



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Alexander Reimer

Entwicklung eines Messsystems und einer Methode für die statistische Vermessung von automatisierten Einparksystemen

*Fakultät Technik und Informatik
Department Fahrzeugtechnik und Flugzeugbau*

*Faculty of Engineering and Computer Science
Department of Automotive and
Aeronautical Engineering*

Alexander Reimer

**Entwicklung eines Messsystems und
einer Methode für die statistische
Vermessung von automatisierten
Einparksystemen**

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang M.Sc. Fahrzeugbau
am Department Fahrzeugtechnik und Flugzeugbau
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr. rer. nat. Rasmus Rettig
Zweitprüfer: Prof. Dipl. Ing. (FH) Gregor Schimming

Abgabedatum: 30.07.2018

Zusammenfassung

Alexander Reimer

Thema der Masterthesis

Entwicklung eines Messsystems und einer Methode für die statistische Vermessung von automatisierten Einparksystemen

Stichworte

Fahrzeugtechnik, autonomes Fahren, Robotik, Lokalisierung, Automatisierung, Trajektorien, Tracking, digitale Bildverarbeitung, digitale Fotografie, Assistenzsysteme

Kurzzusammenfassung

Diese Arbeit umfasst die Entwicklung einer Messmethode für die Quantifizierung von automatisierten Einparkvorgängen. Darüber hinaus werden Messergebnisse im Sinne eines Testverfahrens statistisch ausgewertet.

Abstract

Alexander Reimer

Title of the paper

Development of a measurement system and a method for statistical surveying of automated parking systems.

Keywords

Automotive engineering, autonomous driving, localisation, digital image processing, digital photography, automation, tracking, trajectories

Abstract

This thesis describes the development of a measurement method for quantification of automated parking of automobiles as well as an approach for statistical surveying of the quantities.

Danksagung

An dieser Stelle möchte ich bei allen bedanken, die mich während der Bearbeitung dieser Thesis motiviert und unterstützt haben. Allen die bei den Versuchen mitgewirkt haben, möchte ich für ihre tatkräftige Unterstützung bei der Vorbereitung und Durchführung danken.

Prof. Dr. rer. nat. Rasmus Rettig danke ich für die intensive Betreuung.

Außerdem möchte ich mich bei Prof. Dipl. Ing. (FH) Gregor Schimming für die Übernahme des Zweitgutachtens bedanken.

Schließlich möchte ich mich bei alle bedanken, die diese Arbeit Korrektur gelesen haben und mir damit eine große Hilfe bei der Fertigstellung waren.

Inhaltsverzeichnis

Abkürzungsverzeichnis.....	I
Abbildungsverzeichnis.....	II
Tabellenverzeichnis.....	IV
1. Einführung.....	1
2. Grundlagen	2
2.1. Automatisierte Einparksysteme	2
2.1.1. Automatisierungsgrade und Kategorisierung von Assistenzsystemen und Einparksystemen.....	2
2.1.2. Sensorik und Aktorik von Einparksystem.....	3
2.2. Bildverarbeitung	4
2.3. Kalibrierung von Kameras und Korrektur von Verzerrungen.....	5
2.4. Optik	6
2.4.1. Funktionsprinzip digitaler Bild-Aufnahmen.....	6
2.4.2. Aufbau und Funktion einer Kamera	7
2.4.3. Sensortypen.....	7
2.4.4. Linse	8
2.4.5. Blende.....	9
2.4.6. Objektiv	10
2.4.7. Vogelperspektive.....	10
2.4.8. Schärfentiefe.....	11
2.5. Definition Testverfahren	12
3. Anforderungsanalyse	13
3.1. Anforderungen an Testverfahren.....	17
3.1.1. Effektivitätskriterien	17
3.1.2. Effizienzkriterien.....	18
4. Systementwurf	19
4.1. Definition des Ziels.....	19
4.2. Eingliederung und Funktion des Messsystems.....	19
4.3. Statistische Versuchsplanung	20
4.4. Messmethodik	22
4.4.1. Endposition	23
4.4.2. Positionsbestimmung mit zwei Raum-Koordinaten.....	23
4.5. Architektur der Messtechnik	24
5. Realisierung	25
5.1. Versuchsfahrzeug	25
5.2. Applikation der Messtechnik an Versuchsfahrzeug.....	26
5.3. Sensorik.....	27

5.3.1.	Sphärische-Kamera FLIR Ladybug 5	27
5.3.2.	Seitenkameras und Vogelkamera.....	27
5.3.3.	Bosch Laser-Entfernungsmesser	28
5.3.4.	LIDAR Sensor Velodyne HDL-32E	28
5.3.5.	SBG Systems Apogee-D Inertial Navigation System	28
5.4.	Schnittstellen und Kommunikation.....	29
5.4.1.	Rechnereinheit des Versuchsfahrzeuges	29
5.4.2.	Seitenkameras	29
5.4.3.	Sphärische Kamera.....	29
5.4.4.	LIDAR-Sensor	29
5.5.	Teststrecke, Anordnung	30
5.5.1.	Anpassungen nach der ersten Messreihe	33
5.6.	Versuchsdurchführung	38
5.7.	Bestimmung der Zielmessgrößen.....	39
5.7.1.	Positionsbestimmung	41
5.7.2.	Funktionsprinzip Marker-Tracking für Bildaufnahmen.....	42
5.7.3.	Transformation der Bildposition in Raumkoordinaten	49
5.7.4.	Messung der Einparkdauer	54
5.7.5.	Zählung von Parkzügen	54
5.7.6.	Notierung von Abbruch nach eingeleitetem Einparkvorgang	54
5.7.7.	Messung des Lenkwinkels.....	54
5.7.8.	Messung des Abstandes zwischen Bordstein und Rad-Außenflanke	55
5.7.9.	Bestimmung des Orientierungswinkels am Ende des Einparkvorgangs.....	63
5.7.10.	Messung der Abstände zu Fahrzeugen	64
5.8.	Anforderungen an Testverfahren.....	66
5.9.	Auswertung der Messergebnisse	67
5.9.1.	Protokoll der ersten Messreihe.....	67
5.9.2.	Protokoll der zweiten Messreihe.....	68
5.9.3.	Abstand Rad-Bordstein	71
5.9.4.	Abstände zu geparkten Fahrzeugen.....	82
5.9.5.	Orientierungswinkel.....	90
5.9.6.	Parkzuganzahl	91
5.9.7.	Dauer des Einparkvorgangs.....	92
5.9.8.	Trajektorien	94
6.	Zusammenfassung.....	95
7.	Fazit	97
8.	Ausblick.....	98
9.	Anhang.....	100

9.1. Programmcode.....	100
9.1.1. Seitenkamas100	100
9.1.2. Kamera in Vogelperspektive.....111	111
10. Literaturverzeichnis.....	127

Abkürzungsverzeichnis

AVI	Audio Video Interleave
BAST	Bundesanstalt für Straßenwesen
CAN	Controller Area Network
CCD	Charge-Coupled Device
CMOS	Complementary metal-oxide-semiconductor
Fps	Frames per second
LIDAR	Light Detection And Ranging
RADAR	Radio Detection And Ranging
RMSE	Root mean square error
ROS	Robot Operating System
SAE	Society of Automotive Engineers
USB	Universal Serial Bus
VDA	Verband der Automobilindustrie

Abbildungsverzeichnis

Abbildung 1: Grade der Automatisierung und ihre Definition	2
Abbildung 2: Flussdiagramm für den Datenaustausch zwischen Einparksystem und Fahrzeug	3
Abbildung 3: Beispiel für eine Bild-Entzerrung (links Originalaufnahme, rechts entzerrtes Bild).....	5
Abbildung 4: Bayer-Matrix	6
Abbildung 5: Schematischer Aufbau einer Kamera.....	7
Abbildung 6: Linsenabhängige Größen	8
Abbildung 7: Blendenöffnung	9
Abbildung 8: Brennweite und Bildwinkel	10
Abbildung 9: Konstruktion des Schärfentiefebereichs.....	11
Abbildung 10: Schema zur Schärfentiefe	11
Abbildung 11: Kommunikation zw. Einparksystem, Versuchsfahrzeug und eingegliedertem Messsystem.....	19
Abbildung 12: Allgemeine Form der Normalverteilung.....	20
Abbildung 13: Anordnung der Sensorik am Versuchsfahrzeug (2. Entwurf).....	22
Abbildung 14: Lageplan der Versuchsanordnung bei der 1. Messreihe	31
Abbildung 15: Lageplan der Versuchsanordnung bei der 2. Messreihe	32
Abbildung 16: Position und Ausrichtung der Stativkonstruktion mit Kamera auf Gabelstapler	33
Abbildung 17: Bestimmung der Kameraposition in der x-y-Ebene.....	34
Abbildung 18: Lage und Orientierung der Weitwinkelkamera	35
Abbildung 19: Vogelkamera auf Stativ und Gabelstapler (Seitenansicht)	36
Abbildung 20: schematische Darstellung der gesuchten Zielmessgrößen	39
Abbildung 21: Darstellung von Verschiebungen detektierter "Harris-feature Points"-Paarungen an Orts-Markierung (1. Entwurf)	42
Abbildung 22: Darstellung von Verschiebungen detektierter "Surf-feature Points"-Paaren auf dem Versuchsfahrzeug (2. Entwurf)	42
Abbildung 23: Markierungs-Positions-Verlauf beim Marker Tracking (Messreihe 2, Messung 33)	43
Abbildung 24: Kamera-Bild-Aufnahme der ersten Messreihe	44
Abbildung 25: Beispiel für einen Startpunkt für die „Markermittelfinder“-Funktion.....	44
Abbildung 26: detektierte Zeilen- und Spaltenmittelpunkte.....	44
Abbildung 27: Lage der Markierung auf Fahrzeugdach (1. Entwurf)	46
Abbildung 28: Lage der Markierungen über Vorder- und Hinterachse (2. Entwurf).....	46
Abbildung 29: Markierung auf der Heckklappe des Versuchsfahrzeugs (links im Bild: Einklemmschutz)	47
Abbildung 30: Versuchsfeld-Markierungen	48
Abbildung 31: Kamera-Koordinatensystems	49
Abbildung 32: Bild-Koordinatensystem	49
Abbildung 33: Lage des globalen Koordinatensystems	49
Abbildung 34: Bildpunkanzahl über Breite und Höhe einer entzerrten Aufnahme.....	50
Abbildung 35: Räumliche Lage des ersten Ziel-Koordinatensystems auf der Fahrbahnebenen bei der Transformation von Bildpunkten in Raumkoordinaten	52
Abbildung 37: Lage der Seitenkamera zu Radmarkierungskegel	54
Abbildung 36: CAD-Modell vom entworfenen Radmarkierungskegel	54
Abbildung 38: Seitenansicht der Lage und Orientierung der Seitenkameras am Versuchsfahrzeug ...	55
Abbildung 39: Ausschnitt aus einer Aufnahme der hinteren Seitenkamera bei angelegtem Mess-Raster (1. Entwurf)	55
Abbildung 40: Verlauf von Intensitätswerten für gewählte Zählzeile	56
Abbildung 41: Seiten-Kamera-Aufnahme mit markiertem Bordstein (2. Entwurf).....	58
Abbildung 42: Orientierung und Lage der Seiten-Kameras am Versuchsfahrzeug	59
Abbildung 43: Bordstein-Querschnitt.....	59
Abbildung 44: Relativposition Seitenkameras zu Bordstein.....	59
Abbildung 45: Darstellung der Aufstandslinien am Versuchsfahrzeugs	60
Abbildung 46: Länge zwischen Bordsteinoberkante und -fuß bei Anstellwinkel der Kameras	60
Abbildung 47: Bordsteinschrägen-Geometrie in Schnittebene unter Blickwinkel der Kamera	60
Abbildung 48: Plot Intensitätswerte ab Reifenaufstandspunkt bis Bildrand (Messung 2, Heck).....	61
Abbildung 49: Plot Intensitätswerte ab Reifenaufstandspunkt bis Bildrand (Messung 4, Front)	62

Abbildung 50: Aufnahme der Front-Seitenkamera.....	64
Abbildung 51: manuelle Messungen des Abstandes Rad-Bordstein (1. Messreihe).....	71
Abbildung 52: Histogramm mit 10 Intervallen zu Abstandsmessung Vorderrad-Bordstein bei 1246mm Rangierraum.....	72
Abbildung 53: Histogramm zu Abstandsmessung Hinterrad-Bordstein bei 1246mm Rangierraum	73
Abbildung 54: Histogramm zu Abstandsmessung Vorderrad-Bordstein bei 1328mm Rangierraum	73
Abbildung 55: Kumulierte Häufigkeiten Für Messungen am Vorderrad bei 1246mm und 1328mm Rangierraum.....	74
Abbildung 56: Histogramm zu Abstandsmessung Hinterrad-Bordstein bei 1328mm Rangierraum	74
Abbildung 57: Kumulierte Häufigkeiten Für Messungen am Hinterrad bei 1246mm und 1328mm Rangierraum.....	75
Abbildung 58: Kumulierte Häufigkeiten Für Messungen am Vorderrad bei 1246mm und 2149mm Rangierraum.....	75
Abbildung 59: Kumulierte Häufigkeiten Für Messungen am Hinterrad bei 1246mm und 2149mm Rangierraum.....	76
Abbildung 60: Verlauf Bordsteinabstand über Rangierraum.....	77
Abbildung 61: Differenz aus Hinterradabstand - Vorderradabstand über Rangierraum	78
Abbildung 62: Häufigkeitsverteilung der Differenz aus Hinterrad-Abstand und Vorderrad-Abstand	79
Abbildung 63: Korrelation manueller Messungen mit Messungen mit Seitenkamera-Aufnahmen für den Frontabstand.....	82
Abbildung 64: Differenz aus Ergebnissen manueller Messungen und Ergebnissen bei Messungen mit Seitenkamera.....	83
Abbildung 65: Häufigkeitsverteilung für Differenzen der Front-Abstandsmessungen.....	84
Abbildung 66: Häufigkeitsverteilung für Differenzen der Heck-Abstandsmessungen.....	85
Abbildung 67: Endabstände der 2. Messreihe	87
Abbildung 68: Differenz der Messungen für Front- und Heckabstände (2. Messreihe).....	88
Abbildung 69: Häufigkeitsverteilung Orientierungswinkel (2. Messreihe)	90
Abbildung 70: Parkzuganzahl über Rangierraum.....	91
Abbildung 71: Einparkdauer über Rangierraum	92
Abbildung 72: Zusammenhang zwischen Parkdauer und Parkzuganzahl.....	93
Abbildung 73: Trajektorien aus allen vogelperspektivischen Aufnahmen der 2. Messreihe	94

Tabellenverzeichnis

<i>Tabelle 1: Komponenten von Einparksystemen und ihre Funktionen</i>	3
<i>Tabelle 2: Aus Endkundenanforderungen abgeleitete Kriterien</i>	14
<i>Tabelle 3: abgeleitete Anforderungen an das Messkonzept und die Messtechnik</i>	15
<i>Tabelle 4: Ableitung quantifizierbarer Kriterien aus Anforderungen an die Messtechnik</i>	16
<i>Tabelle 5: Berechnung benötigter Versuche zur Ermittlung von Normalverteilungen</i>	20
<i>Tabelle 6: Anzahl der benötigten Versuche zur Erreichung einer Normalverteilung</i>	21
<i>Tabelle 7: Architektur der Messtechnik</i>	24
<i>Tabelle 8: Herstellerangaben zum Versuchsfahrzeug aus Handbuch des Fahrzeugs und Fahrzeugschein</i>	25
<i>Tabelle 9: Reifen-Außenflanken-Abstände des Versuchsfahrzeugs</i>	25
<i>Tabelle 10: Spezifikationen zur Kamera FLIR Ladybug 5</i>	27
<i>Tabelle 11: Spezifikationen zur Kamera Basler acA 1920-40um/uc</i>	27
<i>Tabelle 12: Spezifikationen zum Objektiv Kowa LM6HC</i>	27
<i>Tabelle 13: Spezifikationen zum Objektiv Kowa LM6HC</i>	27
<i>Tabelle 14: Öffnungswinkel der verwendeten Kameras</i>	28
<i>Tabelle 15: Spezifikationen des Bosch-Laserentfernungsmessers</i>	28
<i>Tabelle 16: Spezifikationen des LIDAR-Sensors</i>	28
<i>Tabelle 17: Spezifikationen des Inertial Navigation Systems</i>	28
<i>Tabelle 18: Übersicht der bei den Messungen eingesetzten Komponenten</i>	39
<i>Tabelle 19: Übersicht über die Methoden zur Ermittlung der Zielmessgrößen</i>	40
<i>Tabelle 20: Auflösung der Messmethode Transformation vogelperspektivische Aufnahmen</i>	53
<i>Tabelle 21: Auflösung für Orientierungswinkel aus vogelperspektivischen Aufnahmen</i>	53
<i>Tabelle 22: Messbereich der Messmethode zur Bestimmung des Bewegungsverlaufs durch Auswertung vogelperspektivischer Aufnahmen</i>	53
<i>Tabelle 23: Vergleich geometrischer Transformation von Bildkoordinaten mit anderen Messmethoden</i>	57
<i>Tabelle 24: Auflösungen der unterschiedlichen Messprinzipien zur Bestimmung des Abstandes zwischen Rad und Bordstein</i>	57
<i>Tabelle 25: Maßnahmen zur Erfüllung von Anforderungen an Testverfahren</i>	66
<i>Tabelle 26: Protokoll der 1. Messreihe</i>	67
<i>Tabelle 27: Protokoll der 2. Messreihe (Teil 1)</i>	69
<i>Tabelle 28: Protokoll der 2. Messreihe (Teil 2)</i>	70
<i>Tabelle 29: Messergebnisse für Abstand Vorderrad-Bordstein (1. Messreihe)</i>	71
<i>Tabelle 30: Messergebnisse für Abstand Hinterrad-Bordstein (1. Messreihe)</i>	71
<i>Tabelle 31: Vergleich manuelle Messung vs. Messung mit Seitenkamera</i>	80
<i>Tabelle 32: Vergleich Orientierungswinkel aus manuellen Messungen und Messungen mit Seitenkamera-Aufnahmen</i>	90
<i>Tabelle 33: Messbereich, Genauigkeit und Auflösung des Messkonzeptes</i>	96

1. Einführung

Im Zuge der Automatisierung von Fahrzeugen haben Systeme wie Notbremsassistenten, Tempomaten mit adaptiver Abstandsregelung, Spurhalteassistenten und Totwinkelassistenten in Serienfahrzeuge Einzug gefunden. Eines der größten Potenziale, die in assistiertem, automatisiertem oder autonomen Fahren gesehen wird, ist darin begründet, dass ein Großteil aller Verkehrsunfälle auf menschliches Versagen zurückzuführen ist (Maurer u. a. 2015, S. 310). Deshalb kann es als erstrebenswert angesehen werden Systeme zur Unterstützung des Fahrers oder sogar als Ersatz für den menschlichen Fahrer weiterzuentwickeln um damit Verkehrsunfälle dieser Art zu vermeiden. Erkenntnisse aus der Entwicklung und Verbesserung von aktuellen Assistenzsystemen in Fahrzeugen können zur gesellschaftlichen Akzeptanz automatisierter und autonomer Fahrzeuge beitragen. Ein weiterer potentieller Vorteil des autonomen Fahrens besteht darin, dass durch autonome Fahrzeuge Verkehrs- und Parkflächen in andere Nutzungen überführt werden könnten (Maurer u. a. 2015, S. 236). Außerdem wäre es möglich die Zeit während der Fahrt anderweitig zu nutzen. Da das autonome Fahren diese Vielfalt an potenziellem Nutzen verspricht, soll diese Arbeit einen Beitrag zur Weiterentwicklung automatisierter Fahrzeugführungs-Systeme leisten. Dazu soll ein Messsystem entwickelt werden, welches nach dem Prinzip Reverse Engineering die Bewegungen eines Fahrzeugs erfasst, während dieses durch ein vollautomatisches Einparksystem geführt wird. Dabei sollen sowohl Erkenntnisse aus der Entwicklung des Messsystems, als auch aus den Messergebnissen zukünftigen Arbeiten als Referenz dienen.

Die vorliegende Arbeit ist am Urban Mobility Lab der HAW Hamburg im Zeitraum 1. Februar 2018 bis 30. Juli 2018 entwickelt und verfasst worden.

2. Grundlagen

2.1. Automatisierte Einparksysteme

2.1.1. Automatisierungsgrade und Kategorisierung von Assistenzsystemen und Einparksystemen

Fahrzeuge können nach dem Grad der Automatisierung kategorisiert werden. Dazu kann, z. B. die Einteilung durch den Verband der Automobilindustrie VDA als Orientierung dienen. Diese Einteilung ist von der Society of Automotive Engineers (SAE) und der Bundesanstalt für Straßenwesen (BASt) übernommen worden.

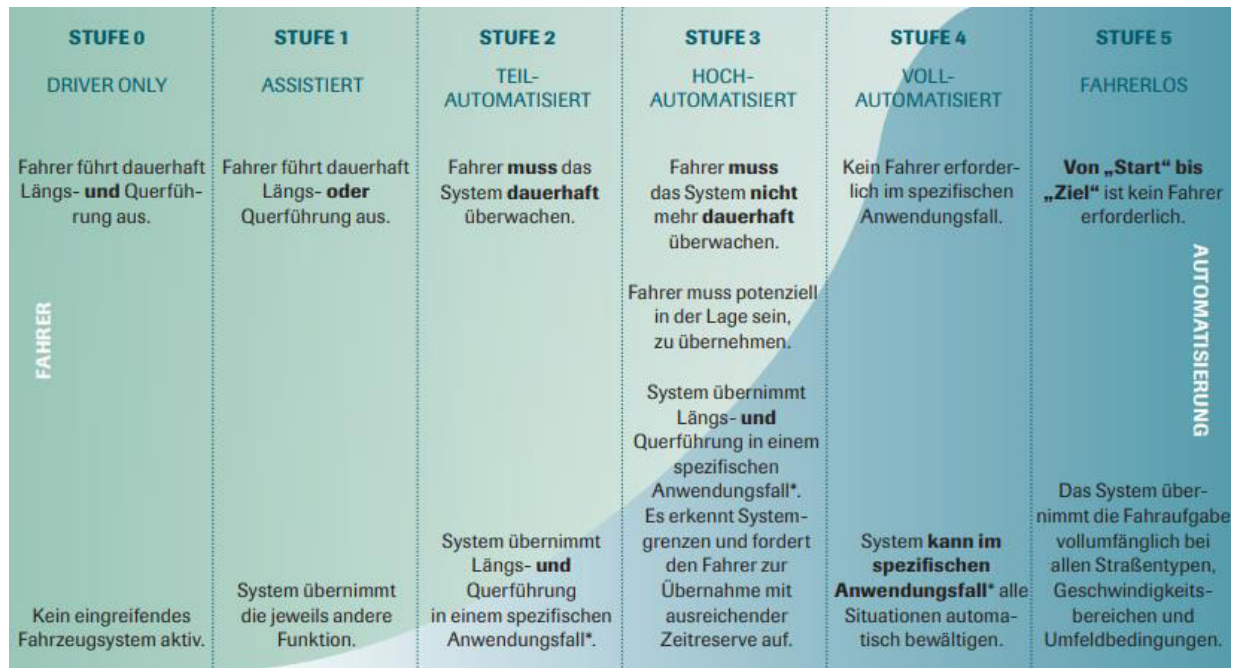


Abbildung 1: Grade der Automatisierung und ihre Definition

Quelle: („SAE Levels of Driving Automation“ o. J.; „VDA“ o. J.; „BASt 2017 - Forschung kompakt“ 2017)

Analog dazu lassen sich Kategorien für Einparkassistenten ableiten. Einparkassistenten werden in informierende, führende, semiautomatische und vollautomatische Systeme eingeteilt (Winner, Hakuli, und Lotz 2015, S. 842). Während informierende Systeme dem Fahrer lediglich Informationen, z. B. zu Abständen zu Hindernissen übermitteln, geben führende Systeme Handlungsempfehlungen, wie Einlenken, Vorfahren oder Zurücksetzen, z. B. durch Anzeige von Hilfslinien in der Anzeige von Bildern einer Rückfahrkamera. Semiautomatische übernehmen die Quer- oder Längsführung des Fahrzeugs, während der Fahrer den anderen Teil übernehmen muss. Vollautomatische Einparksysteme übernehmen sowohl die Quer-, als auch Längsführung des Fahrzeugs. Ein Fahrzeug, das mit solch einem System ausgestattet ist, kann der Grafik nach als teil- bis hochautomatisiert eingestuft werden.

Gegenstand dieser Arbeit ist die Entwicklung eines Messkonzeptes für vollautomatische Einparksysteme. Anhand der Ergebnisse soll die Qualität der Ergebnisse des behandelten Systems statistisch ausgewertet werden und Aussagen über die Reproduzierbarkeit der Ergebnisse gemacht werden. Der folgende Abschnitt behandelt Funktionen und Komponenten von Einparksystemen.

2.1.2. Sensorik und Aktorik von Einparkssystem

Automatische Einparkssysteme, die Quer- und/oder Längsführung des Fahrzeugs übernehmen, erfassen durch Auswertung aller für den Einparkvorgang benötigten Sensordaten den Ist-Zustand. Darüber hinaus wird ein Soll-Zustand berechnet und um diesen zu erreichen, werden Befehle an die Fahrzeugsteuerung gesendet. Das folgende Diagramm zeigt beispielhaft den Austausch von dabei relevanten Daten.

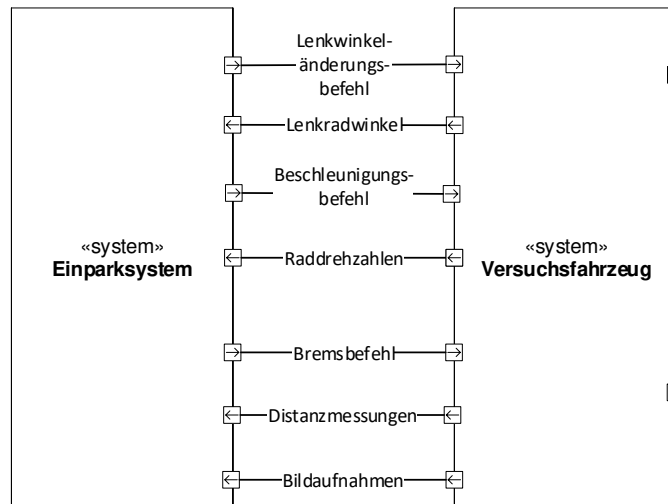


Abbildung 2: Flussdiagramm für den Datenaustausch zwischen Einparksystem und Fahrzeug

Die Daten zu der Umgebung werden dabei meist über Ultraschallsensoren oder Kameras generiert. Weitere Komponenten, welche für den Austausch der Daten sowie für die Kommunikation mit dem Fahrer sind (vgl. (Winner, Hakuli, und Lotz 2015, S. 848, 663):

Komponente	Funktion
Steuergerät des Einparksystems	Implementierung der Funktionen (inkl. Quer- und Längsregelung)
Taster	Aktivieren des Systems
Drehzahlsensoren	Lokalisierung des Fahrzeugs
Lenkwinkelsensor	Lokalisierung des Fahrzeugs
Längs- und Querschleunigungssensoren	Lokalisierung des Fahrzeugs
Seitliche Ultraschallsensoren oder Short-Range-Radar	Vermessung von Parklücken
Ultraschallsensoren oder Short-Range-Radar an Heck und Front	Distanzmessungen
Blinklichtschalter	Auswahl der Parklückenseite durch den Fahrer
Steuergerät für Anhängererkennung	Anhängererkennung
Warnsummer der Einparkhilfe	Fahrer informieren
Elektromechanische Lenkung	Steuerung der Lenkung
Steuergerät der Bremse	Geschwindigkeitsinformation
Kameras an Heck und Front Grafikmodul	grafische Ausgabe von Informationen über das Umfeld über Display

Tabelle 1: Komponenten von Einparksystemen und ihre Funktionen

Die Daten der Sensoren werden bei vollautomatischen Einparksystemen dazu verwendet, die relative Position des Fahrzeugs zur Parklücke, seine Orientierung sowie Abstände zu Objekten in der Umgebung zu berechnen. Mit den berechneten Daten können Steuerungs-Strategien berechnet werden, um das Fahrzeug in die Parklücke zu manövrieren.

2.2. Bildverarbeitung

Die von einer Kamera erzeugten Bilder können als Matrizen abgespeichert werden. Bei Farbbildern haben diese Matrizen 3 Dimensionen, wobei die erste dem Spaltenindex, die zweite dem Zeilenindex und die dritte dem Index des Farbkanals R (rot), G (grün) oder B (blau) entspricht. Jeder Bildpunkt hat für jeden der Farbkanäle einen Intensitäts-Wert.

Außerdem können Schwarzweiß-Bilder mit nur einem Farbkanal eingelesen werden oder RGB-Bilder in solche umgewandelt werden („Convert RGB image or colormap to grayscale - MATLAB rgb2gray - MathWorks Deutschland“ o. J.). Dabei gibt der Intensitätswert die Helligkeit eines Bildpunktes wieder und wird häufig durch einen entsprechend helleren Grauwert bei größeren Werten oder einen dunkleren Grauwert bei kleineren Werten dargestellt. Bei der Umwandlung von RGB-Bildern in Graustufen-Bilder wird der Intensitätswert jedes Bildpunktes aus den drei Werten der Farbkanäle berechnet.

Für die Bildverarbeitung gibt es viele Ansätze zum automatisierten finden von Ecken, Kanten oder anderer markanter Punkte in einem Bild (Canny 1986; Torre und Poggio 1986; Zhang, Zhao, und Su 2005; Muthugnanambika und Padmavathi 2017; Rosten und Drummond 2005; „Detect SURF features and return SURFPoints object - MATLAB detectSURFFeatures - MathWorks Deutschland“ o. J.; „Detect corners using Harris–Stephens algorithm and return cornerPoints object - MATLAB detectHarrisFeatures - MathWorks Deutschland“ o. J.). Dabei werden z. B., Intensitätsgradienten im Bild ausgewertet. Je nach Vorgehensweise und Parametrisierung für Schwellwerte, Abtastfrequenz und anderen Faktoren werden dabei für ein Bild mehr oder weniger Punkte detektiert.

Videos bestehen aus Bildersequenzen, sodass die Einzelbilder extrahiert werden können. Diese oder einzeln aufgenommene Bilder können im sog. „Header“ Informationen, wie einen Zeitstempel, das Datenformat, Zeichenkodierung enthalten. Diese Informationen können ausgelesen werden.

Grafische Darstellungen jeglicher Größen, die in Matrizen geschrieben werden, können mit sogenannten Plots ausgegeben werden. Dabei kann eine oder mehrere Größen gegen eine andere oder gegen einen Index abgetragen werden.

2.3. Kalibrierung von Kameras und Korrektur von Verzerrungen

Die Kalibrierung von Kameras ist notwendig, um mit Kameras messen zu können (Kannala und Brandt, o. J.; „Measuring Planar Objects with a Calibrated Camera - MATLAB & Simulink - MathWorks Deutschland“ o. J.). Eine Kamera kann mit unterschiedlichen Objektiven ausgestattet werden, sodass Aufnahmen, je nach Ausführung des Objektivs, unterschiedliche Arten und Ausprägungen von Verzerrungen aufweisen können (Bühler, Schlaich, und Sinner 2017, S. 54; Jähne 2012, S. 97).

Für den Fall das eine Kamera ohne Verdrehung auf eine gerade Linie gerichtet ist, führt Verzerrung dazu, dass Punkte der Linie im Bild nicht auf der gleichen Höhe liegen. Damit gerade Linien im Bild auch gerade erscheinen, müssen die aufgenommenen Bilder entsprechend korrigiert werden. Dieser Vorgang wird als Kompensation oder Entzerrung bezeichnet. Bei der Kalibrierung kann z. B., ein Schachbrettmuster mit bekannten Kantenlängen mit der zu kalibrierenden Kamera mit aufgesetztem Objektiv aufgenommen werden. Durch einen Abgleich der Lage der Eckpunkte zueinander im Bild mit ihrer Lage zueinander auf dem Muster kann berechnet werden, wie das Bild zu korrigieren ist. Diese Korrekturparameter können ebenso, weitere Aufnahmen der kalibrierten Kamera korrigieren. Das Ergebnis am Beispiel einer Aufnahme mit einem Objektiv der Brennweite 6,5 mm ist im folgenden Bild zu sehen.



Abbildung 3: Beispiel für eine Bild-Entzerrung (links Originalaufnahme, rechts entzerrtes Bild)

2.4. Optik

2.4.1. Funktionsprinzip digitaler Bild-Aufnahmen

Kamera-Sensoren wandeln Licht in elektrische Signale um (vgl. zu diesem Abschnitt (Jähne 2012)).

Dazu wird der sogenannte photoelektrische Effekt (Tipler und Mosca 2015, S. 1162) genutzt, demzufolge Photonen Ladungsträger aus elektrisch leitfähigen Materialien herauslösen können.

Um diesen Effekt zu messen, hat sich der Einsatz von Halbleitern in Bildsensoren durchgesetzt (Jähne 2012, S. 197). Bei Messung der Leitfähigkeit wird die Lichtintensität berechnet. Diese Messung erfolgt für jeden Bildpunkt des Sensors, sodass eine Szene, bzw. ein Gegenstand digital mit einer durch die Anzahl der Bildpunkte beschränkten Auflösung abgebildet werden kann.

Bei der Aufnahme wird im Sensor die Farbinformation direkt in die drei Teilfarben Rot, Grün und Blau aufgeteilt. Sensorelemente und Signalverarbeitung der Chips sind heute meist nach der BayerMatrix (Abbildung 4) angeordnet. Diese ist entsprechend der Empfindlichkeits-Eigenschaften des menschlichen Auges mit Filterschichten belegt. Dabei sind 50% der Sensoren mit einer grünen und jeweils 25% mit roten und blauen Filterschicht belegt, wobei diese gleichmäßig über die Sensorfläche verteilt sind.

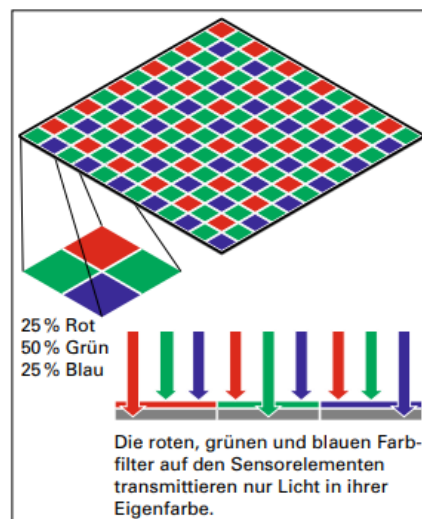


Abbildung 4: Bayer-Matrix

Quelle: (Bühler, Schlaich, und Sinner 2017, 30)

Während die Aufnahme berechnet wird, ist das Bild zunächst in einem Zwischenspeicher abgelegt. Anschließend werden die Teilfarbinformationen zu einem dreifarbigem Pixel zusammengerechnet.

Zur Berechnung des Bildes zählen z. B. Farbwerte, Weißabgleich sowie Scharfzeichnung. Dabei unterscheiden sich die Algorithmen der Chip-Hersteller. Endgültig wird das Bild im Raw-Format oder in einem komprimierten (z. B. JPEG) auf einem Speichermedium abgespeichert.

2.4.2. Aufbau und Funktion einer Kamera

Eine Kamera besteht aus einem lichtempfindlichen Bildsensor mit einem Objektiv, bestehend aus einer oder mehreren Linsen und einer Blende. Der Bildsensor ist im Gehäuse hinter dem Objektiv angeordnet. In der folgenden Grafik ist der schematische Aufbau einer Kamera dargestellt. Im nächsten Abschnitt wird zuerst auf das Funktionsprinzip digitaler Bildaufnahmen eingegangen. Danach folgen Erklärungen zu den Funktionen der einzelnen Elemente und wichtiger Parametern digitaler Bildaufnahmen.

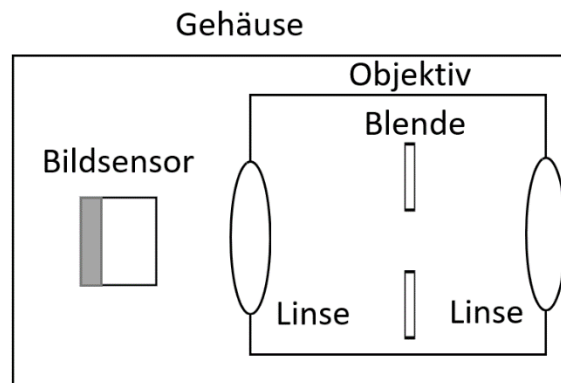


Abbildung 5: Schematischer Aufbau einer Kamera

2.4.3. Sensortypen

Es folgen Beschreibungen der beiden in dieser Arbeit eingesetzten Sensortypen. Die Informationen sind dabei der Literatur entnommen (vgl. (Bühler, Schlaich, und Sinner 2017, S. 30f; Keferstein, Marxer, und Bach 2018, S. 183f)). Bei beiden Sensortypen wird Licht in ein elektrisches Signal umgewandelt.

CCD Sensoren

CCD steht für „Charge-Coupled Device“ und ist ein lichtempfindlicher Sensorchip in Digitalkameras. Die durchschnittliche Kantenlänge liegt bei $10\mu\text{m}$ und die Sensorfläche wird seriell zeilenweise ausgelesen. Die Umwandlung von Analog zu Digital werde durch eine einzige Schaltung bewerkstelligt, wodurch die Aufnahmen rauschärmer seien und der Sensor im Vergleich zu anderen Sensortypen eine größere lichtempfindliche Fläche habe.

CMOS Sensoren

Bei „Complementary Metal Oxide Semiconductor“- Sensoren sind die einzelnen Elemente (d.h. Pixel) einzeln adressierbar, was eine schnellere Bildverarbeitung der Signale und damit schnellere Bildfolgen ermöglicht. Dieser Sensortyp stelle die hochwertigere Variante zu CCD dar und habe einen geringeren Stromverbrauch.

2.4.4. Linse

In folgender Abbildung sind Größen abgebildet, welche den geometrischen Zusammenhang zwischen der Bildebene (rechts im Bild), der Hauptebene (Mitte des Bildes) sowie der Gegenstandsebene (links im Bild) beschreiben (für diesen Abschnitt vgl. (Bühler, Schlaich, und Sinner 2017, S. 55)). Dabei liegt in der Bildebene der Sensor der Kamera, für den Fall dass nur eine Linse vorhanden ist liegt diese in der Hauptebene und in der Gegenstandsebene liegt der betrachtete Gegenstand.

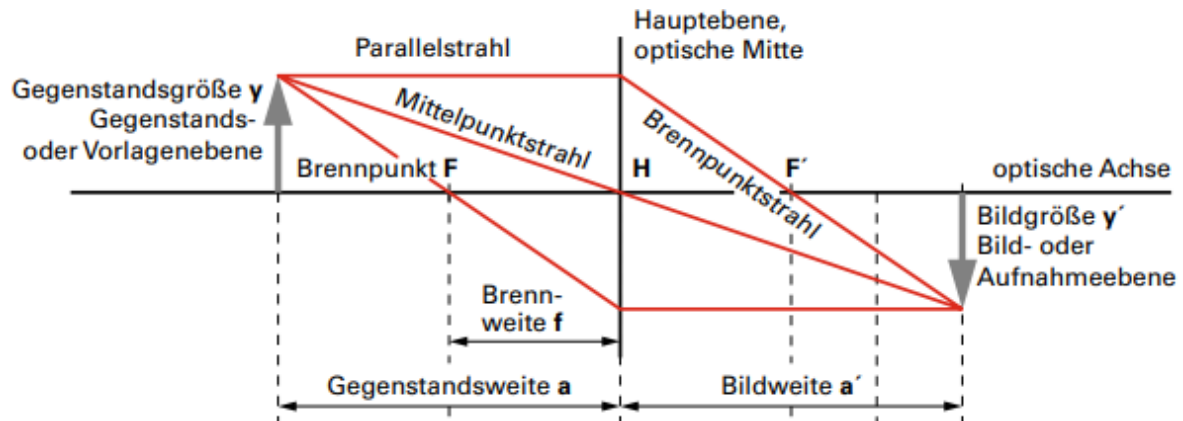


Abbildung 6: Linsenabhängige Größen

Quelle: (Bühler, Schlaich, und Sinner 2017, S. 55)

Optische Achse: Die optische Achse ist die Symmetrieachse der Linsen. Auf ihr steht senkrecht die Hauptebene.

Hauptebene, Hauptpunkt: Objektive haben eine gegenstandsseitige und eine bildseitige Hauptebene. Von ihr aus wird jeweils die Brennweite, Gegenstands- und Bildweite gerechnet. Die Hauptebene schneidet im Hauptpunkt H die optische Achse.

Brennweite, Brennpunkt: Die Brennweite f ist der Abstand des Brennpunkts F vom Hauptpunkt H. Im Brennpunkt treffen sich die von einer Sammellinse gebrochenen Strahlen.

Gegenstandsweite, Bildweite: Die Gegenstandsweite a ist der Abstand zwischen Objekt y und dem Hauptpunkt H. Die Bildweite a' ist die Entfernung des bildseitigen Hauptpunkts zum Bild y' .

Die Öffnungswinkel der Kamera werden entsprechend dieser Zusammenhänge wie folgt berechnet:

$$\text{Vertikaler Öffnungswinkel: } \alpha_v = 2 \cdot \arctan\left(\frac{b_s}{2 \cdot f}\right) \quad (1)$$

$$\text{Horizontaler Öffnungswinkel: } \alpha_h = 2 \cdot \arctan\left(\frac{h_s}{2 \cdot f}\right) \quad (2)$$

b_s : Sensorbreite [mm]

h_s : Sensorhöhe [mm]

f : Brennweite des Objektivs [mm]

2.4.5. Blende

Als Blende wird eine verstellbare Öffnung des Objektivs bezeichnet (vgl. zu diesem Abschnitt (Bühler, Schlaich, und Sinner 2017, S. 37)). Die Größe der Öffnung bestimmt wie viel Licht auf die Bildebene fällt. Die Blendenzahl k gibt den Kehrwert der relativen Öffnung in folgender Form wider.

$$k = \frac{d}{f} \quad (3)$$

k: Blendenzahl

f: Brennweite [mm]

d: Objektivdurchmesser [mm]

Folgende Abbildung soll den Zusammenhang grafisch verdeutlichen. Bei Kameraobjektiven wird die Blendengröße durch Blendenzahlen der internationalen Blendenreihe angegeben. Diese beginnt bei 1 und erhöht sich jeweils um den Faktor 1,4, was einer Verringerung des Blendendurchmessers um Wurzel 2 und damit einer Halbierung der Öffnungsfläche entspricht. Damit gilt der Zusammenhang einer kleinen Blendenzahl mit einer großen Blendenöffnung und umgekehrt.

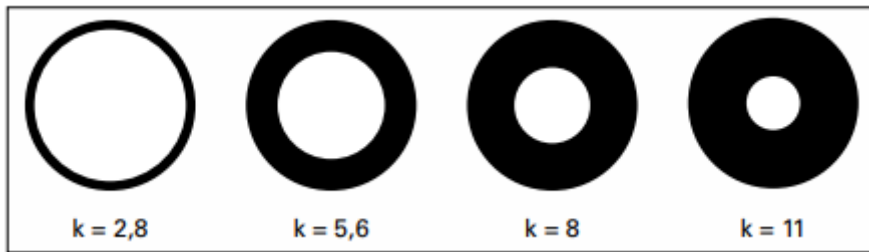


Abbildung 7: Blendenöffnung

Quelle: (Bühler, Schlaich, und Sinner 2017, 37)

Bei einer weiter geschlossenen Blende, d.h. größerer Blendenzahl, muss für die gleiche Helligkeit der Aufnahme länger belichtet werden. Bei Videoaufnahmen gilt es einen Kompromiss zu finden, da eine höhere Blendenzahl eine höhere Schärfentiefe zur Folge hat (siehe folgender Abschnitt). Dagegen nimmt die benötigte Belichtungszeit zu, wodurch die Aufnahme rate der Bilder (Fps = Frames per second) beschränkt wird. Dieses kann bei der Beobachtung von schnell bewegten Objekten dazu führen, dass die Bewegung nicht hinreichend hoch aufgelöst verfolgt wird. Im Extremfall kann eine zu untersuchende Bewegung, die während der Aufnahme eines Bildes startet, im nächsten Bild bereits beendet sein. Dadurch wäre eine Untersuchung des Bewegungsverlaufs nur sehr beschränkt möglich.

2.4.6. Objektiv

Ein Objektiv besteht aus Linsensystem und einer Blende, beides kann verstellbar ausgeführt sein (vgl. zu diesem Abschnitt (Bühler, Schlaich, und Sinner 2017, S. 35)). Die Einteilung der Objektive erfolgt nach der Brennweite in Weitwinkel-, Normal- und Teleobjektive. Dabei hängen mehrere fotografische Parameter von der Brennweite ab und Objektive mit Brennweiten von weniger als 25mm werden als Weitwinkelobjektive bezeichnet (Gockel 2012, S. 19). Der Bildwinkel bestimmt den Bildausschnitt der Abbildung. Dieser nimmt mit abnehmender Brennweite zu, wie in Abbildung 8 dargestellt ist.

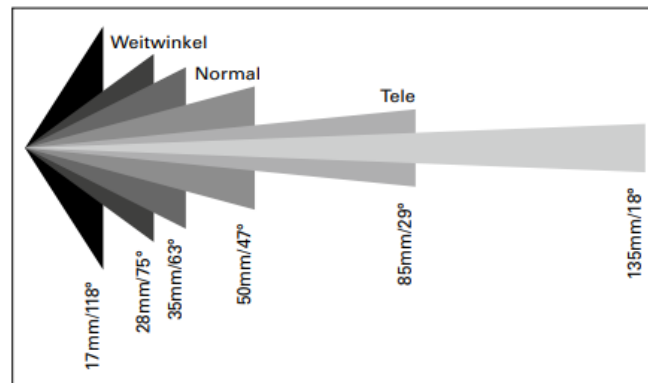


Abbildung 8: Brennweite und Bildwinkel

Quelle: Bühler, Schlaich, und Sinner 2017, S. 36

2.4.7. Vogelperspektive

Als Vogelperspektive wird in Anlehnung an die Literatur (Bühler, Schlaich, und Sinner 2017, S. 10) eine Ausrichtung der Kamera bezeichnet, die unter einem kleineren oder größeren Winkel auf den Boden gerichtet ist. Wenn in dieser Arbeit von vogelperspektivischen Aufnahmen oder Vogel-Kameras gesprochen wird, ist die Kamera mit einer solchen Ausrichtung oder die mit ihr erzeugte Aufnahmen gemeint.

2.4.8. Schärfentiefe

Bild- oder Aufnahmeebene (=Sensorposition) sowie Aufnahmegegenstand sind in ihrer Position fest (für diesen Abschnitt vgl. (Bühler, Schlaich, und Sinner 2017, S. 38)). Deshalb wird zum Scharfstellen die Hauptebene (siehe Abbildung 6) im Objektiv verschoben, bis der Aufnahmegegenstand in der Aufnahme scharf erscheint. Danach werden alle Objekte in dieser Einstellungsebene scharf abgebildet. Um die Einstellungsebene herum gibt es einen Bereich, in dem Objekte durch den Menschen noch scharf erscheinen. Ab einem bestimmten Abstand von der Einstellungsebene zur Aufnahmeebene erreicht die Unschärfe ein für Menschen unakzeptables Maß. Dieser Punkt wird als Nahpunkt bezeichnet. Dieser kann durch den Zerstreungskreis-Durchmesser (siehe folgende Abbildung) berechnet und bewertet werden. Auf der Seite des Aufnahmegegenstandes nennt man den entsprechenden Punkt Fernpunkt. In folgender Abbildung ist der Zusammenhang dargestellt.

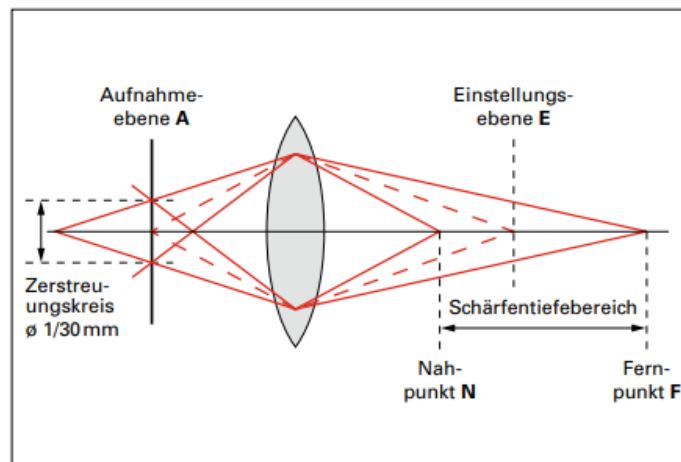


Abbildung 9: Konstruktion des Schärfentiefebereichs

Quelle: (Bühler, Schlaich, und Sinner 2017, S. 38)

Die Schärfentiefe hängt von der Blendenöffnung, der Brennweite sowie dem Abstand der Kamera zum Aufnahmeobjekt (=Aufnahmeabstand) ab, sodass folgende Zusammenhänge gelten:

- Je kleiner die Blendenöffnung, desto größer ist die Schärfentiefe.
- Je kürzer die Brennweite, desto größer ist die Schärfentiefe.
- Je kürzer der Aufnahmeabstand, desto geringer ist die Schärfentiefe.

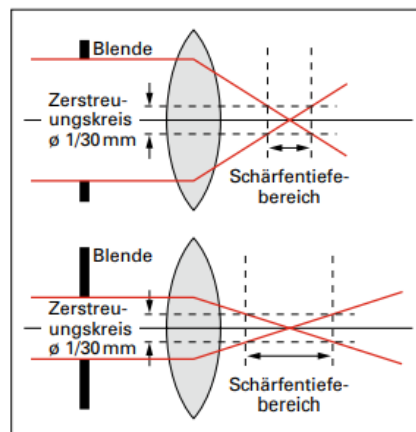


Abbildung 10: Schema zur Schärfentiefe

Quelle: (Bühler, Schlaich, und Sinner 2017, S. 36)

2.5. Definition Testverfahren

Das zu entwickelnde Messsystem soll Ergebnisse liefern, die im Sinne eines Testverfahrens nutzbar sind. Dazu folgt ein Zitat für eine Definition von Testverfahren aus der Literatur (Winner, Hakuli, und Lotz 2015, S. 168):

„Der Begriff Testverfahren bezeichnet eine Methode, nach der ein Test eines Systems auf bestimmte Eigenschaften durchzuführen ist. Hierzu sind auch erforderliche Werkzeuge, Hilfsmittel, Randbedingungen und Auswertemethoden festzulegen. Testverfahren sind ein wesentliches Werkzeug, um zu prüfen, ob gewünschte Produkteigenschaften vorhanden sind, was selbstverständlich auch während der Entwicklung von Fahrerassistenzsystemen gilt. Es existieren entwicklungsbegleitende und freigebende Tests, die im Wesentlichen in Eigenregie vom Fahrzeug- oder Systemhersteller durchgeführt werden. Ferner gibt es Tests, die im Sinne einer unabhängigen Produktprüfung von externen Testorganisationen vorgenommen werden – sei es für die Genehmigung von Fahrzeugtypen zu Zulassung zum Markt, im Rahmen der Anwendung der Norm zur funktionalen Sicherheit (in beiden Fällen beispielsweise durch technische Dienste) oder für die Kundeninformation, dann durchgeführt von Testinstituten für Verbraucherschutz.“

Die Messergebnisse sollen eine Bewertung darüber, ob das Einparksystem im Sinne des Endkunden gewünschte Eigenschaften hat, ermöglichen. Diese Arbeit kann im Sinne der Definition als eine unabhängige Produktprüfung angesehen werden.

3. Anforderungsanalyse

In diesem Kapitel werden die Anforderungen an automatisierte Einparksysteme aus der Sicht des Endkunden formuliert. Aus diesen werden Anforderungen an das Messkonzept sowie letztlich quantifizierbare Kriterien abgeleitet. Dabei werden nur vollautomatische Einparksysteme betrachtet, d.h. Systeme die Parklücken detektieren und auf Befehl die Quer- und Längsdynamik des Fahrzeugs übernehmen und das Fahrzeug vollautomatisch in die Parklücke manövrieren. In der folgenden Tabelle sind direkte Endkundenanforderungen aufgelistet. In den darauffolgenden Spalten stehen die abgeleiteten Kriterien, welche prüf- oder messbar sind.

Aus diesen Kriterien werden wiederum, Anforderungen an das Messkonzept und an die Messtechnik sowie quantifizierbare Kriterien abgeleitet. Für diese Kriterien werden an dieser Stelle Messbereich, benötigte Genauigkeit, Auflösung und Skala für die Bewertung festgelegt. Abschließend wird die Lösung zur Messung der Kriterien in der letzten Spalte beschrieben. Zur besseren Übersicht ist die Tabelle so aufgeteilt, dass 2-5 Spalten pro Seite nebeneinander aufgeführt sind. Auf der jeweils folgenden Seite ist dann noch einmal die letzte Spalte der vorherigen Seite als erste Spalte aufgeführt, sodass die Herleitung der Kriterien Schritt für Schritt dargestellt ist.

Kriterium	Endkundenanforderungen	abgeleitete Kriterien
Qualität des Einparkvorgangs	zufriedenstellende Parkendposition	<p>Abstände beider Rad-Außenflanken zu Bordstein gleich groß (tolerierete Abweichung $\leq 0,02$ m)</p> <p>Abstände beider Rad-Außenflanken zu Bordstein $\leq 0,07$ m</p> <p>Abstand zu Objekten vor und hinter der Parklücke $\geq 0,23$m, falls Rangiererraum $\geq 0,5$ m</p> <p>Abstand zu einem der Objekte $\leq 0,4$ m, falls Rangiererraum $\geq 1,1$ m (Aufrücken)</p> <p>Lenkwinkel an gelenktem Rad $\leq 3^\circ$</p>
	Bei gleichen Bedingungen (=gleiche Startposition, gleiche Parklücke) gleiche Parkendposition erreichen	<p>Abstände beider Rad-Außenflanken zu Bordstein sind unter gleichen Bedingungen (=gleiche Startposition, gleiche Parklücke) am Ende des Einparkvorgangs konstant (tolerierete Abweichung $\leq 0,02$ m)</p> <p>Parkendposition ist unter gleichen Bedingungen konstant (tolerierete Abweichung $\leq 0,07$ m)</p>
Mehrwertterhöhung	Minimierung Parkzugangszahl	<p>Minimierung Parkzugangszahl</p> <p>Definition: Das erste Anfahren der Parklücke ist bereits ein Parkzug, jede danach folgende Schubumkehrung des Fahrzeugs ist ein weiterer Parkzug.</p>
	<p>möglichst zügiger Einparkvorgang</p> <p>Ausschließen eines Abbruchs des Einparkvorgangs nach Bestätigung einer gefundenen Parklücke</p> <p>Erhöhung des Mehrwertes durch Weiterentwicklung des Systems</p>	<p>Minimierung der Einparkdauer</p> <p>Abbruch eingeleiteten Einparkvorgangs ausschließen</p> <p>Erfüllung gleicher Kriterien für unterschiedlicher Einparksysteme am Markt</p> <p>Minimierung der Summe der Fehlerquadrate beim Vergleich zweier räumlicher Bewegungsverläufe mit gleichen Startbedingungen</p>

Tabelle 2: Aus Endkundenanforderungen abgeleitete Kriterien

Aus Endkundenanforderungen abgeleitete Kriterien	abgeleitete Anforderungen an Messkonzept	abgeleitete Anforderungen an Messtechnik
Abstände beider Rad-Außenflanken zu Bordstein gleich groß (tolerierte Abweichung $\leq 0,02$ m)	Abstand zwischen Rad-Außenflanken und Bordstein ist zu vermessen	Abstandsmessung quer zur Fahrtrichtung an der Seite des Fahrzeugs
Abstände beider Rad-Außenflanken zu Bordstein $\leq 0,07$ m	Abstände zu Parklücken begrenzenden Objekten sind zu vermessen	Abstandsmessung entlang der Fahrzeuginnenachse vor der Fahrzeugfront, hinter dem Heck
Abstand zu Objekten vor und hinter der Parklücke $\geq 0,23$ m, falls Rangierraum $\geq 0,5$ m	Lenkwinkel ist zu bestimmen	Orientierung eines der gelenkten Räder ist zu erfassen
Abstand zu einem der Objekte $\leq 0,4$ m, falls Rangierraum $\geq 1,1$ m (Aufbrücken)	Abstand zwischen Rad-Außenflanken und Bordstein ist zu vermessen	Abstandsmessung quer zur Fahrtrichtung realisieren
Lenkwinkel an gelenktem Rad $\leq 3^\circ$	Abstände zu Parklücken begrenzenden Objekten sind zu vermessen	Abstandsmessung entlang der Fahrzeuginnenachse vor der Fahrzeugfront, hinter dem Heck
Abstände beider Rad-Außenflanken zu Bordstein sind unter gleichen Bedingungen (=gleiche Startposition, gleiche Parklücke) am Ende des Einparkvorgangs konstant (tolerierte Abweichung $\leq 0,02$ m)	Parkzüge sind zu erfassen und zählen	Schubumkehrung des Fahrzeugs detektieren
Parkendposition ist unter gleichen Bedingungen konstant (tolerierte Abweichung $\leq 0,07$ m)	Zeitmessung zwischen Start und Ende des Parkvorgangs ist durchzuführen	Start- und Endzeit des Einparkvorgangs detektieren
Minimierung Parkzuganzahl Definition: Das erste Befahren der Parklücke ist bereits ein Parkzug, jede danach folgende Schubumkehrung des Fahrzeugs ist ein weiterer Parkzug.	Abbruch von eingeleiteten Parkvorgang ist zu detektieren	Abbruch von Parkvorgang detektieren
Minimierung der Einparkdauer	Anforderungen an Testverfahren erfüllt	Anforderungen an Testverfahren erfüllt
Abbruch eingeleiteten Einparkvorgangs ausschließen	Position und Orientierung des Fahrzeugs ist während des Einparkvorgangs in festen Zeitintervallen zu bestimmen	Koordinaten und Orientierungswinkel des Fahrzeugs sind zeitdiskret während des Einparkvorgangs bestimmen
Erfüllung gleicher Kriterien für unterschiedliche Einparksysteme am Markt		
Minimierung der Summe der Fehlerquadrate beim Vergleich zweier räumlicher Bewegungsverläufe mit gleichen Startbedingungen		

Tabelle 3: abgeleitete Anforderungen an das Messkonzept und die Messtechnik

abgeleitete Anforderungen an Messtechnik	quantifizierbare Kriterien	Messbereich	geforderte Genauigkeit	geforderte Auflösung
Abstandsmessung quer zur Fahrtrichtung an der Seite des Fahrzeugs	Abstandsdiﬀerenz zwischen Vorderrad Außenﬂanke zu Bordstein und Hinterrad-Außenﬂanke zu Bordstein	0...0,05 m	0,01 m	0,001 m
Abstandsmessung entlang der Fahrzeuglängsachse vor der Fahrzeugfront, hinter dem Heck	Differenz der Abstände zu beiden Parklücken begrenzenden Objekten	0...0,5 m	0,01 m	0,001 m
Abstandsmessung quer zur Fahrtrichtung realisieren	Abstand Rad-Außenﬂanken und Bordstein	0...0,5 m	0,01 m	0,001 m
Abstandsmessung entlang der Fahrzeuglängsachse vor der Fahrzeugfront, hinter dem Heck	Abstände zu den Parklücken begrenzenden Objekten	0...2 m	0,01 m	0,001 m
Orientierung eines der gelenkten Räder ist zu erfassen	Lenkwinkel, Einlenkwinkel am gelenkten Rad	0...50°	1°	0,1°
Schubumkehrung des Fahrzeugs detektieren	Bewegungsrichtungsänderung	0...10	1	1
Start und Ende des Einparkvorgangs detektieren	Dauer des Parkvorgangs ≤ 120/110/100/90/80/70/60/50/40/30/20/10 sec	0...120 sec	1 sec	0,1 sec
Abbruch von Parkvorgang detektieren	Abbruch eingeleiteten Einparkvorgangs	Ja/nein	-	-
Anforderungen an Testverfahren erfüllt	Anforderungen an Testverfahren erfüllt	Ja/nein	-	-
Koordinaten und Orientierungswinkel des Fahrzeugs sind zeitdiskret während des Einparkvorgangs bestimmen	Bewegungsverlauf des Fahrzeugs über der Zeit (2 Koordinaten auf der Fahrbahnebenen)	0...10 m	0,03 m	0,003m bei min. 1 Hz
	Orientierungswinkel von Fahrzeug bezüglich der z-Achse	0...50°	1°	0.1°

Tabelle 4: Ableitung quantifizierbarer Kriterien aus Anforderungen an die Messtechnik

3.1. Anforderungen an Testverfahren

Nach (Maurer u. a. 2015, S. 447 ff) werden Anforderungen an Testkonzepte in Effektivitäts- und Effizienzkriterien unterteilt. Diese sollen in diesem Abschnitt zunächst einzeln aufgeführt werden und in Bezug auf das Testszenario bei Einparkvorgängen erklärt werden. Im Kapitel zur Realisierung, unter Abschnitt 5.8 werden die spezifischen Maßnahmen zur Erfüllung der Anforderungen beschrieben.

3.1.1. Effektivitätskriterien

Repräsentativität

Der Grad an Realität ist im Testszenario zu minimieren, ohne dabei das Verhalten des Testobjektes gegenüber der Umwelt zu beeinflussen. Für Das Einparkszenario bedeutet dies, dass eine Fahrbahn mit Objekten, welche eine Parklücke aufspannen benötigt wird. Diese Objekte sollen durch die Sensorik des Fahrzeugs, bzw. des Einparksystems genauso detektierbar sein, wie reale Fahrzeuge. Darüber hinaus ist es aber nicht notwendig die Versuche auf einer während der Versuchsdurchführung befahrenen Straße durchzuführen.

Validität

Verfälschung des Verhaltens des Testobjektes, oder Verhalten und die Eigenschaften der Umwelt, gegenüber realen Verhalten ist auszuschließen. Das bedeutet für die Messtechnik, dass diese das Testszenario gegenüber Realität nicht verändern darf. Das bedeutet konkret für das Einpark-Szenario, dass Messtechnik auf dem Dach des Fahrzeugs und in Umgebung so zu applizieren ist, dass Einparkvorgänge nicht behindert werden.

Variierbarkeit

Die Testdurchführung muss Möglichkeiten bieten um unterschiedliche Testszenarien abzudecken, welche vorher zu definieren sind. Diese werden in 4.3 festgelegt.

Beobachtbarkeit

Parameter für die Ermittlung von Abständen und Orientierungen, Winkeln, Position des Versuchsfahrzeugs sollen beobachtbar sein. Während des Einparkens ist das Versuchsfahrzeug auf der Versuchsstrecke sichtbar. Gleiches gilt für das Lenkrad, die Räder des Fahrzeugs, den Bordstein und die parklückenaufspannenden Objekte. Weiterhin werden Referenzmarkierungen auf der Versuchsstrecke sowie an dem Fahrzeug angebracht.

3.1.2. Effizienzkriterien

Ökonomisch

Die Versuche sollen möglichst kosten- und zeiteffizient durchgeführt werden. Dazu ist die Versuchsstrecke von Hindernissen frei zu halten und die Aufzeichnung der Daten zeitlich möglichst effizient zu gestalten.

Reproduzierbarkeit

Die Versuchsanordnung soll bei weiteren Messreihen zu späteren Zeitpunkten und an anderen Orten nachgebaut werden können. Dadurch sollen Versuche unter gleichen Bedingungen realisiert werden können.

Sicherheit

Das akzeptierte Risiko aller Beteiligten soll nicht überstiegen werden. Bei den Einparkversuchen akzeptieren die beteiligten ein Restrisiko beim Betreten der Fahrbahn, welches ihnen bewusst ist und durch Koordination untereinander zu minimieren ist. Weitere Risiken sind auszuschließen.

4. Systementwurf

4.1. Definition des Ziels

Das Messsystem soll während des Einparkvorgangs in konstanten Zeitintervallen von maximal 1 Sekunde die Position und Orientierung des Fahrzeugs ermitteln. Außerdem sollen am Ende des Einparkvorgangs die Abstände der Räder zum Bordstein sowie die Abstände des Versuchsfahrzeugs zu den vor und hinter der Parklücke geparkten Fahrzeugen gemessen werden. Weiterhin ist der Einlenkwinkel der Räder am Ende des Einparkvorgangs zu ermitteln. Die Differenz zwischen Parklückengröße und Fahrzeuglänge ergibt den Rangierraum. Dieser soll bei den Messungen variiert werden. Ein Einparkvorgang gilt als abgeschlossen, sobald das Versuchsfahrzeug in der Parklücke steht und eigenständig die Bewegung des Fahrzeugs einstellt. Nach den Messungen sind die Ergebnisse statistisch auszuwerten und das System anhand geeigneter Darstellungen bezüglich der Endkundenanforderungen aus Abschnitt 2.5 zu bewerten.

4.2. Eingliederung und Funktion des Messsystems

In der folgenden Abbildung ist in einem Flussdiagramm dargestellt, welche Daten zwischen einem automatisierten Einparksystem und dem Versuchsfahrzeug ausgetauscht werden. Dabei empfängt das Einparksystem Signale der Sensorik, um für den Einparkvorgang relevante Informationen über Fahrzeugkomponenten und die Umgebung des Fahrzeugs zu erhalten. Anhand dieser Sensordaten werden Befehle an die Aktorik des Fahrzeugs gesendet, um einen Soll-Ist-Abgleich zu erreichen.

Das zu entwickelnde Messsystem soll, wie in Kapitel 2.5 hergeleitet, das Fahrzeug während des Einparkvorgangs stetig lokalisieren, seine Orientierung sowie Lenkwinkel an einem Rad der Vorderachse und am Lenkrad erfassen.

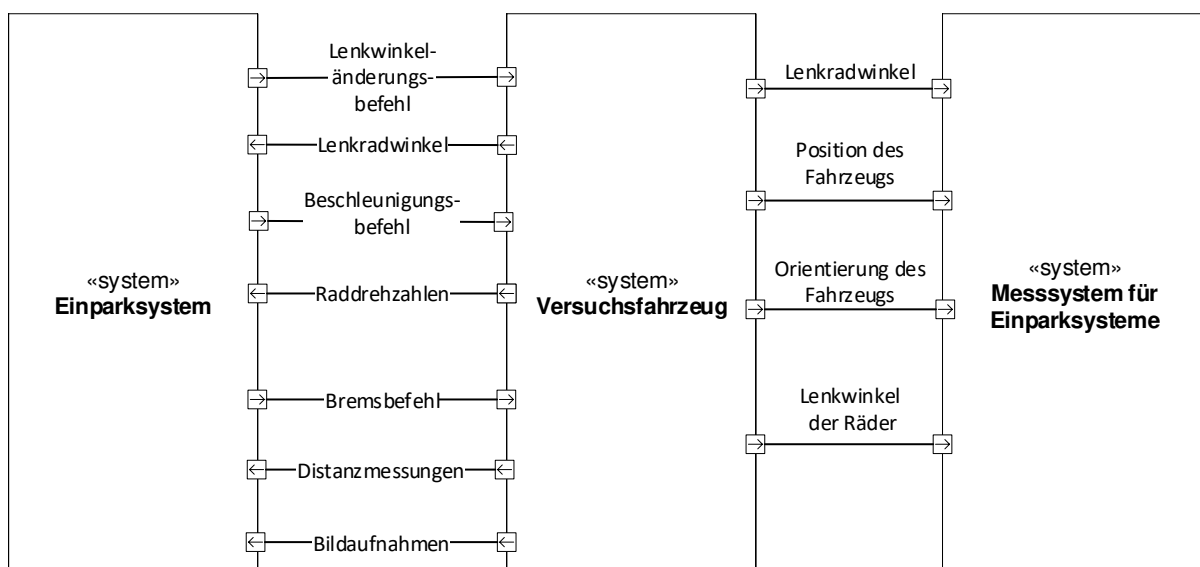


Abbildung 11: Kommunikation zw. Einparksystem, Versuchsfahrzeug und eingegliedertem Messsystem

4.3. Statistische Versuchsplanung

Bei dem zu entwickelnden Messkonzept sollen Häufigkeitsverteilungen (=Histogramme) von Abständen, welche hier die Qualitätsmerkmale darstellen betrachtet werden. Eine der bekanntesten Verteilungen ist die Normalverteilung und wird in seiner allgemeinen Form, wie in Abbildung 12 dargestellt.

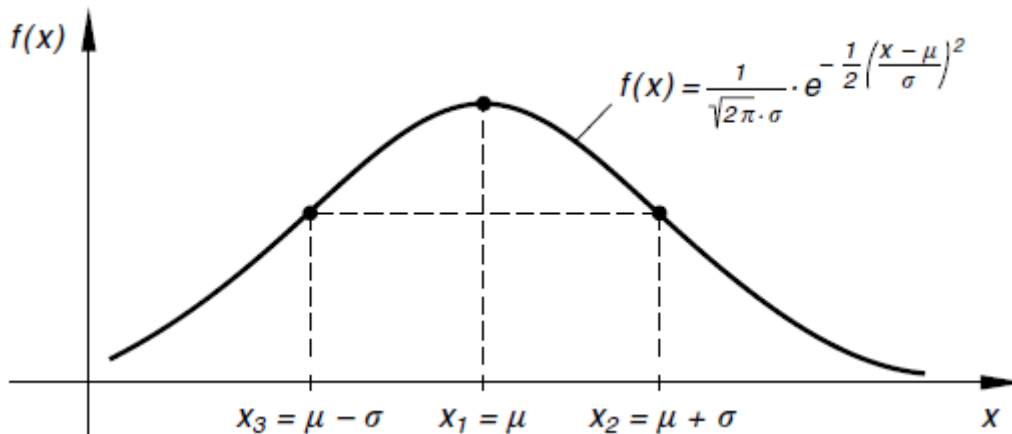


Abbildung 12: Allgemeine Form der Normalverteilung

Quelle: (Papula und Papula 2016, 371)

Bei der Normalverteilung liegen 68,3 % aller Werte im Bereich $\mu - \sigma \dots \mu + \sigma$ liegen, 95,4 % aller Werte im Bereich $\mu - 2\sigma \dots \mu + 2\sigma$ und 99,7% aller Werte im Bereich $\mu - 3\sigma \dots \mu + 3\sigma$ liegen (Papula und Papula 2016, 385).

Für die Messergebnisse aus dieser Arbeit soll untersucht werden, ob diese normalverteilt sind oder nicht und dabei μ und σ ermittelt werden. Dazu stellt sich die Frage, wie viele Versuche mit Messungen benötigt werden, um diese Untersuchung vornehmen zu können. Um dies herauszufinden wird folgender Zusammenhang herangezogen. 95% aller wahren Mittelwerte sollen zwischen

$$\mu - 2 \sqrt{\frac{\sigma^2}{n}} \quad (4)$$

und

$$\mu + 2 \sqrt{\frac{\sigma^2}{n}} \quad (5)$$

liegen, wobei n der Anzahl an Versuchen entspricht. Für die Variablen μ und σ können nun vermutete Werte eingesetzt werden. Für den Abstand zwischen Reifen und Bordstein und die Position in der Parklücke relativ zur Mitte werden bei unterschiedlichen Werten für die Grenzen und σ folgende Werte für n ermittelt.

	$\mu + 2 \sqrt{\frac{\sigma^2}{n}}$ [cm]	σ [cm]	n (gerundet)
Abstand zw. Reifen und Bordstein	1,5	2	7
	1	2	16
Differenz zu Abstand bei mittiger Parkendposition Endposition	2	3	9
	4	1	16

Tabelle 5: Berechnung benötigter Versuche zur Ermittlung von Normalverteilungen

Analog dazu können Berechnungen für die Ermittlung von Verteilungen (im Sinne reproduzierbarer Ergebnisse) bezüglich der Endabstände zum Bordstein oder den parklückenbegrenzenden Objekten erfolgen. Darauf wird hier verzichtet, da durch Einsetzen der selben Werte dieselben Ergebnisse liefert. Ein eindeutiges Ergebnis gibt es hier nicht und somit wird an dieser Stelle für die Anzahl benötigter Versuche **n=15** festgelegt. Das bedeutet um aussagen zu können, ob die Häufigkeitsverteilung der gemessenen Abstände normalverteilt ist, sind für die einzelnen Stufen jeweils 15 Versuche durchzuführen. Des Weiteren sollen der Einfluss des Rangierraums auf die Messgrößen untersucht werden. Dabei wird auf eine Untersuchung von Häufigkeitsverteilungen verzichtet, sodass weitere Stufen mit weniger als 15 Versuchen festgelegt werden. Es folgt eine tabellarische Auflistung der geplanten Stufen mit den Versuchswiederholungen. Die Bezeichnung der Stufe gibt den entsprechenden Rangierraum wieder.

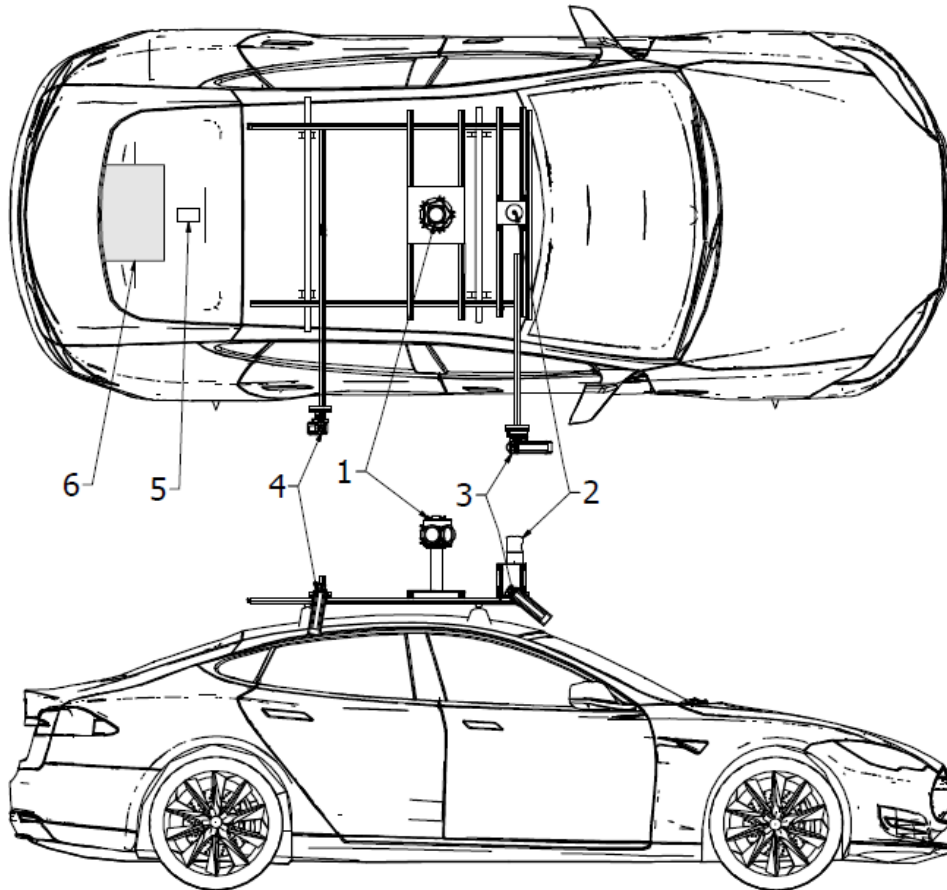
Stufen-Nr.	Stufe (Rangierraum in mm)	n
1	1000	15
2	900	15
3	800	15
4	1100	15
5	1200	15
6	1300	15
7	1400	15
8	1500	2
9	1600	2
10	1700	2
11	1800	2
12	1900	2
13	2000	2
Gesamt	-	117

Tabelle 6: Anzahl der benötigten Versuche zur Erreichung einer Normalverteilung

Da zum Zeitpunkt der Planung noch nicht genau abgeschätzt werden kann wie viel Zeit die Versuche in Anspruch nehmen, werden zunächst die Stufen mit dem kleinstmöglichen Rangierraum betrachtet. Der kleinste von modernen Einparkassistenten bewältigte Rangierraum beträgt etwa 800mm (Renz 2013; „Vollautomatisches, autonomes Einparken“ o. J.). Falls die Stufe Nr. 3 bewältigt wird, ist vorgesehen den Rangierraum weiter um 100mm zu verringern. Bei der ersten Stufe, die nicht bewältigt wird, soll der Rangierraum um 50 mm vergrößert werden. Eine Stufe gilt als nicht bewältigt, sobald mindestens 5 Versuche erfolglos bleiben, d.h. kein Einparkvorgang abgeschlossen wird.

4.4. Messmethodik

In diesem Kapitel werden die Methoden zum Messen der gesuchten Größen beschrieben. Für das Verständnis der Ausführungen dazu folgt eine grafische Darstellung des Versuchsfahrzeugs mit der applizierten Messtechnik.



1. Ladybug 5 Kamera
2. Velodyne HDL-32E LiDAR Sensor
3. Basler acA1920-40uc (Seitenkamera, vorne)
4. Basler acA1920-40uc (Seitenkamera, hinten)
5. Apogee-D Inertial Navigation System
6. Rechner

Abbildung 13: Anordnung der Sensorik am Versuchsfahrzeug (2. Entwurf)

4.4.1. Endposition

Der Abstand zwischen Bordstein und dem vorderen sowie hinteren Rad wird über Aufnahmen der Kameras an der Seite des Sensorträgers, bzw. Versuchsfahrzeug vermessen. Dabei werden die Kameras kalibriert, um mit den internen Parametern Aufnahmen zu entzerren sowie die Lage des optischen Mittelpunktes im Bild sowie der optischen Achse im Raum zu ermitteln. Dazu wird die Position der Kameras auf dem Fahrzeug vermessen. Damit können Koordinaten von Bildpunkten in Raumkoordinaten umgerechnet werden. Aus der relativen Position von Reifen und Bordstein zum optischen Zentrum können bei bekannter Lage der optischen Achse im Raum mit den Öffnungswinkeln und der Auflösung des Bildes die Abstände im Raum berechnet werden. Für die Bestimmung der Abstände zu den Fahrzeugen vor und hinter der Parklücke ist zum Zeitpunkt des Systementwurfs die Auswertung von Distanzmessungs-Daten des LIDAR-Sensors vorgesehen.

4.4.2. Positionsbestimmung mit zwei Raum-Koordinaten

Die Position des Fahrzeugs während des Einparkens wird zum einen über vogelperspektivische Bildaufnahmen und Lokalisierung von am Fahrzeug angebrachten Markierungen bewerkstelligt. Dazu muss Lage und Ausrichtung der Kamera sowie die Position der Markierung auf dem Fahrzeug, bestimmt werden.

Außerdem ist vorgesehen, die Position aus den Aufnahmen einer sphärischen Kamera zu bestimmen. Dazu werden auf der Versuchsstrecke Ortsmarkierungen verteilt, deren Abstände zueinander vermessen werden. Aus der Lage der Markierungen in den Panorama-Bildaufnahmen kann die Position sowie Orientierung des Fahrzeugs bestimmt werden.

Mit beiden Messemethoden soll der Bewegungsverlauf von Start bis Ende des Einparkvorgangs bei einer Abtastfrequenz von mindestens 1 Hz ermittelt werden. Damit kann bei bekannter Position der geparkten Fahrzeuge zum Zielkoordinatensystem auch der Endabstand zu diesen Fahrzeugen bestimmt werden. Primär sollen die beiden Messemethoden zur Aufzeichnung von Trajektorien dienen. Durch Auswertung der Trajektorien sollen Einparkdauer, Parkzuganzahl und Zusammenhänge zwischen Start- und Endposition untersucht werden.

4.5. Architektur der Messtechnik

Am Urban Mobility Lab stehen Kameras sowie ein LIDAR-Sensor und ein Versuchsfahrzeug, welches mit einem Rechner ausgestattet ist zur Verfügung. Um die Messmethoden aus Abschnitt 4.4 zu realisieren werden die Komponenten wie in der folgenden Abbildung dargestellt angeordnet. Die Spezifikationen zu den Sensoren folgen in Abschnitt 5.3..

Die Seitenkameras werden so bezeichnet, da diese die Funktion haben an der Seite des Versuchsfahrzeugs Bilder vom Rad und Bordstein aufzunehmen. Diese Aufnahmen sind für laterale Positionsmessungen vorgesehen. Zentriert auf dem Fahrzeug sitzt eine sphärische Kamera und ein LIDAR-Sensor, deren Funktion es ist das Umfeld zu erfassen. Mit Daten dieser Sensoren soll das Fahrzeug während des Einparkvorgangs bei Aufnahmezeiten von mindestens 1 Hz lokalisiert werden und der Orientierungswinkel ermittelt werden. Außerdem sollen mit einer Kamera vogelperspektivische Bilder (ebenfalls bei mindestens 1Hz) vom einparkenden Versuchsfahrzeug aufgezeichnet werden. Diese über dem Fahrzeug positionierte Kamera wird im Folgenden als Vogelkamera bezeichnet. Alle Sensoren sind von dem Rechner des Versuchsfahrzeugs aus zu steuern und die Aufnahmen sind von dem Rechner aus zu überwachen.

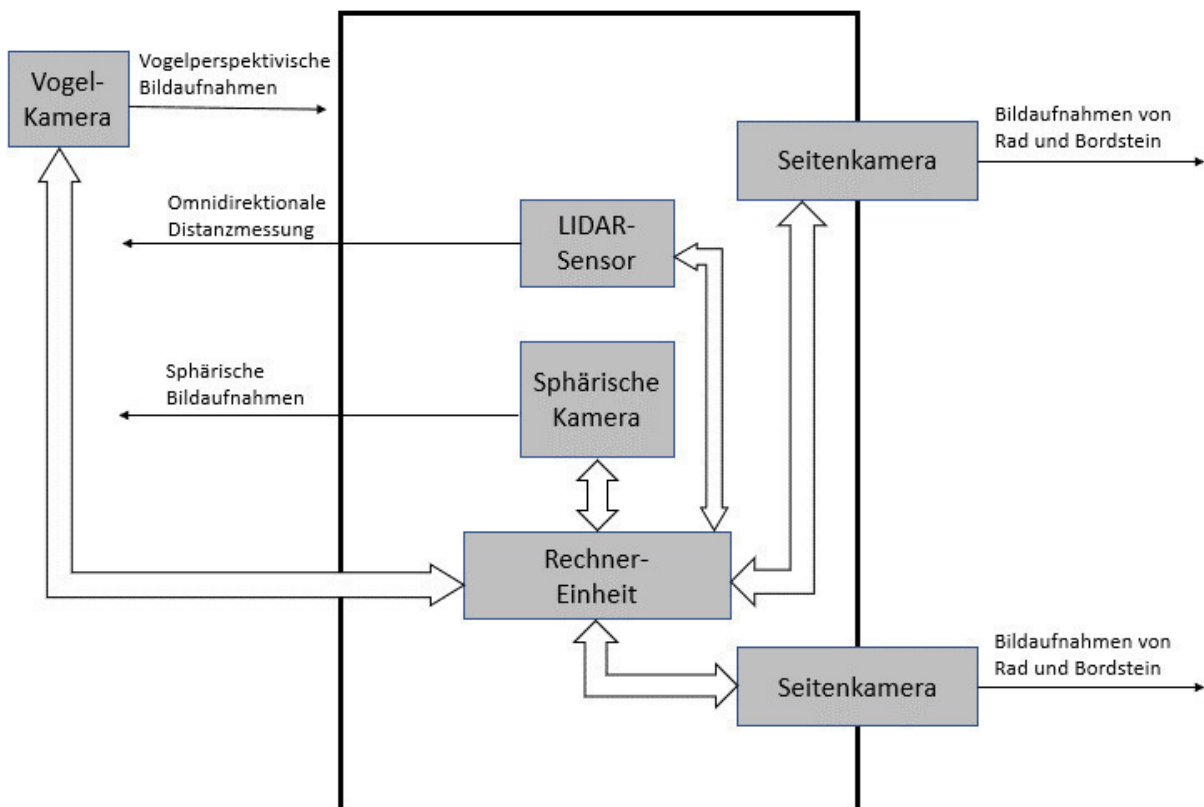


Tabelle 7: Architektur der Messtechnik

5. Realisierung

5.1. Versuchsfahrzeug

Bei dem Versuchsfahrzeug handelt es sich um einen Tesla Model S 75D mit folgenden Daten:

Modell	Model S 75D
Länge [mm]	4979
Breite, ohne Spiegel [mm]	1964
Breite, mit Spiegeln [mm]	2189
Höhe [mm]	1445
Radstand [mm]	2960
Wendekreis [m]	11,8
Max. Dachlast [kg]	75
Überhang, vorne [mm]	937
Überhang, hinten [mm]	1081
Spur, vorne [mm]	1662
Spur, hinten [mm]	1700

Tabelle 8: Herstellerangaben zum Versuchsfahrzeug aus Handbuch des Fahrzeugs und Fahrzeugschein

Darüber hinaus werden die Abstände der Radaußenflanken der verwendeten Konfiguration sowie die Fahrzeugbreite an den Kotflügel-Außenseiten gemessen. Dabei beziehen sich die Messungen an den Reifen auf den am weitesten nach außen ragenden Punkt über der Reifenaufstandsfläche. Ebenso werden an den Kotflügeln die am weitesten nach außen ragenden Punkte vermessen.

Abstand Reifen-Außenflanken, hinten [mm]	1945
Abstand Reifen-Außenflanken, vorne [mm]	1906
Fahrzeugbreite an Kotflügel-Außenseiten, vorne [mm]	1939
Fahrzeugbreite an Kotflügel-Außenseiten, hinten [mm]	1956

Tabelle 9: Reifen-Außenflanken-Abstände des Versuchsfahrzeugs

5.2. Applikation der Messtechnik an Versuchsfahrzeug

Um die Sensorik am Versuchsfahrzeug zu applizieren wird auf Grundlage eines Vorgängermodells ein neuer Träger konstruiert. Dieser soll auf dem Dachgepäckträger appliziert werden können und die Sensorik tragen. Zum Zeitpunkt der Realisierung zählen zu den Sensoren die Ladybug5-Kamera, die 4 Basler Kameras mit Gehäusen, der LIDAR-Sensor sowie seine die GPS-Antenne. Dabei werden Mindestanforderungen an die Konstruktion formuliert:

1. Die Konstruktion soll mit zwei Person an am Fahrzeug applizierbar und entfernbar sein
2. Die Konstruktion soll nicht über die maximale Fahrzeugbreite oder -Länge hinausragen
3. Sensoren sollen spielfrei (Prüfung per Hand) über dem Fahrzeugdach getragen werden
4. Sensoren sollen die Sicht der anderen Sensoren möglichst nicht beeinträchtigen
5. Sensoren sollen bei möglichst geringem Aufwand verschiebbar sein
6. Sensoren sollen bei möglichst geringem Aufwand an- oder abgebaut werden können.
7. Sensoren sollen gekippt angebaut werden können.
8. Die Konstruktion soll weniger als 75 kg wiegen
9. Die Kosten der Konstruktion sollen 999 € nicht überschreiten
10. Das Öffnen und Schließen der Heckklappe soll nicht eingeschränkt werden
11. Die Konstruktion oder Teile von dieser sollen sich nach Stadtfahrten bei bis zu 60 km/h bis 1 Std nicht lösen

Auf dieser Grundlage wird eine Konstruktion aus Alu-Profilen mit einem 30 mm x 30 mm Querschnitt zuerst in der CAD-Software Autodesk Inventor 2018 geplant. Dazu werden vorher die Abstände und Längen der Dachgepäckträger-Schienen vermessen. Außerdem wird der Abstand der hinteren Dachgepäckträger-Schiene bis zur Heckklappe gemessen und es liegen Daten zu den Abmessungen des Versuchsfahrzeugs vor. Weiterhin liegt ein Nachbau des Versuchsfahrzeugs als CAD-Modell vor, sodass die CAD-Konstruktion auf diesem Modell aufgebaut werden kann. Mit diesem Modell werden Abbildungen, wie z. B. Abbildung 13 erstellt.

5.3. Sensorik

5.3.1. Sphärische-Kamera FLIR Ladybug 5

Die Kamera hat folgende Spezifikationen (Auszug aus Datenblatt):

Auflösung [Pixel]	2048 x 2448 pro Sensor
Sensor	Sony ICX655
Sensorgröße	2/3"
Sensortechnologie	CMOS, global shutter
Pixelgröße [μm^2]	3,45 x 3,45
Anzahl an Sensoren	6
Brennweite [mm]	4,4
Max. Bildrate [fps]	30
Ausgabeformate	8, 12 oder 16 bit Raw oder JPEG komprimiert

Tabelle 10: Spezifikationen zur Kamera FLIR Ladybug 5

Quelle: („Ladybug5+ USB3“ o. J.)

5.3.2. Seitenkameras und Vogelkamera

Für die Messungen werden zwei Basler acA1920-40gc als Seitenkameras und eine Basler acA1920-40gm als Vogelkamera eingesetzt. Diese werden in „Salamander“-Gehäuse der Firma Autovimation in der Ausführung „M“ eingesetzt. Zwei der Kameras werden mit dem Objektiv Kowa LM12HC und eine mit dem Objektiv Kowa LM6HC kombiniert. Es folgt zunächst ein Auszug aus dem Datenblatt der Kameras:

Modell	Basler acA 1920-40um/uc
Auflösung [Pixel]	1920 x 1200
Sensor	Sony IMX249
Sensorgröße [mm ²]	11,3 x 7,1
Sensortechnologie	CMOS, global shutter
Pixelgröße [μm^2]	5.86 x 5.86
Max. Bildrate [fps]	42
Bittiefe eines Pixels	10 bis 12
Ausgabeformate (Video)	AVI bis 2GB Datenvolumen

Tabelle 11: Spezifikationen zur Kamera Basler acA 1920-40um/uc

Quelle: (AG o. J.)

Die Spezifikationen der eingesetzten Objektive sind nach den Datenblättern folgende:

Modell	KOWA LM6HC
Brennweite [mm]	6

Tabelle 12: Spezifikationen zum Objektiv Kowa LM6HC

Quelle: („LM6HC | Produkte | Kowa Optimed Deutschland GmbH“ o. J.)

Modell	KOWA LM12HC
Brennweite [mm]	12,5

Tabelle 13: Spezifikationen zum Objektiv Kowa LM12HC

Quelle: („LM12HC | Produkte | Kowa Optimed Deutschland GmbH“ o. J.)

Damit ergeben sich die Öffnungswinkel α_v (vertikal) und α_h (horizontal) der eingesetzten Kameras nach Formeln aus Abschnitt 2.4.4 (Linsen) wie folgt:

	α_v	α_h
Basler acA 1920-40um mit Kowa LM12HC	31,7°	48,6°
Basler acA 1920-40uc mit Kowa LM12HC	31,7°	48,6°
Basler acA 1920-40um mit Kowa LM6HC	61,222°	86,558°

Tabelle 14: Öffnungswinkel der verwendeten Kameras

5.3.3. Bosch Laser-Entfernungsmesser

Für Messungen in der Versuchsumgebung wird ein Laser-Entfernungsmesser eingesetzt. Es folgt ein Auszug aus dem Datenblatt:

Modell	Bosch PLR 25
Wellenlänge des Lasers	635 nm
Messbereich	0,05 bis 25 m
Messgenauigkeit	+/- 2,0 mm
Messzeit	0,5 bis 4s
Laserklasse	2

Tabelle 15: Spezifikationen des Bosch-Laserentfernungsmessers

Quelle: („PLR 25 | Laser-Entfernungsmesser | Digitale Messwerkzeuge | Heimwerker“ o. J.)

5.3.4. LIDAR Sensor Velodyne HDL-32E

Modell	Velodyne HDL-32E
Messbereich	Bis 100m
Genauigkeit	+/- 2cm
Vertikaler Sichtbereich	+10,67° bis -30,67°
Vertikale Auflösung	1,33°
Horizontaler Sichtbereich	360°
Horizontale Auflösung	0,1° bis 0,4°
Rotationsfrequenz	5 – 20 Hz
Wellenlänge des Lasers	903 nm

Tabelle 16: Spezifikationen des LIDAR-Sensors

Quelle: („Velodyne LiDAR“ o. J.)

5.3.5. SBG Systems Apogee-D Inertial Navigation System

Modell	Apogee-D Dual GNSS/INS
Messgrößen	Beschleunigung bezüglich 3 Achsen Orientierungswinkel bezüglich 3 Achsen 3 Koordinaten

Tabelle 17: Spezifikationen des Inertial Navigation Systems

Quelle: („Apogee-D: High Accuracy Dual Antenna GNSS/INS“ o. J.)

5.4. Schnittstellen und Kommunikation

5.4.1. Rechneinheit des Versuchsfahrzeuges

Für die Messungen steht ein Rechner zur Verfügung, welcher im Kofferraum des Versuchsfahrzeuges eingebaut ist. Auf diesem ist als Betriebssystem Linux installiert und alle Sensoren werden mit ihm über **USB 3.0** (Seitenkameras, sphärische Kamera) oder **Ethernet** (LIDAR-Sensor) verbunden. Außerdem wird ein Laptop über Ethernet verbunden, sodass der Rechner per **Remote-Verbindung** gesteuert werden kann. Bei der zweiten Messreihe wird auf einem weiteren Laptop ein Wifi-Hotspot eröffnet und an diesen wird Vogelkamera angeschlossen. Der Laptop im Versuchsfahrzeug wird in den Hotspot ausgewählt und an diesem kann somit auch diese Kamera angesteuert werden.

5.4.2. Seitenkameras

Für die Basler-Kameras werden entsprechende Treiber des Herstellers installiert und sie werden über die mitgelieferte „Plyon“-Software gesteuert. Bei den Aufnahmen handelt es sich um Bilder im bmp-Format mit einer Bittiefe von 24 bit pro Bildpunkt und einer Auflösung von 1920x1200 Bildpunkten.

5.4.3. Sphärische Kamera

Ebenso wie die Seitenkameras, wird die Ladybug-Kamera über einen mitgelieferten Treiber für Linux gesteuert. Die aufgenommenen Stream Files werden mit der mitgelieferten „Ladybug Recap“ Software als Bildsequenzen im JPEG-Format extrahiert. Die Ladybug-Aufnahmen haben eine Auflösung von 8192x4096 Bildpunkten und ebenso wie die Basler-Kamera-Aufnahmen eine Bittiefe von 24 bit.

5.4.4. LIDAR-Sensor

Der LIDAR-Sensor wird über einen ROS-Server, welcher auf dem Rechner installiert ist gesteuert. Die aufgezeichneten Daten werden als Rohdaten oder als Point-Cloud-Daten exportiert. Mit der entsprechenden ROS-Toolbox für Matlab können diese Datensätze welche Zeitstempel mit Winkeln und Distanzen enthalten eingelesen werden.

5.5. Teststrecke, Anordnung

Am Ende dieses Abschnitts folgen Abbildungen (Seiten 31 und 32) der Versuchsanordnungen, wie sie auf der Versuchsstrecke am Department Fahrzeugtechnik und Flugzeugbau der Haw-Hamburg umgesetzt werden. Nachdem die erste Messreihe stattfindet, wird bei der zweiten die Versuchsanordnung im Sinne einer effektiven Versuchsdurchführung und Auswertung angepasst. Bevor die Einparkversuche durchgeführt werden, wird die geplante Anordnung umgesetzt. Dieser Abschnitt beschreibt die praktische Umsetzung der Anordnung.

Als Orts-Markierungen sind Hütchen mit Schildaufsätzen vorgesehen. Um die Positionierung der Orts-Markierungen möglichst genau nach Plan umzusetzen wird zuvor die Versuchsumgebung vermessen. Auf Grundlage eines Lageplans werden vor den Messungen die Positionen für die Orts-Markierungen auf der Versuchsstrecke mit weißem Klebeband und Marker-Stift markiert. Dabei entsteht auf dem Boden eine Kreuzmarkierung. Sobald die jeweilige Orts-Markierung dann positioniert ist, kann das Schild auf dem Hütchen zunächst über einer der Linien positioniert werden und anschließend durch Verdrehen über die andere Linie. Durch erneutes Verdrehen wird geprüft, ob die Position noch mit der ersten Linie übereinstimmt. Dieses wird im Zweifel wiederholt, bis die Mitte der Ortsmarkierung über der Mitte des Kreuzes auf dem Boden liegt.

Als Fahrzeug vor der Parklücke parkendes Fahrzeug wird ein BMW E36 Compact aus dem Fahrzeuglabor der Hochschule ausgewählt, da dieser durch seine geringe Höhe von 1393 mm beste Sicht auf die Ortsmarkierungen Nr. 1 und Nr. 2 vom Startpunkt der Einleitung des Einparkvorgangs aus gewährleistet. Zum anderen wird das Risiko einer Kollision der Seiten-Kameras des Versuchsfahrzeugs mit dem parkenden Fahrzeug minimiert. Für den BMW E36 Compact wird die Position für den Frontabschluss am Bordstein markiert. Während der Versuche wird dieses Fahrzeug nicht um positioniert. Vor Versuchsbeginn wird das Fahrzeug entsprechend des Lageplans an der Markierung positioniert, während eine Person an der Seite gegenüber des Bordsteins die Position prüft und dem Fahrzeugführer Zeichen gibt. Dieser passt die Position des Fahrzeugs an, bis die Position des Nummernschilds mit der Markierung am Bordstein übereinstimmt. Dabei wird eine Abweichung von ca. ± 5 cm zugelassen. Die genaue Position ist hier für das Blickfeld der von oben filmenden Kamera wichtig. Die Länge der Parklücke wird später durch Verschiebung des hinter der Parklücke stehenden Fahrzeugs eingestellt.

Eine 6,1 m-Markierung, welche ausgehend von der Markierung für das vor der Parklücke geparkten Fahrzeugs aus vermessen wird, dient zur späteren Orientierung bei der Einstellung der Länge der Parklücke. Vor den Messungen wird das hinten parkende Fahrzeug an dieser Markierung so positioniert, dass etwa 6,1 m Abstand zwischen dem Heckabschluss dieses Fahrzeugs und dem Nummernschild des BMW 3C eingestellt werden. Anschließend wird der Abstand mit dem Laser-Distanzmesser kontrolliert und das gemessene Maß aufgenommen. Bei späteren Variationen des Abstandes wird analog zu diesem Ablauf vorgegangen. Bei Messungen der Abstände zwischen den Fahrzeugen wird der Laser-Distanzmesser stets an dieselbe Position am Heckabschluss des vorne geparkten Fahrzeugs oder am Versuchsfahrzeug angelegt und stets derselbe Punkt der dahinter geparkten Fahrzeug angepeilt.

Die von oben filmende Vogelkamera wird bei der ersten Messreihe auf einer Stativkonstruktion in einem Raum, welcher von der Parklücke aus gesehen, bezüglich der Fahrbahn auf der gegenüberliegenden Seite liegt positioniert.

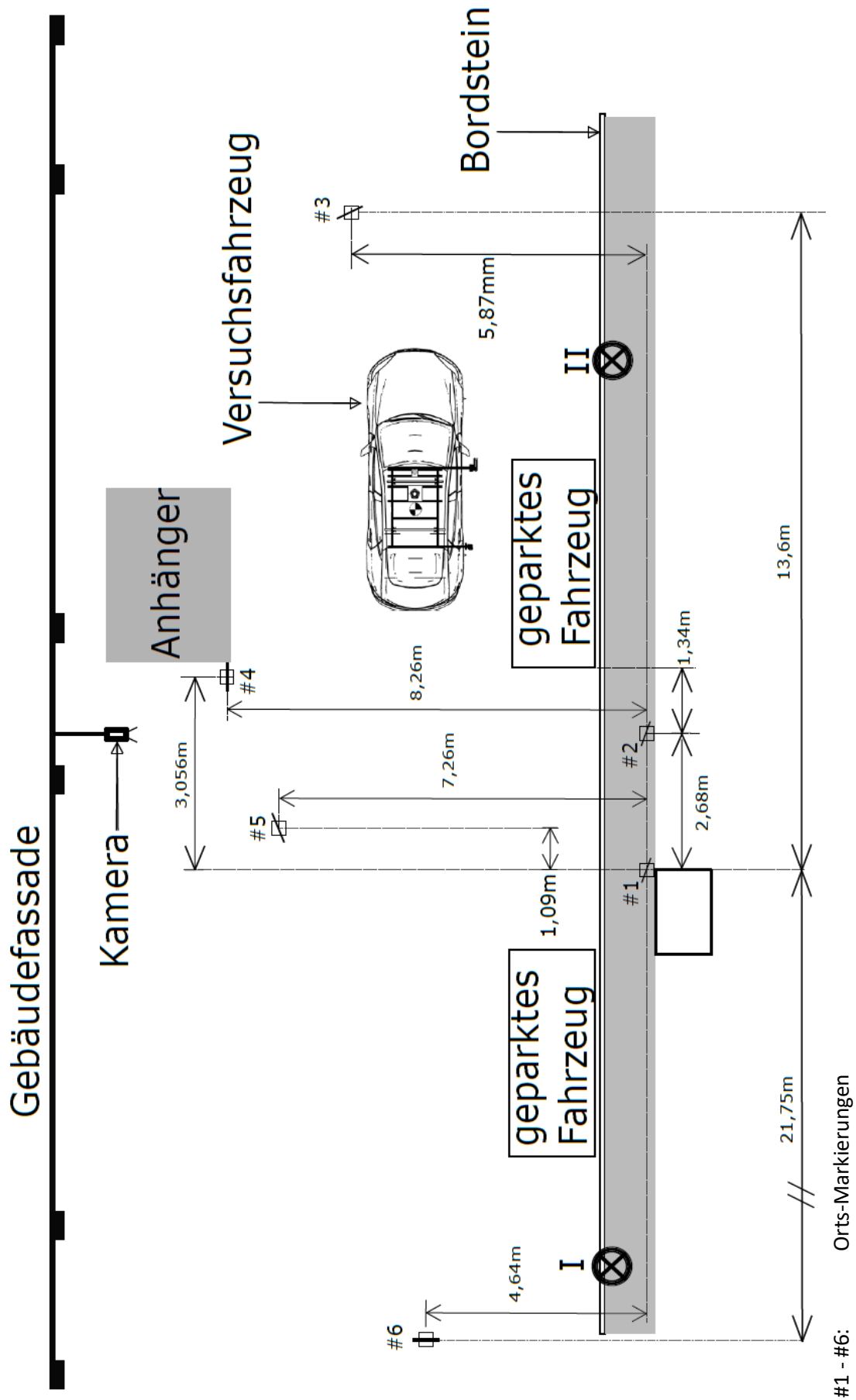


Abbildung 14: Lageplan der Versuchsanordnung bei der 1. Messreihe

#1 - #6: Orts-Markierungen

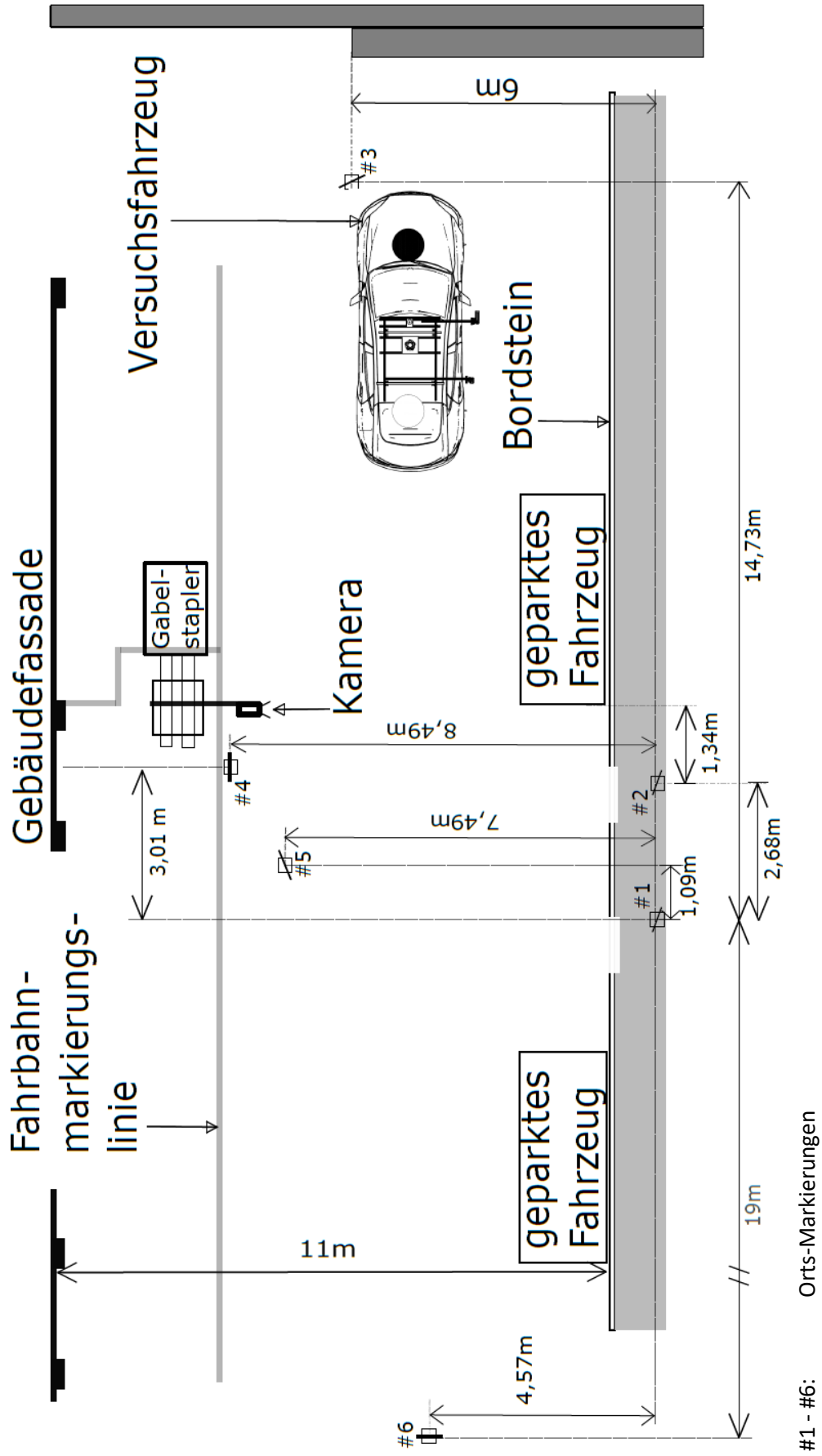


Abbildung 15: Lageplan der Versuchsanordnung bei der 2. Messreihe

5.5.1. Anpassungen nach der ersten Messreihe

Bei der zweiten Messreihe wird anhand der Endpositionen der vorherigen Versuche sowie des Radstandes des Fahrzeugs die Position für die Bordsteinmarkierung markiert, sodass die Markierung bei unterschiedlichen Parklückengrößen bezogen auf das Fahrzeugkoordinatensystem rechts vom Rad in der Aufnahme sichtbar sein soll. Für die Markierung des Bordsteins wird weiß glänzende Kunststoff-Folie mit klebender Unterseite verwendet. Die Folien sind jeweils 1,1 m lang und die Breite von ca. 20 cm ist so gewählt, dass die Folie vom Bordsteinfuß ausgehend die Seite des Bordsteins sowie die gesamte Oberkante abdeckt. Zusätzlich wird die Folie mit weißem Klebeband am Gehweg fixiert.

Nachdem aufgrund von Sanierungsarbeiten am Gebäude die gesamte Versuchsanordnung bei der zweiten Messreihe verschoben werden muss, wird die von oben filmende Kamera mit Hilfe eines Gabelstaplers am Rand der Fahrbahn auf die nötige Höhe gebracht. Näheres zur genauen Position der Kamera folgt im nächsten Abschnitt.

Bestimmung der Position und Orientierung der Kamera

Das Grundkonzept der Messung über vogelperspektivische Aufnahmen ändert sich nach der ersten Messreihe nicht. Deshalb wird das Konzept nur am Beispiel der zweiten Messreihe erklärt.

Die Kamera auf der Stativkonstruktion wird mittig auf der Länge der Gabel positioniert und die Innenseiten der Befestigungspunkt der Räder der Stativkonstruktion werden an die Gabel anliegend positioniert. Damit verlaufen die Profile der Vorderseite der Stativkonstruktion parallel, bzw. die seitlichen Profile senkrecht zur Gabel. Die Konstruktion wird mit Schraubzwingen auf der Gabel des Gabelstaplers befestigt und die Box mit dem Laptop auf dem Regal der Stativkonstruktion mit Spanngurten fixiert. Die folgende Abbildung zeigt diese Anordnung.

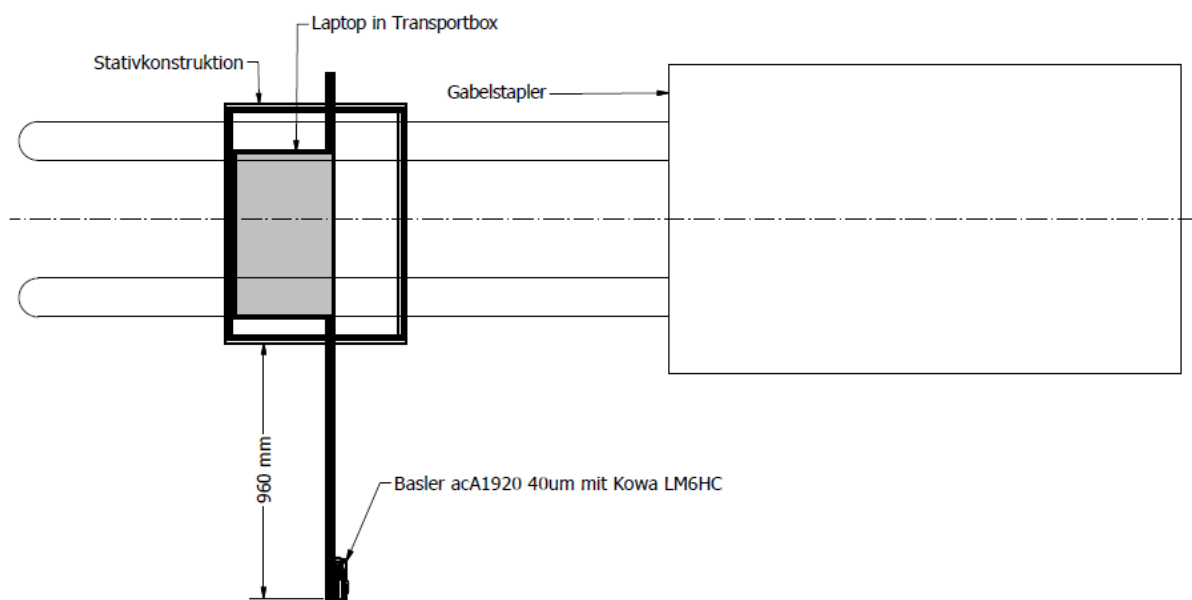


Abbildung 16: Position und Ausrichtung der Stativkonstruktion mit Kamera auf Gabelstapler

Um die Position des Markers im Bild in Raumkoordinaten zu transformieren, ist die Position sowie die Ausrichtung der Kamera, bzw. des Kamerasensors zu ermitteln. Nachdem die Kamera entsprechend Abbildung 15 positioniert und auf die die maximal mögliche Höhe

gefahren ist, wird der Gabelstapler versetzt, bis die Sicht auf das Versuchsfahrzeug von Anfang bis Ende des Einparkvorgangs zu gewährleistet werden kann. Die **Höhe der Gabel von 4634 mm** wird mit dem Laser-Distanzmesser aufgenommen. Dazu wird der Distanzmesser an die Schiene der Gabel und an den Untergrund angelegt. Damit wird näherungsweise senkrecht zur Straße der Abstand von Straße bis zur Gabel gemessen. Vor Anheben der Gabel wird durch den Fahrer sichergestellt, dass diese nicht geneigt ist. Somit entspricht der gemessene Abstand auch näherungsweise dem Abstand zwischen Straße und der Unterseite der Gabel unter dem Objektiv. Zu diesem Abstand wird die **Tiefe der Gabel von 50 mm** sowie die **Höhe des Stativs von 1780 mm** hinzuaddiert. Mit Subtraktion der mit einem Maßband aufgenommenen Höhendifferenz zwischen der Unterseite des Profils, an dem die Kamera auf dem Stativ montiert ist und der Position des Kamerasensors von **150 mm (+- 2 cm)** ergibt dies die Höhe des Kamerasensors relativ zur Straße. Diese beträgt folglich **6314 mm**. Die Anordnung ist in Abbildung 19 zu sehen.

Nach den Messvorgängen wird die Gabel abgesenkt und die Position des Stativs wird senkrecht zur Seitenstreifenbegrenzung gemessen, sodass sich folgende Anordnung ergibt.

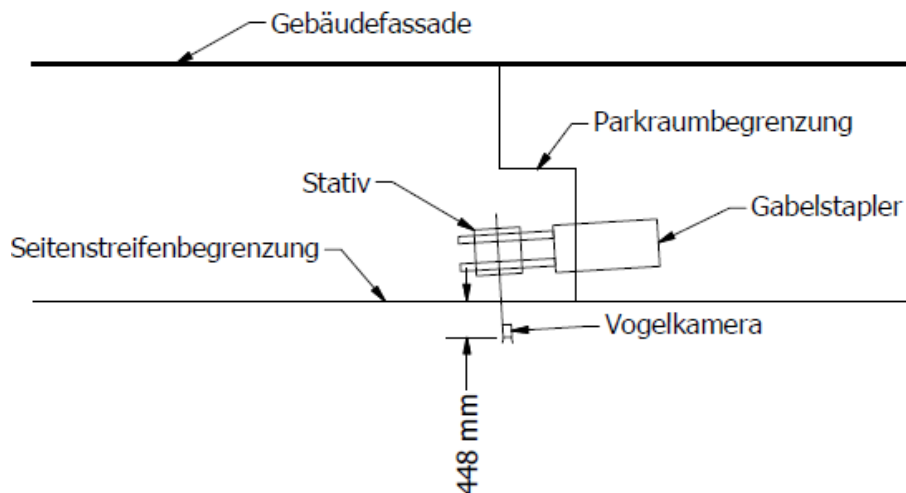


Abbildung 17: Bestimmung der Kameraposition in der x-y-Ebene

Da auch die herausstehende Länge des Profils, welches die Kamera trägt bekannt ist (960mm – siehe 5.5.1), kann der zur Seitenstreifenbegrenzung senkrechte Abstand der Kamera zu dieser bestimmt werden. Da die Seitenstreifenbreite sowie der Abstand zwischen Fassade und Bordstein sowie dem Seitenstreifen auf der gegenüberliegenden Seite der Fahrbahn bekannt sind (siehe Abbildung 15), kann der theoretische Abstand zwischen der Kamera und die Seitenstreifenbegrenzung des gegenüberliegenden Seitenstreifens ermittelt werden. Dieses ist in der folgenden grafischen Darstellung zu sehen. Die **3,27° für den Verdrehwinkel β** werden durch Messungen der Abstände der Eckpunkte der Stativkonstruktion zum Seitenstreifen ermittelt.

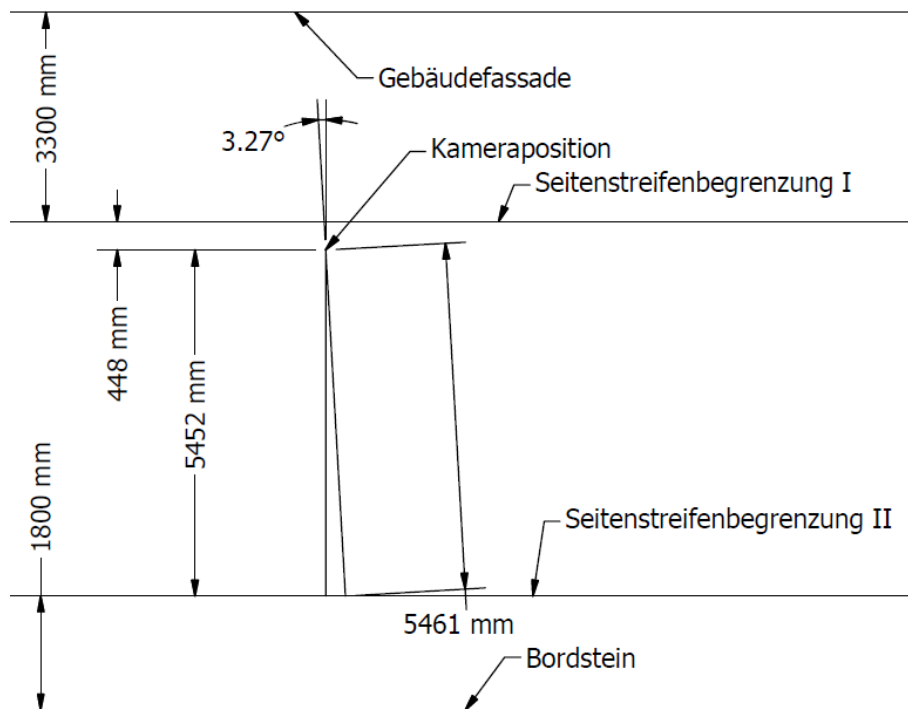


Abbildung 18: Lage und Orientierung der Weitwinkelkamera

Die Seitenstreifenbegrenzung II ist in den Aufnahmen der Kamera sichtbar und seine Relativposition zum optischen Mittelpunkt kann bestimmt werden. Über den vertikalen Öffnungswinkel der Kamera mit der vertikalen Auflösung kann der relative Winkel zum optischen Zentrum berechnet werden. Mit dem theoretischen Winkel, welcher sich aus dem gemessenen Abstand der Kamera zum Seitenstreifen und der Höhe ergibt dies den Neigungswinkel der optischen Achse der Kamera. Der theoretische Winkel bei dem Abstand der Kamera zur Seitenstreifenbegrenzung II (siehe Abbildung 18) und der Höhe der Kamera relativ zur Straße ergibt sich mit dem Tangens zu **40,85°**. Bei einer Lage der Seitenstreifenbegrenzung II bei $x=960$, $y=429$ im entzerrten Bild, befindet sich dieser **168 Bildpunkte** über dem optischen Zentrum (Messung 5, Frame 3). Dieses entspricht bei 1254 Pixeln über einen Bildwinkel von 62,22° einem Relativwinkel von 8,34°. Subtrahiert vom theoretischen Winkel zwischen z-Achse und der Verbindungslinie zwischen Kamera und Seitenstreifenbegrenzung II ergibt dies den Neigungswinkel der Kamera.

Der so ermittelte Neigungswinkel α beträgt 32,51°. An dem Profil, welches die Kamera trägt wird vor den Messungen mit Hilfe eines Geodreiecks ein Winkel von ca. 32° gemessen. Die Abweichung kann auf eine Krümmung der Straße sowie Ungenauigkeit beim Messen mit dem Geodreieck zurückgeführt werden. Der Winkel α von 32,51° wird für die weiteren Berechnungen als Neigungswinkel der Kamera eingesetzt.

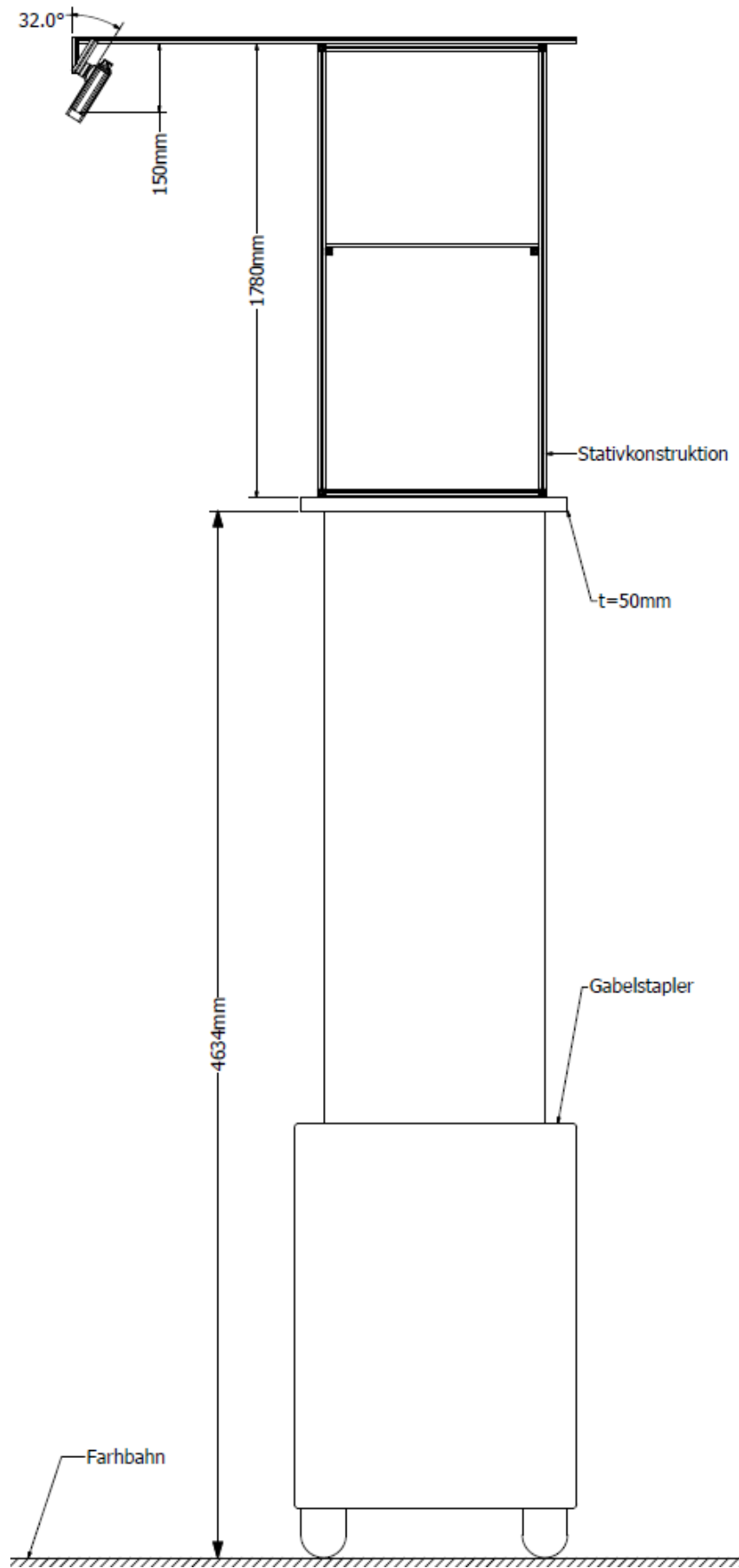


Abbildung 19: Vogelkamera auf Stativ und Gabelstapler (Seitenansicht)

Parameter bei der ersten Messreihe

Bei der ersten Messreihe ist die Vogelkamera, wie in Abbildung 14 (Seite 31) dargestellt, neben der Versuchsstrecke in einem Raum des ersten Stockwerks positioniert. Dabei wird die Höhe des Kamerasensors relativ zur Fahrbahn durch Messungen mit dem Laser-Entfernungsmesser ermittelt. Zunächst wird dabei die Höhe des Fenstersimses entlang der Außenfassade gemessen. Anschließend ergibt sich die Höhe des Sensors zur Fahrbahn durch Messung der relativen Höhe des Sensors und Fenstersimses zum Boden im Inneren des Raums. So wird eine Höhe von **8650 mm** erreicht. Außerdem wird der Neigungswinkel der Kamera durch Messung am Gehäuse mit **34° ± 0,5°** zum senkrechten Profil gemessen. Auch bei dieser Anordnung wird der theoretischen Winkel bei der Höhe und dem senkrechten Abstand zur Seitenstreifenbegrenzung berechnet. Anschließend wird der relative Winkel zwischen Kamerabildhorizontalen und dem Schnittpunkt der Kamerabild-Vertikalen mit dem Seitenstreifen analog zur 1. Messreihe berechnet. Für Aufnahmen der 1. Messreihe wird ein Neigungswinkel von **$\alpha=34,2^\circ$** ermittelt.

Nachdem die Neigung, Höhe sowie y-Position der Kamera bekannt ist, kann mit einer Aufnahme der Kamera, welche mit den internen Kameraparametern entzerrt wird die Verdrehung der Kamera in der x-y-Ebene bestimmt werden. Dazu werden zwei Punkte an der Seitenstreifenbegrenzung in Raumkoordinaten transformiert. Aus diesen Raumkoordinaten kann der gesuchte Verdrehwinkel β bestimmt werden. Dieser beträgt bei den ersten 3 Messungen der 1. Messreihe $7,1^\circ$ und bei den darauffolgenden $1,8^\circ$. Der Grund dafür, dass die Kamera verdreht ist und der Winkel nicht konstant bleibt, liegt darin, dass die Kamera im Zuge der Untersuchungen verschoben wird.

5.6. Versuchsdurchführung

Vor Einleitung des Einparkvorgangs, werden die Aufzeichnungen mit der Ladybug-Kamera bei 1 Fps sowie der Basler-Kamera bei 2 Fps in der Vogelperspektive gestartet. Bevor die Aufzeichnung aus der Vogelperspektive gestartet wird, wird ein Gamma-Ausgleich über die Pylon-Software ausgelöst, um wechselndes Tageslicht durch Anpassung der Belichtungszeit auszugleichen und möglichst konstante Belichtung der Aufnahmen zu erzielen. Das Versuchsfahrzeug wird versetzt zur Parklücke mit dem Heck an der Markierung Nr. 6 positioniert. Sobald die Video- und LIDAR-Aufzeichnungen gestartet sind, wird der rechte Blinker aktiviert und an der Parklücke vorbeigefahren, indem das Bremspedal bei eingelegtem D-Gang losgelassen wird. Dabei werden die Instrumententafel sowie der Eingabe-Bildschirm bis zum Heranfahren an Marker 3 beobachtet. Falls keine Parklücke detektiert und/oder der Befehl für das Starten des Einparkvorgangs auf dem Eingabe-Bildschirm erscheint, wird das Versuchsfahrzeug wieder bis zur Orts-Markierung Nr. 6 zurückgesetzt und ein neuer Anlauf gestartet.

Erscheint der Befehl zum Starten des Einparkvorgangs, so wird dieser über den Eingabe-Bildschirm ausgewählt, der Rückwärtsgang, wie aufgefördert eingelegt und das Bremspedal losgelassen. Daraufhin leitet das System den Einparkvorgang ein. Falls der Einparkvorgang abbricht, wird ein neuer Anlauf gestartet. Bei erfolgreichem Einparken, wird nach dem Bestätigungston die Skala am Lenkrad abgelesen. Außerdem werden Gamma-Ausgleich und anschließend Aufnahmen über die beiden Basler-Kameras an den Seiten des Fahrzeugs ausgelöst.

Die vogelperspektivische Aufnahme wird spätestens alle 10 Minuten neu gestartet, da die Software bei 2GB Dateigröße die Aufzeichnung automatisch abbricht. Die Ladybug-Aufnahmen werden nach 4-6 Messungen neu gestartet, um zum einen die Größe der Stream-File-Serien zu begrenzen und zum anderen das Risiko von Datenverlust zu verringern, falls einzelne Aufzeichnungen verloren gehen.

Zusätzlich werden Stichprobenartig mit dem Maßband Abstände zwischen Vorderrad und Bordstein sowie Hinterrad und Bordstein gemessen. Dabei wird das Maßband so angelegt, dass die 0 unter dem äußersten Punkt des Radaufstandspunktes anliegt und bis zum Fuß des Bordsteins gemessen.

Außerdem wird der Abstand zwischen einem fixen Punkt am hinter der Parklücke geparkten Fahrzeug und einem fixen Punkt am Nummernschild des Versuchsfahrzeugs mit dem Laser-Entfernungsmesser gemessen. Weiterhin wird von einem fixen Punkt am Front-Nummernschild des Versuchsfahrzeugs und einem fixen Punkt am Nummernschild des vorne geparkten Fahrzeugs mit dem Laser-Entfernungsmesser gemessen.

Zeiten zu den Messungen werden bei Einleitung des Einparkvorgangs oder spätestens am Ende des Einparkvorgangs notiert, um so später Aufzeichnungen den entsprechenden Messungen zuordnen zu können.

Es werden insgesamt 2 Messreihen an 2 Tagen durchgeführt worden. Die erste Messreihe erfolgt am 09.04.2018 und die zweite Messreihe am 13.06.2018.

5.7. Bestimmung der Zielmessgrößen

Für die Auswertung der aufgezeichneten Daten wird Matlab R2017b („MathWorks – Entwickler von MATLAB und Simulink“ 2018) verwendet. Dabei werden sowohl eigene Funktionen geschrieben, als auch bereits in Matlab integrierte oder in der Online-Community „mathworks“ angebotene Funktionen verwendet. Bei integrierten Funktionen wird lediglich der Name zitiert. Bei selbstgeschriebenen oder heruntergeladenen wird dies explizit erwähnt. Die verwendeten Toolboxes sind in der Hochschullizenz der Haw-Hamburg enthalten (Stand 21.07.2018). In diesem Kapitel werden die Konzepte zu Ermittlung der einzelnen Zielgrößen beschrieben. Die entsprechenden Programmcodes der Funktionen sind im Anhang zu finden. Die Methoden zur Bestimmung der Zielmessgrößen werden nach der ersten Messreihe überarbeitet, bzw. weiterentwickelt. Es folgt eine tabellarische Übersicht (siehe Tabelle 19) über die Methoden des ersten und zweiten Entwurfs, bevor diese Methoden in den folgenden Abschnitten einzeln im Detail beschrieben werden. Außerdem sind die gesuchten Größen in der folgenden Abbildung schematisch dargestellt.

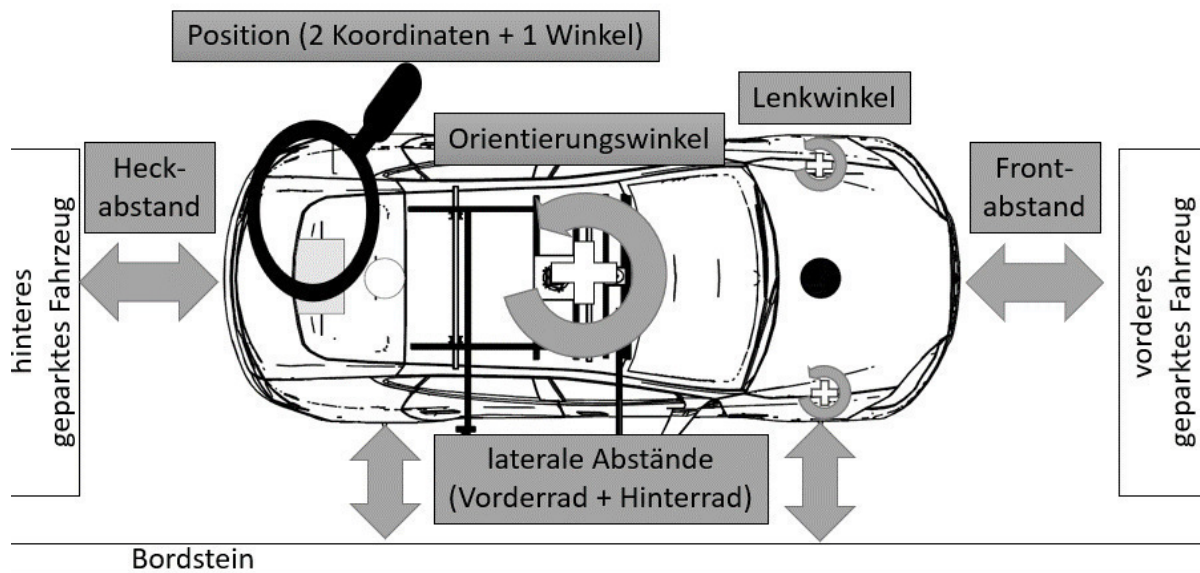


Abbildung 20: schematische Darstellung der gesuchten Zielmessgrößen

Ergänzend zur der Tabelle zur Übersicht der Messmethoden auf der folgenden Seite, folgt eine Übersicht der eingesetzten Komponenten. Aufzeichnungen aller mitgeführten Sensoren liegen vor. Im Rahmen dieser Arbeit konnten nicht alle Daten genutzt werden.

Komponenten des ersten Entwurfs	Komponenten des zweiten Entwurfs
<ul style="list-style-type: none"> • eine Markierung am Versuchs-Fahrzeug • 6 Orts-Markierungen • Vogelkamera • sphärische Kamera * • LIDAR-Sensor * • Lageplan • Messraster • Rad-Markierungskegel * 	<ul style="list-style-type: none"> • zwei Markierungen am Versuchs-Fahrzeug • 6 Orts-Markierungen • Vogelkamera • sphärische Kamera * • Inertial Measurement Unit * • Lageplan • Lenkwinkelraster am Lenkrad • Markierung an Bordstein
<p>*: Auswertung der Aufzeichnungen ist für die Bestimmung der gesuchten Größen im Rahmen dieser Arbeit nicht erfolgt oder nicht vollständig umgesetzt.</p>	

Tabelle 18: Übersicht der bei den Messungen eingesetzten Komponenten

Zielfmessgröße	Methode des ersten Entwurfs	Methode des zweiten Entwurfs	Abschnitt
Verlauf von Position und Orientierungswinkel des Versuchsfahrzeugs während des Einparkvorgangs	Auswertung vogelperspektivischer Aufnahmen mit geometrischer Transformation von Bild- in Raumkoordinaten	Auswertung vogelperspektivischer Aufnahmen	5.7.1 5.7.3
Einparkdauer	Auswertung vogelperspektivischer Aufnahmen	Auswertung vogelperspektivischer Aufnahmen	0
Anzahl der Parkzüge	Auswertung vogelperspektivischer Aufnahmen	Auswertung vogelperspektivischer Aufnahmen	5.7.5
Front- und Heckabstand	Auswertung vogelperspektivischer Aufnahmen mit geometrischer Transformation von Bild- in Raumkoordinaten und Vergleich mit relativer Positionen der geparkten Fahrzeuge zu einer Orts-Markierung	Auswertung von Seitenkamera-Aufnahmen mit geometrischer Transformation von Bild- in Raumkoordinaten und Vergleich mit relativer Positionen der geparkten Fahrzeuge zur Bordsteinmarkierung	5.7.10
lateraler Abstand	Auswertung von Seitenkamera-Aufnahmen bei Einsatz eines Messrasters	Auswertung von Seitenkamera-Aufnahmen mit geometrischer Transformation von Bild- in Raumkoordinaten bei Einsatz einer Bordstein-Markierung	0
Orientierungswinkel am Ende des Einparkvorgangs	-	-	5.7.9
Lenkwinkel	Auswertung von Seitenkamera-Aufnahmen mit markiertem Vorderrad	Ablese von Skala am Lenkrad für Lenkradwinkel	

Tabelle 19: Übersicht über die Methoden zur Ermittlung der Zielfmessgrößen

5.7.1. Positionsbestimmung

Bei der Bestimmung und Verfolgung der Position des Fahrzeugs wird der Fokus der Arbeit auf die Auswertung vogelperspektivischer Aufnahmen gelegt.

Dabei teilt sich die Positionsbestimmung in folgende Arbeitsschritte auf:

1. Verfolgen der Fahrzeugmarkierungen in den Bildsequenzen
2. Markierungsmittelpunkt finden zur Lokalisierung des Markierungspunktes auf dem Fahrzeug
3. Transformation der ermittelten Bildkoordinaten der Markierung in Raumkoordinaten

In den folgenden Abschnitten wird die Umsetzung der einzelnen Arbeitsschritte beschrieben. Die entsprechende Auswertung sphärischer Bildaufnahmen, LIDAR-Sensor-Daten und der Daten des IMU-Sensors kann, auf Grundlage der in dieser Arbeit gewonnenen Erkenntnisse, Gegenstand weiterführender Arbeiten sein.

5.7.2. Funktionsprinzip Marker-Tracking für Bildaufnahmen

Zunächst werden Videoaufnahmen als Bilderserien entsprechend einer Messung extrahiert. Dazu werden Start und Endzeit des jeweiligen Videos mit den notierten Zeiten sowie anderen Aufnahmen abgeglichen. Anschließend wird die Bilderserie von Anfang bis Ende durchgegangen. Sobald die zu verfolgende Markierung erscheint, ist die Startposition dieser Markierung per Mausklick einzugeben, wobei die Mitte nicht getroffen werden muss, die eingegebenen Koordinaten aber innerhalb des Markierungskreises liegen soll. Die eingegebene Position wird durch die selbstgeschriebene „Markermittendifinder“-Funktion korrigiert. Das Prinzip dieser Funktion wird im Anschluss an die Beschreibung des Ablaufs des Marker-Trackings erklärt. Zunächst folgt eine Beschreibung der Verfolgung von Bewegungen beim Vergleich zweier aufeinander folgender Bild-Aufnahmen.

Nachdem der Mittelpunkt der Markierung gefunden ist, wird ein ROI (Region of Interest) als auszuwertender Bildbereich gewählt. Für diesen ROI werden Eckpunkte durch die Algorithmen „detectSurfFeatures“ („Detect SURF features and return SURFPoints object - MATLAB detectSURFFeatures - MathWorks Deutschland“ o. J.) oder „detectHarrisFeatures“ („Detect corners using Harris–Stephens algorithm and return cornerPoints object - MATLAB detectHarrisFeatures - MathWorks Deutschland“ o. J.) gesucht. Bei den durchgeführten Versuchen stellt sich heraus, dass der Algorithmus „detectSurfFeatures“ eine größere Anzahl an Punkten findet. Mit den „extractFeatures“ („Extract interest point descriptors - MATLAB extractFeatures - MathWorks Deutschland“ o. J.) und „matchFeatures“-Algorithmen („Find matching features - MATLAB matchFeatures - MathWorks Deutschland“ o. J.) werden die Verschiebungen von als valide erkannten Punkten von einem Bild zum nächsten, für den gleichen ROI berechnet. Eine Darstellung so erkannter Punktepaare und deren Verschiebungen ist in Abbildung 21 und Abbildung 22 bei Überlagerung zweier aufeinander folgender Aufnahmen zu sehen.

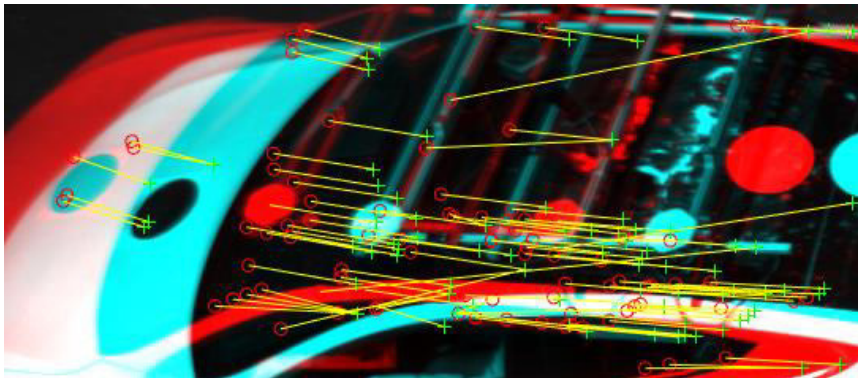


Abbildung 22: Darstellung von Verschiebungen detektierter "Surf-feature Points"-Paaren auf dem Versuchsfahrzeug (2. Entwurf)

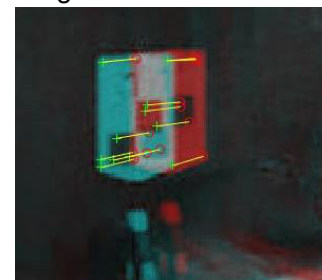


Abbildung 21: Darstellung von Verschiebungen detektierter "Harris-feature Points"-Paarungen an Orts-Markierung (1. Entwurf)

Ausreißer werden mit der Funktion „isoutlier“ („Find outliers in data - MATLAB isoutlier - MathWorks Deutschland“ o. J.) für die x- und y-Spalte der Matrix, welche all die detektierten Verschiebevektoren beinhaltet separat detektiert. Anschließend wird jede Zeile, welche einen Ausreißer besitzt aus der Verschiebematrix eliminiert. Im letzten Schritt wird für die Bestimmung des Verschiebungsvektors der Median für jede der beiden Spalten der Verschiebematrix ausgewählt.

Der so bestimmte Verschiebungsvektor wird zu der Markierungs-Position des ersten Bildes addiert.

In der folgenden Abbildung ist Aus Messung Nr. 25 der zweiten Messreihe ein Frame mit eingezeichneter aktueller Markierungsposition (großes „X“) sowie den vorherigen Positionen verbunden zu einer Trajektorie (Linie mit kleinen „x“) zu sehen. Bei der Darstellung wird das Schwarzweiß-Bild in gelben und blauen Farbtönen ausgegeben.

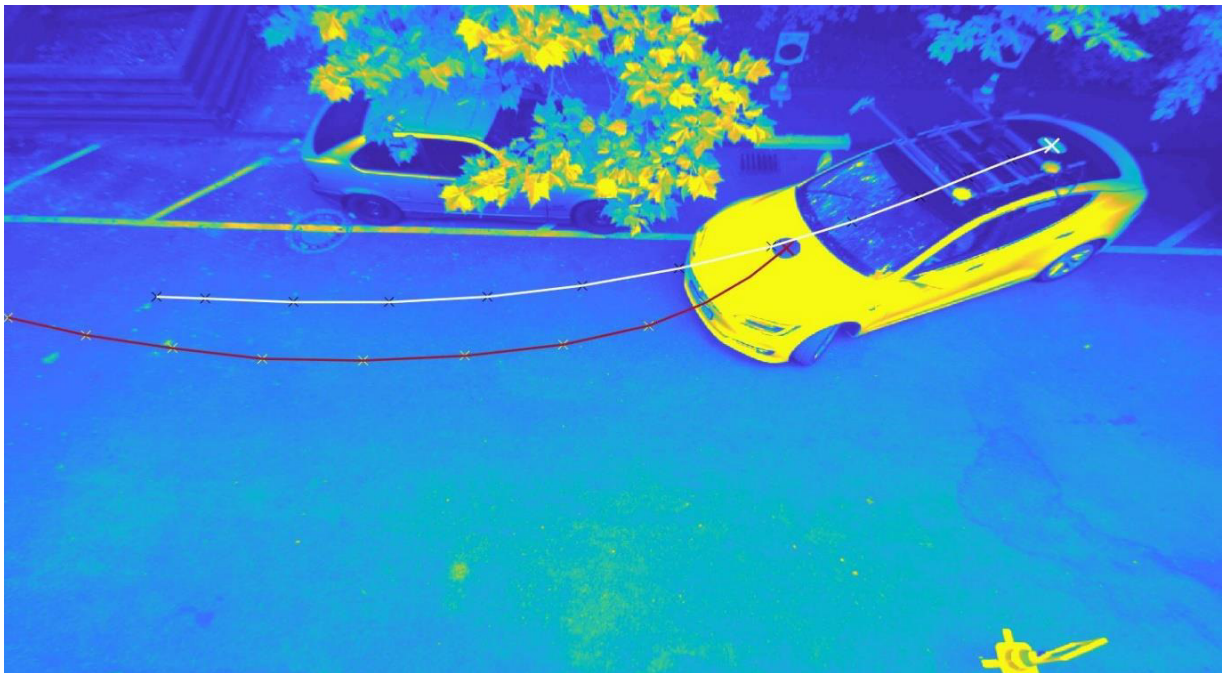


Abbildung 23: Markierungs-Positions-Verlauf beim Marker Tracking (Messreihe 2, Messung 33)

Die Position der Markierung im vorherigen Frame wird mit seinem Intensitätswert als Referenzpunkt gewählt. Falls der Intensitätswert des so bestimmten Punktes nicht stärker von dem des Referenz-Punktes abweicht, als der eingegebene Abweichungs-Parameter dem alten Intensitätswert multipliziert zulässt, wird angenommen, dass die neue Position sich auf dem Marker befindet. Andernfalls wird noch ein parametrisch bestimmter Anteil des Verschiebungsvektors zu der Position hinzuaddiert. Der so bestimmte Punkt wird erneut mit dem Referenzpunkt verglichen. Dieses wird wiederholt, bis die Abweichung von dem Referenzwert den Schwellwert nicht mehr überschreitet. Ab hier fängt der beschriebene Ablauf von neuem an und es wird im nächsten Schritt die Mitte des Markers gesucht. Die Beschreibung dieser Funktion folgt im nächsten Abschnitt.

„Markermittendifinder“-Funktion

Erster Entwurf

In Abbildung 24 ist eine Aufnahme der ersten Messreihe mit dem ersten Entwurf der Fahrzeugmarkierung abgebildet. Für diesen werden die Ränder der Markierung gesucht, indem nach Bildpunkten mit kleineren Intensitätswerten als 200 gesucht werden. Da die schwarzen Figuren innerhalb der Markierung nicht als Rand detektiert werden sollen, wird festgelegt, dass erst bei 15 aufeinander folgenden Bildpunkten der Rand der Markierung detektiert ist. Dies erfolgt dabei nur für eine Zeile und eine Spalte. Aus den Markierungs-Rand-Koordinaten werden die Koordinaten der Markierungs-Mitte als gerundeter arithmetischer Mittelwert berechnet. Die Funktion verliert bei einer der Messungen die Markierung, als der im Bild links neben ihm liegende Träger hell aufleuchtet.



Abbildung 24: Kamera-Bild-Aufnahme der ersten Messreihe

Zweiter Entwurf

Als Startpunkt wird die nach der Auswertung der Verschiebung bestimmte Position übergeben (Bsp.: siehe Abbildung 25). Von diesem ausgehend werden für Zeilen und Spalten in alle Richtungen des Bildes Mittelpunkte gesucht. Dazu wird nach den Grenzen der Markierung gesucht, indem für die Bildpunkte auch hier der Intensitätswert mit dem Startpunkt verglichen wird. Ist die Abweichung kleiner, als durch den eingegebenen Abweichungs-Parameter erlaubt, wird dieser als Punkt innerhalb der Markierung betrachtet und es wird in der Zeile oder Spalte weiter nach der Grenze der Markierung gesucht. Sobald die äußersten Grenzen für die Zeilen und Spalten der Markierung gefunden sind, wird mit den Koordinaten der Mittelpunkt für die Zeile oder Spalte als gerundeter arithmetischer Mittelwert berechnet.

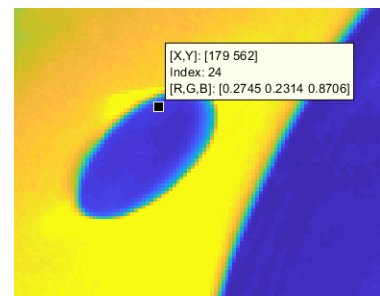


Abbildung 25: Beispiel für einen Startpunkt für die „Markermittendifinder“-Funktion

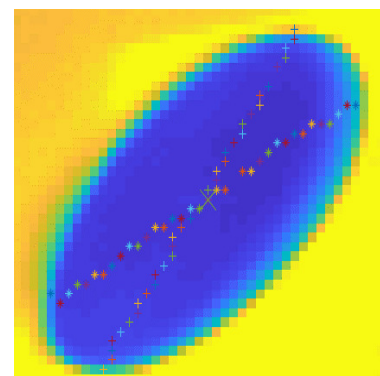


Abbildung 26: detektierte Zeilen- und Spaltenmittelpunkte

Im finalen Schritt, wird aus allen Zeilenmittelpunkten die mittlere y-Koordinate (Zeilenindex) und aus allen Spaltenmittelpunkten die x-Koordinate (Spaltenindex) bestimmt. Diese ergibt sich als arithmetisches Mittel aus größter und kleinster Koordinate, bestimmt mit den Funktionen „max“ („Maximum elements of an array - MATLAB max - MathWorks Deutschland“ o. J.) und „min“ („Minimum elements of an array - MATLAB min - MathWorks Deutschland“ o. J.), mit anschließender Rundung. Die Funktion „round“ („Round to nearest decimal or integer - MATLAB round - MathWorks Deutschland“ o. J.) rundet dabei bis Dezimalstellen kleiner 5 ab und ab 5 auf.

Die Markierung auf der Heckklappe ist stets von einer ausreichenden Anzahl detektierter Eckpunkte umgeben, sodass die Verschiebung in den meisten Fällen die neue Markierung Position trifft. Im Anschluss wird der Markierungs-Mittelpunkt durch einen Algorithmus für die Heckmarkierung gesucht.

Beim Tracking der Markierung auf Fronthaube werden in den meisten Fällen keine Eckpunkte auf der Haube oder am Rand gefunden. Deshalb wird der betrachtete Bildbereich Richtung Heck vergrößert. Dieses führt dazu, dass die Verschiebungen der detektierten Eck-Punkte stärker von der Verschiebung der Markierung abweichen, als dies bei der Heckmarkierung der Fall ist. Um die Markierung trotzdem zu finden, wird zunächst ein Bruchteil der berechneten Verschiebung zur alten Marker-Position addiert und der Intensitätswert für diesen sowie Punkte, welche 20, 30 oder 40 Bildpunkte unterhalb von ihm liegen betrachtet. Da die Markierung von einer ausreichend großen, relativ homogenen Fläche bezüglich der Intensitätswerte von über 200 umgeben ist, kann so die Markierung gefunden werden. Der Parameter, welcher die Schrittweite bei der Suche nach der Markierung bestimmt, wird für Messungen, bei denen die Markierung verloren wird für die jeweiligen Sequenzen angepasst. Dabei wird dieses wiederholt, bis die Markierung gefunden wird.

Markierungen am Fahrzeug

Das Fahrzeug wird mit auf Papier gedruckten Kreisen bei möglichst großem Kontrast zum Hintergrund markiert. Im ersten Entwurf sitzt die Markierung (Durchmesser: 200mm) hinter der Ladybug-Kamera mittig zwischen den Außenkanten des Sonnendachs. Folgende Abbildung zeigt diese Anordnung.

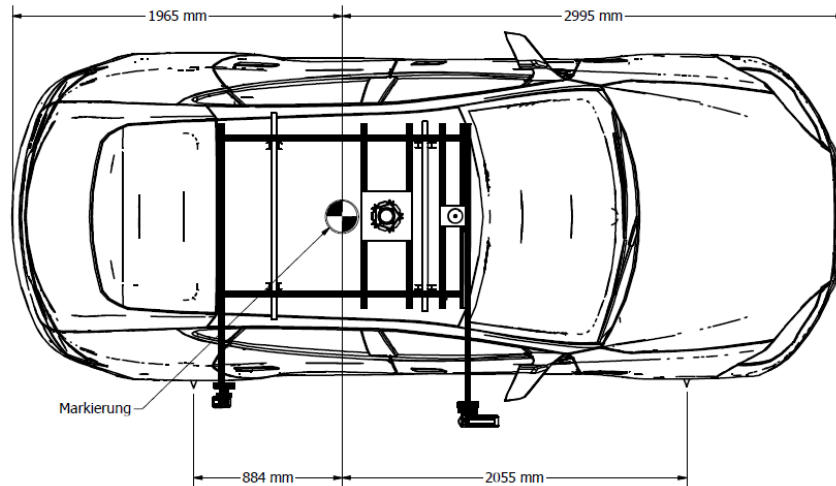


Abbildung 27: Lage der Markierung auf Fahrzeugdach (1. Entwurf)

Im überarbeiteten Entwurf werden zwei einfarbige, runde Markierungen mit größerem Durchmesser von 270mm, wie in folgender Darstellung zu sehen auf dem Fahrzeug positioniert. Diese sind mittig über den Rad-Achsen am Fahrzeug angeklebt. Die Fronthauben-Markierung ist in schwarz ausgeführt, da die Haube weiß lackiert ist. Dagegen ist die Markierung auf der Heck-Klappe in weiß ausgeführt, da das Sonnendach und die Heckklappen-Scheibe mit einem Spalt dazwischen ineinander übergehen und aus verdunkeltem Glas ausgeführt sind (siehe dazu Abbildung 29).

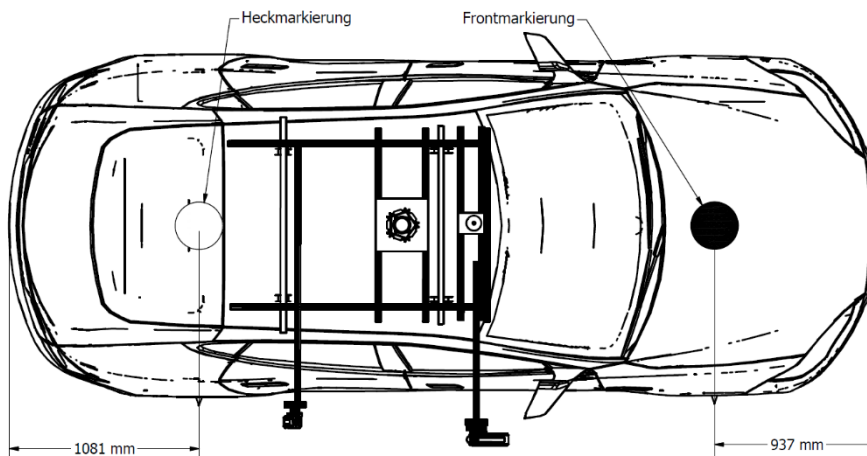


Abbildung 28: Lage der Markierungen über Vorder- und Hinterachse (2. Entwurf)

Die Anpassung der Fahrzeugmarkierungen erfolgt aus folgenden Gründen:

- Eine Verdeckung der Markierung durch Profile, wie beim alten Entwurf soll nicht wieder auftreten
- Ein „Marker-Detection“-Ansatz (Hirzer 2008) wäre bei einer größeren Anzahl an Markierungen zur Identifizierung der Markierungen sinnvoll, da die Umsetzung einen höheren Aufwand benötigt
- Der Orientierungswinkel des Fahrzeugs soll durch Lokalisierung von zwei Markierungen bestimmt werden
- Abstände der Rad-Außenflanken zum Bordstein sollen aus den Markierungsendpositionen ermittelt werden
- Ein größerer Kontrast zum Hintergrund vereinfacht das Tracking der Markierung
- Mit dem neuen Weitwinkel-Objektiv (Kowa LM6HC) erscheinen die Markierungen im Bild kleiner und das Platzangebot lässt eine Vergrößerung der Markierungen zu



Abbildung 29: Markierung auf der Heckklappe des Versuchsfahrzeugs (links im Bild: Einklemmschutz)

Orts-Markierungen auf der Versuchsstrecke

Für die Orientierung im Raum mit sphärischen Bildaufnahmen oder LIDAR-Daten werden Orts-Markierungen entworfen, welche aus Pylonen mit Schildaufsätzen mit aufgeklebten Markierungen im A3-Format bestehen. Beim ersten Entwurf handelt es sich um ein 2 x 2 schwarzweiß-Karomuster und die Mitte der Markierung befindet sich auf einer Höhe von 740 mm zur Pylonenfuß-Unterseite. Außerdem wird bei diesem Entwurf zur Unterscheidung eine eingekreiste Zahl mittig über dem Muster platziert.

Beim überarbeiteten Entwurf sind die Ausdrücke als schwarze Kreise auf weißem Papier ausgeführt. Die Anpassung hat die selben Gründe, wie die im letzten Abschnitt erwähnten. Schwarze Kreise auf hellem Hintergrund sollen mit Detektierung von Verschiebungen umgebender „Feature“-Points verfolgt werden. Durch Abtasten der Kreisfläche wird anschließend die Mitte der Markierung gefunden. D.h. es ist möglich, das Konzept für das Verfolgen der Markierungen, für die Aufnahmen der sphärischen Kamera, mit den dort sichtbaren Orts-Markierungen umzusetzen.

In dieser Arbeit wird bei der ersten Messreihe die Markierung Nr. 1 als Referenz für die Transformation vogelperspektivischer Bildkoordinaten verwendet. Bei dem zweiten Entwurf wird die Funktion für das Finden des Markierungs-Mittelpunktes verwendet, um für jedes Bild der Aufnahme die Position der Markierung automatisiert zu bestimmen und die. Näheres dazu folgt im nächsten Abschnitt.

Bei dem zweiten Entwurf beträgt die Höhe des Markierungs-Mittelpunktes 760 mm zur Pylonenfuß-Unterseite. Die Abmessung des Pylonenfußes beträgt 240 mm x 240 mm und die Markierungsausdrücke werden zu den Seiten des Schildaufsatzes mittig auf diesen angebracht. Demnach kann die Lage des Mittelpunktes der Markierung in der x-y-Ebene mittig bezüglich der Pylonenfuß-Außenseiten angenommen werden. Die Dicke des Schildaufsatzes von ca. 4mm wird dabei vernachlässigt.

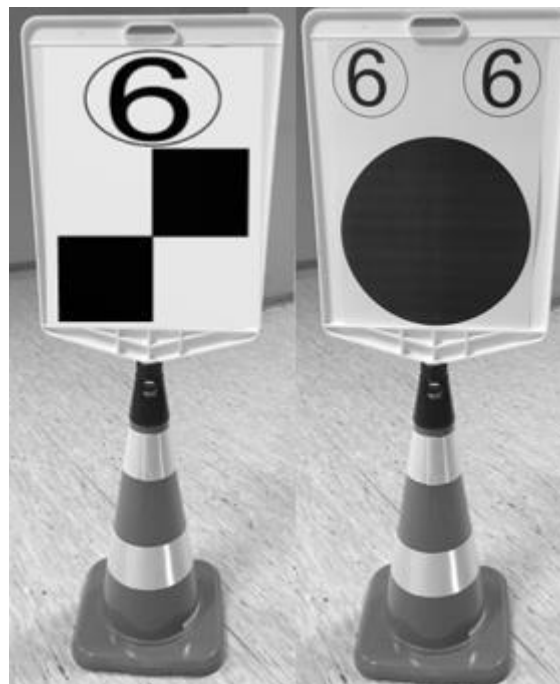


Abbildung 30: Versuchsfeld-Markierungen

5.7.3. Transformation der Bildposition in Raumkoordinaten

Für die Transformation von Bildkoordinaten in Raumkoordinaten wird eine selbstgeschriebene Funktion verwendet. Dazu wird die Lage des **Kamera-Koordinatensystems** wie in Abbildung 31 dargestellt festgelegt. Demnach liegt der Ursprung auf dem Bild-Sensor der Kamera und die z-Achse ist in die Aufnahme-richtung gerichtet. Dieses Koordinatensystem hat **keinen direkten Bezug zu den Bildkoordinaten**. Es dient nur zur Beschreibung der relativen Lage anderer Koordinatensysteme zu der Kamera.

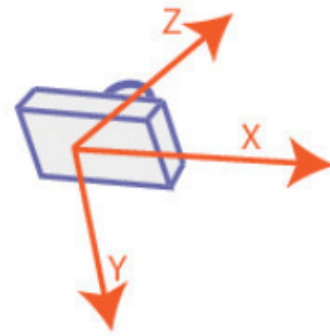


Abbildung 31: Kamera-Koordinatensystems

Quelle: („Coordinate Systems - MATLAB & Simulink - MathWorks Deutschland“ o. J.)

Das **Bild-Koordinatensystem** hat seinen Ursprung in der oberen linken Ecke des Bildes und verläuft damit wie in der folgenden Darstellung abgebildet. Da das Bild nur eine Ebene hat, fehlt hier die z-Achse.

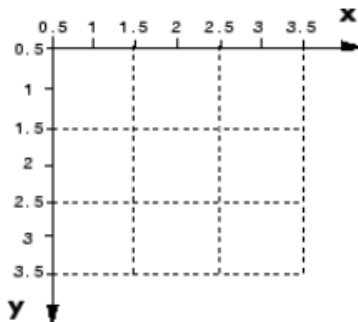


Abbildung 32: Bild-Koordinatensystem

Quelle: („Coordinate Systems - MATLAB & Simulink - MathWorks Deutschland“ o. J.)

Das globale Koordinatensystem wird bei den Untersuchungen auf eine Ortsmarkierung gelegt, sodass die x-y-Ebene der Fahrbahnebene entspricht und die z-Achse senkrecht zur Fahrbahn stehend nach oben gerichtet ist. Folgende Abbildung zeigt die Lage des Koordinatensystems.

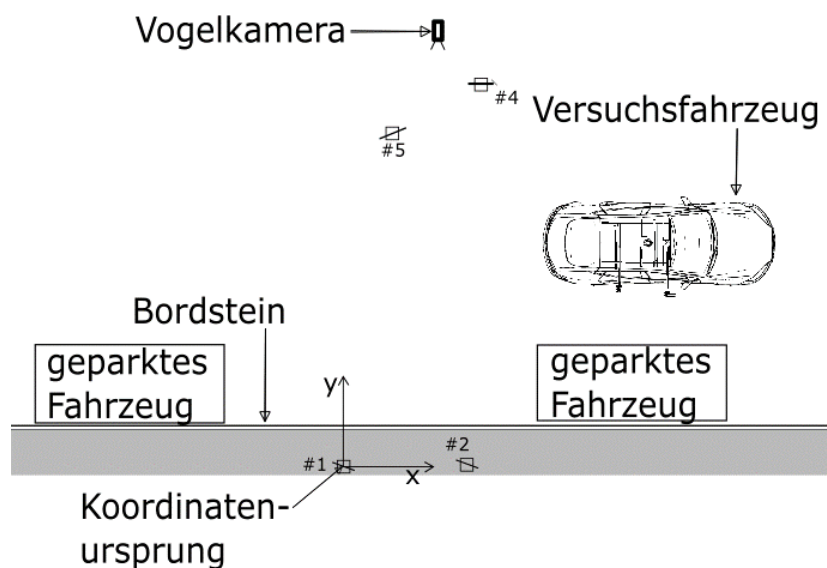


Abbildung 33: Lage des globalen Koordinatensystems

Die Transformation durchläuft folgende Schritte:

1. Transformation in ein Koordinatensystem, welches um $90^\circ + 90^\circ - \alpha$ (Neigungswinkel) gedreht und auf die Höhe der Fahrbahn gelegt wird (Funktion „KoordzuRaumPosV4“ – siehe Anhang)
2. Transformation in ein um $-\beta$ gedrehtes Koordinatensystem bezüglich der z-Achse (Funktion „KoordzuRaumPosV4“ – siehe Anhang)
3. Messung der relativen Lage zum Ursprung des globalen Koordinatensystems durch Transformation (nach Schritt 1 und 2) der Lage des Ursprungs des globalen Koordinatensystems in Raum-Koordinaten (liegt in Markierung Nr. 1)

Die Beschreibung der Umsetzung einzelner Schritte erfolgt im weiteren Verlauf dieses Abschnittes. Der vollständige Programmcode ist im Anhang zu finden. Vorher wird auf einige getroffene Annahmen eingegangen:

Für die Transformationen wird angenommen, dass die optische Achse als Rotations-Symmetrie-Achse des Objektivs in der Mitte des Sensors ihren Ursprung hat, sodass diese im Bild als Punkt, durch das optische Zentrum repräsentiert wird (siehe Abschnitt 2.4.4). Weiterhin wird angenommen, dass der vertikale Öffnungswinkel der Kamera über die gesamte Breite des Bildes konstant ist und der horizontale Öffnungswinkel über die gesamte Höhe des Bildes konstant ist. Dabei bezieht sich die Annahme auf eine ganze, unbeschnittene Aufnahme. Da die Aufnahmen entzerrt werden, werden Zeilen und Spalten des Bildes in unterschiedlichem Maße gestreckt, sodass die Zeilen und Spalten unterschiedliche Anzahlen beschriebener Bildpunkte aufweisen. Die folgende Abbildung zeigt diesen Zusammenhang am Beispiel einer entzerrten Aufnahme der verwendeten monochromen Kamera mit dem Weitwinkelobjektiv. Die Anzahl der Bildpunkte für die Zeilen und Spalten des Bildes wird bei der Transformation eingegeben. Mit den Öffnungswinkeln der Kamera wird für den betrachteten Punkt der entsprechende Winkel pro Pixel ausgerechnet.

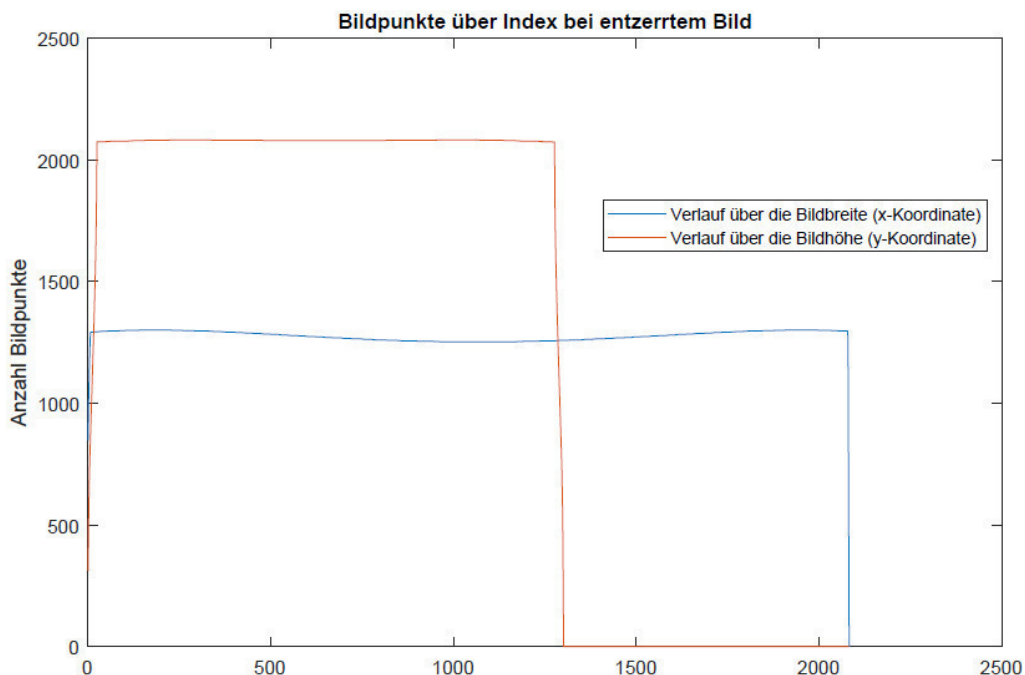


Abbildung 34: Bildpunktanzahl über Breite und Höhe einer entzerrten Aufnahme

In dem entsprechenden Abschnitt der Funktion werden Öffnungswinkel der Kamera wie folgt festgehalten und die Bildpunktanzahlen nach den Indizes der eingegebenen Koordinaten aus dem entsprechenden Vektor ausgelesen.

```

cam_Winkel_V      = 61.222 ;      % mit Kowa LM12HC Objektiv: 31.7;
cam_Winkel_H      = 86.558 ;      % mit Kowa LM12HC Objektiv: 48.6;

cam_Res_V         = Bildpunktzahl(1,Koord(1)); % vertikale Auflösung
cam_Res_H         = Bildpunktzahl(2,Koord(2)); % horizontale Auflösung

```

Für die erste Transformation wird zunächst die relative Lage des betrachteten Punktes zum optischen Zentrum, welcher der Funktion über die Kamera-Parameter übergeben wird berechnet.

```

% Relativer y-Abstand zum optischen Zentrum der Kamera (im Bild)
delta_y_p         = cameraParams.PrincipalPoint(2) - Koord(2);
% Relativer x-Abstand zum optischen Zentrum der Kamera (im Bild)
delta_x_p         = Koord(1) - cameraParams.PrincipalPoint(1);

```

Aus den Abständen zum optischen Zentrum werden die Relativwinkel φ_{km} und τ_{km} mit den entsprechenden Winkeln pro Pixel berechnet.

```

% Relativwinkel zwischen dem betrachteten Punkt und optischer Achse
% bei Rotation um die Kamerabildhorizontale. Der Drehsinn entspricht dabei
% der rechten Hand Regel, sodass Punkte oberhalb des optischen Zentrums
% positive Winkel ergeben.
phi_km            = ( delta_y_p * cam_Winkel_V / cam_Res_V );

% Relativwinkel zwischen dem betrachteten Punkt und optischer Achse
% bei Rotation um die Kamerabildvertikale. Der Drehsinn ist so gewählt, dass
% Punkte rechts vom optischen Zentrum Positive Winkel ergeben (linke Hand
% Regel).
tau_km            = ( delta_x_p * cam_Winkel_H / cam_Res_H );

```

Die Höhe der Kamera bezüglich der Fahrbahnebene wird in der Funktion eingeschrieben und eine Höhe des betrachteten Punktes wird der Funktion bei Anwendung übergeben, sodass eine Differenzhöhe berechnet wird. Die Höhe des betrachteten Punktes ist variabel, um die Funktion auf die unterschiedlichen Markierungen anwenden zu können.

```

% Hier wird die Höhe der Kamera in Relation zur Fahrbahn eingegeben.
% Bei der ersten Messreihe beträgt die Höhe der Kamera 8650mm und
% im zweiten Entwurf 6314mm.
h_cam             = 4634 + 1780 - 150 + 50;

% Höhendifferenz zwischen Kamera und Marker
h_km              = h_cam - h_marker;

```

Die Koordinatenachsen des ersten Ziel-Koordinatensystems, welches entsprechend dem Neigungswinkel der Kamera gedreht und auf der Fahrbahnebene liegt, werden mit d_{km} und b_{km} benannt. Die räumliche Lage ist in folgender Darstellung abgebildet.

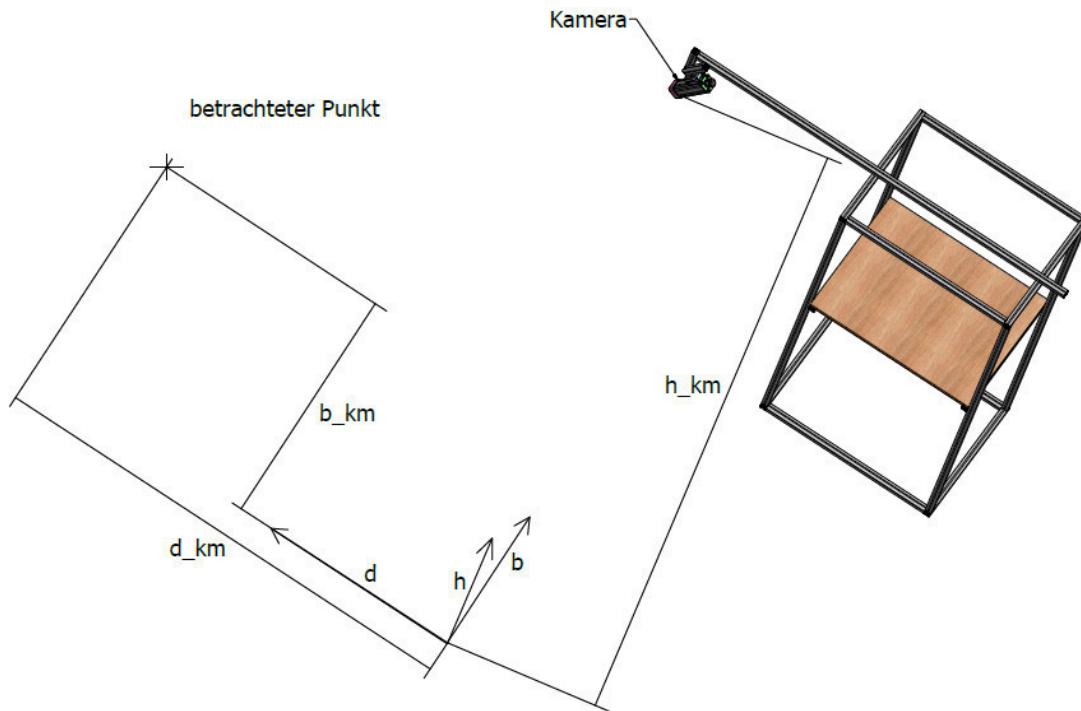


Abbildung 35: Räumliche Lage des ersten Ziel-Koordinatensystems auf der Fahrbahnebenen bei der Transformation von Bildpunkten in Raumkoordinaten

Mit dem Winkel φ_{km} und bekanntem Neigungswinkel α sowie der Höhendifferenz zwischen Kamera und betrachtetem Punkt wird die Koordinate d_{km} berechnet. Diese verläuft entlang einer imaginär auf die Fahrbahnebene projizierten Kamerabildvertikalen.

$$d_{km} = h_{km} * \tan(\alpha_k + \varphi_{km});$$

Mit d_{km} und der Summe der Winkel α_{km} und φ_{km} wird der senkrechte Abstand l_{km} zwischen der Kamerabildhorizontalen und der in den betrachteten Punkt projizierten Kamerabildhorizontalen berechnet.

$$l_{km} = d_{km} / \sin(\alpha_k + \varphi_{km});$$

Mit dem bekannten Relativwinkel τ_{km} und dem Abstand l_{km} wird der zu l_{km} senkrechte Abstand b_{km} zum betrachteten Punkt berechnet. Diese Größe entspricht der zweiten Koordinate im ersten Ziel-Koordinatensystem (siehe Abbildung 35).

$$b_{km} = l_{km} * \tan(\tau_{km});$$

Die ermittelten Koordinaten werden mit dem Verdrehwinkel β_{km} in das zweite Ziel-Koordinatensystem transformiert.

$$\begin{aligned} y_{rel_cam} &= \cos(\beta_k) * d_{km} - (b_{km} * \sin(\beta_k)); \\ x_{rel_cam} &= b_{km} * \cos(\beta_k) + d_{km} * \sin(\beta_k); \end{aligned}$$

Auflösung und Messbereich der Messmethode

Mit der Transformationsfunktion werden für die relevanten Bereiche für die x- und y-Raumkoordinaten folgende Auflösungen erreicht:

Bereich	x-Auflösung [mm]	y-Auflösung [mm]	Zeitliche Auflösung [Hz]
Parklücke, Frontachse	7,1	12,1	2
Parklücke, Hinterachse	8,6	11,6	2
Startposition	9	8,2	2
Orts-Markierung Nr. 2	8,3	14,1	2

Tabelle 20: Auflösung der Messmethode Transformation vogelperspektivische Aufnahmen

Aus den Werten für die Auflösung der Raum-Koordinaten werden Werte für die Auflösung des aus diesen Werten bestimmten Orientierungswinkels des Fahrzeugs berechnet. Dieser ergibt sich mit dem bekannten Radstand aus dem Tangens zu:

$$\theta_v = \arctan\left(\frac{\Delta y_{f,h}}{R_s}\right) \quad (6)$$

- θ_v : Orientierungswinkel aus Vogelkamera-Messung
 $\Delta y_{f,h}$: y-Auflösung für beide Markierungen addiert
 R_s : Radstand

Bereich	Auflösung für θ_v [°]
Parklücke	0,45
Startposition	0,32

Tabelle 21: Auflösung für Orientierungswinkel aus vogelperspektivischen Aufnahmen

Durch Einsetzen der äußersten Bildkoordinaten ergeben sich für den Messbereich 11,277 m in Richtung globaler x-Koordinate und 9,1 m in Richtung y. Bezogen auf den globalen Koordinatenursprung ergeben sich daraus folgende Werte:

y-Messbereich [m]		x-Messbereich [m]	
von	bis	von	bis
0	8,27	-2,65	8,62

Tabelle 22: Messbereich der Messmethode zur Bestimmung des Bewegungsverlaufs durch Auswertung vogelperspektivischer Aufnahmen

Für den Orientierungswinkel des Fahrzeugs werden aus Messungen der Bordsteinabstände Werte von maximal 3,6° ermittelt und arithmetisch gemittelt ein Wert von 1,35°. Deshalb können bei Vernachlässigung des Winkels, Betragswerte der y-Koordinaten mit Betragswerten senkrecht zur Fahrzeuglängsachse sowie Betragswerte der x-Koordinate mit Betragswerten entlang der Fahrzeuglängsachse verglichen werden. In Abschnitt 3.1 sind für Abstände zum Bordstein und zu den geparkten Fahrzeugen geforderte Auflösungen von 1 mm festgelegt. Dieses wird mit dieser Messmethode um ein Vielfaches überschritten. Für die Verfolgung der Fahrzeugposition ist als Auflösung 3mm bei mindestens 1Hz gefordert. Damit wird die geforderte Auflösung der Raumkoordinaten nicht erreicht.

5.7.4. Messung der Einparkdauer

Die Aufnahmen der vogelperspektivischen Kamera zeichnen bei bekannter Frequenz zeitdiskret Videos auf. Aus den Videos werden entsprechend dem Start und Ende des Einparkvorgangs Bilder extrahiert. Dabei wird nach Durchsicht der Videos als Startzeit der letzte Zeitpunkt vor der ersten Bewegung des Fahrzeugs notiert und als Endzeit diejenige, bei dem es zum Stillstand kommt. Die Differenz zwischen Start- und Endzeit ergibt die Einparkdauer. Die Bildrate der Aufzeichnung beträgt 2 Hz und da die verwendete Software die Aufzeichnung nach 2 GB Datenvolumen abbricht, beträgt die maximale Einparkdauer 456 s. Dies entspricht der Länge eines Videos, welches beim Abbruch erzeugt wird.

5.7.5. Zählung von Parkzügen

Nachdem der Verlauf der Bildposition der Fahrzeugmarkierungen verfolgt ist, wird aus diesen Verläufen die Parkzuganzahl berechnet und mit manueller Zählung bei Durchsicht der Bewegungsverläufe in Plots abgeglichen. Die Berechnung der Parkzuganzahl erfolgt dabei durch Zählen von Überschreitungen eines Schwellwertes.

5.7.6. Notierung von Abbruch nach eingeleitetem Einparkvorgang

Bei einem Abbruch eines eingeleiteten Einparkvorgangs ist dieses durch das Versuchspersonal unter Angabe der Zeit zu notieren, bzw. direkt dem jeweiligen Einparkvorgang zuzuordnen.

5.7.7. Messung des Lenkwinkels

Für die Messung des Lenkwinkels am Rad werden mit der CAD-Software Autodesk Inventor 2018 („Inventor 2018“ 2018) Markierungskegel zur Applikation an den Radmuttern des Rades entwickelt. Dieser Entwurf ist in Abbildung 36 zu sehen. Die Sicht der Kamera auf den Kegel nach der Applikation kann mit dem CAD-Modell der gesamten Konstruktion geprüft werden (siehe Abbildung 37). In Abbildung 39 ist eine Aufnahme zu sehen, in der der applizierte Kegel zu sehen ist. Durch Bestimmung der Orientierung des Kegels kann der Lenkwinkel des Rades bestimmt werden. Außerdem wird entsprechend des Lenkraddurchmessers eine Skala entworfen und zusammen mit einem Zeiger am Lenkrad und der dahinter liegenden Konsole appliziert. Mit der Skala können Lenkradwinkel von -4° bis $+4^\circ$ abgelesen werden. Bei den durchgeführten Messungen stellt sich heraus, dass der durch die Skala abgedeckte Bereich nicht ausreicht. Die Entwicklung eines Programms zur Ermittlung der Orientierung des Kegels steht noch aus.

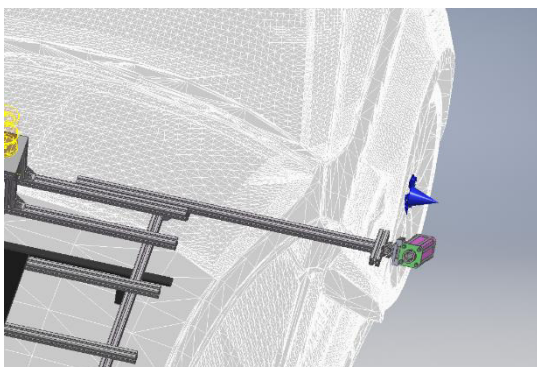


Abbildung 36: Lage der Seitenkamera zu Radmarkierungskegel

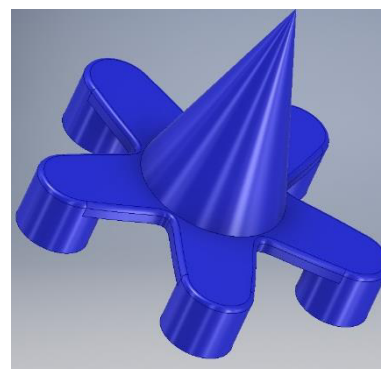


Abbildung 37: CAD-Modell vom entworfenen Radmarkierungskegel

5.7.8. Messung des Abstandes zwischen Bordstein und Rad-Außenflanke

Die Vermessung der Abstände zwischen den Reifen und dem Bordstein ist über Aufnahmen der Seiten-Kameras vorgesehen. Dazu werden die Kameras an der Seite des Sensorträgers angebracht und deren Position sowie Orientierung vermessen. Höhe sowie Abstand zur den Aufstandslinien werden dabei mit Hilfe des Laser-Entfernungsmessers aufgenommen und die Winkel entsprechend dieser Maße berechnet. Im ersten Entwurf sieht diese Anordnung wie in Abbildung 38 dargestellt aus und wird später angepasst. Die entsprechenden Darstellungen folgen im Abschnitt zum überarbeiteten Entwurf.

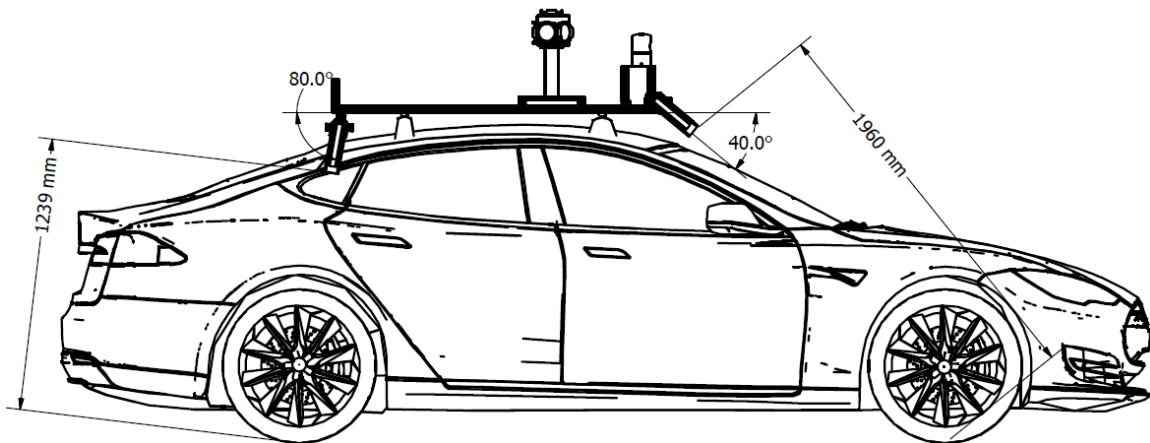


Abbildung 38: Seitenansicht der Lage und Orientierung der Seitenkameras am Versuchsfahrzeug

Erster Entwurf

Im ersten Entwurf der Messmethode wird am Ende des Einparkvorgangs ein Schwarz-Weiß-Raster mit 2mm breiten Streifen an den Bordstein angelegt und an die Radaufstandsfläche herangeschoben. Dieses wird über die Seiten-Kameras aufgenommen. Ein Beispiel für eine solche Aufnahme ist in Abbildung 39 zu sehen.

Mit der Funktion „Rasterauswertung“ werden die Streifen zwischen Radaufstandspunkt und Bordstein gezählt. Dies erfolgt über Zählung von Vorzeichenwechsel des Intensitätsgradienten. In folgender Abbildung ist der Verlauf der Intensitätswerte die in Abbildung 39 markierte Zeile zu sehen. Das Raster verläuft hier ca. bis zum Index 225.

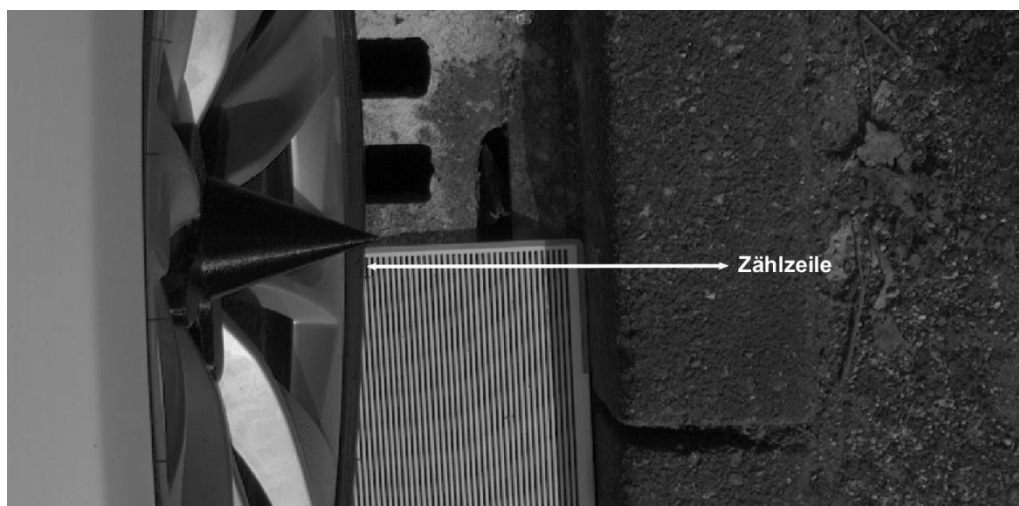


Abbildung 39: Ausschnitt aus einer Aufnahme der hinteren Seitenkamera bei angelegtem Mess-Raster (1. Entwurf)

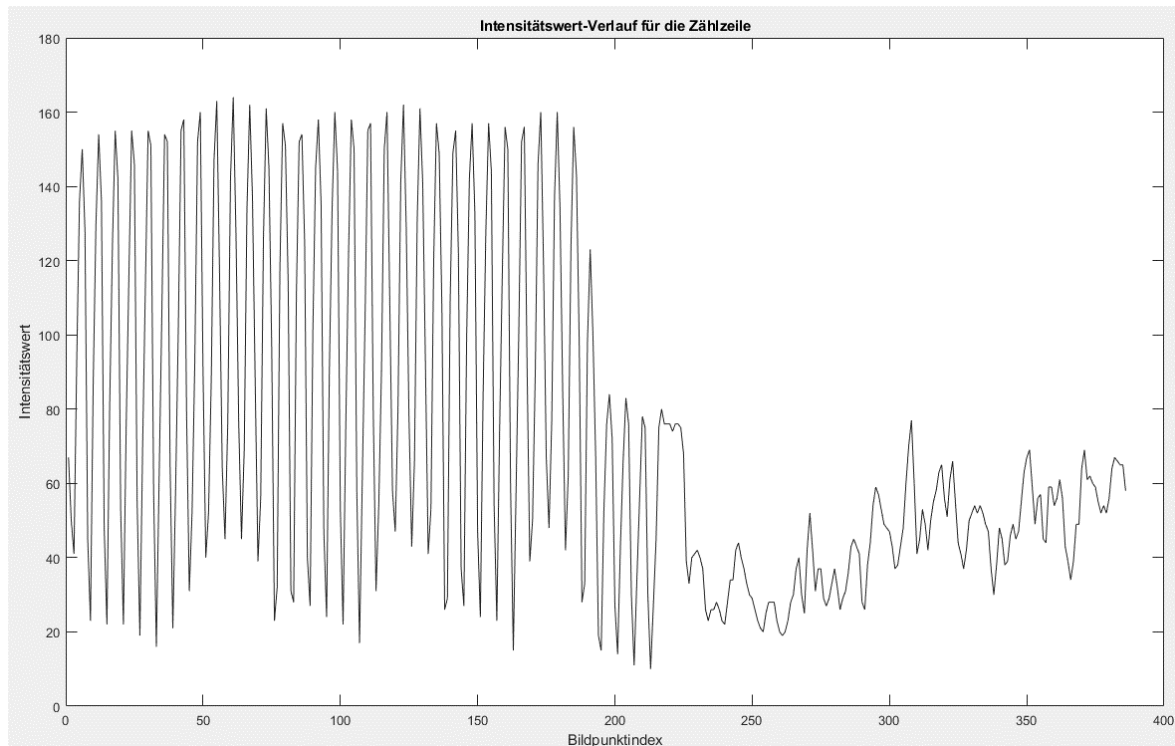


Abbildung 40: Verlauf von Intensitätswerten für gewählte Zählzeile

Dabei werden die Maxima in dem Bereich um den Bildpunktindex 200 kleiner, da dort der Schatten des Bordsteins liegt (siehe Abbildung 39). Hinter dem Bereich mit den kleineren Maxima kommt ein relativ breites lokales Maximum, welches den breiten weißen Streifen am Rand des Messrasters darstellt. Anhand dieses Maximums wird das Ende der Zählzeile detektiert.

Der Startpunkt für die Zählung ist bei allen Messungen fest und wird bestimmt, indem in einer Aufnahme, bei der das Lenkrad sowie die Räder geradestehen und das Raster am Reifen und Bordstein anliegt, wie eben beschrieben. In dieser Aufnahme wird an die Außenseite der Reifenaufstandsfläche herangezoomt und der am höchsten gelegene Bildpunkt auf dem Raster gesucht, der neben dem Reifen zu sehen ist. Von hier aus werden die Koordinaten des Punktes, welcher 5 Bildpunkte abwärts liegt für den Startpunkt der Zählzeile aufgenommen. Diese Verschiebung soll leichte Einlenkung des Rades bei wechselnde Beladung des Fahrzeugs oder Luftdruck kompensieren und damit verhindern, dass der Reifen als schwarzer Streifen des Rasters detektiert wird.

Ergebnisse manueller Zählung der Rasterstreifen zwischen dem Startpunkt bis zum Ende des Streifens mit Addition des Endstreifens werden als Referenz hinzugezogen. Die Abstände zum Bordstein werden zusätzlich manuell mit einem Maßband aufgenommen und mit den Ergebnissen verglichen. Außerdem wird manuell die Bild-Position des Bordsteinfußes eingegeben und mit dem bekannten Öffnungswinkel und Auflösung der Aufnahme der Abstand zum Startpunkt geometrisch bestimmt.

Die folgende Tabelle zeigt eine Auflistung der Messergebnisse für die unterschiedlichen Messmethoden. Die neue Messmethode soll bei der Versuchsdurchführung zeitlich effizienter sein, bzw. den Personalaufwand verringern, indem am Ende der Einparkvorgänge keine Raster mehr anzulegen sind. Deshalb soll festgestellt werden, ob die Methode der geometrischen Transformation im Vergleich zu den anderen Messmethoden reproduzierbare Ergebnisse liefert.

Messung	manuelle Rasterauswertung [mm]	automatische Rasterauswertung [mm]	manuelle Messung [mm]	geometrische Transformation der Bildposition [mm]	Differenz zu geometrischer Transformation [mm]
1 (Front)	73	73	-	68,5	4,5
2 (Front)	27	27	-	24,4	7,4
3 (Front)	71	71	70	68	2...3
1 (Heck)	147	149	-	150	1...3
2 (Heck)	153	155	-	153,6	0,6...1,4
3 (Heck)	123	-	115	120	3...5

Tabelle 23: Vergleich geometrischer Transformation von Bildkoordinaten mit anderen Messmethoden

Die Auflösung der Messmethoden Rasterauswertung ergibt sich aus der Breite der 2 mm breiten Streifen, während bei der geometrischen Transformation die Bildpunkte des Kamerasensors den beschränkenden Faktor darstellen. Die Auflösung wird berechnet, indem zwei nebeneinander liegende Bildpunkte im Bildbereich des Schnittpunktes von Bordsteinfuß und Bildhorizontalen transformiert werden. Die Ergebnisse sind der folgenden Tabelle zu entnehmen.

Bereich	Auflösung Rasterauswertung [mm]	Auflösung geom. Transformation [mm]
Vorderrad-Bordstein	2	0,85
Hinterrad-Bordstein	2	0,59

Tabelle 24: Auflösungen der unterschiedlichen Messprinzipien zur Bestimmung des Abstandes zwischen Rad und Bordstein

D.h. die anderen Messmethoden weisen, im Vergleich zur geometrischen Transformation, Abweichungen von maximal 7,4 mm auf. Im Vergleich zu der geforderten Genauigkeit von 10 mm liegen die Abweichungen niedriger. Außerdem wird mit der Transformation von Bildkoordinaten die geforderte Auflösung von 1 mm für laterale Abstandsmessungen erreicht. Für die Bestimmung der Differenz zwischen dem Abstand am Vorderrad und Hinterrad wird damit eine Auflösung von 1,44 mm erreicht.

Aus den genannten Gründen wird eine neue Messmethode für die Messung der lateralen Abstände entwickelt. Die Beschreibung dieser Methode folgt im nächsten Abschnitt.

Zweiter Entwurf

An Stelle des Messrasters wird der Bordstein im neuen Entwurf mit einer weißen Folie markiert, welche haftend an diesem angebracht wird (siehe folgende Abbildung). Bei der Auswertung wird der äußerste Punkt des Bordsteins durch den Kontrast zwischen Fahrbahn und Bordsteinmarkierung detektiert und so kann der Abstand mit der Anzahl an Bildpunkten zwischen Startpunkt und Bordstein berechnet werden. Dazu wird die Lage und Ausrichtung der Kamera vermessen. Vor allem ist dabei der Abstand zwischen der Linse der Kamera und dem Radaufstandspunkt in der Seitenansicht, wie in Abbildung 38 zu sehen ist wichtig.



Abbildung 41: Seiten-Kamera-Aufnahme mit markiertem Bordstein (2. Entwurf)

Für die Vermessung des Abstandes zwischen Bordstein und der Rad-Außenflanke des vorderen und hinteren Rades wird in der zweiten Messreihe ein neues Programm verwendet. Dieses sucht den Bordstein von fixen Punkt an der Außenflanke am Reifen-Aufstandspunkt aus gehend. Dabei wird in der Zählzeile in Richtung Bordstein, d.h. zur Fahrtrichtung nach rechts gesucht. Als Indikator dient dabei der Intensitätswert des betrachteten Bildpunktes. Um dafür den Parameter für den Schwellwert entsprechend der Belichtung dieser Aufnahme zu wählen, werden die Intensitätswerte der Zählzeile, welche von RGB in Graustufen umgewandelt wird geplottet.

Die Auswertung läuft in 2 Stufen ab: Zunächst wird nach dem ersten Auftreten des höchsten Intensitätswertes in der Zählzeile gesucht. Sobald dieser gefunden ist, wird von dem Punkt aus nach dem äußersten Punkt des Bordsteins gesucht. Dazu wird nach dem ersten Unterschreiten eines parametrisch festgelegten Schwellwertes gesucht. Der so ermittelte Punkt wird in den Plot als vertikale Linie eingefügt und es wird geprüft, ob dies dem äußersten Punkt des Bordsteins entspricht. Der Punkt soll dabei am Anfang des steilen Anstiegs der Intensitätswerte vor dem Bereich konstanter Maximalwerte liegt. Falls dies nicht der Fall ist, werden die Schwellwerte an die Intensitätswerte auf der Markierung und auf dem Asphalt angepasst und es wird erneut gesucht. Die bestimmten Schwellwerte werden für die nächsten Aufnahmen beibehalten und gegebenenfalls an stärkere oder schwächere Belichtung angepasst. Dies ist nötig, da die Bordsteinmarkierung bei abnehmendem Tageslicht, Intensitätswerte des Asphaltes erreicht. Umgekehrt erreicht der Bordstein bei zunehmendem Tageslicht Intensitätswerte der Bordsteinmarkierung.

Um den Abstand zwischen Bordstein und Radaufstandspunkt zu bestimmen, ist die Geometrie des Bordsteins sowie des Blickwinkels der Kamera zu Beachten. Folgende Abbildung zeigt die Lage und Orientierungen der Seitenkameras der zweiten Messreihe. Für die Berechnung des Abstandes zwischen Bordstein und dem Reifen ist hier vor allem der Abstand zwischen Kameralinse und Radaufstandslinie wichtig, damit auf der Höhe der Radaufstandsfläche mit

der Auflösung und dem horizontalen Öffnungswinkel der Kamera Abstände berechnet werden können.

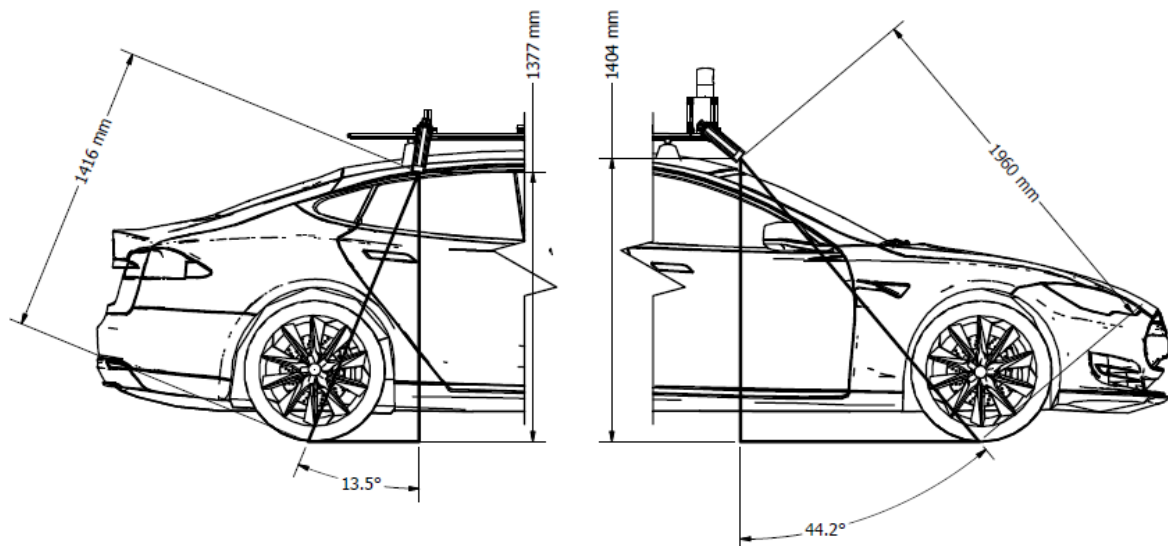


Abbildung 42: Orientierung und Lage der Seiten-Kameras am Versuchsfahrzeug

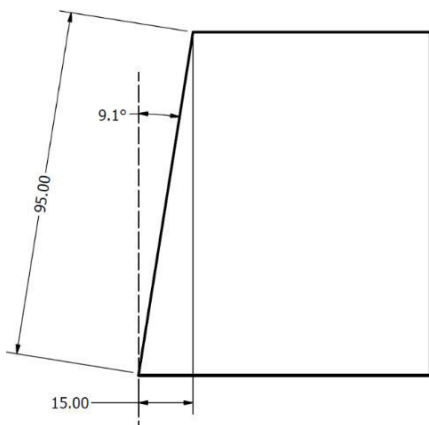


Abbildung 43: Bordstein-Querschnitt

Der Bordstein ist zwischen seinem Fußpunkt und der Oberkante schräg ausgeführt, wie im Querschnitt des Bordsteins in Abbildung 43 dargestellt. Nachdem die Länge der Schräge sowie der Gegenkathete mit einem Maßband gemessen wird, kann so der Winkel der Schrägen ermittelt werden.

Für den Fall, dass eine Kamera die Bordstein-Oberkantenecke unter einem größeren Blickwinkel als dem Bordsteinwinkel sieht, ist der Fußpunkt im Bild nicht zu sehen. Bei kleinerem Blickwinkel ist der Fußpunkt theoretisch unter den geometrischen Bedingungen zu sehen. Deshalb müssen die geometrischen Sichtverhältnisse bei den bekannten Anstellwinkeln der Kameras gesondert betrachtet werden. Abbildung 44 dient zur Veranschaulichung des Zusammenhangs. Dabei werden die Neigungswinkel der Kameras durch Messung der Höhe sowie des Abstandes der Kameragehäusefront zur Aufstandslinie ermittelt. Es folgt eine grafische Darstellung der Aufstandslinien.

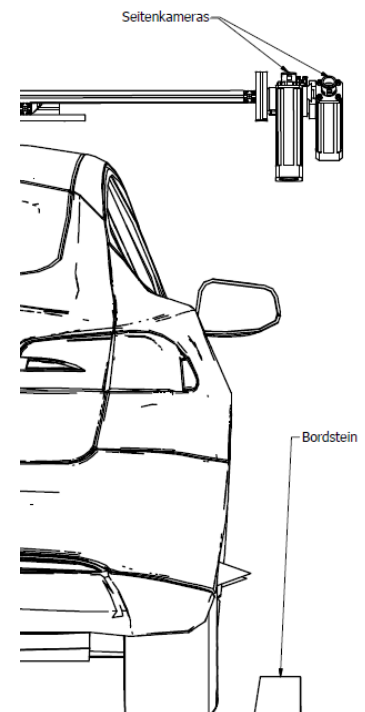


Abbildung 44: Relativposition Seitenkameras zu Bordstein

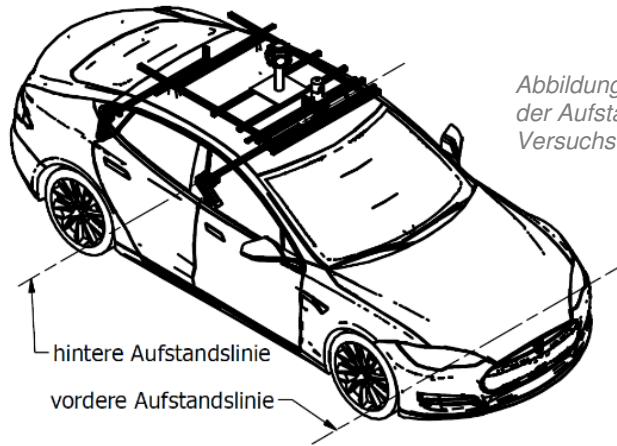


Abbildung 45: Darstellung der Aufstandslinien am Versuchsfahrzeugs

Zur Ermittlung der Lagemaße der Seitenkameras wird der Laser-Entfernungsmesser eingesetzt. Für die Frontkamera ergibt sich bei 1404 mm Höhe und einem Abstand zur Aufstandslinie von 1960 mm ein Winkel von $44,2^\circ$ (siehe Abbildung 42). Für die Heckkamera ergibt sich bei 1377 mm Höhe und einem Abstand zur Aufstandslinie von 1416 mm ein Winkel von $13,5^\circ$ bezüglich der Senkrechten zur Straße. Es folgt eine Darstellung, welche die Länge zwischen Bordsteinoberkante und -Fußpunkt unter dem Blickwinkel der Kameras abbildet.

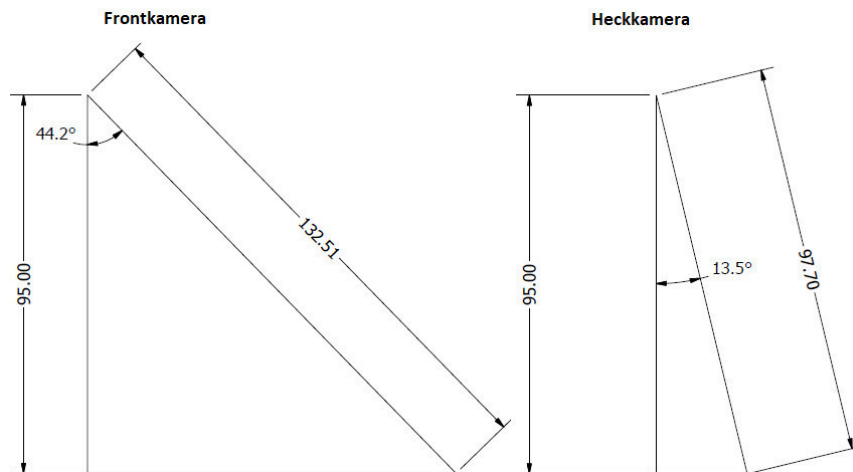


Abbildung 46: Länge zwischen Bordsteinoberkante und -fuß bei Anstellwinkel der Kameras

Abbildung 47 zeigt Darstellungen der Bordsteingeometrie beim Schneiden des Bordsteins unter dem Kamerablickwinkel zur Querschnittsebene, welche die Geometrie des Bordsteins aus Sicht der Kameras verdeutlichen sollen.

Es ergeben sich damit $6,5^\circ$ für die Bordsteinschräge unter dem Neigungswinkel der Frontkamera- und $8,7^\circ$ unter dem Neigungswinkel der Heckkamera. Diese Werte werden verwendet, um mit dem ermittelten Winkel der detektierten Bordsteinkante zu bestimmen, ob es sich bei dem Punkt um den Fußpunkt oder den Oberkanteneckpunkt handelt. Handelt es sich um den Oberkanteneckpunkt, so wird die Länge von 15 mm (siehe Abbildung 47, Abbildung 43) zwischen Fuß- und Oberkante hinzuaddiert, nachdem mit dem vermessenen Abstand zwischen Kameragehäusefront und Bordstein sowie der Bildposition des Oberkanteneckpunktes die Länge zwischen

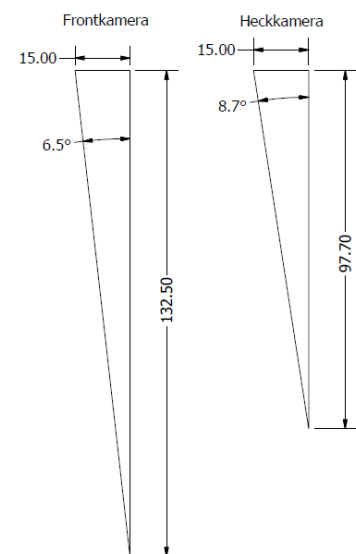


Abbildung 47: Bordsteinschrägen-Geometrie in Schnittebene unter Blickwinkel der Kamera

projizierter Kamerabildvertikalen und dem gefundenen Punkt berechnet wird. Die Summe der beiden Längen ergibt mit der Position des Radaufstandspunktes den gesuchten Abstand zwischen Bordstein und Rad.

Für den Fall, dass der Fußpunkt detektiert wird, ergibt sich mit dem Abstand zwischen Reifenaufstandslinie und Kamera und den Bildpositionen von Aufstandspunkt und Bordsteinoberkante der Abstand zwischen Rad und Bordstein. Es folgt ein Beispiel-Plot für den Bordstein am Heck bei Messung 2. Die vertikale Punkte-Linie zeigt den detektierten Punkt, an dem die Bordsteinoberkante oder der Fußpunkt vermutet wird. In dem Fall wird für diesen Punkt ein Winkel von $2,9^\circ$ berechnet, was bedeutet, dass theoretisch der Fußpunkt im Bild zu sehen ist. Anhand des Plots ist zu sehen, dass vor dem Anstieg bis auf den maximal darstellbaren Wert von 255 zuerst ein lokales Maximum von unter 140 auftritt. Dieses kann durch die im Vergleich zum Straßenasphalt stärker reflektierende (mit der Markierungsfolie beklebten) Schräge des Bordsteins erklärt werden. Jedoch sind hier die Intensitätswerte noch geringer, als an der Oberkante, wo das Sonnenlicht sehr stark reflektiert wird.

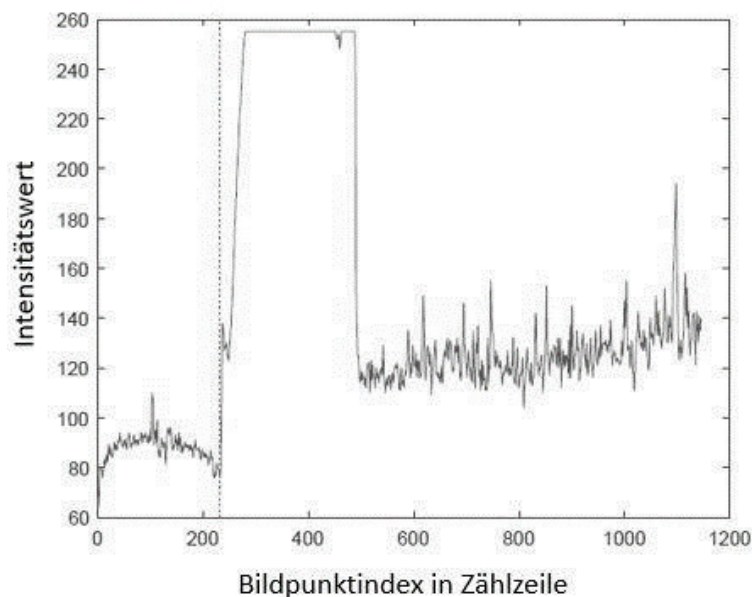


Abbildung 48: Plot Intensitätswerte ab Reifenaufstandspunkt bis Bildrand (Messung 2, Heck)

Als Gegenbeispiel für einen Fall, bei dem die Bordsteinschräge für die Kamera nicht sichtbar ist folgt eine Darstellung aus der Aufnahme der Frontkamera aus Messung 4. Hier wird ein Winkel von $8,72^\circ$ für die Bordsteinoberkante im Bild gemessen. Im Vergleich zu der Messung in Abbildung 48 sind hier kleinere Intensitätswerte zu sehen, da die Sonneneinstrahlung hier abgenommen hat. Dabei ist die relative Differenz zwischen dem ersten Anstieg auf ein en Wert von ca. 100 zum zweiten auf ca. 160 deutlich kleiner, als es bei der Messung in Abbildung 48 der Fall ist. Für beide Messungen konnte mit 170 als Überschreitungsschwellwert und 90 für den Unterschreitungsschwellwert die Bordsteinkante detektiert werden.

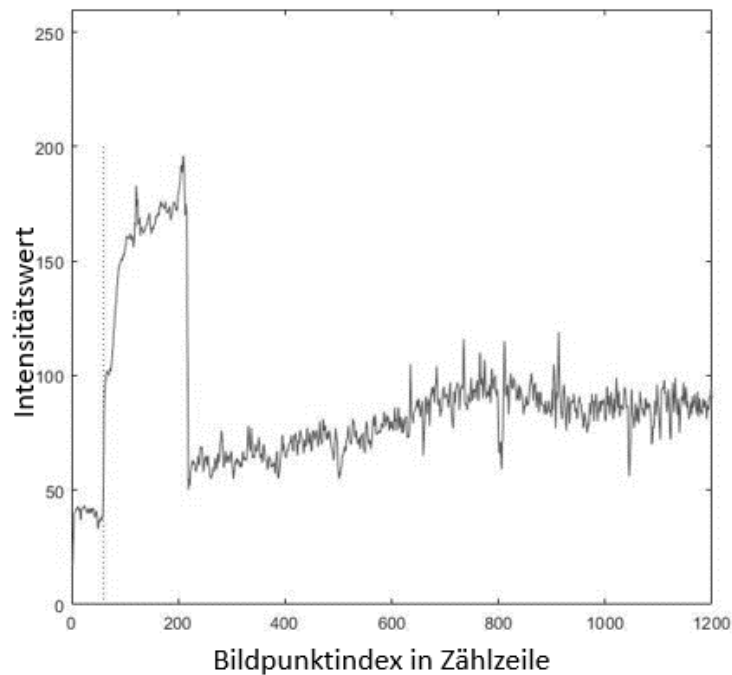


Abbildung 49: Plot Intensitätswerte ab Reifenaufstandspunkt bis Bildrand (Messung 4, Front)

Manuelle Eingabe von Koordinaten für die Position des Bordsteins am rechten Bildrand ergeben Grenzwerte für die Messbereiche von 1088 mm am Vorderrad und 737 mm am Hinterrad.

5.7.9. Bestimmung des Orientierungswinkels am Ende des Einparkvorgangs

Aus den ermittelten Abständen des Vorder- und Hinterrades zum Bordstein wird mit den bekannten Werten für die Abstände der Radaußenflanken der Orientierungswinkel des Fahrzeugs mit dem Tangens bestimmt. Dieser wird dann wie folgt berechnet.

$$\theta_s = \arctan\left(\frac{d_f - d_h + \frac{B_f}{2} - \frac{B_h}{2}}{R_s}\right) \quad (7)$$

θ_s	:	Orientierungswinkel aus Seitenkamera-Messung
d_f	:	Abstand zu Bordstein am Vorderrad
d_h	:	Abstand zu Bordstein am Hinterrad
B_f	:	Frontspurbreite
B_h	:	Heckspurbreite
R_s	:	Radstand

Die Auflösung der verwendeten Messmethode für die Differenz zwischen Vorder- und Hinterrad erreicht nach Abschnitt 0 einen Wert von $\leq 1,44$ mm. Bei kleinsten Differenzen der Abstände hat diese Größe den größten Einfluss auf die Auflösung. Deshalb ergibt sich die minimale Auflösung wie folgt.

$$\Delta\theta_{s,min} = \arctan\left(\frac{1,44 \text{ mm} + \frac{1906 \text{ mm}}{2} - \frac{1945 \text{ mm}}{2}}{2960 \text{ mm}}\right) = 0,35^\circ \quad (8)$$

5.7.10. Messung der Abstände zu Fahrzeugen

Bei den Berechnungen des letzten Abschnittes ist die Position des Bordsteins in der Horizontalen der Seitenkamera-Aufnahme ermittelt worden. Die y-Koordinaten der Markierungsfolie am Bordstein sind mit 273 mm (vorne) und -937 mm (hinten) bekannt. Die Methode zum Finden des Markierungsrandes in der Bild-Horizontalen, wird analog für das Finden des Markierungsrandes entlang der Bildvertikalen umgesetzt. In folgender Abbildung ist die Lage der Markierung in der Aufnahme zu sehen.



Abbildung 50: Aufnahme der Front-Seitenkamera

Auch hier wird wieder der Kontrast zwischen dem Untergrund und der Markierung genutzt, um die Markierung zu finden. Danach wird über die Bildpunkte zwischen optischem Zentrum der relative Winkel zur optischen Achse berechnet. Es folgt der dazu entsprechende Ausschnitt des Programms.

```
% in der Variablen n wird die Anzahl der Bildpunkte der Spalte im
% entzerrten Bild aufgenommen.
n = sum(Front(:,MarkerEnde(i,2))>0);
% Der relative Winkel zur optischen Achse wird berechnet.
MarkerEnde(i,3) = (cameraParamsFront.ImageSize(1)...
- MarkerEnde(i,1) - cameraParamsFront.PrincipalPoint(2))...
* cam_Winkel_V / n;
```

Als Neigungswinkel der optischen Achse im Raum wird der eingestellte Winkel (43°) des Kameragehäuses am Profil eingesetzt. Nach den in Abbildung 42 gezeigten Maßen für die Höhe und den direkten Abstand der Kamera zur entsprechenden Aufstandslinie ist auch die Lage der Kamera entlang der Fahrzeuglängsachse bekannt.

Mit der y-Koordinate des vorne geparkten Fahrzeugs und der gemessenen Parklückenlänge wird der Abstand vor und hinter dem eingeparkten Versuchsfahrzeug berechnet. Nachdem der Frontabstand berechnet ist, wird mit der Fahrzeuglänge und dem berechneten Orientierungswinkel aus den Bordsteinabstands-Messungen des Abstands am Heck berechnet. Die Auflösung dieser Messmethode erreicht 1 mm im Bereich des unteren Endes der Markierung im Bild und 2,2 mm im obersten Bereich des Bildes.

Der Messbereich wird durch manuelles Einsetzen der Bildkoordinaten für das vordere und hintere Ende der Markierung berechnet. Dabei ergeben sich für den Frontabstand Werte von -228 mm bis 2092 mm. Dementsprechend können für den Frontabstand Werte im Bereich 0...2092 mm gemessen werden. Beim maximalen Frontabstand und einem Heckabstand von 0mm wäre die Parklücke 7071 mm lang. Der untere Grenzwert von -228 mm ist so zu interpretieren, dass die Markierung beim Frontabstand von 0mm auch noch sichtbar sein muss. In diesem Fall muss der Heckabstand 2092 mm betragen. Folglich beträgt der Messbereich für den Heckabstand ebenfalls 0 bis 2092 mm. Folglich können mit Parklücken von bis zu 7071 mm noch Abstände in dem Wertebereich bestimmt werden. Die möglichen Werte für die Bestimmung der Abstandsdifferenz liegen ebenfalls in dem genannten Bereich.

5.8. Anforderungen an Testverfahren

Es folgt eine tabellarische Übersicht zu den in Abschnitt 3.1 erwähnten Anforderungen an Testverfahren und den Maßnahmen zu deren Erfüllung.

Repräsentativität	<ul style="list-style-type: none"> • Für die Bildung der Parklücke werden Fahrzeuge verwendet. • Die Parklücke ist durch einen Bordstein mit Gehweg begrenzt. Die Versuche finden auf einer Versuchsstrecke statt, wo kein normaler Verkehr herrscht.
Validität	<ul style="list-style-type: none"> • Über dem Fahrzeug applizierte Messtechnik verdeckt die Sicht der Fahrzeugsensoren auf den Untergrund und die Fahrzeuge nicht. Dieses gilt auch für die Markierungen am Versuchsfahrzeug und auf der Versuchsstrecke.
Variierbarkeit	<ul style="list-style-type: none"> • Testszenarien zum Längseinparken bei unterschiedlichen Rangierräumen sind mit dem entwickelten Konzept realisierbar
Beobachtbarkeit	<ul style="list-style-type: none"> • Das parkende Versuchsfahrzeug ist während des Vorgangs auf der Versuchsstrecke sichtbar. Gleiches gilt für das Lenkrad und die Räder des Versuchsfahrzeugs • Für die Messungen ist der Bereich zwischen Versuchsfahrzeug und Bordstein sowie Versuchsfahrzeug und den anderen geparkten Fahrzeugen sichtbar und frei von Objekten
Ökonomisch	<ul style="list-style-type: none"> • Für die Messungen wird bereits im Urban Mobility Lab vorhandene Hardware verwendet (siehe 5.2, 5.3). • Ein durch Rapid-Prototyping gefertigter Klemmschutz verhindert das Einklemmen von Kabeln durch ein Türfenster (siehe Abbildung 29)
Reproduzierbarkeit	<ul style="list-style-type: none"> • Die Positionen aller Markierungen auf der Versuchsstrecke relativ zueinander ist dokumentiert und kann nachgestellt werden • Ebenso ist die Position des vor der Parklücke stehenden Fahrzeugs dokumentiert. Bei der Variierung der Parklückengröße wird das hinten geparkte Fahrzeug entlang des Bordsteins bewegt • Die Markierungen am Versuchsfahrzeug sind über Vorder- und Hinterachse mittig bezüglich der Fahrzeugbreite appliziert • Die Lage des globalen Koordinatensystems ist auf Höhe auf die Orts-Markierung Nr. 1 vorgesehen. Der Verlauf der Achsen sowie Details zu Ausführung der Markierung sind den Kapiteln 5.5 und 5.7.3 zu entnehmen. Weitere Details sind in den entsprechenden Abschnitten zur Realisierung zu finde
Sicherheit	<ul style="list-style-type: none"> • Während der Einparkvorgänge befindet sich das Versuchspersonal im Fahrzeug oder auf dem Gehweg • Beim Betreten der Fahrbahn oder Herantreten an das Versuchsfahrzeug wird darauf geachtet, dass das Versuchsfahrzeug stillsteht und bei heruntergelassenen Fahrzeugfenstern untereinander kommuniziert • Für unbeteiligte wird die Versuchsstrecke mit Pylonen und anderen Komponenten abgesperrt und das Versuchspersonal warnt herannahende Personen

Tabelle 25: Maßnahmen zur Erfüllung von Anforderungen an Testverfahren

5.9. Auswertung der Messergebnisse

In diesem Abschnitt werden die aufgezeichneten Messdaten statistisch ausgewertet. Dabei sollen Häufigkeitsverteilungen und andere Diagrammtypen zum Aufzeigen von Zusammenhängen zwischen Parklückenlänge und Größen, welche die Parkendposition quantifizieren, untersucht werden. Außerdem werden Raumkoordinaten und Orientierungswinkel von Bewegungsverläufen des parkenden Versuchsfahrzeugs untersucht. Weitere Darstellungen werden herangezogen, um Korrelationen unterschiedlicher Messungen für gleiche Größen zu untersuchen.

5.9.1. Protokoll der ersten Messreihe

Die erste Messreihe findet am 9.4.2018 statt. Die folgende Tabelle zeigt eine Übersicht über die durchgeführten Messungen sowie die dabei aufgezeichneten Daten. Dabei werden insgesamt 7 Messungen bei zwei Parklückenlängen durchgeführt. Das „x“-Symbol steht für eine vorhandene Aufzeichnung des jeweiligen Sensors bei der jeweiligen Messung. Die Aufzeichnung über die sphärische Kamera und den LIDAR-Sensor laufen bis nach Messung 6 die ganze Zeit über. Bilder der Seitenkameras von dem Bereich an der Seite des Fahrzeugs liegen für die Messungen 1 bis 3 vor. Aufgrund technischer Probleme liegen hier für die restlichen Messungen keine Aufzeichnungen der Seitenkameras vor. Diese bestehen darin, dass die Stromversorgung des Laptops für den Zugriff auf den Rechner nicht gewährleistet werden kann. Nachdem der Akku dieses Laptops leer ist kann der Rechner deshalb nicht mehr gesteuert werden und es laufen nur noch die Messungen mit dem LIDAR-Sensor und der sphärischen Kamera weiter. Für Messung 7 konnten auch diese nicht mehr gestartet werden. Mit Vogelkamera, welche durch einen anderen Laptop gesteuert wird, werden alle Messungen bis auf die 5. aufgezeichnet. Nachdem nach der dritten Messung bei 6 Anläufen keine Parklücke detektiert wird, wird die neue Stufe von 6940mm eingestellt und damit 4 weitere Messungen durchgeführt.

Messung Nr.	Rangier- raum [mm]	Aufzeichnungen						
		SP	SE	VK	L	Manuell gemessener Endabstand Vorderrad- Bordstein [mm]	Manuell gemessener Endabstand Hinterrad- Bordstein [mm]	Endabstand zu BMW (Nummernschild zu Nummernschild) [mm]
1	1121	x	x	x	x	70	115	558
2	1121	x	x	x	x	-	-	-
3	1121	x	x	x	x	-	-	570
4	1961	x	-	x	x	70	10	-
5	1961	x	-	-	x	90	195	-
6	1961	x	-	x	x	40	173	999
7	1961	-	-	x	-	170	214	963

x = vorhanden - = nicht vorhanden
 SP=sphärische Kamera; SE=Seitenkamera; VK= Vogelkamera; L=LIDAR-Sensor

Tabelle 26: Protokoll der 1. Messreihe

5.9.2. Protokoll der zweiten Messreihe

Die zweite Messreihe findet am 13.06.2018 statt und es werden 46 Einparkversuche durchgeführt. Als erste Stufe wird ein Rangierraum von 1115 mm eingestellt. Nach 6 Versuchen ohne Erkennung der Parklücke durch das System, wird der Rangierraum auf 1246mm vergrößert und es werden 15 Versuche in dieser Stufe durchgeführt. Danach wird der Rangierraum erneut verkleinert auf 1139 mm. Nach 4 erfolglosen Anläufen wird als nächstes die Stufe 1328mm eingestellt und mit dieser Stufe 15 Versuche durchgeführt. Danach werden mehrere weitere Stufen im Abstand von ca. 100 mm eingestellt und es werden pro Stufe jeweils 2 Versuche durchgeführt. Mit der letzten Stufe sind 2 weitere Versuche durchgeführt worden, nachdem die Aufzeichnung mit der Vogelkamera unterbrochen wird. Somit werden 2 Stufen mit 15 Versuchswiederholungen und insgesamt 10 Stufen bei mindestens einem Versuch, bzw. 9 Stufen bei mindestens 2 Versuchen untersucht. Zusätzlich werden dieses Mal auch auf dem Display des Versuchsfahrzeugs angezeigte Abstände zu erkannten Objekten vor und hinter dem Fahrzeug als Referenz für eigene Messungen notiert. Auch bei dieser Messreihe wird der Abstand zwischen Reifenaufstandspunkt und Bordsteinfußpunkt mit einem Maßband gemessen. Dieses erfolgt nicht bei jeder Messung, sondern stichprobenartig, um die Messmethoden, aber nicht jedes einzelne Messergebnis zu prüfen. Der LIDAR-Sensor ist vor diesen Messungen aufgrund eines technischen Defekts ausgefallen und ist deshalb nicht aufgelistet. Die sphärische Kamera läuft die ganze Zeit über und wird nur nach ca. 7-10 GB Stream-Daten Volumen neu gestartet, um so Versuche in kürzeren Abständen durchführen zu können.

Nr.	Rangier-Raum [mm]	manuelle Messung Endabstand (Front) [mm]	manuelle Messung Endabstand (Heck) [mm]	manuelle Messung Endabstand Vorderrad-Bordstein [mm]	manuelle Messung Endabstand Hinterrad-Bordstein [mm]	SP	SE	VK
1	1246	594	-	-	-	-	-	-
2	1246	629	649	121	146	-	x	-
3	1246	604	655	-	-	x	x	x
4	1246	620	-	-	-	x	x	x
5	1246	608	-	185	185	x	x	x
6	1246	-	-	35	183	x	x	x
7	1246	608	-	-	-	x	x	x
8	1246	596	-	-	-	x	x	x
9	1246	610	-	-	-	x	x	x
10	1246	605	-	-	-	x	x	-
11	1246	620	-	-	-	x	x	x
12	1246	603	-	-	-	x	x	x
13	1246	601	-	-	-	x	x	x
14	1246	614	-	-	-	x	x	x
15	1139	574	-	80	165	x	x	x
16	1328	644	-	-	-	x	x	x
17	1328	656	-	-	-	x	x	x
18	1328	650	-	90	175	x	x	x
19	1328	628	-	-	-	x	x	x
20	1328	635	-	-	-	x	x	x
21	1328	643	720	-	-	x	x	x
22	1328	631	725	-	-	x	x	x
23	1328	628	729	-	-	x	x	x
24	1328	629	735	-	-	x	x	x
25	1328	633	726	-	-	x	x	x
26	1328	636	720	-	-	x	x	x
27	1328	636	714	-	-	x	x	x
28	1328	646	718	-	-	x	x	x
29	1328	634	718	-	-	x	x	x
30	1328	651	707	-	-	x	x	x
31	1461	664	804	-	-	x	x	x
32	1461	673	796	-	-	x	x	x
33	1584	726	887	-	-	x	x	x
34	1584	762	852	-	-	x	x	x
35	1718	880	877	-	-	x	x	x
36	1718	873	972	-	-	x	x	x

Tabelle 27: Protokoll der 2. Messreihe (Teil 1)

Nr.	Rangier-Raum [mm]	manuelle Messung Endabstand (Front) [mm]	manuelle Messung Endabstand (Heck) [mm]	manuelle Messung Endabstand Vorderrad-Bordstein [mm]	manuelle Messung Endabstand Hinterrad-Bordstein [mm]	SP	SE	VK
37	1843	974	884	-	-	x	x	x
38	1843	961	920	-	-	x	x	x
39	1936	992	952	-	-	x	x	x
40	1936	944	1005	-	-	x	x	x
41	2041	887	1102	-	-	x	x	x
42	2041	1025	1032	-	-	x	x	x
43	2149	1032	1012	-	-	x	x	-
44	2149	1049	1117	-	-	x	x	-
45	2149	1045	1116	-	-	x	x	x
46	2149	1015	1149	-	-	x	x	x

Tabelle 28: Protokoll der 2. Messreihe (Teil 2)

5.9.3. Abstand Rad-Bordstein

Statistische Auswertung der 1. Messreihe

Für die ersten drei Messungen der 1. Messreihe werden Bilder über die Seitenkameras aufgenommen. In den folgenden Tabellen sind die Ergebnisse bei Anwendung der unterschiedlichen Messmethoden aufgelistet. Bei allen 3 Messungen ist ein Rangierraum von 1121 mm eingestellt.

	manuelle Rasterauswertung [mm]	autom. Rasterauswertung [mm]	geometrische Transformation [mm]	man. Messung [mm]
Messung 1	73	73	68,5	-
Messung 2	27	27	24,4	-
Messung 3	71	71	68	70

Tabelle 29: Messergebnisse für Abstand Vorderrad-Bordstein (1. Messreihe)

	manuelle Rasterauswertung [mm]	autom. Rasterauswertung [mm]	geometrische Transformation [mm]	man. Messung [mm]
Messung 1	147	149	150	-
Messung 2	153	155	153,6	-
Messung 3	123	15	120	115

Tabelle 30: Messergebnisse für Abstand Hinterrad-Bordstein (1. Messreihe)

Darüber hinaus erfolgen manuelle Messungen mit dem Maßband für die weiteren 4 Messungen. Wie in folgender Darstellung zu sehen ist, ist der Abstand am Vorderrad im Vergleich zum Abstand am Hinterrad 20 bis 76,9% kleiner.

	manuelle Messung Vorderrad-Bordstein [mm]	manuelle Messung Hinterrad-Bordstein [mm]	Differenz in Relation Abstand Hinterrad-Bordstein
Messung 3	70	130	46,2 %
Messung 4	70	100	30,0 %
Messung 5	90	195	53,9 %
Messung 6	40	173	76,9 %
Messung 7	170	214	20,6 %

Abbildung 51: manuelle Messungen des Abstandes Rad-Bordstein (1. Messreihe)

Statistische Auswertung der 2. Messreihe

Die Bestimmung der Intervallgrößen für Histogramme erfolgt nach (Scott 1979). Demnach ergibt sich diese mit der Standardabweichung s für n Werte nach der Formel:

$$\Delta = 3,49sn^{-1/3} \quad (9)$$

Es folgt eine Darstellung der Verteilung absoluter Häufigkeiten für gemessene Abstände zwischen Vorderreifen und Bordstein bei der ersten Stufe der Messreihe. Dabei sticht ein Intervall mit einer Häufigkeit von 7 heraus. Dieses liegt im Bereich 63,2 mm bis 126,5 mm. Für größere Werte treten kleinere Häufigkeiten auf als bei Werten, die kleiner sind als das am häufigsten auftretende Intervall. Bei einem arithmetischen Mittelwert von 92,8 mm beträgt die Standardabweichung hier 42,6 mm.

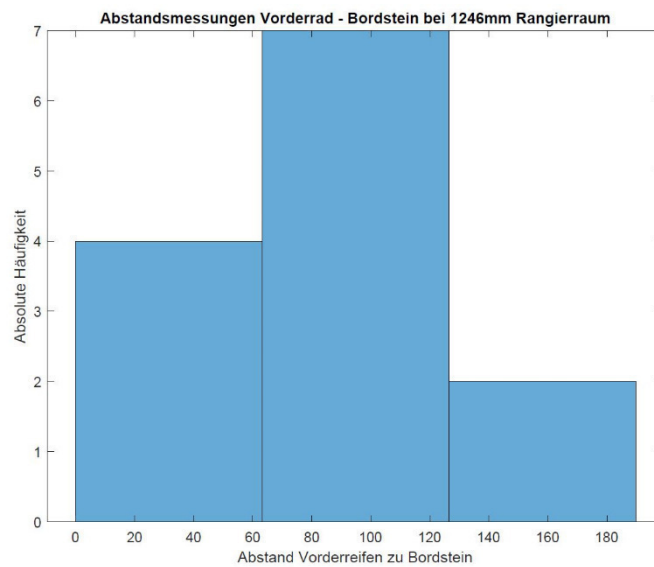


Abbildung 52: Histogramm mit 10 Intervallen zu Abstandsmessung Vorderrad-Bordstein bei 1246mm Rangierraum

Analog zu den Ergebnissen aus Messungen am Hinterrad, wird die Häufigkeitsverteilung für die Ergebnisse der Messungen am Hinterrad untersucht und ist in der folgenden Abbildung zu sehen. Hier beträgt der arithmetische Mittelwert 160,2 mm und die Standardabweichung liegt bei 17 mm.

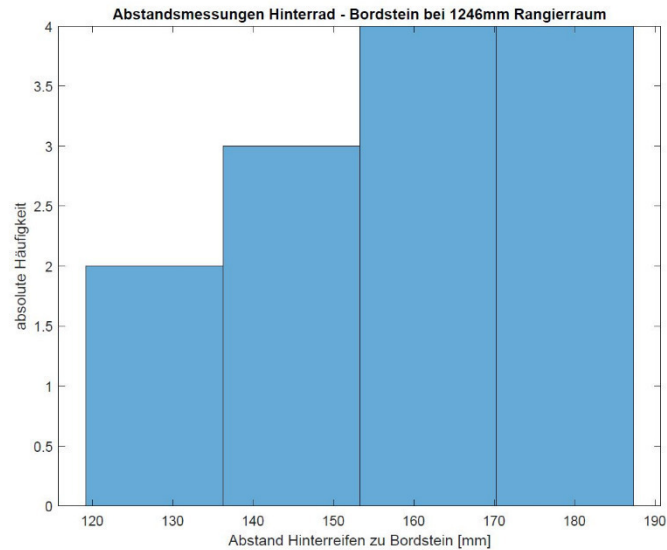


Abbildung 53: Histogramm zu Abstandsmessung Hinterrad-Bordstein bei 1246mm Rangierraum

Für die Stufe 1328 mm werden ebenfalls Histogramme für die Ergebnisse untersucht. Zunächst erfolgt dies für die Messungen am Vorderrad (siehe folgende Abbildung). Hier tritt ein Maximum mit einer Häufigkeit von 7 für das Intervall 82,7 mm bis 110 mm auf, welches vollständig in dem Intervall der vorherigen Stufe mit der Häufigkeit von 7 liegt. Der Mittelwert beträgt hier 109,5 mm bei einer Standardabweichung von 27,6 mm. Nachdem dieser Zusammenhang deutlich wird, soll als nächstes eine kumulierte Darstellung (Abbildung 55) betrachtet werden.

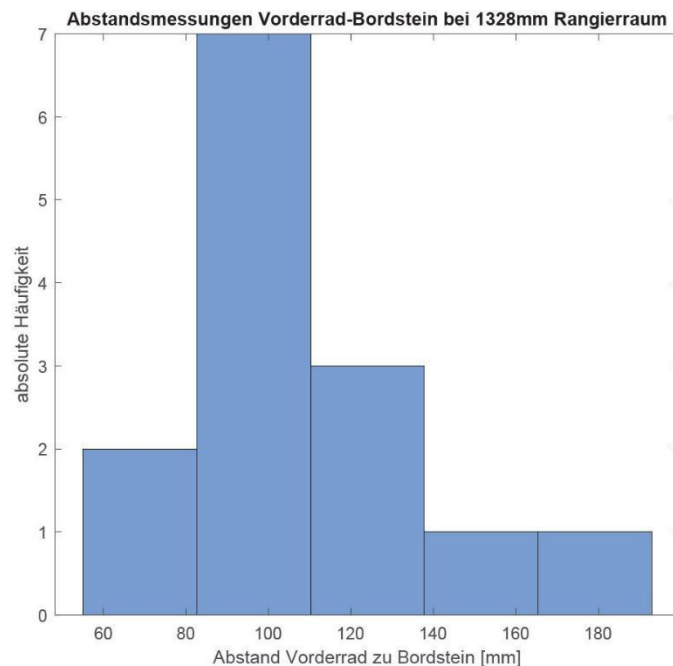


Abbildung 54: Histogramm zu Abstandsmessung Vorderrad-Bordstein bei 1328mm Rangierraum

Der arithmetische Mittelwert beträgt für diese Ergebnisse 101,5 mm und die Standardabweichung liegt bei 35,9 mm.

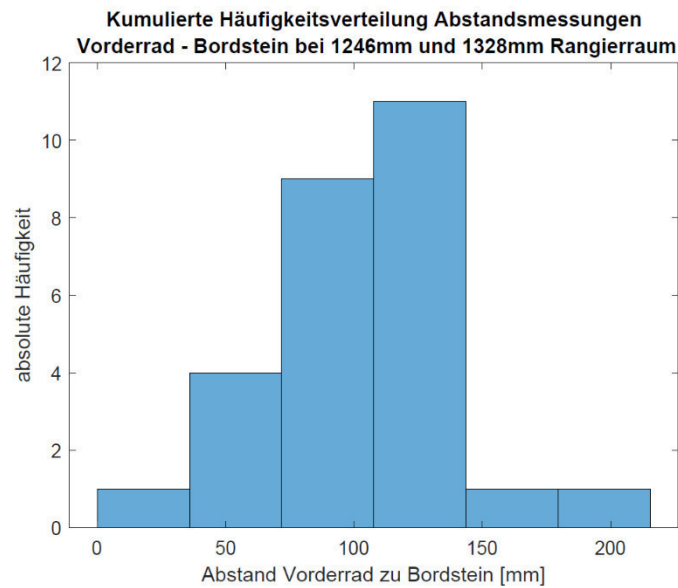


Abbildung 55: Kumulierte Häufigkeiten Für Messungen am Vorderrad bei 1246mm und 1328mm Rangierraum

Als nächstes werden die Messergebnisse für das Hinterrad bei der Stufe 1328 mm untersucht. Auch hier liegt der Mittelwert mit 164 mm deutlich höher, als es am Vorderrad bei dieser Stufe der Fall ist. Auch die Standardabweichung liegt mit 23,4 mm unter der von den Messergebnissen am Vorderrad.

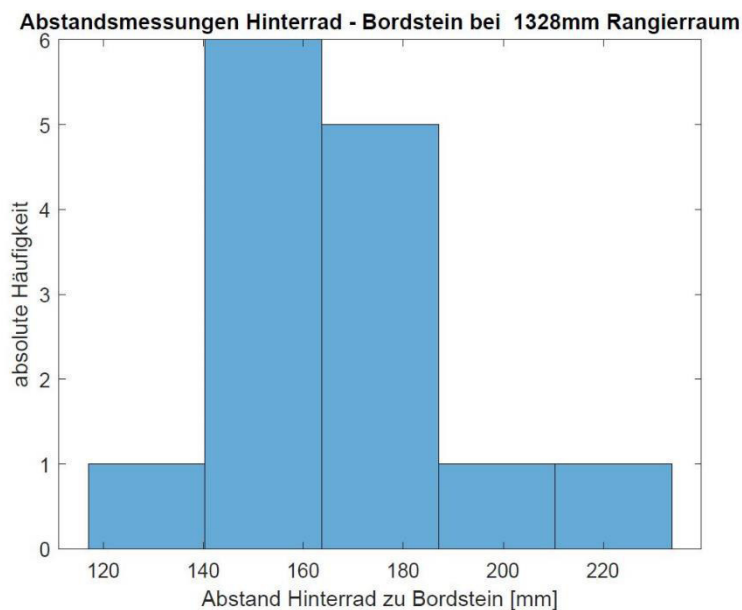


Abbildung 56: Histogramm zu Abstandsmessung Hinterrad-Bordstein bei 1328mm Rangierraum

Eine Kumulierung, für beide Stufen liefert eine Verteilung, die wie folgt aussieht. Der arithmetische Mittelwert beträgt für diese Ergebnisse 161,9 mm. Die Standardabweichung ist mit 20,3 mm deutlich kleiner, als beim Vorderrad mit 35,9 mm.

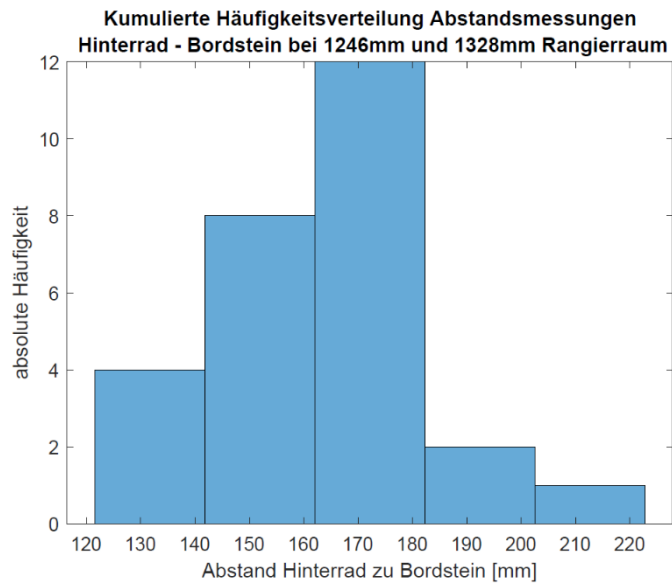


Abbildung 57: Kumulierte Häufigkeiten Für Messungen am Hinterrad bei 1246mm und 1328mm Rangierraum

Nach den Untersuchungen kann ausgesagt werden, dass Abstände zum Bordstein am Hinterrad bei den durchgeführten Messungen für die zwei Reihen mit 1246 mm und 1328 mm Rangierraum im Vergleich zu den Abständen am Vorderrad im Mittel um ca. 60% größer sind. Außerdem ist eine Tendenz zu einer Anhäufung von Endabständen im Bereich der berechneten Mittelwerte anhand der gesammelten Daten zu vermuten. Als nächstes sollen Ergebnisse aller Versuche bei allen Stufen untersucht werden (siehe folgende Abbildung). Der Mittelwert für die Messungen am Vorderrad verschiebt sich hier von 101,5 mm auf 122,9 mm.

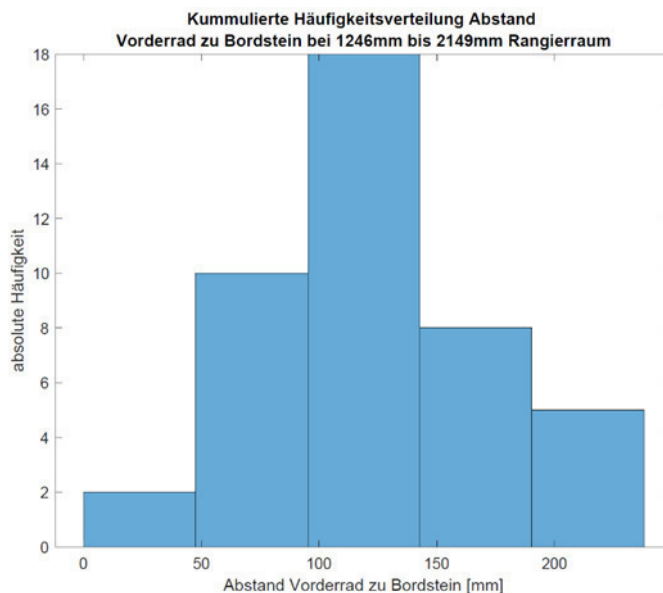


Abbildung 58: Kumulierte Häufigkeiten Für Messungen am Vorderrad bei 1246mm und 2149mm Rangierraum

Ähnliches kann bei Betrachtung der kumulierten Messungen am Hinterrad für alle durchgeführten Messungen festgestellt werden. Der arithmetische Mittelwert verschiebt sich auch hier von 161,9 mm zu einem größeren Wert von 174,3 mm. Dabei ist die Veränderung mit einem Zuwachs von ca. 7% geringer, als beim Vorderrad mit ca. 20%.

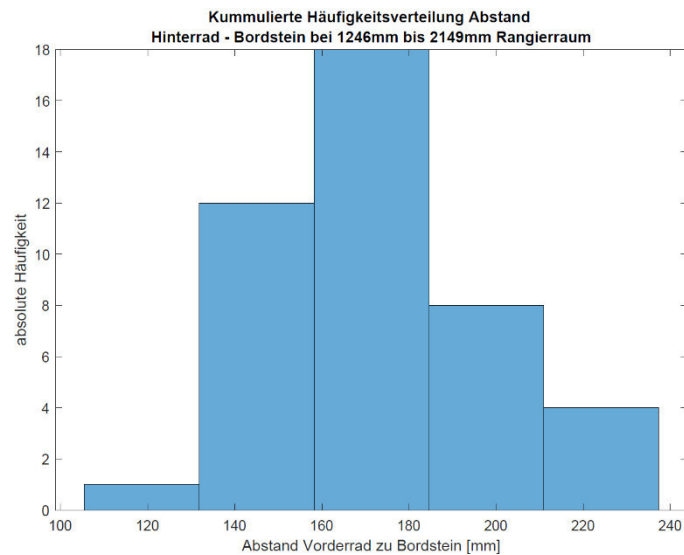


Abbildung 59: Kumulierte Häufigkeiten Für Messungen am Hinterrad bei 1246mm und 2149mm Rangierraum

Die Untersuchung von Histogrammen zeigen, dass Mittelwerte der Messergebnisse sich zu größeren Werten verschieben, wenn Ergebnisse aus Messungen bei größerem Rangierraum mit einbezogen werden. Deshalb soll der Zusammenhang zwischen Rangierraum und dem Endabstand zum Bordstein näher untersucht werden. Dazu sind in der folgenden Grafik Alle für die Stufen gemittelten (arithmetisch) Messergebnisse für beide Räder über die Stufen abgetragen.

Auch hier ist zu sehen, dass Abstände am Vorderrad kleiner sind, als am Hinterrad. Für die Messungen am Vorderrad ist bis zur Stufe 1843 mm ein Zuwachs auf 228 mm und danach eine Abnahme auf 166 mm bei der Stufe 2041 mm zu sehen. Für den Bereich 1461 mm bis zur letzten Stufe ist dieser Verlauf auch bei den Messungen am Hinterrad zu erkennen. Für den davor liegenden Bereich sinkt das Ergebnis von 166 mm auf 152 mm. Darüber hinaus ist in der Darstellung für größere Rangierräume eine Annäherung der beiden Verläufe zu sehen. Die Standardabweichungen für die Stufen 1246 mm und 1328 mm sind ebenfalls eingetragen.

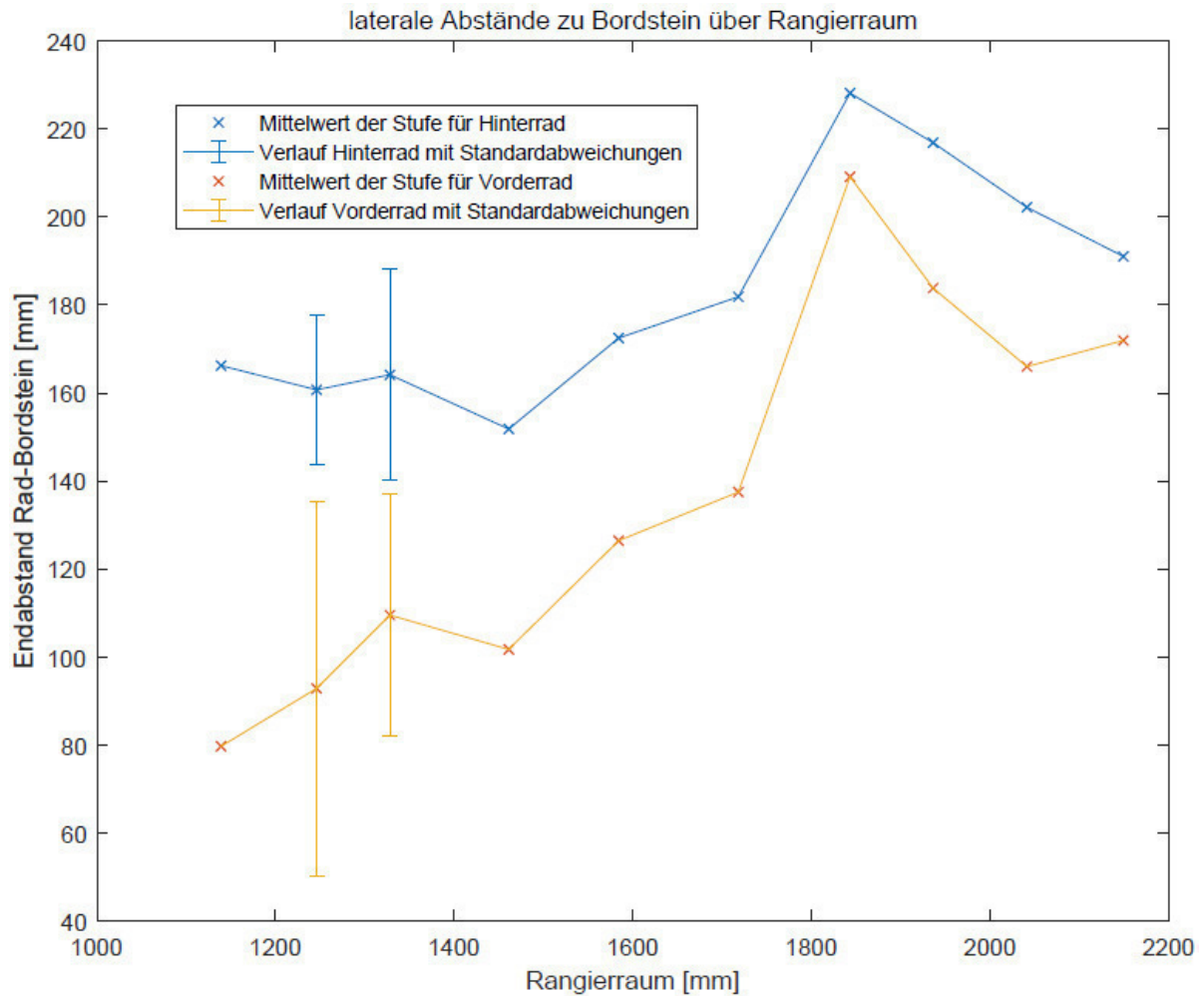


Abbildung 60: Verlauf Bordsteinabstand über Rangierraum

Differenz aus Abstand am Vorderrad und Abstand am Hinterrad

Die Differenz aus beiden Abständen ist in Abbildung 61 über dem Rangierraum aufgetragen. Dabei wird für jede Stufe nach der Subtraktion der Vorderrad-Abstände von den Hinterrad-Abständen der arithmetische Mittelwert berechnet. Dieser Wert ist für jede Stufe in der Grafik als Kreuz eingetragen. Für die Stufen mit mindestens 4 Ergebnissen ist außerdem die Standardabweichung als vertikale Klammer eingetragen. Hier ist zu sehen, dass die Differenz bei größeren Rangierräumen kleiner wird. Dabei ist der Verlauf nicht stetig und steigt nach der Stufe 1843 mm wieder an, bevor er wieder sinkt.

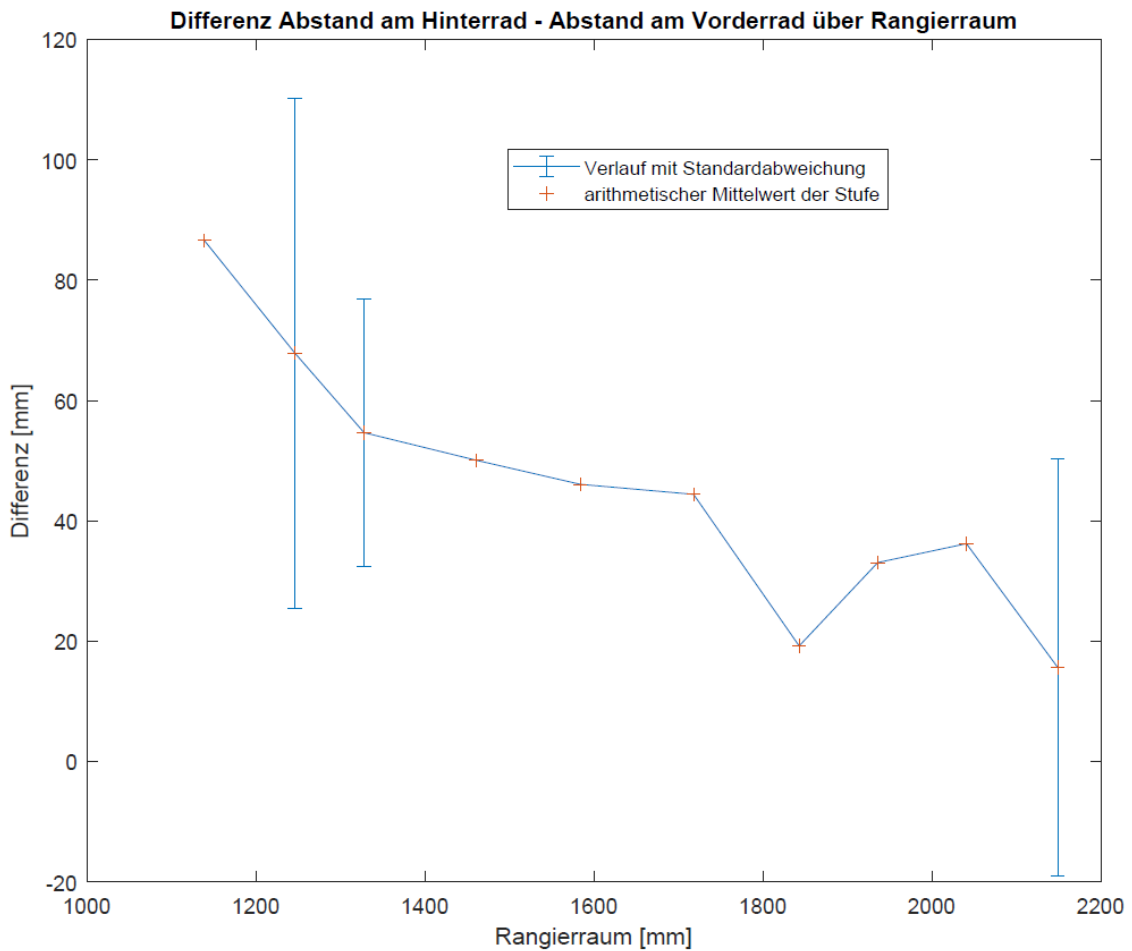


Abbildung 61: Differenz aus Hinterradabstand - Vorderradabstand über Rangierraum

Die Häufigkeitsverteilung der berechneten Differenzen sind in der folgenden Darstellung zu sehen. Dabei liegt der Mittelwert bei 51 mm. Das bedeutet, der Hinterrad-Abstand zum Bordstein ist im Mittel 51 mm größer, als der Abstand des Vorderrades zum Bordstein. Dabei liegt die Standardabweichung aller Werte bei 34 mm.

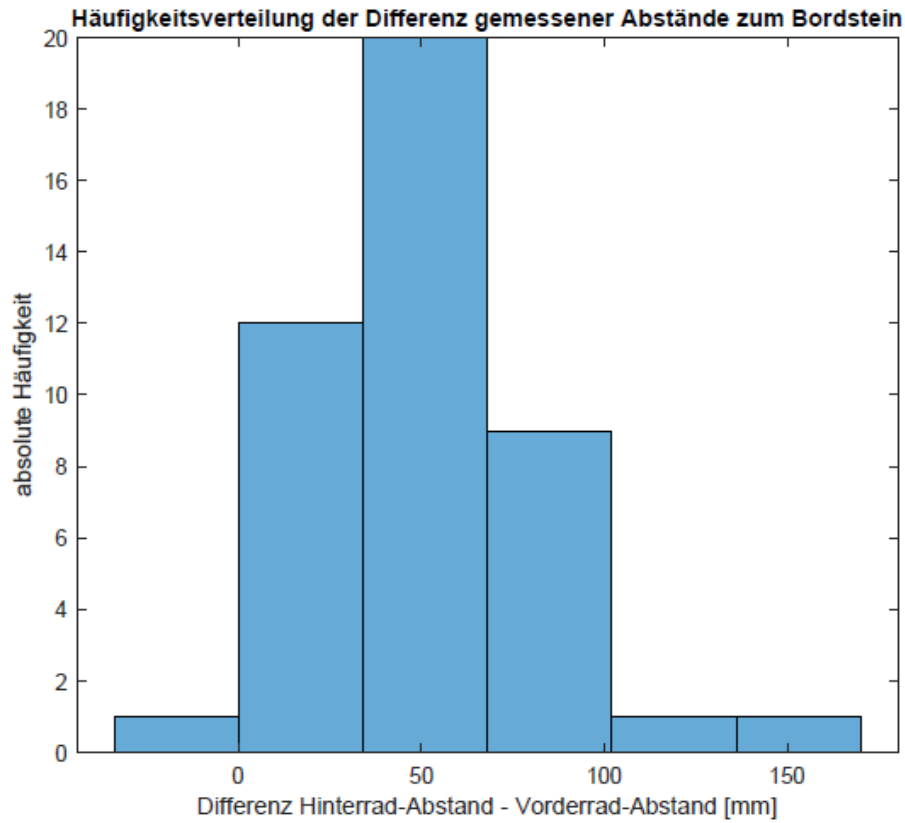


Abbildung 62: Häufigkeitsverteilung der Differenz aus Hinterrad-Abstand und Vorderrad-Abstand

Vergleich mit manuellen Messungen

Differenz aus beiden Abständen

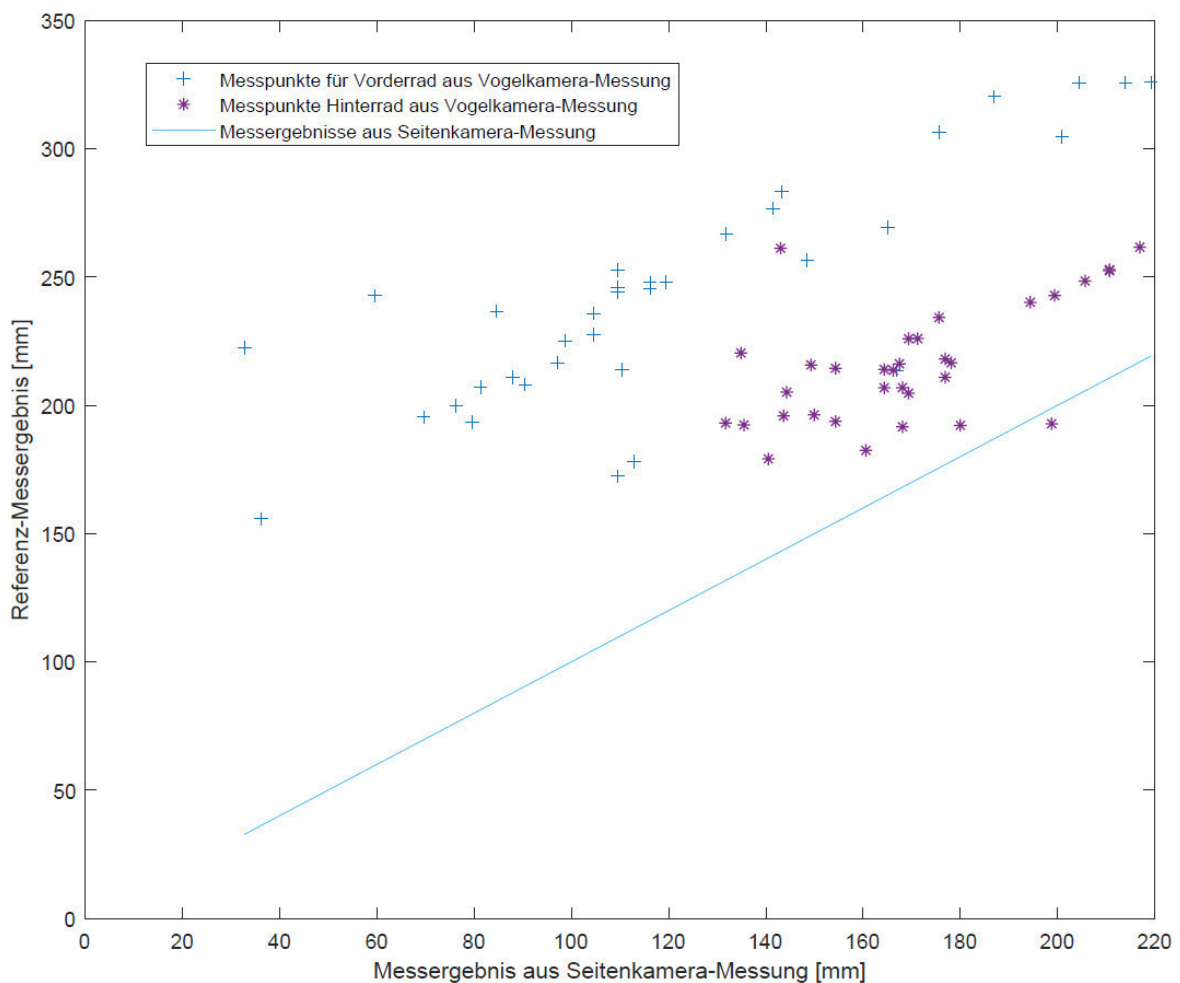
Für die Messungen, bei denen die Bordsteinabstände manuell aufgenommen werden, soll die Differenz zwischen Vorder- und Hinterrad mit der entwickelten Messmethode verglichen werden. Dazu folgt eine Übersicht in der folgenden Tabelle. Auch hier wird als Referenz für die Bewertung der Genauigkeit der arithmetische Mittelwert der Beträge der Differenzen zwischen beiden Messmethoden herangezogen. Dieser ergibt sich zu 1,63 mm.

manuelle Messung Vorderrad [mm]	manuelle Messung Hinterrad [mm]	Differenz [mm]	Messung mit Seitenkamera Vorderrad [mm]	Messung mit Seitenkamera Hinterrad [mm]	Differenz [mm]
121	146	25	119,5	148	28,5
185	185	0	186,1	186,9	0,8
35	183	148	32,8	180	147,2
80	165	85	79,6	166,2	86,6
90	175	85	90,5	176,9	86,4

Tabelle 31: Vergleich manuelle Messung vs. Messung mit Seitenkamera

Vergleich Messung mit vogelperspektivischen Aufnahmen vs. Messung mit Seitenkamera-Aufnahmen

Nach Transformation von Bildkoordinaten der Mittelpunkte der Fahrzeugmarkierungen am Ende des Einparkvorgangs werden die halben zuvor gemessenen Abstände der Fahrzeug-Außenflanken entsprechend der Orientierung des Fahrzeugs von den y-Koordinaten subtrahiert. Nach Subtraktion der y-Koordinate des Bordsteinfußes wird für die Differenz der theoretische Abstand zwischen Rad-außenflanke und Bordstein angenommen. Diese Ergebnisse sind tendenziell größer als die Ergebnisse aus den Messungen in den Seitenkamera-Aufnahmen, wie in folgender Abbildung zu sehen ist. Hier sind die Ergebnisse aus Messungen mit vogelperspektivischen Aufnahmen gegen die Ergebnisse aus Messungen mit Seitenkamera als Punkte aufgetragen. Als Referenz ist eine Linie eingetragen, die das Verhältnis 1:1 für die Ergebnisse aus Seitenkamera-Messungen wiedergibt.



Für einen Vergleich wird der RMSE-Wert für die Messergebnisse der beiden Messmethoden herangezogen. Dieser beträgt 129 mm und liegt damit deutlich höher als die geforderten Genauigkeiten zur Positionsbestimmung von 30 mm oder für die Bestimmung lateraler Abstände von 10 mm.

Nachdem die Messungen der Bordsteinabstände mit anderen Messmethoden verifiziert sind, ist hier eine Ungenauigkeit in der Messmethode der Transformation von Bildkoordinaten der vogelperspektivischen Aufnahmen in Raumkoordinaten zu vermuten.

5.9.4. Abstände zu geparkten Fahrzeugen

Seitenkamera-Aufnahmen

Vergleich mit manuellen Messungen

Die Messung der Abstände an der Fahrzeugfront und am Heck sind nach der Beschreibung in Abschnitt 5.7.10 erfolgt. Ein Vergleich mit manuellen Referenzmessungen soll die Genauigkeit der Messmethode zeigen. Dazu wird eine Regression vorgenommen. Dabei wird ein linearer Zusammenhang der Ergebnisse bei einem Bestimmtheitsmaß von **$R^2=0.9733$** und einem **RMSE-Wert von 24,81 mm** bei einem **Wertebereich von 575 mm bis 1044 mm** festgestellt. Dabei gilt folgender Zusammenhang, welcher in der folgenden Abbildung zusammen mit den Messwerten dargestellt ist, wobei x für das manuelle Messergebnis steht:

$$f(x) = 0,9307 * x + 54,19\text{mm} \quad (10)$$

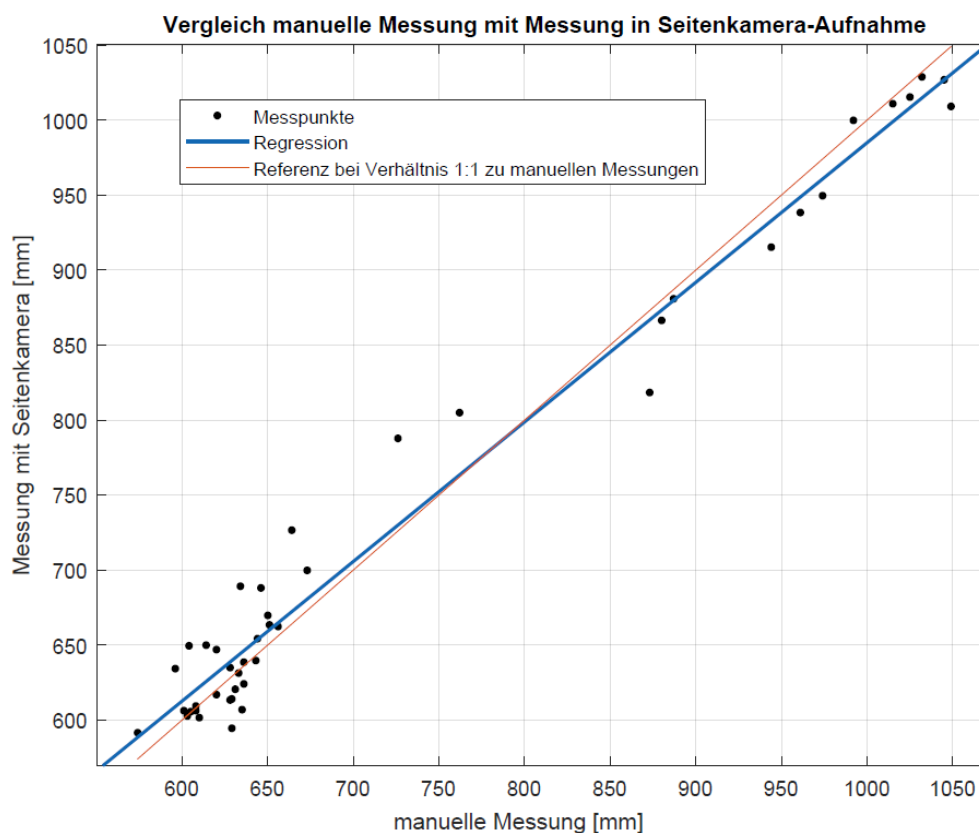


Abbildung 63: Korrelation manueller Messungen mit Messungen mit Seitenkamera-Aufnahmen für den Frontabstand

Die Funktion zeigt, dass mit steigenden Werten das Verhältnis der beiden Größen sich 1:1 annähert. Der RMSE-Wert liegt für die Regression mit 24,85 mm höher als die **geforderte Genauigkeit von 10mm** für die Bestimmung von Abständen in Längsrichtung deutlich. Ein direkter Vergleich der Messergebnisse liefert einen RMSE-Wert von 26,9 mm. Das arithmetische Mittel der Beträge aller **Differenzen** der Ergebnisse für die jeweiligen Messungen ergibt einen Wert von **20,16 mm**.

Es folgt eine grafische Darstellung aller Differenzbeträge. Dabei sind die Ergebnisse aus Messungen mit Seitenkamera von den Messergebnissen manueller Messung subtrahiert. Für Stufen mit mindestens 4 vorliegenden Ergebnissen sind die Standardabweichungen eingetragen. Hier ist zu erkennen, dass sich der Verlauf für den Frontabstand nicht exakt gegenläufig zu dem Verlauf für den Heckabstand verhält. Dieses kann damit begründet werden, dass manuelle Messungen nicht entlang der globalen x-Koordinate orientiert sind. Stattdessen sind diese bei Anlegen des Laser-Distanzmessers an einem festen Punkt des einen Fahrzeugs und Anpeilen eines festen Punktes am jeweils anderen Fahrzeugs erfolgt. Dagegen werden die Messungen mit Seitenkamera-Aufnahmen weniger durch eine Verdrehung des Fahrzeugs in der Parklücke beeinflusst. Die Position des Heckabschlusses wird dabei unter Berücksichtigung der Verdrehung (ermittelt aus lateralen Abständen) berechnet.

Bis zur Stufe 1718 mm ist ein gegenläufiger Verlauf deutlich zu erkennen. Bei größeren Rangierräumen können Fehler durch schräge Messung mit dem Laser-Distanzmesser zunehmen.

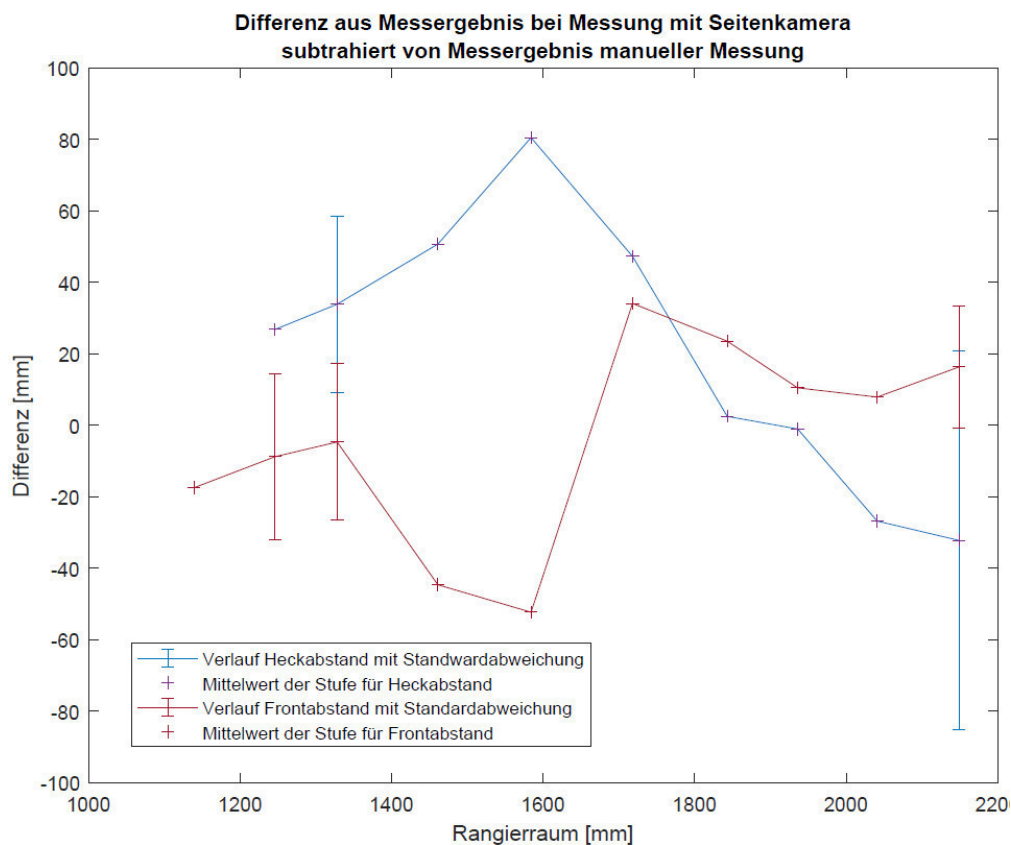


Abbildung 64: Differenz aus Ergebnissen manueller Messungen und Ergebnissen bei Messungen mit Seitenkamera

Die entsprechende Häufigkeitsverteilung zu den Differenzen für die Frontabstände ist in folgender Abbildung zu sehen. Der arithmetische Mittelwert beträgt für Ergebnisse aller Stufen -3,54 mm und die Standardabweichung liegt bei 28,2 mm. Bei den Stufen mit 13 und 15 Wiederholungen sind

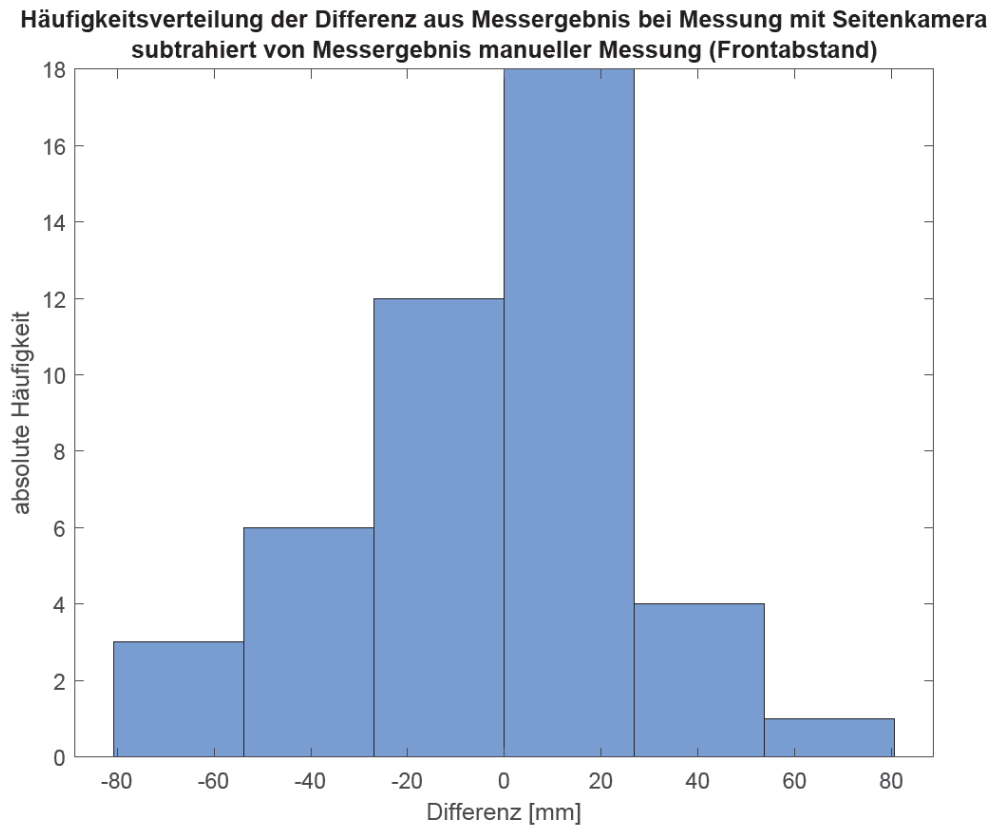


Abbildung 65: Häufigkeitsverteilung für Differenzen der Front-Abstandsmessungen

Die gleiche Untersuchung für den Heckabstand ergibt einen Mittelwert von 20,4 mm bei einer Standardabweichung von 43 mm.

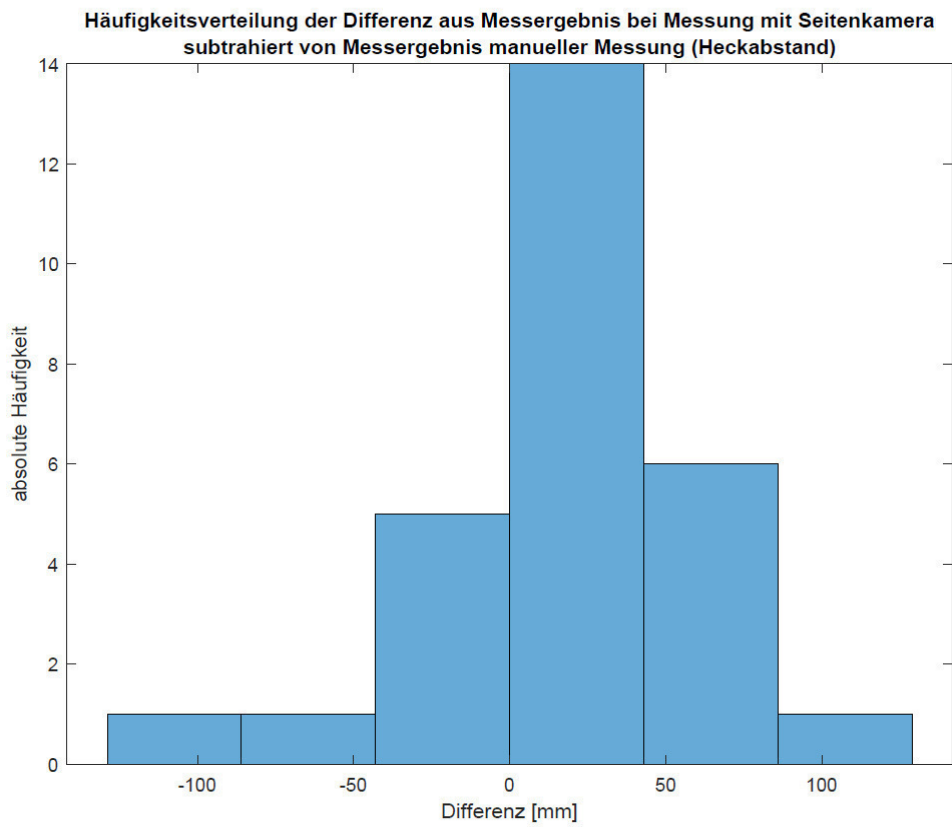
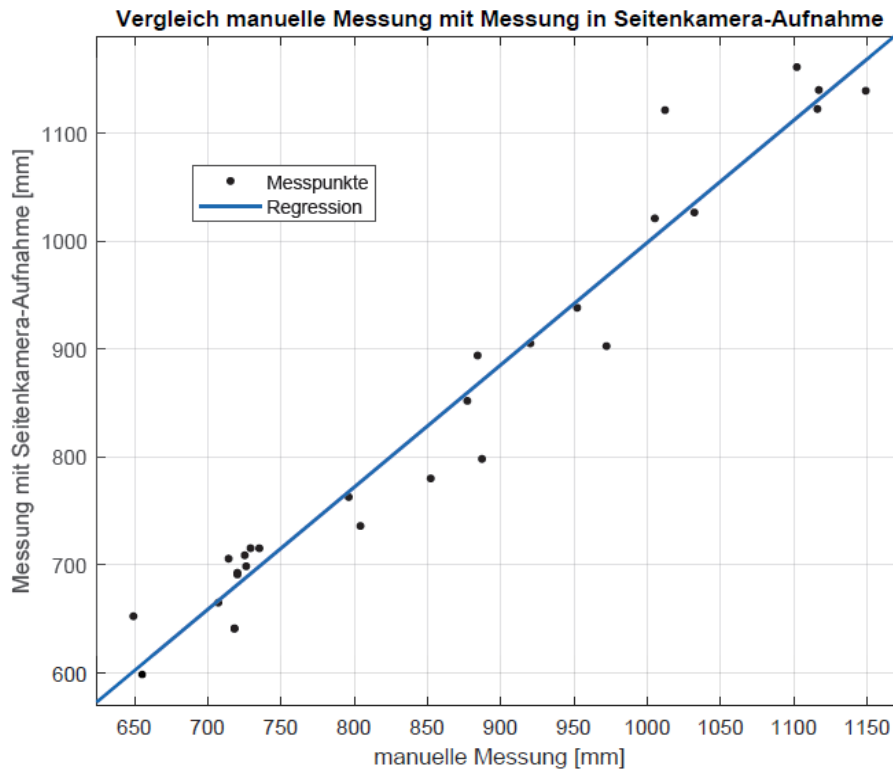


Abbildung 66: Häufigkeitsverteilung für Differenzen der Heck-Abstandsmessungen

Analog zum Frontabstand werden die Ergebnisse für den Heckabstand untersucht. Dabei ergeben sich bei der Regression Maßzahlen von $R^2= 0,957$ und $RMSE= 38.3 \text{ mm}$ bei folgendem Zusammenhang:

$$f(x) = 1,133 * x - 134,5\text{mm} \quad (11)$$

Damit ist auch hier ein deutlicher linearer Zusammenhang zu sehen, wobei der RMSE-Wert einen größeren Fehler, als bei dem Frontabstand widerspiegelt. Dementsprechend kann angenommen werden, dass die geforderte Genauigkeit hier noch deutlicher verfehlt wird. Der arithmetische Mittelwert aller Differenzbeträge ist bei diesem Vergleich **36,6 mm**.



Statistische Auswertung

Für die statistische Auswertung der Abstände entlang der Fahrzeuglängsachse werden die Ergebnisse aus Messungen mit den Seitenkameras herangezogen. Zunächst folgt eine Übersicht aller Ergebnisse der zweiten Messreihe. Die konstanten Summen der beiden Abstände für die jeweiligen Stufen sind hier deutlich zu erkennen. Dabei ist die Stufe von ca. 1150 mm Rangierraum, bei der nur ein erfolgreicher Versuch in dieser Messreihe durchgeführt wird. Weiterhin sind die zwei Stufen mit jeweils 13 und 15 Versuch Wiederholungen sowie die Stufen mit jeweils 2 Wiederholungen und die letzte mit 4 Wiederholungen zu erkennen.

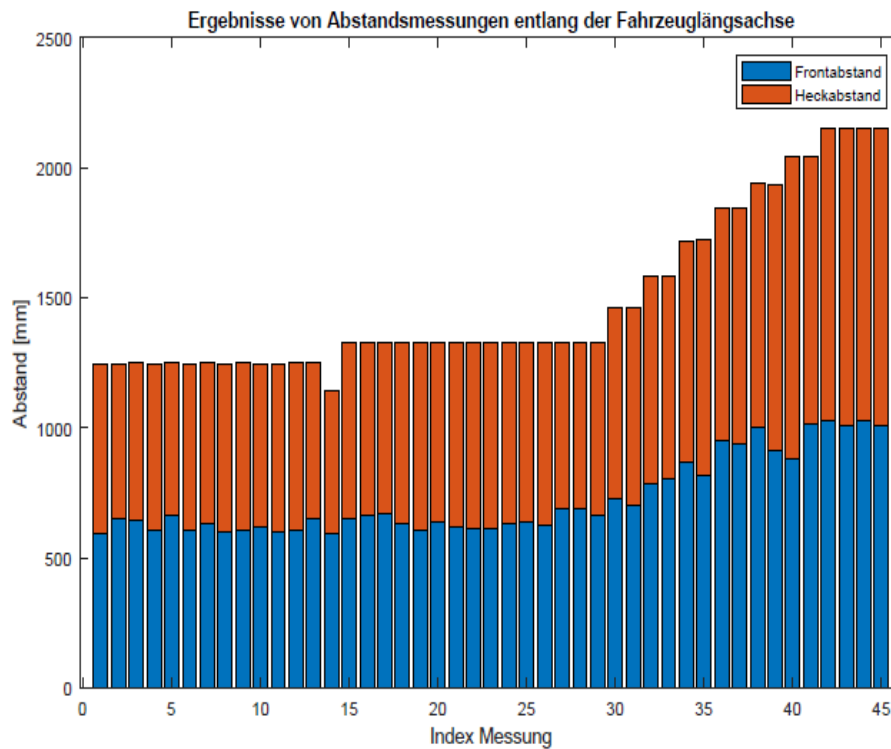


Abbildung 67: Endabstände der 2. Messreihe

Als nächstes soll die Differenz der beiden Abstände untersucht werden, wobei der Front- vom Heckabstand subtrahiert wird. Der Mittelwert aller Ergebnisse liegt mit 32,9 mm bei einer Standardabweichung von 68,4 mm. Das bedeutet der Heckabstand ist im Mittel größer als der Frontabstand.

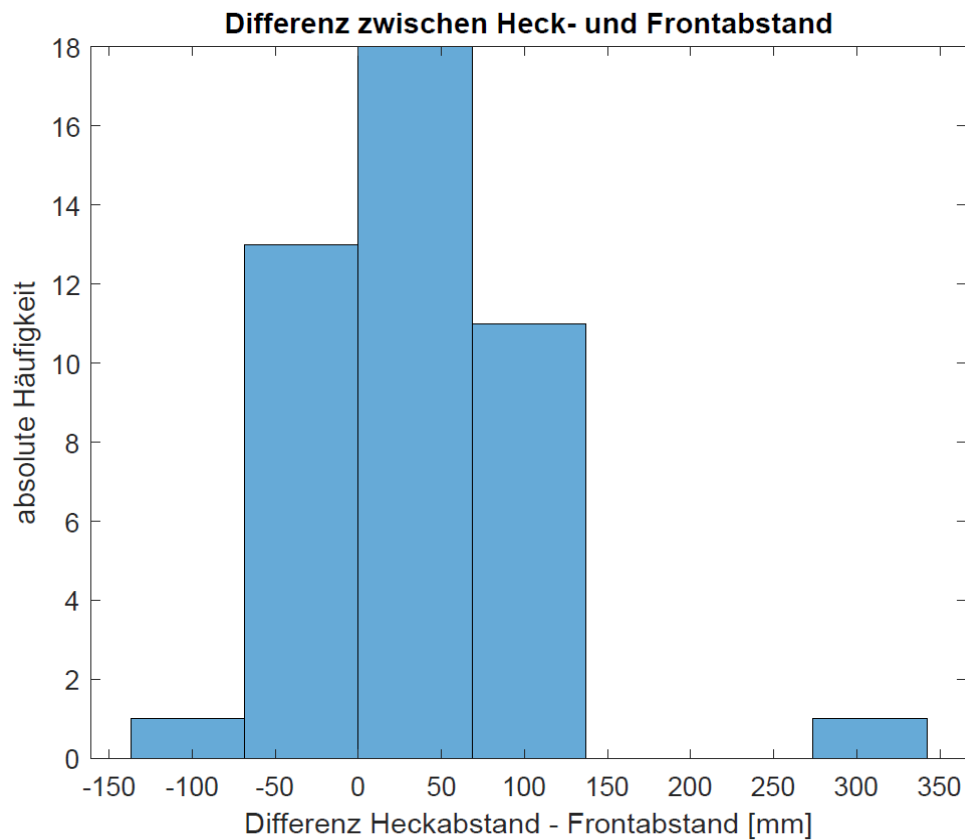


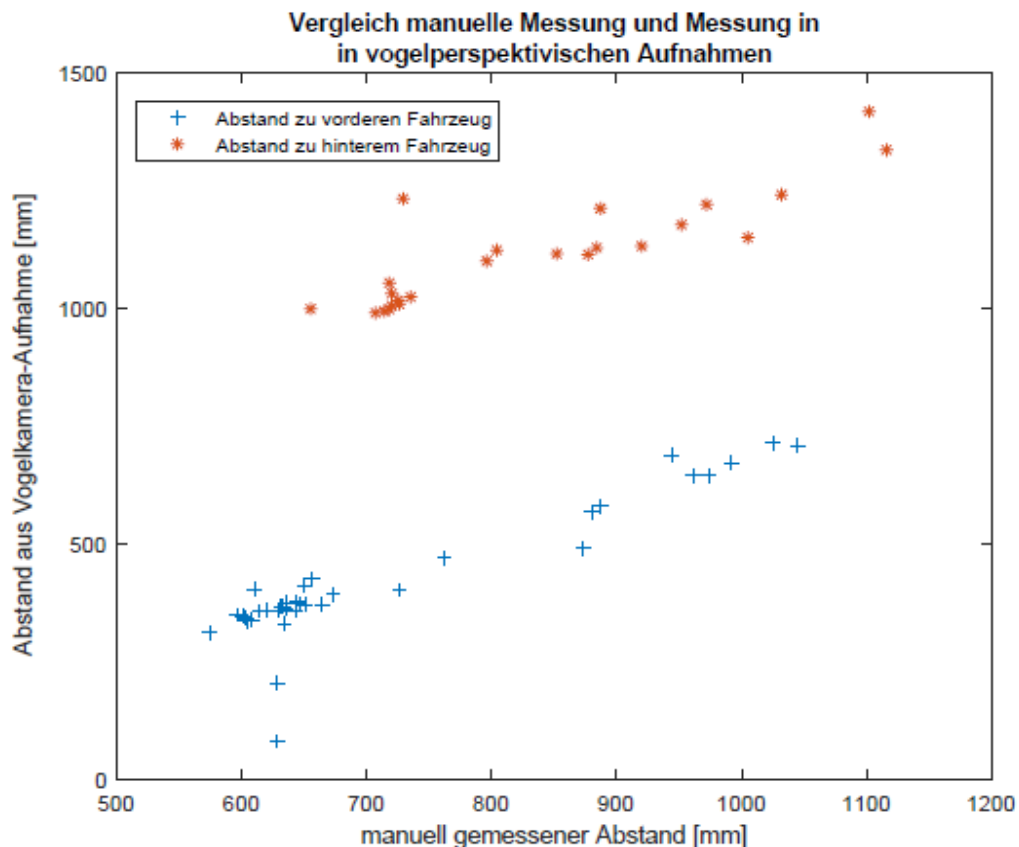
Abbildung 68: Differenz der Messungen für Front- und Heckabstände (2. Messreihe)

Außerdem soll die Genauigkeit der Messung der **Differenz** zwischen Front- und Heckabstand bewertet werden. Dazu liegen keine Absolutwerte vor, sodass nur eine Relativierung anhand manueller Messungen betrachtet werden kann. Zu diesem Zweck wird die Differenz zusätzlich für die manuellen Messungen berechnet. Eine Subtraktion der Ergebnisse von den in der letzten Darstellung gezeigten Ergebnissen liefert Werte im **Bereich -112 mm bis 150 mm** und der arithmetische Mittelwert aller Beträge einen Wert von **50 mm**. Bezogen auf das arithmetische Mittel aller manuell gemessenen Frontabstände beträgt der Anteil **6,8%**. Die geforderte Genauigkeit für Messungen entlang der Fahrzeuglängsachse beträgt 10 mm. Im Vergleich dazu liegt der Mittelwert deutlich höher.

Vogelperspektivische Aufnahmen

Vergleich mit manuellen Messungen

Aus den vogelperspektivischen Aufnahmen werden durch Transformation der Bildpositionen der Fahrzeugmarkierungen am Ende des Einparkvorgangs in Raumkoordinaten ermittelt. Mit den relativen Positionen der Markierungen zueinander wird der Orientierungswinkel des Fahrzeugs berechnet. Da der Fahrzeugüberhang der Front und des Hecks vom Versuchsfahrzeug bekannt sind, kann mit den Raumpositionen der Markierungen die Positionen des Frontabschlusses und Heckabschlusses des Versuchsfahrzeugs berechnet werden. Mit der x-Koordinate des Abschlusses des vorne geparkten Fahrzeugs und der Parklückengröße werden die Abstände zu den geparkten Fahrzeugen berechnet. Die Ergebnisse werden zunächst mit den manuellen Referenzmessungen verglichen. In der folgenden Darstellung ist zu sehen, dass die Werte-Bereiche der manuellen Messungen für Front- und Heckabstand sich deutlich überschneiden. Dagegen ist bei Messungen aus den vogelperspektivischen Aufnahmen ein Offset von ca. 500 mm zu erkennen. Dies lässt auf einen Fehler bei einem der Transformationsschritte von einem Koordinatensystem ins nächste schließen. Da der Offset über den Verlauf der Abszisse annähernd konstant erscheint, könnte der Fehler bei der Lokalisierung der Ortsmarkierung relativ zur Kamera liegen.



5.9.5. Orientierungswinkel

Die durch Messungen mit der Seitenkamera ermittelten Orientierungswinkel θ_M werden in folgender Übersicht mit Orientierungswinkeln aus den manuellen Messungen der Rad-Bordsteinabstände θ_S gegenübergestellt. Der arithmetische Mittelwert der **Differenzen** beträgt **-0,0058°**. Bezogen auf den Mittelwert der Orientierungswinkel aus manuellen Messungen beträgt der Anteil der gemittelten Differenz **0,33%**.

θ_M [°]	θ_S [°]	$\Delta\theta = \theta_M - \theta_S$ [°]
-0,93918225	-0,93602059	-0,00316167
-0,40265346	-0,40129777	-0,00135569
-3,23233637	-3,22147604	-0,01086033
-2,06195291	-2,05501627	-0,00693664
-2,0586009	-2,05167552	-0,00692538

Tabelle 32: Vergleich Orientierungswinkel aus manuellen Messungen und Messungen mit Seitenkamera-Aufnahmen

In folgender Darstellung ist zu sehen, dass die Orientierungswinkel des Fahrzeugs am Ende des Einparkvorgangs im Bereich um -1° bis -2° häufen. Der Mittelwert liegt hier bei -1,35 ° bei einer Standardabweichung von 0,66°.

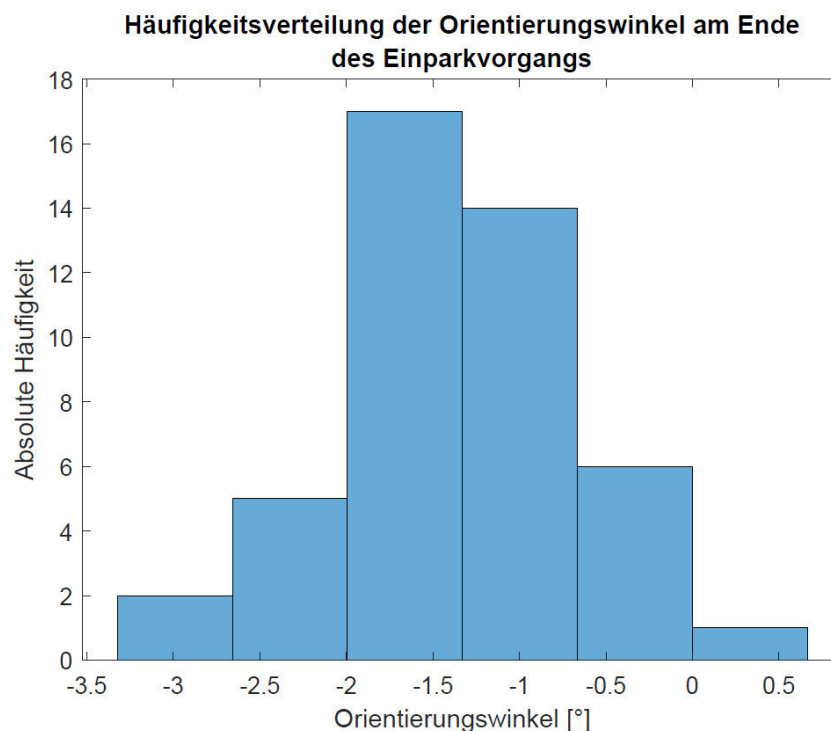


Abbildung 69: Häufigkeitsverteilung Orientierungswinkel (2. Messreihe)

5.9.6. Parkzuganzahl

Nachdem die Parkzüge gezählt sind, wird für jede Stufe der Mittelwert aus den aufgetretenen Maximal- und Minimalwert gebildet. Die Ergebnisse sind in folgendem Diagramm über dem Rangiererraum aufgetragen. Hier ist zu erkennen, dass die benötigte Anzahl an Parkzügen mit Vergrößerung des Rangiererraums abnimmt. Eine Ausnahme stellt die letzte Stufe dar, bei der die gemittelte Parkzuganzahl um 0,5 im Vergleich zur der vorherigen Stufe zunimmt.

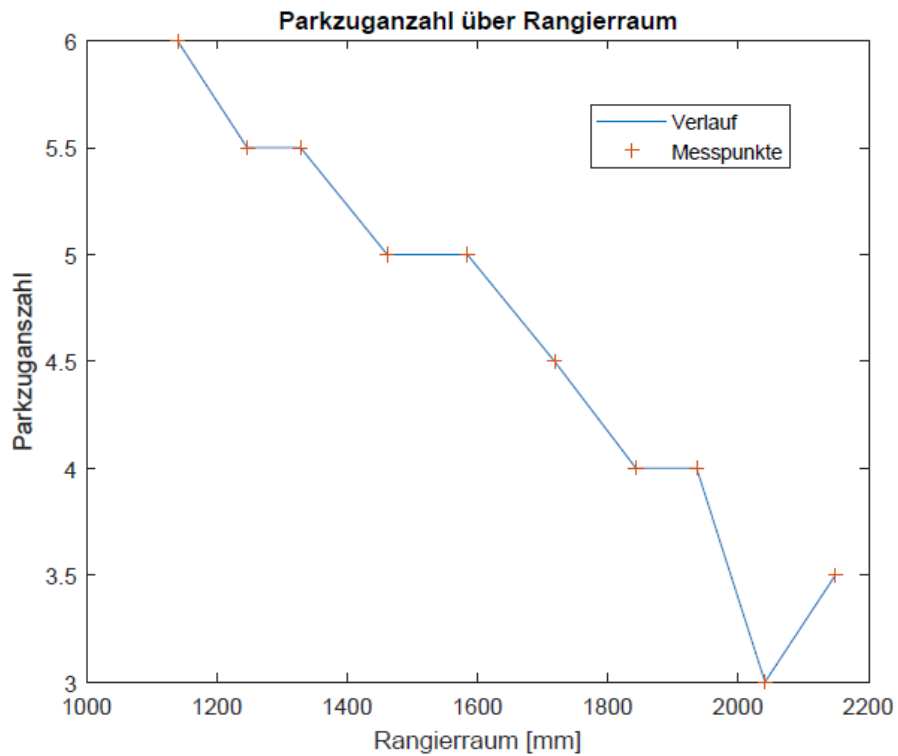


Abbildung 70: Parkzuganzahl über Rangiererraum

5.9.7. Dauer des Einparkvorgangs

Aus den ermittelten Zeiträumen zwischen Start und Ende der Einparkvorgänge wird für jede Reihe der arithmetische Mittelwert gebildet, um eine Abhängigkeit von dem Rangierraum zu untersuchen. Wie ein Vergleich zwischen der folgenden Grafik und der Darstellung zu der Anzahl der Parkzüge zeigt, ist die Abhängigkeit der Einparkdauer weniger kontinuierlich, als es bei den Parkzügen der Fall ist. Während die Tendenz der Parkzuganzahl bei größeren Parklücken eindeutig abnehmend ist, nehmen die Werte für die Einparkdauer in einigen Intervallen auch etwas zu.

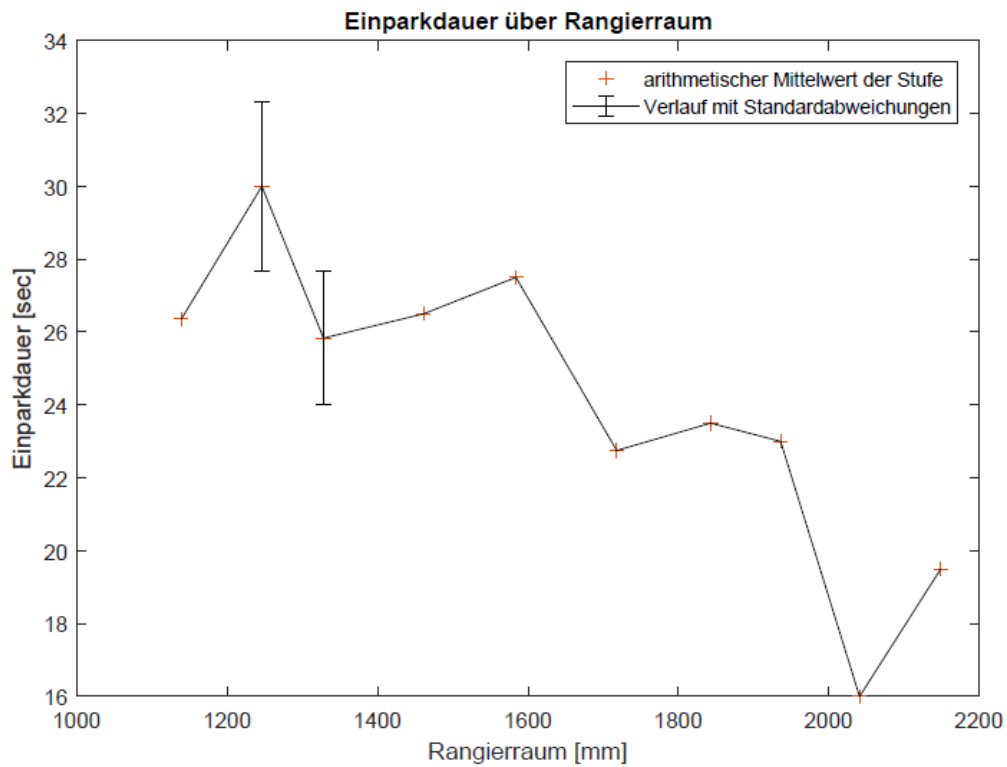


Abbildung 71: Einparkdauer über Rangierraum

Abschließend soll untersucht werden, ob und wie die Größen Parkzuganzahl und Parkdauer korrelieren. In der folgenden Darstellung ist zu erkennen, dass die Parkdauer mit der Parkzuganzahl ansteigt. Außerdem werden die Standardabweichungen bei größerer Anzahl an Parkzügen größer. Dieses kann auch mit der größeren Anzahl an Wiederholungen für Stufen mit kleinerem Rangierraum erklärt werden.

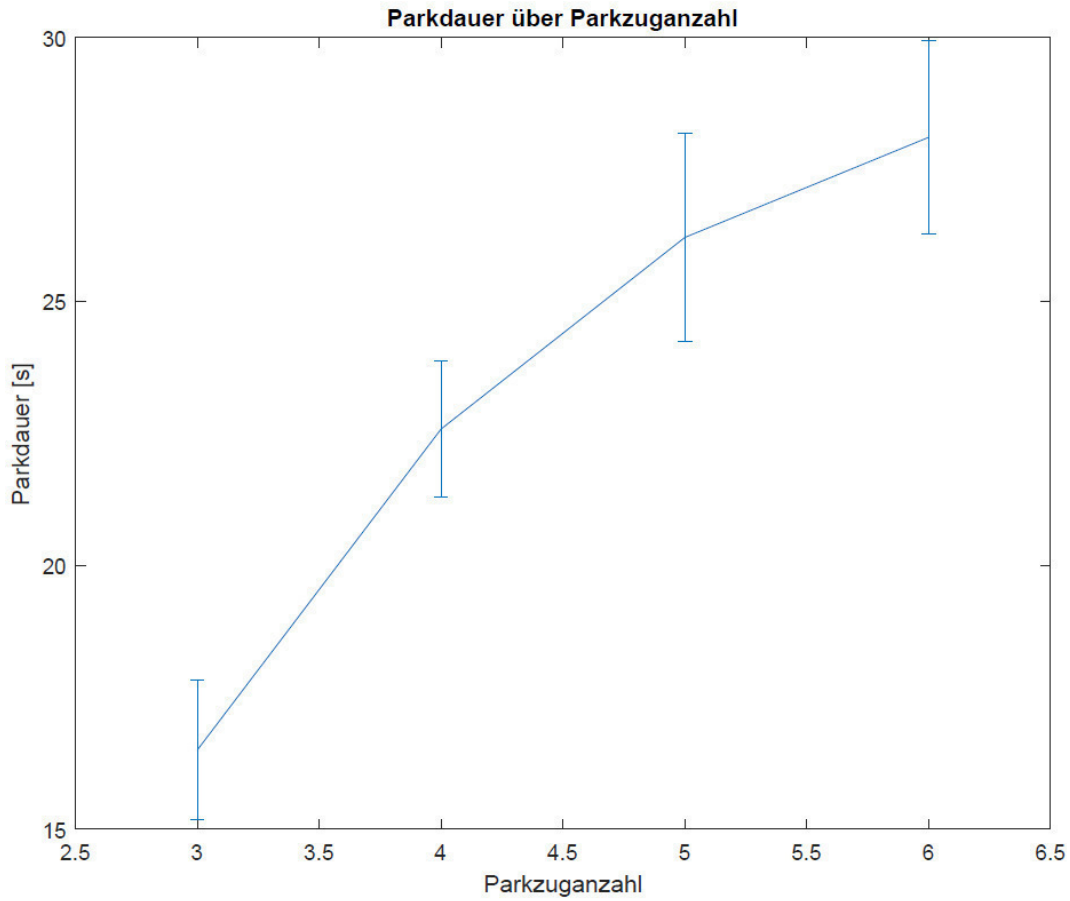


Abbildung 72: Zusammenhang zwischen Parkdauer und Parkzuganzahl

5.9.8. Trajektorien

In der folgenden Darstellung sind die Trajektorien in Raumkoordinaten für alle Messungen abgebildet. Die Referenz der Koordinaten ist dabei die Orts-Markierung Nr. 1. Die Trajektorien der Frontachs-Markierung enden alle bei x-Werten von über 1900 mm. Die Heckmarkierungs-Trajektorien enden alle etwa im Bereich -2000 mm ... 200 mm.

Bei Betrachtung der Positionen der Markierungen am Ende des Einparkvorgangs im unteren Teil des Bildes ist ein Abstand von ca. 3m zwischen den Markierungen zu erkennen. Dieses entspricht auch in etwa dem Radstand des Versuchsfahrzeugs. Außerdem enden die Trajektorien etwa im Bereich $y = 2200 \text{ mm} \dots 2300 \text{ mm}$.

Außerdem ist zu sehen, dass die Startpositionen der Einparkvorgänge auf ca. 1000 mm in y-Richtung verteilt sind. Um die Reproduzierbarkeit der Ergebnisse des Einparksystems zu prüfen, müssten mehrere Versuche mit gleichem Startpunkt und Startorientierung durchgeführt werden.

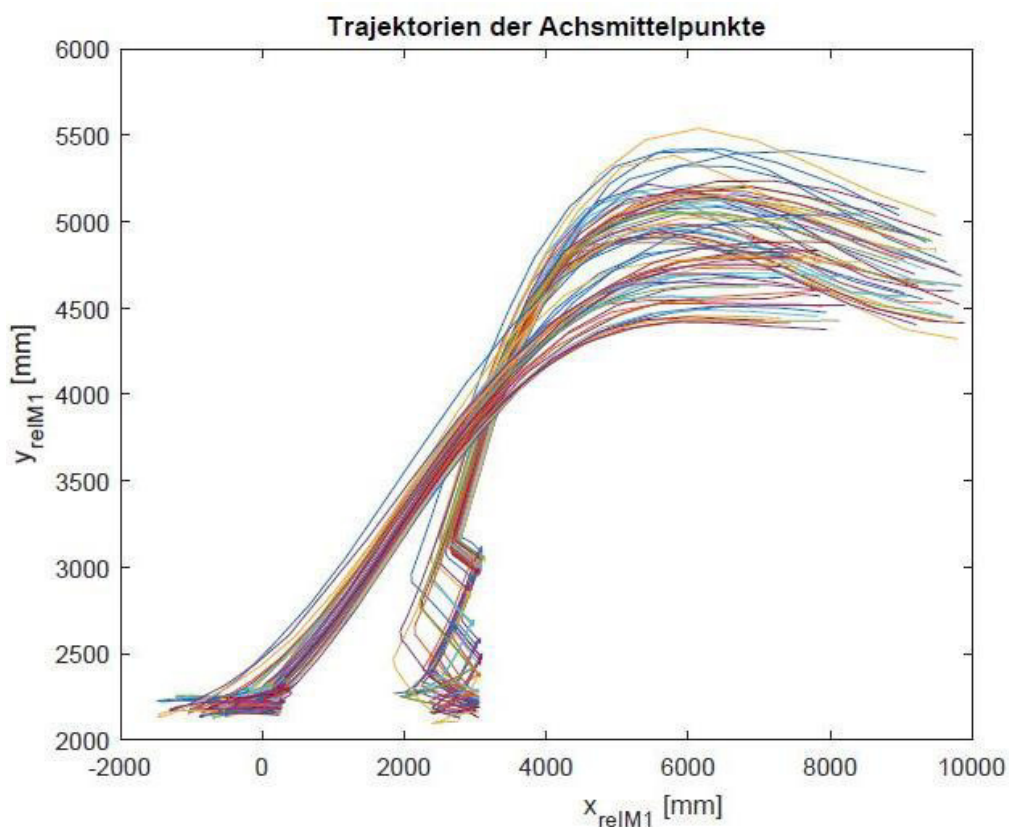


Abbildung 73: Trajektorien aus allen vogelperspektivischen Aufnahmen der 2. Messreihe

6. Zusammenfassung

In diesem Abschnitt soll das entwickelte Messsystem unter den in Abschnitt 2.5 festgelegten Kriterien geprüft werden. Dazu folgt eine tabellarische Übersicht der einzelnen Messgrößen. Um die Genauigkeit der Messmethoden zu bewerten, liegen keine absoluten Größen für Abstände und Positionen vor. Deshalb werden Differenzen zu Referenzmessungen betrachtet und diese mit den geforderten Genauigkeiten verglichen. Größen, welche die geforderten Werte erreichen, sind grau unterlegt.

Für die Bestimmung der Endposition des Fahrzeugs erreicht die Methode der geometrischen Transformation von Bildpunkten der Seitenkamera-Aufnahmen die besten Ergebnisse für die Auflösung. Außerdem sind hier die Abweichungen zu manuellen Referenzmessungen am geringsten. Dieses gilt sowohl für Abstände entlang der Fahrzeuglängsachse, als auch laterale Abstände und den aus diesen berechneten Orientierungswinkeln. Deshalb sind in der Tabelle unter den entsprechenden Messgrößen die erreichten Werte der Messungen mit Seitenkamera-Aufnahmen eingetragen.

Die Anzahl der Parkzüge sowie die Dauer des Einparkvorgangs und der Bewegungsverlauf werden mit vogelperspektivischen Aufnahmen ermittelt.

Eine Methode für die Bestimmung des Lenkwinkels konnte nicht vollständig implementiert werden, sodass an der entsprechenden Stelle keine Einträge stehen.

quantifiziertes Kriterium	Messbereich		Genauigkeit		Auflösung	
	gefordert	erreicht	gefordert	*	gefordert	erreicht
Differenz: Abstand Hinterrad zu Bordstein – Abstand Vorderrad zu Bordstein	0...0,05m	-1,09 m ...0,74 m/	10 mm	1,63 mm.	1 mm	1,44 mm
Differenz: Heckabstand - Frontabstand	0...1m	-2,09... 2,09 m	10 mm	20,2 mm	1 mm	2,2 mm
Abstand Rad-Außenflanke zu Bordstein	0...0,5m	0...1,09 m (vorn) 0...0,74 m (hinten)	10 mm	3,1 mm	1 mm	0,85 mm (vorn) 0,59 mm (hinten)
Abstände zu Fahrzeugen vor und hinter der Parklücke	0...30m	0...2,09 m	10 mm	20,2 mm	1 mm	1 mm...2,2 mm (³)
Lenkwinkel, Einlenkwinkel am gelenkten Rad	0...50°	-	1°	-	0,1°	-
Bewegungsrichtungsänderung	0...10	≥ 10	1	1	1	1
Dauer des Parkvorgangs ≤ 120/110/100/90/80/70/60/50/40/30/20/10 sec.	0...120 sec.	0...456 sec.	1 sec	> 1 sec	0,1 sec.	0,5 sec.
Anforderungen an Testverfahren erfüllt	Ja/nein	Ja/nein	-	-	ja	ja
Abbruch eingeleiteten Einparkvorgangs	Ja/nein	Ja/nein	-	-	-	-
Bewegungsverlauf des Fahrzeugs über der Zeit (2 Koordinaten auf der Fahrbahnebenen + Orientierungswinkel bezüglich der z-Achse)	0...10m	lateral: 0...9,1 m längs: 0...11,3 m	30 mm	129 mm **	3 mm	7,1 mm ... 14,1 mm (²³)
	0...50°	-0...>50°	1°	-	1 Hz	2 Hz
					0,3°	0,32...0,45° (³)
					1 Hz	2 Hz

*: Zum Vergleich wird der Betrag der arithmetisch gemittelten Differenzen der Messergebnisse zu Referenzmessungen mit Laser-Entfernungsmesser herangezogen.
**: RMSE aus Vergleich lateraler Endposition mit Ergebnis aus Messung mit Seitenkamera
²: abhängig von Richtung
³: abhängig von Bildbereich
-: keine Referenz vorhanden

Tabelle 33: Messbereich, Genauigkeit und Auflösung des Messkonzeptes

7. Fazit

Die vorliegende Arbeit mit dem Titel „Entwicklung eines Messsystems und einer Methode für die statistische Vermessung von automatisierten Einparksystemen“ hatte die Aufgabe ein System zu entwickeln, um die Endposition des Versuchs-Fahrzeugs bei automatisierten Einparkvorgängen präzise zu bestimmen. Dazu sollte Hardware am Fahrzeug appliziert werden, sodass diese über den im Versuchsfahrzeugs integrierten Rechner gesteuert werden kann. Darüber hinaus gehörte zu der Aufgabe eine Festlegung eines quantifizierten Versuchsplans zur statistischen Auswertung und Bewertung von Einparkvorgängen. Bei der Arbeit sollte im Urban Mobility Lab vorhandene Hardware eingesetzt werden, um zeitsynchrone Aufnahmen von Daten mit Zeitstempeln zu realisieren. Die Auswertung der Daten sollte zunächst offline erfolgen. Außerdem war der Einsatz von Simulations-Software geplant, um Messungen vor der Versuchsdurchführung auf der Versuchsstrecke zu simulieren. Die Beschaffung der ausgewählten Software war an der Hochschule nicht möglich.

Die Bestimmung der gesuchten Abstände längs und lateral zum Fahrzeug sowie des Orientierungswinkels zur Bestimmung der Endposition ist mit Kameras und Markierung des Bordsteins realisiert worden. Dabei sind Kameras über eine Sensorträger-Konstruktion an dem Dachgepäckträger des Fahrzeugs appliziert, sodass diese den Fahrbahnbereich an der Seite des Fahrzeugs aufnehmen. Nach einer Weiterentwicklung des ersten Entwurfes der Messmethode für die Bestimmung der Endposition, konnte zeitlicher und personeller Aufwand reduziert und die Auflösung erhöht werden.

Für die Bestimmung des Lenkwinkels ist ein Ansatz entwickelt worden, deren Umsetzung, bzw. Weiterentwicklung als Bestandteil weiterführender Arbeiten empfehlenswert ist. Der entwickelte Markierungskegel wird am Rad appliziert und bietet die Möglichkeit über die Bildaufnahmen, den Lenkwinkel direkt am Rad zu messen. Dieses kann automatisiert erfolgen.

Für Aufzeichnungen des Verlaufs von Position und Orientierungswinkel während des Einparkvorgangs ist ein auf Verarbeitung vogelperspektivischer Bild-Aufnahmen basierendes Konzept entwickelt worden. Mit den Aufnahmen konnten Einparkdauer, Anzahl benötigter Parkzüge und der Verlauf der Position von zwei Markierungen über den Achsmitten des Fahrzeugs verfolgt werden. Dazu sind nach einer Überarbeitung des ersten Entwurfs Markierungselemente entstanden, die durch die entwickelten Algorithmen in Bildsequenzen verfolgt werden.

Dabei funktioniert der Tracking-Algorithmus für unterschiedliche Blickwinkel. Damit die Algorithmen funktionieren, müssen Umriss der Markierung im Bild erkennbar sein. Die Transformation vogelperspektivischer Bildaufnahmen in Raumkoordinaten liefert für die Endposition deutlich von den Referenzmessungen abweichende Ergebnisse. Dennoch ist ein Vergleich von Trajektorien bei gleichen Startbedingungen möglich. Bei einer Weiterentwicklung des Systems wäre zeitsynchrone Aufnahme mehrerer Sensoren für Vergleiche von Messergebnissen oder Fusionierung empfehlenswert.

Bei Anpassungen einiger Parameter kann das Konzept durch seine Unabhängigkeit von elektronischen Schnittstellen des Fahrzeugs auch mit anderen Versuchsfahrzeugen umgesetzt werden. Die Erprobung des Konzepts an zwei Messtagen über mehrere Stunden zeigten, dass Messungen vergleichbare Ergebnisse liefern.

8. Ausblick

Die Genauigkeit und Auflösung der Messmethoden bietet noch Potential zur Steigerung. Erreicht werden kann dieses durch Einbeziehung von Daten über die Form der Fahrbahnoberfläche, bzw. mit Wahl einer vollkommen ebenen Fahrbahn bei Versuchen. Vor allem für geometrische Transformation von Bildkoordinaten in vogelperspektivischen Aufnahmen bietet dies ein großes Potential zur Steigerung der Genauigkeit. Außerdem kann durch optimierte Kalibrierungsmethoden die Lage und Orientierung von Kameras im Versuchsfeld genauer bestimmt werden, als es in dieser Arbeit geschehen ist. Weiterhin bietet eine exakte Bestimmung der Brennweite der eingesetzten Konfiguration Potential zur Steigerung der Messgenauigkeit. Dabei kann z. B. der Umstand, dass die Brennweite eines Objektivs oder einer Linse bedingt durch die Form von Linsen nicht an jedem Punkt gleich ist mit in die Berechnungen einbezogen werden. Zur Erhöhung der Auflösung können Abstände der Kameras und Blickwinkel optimiert werden oder in Sensoren mit einer höheren Anzahl an Bildpunkten investiert werden.

Das erwähnte Problem der festen Belichtungszeit bei Video-Aufnahmen kann gelöst werden, indem mit der angebotenen „Pylon API“ für das Programmierinterface Python ein eigenes Programm zur Steuerung der Kamera geschrieben wird. Das Problem der Verdeckung von Markierungen kann dadurch gelöst werden, dass für freie Sicht gesorgt wird. Alternativ kann die Größe der Markierung während des Trackings geprüft werden. Überdeckung der Markierung führt dazu, dass der Abstand der Markierungsgrenzen sich plötzlich stark verändert. Wenn dieses detektiert wird, kann kurzzeitige Verdeckung, z. B. durch Aussetzen der Suche nach der Mitte der Markierung kompensiert werden.

Weiterhin können neben oder anstelle von Kameras auch andere Sensoren zur Lokalisierung eingesetzt und Daten auch fusioniert werden. Dazu bieten sich z. B. SLAM-Algorithmen, ggf. in Kombination mit Kalman-Filtern an. Das Thema SLAM in Kombination mit EKF-Filtern ist von Dr. Joan Sola behandelt worden (Sola 2014) Diese Kombination ist in der Robotik (Negenborn 2013; Burgard u. a., o. J.) und auch Pfadplanung bei Einparkvorgängen (Tay 2007; Hahn 2016) oder anderen Systemen verbreitet.

Auch Ansätze für SLAM mit nur Winkeln, unter denen Markierungen detektiert werden existieren (Deans 2005; Strasdat u. a. 2007; Phan und Ovchinnikov 2013; Lemaire, Lacroix, und Sola 2005). Dieser Ansatz bietet sich vor allem beim Einsatz von Kameras an. Weiterhin gibt es bereits umgesetzte Konzepte zur Fusionierung von Daten sphärischer Kameras mit LIDAR-Sensoren (Zhao 2015). Automatisierte Detektierungen von Markierungen werden z. B. für Augmented-Reality Anwendungen entwickelt (Hirzer 2008).

Für die Bestimmung des Lenkungswinkels kann der Ansatz einer Messung des Lenkwinkels am Rad durch Applikation eines durch Rapid-Prototyping gefertigten Radmarkierungskegels weiterverfolgt werden. Dabei wird dieser in einer Farbe ausgeführt, die sich von der Umgebung abhebt. Dadurch kann seine relative Orientierung zu einem Referenzbild bestimmt werden, indem z. B. die Bildkoordinate der Spitze des Kegels gefunden und verglichen wird. Weiterhin wäre eine Kombination mit „Shape Recognition“-Ansätzen („Shape Recognition - MATLAB Central“ o. J.) denkbar.

Der Ansatz einer Skala am Lenkrad zum Ablesen des Lenkradwinkels konnte keine exakten Messergebnisse liefern, da das Lenkrad häufig außerhalb der Skala stand. Wenn diese einen Bereich von -10° bis $+10^\circ$ abdeckt, sollte damit ein ausreichender Bereich abgedeckt werden. Für die Festlegung eines Grenzwertes sollte die Lenkübersetzung vorher bestimmt werden, um unterschiedliche Einparksysteme testen zu können.

Über die Messmethodik hinaus kann auch die Versuchsdurchführung automatisiert werden. Mit einem Versuchs-Fahrzeug z. B., wäre es möglich, dieses über die CAN-Schnittstelle zu steuern. Am Beispiel des vollautomatischen Einparkens müssten dazu die Bremse, der Gangwechsel, Blinker und das entsprechende Menü gesteuert werden. Zusätzlich könnten Online-Auswertungen es ermöglichen Versuche bei Überwachung aus der Ferne und gegebenenfalls Anpassung der Versuchszyklen bei neu gewonnen Erkenntnissen durchzuführen.

Die Start- oder Endzeit des Einparkvorgangs kann automatisiert durch Auswertung von Verschiebungen von „feature-Points“ erfolgen. Entsprechend der Start- und Endzeit können gemessene Abstände am Ende eines Einparkvorgangs, die Trajektorie, Anzahl der Parkzüge, Orientierungswinkel des Fahrzeugs und Lenkungswinkel dem Parkvorgang zugeordnet werden.

9. Anhang

9.1. Programmcode

In diesem Abschnitt ist der Programmcode der verwendeten Funktionen zu finden. Dieser ist im Code kommentiert, wobei sich die Kommentare jeweils auf den folgenden Code beziehen. Die Dokumentation erfolgt in der Art und Weise, dass zuerst die übergeordnete Hauptfunktion zitiert wird. Die durch die Hauptfunktion aufgerufenen Unterfunktionen folgen danach. Die Programme werden weiterhin nach den Sensoren, deren erzeugte Daten diese verarbeiten sollen unterteilt.

9.1.1. Seitenkameras

Messung des Abstandes Rad-Bordstein durch Rasterauswertung

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  Hauptprogramm Rasterauswertung %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Zunächst wird der Abstand der Vorderrad-Außenflanke zum Bordstein anhand
% der Lage des Radaufstandspunktes im Bild ermittelt. Außerdem sind hier
% die Abmaße der angelegten Raster zur Messung einzugeben, welche durch die
% Funktionen dann ausgewertet werden. Für die Ermittlung des
% Hinterradabstandes werden lediglich andere Koordinaten für den
% Radaufstandspunkt an der Rad-Außenflanke eingegeben.

Breite_Rasterstreifen = 2;           % Angabe in mm
Breite_Endstreifen   = 11;          % Angabe in mm

y_latsch_v           = 359;
x_latsch_v           = 685;
Image_v              = imread('Front.bmp');

dv                   = Rasterauswertung(x_latsch_v, y_latsch_v
,Image_v, Breite_Rasterstreifen, Breite_Endstreifen);

y_latsch_h           = 550;
x_latsch_h           = 735;
Image_h              = imread('Heck.bmp');

dh                   = Rasterauswertung(x_latsch_h, y_latsch_h
,Image_h, Breite_Rasterstreifen, Breite_Endstreifen);

% Anhand der ermittelten Abstände und der hier einzugebenden Größen für den
% Radstand und Position des Markers relativ zur Hinterachse wird der
% theoretische Bordstein-Abstand auf der Höhe des Markers mit dem
% Strahlensatz berechnet.

Rs                   = 2960;         % Radstand Tesla Model S in mm
S_mh                 = 840;         % Abstand Marker zu Hinterachse

d_FH                 = dv - dh;     % Abstandsdifferenz bezüglich des
                                   % Bordsteins zwischen Vorderrad und
                                   % Hinterrad

dm                   = dh - S_mh * abs(d_FH) / Rs;
```

Funktion zur Berechnung des Abstandes durch Auswertung des Messrasters (1. Entwurf)

```
function Abstand_zu_Bordstein = Rasterauswertung(x_latsch, y_latsch ,Image,
Breite_Rasterstreifen, Breite_Endstreifen)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rasterauswertungs-Funktion für die Vermessung des Abstandes zwischen %
% Reifen und Bordstein %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Das auszuwertende Bild wird eingelesen und anschließend in ein
% grayscale-Image umgewandelt. Dieses wird in ein Array umgewandelt,
% welches für jeden Pixel des Bildes den Intensitätsgradienten enthält.
    I          = Image;
    Igr        = rgb2gray(I);

    [Gxr, ~] = imgradientxy(Igr);

% Der Endstreifen, welcher um ein Vielfaches breiter ist, als die
% Zählstreifen wird gesucht, indem Die Pixel der Zählzeile, welche ab der
% Rad-Außenflanke mit den aufeinanderfolgenden verglichen wird. Sobald 5
% Pixel nacheinander einen Intensitätsunterschied von weniger als 10%
% aufweisen, wird der erste Pixel dieser 5 als Koordinate für das Ende der
% Zählzeile in der Variable x-Ende aufgenommen.
    Bordstein_detektiert=0;
    i=0;
    xi=1;
    while Bordstein_detektiert==0
        if xi==5;
            Bordstein_detektiert=1;
            break
        elseif abs((double(abs(double(Igr(y_latsch,x_latsch+i)) -
double(Igr(y_latsch,x_latsch+xi+i)))))) /
double((Igr(y_latsch,x_latsch+i))))<0.1
            xi=xi+1;
        else
            xi=1;
            i=i+1;
        end
    end
    x_ende=x_latsch+i;

% Die Zählzeile wird in den Bereich zwischen der Rad-Außenflanke und dem
% breiten weißen Streifen des Rasters ins Bild gelegt und die
% Vorzeichenwechsel im Verlauf des Intensitätsgradienten werden gezählt.
% Die Vorzeichenwechsel ergeben mit der Breite der Streifen multipliziert
% und dazu die Breite des ersten Streifens sowie des breiten am Ende des
% Rasters den Gesamtabstand Rad-Außenflanke zum Bordstein.

    zaehlzeile          = Gxr(y_latsch,x_latsch:x_ende);
    pos                 = zaehlzeile>0;
    changes              = xor(pos(1:end-1),pos(2:end));
    Vorzeichenwechsel   = sum(changes);

    Abstand_zu_Bordstein = Vorzeichenwechsel * Breite_Rasterstreifen +
Breite_Endstreifen;
end
```



```

        Abstand_Latsch_Vertikale(1)          =
tand(Winkel_Latsch_Vertikale(1)) * z_latsch_v;

        Abstand_Latsch_Bordstein(1)         =
Abstand_Latsch_Vertikale(1) - Abstand_Bordstein_Vertikale(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Heckkamera %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        y_latsch_h          = 550;
        x_latsch_h          = 735;
        z_latsch_h          = 1440;
% Berechnung der maximalen Gegenstandsweite am Horizont der Kamera über die
% Linsengleichung

        G_hh                = b_h * z_latsch_h / f;
        l_hp                = G_hh / cameraParamsHeck.ImageSize(2);

% Bild von dem Hinterrad gestreckt mit Achsen wird in einer Figure
% angezeigt.
        imagesc(undistortImage((Heck),cameraParamsHeck));
        MaximizeFigureWindow
        Bordsteinposition_heck = ginput(1);

% Die Pixelanzahl zwischen Latsch und Bordstein wird mit der entsprechenden
% Länge pro Pixel in die einen Abstand in mm umgerechnet.

        PixelAbstand_Bordstein_Vertikale(2) =
cameraParamsHeck.PrincipalPoint(1) - Bordsteinposition_heck(1);

        Winkel_Bordstein_Vertikale(2)      =
PixelAbstand_Bordstein_Vertikale(2) * cam_Winkel_H / cam_Res_H;

        Abstand_Bordstein_Vertikale(2)     =
tand(Winkel_Bordstein_Vertikale(2)) * z_latsch_h;

        PixelAbstand_Latsch_Vertikale(2)   =
cameraParamsHeck.PrincipalPoint(1) - x_latsch_h;

        Winkel_Latsch_Vertikale(2)        =
PixelAbstand_Latsch_Vertikale(2) * cam_Winkel_H / cam_Res_H;

        Abstand_Latsch_Vertikale(2)       =
tand(Winkel_Latsch_Vertikale(2)) * z_latsch_h;

        Abstand_Latsch_Bordstein(2)       =
Abstand_Latsch_Vertikale(2) - Abstand_Bordstein_Vertikale(2);

        FileName=[datestr(now, 'dd-mmm-yyyy'),'Winkelmessung.mat'];
        save(FileName,'Abstand_Latsch_Bordstein')

```

Bordstein-Abstandsvermessung mit Detektierung des markierten Bordsteins (2. Entwurf)

```
for i = 1:46
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Bordstein-Abstandsvermessung über Winkel %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Der Index bestimmt den zu öffnenden Ordner
    oldfolder = cd(int2str(i));

% In diesen Variablen werden der obere und untere Grenzwert festgelegt, um
% die Bordsteinmarkierung zu finden.
    og = 170;
    ug = 90;

% Zwischen einem fixen Punkt am Reifen und dem Bordstein wird über die
% Bordsteinposition im Bild mit den geometrischen Verhältnissen der Abstand
% ermittelt. Um den Bordstein zu detektieren wird
    Front = imread('Front.bmp');
    Heck = imread('Heck.bmp');

% Öffnungswinkel der Kamera errechnet über die Linsengleichung mit den
% Abmessungen des Sensors.
    cam_Winkel_V = 31.7;
    cam_Winkel_H = 48.6;

% Auflösung der Kamera über die in der Camera-Calibration-App erzeugten
% Kamera-Parameter ausgelesen.
    cam_Res_V = cameraParamsFront.ImageSize(1);
    cam_Res_H = cameraParamsFront.ImageSize(2);

% Daten des Sensors & Kamera
    b_h = 11.3; % Sensorgröße horizontal in mm
    b_v = 7.1; % Sensorgröße vertikal in mm
    f = 12.5; % Bildweite der Kamera (Back nach
              % Datenblatt Kamera und Objektiv

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Frontkamera %%%%%%%%%

% Die fixen Startpunkte für die Abstandsvermessung zum Bordstein werden für
% die Vergleichbarkeit aus der Rasterauswertung übernommen. z_latsch_v
% wurde mit dem Lasermessgerät, welches an das Kameragehäuse angelegt wurde
% und
    y_latsch_v = 455;
    x_latsch_v = 701;
    z_latsch_v = 1960;
    z_Bordstein_v = 1849;

% Bild der Frontkamera wird mit den Kalibrierungs-Parametern der Front-
% Kamera entzerrt eingelesen.
    Img_v = undistortImage((Front), cameraParamsFront);
    ROI = rgb2gray(Front(y_latsch_v,x_latsch_v:1920,:));
    bs_v = find(ROI > og,1);
    bs_v_sub = find(fliplr(ROI(1:bs_v)) < ug,1);
    Bordsteinposition_front = bs_v + x_latsch_v - bs_v_sub;

% Mit dem Abstand zwischen Kamerasensor und Radaufstandspunkt in der
% z-x-Ebene wird aus den Bildpunkten mit dem Öffnungswinkel sowie Lage des
% optischen Zentrums der Kamera im Bild der Abstand zwischen Reifen und
% Bordstein berechnet.
```

```

PixelAbstand_Bordstein_Vertikale(1) =
cameraParamsFront.PrincipalPoint(1) - Bordsteinposition_front;

Winkel_Bordstein_Vertikale(1) =
PixelAbstand_Bordstein_Vertikale(1) * cam_Winkel_H / cam_Res_H;

Abstand_Bordsteinoberkante_Vertikale(1)=
tand(Winkel_Bordstein_Vertikale(1)) * z_Bordstein_v;

Abstand_Bordsteinfuss_Vertikale(1) =
tand(Winkel_Bordstein_Vertikale(1)) * z_latsch_v;

PixelAbstand_Latsch_Vertikale(1) =
cameraParamsFront.PrincipalPoint(1) - x_latsch_v;

Winkel_Latsch_Vertikale(1) =
PixelAbstand_Latsch_Vertikale(1) * cam_Winkel_H / cam_Res_H;

Abstand_Latsch_Vertikale(1) =
tand(Winkel_Latsch_Vertikale(1)) * z_latsch_v;

% Der Abstand wird in Abhängigkeit des Winkels zwischen optischer
% Achse und der detektierten Bordsteinkante berechnet.

if ((cameraParamsFront.PrincipalPoint(1) - Bordsteinposition_front)
* cam_Winkel_H / cameraParamsFront.ImageSize(2)) > 6.5
    Abstand_Latsch_Bordstein(1) =
Abstand_Latsch_Vertikale(1) - Abstand_Bordsteinoberkante_Vertikale(1) - 15;
else
    Abstand_Latsch_Bordstein(1) =
Abstand_Latsch_Vertikale(1) - Abstand_Bordsteinfuss_Vertikale(1);
end

Bordsteinmessungen(i,5) =
((cameraParamsFront.PrincipalPoint(1) - Bordsteinposition_front) *
cam_Winkel_H / cameraParamsFront.ImageSize(2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Heckkamera %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Daten werden analog zur Frontkamera ermittelt und eingesetzt.

y_latsch_h = 614;
x_latsch_h = 776;
z_latsch_h = 1416;
z_Bordstein_h = 1377;

% Berechnung der maximalen Gegenstandsweite am Horizont der Kamera über die
% Linsengleichung

G_hh = b_h * z_latsch_h / f;
l_hp = G_hh / cameraParamsHeck.ImageSize(2);

% Bild von dem Hinterrad wird analog zur Frontkamera-Aufnahme entsprechend
% der Parameter für Verdrehung der Heck-Kamera und Kalibrierung der
% Heck-Kamera eingelesen.
Img_h =
undistortImage((Heck), cameraParamsHeck);
ROI =
rgb2gray(Heck(y_latsch_h,x_latsch_h:1920,:));
bs_h = find(ROI > og,1);

```

```

        bs_h_sub          = find(fliplr(ROI(1:bs_h)) < ug,1);
        Bordsteinposition_heck = x_latsch_h + bs_h - bs_h_sub;

% Analog zur Frontkamera wird hier der Abstand zwischen Hinterrad und
% Bordstein berechnet.

        PixelAbstand_Bordstein_Vertikale(2) =
cameraParamsHeck.PrincipalPoint(1) - Bordsteinposition_heck;

        Winkel_Bordstein_Vertikale(2) =
PixelAbstand_Bordstein_Vertikale(2) * cam_Winkel_H / cam_Res_H;

        Abstand_Bordsteinoberkante_Vertikale(2) =
tand(Winkel_Bordstein_Vertikale(2)) * z_Bordstein_h;

        Abstand_Bordsteinfuss_Vertikale(2) =
tand(Winkel_Bordstein_Vertikale(2)) * z_latsch_h;

        PixelAbstand_Latsch_Vertikale(2) =
cameraParamsHeck.PrincipalPoint(1) - x_latsch_h;

        Winkel_Latsch_Vertikale(2) =
PixelAbstand_Latsch_Vertikale(2) * cam_Winkel_H / cam_Res_H;

        Abstand_Latsch_Vertikale(2) =
tand(Winkel_Latsch_Vertikale(2)) * z_latsch_h;

% Der Abstand wird in Abhängigkeit des Winkels zwischen optischer
% Achse der Kamera und der detektierten Bordsteinkante berechnet.

        if ((cameraParamsHeck.PrincipalPoint(1) - Bordsteinposition_heck) *
cam_Winkel_H / cameraParamsHeck.ImageSize(2)) > 8.7
            Abstand_Latsch_Bordstein(2) =
Abstand_Latsch_Vertikale(2) - Abstand_Bordsteinoberkante_Vertikale(2) - 15;
        else
            Abstand_Latsch_Bordstein(2) =
Abstand_Latsch_Vertikale(2) - Abstand_Bordsteinfuss_Vertikale(2);
        end

        Bordsteinmessungen(i,1:2) = Abstand_Latsch_Bordstein;
        Bordsteinmessungen(i,3) = og;
        Bordsteinmessungen(i,4) = ug;
        Bordsteinmessungen(i,6) =
((cameraParamsHeck.PrincipalPoint(1) - Bordsteinposition_heck) *
cam_Winkel_H / cameraParamsHeck.ImageSize(2));

%     FileName=[datestr(now, 'dd-mmm-yyyy'),'Bordsteinmessung.mat'];
%     save(FileName,'Bordsteinmessungen')
cd(oldfolder)
    close
    close
end

```

Abstände zu Fahrzeugen

Messungen mit vorderem Ende der Markierung im Bild

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Messungen Längsposition aus %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Seitenkamera-Aufnahmen %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Protokoll=xlsread('Protokoll.xlsx');
load('Bordsteinmessungen3.mat');
%%
UeberhangFront          = 937;
UeberhangHeck           = 1081;
cam_Winkel_V            = 31.7;
cam_Winkel_H            = 48.6;
FrontFzgPos             = 1340;
Abstand_VorderradAufstandspunkt = 1367;
Abstand_M2_Streifenende = 273-50;
Hoehe_Cam_Bordstein     = 1404-95;
% Für die Messungen 3 bis 36 ist das obere Ende der Markierung im Bild zu
% sehen. Dieses wird für die Messungen im entzerrten Bild gesucht und mit
% den bekannten Abständen und Positionen ergibt sich der Abstand an
% der Fahrzeugfront.

for i= 3 : 36

    oldfolder=cd(int2str(i));
    Front=imread('Front.bmp');
    Front=undistortImage(Front,cameraParamsFront,'OutputView','full');
    imagesc(Front)
    hold on
% Hier wird die horizontale Position der Bordsteinecke ausgelesen und
% mit ihr durch Verschiebung um 150 Bildpunkte nach rechts die
% Zählspalte festgelegt.
    Spaltenindex = cameraParamsFront.PrincipalPoint(1) -
Bordsteinmessungen(i,5) * 1920 / cam_Winkel_H + 150;
    Zaehlspalte=(Front(:,round(Spaltenindex)));

% In der Zählspalte wird nach dem Index des ersten Bildpunktes mit
% einem Intensitätswert von über 200 gesucht und zur Kontrolle wird der
% gefundene Punkt angezeigt.
    ix = find(Zaehlspalte > 200,1);
    plot(round(Spaltenindex),ix,'+', 'MarkerSize',20)
    MarkerEnde(i,1:2)=[ix,round(Spaltenindex)];
    Orientierungswinkel=abs(Bordsteinmessungen(i,7));

% in der Variablen n wird die Anzahl der Bildpunkte der Spalte im
% entzerrten Bild aufgenommen.
    n = sum(Front(:,MarkerEnde(i,2))>0);
% Der relative Winkel zur optischen Achse wird berechnet.
    MarkerEnde(i,3) = (cameraParamsFront.PrincipalPoint(2) - ...
    MarkerEnde(i,1)) * cam_Winkel_V / n;
    MarkerEnde(i,4) = tand(MarkerEnde(i,3)) * Hoehe_Cam_Bordstein;
% Der zur Fahrzeuglängsachse parallele Abstand zwischen Kamera und dem
% betrachteten Markierungspunkt im Raum wird berechnet. Mit den bekannten
% Relativ-Positionen ergeben sich Front- und Heckabstand.
    MarkerEnde(i,5) = tand(MarkerEnde(i,3) + 43) *
Hoehe_Cam_Bordstein;
    MarkerEnde(i,6) = MarkerEnde(i,5) -
Abstand_VorderradAufstandspunkt;
    MarkerEnde(i,6) = Abstand_M2_Streifenende - MarkerEnde(i,6);
    MarkerEnde(i,7) = MarkerEnde(i,6) + UeberhangFront;
end
```

```
MarkerEnde(i,8) = FrontFzgPos - MarkerEnde(i,7);
if i==36
    MarkerEnde(:,9) = Protokoll(1:i,13);
    MarkerEnde(:,10) = MarkerEnde(:,8) - MarkerEnde(:,9);
    MarkerEnde(2,11) = mean(abs(MarkerEnde(7:36,10)));
end
cd(oldfolder)
end
```

Messungen mit hinteren Ende der Markierung im Bild

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Messungen Längsposition aus %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Seitenkamera-Aufnahmen %%%%%%%%%
Protokoll=xlsread('Protokoll.xlsx');
load('Bordsteinmessungen.mat');
UeberhangFront = 937;
UeberhangHeck = 1081;
cam_Winkel_V = 31.7;
cam_Winkel_H = 48.6;
FrontFzgPos = 1340;
Abstand_VorderradAufstandspunkt = 1367;
Abstand_M2_Streifenende = 273;
Abstand_M2_Streifenanfang = 815+122;
Hoehe_Cam_Bordstein = 1404-95;
Parklueckenlaenge = Protokoll(:,12);
% Für die Messungen 37 bis 46 ist das untere Ende der Markierung im Bild zu
% sehen. Dieses wird für die Messungen im entzerrten Bild gesucht und mit
% den bekannten Abständen und Positionen ergibt sich der Abstand an
% der Fahrzeugfront.

for i= 37 : 46
    oldfolder=cd(int2str(i));
    Front=imread('Front.bmp');
    Front=undistortImage(Front,cameraParamsFront);
    imagesc(Front)
    hold on
% Hier wird die horizontale Position der Bordsteinecke ausgelesen und
% mit ihr durch Verschiebung um 70 Bildpunkte nach rechts die
% Zählspalte festgelegt.
    Spaltenindex=cameraParamsFront.PrincipalPoint(1)-...
        Bordsteinmessungen(i,5) * 1920/cam_Winkel_H + 70;
    Zaehlspalte=flipud(Front(:,round(Spaltenindex)));
% In der Zählspalte wird nach dem Index des ersten Bildpunktes mit
% einem Intensitätswert von über 200 gesucht und zur Kontrolle wird der
% gefundene Punkt angezeigt.
    ix = find(Zaehlspalte > 200,1);
    plot(ix,round(Spaltenindex),'+', 'MarkerSize',20)
    MarkerEnde(i,:) = [ix,round(Spaltenindex)];
    Orientierungswinkel= abs(Bordsteinmessungen(i,7));
% in der Variablen n wird die Anzahl der Bildpunkte der Spalte im
% entzerrten Bild aufgenommen.
    n = sum(Front(:,MarkerEnde(i,2))>0);
% Der relative Winkel zur optischen Achse wird berechnet.
    MarkerEnde(i,3) = (cameraParamsFront.ImageSize(1)...
        - MarkerEnde(i,1) - cameraParamsFront.PrincipalPoint(2))...
        * cam_Winkel_V / n;
% Der zur Fahrzeuglängsachse parallele Abstand zwischen Kamera und dem
% betrachteten Markierungspunkt im Raum wird berechnet.
    MarkerEnde(i,4) = tand(- MarkerEnde(:,3) + 43) *
Hoehe_Cam_Bordstein;
% Aus der Differenz zum Abstand des Vorderradaufstandspunktes zur Kamera
% ergibt sich der Abstand zwischen Radaufstandspunkt und dem
% Markierungspunkt.
    MarkerEnde(i,5) = Abstand_VorderradAufstandspunkt -
MarkerEnde(:,5);
% Mit dem relativen Abstand zur Ortmarkierung 2 ergibt sich die Position
% des Radaufstandspunktes im globalen Koordinatensystem.
    MarkerEnde(i,5) = Abstand_M2_Streifenanfang - MarkerEnde(:,6);
% Mit dem Front-Überhang des Fahrzeugs ergibt sich die globale Lage der
% Fahrzeugfront.
```



```
MarkerEnde(i,6) = UeberhangFront - MarkerEnde(:,6);
% Die Differenz zwischen der Lage des Abschlusses des vorderen geparkten
% Fahrzeugs und der Lage der Fahrzeugfront ergibt den Front-Abstand.
MarkerEnde(i,7) = FrontFzgPos - MarkerEnde(:,7);
EnbabstandFront(i) = MarkerEnde(:,7);
% Mit dem Orientierungswinkel aus den Bordsteinmessungen und der
% Parklückenlänge aus dem Protokoll ergibt sich der Heckabstand.
EnbabstandeHeck(i) = Parklueckenlaenge(i) - ...
    cosd(Orientierungswinkel)*4979 - EnbabstandFront(i)
cd(oldfolder)
end
```

9.1.2. Kamera in Vogelperspektive

Hauptprogramm Tracking von schwarzen Fronthauben-Marker

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% _Tracking %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% von schwarzen Fronthauben-Marker %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Die Aufnahme wird Bild für Bild durchlaufen und dabei die lokalen
Verschiebung
% von Ecken und Kanten von einem Bild zum nächsten betrachtet.

% Kameraparameter aus der Kalibrierung der hier eingesetzten Kamera
% werden geladen, um die Bilder zu entzerren.
load('BaslerWeitwinkelParams.mat');

% Für die Verschiebung wird ein Startwert eingegeben.
    Markerverschiebung      = [100 -50];
% Images, welche in einem Array gespeichert sind, werden geladen.
    load('Images.mat');

% Arbeitsspeicher für die Variablen wird belegt.
    MarkerInitPos      = zeros(1,2);
    Marker_ImagePos    = zeros(size(Images, ndims(Images)),2);
    QM                  = zeros(size(Images, ndims(Images)),1);
    figure
    MaximizeFigureWindow;
    %%

for i = 1 : (size(Images, ndims(Images)) - 1)

    I1      = undistortImage(Images(:,:,i),cameraParamsWeitwinkel);
    I2      = undistortImage(Images(:,:,i + 1),cameraParamsWeitwinkel);

% Falls noch keine Initiale Markerposition aufgenommen wurde (für
% den Fall das dieser erst später in der Sequenz im Bild
% erscheint), soll dieser per Maus eingegeben werden.
    if MarkerInitPos(1,:) == [0, 0]
        imagesc(I1);
        w = waitforbuttonpress;
        if w == 0 % Bei Klicken Markerposition aufnehmen
            MarkerInitPos(1,:) = ginput(1);
            % Die eingegebene Position des Markers wird durch die Funktion zum
            % finden der Markermitte korrigiert und in dass entsprechende Array
            % als erste Markerposition in der Bildsequenz aufgenommen.
            Marker_ImagePos(i,:) = ...
                MarkerMittefindenV4(I1,
[round(MarkerInitPos(2)),round(MarkerInitPos(1))], 2);
        else
% Bei Tastendruck zum nächsten Bild gehen, ohne Ausführung der
% weiteren Anweisungen in der Schleife.
            continue
        end
    end

% Für den gefundenen Mittelpunkt wird ein Qualitätsmerkmal berechnet.
% Dieser ergibt sich aus den arithmetisch gemittelten
% Intensitätswerten der umgebenden Pixel. Der Parameter s bestimmt, wie
% viele Pixel der Umgebung dazu betrachtet werden.
    s      = 15;
    QM(i)  = mean(mean(I1((Marker_ImagePos(i ,1) - 15) : ...
```

```

                (Marker_ImagePos(i ,1) + 15),...
                (Marker_ImagePos(i ,2) - 15) : ...
                (Marker_ImagePos(i ,2) + 15)))));

% Die Variable range legt den um die Position des Markers
% aufgespannten Bereich zur Betrachtung von Verschiebungen fest.
range          = [Marker_ImagePos(i,2) - 50 , Marker_ImagePos(i,2) + 500;...
                  Marker_ImagePos(i,1) - 115 , Marker_ImagePos(i,1) + 130];

% Es sollen nur Bildpunkte innerhalb des Images betrachtet werden.
range(range<0)          = 1;
range(range>1920)       = 1920;

Markerverschiebung      = (verschiebung(range, I1, I2));

% Das folgende Bild wird im offenen figure angezeigt und im
% weiteren Verlauf des Programms können hier die Abtastungen,
% Verschiebungen oder gefundenen Mittelpunkte sowie deren Verläufe
% angezeigt werden.
    imagesc(I2);

% Es wird zunächst ein Bruchteil der berechneten Verschiebung zur
% alten Marker-Position addiert und Punkte oberhalb und unterhalb
% der neuen Position betrachtet, bis ein Punkt mit einem
% Intensitätswert, welche betraglich nicht größer als 140% des
% vorherigen Qualitätsmerkmals ist gefunden wird.
Marker_ImagePos(i + 1,:) = round(Marker_ImagePos(i,:) + ...
                                fliplr(Markerverschiebung)*2/3);
    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), 'o');
QM(i + 1)          = mean(mean(I2((Marker_ImagePos(i + 1,1) - 15):...
                                (Marker_ImagePos(i + 1,1) + 15),...
                                (Marker_ImagePos(i + 1,2) - 15):...
                                (Marker_ImagePos(i + 1,2) + 15)))));

while QM(i + 1) > QM(i) * 1.4
    Marker_ImagePos(i + 1,:) = round(Marker_ImagePos(i + 1,:) + ...
                                    fliplr(Markerverschiebung)/8);
    QM(i + 1)          = mean(mean(I2((Marker_ImagePos(i + 1,1) - 15):...
                                    (Marker_ImagePos(i + 1,1) + 15),...
                                    (Marker_ImagePos(i + 1,2) - 15) : (Marker_ImagePos(i + 1,2) + 15)))));
    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '*');
    if I2(Marker_ImagePos(i + 1,1) + ...
        20 ,Marker_ImagePos(i + 1,2)) < QM(i) * 1.5 &&...
        I2(Marker_ImagePos(i + 1,1) + 20 ,Marker_ImagePos(i + 1,2)) <...
        I2(Marker_ImagePos(i + 1,1),Marker_ImagePos(i + 1,2))

        Marker_ImagePos(i + 1,1) = Marker_ImagePos(i + 1,1) + 20;
        QM(i + 1) = mean(mean(I2((Marker_ImagePos(i + 1,1) - 15): ...
                                (Marker_ImagePos(i + 1,1) + 15),...
                                (Marker_ImagePos(i + 1,2) - 15):...
                                (Marker_ImagePos(i + 1,2) + 15)))));
        plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '*');
        continue
    elseif I2(Marker_ImagePos(i + 1,1) + 30 ,Marker_ImagePos(i + 1,2)) <...
        QM(i) * 1.5 &&...
        I2(Marker_ImagePos(i + 1,1) + 30 ,Marker_ImagePos(i + 1,2))...
        < I2(Marker_ImagePos(i + 1,1),Marker_ImagePos(i + 1,2))

        Marker_ImagePos(i + 1,1) = Marker_ImagePos(i + 1,1) + 20;
        QM(i + 1) = mean(mean(I2((Marker_ImagePos(i + 1,1) - 15):...
                                (Marker_ImagePos(i + 1,1) + 15),...
                                (Marker_ImagePos(i + 1,2) - 15):...
                                (Marker_ImagePos(i + 1,2) + 15)))));

```

```

                                (Marker_ImagePos(i + 1,2) + 15)));
    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '*');
    continue
elseif I2(Marker_ImagePos(i + 1,1) + 40 ,Marker_ImagePos(i + 1,2)) < ...
    QM(i) * 1.5  &&...
    I2(Marker_ImagePos(i + 1,1) + 40 ,Marker_ImagePos(i + 1,2)) < ...
    I2(Marker_ImagePos(i + 1,1),Marker_ImagePos(i + 1,2))
        Marker_ImagePos(i + 1,1) = Marker_ImagePos(i + 1,1) + 20;
        QM(i + 1) = mean(mean(I2((Marker_ImagePos(i + 1,1) - 15):...
                                (Marker_ImagePos(i + 1,1) + 15),...
                                (Marker_ImagePos(i + 1,2) - 15):...
                                (Marker_ImagePos(i + 1,2) + 15))));
        plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '*');
    continue
end

if I2(Marker_ImagePos(i + 1,1) - 20 ,Marker_ImagePos(i + 1,2)) <...
    QM(i) * 1.5  &&...
    I2(Marker_ImagePos(i + 1,1) - 20 ,Marker_ImagePos(i + 1,2)) <...
    I2(Marker_ImagePos(i + 1,1),Marker_ImagePos(i + 1,2))
        Marker_ImagePos(i + 1,1) = Marker_ImagePos(i + 1,1) - 20;
        QM(i + 1) = mean(mean(I2((Marker_ImagePos(i + 1,1) - 15) :...
                                (Marker_ImagePos(i + 1,1) + 15),...
                                (Marker_ImagePos(i + 1,2) - 15) : (Marker_ImagePos(i + 1,2) + 15))));
        plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '*');
    continue
elseif I2(Marker_ImagePos(i + 1,1) - 30 ,Marker_ImagePos(i + 1,2)) <...
    QM(i) * 1.5  &&...
    I2(Marker_ImagePos(i + 1,1) - 30 ,Marker_ImagePos(i + 1,2)) <...
    I2(Marker_ImagePos(i + 1,1),Marker_ImagePos(i + 1,2))
        Marker_ImagePos(i + 1,1) = Marker_ImagePos(i + 1,1) - 20;
        QM(i + 1) = mean(mean(I2((Marker_ImagePos(i + 1,1) - 15):...
                                (Marker_ImagePos(i + 1,1) + 15),...
                                (Marker_ImagePos(i + 1,2) - 15):...
                                (Marker_ImagePos(i + 1,2) + 15))));
        plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '*');
    continue
end
end

    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), 'o');

    Marker_ImagePos(i + 1,1)=...
    FindeMitteHorizontalRelSP(I2, Marker_ImagePos(i + 1,:), 2);
    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '+');

    Marker_ImagePos(i + 1,2)=...
    FindeMitteVertikalRelSP(I2, Marker_ImagePos(i + 1,:), 2);
    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '+');

    Marker_ImagePos(i + 1,1)=...
    FindeMitteHorizontalRelSP(I2, Marker_ImagePos(i + 1,:), 2);
    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '+');

    % Hier wird die Funktion für das finden des Mittelpunktes des Markers
    % aufgerufen. Dieser wird das aktuelle Image sowie die durch die
    % Funktion Verschiebung bestimmte Position übergeben.
    Marker_ImagePos(i + 1,:) = ...
    MarkerMittefindenV4(I2, [Marker_ImagePos(i + 1, 1),...
    Marker_ImagePos(i + 1, 2)], 2);
    plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1), '+');
    k
                                = find(Marker_ImagePos(:,1),1);

```

```
plot(Marker_ImagePos(k:i,2),Marker_ImagePos(k:i,1),'x');  
plot(Marker_ImagePos(i+1,2),Marker_ImagePos(i+1,1),'x','MarkerSize',20)  
plot(Marker_ImagePos(k:i + 1,2), Marker_ImagePos(k:i + 1,1));  
end
```

Markermitte finden in schwarzen Fronhauben-Marker

```
function MarkerMitte = MarkerMittefindenV4(Image, Startpunkt, f)
% Koordinaten in der Form [Y, X] eingeben! ginput gibt [X, Y] aus!
% Es wird nach den Markergrenzflächen gesucht und aus den Koordinaten
% wird die Mitte des Markers errechnet. % Den Funktionen
"FindeMitteHorizontalRelSP"
% und "FindeMitteVertikalRelSP" wird je nach Suchrichtung der nächste
% Bildpunkt neben dem Referenzpunkt übergeben.
figure; imagesc(Image);
MaximizeFigureWindow
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Vertikal %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Der Referenzpunkt ist zu Anfang der eingegebene Startpunkt. Danach wird
% der Mittelpunkt der zuletzt gefundenen Zeile als Referenzpunkt
eingesetzt.
    clear MitteVertikal
    MitteVertikal(1,2) = FindeMitteVertikalRelSP(Image, Startpunkt, f);
    MitteVertikal(1,1) = Startpunkt(1);
    i=1;
    Referenzpunkt      = Startpunkt;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% nach tiefster Zeile im Marker suchen
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Vom Startpunkt aus gehend wird für den nächsten Bildpunkt unter dem
% Referenzpunkt der Mittelpunkt der Zeile gesucht, falls der tiefer
% liegende Punkt einen Intensitätswert hat der unter 200 liegt und unter
% dem des Referenzpunkt mit dem Parameter f multipliziert.
    while Image(Referenzpunkt(1) , Referenzpunkt(2) + 1)...
        < f * double(Image(Startpunkt(1),Startpunkt(2))) &&...
            Image(Referenzpunkt(1) , Referenzpunkt(2) + 1)...
            < 200
        % Der Faktor wird für die Bestimmung des Mittelpunktes der
        % Zeile entsprechend dem Verhältnis zwischen aktuellem
        % Referenzpunkt zum Startpunkt angepasst.
        Faktor = double(Image(Referenzpunkt(1), Referenzpunkt(2) + 1))...
            / double(Image(Startpunkt(1),Startpunkt(2)))*1.5
        MitteVertikal(length(MitteVertikal(:,1)) + 1,2) = ...
        FindeMitteVertikalRelSP(Image, Referenzpunkt + [1 0], f /Faktor);
        MitteVertikal(length(MitteVertikal(:,1)),1) =
            Referenzpunkt(1) +1;
        i=i+1;
        Referenzpunkt      = MitteVertikal(length(MitteVertikal(:,1)),:);
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% nach höchster Zeile im Marker suchen %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Vom Startpunkt aus gehend wird für den nächsten Bildpunkt über dem
% Referenzpunkt der Mittelpunkt der Zeile gesucht, falls der tiefer
% liegende Punkt einen Intensitätswert hat der unter 200 liegt und unter
% dem des Referenzpunkt mit dem Parameter f multipliziert.
    i=i+1;
    Referenzpunkt      = Startpunkt;
    while Image(Referenzpunkt(1) , Referenzpunkt(2) - 1)...
        < f * double(Image(Startpunkt(1) , Startpunkt(2))) &&...
            Image(Referenzpunkt(1) , Referenzpunkt(2) - 1)...
            < 200
        % Der Faktor wird für die Bestimmung des Mittelpunktes der
        % Zeile entsprechend dem Verhältnis zwischen aktuellem
        % Referenzpunkt zum Startpunkt angepasst.
        Faktor = double(Image(Referenzpunkt(1), Referenzpunkt(2) + 1))...
            / double(Image(Startpunkt(1),Startpunkt(2)))*1.5;
        MitteVertikal(length(MitteVertikal(:,1)) + 1,2) = ...
```

```

    FindeMitteVertikalRelSP(Image, Referenzpunkt - [1 0], f /
Faktor);
    MitteVertikal(length(MitteVertikal(:,1)),1) = Referenzpunkt(1) -
    i=i+1;
    Referenzpunkt      = MitteVertikal(length(MitteVertikal(:,1)),:);
end
% Die vertikale Mittelpunktskoordinate des Markers ergibt sich aus
% dem arithmetischen Mittelwert der größten und kleinsten
% y-Koordinate der gefundenen Mittelpunkte und wird für die Suche nach der
% horizontalen Mittelpunktskoordinate als Startkoordinate eingesetzt.
    MarkerMitteVertikal(1) = round(mean([max(MitteVertikal(:,1)),...
                                         min(MitteVertikal(:,1))]));
    MarkerMitteVertikal(2) = Startpunkt(2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Horizontal %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Der Referenzpunkt ist zu Anfang der eingegebene Startpunkt.
clear MitteHorizontal
MitteHorizontal(1,1) = FindeMitteHorizontalRelSP(Image,...
MarkerMitteVertikal, f);
MitteHorizontal(1,2) = MarkerMitteVertikal(1,2);
i=1;
Referenzpunkt      = MarkerMitteVertikal;

%%% nach äußerster rechts liegender Spalte innerhalb des Markers suchen %%
% Vom Startpunkt aus gehend wird für den nächsten Bildpunkt rechts vom
% Referenzpunkt der Mittelpunkt der Spalte gesucht, falls der rechts
liegende Punkt einen Intensitätswert hat der unter 200 liegt und unter dem
des Referenzpunkt mit dem Parameter f multipliziert.
while Image(Referenzpunkt(1) + 1 , Referenzpunkt(2))...
    < f * double(Image(Startpunkt(1) , Startpunkt(2))) &&...
    Image(Referenzpunkt(1) + 1 , Referenzpunkt(2))...
    < 200
    % Der Faktor wird für die Bestimmung des Mittelpunktes der
    % Zeile entsprechend dem Verhältnis zwischen aktuellem
    % Referenzpunkt zum Startpunkt angepasst.
    Faktor = double(Image(Referenzpunkt(1) + 1 , Referenzpunkt(2)))::
    / double(Image(Startpunkt(1),Startpunkt(2)))*1.5;
    MitteHorizontal(length(MitteHorizontal(:,1)) + 1,1) = ...
    FindeMitteHorizontalRelSP(Image, Referenzpunkt + [0 1], f / Faktor);
    MitteHorizontal(length(MitteHorizontal(:,1)),2) = Referenzpunkt(2)+1;
    i=i+1;
    Referenzpunkt = MitteHorizontal(length(MitteHorizontal(:,1)),:);
end
Referenzpunkt      = MarkerMitteVertikal;
k=1;

%%% nach äußerster links liegender Spalte innerhalb des Markers suchen %%
% Vom Startpunkt aus gehend wird für den nächsten Bildpunkt links vom
% Startpunkt der Mittelpunkt der Spalte gesucht, falls der links vom
Referenzpunkt
% liegende Punkt einen Intensitätswert hat der unter 200 liegt und unter
dem des
% Referenzpunkt mit dem Parameter f multipliziert.

while Image(Referenzpunkt(1) - 1, Referenzpunkt(2))...
    < f * double(Image(Startpunkt(1) , Startpunkt(2))) &&...
    Image(Referenzpunkt(1) - 1, Referenzpunkt(2))...
    < 200
    Faktor = double(Image(Referenzpunkt(1) - 1 , Referenzpunkt(2))) /...
    double(Image(Startpunkt(1),Startpunkt(2)))*1.5;
    MitteHorizontal(length(MitteHorizontal(:,1)) + 1,1) = ...
    FindeMitteHorizontalRelSP(Image, Referenzpunkt - [0 1], f / Faktor);

```

```

        MitteHorizontal(length(MitteHorizontal(:,1)),2) = Referenzpunkt(2)-1;
        k=k+1;
        i=i+1;
        Referenzpunkt = MitteHorizontal(length(MitteHorizontal(:,1)),:);
    end
% Die horizontale Mittelpunktskoordinate ergibt sich als Mittelwert aus den
% Koordinaten des linken und rechten Randes des Markers.
    MarkerMitteHorizontal = round(mean([max(MitteHorizontal(:,2)), ...
                                         min(MitteHorizontal(:,2))]));
% Die vertikale Mittelpunktskoordinate ergibt sich als Mittelwert aus den
% Koordinaten des oberen und unteren Randes des Markers.
    MarkerMitteVertikal = round(mean([max(MitteVertikal(:,1)), ...
                                       min(MitteVertikal(:,1))]));
% Mit der vertikalen und horizontalen Mittelpunktskoordinate ergibt sich
% der Mittelpunkt wie folgt.
    MarkerMitte=[round(MarkerMitteVertikal), round(MarkerMitteHorizontal)];
end

```


Vertikale Markermitten-Koordinate finden

```
function MitteVertikal = FindeMitteVertikalRelSP(Image, Startpunkt, f)
% Startpunkt ist in der Form X,Y einzugeben. Die Funktion sucht bei
% eingegebenem Startpunkt den Mittelpunkt der Zeile im übergebenen Bild.
Dabei werden die Grenzen des Markers entsprechend dem Parameter f sowie dem
Maximalwert von 200 gesucht.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FindeMitteVertikal %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Zeiger(1) = 0;
    Zeiger(2) = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% nach unten %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while double(Image(Startpunkt(1), Startpunkt(2) + Zeiger(1)))...
        < f * double(Image(Startpunkt(1), Startpunkt(2))) &&...
        double(Image(Startpunkt(1), Startpunkt(2) + Zeiger(1)))...
        < 200
        Zeiger(1) = Zeiger(1) + 1;
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% nach oben %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    while double(Image(Startpunkt(1), Startpunkt(2) - Zeiger(2)))...
        < f * double(Image(Startpunkt(1), Startpunkt(2))) &&...
        double(Image(Startpunkt(1), Startpunkt(2) - Zeiger(2)))...
        < 200
        Zeiger(2) = Zeiger(2) + 1;
    end
    MitteVertikal = round(Startpunkt(2) - (Zeiger(2) - (Zeiger(1)...
        + Zeiger(2))/2));
end
```

Horizontale Markermitten-Koordinate finden

```
function MitteHorizontal = FindeMitteHorizontalRelSP(Image, Startpunkt, f)

% Startpunkt ist in der Form X,Y einzugeben. Die Funktion sucht bei
% eingegebenem Startpunkt den Mittelpunkt der Spalte im übergebenen Bild.
Dabei werden die Grenzen des Markers entsprechend dem Parameter f sowie dem
Maximalwert von 200 gesucht.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FindeMitteHorizontal %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Zeiger(1) = 0;
    Zeiger(2) = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% rechts %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while double(Image(Startpunkt(1) + Zeiger(1), Startpunkt(2)))...
        < f * double(Image(Startpunkt(1), Startpunkt(2))) &&...
        double(Image(Startpunkt(1) + Zeiger(1), Startpunkt(2)))...
        < 200
        Zeiger(1) = Zeiger(1) + 1;
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% links %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while double(Image(Startpunkt(1) - Zeiger(2), Startpunkt(2)))...
        < f * double(Image(Startpunkt(1), Startpunkt(2))) &&...
        double(Image(Startpunkt(1) - Zeiger(2), Startpunkt(2)))...
        < 200
        Zeiger(2) = Zeiger(2) + 1;
    end
    MitteHorizontal = round(Startpunkt(1) - (Zeiger(2) - (Zeiger(1)+...
        Zeiger(2))/2));
end
```

Hauptprogramm Tracking von weißen Heckklappen-Marker

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Basler_Oben_Marker_Tracking %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% für weißen Heckklappen-Marker %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Die Aufnahme wird Bild für Bild durchlaufen und dabei die lokalen
% Verschiebung von Ecken und Kanten von einem Bild zum nächsten betrachtet.

% Kameraparameter aus der Kalibrierung der hier eingesetzten Kamera
% werden geladen, um die Bilder zu entzerren.
load('BaslerWeitwinkelParams.mat');
%%
Markerverschiebung      = [100 -50];
% Images, welche in einem Array gespeichert sind, werden geladen.
load('Images.mat');
MarkerInitPos          = zeros(1,2);
Marker_ImagePos        = zeros(size(Images, ndims(Images)),2);

for i = 1 : (size(Images, ndims(Images))-1)
    I1                = undistortImage(Images(:,:,i),cameraParamsWeitwinkel);
    I2                = undistortImage(Images(:,:,i + 1),cameraParamsWeitwinkel);
    % Falls noch keine Initiale Markerposition aufgenommen wurde (für
    % den Fall das dieser erst später in der Sequenz im Bild
    % erscheint), soll dieser per Maus eingegeben werden.
    if MarkerInitPos(1,:) == [0, 0]
        imagesc(I1);
        w = waitforbuttonpress;
        if w == 0 % Bei Klicken Markerposition aufnehmen
            MarkerInitPos(1,:) = ginput(1);
        end
    end
    % Die eingegbene Position des Markers wird durch die Funktion zum
    % finden der Markermittle korrigiert und in das entsprechende Array
    % als erste Markerposition in der Bildsequenz aufgenommen.
    Marker_ImagePos(i,:) = ...
        MarkerMittefindenV3(I1, [round(MarkerInitPos(2)), ...
                                round(MarkerInitPos(1))], 0.14);
    else % Bei Tastendruck zum nächsten Bild gehen
        continue
    end
end
range = [Marker_ImagePos(i,2) - 150 , ...
        Marker_ImagePos(i,2) + 250; ... % Horizontal
        Marker_ImagePos(i,1) - 100 , ...
        Marker_ImagePos(i,1) + 180]; % Vertikal
range(range<0) = 1;
range(range>1920) = 1920;

Markerverschiebung      = (verschiebung(range, I1, I2));
    if Marker_ImagePos(i,2) > 750 && Marker_ImagePos(i,2) < 1400
        Markerverschiebung      = Markerverschiebung * 5 / 6;
    end
imagesc(I2);
Marker_ImagePos(i + 1,:) = round(Marker_ImagePos(i,:) + ...
                                fliplr(Markerverschiebung));
plot(Marker_ImagePos(i + 1,2),Marker_ImagePos(i + 1,1),'o');
Marker_ImagePos(i + 1,:) = ...
MarkerMittefindenV3(I2, [Marker_ImagePos(i + 1,1), ...
                        Marker_ImagePos(i + 1,2)], 0.14);
k = find(Marker_ImagePos(:,1),1);
end
```

Berechnung der Bild-Verschiebung

```
function [vector]=verschiebung(range, I1, I2)
% Der Funktion werden 2 Images sowie über die Variable range Koordinaten
% für den zu untersuchenden Bildbereich übergeben. Für Diesen bildbereich
% werden Eckpunkte in beiden Bildern gesucht.
    if size(I1,3)>1
        I1=rgb2gray(I1);
    end
    if size(I2,3)>1
        I2=rgb2gray(I2);
    end

    points1 =
detectSURFFeatures(I1(range(2,1):range(2,2),range(1,1):range(1,2),:));
    points2 =
detectSURFFeatures(I2(range(2,1):range(2,2),range(1,1):range(1,2),:));

    [features1,valid_points1] = extractFeatures(I1(range(2,1):...
        range(2,2),range(1,1):range(1,2),:),points1);
    [features2,valid_points2] = extractFeatures(I2(range(2,1):...
        range(2,2),range(1,1):range(1,2),:),points2);
% Die gefundenen Punkte werden verglichen und korrespondierende Punkte
% gesucht. Für diese werden die Verschiebungen berechnet.
    indexPairs = matchFeatures(features1,features2);
    matchedPoints1 = valid_points1(indexPairs(:,1),:);
    matchedPoints2 = valid_points2(indexPairs(:,2),:);
    figure
    showMatchedFeatures(I1(range(2,1):range(2,2),range(1,1):...
        range(1,2),:),I2(range(2,1):...
        range(2,2),range(1,1):range(1,2),:),...
        matchedPoints1,matchedPoints2);
    if isempty(matchedPoints1) == 1
        range(1,2) = range(1,2) + 130 ;
        range(1,2) = range(1,1) - 130 ;
        vector=verschiebung(range, I1, I2);
    else
        matchedPoints1.Location(:,1)=
matchedPoints1.Location(:,1)+range(1,1);
        matchedPoints1.Location(:,2)=
matchedPoints1.Location(:,2)+range(2,1);
        matchedPoints2.Location(:,1)=
matchedPoints2.Location(:,1)+range(1,1);
        matchedPoints2.Location(:,2)=
matchedPoints2.Location(:,2)+range(2,1);
        verschiebungen = matchedPoints2.Location-matchedPoints1.Location;
% Mit der Funktion "isoutlier" werden Ausreisser detektiert und
aussortiert.
        TF = isoutlier(verschiebungen);
        verschiebungen = verschiebungen - ...
            (abs(TF(:,2)-TF(:,1))+ ...
            TF(:,2)).*verschiebungen;
        verschiebungen(~any(verschiebungen,2), :) = [];
        vector = [median(verschiebungen(:,1)),...
            median(verschiebungen(:,2))];
    end
end
hold on
figure;
showMatchedFeatures(I1(range(2,1):range(2,2),range(1,1):...
    range(1,2),:),I2(range(2,1):range(2,2),range(1,1):...
    range(1,2),:),matchedPoints1,matchedPoints2);
```

Marker-Mitte finden in weißer Heck-Markierung

```
function MarkerMitte = MarkerMittefindenV3(Image, Startpunkt, f)
    % Koordinaten in der Form [Y, X] einzugeben!
    % Es wird nach den Markergrenzflächen gesucht und aus den Koordinaten
    % wird die Mitte des Markers errechnet.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Vertikal %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear MitteVertikal
MitteVertikal(1,1) = FindeMitteVertikal(Image, Startpunkt, f);
MitteVertikal(1,2) = Startpunkt(2);
i=1;
Referenzpunkt      = Startpunkt;
while abs(double(Image(Referenzpunkt(1) , Referenzpunkt(2) + 1)) - ...
          double(Image(Referenzpunkt(1) , Referenzpunkt(2))))...
        < f * double(Image(Referenzpunkt(1),Referenzpunkt(2)))

    MitteVertikal(length(MitteVertikal(:,1)) + 1,1) = ...
    FindeMitteVertikal(Image, ...
    [MitteVertikal(length(MitteVertikal(:,1)),1),...
    MitteVertikal(length(MitteVertikal(:,1)),2) + 1], f);
    MitteVertikal(length(MitteVertikal(:,1)),2) = Referenzpunkt(2)+1;
    i=i+1;
    Referenzpunkt      = MitteVertikal(length(MitteVertikal(:,1)),:);
end
i=i+1;
Referenzpunkt      = Startpunkt;

while abs(double(Image(Referenzpunkt(1) , Referenzpunkt(2) - 1)) - ...
          double(Image(Referenzpunkt(1) , Referenzpunkt(2))))...
        < f * double(Image(Referenzpunkt(1) , Referenzpunkt(2)))

    MitteVertikal(length(MitteVertikal(:,1)) + 1,1) = ...
    FindeMitteVertikal(Image, [Referenzpunkt(1), Referenzpunkt(2)], f);
    MitteVertikal(length(MitteVertikal(:,1)),2) = Referenzpunkt(2)-1;
    i=i+1;
    Referenzpunkt      = MitteVertikal(length(MitteVertikal(:,1)),:);
end
EndPunktVertikal      = MitteVertikal(length(MitteVertikal(1,:)),:);
MarkerMitteVertikal(1) = round(mean([max(MitteVertikal(:,1)),...
                                     min(MitteVertikal(:,1))]););
MarkerMitteVertikal(2) = Startpunkt(2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Horizontal %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear MitteHorizontal
MitteHorizontal(1,2) =FindeMitteHorizontal(Image,MarkerMitteVertikal,f);
MitteHorizontal(1,1) = MarkerMitteVertikal(1,1);
i=1;
Referenzpunkt      = MarkerMitteVertikal;
while abs(double(Image(Referenzpunkt(1) + 1 , Referenzpunkt(2))) - ...
          double(Image(Referenzpunkt(1) , Referenzpunkt(2))))...
        < f * double(Image(Referenzpunkt(1) , Referenzpunkt(2)))

    MitteHorizontal(length(MitteHorizontal(:,1))+1,2) = ...
    FindeMitteHorizontal(Image,...
    [MitteHorizontal(length(MitteHorizontal(:,1)),1),...
    MitteHorizontal(length(MitteHorizontal(:,1)),2)], f);
end
```

```

        MitteHorizontal(length(MitteHorizontal(:,1)),1) =...
        MarkerMitteVertikal(1) + i;
        i=i+1;
        Referenzpunkt = MitteHorizontal(length(MitteHorizontal(:,1)),:);
    end

Referenzpunkt          = MarkerMitteVertikal;
k=1;
    while abs(double(Image(Referenzpunkt(1) , Referenzpunkt(2))) - ...
        double(Image(Referenzpunkt(1) - 1 , Referenzpunkt(2))))...
        < f * double(Image(Referenzpunkt(1) , Referenzpunkt(2)))

        MitteHorizontal(length(MitteHorizontal(:,1)) + 1,2) = ...
        FindeMitteHorizontal(Image, [Referenzpunkt(1) - 1 ,
Referenzpunkt(2)], f);
        MitteHorizontal(length(MitteHorizontal(:,1)),1) = Startpunkt(1) - k;
        k=k+1;
        i=i+1;
        Referenzpunkt = MitteHorizontal(length(MitteHorizontal(:,1)),:);
    end
        Referenzpunkt          = EndPunktVertikal;
        i=1;
    while abs(double(Image(Referenzpunkt(1) + 1 , Referenzpunkt(2))) - ...
        double(Image(Referenzpunkt(1) , Referenzpunkt(2))))...
        < f * double(Image(Referenzpunkt(1) , Referenzpunkt(2)))
        MitteHorizontal(length(MitteHorizontal(:,1))+1,2) = ...
        FindeMitteHorizontal(Image, [Referenzpunkt(1) - 1 ,
Referenzpunkt(2)], f);
        MitteHorizontal(length(MitteHorizontal(:,1)),1) = EndPunktVertikal(1)
+ i;
        i=i+1;
        Referenzpunkt = MitteHorizontal(length(MitteHorizontal(:,1)),:);
    end

        Referenzpunkt          = EndPunktVertikal;
        k=1;

    while abs(double(Image(Referenzpunkt(1) - 1 , Referenzpunkt(2))) - ...
        double(Image(Referenzpunkt(1) , Referenzpunkt(2))))...
        < f * double(Image(Referenzpunkt(1) , Referenzpunkt(2)))

        MitteHorizontal(length(MitteHorizontal(:,1))+1,2) = ...
        FindeMitteHorizontal(Image, [Referenzpunkt(1) - 1 ,
Referenzpunkt(2)], f);
        MitteHorizontal(length(MitteHorizontal(:,1)),1) = EndPunktVertikal(1)
- k;
        k=k+1;
        i=i+1;
        Referenzpunkt = MitteHorizontal(length(MitteHorizontal(:,1)),:);
    end
MarkerMitteHorizontal = round(mean([max(MitteVertikal(:,2)),...
        min(MitteVertikal(:,2))]));
MarkerMitteVertikal = round(mean([max(MitteHorizontal(:,1)),...
        min(MitteHorizontal(:,1))]));
MarkerMitte=[round(MarkerMitteVertikal) , round(MarkerMitteHorizontal)];
end

```

Vertikale Markermitten-Koordinate finden

```
function MitteVertikal = FindeMitteVertikal(Image, Startpunkt, f)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  FindeMitteVertikal  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Zeiger(1) = 0;
Zeiger(2) = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  nach unten  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while abs(double(Image(Startpunkt(1) + Zeiger(1), Startpunkt(2))) - ...
    double(Image(Startpunkt(1) + Zeiger(1) + 1, Startpunkt(2))))...
    < f * double(Image(Startpunkt(1) + Zeiger(1), Startpunkt(2)))
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  nach oben  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while abs(double(Image(Startpunkt(1) - Zeiger(2), Startpunkt(2))) - ...
    double(Image(Startpunkt(1) - Zeiger(2) - 1, Startpunkt(2))))...
    < f * double(Image(Startpunkt(1) - Zeiger(2), Startpunkt(2)))
    Zeiger(2) = Zeiger(2) + 1;
end
MitteVertikal = round(Startpunkt(1) - ....
    (Zeiger(2) - (Zeiger(1) + Zeiger(2))/2));
end
```

Horizontale Markermitten-Koordinate finden

```
function MitteHorizontal = FindeMitteHorizontal(Image, Startpunkt, f)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  FindeMitteHorizontal  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Zeiger(1) = 0;
    Zeiger(2) = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  rechts  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while abs(double(Image(Startpunkt(1), Startpunkt(2) + Zeiger(1))) - ...
    double(Image(Startpunkt(1), Startpunkt(2) + Zeiger(1) + 1)))...
    < f * double(Image(Startpunkt(1), Startpunkt(2) + Zeiger(1)))
    Zeiger(1) = Zeiger(1) + 1;
end
plot(Startpunkt(1) + Zeiger + s ,Startpunkt(2), '*');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  links  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while abs(double(Image(Startpunkt(1), Startpunkt(2) - Zeiger(2))) - ...
    double(Image(Startpunkt(1), Startpunkt(2) - Zeiger(2) - 1)))...
    < f * double(Image(Startpunkt(1), Startpunkt(2) - Zeiger(2)))
    Zeiger(2) = Zeiger(2) + 1;
end
MitteHorizontal = round(Startpunkt(2) - ...
    (Zeiger(2) - (Zeiger(1) + Zeiger(2))/2));
end
```

Transformation von Bildkoordinaten in Raumkoordinaten

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Hauptprogramm %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Transformation Vogelperspektivische Aufnahmen in Weltkoordinaten %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%s%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Relevante Parameter für die Transformation von Bildkoordinaten in
% Raumkoordinaten und zur Berechnung von Abständen werden hier eingegeben.
    h_marker           = 760 + 100;
    h_Frontmarker      = 995;
    h_Heckmarker       = 1330;
    alpha_k            = 32.46;
    beta_k             = -3.3;

% Kalibrierungsparameter, die Bildpunktanzahl der Zeilen und Spalten des
% Bildes werden sowie das Protokoll der Messreihe werden geladen.
    load('BaslerWeitwinkelParams.mat')
    load('BildpunktzahlWeitwinkel.mat')
    Protokoll=xlsread('Protokoll.xlsx');
%% Eine Schleife wird entsprechend der Anzahl der Messungen durchlaufen.
% Zuvor werden Ordner mit dem Index der Messung als Namen angelegt und die
% Bildpositionen der Marker, welche durch den Tracking-Algorithmus
% berechnet sind dort abgelegt. Diese tragen die Dateinamen
%"Marker_ImagePos_Heck.mat" und "Marker_ImagePos_Front.mat". Nachdem die
% Position des Heckmarkers transformiert ist, werden alle Messungen noch
% ein mal für den Frontmarker durchlaufen.

for i= 1:46
    FileName           =      int2str(i);
    oldfolder          =      cd(FileName);
    if exist('Marker_ImagePos_Heck.mat') == 2
        load('Marker_ImagePos_Heck.mat')
        load('Images.mat')
% Zur Reduzierung der Rechenzeit beim Beschreiben der Variablen
% wird für diese Speicher entsprechend der Anzahl an Aufnahmen
% der Messung vor dem Beschreiben belegt.
        clear MarkerPosGlob
        clear MarkerPosRelCam
        MarkerPosRelCam =      zeros(length(Marker_ImagePos),2);
        MarkerPosRelCam =      zeros(length(Marker_ImagePos),2);
        %%
        start=1;
        for k = 1 : length(Marker_ImagePos)
% Zunächst wird für das betrachtete Bild die Position von Marker 2
% relativ zur Kamera bestimmt. Für Aufnahmen, bei denen dieser in keinem
% Bild verdeckt ist, wird diese Anweisung für alle Bilder ausgeführt. Die
% Bedingung k==1 passt dies für Aufnahmen, bei denen dies nicht der Fall
% ist an, sodass die Anweisung nur für das erste Bild der Sequenz
% ausgeführt wird.
            Image           =      undistortImage(Images(:, :, k), ...
                                                cameraParamsWeitwinkel);
            if k==1
                Marker2ImagePos =      MarkerMittefindenV4(Image, ...
                                                            [117 1245], 4);
            end

% Die x- und y- Koordinate des Markers als Ursprung des
% Globale Koordinatensystems relativ zur Kamera entspricht den
% Welt-Koordinaten der Kamera. Dabei beträgt die Höhe der Markermittle mit
% der Gehweghöhe 860mm.
                [CamPosGlob(1), CamPosGlob(2)] = ...
```

```

        Basler_KoordzuRaumposV4(fliplr(Marker2ImagePos), ...
        alpha_k, beta_k, 860, cameraParamsWeitwinkel, Bildpunktzahl);

% Die Bildpositionen der Marker werden in x- und y-Raumkoordinaten relativ
% zur Kamera umgewandelt. Für die Transformation eines Markers mit einer
% anderen Höhe, als hier dem Heckmarker ist der Parameter anzupassen
    [MarkerPosRelCam(k,1), MarkerPosRelCam(k,2)] = ...
    Basler_KoordzuRaumposV4([Marker_ImagePos(k, 2),...
    Marker_ImagePos(k, 1)], alpha_k, beta_k, h_Heckmarker, ...
    cameraParamsWeitwinkel, Bildpunktzahl);

% Mit den x- und y-Welt-Koordinaten der Kamera wird die Position des
% Markers relativ zur Kamera in Welt-Koordinaten umgerechnet.
    MarkerPosGlob(k,1) =...
    CamPosGlob(1) - MarkerPosRelCam(k,1);
    MarkerPosGlob(k,2) =...
    CamPosGlob(2) - MarkerPosRelCam(k,2);
end
else
    continue
end
end
end

```

Koordinatentransformation von Bild- in Raumkoordinaten

```

function [x_rel_cam, y_rel_cam] =...
Basler_KoordzuRaumposV4(Koord, alpha_k, beta_k, h_marker, ...
    cameraParams, Bildpunktzahl)

% Ein Image wird eingelesen. Dieses ist von einer Kamera aufgenommen,
% welche um alpha_k (Drehung um die x-Achse) geneigt ist. Dabei ist der
% kleinste Winkel zwischen der z-Achse der Kamera der z-Achse der
% Fahrbahnebene einzugeben.
% Außerdem ist die Höhendifferenz zwischen Kamera und betrachtetem
% Punkt einzugeben, welche sich aus der Differenz h_cam - h_marker ergibt.
% Diese Punkte werden auf ein Ziel-Koordinatensystem projiziert, welches
% seinen Ursprung auf der Fahrbahnebene unter der Kamera hat, deren z-Achse
% orthogonal zur Straße in Richtung Kamera zeigt sowie y-Achse und x-Achse
% entgegen der entsprechenden Achsen des Globale Koordinatensystems
% gerichtet sind (Ortsmarker 1/2).

% Hier wird die Höhe der Kamera in Relation zur Fahrbahn eingegeben.
% Bei der ersten Messreihe beträgt die Höhe der Kamera 8650mm und
% im zweiten Entwurf 6314mm.
    h_cam = 4634 + 1780 - 150 + 50;

% Höhendifferenz zwischen Kamera und Marker
    h_km = h_cam - h_marker;

% Öffnungswinkel der Kamera errechnet über die Linsengleichung mit den
% Abmessungen des Sensors.
    cam_Winkel_V = 61.222 ; % mit Kowa LM12HC Objektiv 31.7;
    cam_Winkel_H = 86.558 ; % mit Kowa LM12HC Objektiv 48.6;

% Auflösung der betrachteten Zeile, Spalte wird aus der Matrix
% Bildpunktzahl ausgelesen.
    cam_Res_V = Bildpunktzahl(1,Koord(1));
    cam_Res_H = Bildpunktzahl(2,Koord(2));

% Relativer y-Abstand zum optischen Zentrum der Kamera (im Bild)

```



```

delta_y_p          = cameraParams.PrincipalPoint(2) - ...
                    Koord(2);

% Relativer x-Abstand zum optischen Zentrum der Kamera (im Bild)
delta_x_p          = Koord(1) - ...
                    cameraParams.PrincipalPoint(1);

% Relativwinkel zwischen dem betrachteten Punkt und optischer Achse
% bei Rotation um die x-Achse. Der Drehsinn entspricht dabei der rechten
% Hand Regel, sodass Punkte oberhalb des optischen Zentrums positive Winkel
% ergeben.
phi_km             = ( delta_y_p * cam_Winkel_V / cam_Res_V );

% Relativwinkel zwischen dem betrachteten Punkt und optischer Achse
% bei Rotation um die y-Achse. Der Drehsinn ist so gewählt, dass Punkte
% rechts vom optischen Zentrum Positive Winkel ergeben (linke Hand Regel).
tau_km             = ( delta_x_p * cam_Winkel_H / cam_Res_H );

% Aus der Höhe und dem Neigungswinkel der Kamera wird mit dem
% Relativwinkel phi_km für den betrachteten Punkt der Abstand entlang der
auf
% die Fahrbahn projizierten Bild-Vertikalen im Raum ermittelt.
d_km               = h_km * tand ( alpha_k + phi_km );

% Mit d_km und der Summe der Winkel alpha_k und phi_km wird der senkrechte
% Abstand l_km zwischen der Kamerabildhorizontalen und der in den
% betrachteten Punkt projizierten Kamerabildhorizontalen berechnet.
l_km               = d_km / sind(alpha_k + phi_km);

% Mit dem bekannten Relativwinkel tau_km und dem Abstand
% l_km wird der zu l_km senkrechte Abstand b_km zum betrachteten Punkt
% berechnet.
b_km               = l_km * tand(tau_km);

% Die berechneten Relativabstände d_km und b_km werden mit dem
% Verdrehwinkel beta_k in das Ziel-Koordinatensystem transformiert.
y_rel_cam         = cosd(beta_k) * d_km - (b_km * sind(beta_k));
x_rel_cam         = b_km * cosd(beta_k) + d_km * sind(beta_k);

end

```

10. Literaturverzeichnis

- AG, Basler. o. J. „Basler ace acA1920-40um - Flächenkamera“. Basler AG. Zugegriffen 28. Juli 2018. <https://de/produkte/kameras/flaechenkameras/ace/aca1920-40um/>.
- „Apogee-D: High Accuracy Dual Antenna GNSS/INS“. o. J. Zugegriffen 28. Juli 2018. <https://www.sbg-systems.com/products/apogee-d-dual-antenna-gnss-ins>.
- „BAST 2017 - Forschung kompakt“. 2017. 2017. https://www.bast.de/BAST_2017/DE/Publikationen/Foko/2013-2012/2012-11.html.
- Bühler, Peter, Patrick Schlaich, und Dominik Sinner. 2017. *Digitale Fotografie: fotografische Gestaltung - Optik -ameratechnik*. Bibliothek der Mediengestaltung. Berlin: Springer Vieweg.
- Burgard, Wolfram, Cyrill Stachniss, Kai Arras, und Maren Bennewitz. o. J. „Introduction to Mobile Robotics“.
- Canny, J. 1986. „A Computational Approach to Edge Detection“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (6): 679–98. <https://doi.org/10.1109/TPAMI.1986.4767851>.
- „Convert RGB image or colormap to grayscale - MATLAB rgb2gray - MathWorks Deutschland“. o. J. Zugegriffen 24. Juli 2018. <https://de.mathworks.com/help/matlab/ref/rgb2gray.html>.
- Deans, Matthew Charles. 2005. „Bearings-Only Localization and Mapping“. Pittsburgh, PA 15213: School of Computer Science Carnegie Mellon University.
- „Detect corners using Harris–Stephens algorithm and return cornerPoints object - MATLAB detectHarrisFeatures - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. https://de.mathworks.com/help/vision/ref/detectharrisfeatures.html?s_tid=doc_ta.
- „Detect SURF features and return SURFPoints object - MATLAB detectSURFFeatures - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. <https://de.mathworks.com/help/vision/ref/detectsurffeatures.html>.
- „Extract interest point descriptors - MATLAB extractFeatures - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. https://de.mathworks.com/help/vision/ref/extractfeatures.html?s_tid=doc_ta.
- „Find matching features - MATLAB matchFeatures - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. https://de.mathworks.com/help/vision/ref/matchfeatures.html?s_tid=doc_ta.
- „Find outliers in data - MATLAB isoutlier - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. https://de.mathworks.com/help/matlab/ref/isoutlier.html?s_tid=doc_ta.
- Gockel, Tilo. 2012. *Kompendium digitale Fotografie*. X.media.press. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-11239-3>.
- Hahn, Dr. Markus. 2016. „SLAM technologies for parking applications part II: A showcase AdaptIVe“. Technical Workshop gehalten auf der SLAM technologies for parking applications, Athen, April 22.
- Hirzer, Martin. 2008. „Marker Detection for Augmented Reality Applications“, Oktober, 27.
- „Inventor 2018“. 2018. 2018. <http://help.autodesk.com/view/INVNTOR/2018/ENU/>.
- Jähne, Bernd. 2012. *Digitale Bildverarbeitung und Bildgewinnung*. 7., neu Bearb. Aufl. Berlin: Springer Vieweg.
- Kannala, Juho, und Sami S Brandt. o. J. „A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses“, 15.
- Keferstein, Claus P., Michael Marxer, und Carlo Bach. 2018. *Fertigungsmesstechnik: alles zu Messunsicherheit, konventioneller Messtechnik und Multisensorik*. 9., überarbeitete und erweiterte Auflage. Lehrbuch. Wiesbaden: Springer Vieweg.
- „Ladybug5+ USB3“. o. J. Zugegriffen 28. Juli 2018. <https://www.ptgrey.com/ladybug5plus>.
- Lemaire, T., S. Lacroix, und J. Sola. 2005. „A Practical 3D Bearing-Only SLAM Algorithm“. In , 2449–54. IEEE. <https://doi.org/10.1109/IROS.2005.1545393>.
- „LM6HC | Produkte | Kowa Optimed Deutschland GmbH“. o. J. Zugegriffen 28. Juli 2018. <https://www.kowa-lenses.com/produkte/303/lm6hc>.

- „LM12HC | Produkte | Kowa Optimed Deutschland GmbH“. o. J. Zugegriffen 28. Juli 2018. <https://www.kowa-lenses.com/produkte/305/lm12hc>.
- „MathWorks – Entwickler von MATLAB und Simulink“. 2018. 2018. <https://de.mathworks.com/>.
- Maurer, Markus, Hermann Winner, Barbara Lenz, und J. Christian Gerdes, Hrsg. 2015. *Autonomes Fahren: technische, rechtliche und gesellschaftliche Aspekte*. Springer Open. Berlin: Springer Vieweg.
- „Maximum elements of an array - MATLAB max - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. https://de.mathworks.com/help/matlab/ref/max.html?searchHighlight=max&s_tid=doc_srchtile.
- „Measuring Planar Objects with a Calibrated Camera - MATLAB & Simulink - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. <https://de.mathworks.com/help/vision/examples/measuring-planar-objects-with-a-calibrated-camera.html>.
- „Minimum elements of an array - MATLAB min - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. https://de.mathworks.com/help/matlab/ref/min.html?searchHighlight=min&s_tid=doc_srchtile.
- Muthugnanambika, M., und S. Padmavathi. 2017. „Feature detection for color images using SURF“. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1–4. <https://doi.org/10.1109/ICACCS.2017.8014572>.
- Negenborn, Rudy. 2013. „Robot Localization and Kalman Filters on finding your position in a noisy world“. Thesis, UTRECHT UNIVERSITY. <https://svs.informatik.uni-hamburg.de/publications/2016/2016-01-15-Herrmann-Machine-Learning-autonomes-Fahrzeug.pdf>.
- Papula, Lothar, und Lothar Papula. 2016. *Vektoranalysis, Wahrscheinlichkeitsrechnung, Mathematische Statistik, Fehler- und Ausgleichsrechnung: mit 550 Abbildungen, zahlreichen Beispielen aus Naturwissenschaft und Technik sowie 295 Übungsaufgaben mit ausführlichen Lösungen*. 7., überarbeitete und erweiterte Auflage. Mathematik für Ingenieure und Naturwissenschaftler, Lothar Papula ; Band 3. Wiesbaden: Springer Vieweg.
- Phan, Khoa D, und Aleksandr V Ovchinnikov. 2013. „Indoor Slam Using an Omnidirectional Camera“, 7.
- „PLR 25 | Laser-Entfernungsmesser | Digitale Messwerkzeuge | Heimwerker“. o. J. Bosch Elektrowerkzeuge für Heimwerker. Zugegriffen 28. Juli 2018. <https://www.bosch-do-it.de/de/de/bosch-elektrowerkzeuge/werkzeuge/plr-25-3165140847308-199929.jsp>.
- Renz, Sebastian. 2013. „Einparkassistenten: 10 Systeme im Test“. *auto motor und sport*. 12. März 2013. <https://www.auto-motor-und-sport.de/test/einparkassistenten-10-systeme-im-test/>.
- Rosten, E., und T. Drummond. 2005. „Fusing Points and Lines for High Performance Tracking“. In , 1508-1515 Vol. 2. IEEE. <https://doi.org/10.1109/ICCV.2005.104>.
- „Round to nearest decimal or integer - MATLAB round - MathWorks Deutschland“. o. J. Zugegriffen 28. Juli 2018. https://de.mathworks.com/help/matlab/ref/round.html?s_tid=doc_ta.
- „SAE Levels of Driving Automation“. o. J. Zugegriffen 24. Juli 2018. </blog/2013/12/sae-levels-driving-automation>.
- Scott, David W. 1979. „On Optimal and Data-Based Histograms“. *Biometrika* 66 (3): 605. <https://doi.org/10.2307/2335182>.
- „Shape Recognition - MATLAB Central“. o. J. Zugegriffen 27. Juli 2018. <https://de.mathworks.com/matlabcentral/fileexchange/15491>.
- Sola, Joan. 2014. „Simultaneous Localization and Mapping with the Extended Kalman filter“, Oktober, 35.
- Strasdat, Hauke, Cyrill Stachniss, Maren Bennewitz, und Wolfram Burgard. 2007. „Visual Bearing-Only Simultaneous Localization and Mapping with Improved Feature Matching“. In *Autonome Mobile Systeme 2007*, herausgegeben von Karsten Berns und

- Tobias Luksch, 15–21. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74764-2_3.
- Tay, Christopher. 2007. „Car Park Mapping with Simultaneous Localisation and Mapping (SLAM)“, September, 52.
- Tipler, Paul Allen, und Gene Mosca. 2015. *Physik für Wissenschaftler und Ingenieure*. Herausgegeben von Jennifer Wagner. Übersetzt von Michael Basler, Renate Dohmen, Carsten Heinisch, Anna Schleitzer, und Michael Zillgitt. 7. deutsche Auflage. Lehrbuch. Berlin Heidelberg: Springer Spektrum.
- Torre, V., und T. A. Poggio. 1986. „On Edge Detection“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (2): 147–63. <https://doi.org/10.1109/TPAMI.1986.4767769>.
- „VDA“. o. J. Zugegriffen 24. Juli 2018. <http://vda.de>.
- „Velodyne LiDAR“. o. J. Zugegriffen 28. Juli 2018. <http://www.velodynelidar.com/>.
- „Vollautomatisches, autonomes Einparken“. o. J. Zugegriffen 14. Juli 2018. <https://www.automobil-industrie.vogel.de/vollautomatisches-autonomes-einparken-a-367071/>.
- Winner, Prof. Dr. rer. nat. Hermann, Dipl.-Ing. Stephan Hakuli, und Dipl.-Ing. Felix Lotz. 2015. „Handbuch Fahrerassistenzsysteme Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort 3., überarbeitete und ergänzte Auflage“. Herausgegeben von Christina Singer M. Sc. Springer Vieweg.
- Zhang, Renyan, Guoling Zhao, und Li Su. 2005. „A new edge detection method in image processing“. In *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005.*, 1:445–48. <https://doi.org/10.1109/ISCIT.2005.1566889>.
- Zhao, Guanyi. 2015. „Fusion of Ladybug3 Omnidirectional Camera and Velodyne Lidar“. Masterthesis, Stockholm: KTH Royal Institute of Technology. <http://www.diva-portal.se/smash/get/diva2:848039/FULLTEXT01.pdf>.