fahrspur im letzten Bild auf die Position der aktuellen Fahrspur geschlossen werden. Dabei wird allerdings zur Vermeidung von Fehlern bei unklaren Aussagen immer das Modell aus dem derzeitigen Bild bevorzugt. Nun kann bei Straßen mit nur einer Fahrbahnmarkierung die Fahrspur gehalten werden, wenn bereits



**Abb. 15:** Drei aufeinander folgende Bilder nach der Vorverarbeitung durch die Kantenerkennung

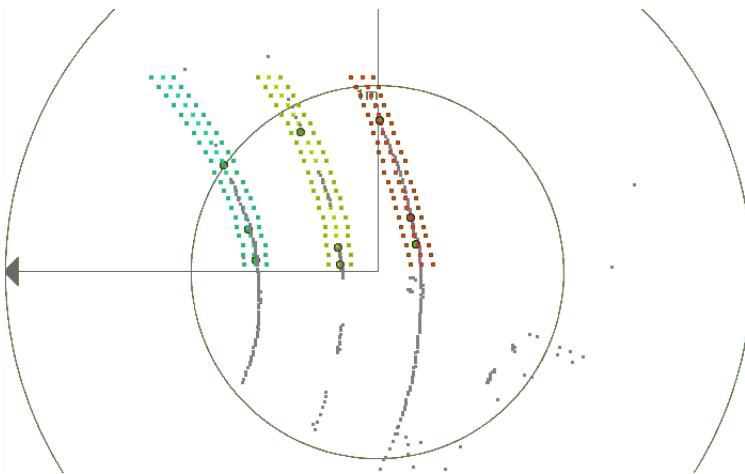
Vorwissen aus dem vorherigen Bild existiert.

### 3.4 Laufzeitverbesserung

Eine weitere Verbesserung stellt die Verwendung eines Teilaspekts aus der Arbeit von Conrad Läßig dar[5]. Hierbei handelt es sich ebenfalls um das Nutzen von Vorwissen aus dem vorherigen Bild, diesmal allerdings nicht zur Verbesserung der Funktionalität, sondern der Laufzeit.

Die Idee dieses Verfahrens stammt aus [25]. Dort werden um die Straßenmarkierungen der Fahrbahn des vorherigen Bildes dynamische Schläuche gelegt. Dieser Vorgang ist in Abbildung 16 auf der nächsten Seite zu sehen. In diesen Schläuchen wird im aktuellen Bild gezielt nach Fahrbahnmarkierungen gesucht, indem die Punkte in eigene Teilmengen der Datenmenge überführt werden und auf diesen dann RANSAC ausgeführt wird. Dies ist natürlich nur möglich, wenn Vorwissen über die Straßenmodelle des letzten Bildes existiert.

Unter der Annahme, dass das Modell des letzten Bildes korrekt ist, kann nun RANSAC auf eine Datenmenge mit einem sehr hohem Inlier-Anteil ausgeführt werden. Dadurch lässt sich durch sehr wenige Iterationen ein repräsentatives Polynom finden. Das Polynom wird zusätzlich mit zusätzlichen Kriterien bewertet, wie beispielsweise der Ähnlichkeit zum Vorgänger[5]. Können die Kriterien nicht erfüllt werden, sind die Datenpunkte aus dem entsprechenden Schlauch in die Datenmenge zu verschieben, damit dort wieder, im Anschluss an das Suchen im Schlauch, SequentialRANSAC ausgeführt werden kann.

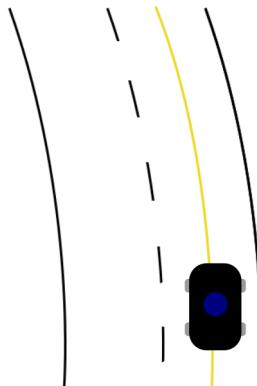


**Abb. 16:** Darstellung der Fahrspurenpolynome mit ihren schlauchartigen Suchbereichen[5]

### 3.5 Fahrspurverfolgung

Als Letztes muss geklärt werden, wie aus den Straßenmodellen die Fahrspuren bestimmt werden und diese dann verfolgt werden können. Die Darstellung der Fahrspur geschieht durch die Interpolation der Mittelpunkte zwischen den Polynomen des Straßenmodells. Dadurch entsteht ein Polynom, das in der Fahrspurmitte einer Fahrspur verläuft (siehe Abbildung 17 für ein Beispiel für die rechte Fahrspur). Dieses Polynom wird an die Pfadverfolgung von Daniel Krakowczyk [30] übergeben, die einen Pfad bestimmt, der dann abgefahren werden muss. Kann keine Straße und damit keine Straßenmitte gefunden werden, wird in Ermangelung von Informationen ein gerades Polynom übergeben.

Durch die Modellierung der Fahrspuren durch Polynome, die wiederum in Straßenmo-



**Abb. 17:** Darstellung einer Straße mit Polynom durch die rechte Fahrspurmitte (Gelb)

delle gegliedert sind, ist das Planen eines Überholmanövers sehr einfach. Dazu muss lediglich die Mitte der linken Fahrspur an die Pfadplanung übergeben werden. Dies hat zur Folge, dass ein Pfad in die linke Fahrspur eingeschlagen wird und das Fahrzeug dort bis zur Rückkehr in die rechten Fahrspur verweilt. Daher ist der Überholvorgang dynamisch und kann in einer Fahrplanung mit Hinderniserkennung verwendet werden.

## 4 Evaluation

In diesem Abschnitt soll der vorgestellte Ansatz evaluiert werden. Dazu sollen verschiedene Experimente durchgeführt werden, die die Funktionalität und Robustheit des Ansatzes unter Beweis stellen sollen. Die Experimente werden direkt mit dem letzten Ansatz des Berlin United Racing Teams von Conrad Läßig verglichen, welcher im Folgenden auch RANSAC-Ansatz genannt wird. Da die Implementierung dieses Ansatzes fehlerhaft ist, wurde stattdessen eine Referenzimplementierung erstellt und in den Experimenten verwendet.

Als Erstes soll die allgemeine Funktionalität bewertet werden. Dazu wurden die Ansätze auf einem optimalem Pfad getestet. Im zweiten Experiment weicht das Auto oft vom optimalem Pfad ab und verlässt auch die Fahrspuren. Dies geschieht sowohl nach rechts von der Straße weg, als auch nach links in die andere Fahrspur, da dies übliche Fehler im Fahrverhalten sind. Das dritte Experiment testet das Verhalten des Fahrzeugs beim Fehlen von Fahrspuren, da damit im Carolo-Cup zu rechnen ist. Im letzten Experiment soll die Durchführbarkeit eines Überholmanövers geprüft werden.

### 4.1 Bewertungssystem

Für die Experimente 1 bis 3 wurde mit Hilfe der Fernsteuerung die Teststrecke abgefahren und die Kamerabilder aufgezeichnet. Dabei wurde für das erste Experiment ein Pfad, der genau in der Mitte der Fahrspur lag, gewählt. Beim zweiten Experiment wurde dies bewusst vernachlässigt. Dadurch soll das Fahrverhalten des Fahrzeugs simuliert werden. Um fehlende Fahrspuren für das dritte Experiment zu simulieren, wurden die Kamerabilder entsprechend manipuliert, sodass in einigen Bildern nur eine oder zwei Fahrspuren zu sehen sind.

Auf diesen Kamerabildern wurden im Nachhinein die Ansätze ausgeführt und die Richtigkeit der einzelnen Fahrspurmarkierungen und der Bestimmung der Fahrspurmitte überprüft. Dafür wird ein Punktesystem eingeführt, welches die Qualität darstellen soll. Wird die Fahrspurmarkierung richtig erkannt und modelliert, erhält der Ansatz für dieses Bild einen Punkt. Bei fehlerhafter Einschätzung der Fahrspurmarkierung wird ein Punkt abgezogen. Sollte der Ansatz keine Angabe zur Fahrspurmarkierung machen, bleibt der Punktestand unverändert. Das System wird in den Abbildungen in Experiment 1, 2 und 3 verwendet. Dort sind die erhaltenen Punkte pro Bild aufsummiert, um den Qualitätsunterschied der Ansätze im Verlauf der Bilder deutlich zu machen. Außerdem befindet

sich dort auch ein Farbbalken parallel zur  $x$ -Achse, der die einzelnen Streckenabschnitte kodiert.

Im vierten Experiment wurde ein Überholmanöver in die linke Fahrspur durchgeführt und die benötigte Strecke bei unterschiedlichen Geschwindigkeiten gemessen.

## 4.2 Vorbetrachtung

Zu erwarten ist, dass sich im ersten Experiment auf Grund der Vorbedingungen des optimalen Pfades die Resultate der beiden Ansätze sehr ähneln. Außerdem sollte dort auch eine hohe Qualität erreicht werden, da dies eine Anforderung an den Ansatz darstellt. Im zweiten Experiment sollte der Ansatz der Punktvalidierung deutlich bessere Resultate liefern, da der RANSAC-Ansatz nicht für diese Aufgabe ausgelegt ist. Allerdings sollte die Punktvalidierung auch in diesem Experiment gute Resultate erbringen, da dies die wesentliche Verbesserung gegenüber dem vorherigen Ansatz darstellt. Beim dritten Experiment sollte der RANSAC-Ansatz jedoch bessere Ergebnisse liefern, da die benötigten Vorbedingungen der Fahrzeugposition einen Vorteil bedeutet. Da die Implementierung der Punktvalidierung das Überholmanöver in der Theorie sehr einfach macht, ist zu erwarten, dass das vierte Experiment auch in der Praxis erfolgreich ist.

## 4.3 Experiment 1: Funktionalität mit optimalem Fahrpfad

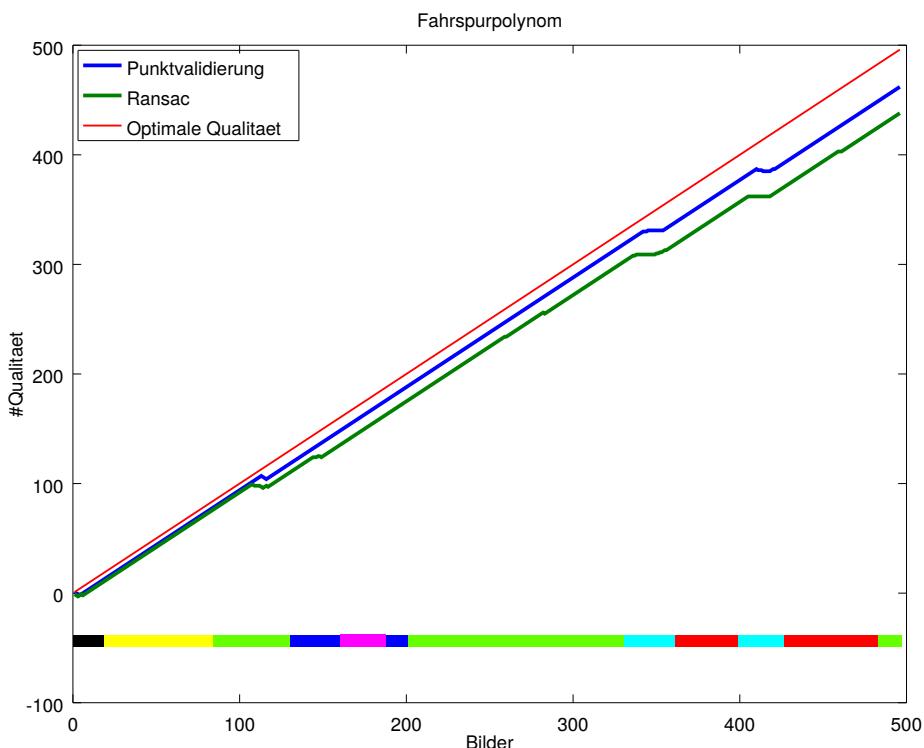
In diesem Experiment wurden 496 Bilder ausgewertet. Dabei wurde bei der Aufnahme der Bilder mit der Fernsteuerung ein optimaler Fahrpfad in der Mitte der rechten Fahrspur erreicht. Tabelle 1 auf der nächsten Seite zeigt die Klassifikationsergebnisse der Punktvalidierung und des RANSAC-Ansatzes. Dort sind die Anzahl der Bilder zu sehen, in denen die Verfahren eine korrekte Klassifizierung, eine falsche Klassifizierung oder keine Angabe ausgegeben haben. Dabei sind sämtliche Fahrspurmarkierungen einzeln betrachtet. Außerdem wird das Fahrspurpolynom, das aus den Fahrspurmarkierungen entsteht, dort aufgeführt. Dies ist die wichtigste Aussage der Tabelle, da dies das Endergebnis der Verfahren darstellt. Die Werte der linken Fahrspur und des Fahrspurpolynoms sind bei beiden Ansätzen ähnlich, während sich bei der Erkennung der mittleren und rechten Fahrspurmarkierung eine deutliche Differenz bemerkbar macht. Außerdem haben beide Ansätze bei der Bestimmung des Fahrspurpolynoms den höchsten Prozentwert.

Abbildung 18 zeigt die Qualität der beiden Ansätze für die befahrene Strecke. Dazu wird nach dem in Abschnitt 4.1 vorgestellten Bewertungssystem die Bestimmung des

Markierung	Links		Mitte		Rechts		Fahrspurpolynom	
	P	R	P	R	P	R	P	R
Korrekt in %	80,4	76,2	84,7	70,2	84,9	77,4	94,8	90,3
Keine Aussage in %	18,3	20,2	14,1	29,2	14,5	21,7	3,6	7,7
Fehler in %	1,2	3,6	1,2	0,6	0,6	0,8	1,6	2

**Tab. 1:** Klassifizierungen der Punktvalidierung (P) und des RANSAC-Ansatzes (R) bei der Einhaltung eines optimalen Fahrpfades von 496 Bildern

Fahrspurpolynoms bepunktet. Entlang der  $x - Achse$  ist die Zeit in Form der bewerteten Bilder zu sehen, während auf der  $y - Achse$  die Qualität der Ansätze abzulesen ist. Dort ist zu erkennen, dass der Ansatz der Punktvalidierung einen leichten Vorsprung zum



**Abb. 18:** Qualität der Ansätze auf bestimmten Streckenabschnitten mit optimaler Pfadverfolgung, wobei die Qualitätspunkte pro Bild aufsummiert sind. Schwarz = Startbucht, Gelb = Gerade mit Parkspur, Grün = Linkskurve mit starker Krümmung, Blau = Kreuzung ohne Stoplinie, Violett = Gerade, Cyan = Kreuzung mit Stoplinie, Rot = Rechtskurve

Ende der Bildserie herausarbeitet. Allerdings haben beide Ansätze an den selben Streckenabschnitten ihre Schwierigkeiten, nämlich der Linkskurve mit starker Krümmung und den Kreuzungen, an denen Vorfahrt gewährt werden muss.

#### 4.4 Experiment 2: Funktionalität mit fehlerhaftem Fahrverhalten

Da das Fahrzeug nicht immer im optimalen Pfad fährt, soll dieses Experiment ein Fahren mit bewussten Schwankungen simulieren. Auch hier wurden Kamerabilder ausgewertet, die durch das Befahren der Teststrecke mit der Fernsteuerung entstanden sind.

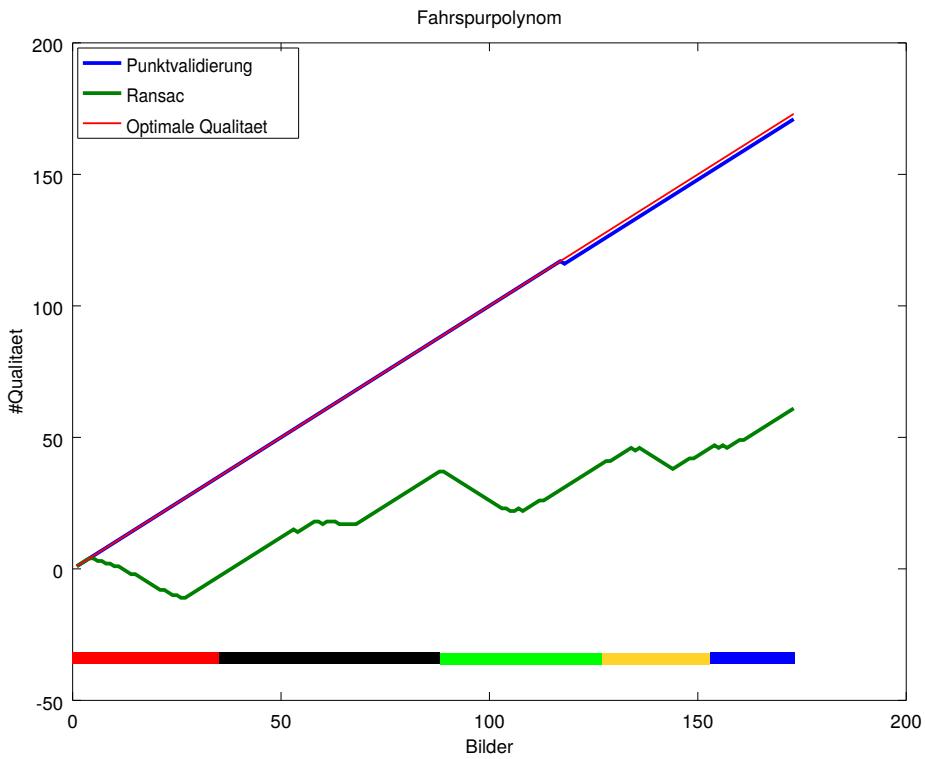
Die Klassifikationsergebnisse sind wie im letzten Experiment in einer Tabelle dargestellt. Dort ist zu erkennen, dass die Punktvalidierung kaum Fehler bei der Klassifizierung gemacht hat. Die Erfolgsrate liegt bei 98,8%. Dies ist sogar noch besser als mit einem optimalen Pfad. Dem RANSAC-Ansatz bereitet besonders die Modellierung der linken und mittleren Fahrspurmarkierungen Probleme. Dort konnten nur knapp ein Drittel der Markierungen korrekt bestimmt werden. Auch die Bestimmung des Fahrspurpolynoms ist mit 61,3% deutlich schlechter als mit Optimalpfad.

Markierung Verfahren	Links		Mitte		Rechts		Fahrspurpolynom	
	P	R	P	R	P	R	P	R
Korrekt in %	72,6	30	83,2	31,8	94,8	59,5	98,8	61,3
Keine Aussage in %	27,2	55,5	16,2	49,7	4,6	23,1	0	12,7
Fehler in %	0	14,5	0,6	18,5	0,6	17,3	1,2	26

**Tab. 2:** Klassifizierungen der Punktvalidierung (P) und des RANSAC-Ansatzes (R) mit fehlerhaftem Fahrverhalten von 173 Bildern

In der Betrachtung der Qualität pro Streckenabschnitt in Abbildung 19 auf der nächsten Seite wird deutlich, dass der RANSAC-Ansatz mit Schwankungen innerhalb der Fahrspur und in der engen Linkskurve noch vernünftige Resultate liefert, bei den anderen Streckenabschnitten jedoch größtenteils versagt.

Bei diesem Experiment wurden zusätzlich noch die Laufzeiten der Ansätze verglichen. Die durchschnittliche Laufzeit der Punktvalidierung betrug auf dem Fahrzeug 1,699 Millisekunden, während der RANSAC-Ansatz nur 0,933 Millisekunden benötigt.

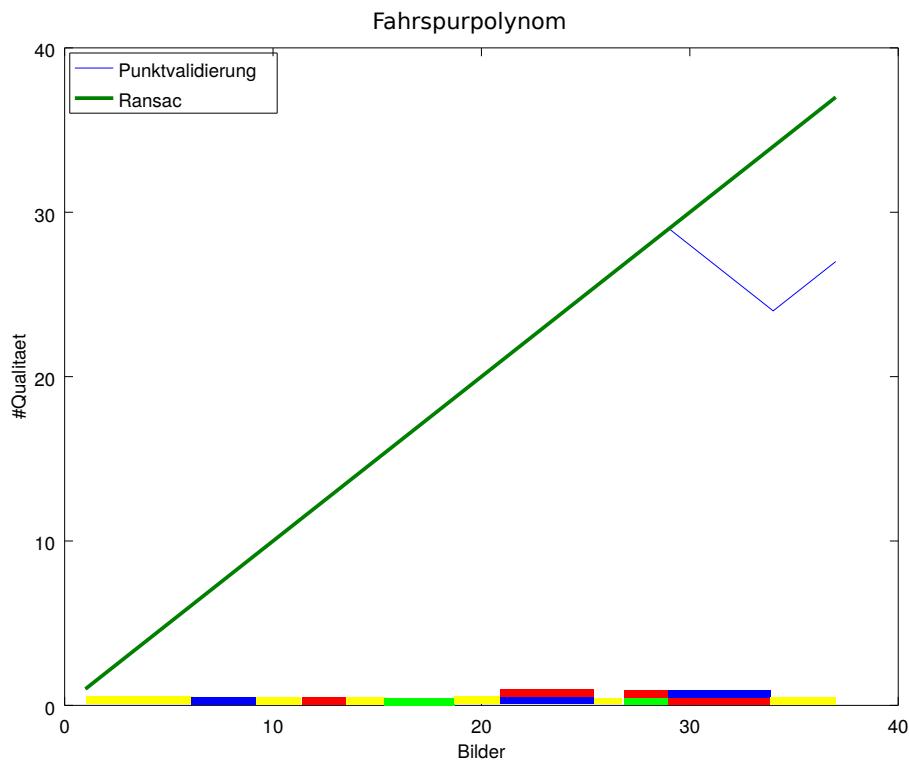


**Abb. 19:** Qualität der Ansätze auf bestimmten Streckenabschnitten mit Fehlerverhalten, wobei die Qualitätspunkte pro Bild aufsummiert sind. Rot = Abdriften in linke Fahrspur auf gerader Strecke, Schwarz = enge Linkskurve, Grün = Abdriften in linke Fahrspur in Linkskurve, Gelb = Überfahren der rechten Straßenbegrenzung in einer Rechtskurve, Blau = Schwankungen in der Fahrspur

#### 4.5 Experiment 3: Funktionalität im Ausnahmefall

Mit Hilfe des Positionstrackings der Punktvalidierung soll die Straße auch dann erkannt werden, wenn mehrere Fahrspurmarkierungen fehlen. Dies soll in diesem Experiment getestet werden. Dafür wurden Markierungen aus den Bildern entfernt, sodass nicht immer alle Markierungen zu sehen waren.

In Abbildung 20 auf der nächsten Seite wird wieder die Qualität bewertet. Dort sind die fehlenden Fahrspurmarkierungen farblich kodiert, die Qualität wird hier anhand der Bestimmung des Fahrspurpolynoms bewertet. Wenn mehrere Farbbalken überein-

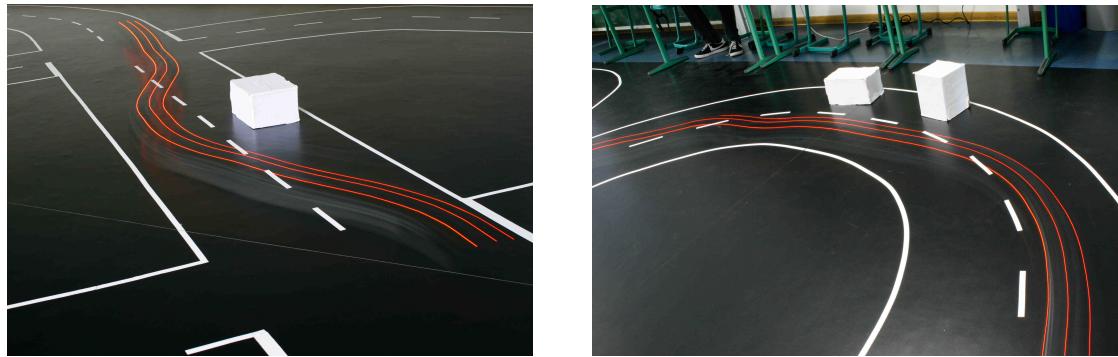


**Abb. 20:** Qualität beim Fehlen von Fahrspurmarkierungen von 37 Bildern. Der Farbstreifen gibt Aufschluss darüber, welche Markierungen fehlen: Gelb = Keine, Blau = Rechts, Rot = Mitte, Grün = Links. Dabei können auch mehrere Markierungen fehlen.

ander angeordnet sind, fehlen dort alle dadurch kodierten Fahrspurmarkierungen. Aus der Darstellung wird deutlich, dass keine Fehler auftreten, solange vor dem Fehlen einer Markierung zuvor alle Markierungen erkannt wurden. Tritt jedoch der Fall auf, dass unterschiedliche Markierungen in Reihe fehlen, wie in der Abbildung von Bild 27 - 34, so wird das Fahrspurpolynom inkorrekt bestimmt. Der RANSAC-Ansatz bleibt davon unbeeinflusst.

## 4.6 Experiment 4: Überholmanöver

Da noch keine Hinderniserkennung für das Fahrzeug funktionstüchtig ist, wurde statt des Umfahrens eines realen Objektes auf der Fahrbahn ein zeitgesteuertes Überholmanöver durchgeführt. Das Fahrzeug fährt dabei eine bestimmte Zeit in der rechten Fahrspur und wählt nach einer gewissen Zeit die linke Fahrspur zur Befahrung aus. Dort verweilt es, bevor es wieder die rechte Fahrspur ansteuert.



**Abb. 21:** Langzeitbelichtungsaufnahmen während der Überholmanöver. Zur Veranschaulichung sind die Bremslichter dauerhaft eingeschaltet und ein Objekt auf der Fahrbahn aufgestellt, welches jedoch nicht erkannt wird. Fotos: Severin Junker

Dieses Manöver wurde auf gerader Strecke und in einer Linkskurve getestet. In beiden Szenarien konnte das Fahrzeug das Überholmanöver ausführen (siehe Abbildung 21). Während in der Kurve nur mit einer Geschwindigkeit von 1 Meter/Sekunde getestet wurde, konnten auf der Geraden verschiedene Geschwindigkeiten gefahren werden. Dort wurde auch die Distanz parallel zur Straße gemessen, die das Fahrzeug für das Fahrspurwechseln benötigt. Dabei benötigt das Fahrzeug für den Wechsel von links nach rechts die identische Entfernung wie beim umgekehrten Wechsel. Die Ergebnisse sind in Tabelle 3 dargestellt. Es ist deutlich zu sehen, dass je schneller das Fahrzeug fährt, desto weniger Strecke für den Vorgang benötigt wird.

Geschwindigkeit im m/s	Strecke in cm
0.5	140
1	110
1.5	70

**Tab. 3:** Vergleich der zurückgelegten Strecke während des Überholvorgangs bei verschiedenen Geschwindigkeiten

## 4.7 Diskussion

Experiment 1 zeigt, dass die Ergebnisse der Ansätze bei Verwendung eines optimalen Pfades auch sehr nah am Optimalwert liegen. Die einzigen Schwächen liegen in der Krümmung der ersten Linkskurve und den Kreuzungen mit Stoplinien. In der Linkskurve ist die Krümmung auf Grund der räumlichen Gegebenheiten sehr stark ausgeprägt. Dies sorgt bei beiden Ansätzen für Fehler, wobei die negativen Resultate beim RANSAC-Ansatz zuerst beginnen und dadurch ausgeprägter sind. Kreuzungen mit Stoplinien stellen ein schwieriges Szenario dar. Wenn sich das Fahrzeug unmittelbar vor und auf der Kreuzung befindet, kann der Beginn der Straße noch nicht erkannt werden. Dies hat zur Folge, dass die Fahrspuren in diesem Zeitraum nicht erkannt werden können. Prinzipiell ist die linke Fahrspurmarkierung am schlechtesten erkannt worden, was allerdings wenig verwunderlich ist, da diese Markierung am Weitesten vom Fahrzeug entfernt ist und daher auch durch am wenigsten Linienpunkte repräsentiert wird. Die Haupterkennnis ist, dass die Qualität der Punktvalidierung unter optimalen Verhältnissen dem RANSAC-Ansatz mindestens ebenbürtig ist.

Das zweite Experiment testet die verbesserte Funktionalität dieses Ansatzes. Dies geschieht durch das Abfahren eines Pfades, der starke Schwankungen des Fahrzeugs simuliert. Die Ergebnisse bekräftigen, dass der RANSAC-Ansatz definitiv nicht auf ein Verlassen der Fahrspur ausgelegt ist. Im Gegensatz dazu zeigt der Ansatz der Punktvalidierung hier seine Vorteile. Durch die Modellierung der Fahrspuren in Straßen kann beim Verlassen der Fahrspur und auch der Straße trotzdem noch die rechte Fahrspurmitte bestimmt werden. Die Punktvalidierung ist zwar nicht so effizient in der Laufzeit wie der RANSAC-Ansatz, stellt jedoch auch deutlich mehr Funktionalität bereit. Allerdings wird mit diesem Ergebnis immer noch deutlich die Echtzeitanforderung erfüllt.

Wenn Straßenmarkierungen fehlen, sollte trotzdem die Fahrspur gehalten werden. Dies wird in Experiment 3 auf die Probe gestellt. Dabei liefert hier der RANSAC-Ansatz bessere Ergebnisse, da er auf eine Vorbedingung der relativen Fahrspuren zum Fahrzeug zurückgreifen kann (siehe Abschnitt 2.4 auf Seite 9). Da im Carolo-Cup derzeit kein Fehlen von Fahrspurmarkierungen in Reihe vorkommt, ist das Resultat auch hier zufriedenstellend.

Durch den Praxistest des Überholmanövers können in Zukunft nun Fahrspurwechsel vollzogen werden. Dabei ist allerdings auffällig, dass sich das Fahrzeug nicht menschlich verhält: Statt bei einer höheren Geschwindigkeit über eine größere Distanz die Fahrspur zu wechseln, verkürzt sich die zurückgelegte Strecke beim Überholvorgang.

Es lässt sich sagen, dass die Funktionalität des RANSAC-Ansatzes durch den in dieser Arbeit beschriebenen Ansatz verbessert wurde, da die Vorbedingungen reduziert wurden. An dieser Stelle ist zudem noch anzumerken, dass das Fahrzeug in der Praxis, obwohl die Fahrspurpolynome korrekt bestimmt wurden, in Kurven oftmals die mittlere Fahrspurmarkierung überfährt. Da der Pfadplanung allerdings ein korrektes Fahrspurpolynom übergeben wurde und dies bei beiden Ansätzen auftritt, ist die Lösung dieses Problems dort zu suchen. Das Problem der Vorfahrtskreuzungen aus Experiment 1 macht sich zudem in der Praxis durch ein ungewolltes Abbiegen in die linke Straße deutlich. Dieses Phänomen wird verstärkt, je schneller das Fahrzeug fährt. Bis zu einer Geschwindigkeit von 1 Metern pro Sekunde ist das Verfahren bei guten Lichtverhältnissen allerdings stabil.

## 5 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Ansatz zur Erkennung, Validierung und Bestimmung der Position der Fahrspurmarkierungen vorgestellt. Dabei wird zur Erkennung von Linien der Algorithmus SequentialRANSAC und zur Validierung und Postionsbestimmung das Verfahren der Punktvalidierung mit Erweiterungen durch Positionstracking und Modelladaption verwendet. Dafür werden allerdings genormte Straßen benötigt, wie sie im Carolo-Cup oder auf Autobahnen (siehe Straßenverkehrs-Ordnung und VwV-StVO) zu finden sind.

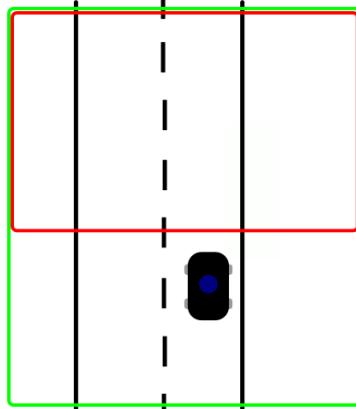
Das Verfahren ist in Experimenten im allgemeinen Straßenverkehr und im Korrigieren von Fehlern getestet worden. Dabei hat sich gezeigt, dass dieser Ansatz beim Fahren mit einem optimalen Pfad und mit Fahrfehlern auf den meisten Streckenteilen eine korrekte Lösung erstellt.

Allerdings haben sich in der Evaluation auch Probleme offenbart. Ein Problem ist das unbewusste Abbiegen bei Kreuzungen, an denen der anderen Straße Vorfahrt gewährt werden soll. Da das Abbiegen unabsichtlich erfolgt, entstehen dadurch weitere Fehler. So wird kein Blinker gesetzt und auch keine Vorfahrtsregeln eingehalten. Im Carolo-Cup resultieren aus Brüchen der Verkehrsregeln Zeitstrafen, die vermieden werden sollten. Dies betrifft auch das Überfahren der mittleren Fahrspurmarkierung in Kurven, daher ist der Ansatz zur Pfadverfolgung zu überarbeiten.

Eine Lösung für das Kreuzungsproblem wäre, statt alle Markierungen vor dem Fahrzeug, auch Straßenmarkierung hinter dem Fahrzeug in die Erstellung der Polynome einzubeziehen, wie in Abbildung 22 auf der nächsten Seite dargestellt ist. Dies könnte passieren, sobald die Kreuzungserkennung von Benjamin Zengin[31] eine Vorfahrtskreuzung erkennt. Dadurch könnten mehr Informationen in diesen Ansatz einfließen und zu einem besseren Ergebnis führen.

Das andere Problem entsteht, wenn hintereinander verschiedene Fahrspurmarkierungen im Bild fehlen. Dann entsteht ein falsches Ergebnis. Dies könnte behoben werden, indem durch die erkannten Fahrspuren auf die unerkannten Fahrspuren geschlussfolgert wird. Dadurch würde das Modell allerdings stärker auf Annahmen basieren.

Eine weitere Verbesserung stellt das Bilden einer Karte der Strecke während des Fahrens dar. Dadurch, dass mehrere Straßen in einem Bild erkannt werden können, sollte dies mit Hilfe dieses Verfahrens möglich sein. Mit der Karte könnte die Pfadplanung verbessert



**Abb. 22:** Verschiedene Suchbereiche des Verfahrens. Der derzeitig verwendete Bereich ist in rot eingezeichnet, während der Bereich für das Überqueren einer Kreuzung in grün eingezeichnet ist.

werden, da ein Blick auf die Straße weit vor dem Auto möglich wird. Dadurch kann auf geraden Strecken zusätzlich beschleunigt werden.

In diesem Zusammenhang wäre die Nutzung der Stereokamera von Jannis Ihrig[32] sinnvoll, da hierdurch das Sichtfeld vor dem Fahrzeug stark erweitert wird. Damit verbunden wäre eine Hinderniserkennung, die durch diesen Ansatz nun dynamische Überholmanöver einleiten könnte.

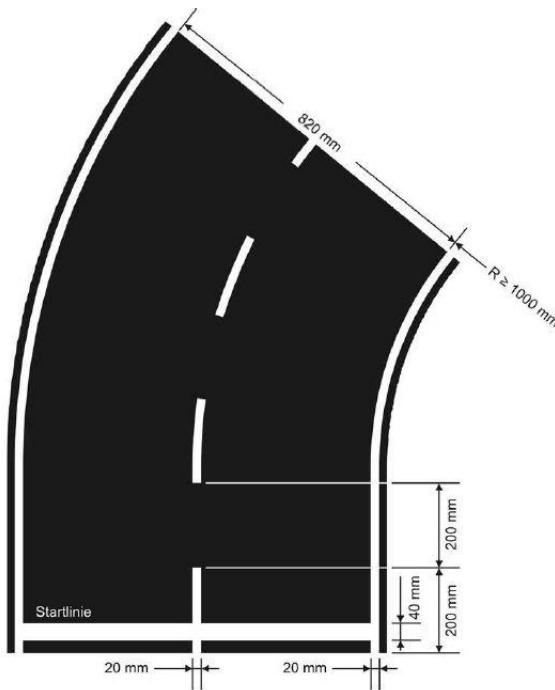
## Literatur

- [1] Dietmar H. Lamparter. *Wer hat das Kommando?* <http://www.zeit.de/2015/09/autonomes-fahren-assistenzsystem-autotechnik>. [Online; accessed 04-September-2015]. Die Zeit, 2015.
- [2] Axel Kannenberg. *Googles eigene autonome Autos kommen auf die Straße.* <http://www.zeit.de/2015/09/autonomes-fahren-assistenzsystem-autotechnik>. [Online; accessed 07-September-2015]. heise online, 2015.
- [3] *Autonom in Zürich unterwegs.* [http://www.fu-berlin.de/presse/informationen/fup/2015/fup\\_15\\_131-autonom-in-zuerich-unterwegs/index.html](http://www.fu-berlin.de/presse/informationen/fup/2015/fup_15_131-autonom-in-zuerich-unterwegs/index.html). [Online; accessed 07-September-2015]. Freie Universität Berlin, 2015.
- [4] Xin Liu u. a. „Real-time long-range lane detection and tracking for intelligent vehicle“. In: *Image and Graphics (ICIG), 2011 Sixth International Conference on*. IEEE. 2011, S. 654–659.
- [5] Conrad Läßig. „Fahrspurerkennung mit RANSAC bei autonomen Modellfahrzeugen“. In: (2015).
- [6] Technische Universität Braunschweig. *Carolo-Cup Regelwerk 2015.* <https://wiki.ifr.ing.tu-bs.de/carolocup/system/files/Hauptwettbewerb2015.pdf>. [Online; accessed 04-September-2015]. 2014.
- [7] Lukas Maischak. „Lane Localization for Autonomous Model Cars“. In: (2014).
- [8] Naja v. Schmude. „Farb- und Kantenbasierte Objekterkennung Humanoider Roboter im RoboCup-Szenario“. In: (2011).
- [9] Irwin Sobel. „History and definition of the sobel operator“. In: *Retrieved from the World Wide Web* (2014).
- [10] Chao Ma und Mei Xie. „A method for lane detection based on color clustering“. In: *Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on*. IEEE. 2010, S. 200–203.
- [11] Robert Schaback und Holger Wendland. *Numerische Mathematik (Springer-Lehrbuch) (German Edition)*. Springer, 2004. ISBN: 3540213945.
- [12] Martin A Fischler und Robert C Bolles. „Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography“. In: *Communications of the ACM* 24.6 (1981), S. 381–395.

- [13] Philip HS Torr und David W Murray. „Outlier detection and motion segmentation“. In: *Optical Tools for Manufacturing and Advanced Automation*. International Society for Optics und Photonics. 1993, S. 432–443.
- [14] John C Clarke, S Carlsson und Andrew Zisserman. „Detecting and Tracking Linear Features Efficiently.“ In: *BMVC*. Citeseer. 1996, S. 1–10.
- [15] Etienne Vincent und Robert Laganiére. „Detecting planar homographies in an image pair“. In: *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*. IEEE. 2001, S. 182–187.
- [16] Roberto Toldo und Andrea Fusiello. „Automatic estimation of the inlier threshold in robust multiple structures fitting“. In: *Image Analysis and Processing–ICIAP 2009*. Springer, 2009, S. 123–131.
- [17] Marco Zuliani, Charles S Kenney und BS Manjunath. „The multiransac algorithm and its application to detect planar homographies“. In: *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. Bd. 3. IEEE. 2005, S. III–153.
- [18] David F Fouhey. „Multi-model Estimation in the Presence of Outliers“. Diss. Citeseer, 2011.
- [19] Roberto Toldo und Andrea Fusiello. „Robust multiple structures estimation with j-linkage“. In: *Computer Vision–ECCV 2008*. Springer, 2008, S. 537–547.
- [20] Lei Xu, Erkki Oja und Pekka Kultanen. „A new curve detection method: randomized Hough transform (RHT)“. In: *Pattern recognition letters* 11.5 (1990), S. 331–338.
- [21] Roberto Toldo und Andrea Fusiello. „Real-time incremental j-linkage for robust multiple structures estimation“. In: *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*. Bd. 1. 2. 2010, S. 6.
- [22] Shengyan Zhou u. a. „Road detection using support vector machine based on online learning and evaluation“. In: *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE. 2010, S. 256–261.
- [23] Yaniv Alon, Andras Ferencz und Amnon Shashua. „Off-road path following using region classification and geometric projection constraints“. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Bd. 1. IEEE. 2006, S. 689–696.
- [24] Juan M Collado u. a. „Detection and classification of road lanes with a frequency analysis“. In: *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*. IEEE. 2005, S. 78–83.

- [25] Christian Lipski u. a. „A fast and robust approach to lane marking detection and lane tracking“. In: *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*. IEEE. 2008, S. 57–60.
- [26] Johannes Clauberg. *Logica vetus et nova*. 1654.
- [27] Ubbo Visser u. a. *Robocup 2007: Robot soccer world cup xi*. Bd. 5001. Springer, 2008.
- [28] Klaus Jänich. *Mathematik 1: Geschrieben für Physiker (Springer-Lehrbuch) (German Edition)*. Springer, 2005. ISBN: 3540213929.
- [29] Christian Karpfinger. *Höhere Mathematik in Rezepten: Begriffe, Sätze und zahlreiche Beispiele in kurzen Lerneinheiten (German Edition)*. Springer Spektrum, 2015. ISBN: 3662438100.
- [30] Daniel Krakowczyk. „A path following control architecture for autonomous vehicles“. In: (2014).
- [31] Benjamin Zengin. „Kreuzungserkennung und -verhalten bei autonomen Modellfahrzeugen“. In: (2014).
- [32] Jannis Ihrig. „Obstacle Detection for an autonomous model car using stereo vision“. In: (2014).
- [33] Konstantinos G Derpanis. „Overview of the RANSAC Algorithm“. In: *York University, Toronto, Canada* (2010).

## Anhang



**Abb. 23:** Kurve mit Angaben zu den normierten Fahrbahnmarkierungen

---

### Algorithmus 2 : RANSAC[33]

---

**Data :** Datenmenge  $D$ , bestehend aus Punkten

**Result :** Instanz  $M^*$  des Modell  $M$ , falls gefunden

- 1: Wähle zufällig Teilmenge  $d$  mit Kardinalität gleich  $n$  aus  $D$ , wobei  $n$  gleich der minimalen Anzahl an Punkten ist, die für  $M$  benötigt werden
  - 2: Erstelle  $M^*$  aus  $d$
  - 3: Bestimme die Anzahl der Punkte  $S$  aus  $D$ , die einen maximalen Abstand von  $\epsilon$  zu  $M^*$  haben
  - 4: Wenn  $S$  größer als ein Schwellwert  $\mathcal{T}$  ist, gib  $M^*$  zurück
  - 5: Wiederhole sonst Schritt 1 bis 4, aber maximal  $N$  mal
-