

# Automatisierte Auswertung von Naturalistic-Cycling-Versuchsdaten

vorgelegt von  
M.Sc.  
Benjamin Rippel

an der Fakultät V - Verkehrs- und Maschinensysteme  
der Technischen Universität Berlin  
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften  
- Dr.-Ing. -

Promotionsausschuss:

Vorsitzende: Prof. Dr. Christina Völlmecke

Gutachterin: Prof. Dr. rer. nat. Stefanie Marker

Gutachter: Prof. Dr.-Ing. Borislav Hristov

Tag der wissenschaftlichen Aussprache: 30.11.2020

Berlin 2021



---

# Danksagung

Die vorliegende Dissertation resultiert aus meiner Arbeit am Fachgebiet Fahrerverhaltensbeobachtung für energetische Optimierung und Unfallvermeidung der Technischen Universität Berlin. Als begeisterter Radfahrer war es naheliegend, ein Dissertationsthema zu wählen, welches mein Hobby in den beruflichen Alltag integriert und dennoch mein Interesse an modernen Methoden der Umfelderkennung bedient. Diese Motivation hat mich die letzten Jahre begleitet und mir den nötigen Antrieb gegeben, nach meinem Ausscheiden aus dem Fachgebiet FVB die Arbeit in meiner Freizeit fertigzustellen. Auf diesem Weg haben mich viele Menschen begleitet.

Mein Dank gilt Prof. Dr. Christina Völlmecke, die als Vorsitzende des Prüfungsausschusses die wissenschaftliche Aussprache unter den besonderen Corona-Auflagen ermöglicht hat. Ebenso möchte ich Prof. Dr. Borislav Hristov für die Begutachtung der Dissertation und die hilfreichen Anmerkungen danken.

Ganz besonders möchte ich mich bei meiner Doktormutter Prof. Dr. Stefanie Marker bedanken. Während meiner Zeit am Fachgebiet FVB konnte ich mich auf die stets guten Ratschläge verlassen, egal ob fachlicher oder privater Natur. Auch nach meinem Umzug an den Bodensee war der Kontakt immer herzlich und ein für den Abschluss essenzieller Bestandteil. Vielen Dank!

Der Unfallanalyse Berlin, besonders Daniel Müller, danke ich für die Unterstützung bei den Versuchen auf der Crashbahn und die Bereitstellung des Materials.

Natürlich gilt mein Dank auch den vielen Freunden und Kollegen in Berlin. Den technischen Mitarbeitern des Fachgebiets, die mich am Versuchsträger unterstützten, aber auch Frau Ipta im Sekretariat für die Hilfe bei allen bürokratischen Belangen. Ich danke meinen Freunden und studentischen Mitarbeitern Tom und Kilian Köppchen für die Hilfe bei der Umsetzung vieler Ideen, auch wenn sich einige davon als Sackgassen entpuppten. Ohne die vielen wissenschaftlichen Mitarbeiter der Fachgebiete KFZB und FVB wäre das Leben an der Universität nur halb so schön gewesen. Insbesondere Jan Grüner, Dr. Peipei Liu, Jens Jerratsch, Dr. Gerd Müller, Johannes Holtz, Tobias Trabert, Dr. Torsten Dunger und Danny Klinder stechen dabei heraus. Danke für die gute Laune, die Grillabende, die Entdeckungstouren durch das TIB und das Hochhalten der Moral.

Ich danke auch meinen Kollegen der ZF Friedrichshafen AG, Dr. Edoardo Casapietra, Dr. Frasher Loshaj und Dr. Michael Walter für die vielen Fachgespräche in der Kaffeeküche und die hilfreichen Tipps. Ebenso möchte ich meinem Teamleiter Dr. Benjamin Wilking für die moralische Unterstützung danken.

Schlussendlich bedanke ich mich ganz besonders bei meiner Familie für die Unterstützung über all die Jahre. Meine Frau Bianca und meine Tochter Alva haben mir immer Halt gegeben und mich zum Weitermachen ermutigt, auch wenn die Moral am Boden war. Sie haben einen entscheidenden Beitrag zum Gelingen der Arbeit geleistet.

---

# Kurzfassung

Die steigende Beliebtheit des Fahrrads hat nicht nur positive Effekte. Eine vermehrte Anzahl an verletzten Radfahrer\*innen, insbesondere in Verbindung mit elektrifizierten Fahrrädern, zeigen dies eindrücklich. Während die Zweiradindustrie boomt, versuchen Bund, Länder und Kommunen den steigenden Anforderungen an die Infrastruktur gerecht zu werden. Diese muss nun für verschiedene Fahrradkonzepte mit unterschiedlichen Geschwindigkeitsprofilen ausgelegt werden. Um die neuen Aufgaben bewältigen zu können, werden vermehrt Probandenstudien unternommen, welche das Verhalten der Fahrradnutzer\*innen abbilden sollen. Insbesondere die Unfallentstehung, ihr Zusammenspiel mit der Infrastruktur und der Einfluss erhöhter Geschwindigkeit durch Elektrifizierung stehen hierbei im Fokus.

Diese *Naturalistic Cycling Studies* (NCS) verwenden mit Sensoren ausgestattete Fahrräder, welche die täglichen Fahrten ausgewählter Probanden und Probandinnen aufzeichnen. Um die Prinzipien von naturalistischen Studien nicht zu verletzen, müssen die Anbauten möglichst unaufdringlich sein. Diese Anforderung grenzt das verwendbare Sensorset stark ein, sodass üblicherweise keine redundante Sensorik verfügbar ist. Dennoch werden während der Studien enorme Datenmengen, hauptsächlich Kameradaten, aufgezeichnet. Die Sichtung der Daten und die Extraktion von studienrelevanten Situationen, sogenannte Events, wird bisher manuell bewältigt, da verlässliche automatisierte Methoden fehlen. Dieser Zustand führt zwangsläufig zu hohem personellem Aufwand und einer suboptimalen Probandenauswahl, getrieben durch die enormen Kosten der Auswertung. Die vorliegende Arbeit greift diese und weitere Probleme auf und bietet einen Ansatz zur Lösung des Dilemmas.

Eine Analyse der aktuellen Auswertungsmethoden von *Naturalistic-Cycling*-Daten und dem empfohlenen Auswertungsprozess von Probandenstudien im Straßenverkehr zeigen die Eventdetektion als den Ansatzpunkt für eine automatisierte Auswertung auf. Verschiedene Ansätze werden vorgestellt, basierend auf potenziell in NCS auftretenden Forschungsfragen. Der Fokus liegt hierbei auf der Detektion von kritischen Situationen, welche sich durch ein erhöhtes Unfallrisiko auszeichnen. Hierfür werden zwei Prozessketten mit Algorithmen zur Objekterkennung, Objektverfolgung und Berechnung der Situationskritikalität prototypisch implementiert und anhand konstruierter Szenarien auf ihre Funktionalität überprüft. Weitere Szenariobetrachtungen im Realverkehr schließen die Funktionstests ab. Anhand der Ergebnisse wird beantwortet, ob eine Auswertung der *Naturalistic-Cycling*-Daten automatisiert werden kann, und welche Schlüsselfaktoren bei der Auswahl des Sensorsets sowie der Umsetzung berücksichtigt werden müssen. Dabei werden die spezifischen Anforderungen an naturalistische Methoden stets geachtet. Ausblicke auf Verfahren, welche die prototypische Implementierung erweitern oder optimieren können, schließen die Arbeit ab.

---

# Abstract

The increasing popularity of bicycles leads to many positive effects, but also negative impacts appear. This is clearly shown by the rising numbers of injured bicyclists, many of whom riding electrified bikes. While the bike industry is flourishing, the government tries to meet the new demands regarding cycling infrastructure which must be usable by diverse bicycle concepts with different velocity profiles. To assess and finally meet the requirements w.r.t. the infrastructure, an increased amount of subject studies are carried out. Their goal is to evaluate the behaviour of cyclists with a focus on accident causation and the interplay of infrastructure design and increased velocity due to electrification.

The *Naturalistic Cycling Studies* (NCS) use sensor equipped bicycles to record the daily rides of selected subjects. The principles of naturalistic studies demand unobtrusive modifications to the bike, which limit the number of applicable sensors to a minimum and usually does not allow redundant sensors. Nevertheless, the used sensors record huge amounts of data, mainly camera data. The inspection of the data and the extraction of relevant scenarios w.r.t. the goal of the study, so-called events, is executed in a manual fashion as there are no reliable methods for automation. This leads to high personnel expenses for data analysis and therefore to a reduced and thus suboptimal selection of subjects. This thesis addresses these and other issues and provides a possible solution.

Current state of the art methods for analysing *Naturalistic Cycling* data as well as the recommendations for the analysis of subject studies in traffic indicate that the key for automation is the event detection. Different approaches are presented, based on potential NCS research questions. The detection of critical events, which are characterized by an increased risk of accident, is in focus. Therefore, two different approaches are implemented. Both implement algorithms for object detection, object tracking and calculation of criticality for specific events. Functional tests are performed on artificially created scenarios in a controlled environment as well as on real traffic data. Based on the results, the thesis answers the question whether the analysis of *Naturalistic Cycling* data can be automated. Additionally, key factors for the design of the sensor set as well as the realization of the analysis are highlighted. The specific requirements regarding naturalistic methods are respected at all times. Finally, methods for the extension and optimization of the provided implementations are shown.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
1.1	Naturalistische Untersuchungen im Straßenverkehr . . . . .	10
1.1.1	Die Anwendung der naturalistischen Methoden im Radverkehr	10
1.1.2	Kritische Situationen, Konflikte und ihre Beschreibung . . . . .	11
1.1.3	Die Grenzen der manuellen Auswertung von Naturalistic Cycling Versuchsdaten . . . . .	13
1.2	Ziele der Arbeit . . . . .	15
1.3	Vorgehen und Aufbau . . . . .	16
<b>2</b>	<b>Grundlagen der Stereovision</b>	<b>19</b>
2.1	Das Lochkameramodell . . . . .	19
2.2	Die Geometrie zweier Perspektiven . . . . .	22
2.3	Binokulare Stereokameras . . . . .	23
2.4	Kamerakalibration . . . . .	25
2.5	Systemdesign und Einflüsse der Parameterwahl . . . . .	27
2.6	Anwendung der Grundlagen in dieser Arbeit . . . . .	27
<b>3</b>	<b>Aufbau und Funktion des Analysetools</b>	<b>31</b>
3.1	Bisherige Arbeiten . . . . .	31
3.2	Anforderungen an die Funktionalität . . . . .	33
3.3	Aufbau und Workflow . . . . .	36
3.4	Erstellung der Disparitätenkarte . . . . .	38
3.4.1	Lokale und Globale Methoden . . . . .	39
3.4.2	Algorithmenauswahl . . . . .	39
3.5	Objektdetektion . . . . .	40
3.5.1	Kriterien zur Algorithmenwahl . . . . .	41
3.5.2	Regelbasierter Ansatz . . . . .	42
3.5.2.1	Regelbasierte Methoden zur Objektdetektion . . . . .	42
3.5.2.2	Auswahl der Algorithmenklasse . . . . .	44
3.5.2.3	Stereovisionsbasierte Objektdetektion . . . . .	44
3.5.2.4	Objektdetektion basierend auf U-Disparity . . . . .	49
3.5.3	Modellbasierte Objekterkennung . . . . .	54
3.5.3.1	Maschinelles Lernen und Deep Learning . . . . .	55
3.5.3.2	Stand der Technik . . . . .	58
3.5.3.3	Mask R-CNN . . . . .	59
3.6	Objektverfolgung . . . . .	61
3.6.1	Stand der Technik . . . . .	62
3.6.1.1	Das Kalman-Filter . . . . .	62

3.6.1.2	Das Erweiterte Kalman-Filter . . . . .	64
3.6.1.3	Datenassoziation . . . . .	64
3.6.1.4	Objektverwaltung . . . . .	65
3.6.2	Umsetzung . . . . .	65
3.7	Konfliktindikatoren . . . . .	67
3.8	Zusammenfassung des Kapitels . . . . .	69
<b>4</b>	<b>Funktionsanalyse und Bewertung</b>	<b>71</b>
4.1	Genauigkeit der Positionsbestimmung bei bekannten Objekten . . . . .	71
4.2	Erkennung von Objekten im realen Verkehr . . . . .	74
4.2.1	Regelbasierter Ansatz . . . . .	74
4.2.2	Modellbasierter Ansatz . . . . .	84
4.3	Funktionsanalyse - Grundwahrheit und realer Verkehr . . . . .	85
4.3.1	Funktionsanalyse mit Hilfe von konstruierten Szenarien . . . . .	86
4.3.1.1	Beschreibung der Versuche zur Bestimmung der TTC	87
4.3.1.2	Auswertung TTC . . . . .	88
4.3.1.3	Beschreibung der Versuche zur Bestimmung der PET	100
4.3.1.4	Auswertung PET . . . . .	101
4.3.2	Reale Szenarien im Straßenverkehr . . . . .	109
4.4	Bewertung der Ergebnisse . . . . .	117
<b>5</b>	<b>Mögliche algorithmische Anpassungen und Erweiterungen</b>	<b>121</b>
5.1	Online-Kalibration . . . . .	121
5.2	Tiefen- und Disparitätenkarten durch maschinelles Lernen . . . . .	122
5.3	Erweiterung des Messmodells um die Objektdimensionen . . . . .	123
5.4	Erweiterung des Messmodells um Geschwindigkeitssignale . . . . .	126
5.5	Schätzung der Eigenbewegung zur Sensordatenbereinigung . . . . .	127
<b>6</b>	<b>Weitere Analysemöglichkeiten für NCS-Daten</b>	<b>129</b>
6.1	Fahrradinfrastrukturerkennung . . . . .	130
6.2	Verkehrszeichen- und Ampelphasenerkennung . . . . .	134
6.3	Freiraumerkennung . . . . .	135
6.4	Weitere Anwendungen . . . . .	136
<b>7</b>	<b>Fazit</b>	<b>139</b>
7.1	Beantwortung der Forschungsfragen . . . . .	139
7.2	Fazit . . . . .	146
<b>Literaturverzeichnis</b>		<b>IX</b>
<b>Abbildungsverzeichnis</b>		<b>XXV</b>
<b>Abkürzungsverzeichnis</b>		<b>XXIX</b>
<b>Symbolverzeichnis</b>		<b>XXXI</b>
<b>Tabellenverzeichnis</b>		<b>XXXVII</b>
<b>Anhang</b>		<b>XXXIX</b>



---

# Kapitel 1

## Einleitung

Fahrradfahren erfreut sich wachsender Beliebtheit. Ob von Jung oder Alt, immer mehr Wege werden mit Fahrrädern bewältigt. So stieg der Anteil des Radverkehrs am Modal Split in Deutschland von 9 % im Jahr 2002 auf 11 % in 2017 [1]. Im Jahr 2020 wird eine Steigerung auf 15 % für möglich gehalten [2]. Die regelmäßige Nutzung von Fahrrädern und Pedelecs stieg laut [3] von 41 % in 2017 auf 44 % im Jahr 2019. Das freut nicht nur die Fahrradindustrie, welche von Jahr zu Jahr neue Umsatzrekorde insbesondere durch den wachsenden Markt für Pedelecs aufstellt und dabei ihr Portfolio stetig erweitert [4]. Selbst während der Corona-Pandemie 2020 konnte die Branche Umsatzsteigerungen generieren [5]. Auch die Politik hat diese Entwicklung mit Freude angenommen, da die positiven Effekte des Radverkehrs auf das gesellschaftliche Leben, die Umwelt als auch die Gesundheit der Fahrer\*innen nicht von der Hand zu weisen sind. Projekte und Förderpläne auf verschiedenen Ebenen sollen den Anteil des Radverkehrs in Deutschland weiter erhöhen, um somit einen direkten, positiven Beitrag zu aktuellen gesellschaftlichen Herausforderungen zu leisten [2].

Die dominierende Herausforderung im Jahr 2020 ist die Eindämmung der Verbreitung des COVID-19 Virus, weshalb der öffentliche Nahverkehr nur reduzierte Personenzahlen transportieren kann, ohne die Vorschriften bezüglich des Mindestabstands zu missachten. Das Fahrrad stellt auch hier eine Alternative dar, welche die Mobilität zumindest in urbanen Räumen bei geringem Ansteckungsrisiko sicherstellen kann. Um die speziellen Anforderungen dieser Situation zu bedienen, werden vermehrt temporäre Pop-Up-Radwege eingesetzt, deren Planung, Bewilligung und Umsetzung zügig von statten gehen [6]. Dabei wird geteilter Verkehrsraum für die alleinige Fahrradnutzung umgewidmet. Die Markierung einer Fahrspur durch gelbe Linien oder Baken ist ein einfaches Mittel der Umsetzung.

Die Entwicklung der Mobilität hat für den Radverkehr jedoch auch Nachteile. Die in Deutschland gemessene tägliche Verkehrsleistung in Personenkilometern hat seit 2002 für Pkw-Fahrer\*innen sowie Fahrradfahrer\*innen zugenommen. Während für Pkw die Personenkilometer in diesem Zeitraum um knapp 3 % stiegen, erhöhte sich die Verkehrsleistung von Radfahrer\*innen um fast 29 % [1]. In urbanen Räumen können die Extreme dieser Entwicklung beobachtet werden. Pegelzählungen in Berlin konnten an einem Messpunkt im Stadtteil Kreuzberg eine Zunahme der Radverkehrsmenge um 175 % erfassen [7]. Dies führt zu einer höheren Aus- und potenziellen Überlastung der Radinfrastruktur, wenn diese nicht entsprechend angepasst wird. Weiterhin werden durch die Elektrifizierung von Fahrrädern höhere

durchschnittliche Geschwindigkeiten erreicht [8] und der Radverkehr auch für ältere Nutzergruppen wieder attraktiv [9]. Diese Effekte erzeugen ein gefährliches Spannungsfeld, welches sich auch in den Unfallzahlen ausdrückt. Während nach [10] von 2010 bis 2017 die Zahl der Verkehrstoten in Deutschland um 13 % sank, stagnierte die Zahl der im Straßenverkehr getöteten Radfahrer\*innen. Die Zahl der Verletzten im Radverkehr stieg in diesem Zeitraum um über 20 %, die Gesamtzahl der Verletzten um ca. 5 %. Die Radunfälle ereigneten sich dabei zu 91 % innerorts. Seit der getrennten Erhebung von Pedelecunfällen im Jahr 2014 haben sich Pedelecunfälle mehr als verdoppelt. Sie stellen 2017 6,5 % der Gesamtzahl der Unfälle im Radverkehr dar, wobei Personen ab 75 Jahren überrepräsentiert sind. Von den im Zeitraum 2014 bis 2017 getöteten Pedelecnutzer\*innen gehörten 64 % zu dieser Altersgruppe, hingegen nur 34 % der Getöteten mit herkömmlichem Fahrrad.

Die Gesetzgebung und Infrastruktur an die Veränderungen des Rad- sowie des Gesamtverkehrs anzupassen ist eine Herausforderung. Hierfür müssen vielfältige Einflüsse auf die Fahrradsicherheit identifiziert und untersucht werden, bevor neue Rahmenbedingungen mit Hinblick auf den Verkehr der Zukunft geschaffen werden können. Die Untersuchungen müssen sich am realen Fahrverhalten orientieren, nicht an reinen Unfallzahlen [11] und langjährigen Verkehrsstatistiken, welche das Bild des Straßenverkehrs verzerrt wiedergeben können und folglich die Bedürfnisse der verschiedenen Nutzergruppen nicht abbilden.

### 1.1 Naturalistische Untersuchungen im Straßenverkehr

Naturalistische Feldstudien zur Untersuchung des Fahrverhaltens (*Naturalistic Driving Studies*, NDS) beschreiben eine Methodik, welche das Verhalten von Fahrer\*innen in ihrer natürlichen Umgebung über eine längere Zeit erfassen. Zu den gesammelten Daten zählen üblicherweise die Fahrzeugbewegung, Verhaltensweise der Fahrer\*innen sowie verschiedene Umfelddaten. Die Datenerfassung geschieht so unaufdringlich wie möglich und bevorzugt im Fahrzeug der Proband\*innen während der üblichen, täglichen Fahrten. Die gesammelten nicht-reaktiven Daten lassen Rückschlüsse auf das Spannungsfeld zwischen Fahrzeug-, Fahrer\*innen- und Umwelteinflüssen und deren Wirkung auf das Verhalten der Proband\*innen zu [12]. Andere Untersuchungsmethoden wie Simulatortests oder Kurzzeitbeobachtungen können hingegen meist keine belastbaren Daten bezüglich des Fahrverhaltens im natürlichen Umfeld bieten, da sich die Proband\*innen der Untersuchungssituation bewusst sind und die Daten somit reaktiv sind. Ihr Nutzen bezogen auf die Erforschung von Unfallursachen ist darum begrenzt. Während naturalistischer Langzeitstudien kann dieser Umstand vergessen werden und die Daten spiegeln das reale Fahrverhalten wider [13].

#### 1.1.1 Die Anwendung der naturalistischen Methoden im Radverkehr

Die Methodik der NDS lässt sich nicht nur auf motorisierte Fahrzeuge, sondern auch auf Fahrräder anwenden. Hierfür hat sich die Bezeichnung *Naturalistic Cycling Study* (NCS) durchgesetzt. Eine der bisher größten Studien dieses Forschungsbereichs stellt

die *Pedelec-Naturalistic Cycling Study* dar, welche die Verkehrssicherheit von Pedelecs im Vergleich zu herkömmlichen Fahrrädern als zentrale Fragestellung behandelt [14]. Die Auswahl der Sensoren muss sich bei NDS auf die Forschungsfragen und die daraus abgeleiteten, zu untersuchenden Kennzahlen (*Performance Indicator*, PI) stützen, ohne die Anforderungen der naturalistischen Methodik zu vergessen. Dies führt gerade bei Fahrrädern zu besonderen Ansprüchen. Pkw und Lkw haben heutzutage meist Sensorik zur Umfeld-, Fahrzeugzustands- und Fahrerzustandserfassung an Bord, um eine Vielzahl von Fahrerassistenzsystemen zu speisen. Die Informationen dieser Sensoren können für NDS verwendet werden, was von [13] vorgeschlagen wird. Weitere, auch redundante Sensoren können ohne Schwierigkeiten an diesen Fahrzeugen angebracht werden, ohne die Anforderungen an eine unaufdringliche Erhebung der nicht-reaktiven Daten zu verletzen. Aufgrund des geringen Stauraums und Gewichts von Fahrrädern, muss die Auswahl der Sensoren sowie des zusätzlich benötigten Datenaufzeichnungssystems (DAS) mit großer Sorgfalt geschehen, um das Fahrverhalten und das Erscheinungsbild des Fahrrads möglichst nicht zu verändern. Dennoch muss das Gesamtsystem strenge Anforderungen an Robustheit, Laufzeit, Budget u. s. w. erfüllen [15].

Die oben erwähnte *Pedelec-Naturalistic Cycling Study* und andere bekannte NCS verwenden meist Videokameras zur Umfelderkennung, das globale Navigationssatellitensystem (GNSS) zur Positionsbestimmung und weitere Sensoren für die Messung kinematischer Größen. So haben nach [16] sieben von neun gelisteten Versuchsfahrern mindestens eine Kamera als Sensor verbaut. Zur Beobachtung der Umgebung und des Fahrerhaltens entsprechen diese Sensoren dem Stand der Technik, da sie dem Auswertungspersonal den Kontext zum Fahrerhalten liefern [13]. Videodaten bieten den Vorteil hoher Informationsdichte und können, anders als z. B. Ultraschallsensoren, während der Wiedergabe und ohne Nachprozessierung von einem Menschen interpretiert werden.

Wenn NDS-Datensätze mit Bedacht erzeugt werden und genügend Varianz im Sinne der Proband\*innen und Fahrzenarios aufweisen, können eventuell weitere Forschungsfragen beantwortet werden. Die Daten können somit für andere Studien von großem Nutzen sein. Daher sollte von Beginn einer Studie an ein starker Fokus auf die Nachnutzung der Daten und den damit verbundenen Hürden, wie z. B. Datenschutz, gelegt werden [17]. Videodaten sind aufgrund der hohen Fülle an Informationen gut für eine Nachnutzung geeignet, bergen jedoch bezogen auf den Datenschutz erhebliches Konfliktpotenzial.

### 1.1.2 Kritische Situationen, Konflikte und ihre Beschreibung

Die während einer Feldstudie generierten Daten können nach [13] vier unterschiedliche Ausprägungen aufweisen:

- **Direkt gemessene Daten** (*Direct Measures*): Messungen von einem Sensor ohne Nachbearbeitung, z. B. Beschleunigungsdaten eines Accelerometers oder Videodaten.
- **Abgeleitete Daten** (*Derived Measures*): Aus direkten Messungen abgeleitete Daten, z. B. die Geschwindigkeit aus Beschleunigungsdaten oder die aus Umfelddaten berechnete Zeit bis zum Aufprall auf eine\*n Unfallgegner\*in (*Time to Collision*, TTC).

- **Probandenangaben** (*Self-Reported Measures*): Subjektive Daten aus Probandenangaben, z. B. erhoben durch Interviews oder Fragebögen.
- **Situationsbezogene Variablen** (*Situational Variables*): Variablen, welche den PIs eine Basis für weitere Analysen bieten, z. B. Angaben über das Wetter, Reibwert der Fahrbahn oder die verwendete Infrastruktur. So kann beispielsweise die Kennzahl „Unfälle pro Millionen gefahrener Kilometer“ weiter nach der verwendeten Infrastruktur untergliedert werden.

Eine weitverbreitete Methode für die Auswertung von NDS-Daten ist die event-basierte Analyse (*Event Based Analysis*, EBA). Hierfür werden Kriterien definiert, welche in den Messdaten identifiziert werden können, und somit einen bestimmten Zeitbereich der gemessenen Daten als Event deklarieren. Die in [14] beschriebenen kritischen Ereignisse beinhalten neben Stürzen und Beinahestürzen ohne Einwirkung anderer Verkehrsteilnehmer\*innen auch Situationen, welche durch die Interaktion zwischen Radfahrer\*innen und anderen Subjekten hervorgerufen werden. In der Terminologie von Verkehrssicherheitsstudien, welche Unfallursachen mit anderen Verkehrsteilnehmerinnen und -teilnehmern erforschen, werden diese Events oft als *Crash Relevant Events* [13], Konflikte [14], *Conflict Events* [18] oder *Traffic Conflicts* [11] bezeichnet. Sie beschreiben Situationen mit erhöhtem Unfallrisiko, welche eine Korrelation zur Entstehung tatsächlicher Unfälle aufweisen [19]. Da Unfälle in Feldstudien selten sind, werden Konflikte und kritische Situationen als Ersatz zur Erforschung der Unfallentstehung verwendet [20].

In dieser Arbeit werden die Begriffe Event, kritische Situation und Konflikt in Anlehnung an obige Definitionen wie folgt beschrieben:

- **Event**: Ein Event beschreibt einen für die Beantwortung der Forschungsfragen relevanten Zeitbereich in den Messdaten. Dies kann z. B. ein Wechsel der verwendeten Infrastruktur oder das Überfahren einer signalisierten Kreuzung sein. Bei Studien zur Verkehrssicherheit werden meist kritische Situationen als relevante Events deklariert.
- **Kritische Situation**: Tritt beim Aufenthalt im Verkehrsraum eine Situation auf, welche ein erhöhtes Unfallrisiko gegenüber der Normalfahrt aufweist, so gilt diese als kritisch. Auch tatsächlich ereignete Unfälle fallen in diese Klassifizierung. Dabei spielt es keine Rolle, ob sich das Unfallrisiko durch die Einwirkung anderer Verkehrsteilnehmer\*innen, die Infrastruktur, sonstiger Objekte oder ohne Fremdeinwirkung verändert hat.
- **Konflikt**: Ist an einer kritischen Situation mindestens ein\*e weitere\*r Verkehrsteilnehmer\*in beteiligt, so wird dies als Konflikt bezeichnet. Konflikte mit Unfallfolge können bei fehlendem direktem Kontakt nicht immer zweifellos als solcher identifiziert werden. Beispielhaft sei hier die Bremsung einer radfahrenden Person auf Glatteis mit Sturzfolge genannt, wobei die Person an einer Kreuzung die Vorfahrt gewähren musste. Die Bildung der Kausalkette ist schwierig, insbesondere die Ursache der Bremsung ist schwer zu definieren. Sie kann beispielsweise aus der Verkehrsführung, aber auch aus dem Konflikt an sich resultieren.

Nach [13] können die Kriterien zur Extraktion der kritischen Situationen in vier Kategorien eingeordnet werden:

- **Fahrerreaktion** (*Driver Response Based*): Die Fahrerreaktion weist hierbei auf eine kritische Situation hin. Dabei kann die Reaktion beispielsweise durch eine signifikante Änderung der Fahrzeugkinematik, wie z. B. starke Bremsmanöver, beschrieben werden [21]. Andere Studien verwenden maschinelles Sehen, um aus Oberkörperbewegungen von Pkw-Fahrer\*innen kritische Situationen zu detektieren [22]. Auch die direkte Einschätzung der Fahrer\*innen kann hierfür verwendet werden, indem mittels Fahrtenbuch nach entsprechenden Situationen gefragt wird oder das Messsystem einen Trigger zur direkten Kennzeichnung der Messdaten durch den\*die Probanden\*in aufweist [15].
- **Sicherheitsfunktion** (*Safety Function Response*): Weist das Versuchsfahrzeug Sicherheitsfunktionen auf, wie z. B. einen Notbremsassistenten, so können diese Signale als sichere Indikatoren für kritische Situationen dienen. Für Fahrräder sind jedoch keine entsprechenden Funktionen verfügbar.
- **Fahrkontext** (*Driving Context Based*): Kontextbasierte Kriterien verwenden Umfeldinformationen zur Klassifizierung der Events, wie beispielsweise die TTC oder auf der Nebenspur passierende Fahrzeuge. Dabei können die Situationen zusätzlich nach ihrer Schwere eingeordnet werden [11]. Neben kritischen Situationen können auch andere Events, wie z. B. Infrastrukturwechsel, aus dem Fahrkontext abgeleitet werden.
- **Fahrverhaltenshistorie** (*Driving History Based*): Treten in den Messdaten signifikante Ausreißer bezogen auf die Verhaltenshistorie der einzelnen Fahrer\*innen auf, so kann die Annahme getroffen werden, dass dieses Verhalten ungewollter Natur ist. Eine Einordnung als kritisch im Sinne der Verkehrssicherheit ist jedoch fraglich.

Diese Kriterien können auch kombiniert werden. Ein konfliktbehafteter Überholvorgang zweier Pkw kann beispielsweise durch geringe Abstände der Subjekte (Fahrkontext), dem Signal des Totwinkelwarnsystems (Sicherheitsfunktion) sowie einer potenziellen Gefahrenbremsung oder ein Ausweichmanöver (Fahrerreaktion) detektiert werden. Die nach Event klassifizierten Messdaten können nun zur Berechnung der PIs verwendet werden, um die aufgestellten Hypothesen der Studie zu testen und somit die Forschungsfragen zu beantworten.

### 1.1.3 Die Grenzen der manuellen Auswertung von Naturalistic Cycling Versuchsdaten

Vergleicht man die generierten Datenmengen verschiedener NCS und die daraus extrahierten kritischen Situationen, werden die Dimensionen der Datenauswertung verdeutlicht. Tabelle 1.1 zeigt Kennzahlen von drei Studien mit dem Fokus Verkehrssicherheit.

Hierbei fallen insbesondere, trotz der erheblichen Umfänge der Studien, die geringe Anzahl an kritischen Situationen sowie an Unfällen auf. Folglich ist eine maximale Detektionsrate dieser Events von großer Bedeutung. Dies ist auch der Grund, weshalb diese Studien auf die manuelle Auswertung der Daten durch geschultes Personal

**Tabelle 1.1**

Datenmengen und Anzahl kritischer Situationen verschiedener NCS

Studie	[23]	[24]	[14]	Einheit
Gesamte Distanz	1549	8986	16 986	km
Gesamte Dauer	114	466	2300	h
Anzahl Proband*innen	20	36	86	-
Erhebungszeitraum	2	4	4	Wochen
kritische Situationen	63	91	202	-
Unfälle	6	0	3	-

setzen, obwohl alternative Ansätze existieren [24]. Hierbei werden die Rohdaten mit Hilfe einer Analysesoftware gesichtet und entsprechend der Beobachtungen kodiert [25]. Dieser Vorgang benötigt ein Vielfaches der eigentlichen Aufnahmezeit, wobei die Dauer mit dem Detailgrad variiert. In [14] wird für die Kodierung eines einstündigen, nach vorne ausgerichteten Videostreams ein Aufwand von dreieinhalb Stunden genannt. Dabei wurden aus dem Video Informationen über das Infrastrukturangebot, deren regelkonformer Nutzung sowie der erreichten Geschwindigkeit extrahiert. Auch die Extraktion von kritischen Situationen geschieht anhand dieser Methodik. Um die Kosten einer NCS nicht unverhältnismäßig steigen zu lassen, kann folglich mit diesen Methoden nur ein begrenztes Datenvolumen bearbeitet werden. Ein weiterer Kritikpunkt an der manuellen Auswertung von Videodaten ist der Einfluss der persönlichen Wahrnehmung des auswertenden Personals [22]. Unterschiede in der Bewertung von Verkehrssituationen sind trotz aufwendiger Schulungen und Reviews unausweichlich.

Die Kritik an der manuellen Auswertung von NDS hat verschiedene Ansätze zur automatisierten Detektion von kritischen Situationen hervorgebracht. Weitere Events, wie Infrastrukturwechsel oder das Passieren signalisierter Kreuzungen, werden bisher ausschließlich manuell erkannt. Für die Extraktion der kritischen Situationen spielen insbesondere die kinematischen Signale eine Rolle, mit deren Hilfe Brems- oder Ausweichmanöver erkannt werden [22]. Nach [24] werden bei der Anwendung in NCS jedoch viele Events aufgrund zu geringer Dynamik nicht detektiert. Weiterhin können nur Events extrahiert werden, welche eine Reaktion der Proband\*innen zur Folge hatten. Bei einer alleinigen Reaktion der Konfliktpartner\*innen wird die kritische Situation nicht erkannt. Als weitere Fahrerreaktion wurde von [22] die Bewegung des Oberkörpers von Pkw-Fahrer\*innen verwendet, allerdings kann diese Methodik in NCS aufgrund der häufigen und vielfältigen Bewegungen auf dem Fahrrad nicht zum Einsatz kommen. Die Verwendung von fahrkontextbasierten Daten findet bei der automatisierten Eventdetektion in NCS noch keine Anwendung, ist bei der Verwertung stationärer Kameradaten in der naturalistischen Datenaufzeichnung jedoch Stand der Technik [11] [26]. Mit Methoden des maschinellen Sehens (*Computer Vision*) werden relevante Objekte erkannt und mit Hilfe eines Algorithmus zum *object tracking* verfolgt. Die daraus generierten Trajektorien und kinematischen Daten können genutzt werden, um Konflikte zu de-

tektieren und Kennzahlen für die Kritikalität zu berechnen. Dabei spielt es für die Detektion des Events keine Rolle, ob die Konfliktpartner\*innen eine Reaktion zeigen, da Kriterien wie TTC unabhängig von der Fahrerreaktion definiert sind und sich auf objektiv messbare räumliche und kinematische Beziehungen zwischen den Objekten stützen.

Die Adaption dieser Methodik auf NCS ist ein logischer nächster Schritt für die automatisierte Detektion von Konflikten. Dies kann durch die Möglichkeiten moderner Bildverarbeitung auf kritische Situationen in Verbindung mit Objekten, welche nicht als Verkehrsteilnehmer\*innen klassifiziert wurden, erweitert werden. Zusätzlich können weitere Kriterien für Events aus dem Fahrkontext gewonnen werden. Die Wahrnehmung und Interpretation des Umfelds entspricht dabei der Kernaufgabe von modernen, umfelderfassenden Fahrerassistenzsystemen (*Advanced Driver Assistance Systems*, ADAS) und autonomen Fahrzeugen. In der Umsetzung ergeben sich jedoch große Unterschiede, da ADAS und *Autonomous Driving* (AD) Systeme meist mehrere, redundante Umfeldsensoren wie Kameras, LiDAR oder RaDAR aufweisen, deren Daten fusioniert werden. Die NCS-Methodik erfordert jedoch minimalistische Sensorsets, deren Daten weiterhin für manuelles *Postprocessing* verfügbar sein sollten, um die Auswertung prüfen und bei Bedarf durch manuelle Prozesse erweitern zu können.

## 1.2 Ziele der Arbeit

Die Beschreibung der bisherigen Anwendung der automatisierten Detektion von kritischen Situationen in NCS-Versuchsdaten, siehe Kapitel 1.1.3, zeigt eindeutig die Notwendigkeit der Erweiterung auf. Bisher wurde nur die Eventdetektion basierend auf Fahrerreaktionen umgesetzt. Fahrkontextbasierte Kriterien wurden nicht berücksichtigt, obwohl vielversprechende Ergebnisse aus Studien mit statischen Kameras verfügbar sind. Diese Arbeit leistet einen Beitrag zum Schließen der Lücke und überträgt erstmalig diese Ansätze auf Kameradaten aus NCS, wobei die Verwendung von Fahrrädern als Versuchsfahrzeuge neben dem nicht-redundanten Sensorset eine besondere Hürde darstellen. Da dynamische Fahrerreaktionen bereits gut detektiert werden können, sollen nun auch kritische Situationen ohne oder mit geringer Fahrerreaktion erkannt werden. Hierfür werden objektiv messbare Kriterien herangezogen, welche auf räumlichen Beziehungen und kinematischen Größen des eigenen Fahrzeugs und Objekten der näheren Umgebung basieren.

Die kombinierte Detektionsrate aus fahrkontext- und fahrerreaktionsbasierten Kriterien kann das grundlegende Problem der eventbasierten Analyse, nämlich die zeitaufwendige und oft subjektive Bewertung von Verkehrssituationen, entschärfen oder in bestimmten Fällen gar lösen. Folglich wäre das Design einer NCS nicht mehr durch die Auswertung als Kostenfaktor beschränkt, sondern kann in weit größeren Dimensionen mit repräsentativeren Probandengruppen umgesetzt werden. Beispielfähig könnte der 2300 Stunden umfassende Datensatz aus [14] bei einer Auswertung bezüglich der Verkehrssicherheit auf 202 kritische Situationen reduziert werden. Nimmt man eine für die Dauer einer kritischen Situation, inklusive Entstehung, Konfliktlösung und Dokumentation der Konfliktfolgen, in etwa 30 Sekunden an, so reduziert sich die Dauer der auszuwertenden Daten auf knapp zwei Stunden. Die Auswertung nach PIs benötigt somit weniger als ein Promille der ursprüngli-

chen Zeit. Die fahrkontextbasierten Kriterien können dabei gleichzeitig als objektive Grundlage der zu extrahierenden PIs dienen.

Ziel dieser Arbeit ist, die fahrkontextbasierte Detektion von relevanten Events auf NCS anzuwenden. Dabei wird ein komplementärer Beitrag zur bestehenden automatisierten Erkennung basierend auf Fahrerreaktionen geleistet. Im Vordergrund steht die Erkennung kritischer Situationen im Kontext der Verkehrssicherheit, da dieses Kriterium in bisherigen NCS der Haupttreiber der EBA ist. Weiterhin stellt die Umsetzung im Spannungsfeld der NCS-Methodik im Vergleich zu anderen EBA Kriterien die größte Herausforderung dar. Hierfür wird prototypisch ein kostengünstiges Stereokamerasystem verwendet, um die benötigten Umfelddaten extrahieren zu können und gleichzeitig eine manuelle Interpretierbarkeit sicherzustellen. Die Events werden anhand des aktuellen Stands der Technik im Bereich maschinelles Sehen aus den Videodaten extrahiert. Die Entwicklung von neuen Algorithmen für diese Extraktion, oder Teilprozesse ebendieser, sind hingegen nicht Gegenstand dieser Arbeit. Vielmehr liegt der Fokus in der Übertragung von bekannten Methoden auf den Anwendungsbereich NCS, der mit spezifischen Anforderungen aufwartet. Auch weitere Kriterien für Events, wie beispielsweise der Wechsel der genutzten Infrastruktur, werden behandelt und Implementierungen oder Ansätze für ebensolche vorgestellt, um eine ganzheitliche Betrachtung der NCS-Daten zu ermöglichen. Diese Arbeit stellt somit das fehlende Glied in der automatisierten Auswertung von NCS-Daten dar und ermöglicht weitere Freiheiten im Studiendesign durch die Entkopplung eines Großteils der Auswertung von den verfügbaren Personalressourcen.

Im Detail formulieren sich folgende Forschungsfragen:

1. Wie kann ein Stereokamerasystem für NCS aussehen und wie beeinflusst die Auslegung die Ergebnisse?
2. Wie lauten die Anforderungen an ein Auswertungstool?
3. Wie kann eine Implementierung auf Basis des aktuellen Stands der Technik im Bereich maschinelles Sehen gestaltet werden?
4. Wie gut ist die Erkennungsrate relevanter Objekte?
5. Wie gut ist die Genauigkeit und Aussagekraft von abgeleiteten Messdaten?
6. Wie zuverlässig ist das Auswertungstool bezogen auf kritische Situationen und Konflikte?
7. Welche Aspekte der manuellen Sichtung kann die automatisierte Auswertung nicht leisten?
8. Welche sinnvollen Erweiterungen des Auswertungstools sind denkbar?

### 1.3 Vorgehen und Aufbau

Die Verwendung einer Stereokamera erlaubt es, Tiefeninformationen aus dem gesammelten Bildmaterial zu extrahieren. Für detaillierte Untersuchungen zur Objekterkennung werden die Grundlagen der Stereovision benötigt, welche in Kapitel

2 erarbeitet werden. Kapitel 3 bietet einen Überblick über den Stand der Technik der fahrkontextbasierten Erkennung von Konflikten mit statischen Kameras im Kontext von naturalistischen Datenerhebungen im Straßenverkehr. Diese Methodik bietet die Grundlage für eine Übertragung auf NCS. In der Folge werden Kriterien an die Funktionalität eines Auswertungstools definiert, und dessen genereller Aufbau beschrieben. Der Kernteil des Kapitels wird durch den Stand der Technik der einzelnen Unterkomponenten und den spezifischen, teilweise konkurrierenden Umsetzungen eingenommen. Alle Komponenten werden in Kapitel 4 entsprechend ihrer Funktionalität bewertet und ggf. verglichen. Dabei kommen Testszenarien aus einer kontrollierten Testumgebung mit Referenzsensoren sowie Szenarien aus dem realen Straßenverkehr ohne Referenz zum Einsatz. Mögliche Anpassungen und Erweiterungen des entwickelten Analysetools werden in Kapitel 5 vorgestellt. Dabei liegt der Fokus auf der algorithmischen Optimierung des Auswertungstools. In Kapitel 6 werden weitere Möglichkeiten zur automatisierten Analyse von NCS-Daten abseits der kritischen Situationen vorgestellt, wobei unter anderem die verwendete Infrastruktur klassifiziert wird. Kapitel 7 schließt die Arbeit mit der Beantwortung der Forschungsfragen und einem Fazit ab.



---

# Kapitel 2

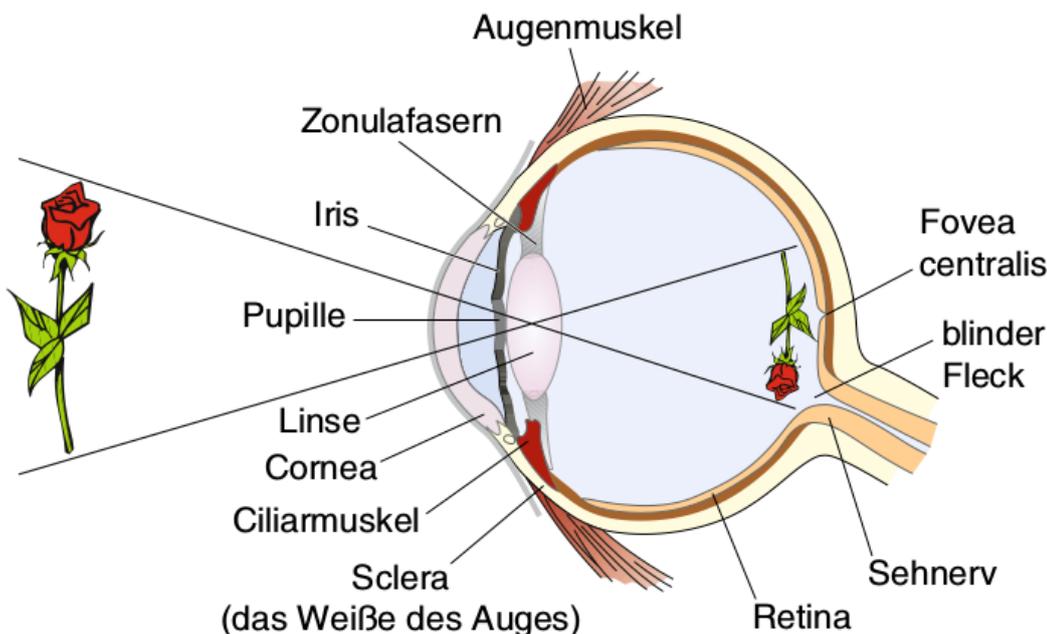
## Grundlagen der Stereovision

Die visuelle Wahrnehmung des Menschen basiert auf der Funktionsweise des Auges, welches als Sinnesorgan Parallelen zu einer Kamera aufweist. Wird eine Szene durch eine Lichtquelle beleuchtet und die Objekte dieser Szene reflektieren das sichtbare Licht, so kann das Auge eine zweidimensionale Abbildung dieser Szene detektieren. Der Aufbau des Auges, dargestellt in Abbildung 2.1, limitiert den Lichteinfall durch die Pupille, bündelt das Licht mit Hilfe einer Linse und sensiert das seiten- und höhenverkehrte Abbild mit Hilfe der Retina [27]. Die Beweglichkeit der Iris lässt eine Steuerung des Lichteinfalls zu, während die ebenfalls flexible Linse das Fokussieren von Objekten in unterschiedlichen Entfernungen ermöglicht. Weitere Besonderheiten stellen die inhomogene Verteilung der verschiedenen Rezeptoren über die Fläche der Retina sowie deren Konvergenz zu rezeptiven Feldern dar [28].

Der Mensch kann nach [28] aus verschiedenen Kriterien Tiefeninformationen extrahieren. Aus einer zweidimensionalen Abbildung einer Szene, d. h. diese ist mit nur einem Auge wahrgenommen, werden monokulare Tiefenkriterien entnommen. Dabei wird, neben der Linearperspektive und Teilverdeckung von Objekten, auch das Wissen über die Größe von bekannten Objekten verwendet. Durch die paarweise Anordnung der Augen können auch binokulare Tiefenkriterien verwendet werden. Neben den okulomotorischen Kriterien, welche sich auf die Rückmeldung der Muskeln bezüglich der Augenstellung und Linsenanpassung stützen, existiert die Querdisparation. Dieses Kriterium resultiert aus den unterschiedlichen Blickwinkeln der beiden Augen auf dieselbe dreidimensionale Szene. Abgesehen von den okulomotorischen Kriterien können diese Tiefenkriterien, ob monokular oder binokular, auch für die Extraktion von Tiefeninformationen aus Kamerabildern verwendet werden [29]. Somit existieren nicht nur Parallelen zwischen Auge und Kamera als Sensor, sondern auch für die Verarbeitung der gewonnenen Daten in Bezug auf Tiefeninformationen und einhergehend auf die Rekonstruktion dreidimensionaler Szenen.

### 2.1 Das Lochkameramodell

In [30] wird eine Kamera als Abbildung der dreidimensionalen Welt auf ein zweidimensionales Bild beschrieben. Das einfachste geometrische Modell einer perspektivischen Projektion ist das Lochkameramodell, siehe Abbildung 2.2. Nach [31] werden die meisten realen Kameras durch dieses Modell gut beschrieben, wenn auch zusätzliche Effekte, wie z. B. die Linsenverzerrung, zusätzlich betrachtet werden müssen. Der Kern des Modells besteht laut [32] aus der Aussage, dass sich alle Punkte  $\tilde{M}$

**Abbildung 2.1**

Das menschliche Auge, entnommen aus [28].

im projektivem Raum  $\mathcal{P}^3$  über ein optisches Zentrum  $C$ , auch Brennpunkt genannt, auf eine Bildebene  $I$  im  $\mathcal{P}^2$  abbilden lassen. Ein Lot durch das optische Zentrum auf die Bildebene bildet die optische Achse  $Z_c$ , welche die Bildebene im Kamerahauptpunkt  $c$  schneidet. Der Abstand  $\overline{CC} = Z_0$  wird als Brennweite  $f$  bezeichnet. Die Abbildung einer Kamera kann in drei einzelne Transformationen unterteilt werden, welche sukzessive einen Punkt in Weltkoordinaten auf Sensorkoordinaten projiziert. Die einzelnen Schritte sind dabei die externe Transformation, die perspektivische Transformation und die interne Transformation [30] [31] [32].

### Externe Transformation

Der Ursprung des Kamerakoordinatensystems liegt im Brennpunkt  $C$  der Kameralinse. Die  $Z$ -Achse liegt auf der optischen Achse. Ein Punkt  $\mathbf{M}_w$  im beliebigen Weltkoordinatensystem  $O$  kann durch Translation um  $\mathbf{t}$  und Rotation um  $\mathbf{R}$  in das Kamerakoordinatensystem überführt werden. Mit der extrinsischen Matrix  $\mathbf{D}$  gilt für homogene Koordinaten:

$$\tilde{\mathbf{M}}_c = \mathbf{D} \cdot \tilde{\mathbf{M}}_w \quad (2.1)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3 & 1 \end{bmatrix}, \quad \text{mit } \mathbf{0}_3 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad (2.2)$$

### Perspektivische Transformation

Die perspektivische Transformation beschreibt die Abbildung eines Punktes in homogenen Kamerakoordinaten  $\tilde{\mathbf{M}}_c$  auf die zweidimensionale Bildebene mit dem dar-

aus resultierenden Punkt  $\tilde{\mathbf{m}}'$ , ebenfalls in homogenen Koordinaten, siehe Formel 2.3.

Dabei gilt

$$\tilde{\mathbf{m}}' = \begin{bmatrix} U \\ V \\ S \end{bmatrix}, \quad \tilde{\mathbf{M}}_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

und somit

$$\tilde{\mathbf{m}}' = \mathbf{P}' \cdot \tilde{\mathbf{M}}_c \quad (2.3)$$

Die Umrechnung in kartesische Koordinaten der Bildebene erfolgt durch:

$$\mathbf{m}' = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} U/S \\ V/S \end{bmatrix}, \quad \text{mit } S \neq 0 \quad (2.4)$$

### Interne Transformation

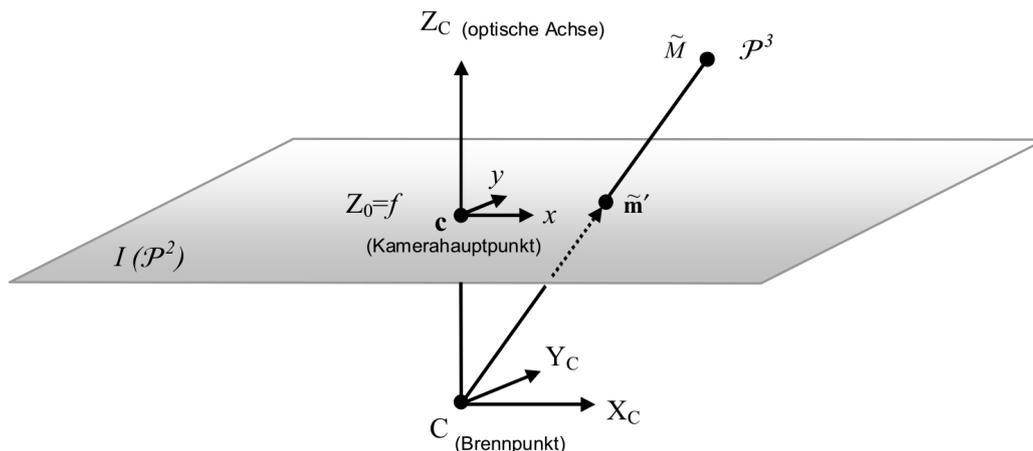
Um die metrische Darstellung der Bildkoordinaten in diskrete Sensorkoordinaten zu wandeln, muss die interne Transformation angewendet werden. Durch die horizontalen und vertikalen Skalierungsfaktoren  $k_u$  und  $k_v$  findet die Diskretisierung statt. Zusätzlich wird eine Translation um  $\mathbf{t} = [u_0, v_0]^T$  innerhalb der Bildebene angewandt um den Ursprung vom Kamerahauptpunkt  $c$  in eine Bildecke zu verschieben. Dies erleichtert die Indexierung der diskreten Bildelemente, da negative Indizes vermieden werden. Der Scherungsfaktor  $\gamma$  spiegelt eine mögliche nicht-orthogonale Ausrichtung der Bildkoordinaten wider. Die Transformation der homogenen Bildkoordinaten ergibt sich somit wie in Formel 2.5 beschrieben.

$$\tilde{\mathbf{m}} = \mathbf{H} \cdot \tilde{\mathbf{m}}', \quad \text{mit } \mathbf{H} = \begin{bmatrix} k_u & \gamma & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

### Projektionsmatrix

Die drei Transformationen lassen sich, wie in Formel 2.6 dargestellt, zusammenfassen. Dabei ist  $\mathbf{A}$  die intrinsische Matrix (Formel 2.7), welche die perspektivische und interne Transformation vereint. Die intrinsische Matrix besitzt fünf Freiheitsgrade. Die extrinsischen Parameter weisen weitere sechs Freiheitsgrade auf, jeweils drei für die Rotation und Translation von Welt- in Kamerakoordinaten. Folglich besitzt die Projektionsmatrix  $\mathbf{P}$  insgesamt elf Freiheitsgrade, und stellt damit eine bis auf einen Skalierungsfaktor eindeutige Abbildung eines dreidimensionalen Raumpunkts auf die zweidimensionale Bildebene dar.

$$\tilde{\mathbf{m}} = \mathbf{A} \cdot [\mathbf{R} \ \mathbf{t}] \cdot \tilde{\mathbf{M}}_w = \mathbf{P} \cdot \tilde{\mathbf{M}}_w \quad (2.6)$$

**Abbildung 2.2**

Die perspektivische Transformation des Lochkameramodells, entnommen aus [32].

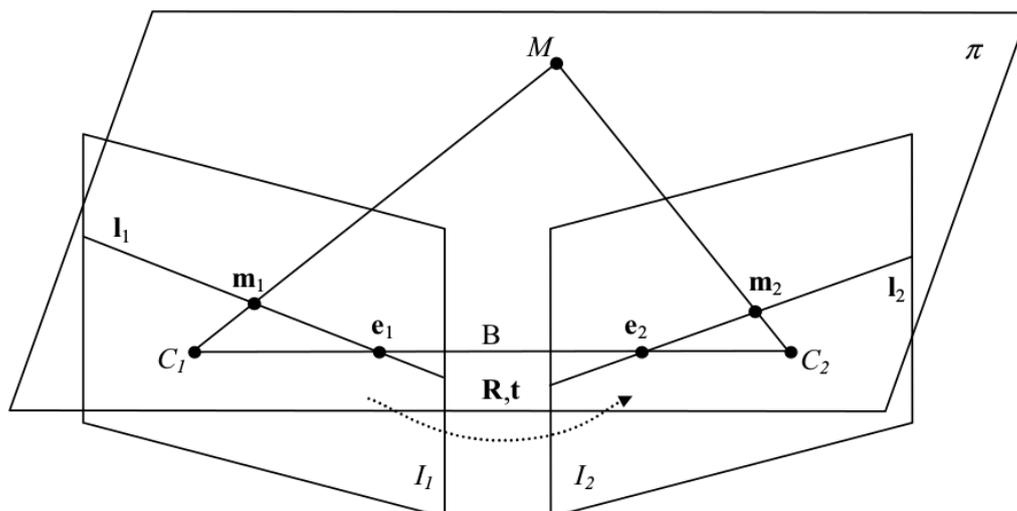
$$\mathbf{A} = \begin{bmatrix} a_u & \gamma & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{mit } a_u = f \cdot k_u \text{ und } a_v = f \cdot k_v \quad (2.7)$$

### Tiefeninformationen aus Einzelbildern

Nach [33] können unendlich viele dreidimensionale Szenen eine konkrete, zweidimensionale Abbildung erzeugen, was die Rekonstruktion der Tiefe bzw. der dreidimensionalen Szene aus einem einzelnen Abbild erschwert. Viele der in Frage kommenden Szenen sind physikalisch nicht sinnvoll, weshalb die möglichen Lösungen durch Nebenbedingungen eingegrenzt werden können. Die globale Skalierung stellt jedoch eine bleibende Mehrdeutigkeit dar, welche ohne weitere Informationen nicht aufzulösen ist. [29] beschreibt beispielsweise Verfahren wie *Shape from Shading* oder *Shape from Texture*, welche relative Tiefeninformationen aus einem einzelnen Bild extrahieren. Die Anwendbarkeit auf realistische Szenen abseits von Laborbedingungen ist jedoch nicht gegeben. Eine mit steigender Rechenleistung und öffentlich verfügbaren und annotierten Datensätzen, wie z. B. [34], emporkommende Methode im Bereich *Computer Vision* stellt das maschinelle Lernen dar. Dies wird auch auf die Rekonstruktion von dreidimensionalen Szenen aus Einzelbildern angewendet und dabei werden beachtliche Fortschritte erzielt, wie beispielsweise [33] und [35] anhand von realistischen Datensätzen zeigen.

## 2.2 Die Geometrie zweier Perspektiven

Eine dreidimensionale Szene kann, wie in Abbildung 2.3 dargestellt, aus mehreren Perspektiven abgebildet werden. Dabei spielt es prinzipiell keine Rolle, ob dafür eine einzelne Kamera verwendet wird, welche die Szene zu unterschiedlichen Zeitpunkten aufnimmt, oder ein System aus mehreren Kameras, welches zeitlich synchronisiert

**Abbildung 2.3**

Die allgemeine Epipolargeometrie, entnommen aus [32].

arbeitet. Nach [30] kann ein Punkt  $M$  im Kamerakoordinatensystem  $C_1$  durch eine Rotation  $\mathbf{R}$  und Translation  $\mathbf{t}$  in das Kamerakoordinatensystem  $C_2$  überführt werden. Die Brennpunkte beider Kameras, mit dem als Basislänge bezeichneten Abstand  $B$ , definieren mit dem Punkt  $M$  die Epipolarebene  $\pi$ . Die Epipolarlinien  $\mathbf{I}_1$  und  $\mathbf{I}_2$  sind die Schnittgeraden der Bildebenen  $I_1$  und  $I_2$  mit der Epipolarebene, auf welchen auch die jeweiligen Abbildungen  $\mathbf{m}_1$  und  $\mathbf{m}_2$  liegen. Alle möglichen Punkte der Szene spannen eine Schar an Epipolarebenen auf, deren Epipolarlinien sich in der jeweiligen Bildebene im Epipol  $\mathbf{e}_1$  bzw.  $\mathbf{e}_2$  schneiden.

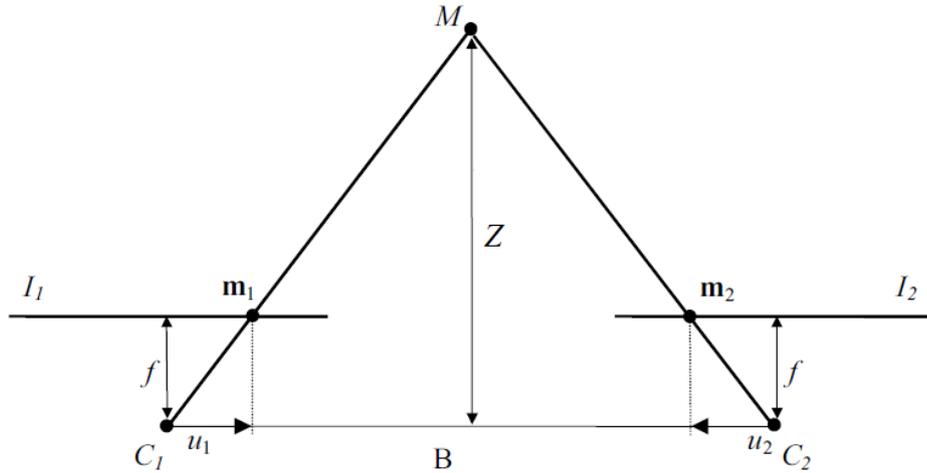
Die Epipolargleichung beschreibt mit Hilfe der Essential-Matrix  $\mathbf{E}$  die Beziehung der korrespondierenden Abbildungen beider Kameras, siehe Formel 2.8, wobei  $[\mathbf{t}]_{\times}$  die Matrixschreibweise des Kreuzprodukts mit  $\mathbf{t}$  darstellt [32].

$$\tilde{\mathbf{m}}_2^T \cdot \mathbf{E} \cdot \tilde{\mathbf{m}}_1 = 0, \quad \text{mit } \mathbf{E} = [\mathbf{t}]_{\times} \cdot \mathbf{R} \quad (2.8)$$

Sind die Bildebenen  $I_1$  und  $I_2$  nicht zueinander verdreht, d. h. die Drehmatrix  $\mathbf{R}$  entspricht der Einheitsmatrix, und die Kamerakoordinatensysteme sind nur horizontal zueinander verschoben, so spricht man von einer achsparallelen Stereogeometrie. Hierbei liegen die Epipole im Unendlichen, wodurch Epipolarlinien parallel zueinander ausgerichtet sind. Betrachtet man die Bildebenen in diskreten Sensorkoordinaten, so fallen die Epipolarlinien nun auf einzelne Zeilen des Bildsensors. Dies bedeutet, dass die korrespondierenden Abbildungen  $\mathbf{m}_1$  und  $\mathbf{m}_2$  sich in der gleichen Bildzeile befinden, nur der Spaltenindex unterscheidet sich.

## 2.3 Binokulare Stereokameras

Wie in Kapitel 2.2 beschrieben, besteht ein achsparalleles Stereosystem aus zwei parallel ausgerichteten Kameras. Sie sind per Definition nur in der Horizontale zueinander verschoben und nicht gegeneinander verdreht. In der Praxis sind diese Bedingungen für ein binokulares Stereokamerasystem nicht erfüllt [36], jedoch können die Kameras mittels Rektifikation virtuell in ein achsparalleles System transformiert



**Abbildung 2.4**

Achsparalleles Stereosystem, entnommen aus [32] (Notation angepasst).

werden [37]. In Abbildung 2.4 ist ein achsparalleles Stereosystem abgebildet. Die Bildebenen  $I_1$  und  $I_2$  sowie die optischen Zentren  $C_1$  und  $C_2$  sind horizontal um die Basislänge  $B$  verschoben. Die optischen Zentren sind um die Brennweite  $f$  von den Bildebenen entfernt. Die Abbildungen  $m_1$  und  $m_2$  des Punktes  $M$  weisen auf den Bildebenen einen Versatz  $\delta$  auf, der Disparität genannt wird. In achsparallelen Stereosystemen befinden sich beide Abbildungen  $m_1$  und  $m_2$ , dargestellt in Sensorkoordinaten, in derselben Bildzeile. Es gilt für die Sensorkoordinaten  $u$  und  $v$  beider Abbildungen folgende Formel 2.9.

$$u_2 = u_1 - \delta, \quad \text{und } v_2 = v_1, \quad \text{mit } \mathbf{m}_{1/2} = \begin{bmatrix} u_{1/2} \\ v_{1/2} \end{bmatrix} \quad (2.9)$$

Der Versatz ist somit rein horizontal und berechnet sich nach Formel (2.10).

$$\delta = u_1 - u_2 \quad (2.10)$$

Die Disparität wird in Bildpunkten angegeben, und kann durch den Faktor  $k_u$ , welcher die Breite eines Bildpunktes auf dem Bildsensor beschreibt, in Meter umgerechnet werden. Nach dem Strahlensatz ergibt sich somit für die Tiefe  $Z_c$  des Punktes  $M$  in Kamerakoordinaten die Formel 2.11. Die durch die Abbildung einer Kamera verlorene Tiefeninformation eines Punktes in Weltkoordinaten kann somit durch die geometrischen Eigenschaften von zwei unterschiedlichen Perspektiven auf diesen Punkt rekonstruiert werden.

$$Z_c = \frac{B \cdot f}{\delta \cdot k_u} \quad (2.11)$$

Wird ein Objekt, welches aus einer Punktmenge besteht, auf beiden Bildebenen dargestellt, muss für jeden Punkt der Punktmenge eines Bildes der korrespondierende Punkt auf dem anderen Bild ermittelt werden. Dieser Sachverhalt nennt sich Korrespondenzproblem, und nur dessen Lösung erlaubt die Berechnung der Disparitäten und folglich der 3D-Koordinaten der Punktmenge und somit des Objekts.

Die Komplexität dieses Problems ergibt sich durch die verschiedenen Ansichten der Bilder. So können z. B. einzelne Bildpunkte des Objekts in einer der beiden Ansichten verdeckt sein, womit keine Korrespondenz existiert. Weiterhin lassen sich schwach texturierte oder periodisch wiederholende Bildbereiche nur schwer punktgenau einander zuordnen. Die achsparallele Anordnung des Stereosystems reduziert die Komplexität des Korrespondenzproblems, da die Abbildungen  $\mathbf{m}_1$  und  $\mathbf{m}_2$  in derselben Bildzeile auftreten. Somit wird das ursprünglich zweidimensionale Problem auf eine Dimension reduziert [32] [37] [36]. Sind die Sensorkoordinaten und die Disparität eines Punktes  $\mathbf{m}$  bekannt, können mit Hilfe der Reprojektionsmatrix  $\mathbf{Q}_R$  die Koordinaten des Punktes  $\mathbf{M}$  berechnet werden.

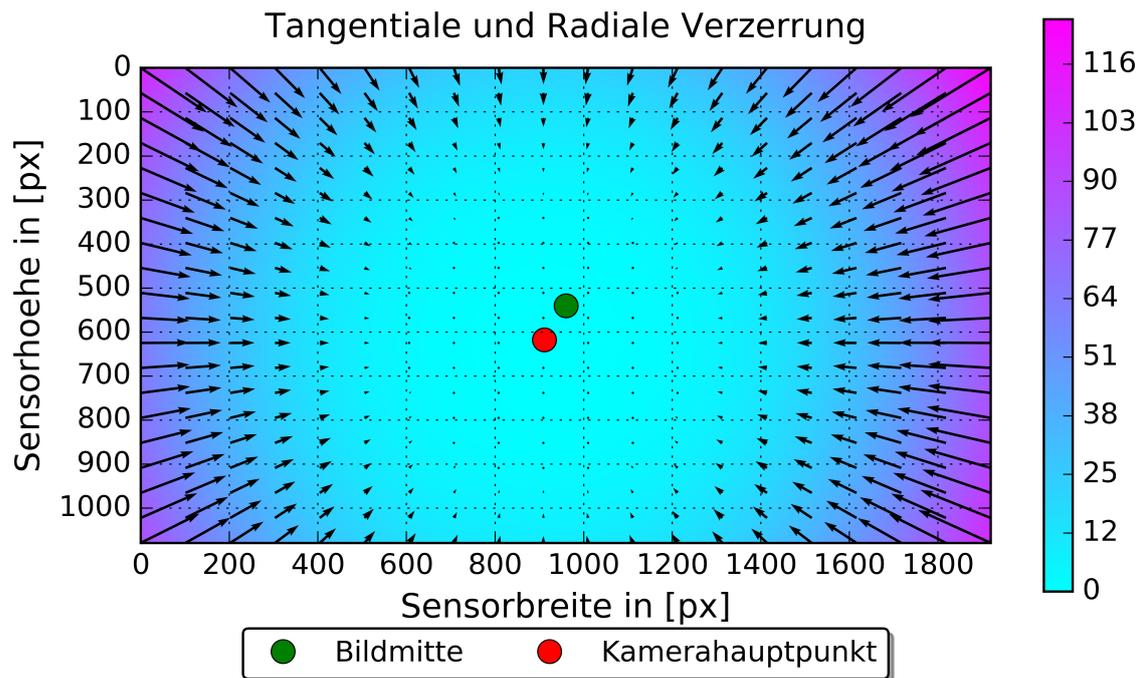
$$\tilde{\mathbf{M}} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \mathbf{Q}_R \cdot \begin{bmatrix} u \\ v \\ \delta \\ 1 \end{bmatrix}, \quad \text{mit } \mathbf{Q}_R = \begin{bmatrix} 1 & 0 & 0 & -u_0 \\ 0 & 1 & 0 & -v_0 \\ 0 & 0 & 0 & f/k_u \\ 0 & 0 & 1/B & 0 \end{bmatrix} \quad (2.12)$$

## 2.4 Kamerakalibration

Um Punkte im dreidimensionalen Raum eindeutig rekonstruieren zu können, muss das Stereosystem vollständig kalibriert sein [36]. Zur Verwendung des in Kapitel 2.1 eingeführten Lochkammermodells müssen die intrinsischen Parameter der Matrix  $\mathbf{A}$  mittels Kamerakalibration für beide Kameras eines Stereosystems bestimmt werden. Zusätzlich müssen durch Stereokalibration die Rotation und Translation der beiden Kamerakoordinatensysteme zueinander bekannt sein, um mittels Rektifikation ein virtuell achsparalleles Stereosystem erzeugen zu können.

Bisher wurden Verzeichnungseffekte durch die Verwendung von Linsen in Kameras vernachlässigt. Diese können ebenso modelliert und die benötigten Parameter per Kalibration bestimmt werden. [36] beschreibt die radiale sowie die tangentialen Verzerrung als Hauptbestandteile der Linsenverzeichnung, deren Modellierung in [38] beschrieben wird. Die radiale Verzerrung resultiert dabei aus der Linsenform, beträgt im Kamerahauptpunkt null und nimmt mit der radialen Entfernung zum selbigen zu. Eine schiefe Ausrichtung der Linse zur Sensorebene führt zur tangentialen Verzerrung, welche eine elliptische Form um den Kamerahauptpunkt annimmt. Die kameraspezifische Verzeichnung würde ohne Korrektur zu einem erheblichen Fehler in der Rektifikation und somit der Disparitätenberechnung führen, weshalb eine Korrektur unabdingbar ist. Abbildung 2.5 zeigt den Effekt der Verzeichnung über die Sensorebene für eine der in dieser Arbeit verwendeten Kameras. Die Pfeile stellen die nicht maßstabsgetreue Verschiebung einer Abbildung dar. Insbesondere der radiale Anteil ist zu erkennen, dessen Maximalwert in diesem Beispiel um den Faktor 100 größer ist als das tangentiale Maximum. Letztere ist somit in der Abbildung kaum auszumachen. Die farbliche Kodierung beschreibt den Absolutwert der kombinierten Verzeichnung, welche am Rand des Bildsensors Werte größer als 100 Bildpunkte annimmt.

Es existiert eine Vielzahl verschiedener Methoden für die Kalibrierung von Kameras, wobei das verwendete Abbildungsmodell eine wichtige Rolle spielt. [39] und [40] geben einen Überblick über den Stand der Technik. In dieser Arbeit wird die

**Abbildung 2.5**

Verzerrung der Kameraabbildung durch tangentiale und radiale Verzerrung entlang der schwarzen Pfeile (nicht maßstabsgetreu). Aufgrund der geringen tangentialen Verzerrung ist diese kaum zu erkennen.

Methode von [41] verwendet, welche sich durch ihre Einfachheit in der Durchführung sowie durch ihre Robustheit auszeichnet. Hierbei wird ein Schachbrettmuster als ebenes Kalibrierobjekt verwendet, welches aus mindestens zwei unterschiedlichen, frei wählbaren Perspektiven aufgezeichnet wird. Dabei muss die jeweilige Position der Kamera zum Kalibrierobjekt nicht bekannt sein. Durch diese Methode werden die intrinsischen Parameter der Matrix  $\mathbf{A}$  des Lochkameramodells, die Verzerrungsparameter sowie Translation und Rotation für jede Aufzeichnung des Kalibrierobjekts bezogen auf das Kamerakoordinatensystem bestimmt. Die verwendete Implementierung der freien Programmiersbibliothek „OpenCV“ [42] verwendet das Verzerrungsmodell aus [38] anstatt dem in [41] beschriebenen Modell, welches keine tangentiale Verzerrung berücksichtigt.

Zur Bestimmung der Rotation und Translation zwischen beiden Kamerakoordinatensystemen des Stereosystems wird ebenfalls das planare Kalibrierobjekt verwendet [36]. Wird dieses Objekt gleichzeitig von beiden Kameras aufgezeichnet, so können die mittels der oben erwähnten Kamerakalibration bestimmten Transformationen zwischen Kalibrierobjekt und Kamerakoordinatensystem verwendet werden, um die Transformation zwischen beiden Kamerakoordinatensystemen zu bestimmen. Legt man das frei wählbare Weltkoordinatensystem in eines der beiden Kamerakoordinatensysteme, so sind alle benötigten Transformationen eindeutig bestimmt.

## 2.5 Systemdesign und Einflüsse der Parameterwahl

Ein Stereokamerasystem wird durch drei primäre Designparameter bestimmt. Dies sind die Brennweite der Kameras, die Basislänge sowie die Auflösung des Bildsensors. Die Parameter haben u. a. Auswirkungen auf die Tiefenauflösung, die Komplexität beim Lösen des Korrespondenzproblems und auf die Größe der Blindzone vor dem Stereokamerasystem [43]. Durch den Versatz der Kameras um die Basislänge entsteht, in Abhängigkeit des Bildwinkels, eine frontale Blindzone. Dies ist ein Bereich vor den Kameras, der nicht auf beiden Bildsensoren abgebildet wird. Folglich können für diesen Bereich keine korrespondierenden Punkte auf den Bildebenen gefunden werden. Die Größe der Blindzone ist direkt proportional zur Brennweite und der Basislänge. Eine Vergrößerung der Basislänge führt weiterhin zu einer Verbesserung der Tiefenauflösung, wirkt sich durch die größere Veränderung in der Perspektive jedoch negativ auf die Korrespondenzanalyse aus [44]. Durch die Vergrößerung der Brennweite und der Sensorauflösung kann die Tiefenauflösung ebenfalls verbessert werden. Ersteres führt jedoch zu einem kleineren Sichtfeld, zweiteres vergrößert den möglichen Disparitätenbereich und somit die Komplexität des Korrespondenzproblems. Die Tabelle 2.1 fasst die Auswirkungen einer Änderung der Parameter zusammen. Dabei sind positive Effekte grün markiert, negative Effekte rot.

**Tabelle 2.1**

Auswirkungen der Parameter eines Stereokamerasystems nach [43]

Auswirkung auf:	Tiefenfehler	Blindzone	Disparitätenbereich
Größere Brennweite	↓	↑	↑
Größere Basislänge	↓	↑	↑
Größere Auflösung	↓	-	↑

Bei gegebenen Kameras sind die Parameter nur bedingt wählbar. Die in dieser Arbeit verwendeten Kameras des Typs „HERO3+ Black“ des Herstellers GoPro bieten die Möglichkeit, verschiedene Auflösungen, Sichtfelder und Bildraten zu wählen. Die Basislänge ist bei der Verwendung des „Dual HERO System“, ein Adapter zur Synchronisation von zwei Kameras, theoretisch frei wählbar. Praktisch wird die Basislänge durch die Länge des Verbindungskabels limitiert. Abbildung 2.6 zeigt den Tiefenfehler aufgrund der Quantisierung der Disparität für die gewählte Sensorauflösung von  $1920 \times 1080$  px und einer Brennweite von 2,6 mm bei variabler Basislänge. Der mit steigender Entfernung exponentiell steigende Fehler ist leicht erkennbar, ebenso wie der positive Einfluss einer größeren Basislänge. Die Designparameter des verwendeten achsparallelen Systems werden in Tabelle 2.2 dargestellt.

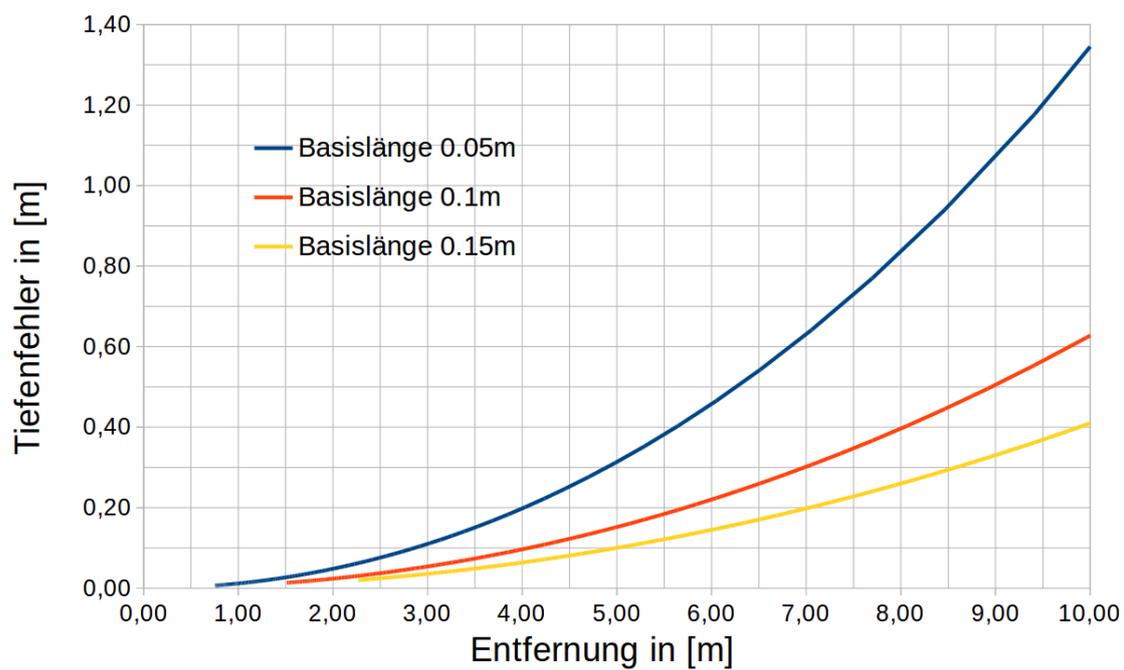
## 2.6 Anwendung der Grundlagen in dieser Arbeit

Die in Kapitel 2 erarbeiteten Grundlagen bilden die Basis für die Entwicklung eines Analysetools in Kapitel 3. Abbildung 2.7 zeigt die benötigten Prozesse auf. Blaue Rechtecke kennzeichnen die einzelnen Prozesse, während die Linien den Datenfluss beschreiben. Die durchgezogenen grünen Linien stellen die Verarbeitungskette der

**Tabelle 2.2**

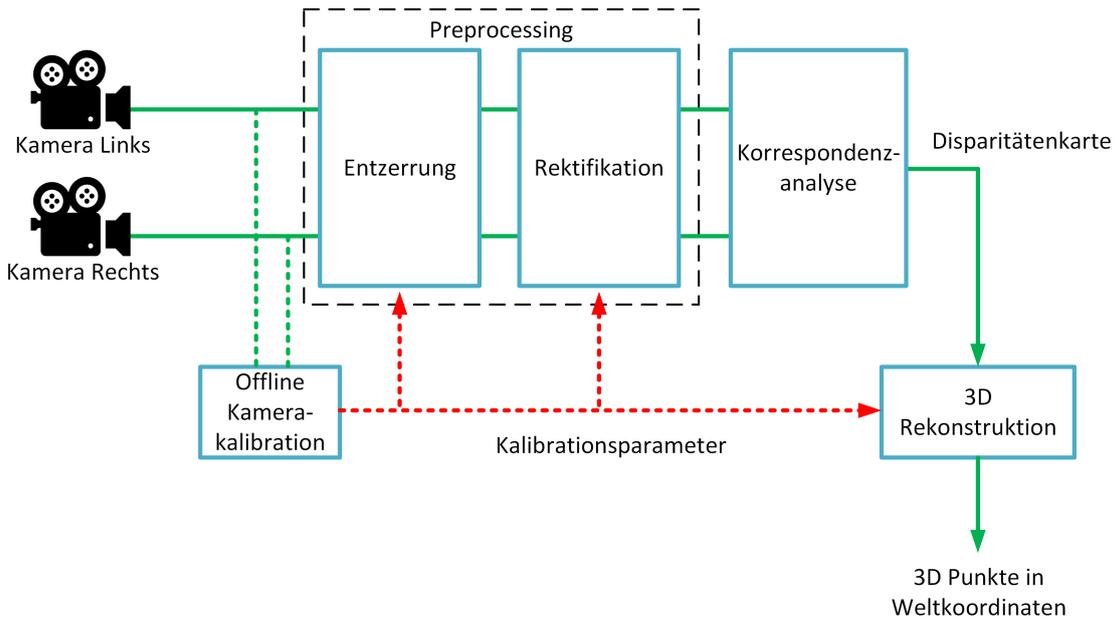
Designparameter des verwendeten achsparallelen Stereosystems

Parameter	Wert	Einheit
Brennweite	2,6	mm
Basislänge	0,112	m
Auflösung	1920 × 1080	px



**Abbildung 2.6**

Tiefenfehler des verwendeten Stereokamerasystems für verschiedene Basislängen.

**Abbildung 2.7**

Darstellung der Prozesskette: von Kamerarohdaten zu 3D-Punkten.

Bildrohdaten im zu entwerfenden Analysetool dar, während die gestrichelten grünen Linien die Bereitstellung der Kameradaten für die Kalibrierung kennzeichnen. Die rote Linie stellt die Verwendung von Kalibrationsparametern in den einzelnen Prozessen dar.

Das Ziel der Prozesskette ist die dreidimensionale Rekonstruktion einer beobachteten Szene. Hierfür werden Tiefenkarten bzw. Disparitätenkarten berechnet, welche wiederum benutzt werden können um Objekte zu extrahieren. Kennt man die Position der relevanten Objekte auf dem Bildsensor (*Region of Interest, ROI*) sowie die entsprechende Disparität im achsparallelen System, so können die dreidimensionalen Weltkoordinaten berechnet werden. Die Szene ist somit für den Zeitpunkt der Bildaufnahme beschrieben. Um nicht nur eine räumliche, sondern auch eine zeitliche Beziehung zwischen detektierten Objekten herzustellen, müssen diese in Folgeaufnahmen assoziiert werden. Die Objektdetektion sowie das Objekttracking werden in Kapitel 3 behandelt.

Die in Kapitel 2 beschriebene Kalibration findet *offline*, also nicht zeitlich parallel zur eigentlichen Prozesskette statt, was durch die benötigten Kalibrierobjekte bedingt ist. Die berechneten intrinsischen und extrinsischen Parameter werden den Prozessen des Analysetools zur Verfügung gestellt. Die Bereinigung der Kamerabilder von Linsenverzerrungen und die Rektifikation können dabei als *Preprocessing* zusammengefasst werden. Die Ausgangsdaten entsprechen nun den Bildern eines achsparallelen Stereosystems, wodurch sich die Komplexität der folgenden Korrespondenzanalyse auf eine Dimension reduziert. Die in diesem Schritt berechnete Disparitätenkarte erlaubt eine eindeutige Rekonstruktion der beobachteten dreidimensionalen Punkte im Weltkoordinatensystem, siehe Gleichung 2.12.



---

# Kapitel 3

## Aufbau und Funktion des Analysetools

Die fahrkontextbasierte Erkennung kritischer Situationen wurde in Kapitel 1 als primäres Ziel dieser Arbeit definiert. Die verwendeten Sensoren sind ein am Fahrradlenker befestigtes, in Fahrtrichtung ausgerichtetes Stereokamerasystem, dessen Eigenschaften in Kapitel 2.5 beschrieben werden und sich aufgrund der Variabilität der Parameter gut für diese Untersuchung eignet. Das Kamerasystem wurde entsprechend der Grundlagen in Kapitel 2 kalibriert, die Bilder entzerrt und rektifiziert.

Anhand dieser Daten sollen nun kritische Situationen im Sinne der EBA mit Hilfe des prototypisch zu entwickelnden Programms extrahiert werden. Für die Entwicklung werden bereits vorhandene Werkzeuge zur Datenanalyse im Bereich NCS und NDS vorgestellt. Anhand dieser Erkenntnisse sowie den Erläuterungen aus Kapitel 1 werden Kriterien an die Funktionalität definiert, woraus sich der grundlegende Aufbau des Analysetools ableitet. Für alle Elemente des Programms wird sukzessive der Stand der Technik erläutert und eine Auswahl des jeweils verwendeten Algorithmus getroffen.

### 3.1 Bisherige Arbeiten

In Kapitel 1 wurde aufgezeigt, dass sich die Auswertung von NDS-Daten bisher weitestgehend auf die manuelle Analyse beschränkt. Der Einsatz von Software ist somit auf Programme zur Annotation und Aggregation der Daten sowie zur statistischen Analyse limitiert. Während für letzteres eine Vielzahl kommerzieller Programme wie „Matlab“ oder „SPSS Statistics“ zur Verfügung steht, werden für die Annotation und Aggregation meist Eigenentwicklungen von Forschergruppen verwendet, um die spezifischen Anforderungen zu erfüllen [25]. Ein im NCS-Umfeld verwendetes Programm, welches z. B. in [14] verwendet wurde, stellt „ELAN“ [45] dar.

Es existieren verschiedene Studien, welche Videodaten aus dem Verkehrsgeschehen automatisiert auswerten, z. B. [46] und [47]. [46] stellt eine Open Source Videoanalyse Software vor, welche Videos aus einer statischen Vogelperspektive automatisch auswertet und Trajektorien von detektierten Objekten erstellt. Mit Hilfe des Programms wurden eine Vielzahl weiterer Studien ermöglicht, wie z. B. [48] [49] [50], deren manuelle Auswertung nicht tragbar gewesen wäre. So konnte z. B. in der Studie [49] die Bewegung von 80 000 motorisierten Fahrzeugen in Kreisverkehren analysiert

werden. Eine ähnliche Anwendung für bewegte Kameras, wie im Fall einer NDS, existiert nach Wissen des Autors nicht.

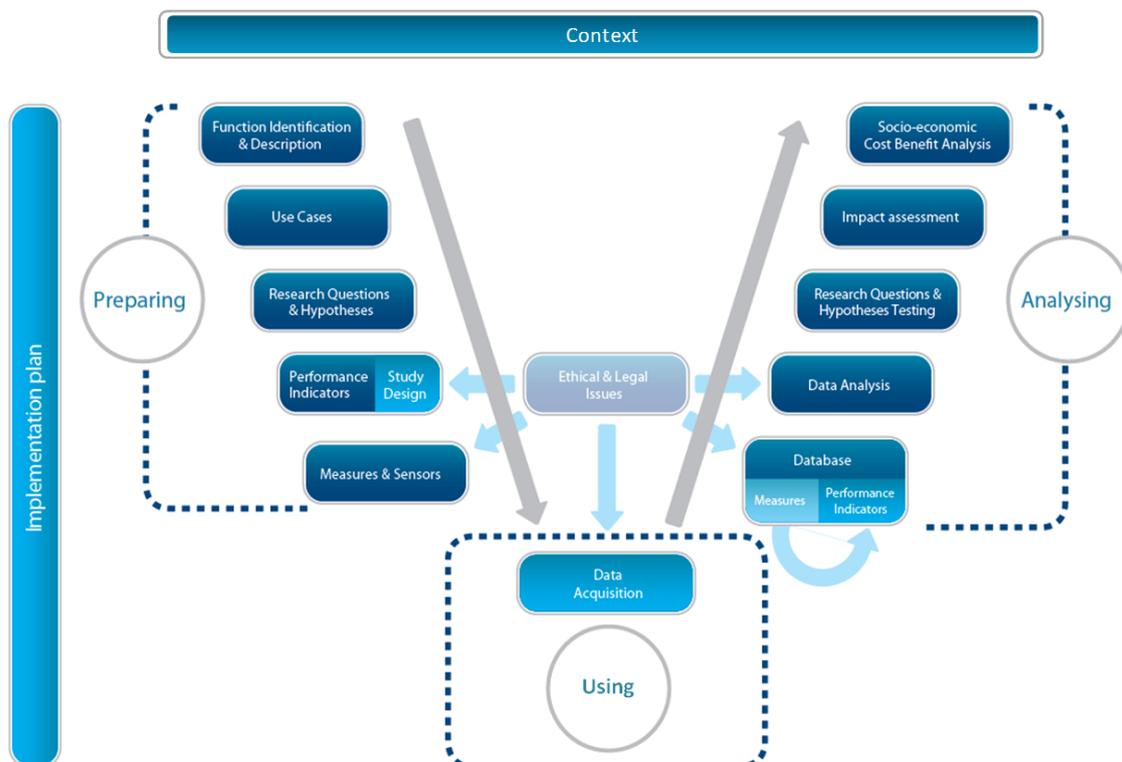
Da die Anforderungen einer Datenaufzeichnung und -analyse einer NDS, und insbesondere einer NCS, sich auf dem ersten Blick mit denen von ADAS und AD überschneiden, bedarf es an dieser Stelle einer Abgrenzung. Für die Anwendung im ADAS- und AD-Bereich werden meist redundante Sensorkonzepte gewählt. Die Analyse der Sensordaten ist daher nicht nur auf Einzelsensoren wie z. B. Kameras konzentriert. Solche werden nur für unkritische Systeme gewählt, deren Störanfälligkeit voraussichtlich keine schwerwiegenden Folgen hätte, wie z. B. Abstandsregelautomaten (*Adaptive Cruise Control*, ACC) oder Spurverlassenswarnung (*Lane Departure Warning*, LDW). Andere ADAS mit kritischeren Funktionen greifen auf Sensorfusion zurück, um die Stärken und Schwächen einzelner Sensoren auszugleichen. Beispielhaft kann hier eine Objektklassifikation durch Kameradaten und Entfernungs- und Geschwindigkeitsmessung des Objekts durch RaDAR oder LiDAR genannt werden [51]. Die im maschinellen Sehen verbreiteten Standards finden auch hier Anwendung. Eine Sammlung von Funktionen, welche im kommerziellen wie auch wissenschaftlichen Umfeld häufig verwendet wird, ist die „OpenCV“-Bibliothek [42]. Die Funktionen von Fahrerassistenzsystemen können auf spezielle Anwendungsfälle getrimmt werden, welche dadurch hohe Detektionsraten erreichen. Ein Beispiel ist die Erkennung von PKW-Heckansichten auf Autobahnen und gut ausgebauten Landstraßen [52].

Aufgrund der Sensorkonzepte und der spezifischen Anwendungsfälle sind Auswertekonzepte des ADAS- und AD-Umfelds oft nicht direkt auf NDS und NCS übertragbar. Wie in Kapitel 1 beschrieben, untersagt die Methodik einen Einsatz von aufdringlichen Sensorkonzepten, was sich hierbei auf das Aussehen sowie die veränderten Fahreigenschaften durch eine Fahrzeugumrüstung bezieht. Versuche mit Fahrrädern stellen hier sicherlich eine extreme Ausprägung dieses Dilemmas dar. Trotzdem können Entwicklungen, welche sich auf Einzelsensorlösungen beziehen und eine entsprechende Funktionalität bieten, auf die automatisierte Auswertung von NDS / NCS-Daten angewendet werden. Dies betrifft neben Bildverarbeitungsalgorithmen auch Ansätze für die Verfolgung von Objekten im Straßenverkehr. Weiterhin können sich ADAS und AD-Systeme hinsichtlich ihrer Auslegung bezüglich Spezifität und Sensitivität von Auswertungsmethoden für NCS unterscheiden. Bei der Verwendung eines automatischen Notbremssystems (*Automatic Emergency Braking*, AEB) in einem Serienfahrzeug kann beispielsweise nur eine minimale falsch-positive Ereigniswahrscheinlichkeit akzeptiert werden. Vermehrte Vollbremsungen des Fahrzeugs ohne tatsächlich existierende Gefahr würde die Akzeptanz der Nutzer\*innen und somit die Verkaufszahlen des Systems zu Recht negativ beeinträchtigen, während eine erhöhte falsch-negative Ereigniswahrscheinlichkeit den Nutzen des Systems in Frage stellt. Im NDS-Kontext kann sich dieses Dilemma anders darstellen. Aufgrund der geringen Anzahl von kritischen Situationen über die Laufzeit einer Studie muss die Wahrscheinlichkeit für eine ausbleibende Detektion minimal sein. Dies führt zwangsläufig zu erhöhten Falschdetektionen, welche im Auswertungsprozess entsprechend behandelt werden müssen. Der Nutzen einer automatisierten Analyse von NCS-Daten muss sich folglich anhand der (Falsch-) Detektionsrate der relevanten Events messen lassen.

## 3.2 Anforderungen an die Funktionalität

Die prototypische Implementierung zur Analyse von NCS-Versuchsdaten muss Anforderungen erfüllen, deren Definition mit Hilfe des bisher erarbeiteten Wissens erstellt werden. Betrachtet man den Gesamtprozess einer Feldstudie, so kann dieser nach [13] als V-Prozess dargestellt werden. Der eigentliche Anwendungsfall ist hierbei nicht eine NDS oder NCS, sondern *Field Operational Tests* (FOT), welche eine neue Funktionalität, z. B. ein Fahrerassistenzsystem, im Feld prüft und dessen Auswirkungen auf Zielkonflikte untersucht. Dennoch ist der Gesamtprozess nach den Autor\*innen des Prozesses auch auf NDS und NCS übertragbar.

Das in Abbildung 3.1 dargestellte „FESTA V“ kann in drei Teile aufgelöst werden. Der linke Teil der V-Struktur, hier gekennzeichnet als „*Preparing*“, beschreibt alle vorbereitenden Maßnahmen. Dabei wird für FOTs die zu untersuchende Funktion und deren Anwendungsfälle definiert sowie die zu untersuchenden Hypothesen formuliert. Aus diesen wird das eigentliche Studiendesign abgeleitet. Die Wahl der Sensoren ergibt sich dabei aus den zu berechnenden PIs, welche die Bearbeitung der Hypothesen bzw. Beantwortung der Forschungsfragen ermöglichen sollen, und den dafür benötigten Rohsignalen. Im Falle von NDS/NCS existiert keine zu untersuchende Funktion, vielmehr werden Forschungsfragen definiert, deren Beantwortung die zentrale Motivation darstellt. Das Studiendesign muss sich an den Anforderungen der naturalistischen Methodik orientieren, mit allen bereits in Kapitel 1.1 beschriebenen Konsequenzen für das DAS und die Sensorik.



**Abbildung 3.1**

Das „FESTA V“ als Prozessdefinition für Feldstudien [13].

Die Durchführung der Datenaufzeichnung wird im „FESTA V“ als „*Using*“ gekennzeichnet. Die dort akquirierten Daten werden im Bereich „*Analysing*“ verarbeitet. Die einzelnen Schritte der Datenanalyse in Feldstudien bestehen nach [13] aus folgenden Punkten, welche in einer leicht abgewandelten Form auch im „FESTA V“ wiederzufinden sind:

1. Analyse der Datenqualität
2. Datenverarbeitung
3. PI-Berechnung
4. Hypothesentest
5. Verallgemeinerung der Ergebnisse

Die Prozesskette zur Detektion von relevanten Events deckt für diese Umsetzung nur die Datenverarbeitung ab. Die Analyse der Datenqualität beinhaltet kritische Punkte einer Versuchsauswertung, wie z. B. Tests auf fehlende Daten, korrekte Einheiten oder Erfüllung der Anforderungen. Dennoch werden diese Unterprozesse an dieser Stelle übergangen, um den Fokus auf die Datenverarbeitung zu legen. Die letzten drei Schritte der Datenanalyse stellen die eigentliche Auswertung der Studie dar, beginnend mit der PI-Berechnung, welche durch die Konfliktindikatoren zur Eventdetektion eine Überschneidung aufweisen können. Die Hypothesentests und die Verallgemeinerung der Ergebnisse schließen die Auswertung ab.

Die Datenverarbeitungskette nach [13] beinhaltet folgende Unterprozesse, welche nicht zwangsläufig Teil des Gesamtprozesses sein müssen:

- a. Datenfilterung
- b. Erzeugen abgeleiteter Daten
- c. Eventannotation bzw. -detektion
- d. Datenreorganisation

Bezieht man diese Einzelschritte auf die in Abbildung 2.7 dargestellte Prozesskette zur 3D-Rekonstruktion, so entspricht die Datenfilterung dem *Preprocessing* der Kameradaten. Die Erstellung einer Disparitätenkarte und alle darauffolgenden Schritte bis zur signalbasierten Eventdetektion zählen zur Erzeugung abgeleiteter Daten. Dies beinhaltet neben der 3D-Rekonstruktion auch die Objektdetektion und -verfolgung sowie die Berechnung von Konfliktindikatoren. Erst letztere ermöglichen direkt die Eventannotation bzw. -detektion anhand definierter Schwellenwerte. Die Datenreorganisation ist hier nicht vorgesehen und kann bei Bedarf in Abhängigkeit der zu berechnenden PIs implementiert werden.

Zuzüglich zur bisherigen Eingrenzung des Datenanalyseprozesses nach [13] sollen die zu erfüllenden Aufgaben weiter abgegrenzt werden. Der Fokus von Kapitel 3 liegt auf der Funktionalität der oben eingegrenzten Teilschritte. Aufgrund der prototypischen Implementierung existieren keine Kriterien an die Laufzeit, obwohl der zeitliche Aspekt der Datenauswertung, wie in Kapitel 1 beschrieben, bei der Bewertung des Nutzens eines solchen Programms eine wichtige Rolle spielt. Zusätzlich werden

keine Lösungen für optimierte Konzepte der Datenspeicherung und -bereitstellung erarbeitet.

Die Eingangsdaten des Programms sind die Videodaten des verwendeten Stereokamerasystems sowie die entsprechenden intrinsischen und extrinsischen Kalibrationsparameter. Ausgabedaten sind die auf detektierte Objekte bezogene Konfliktindikatoren, welche in Abhängigkeit von zu definierenden Schwellenwerten die Kritikalität beschreiben. Dabei wird an dieser Stelle die Definition des Objekts bewusst nicht weiter eingegrenzt. Weiterhin werden die abgeleiteten Daten als Output definiert, welche der Bewertung der Kritikalität zugrunde liegen und in Kapitel 3.7 definiert werden. Somit kann für die Videodaten des Inputs ein kontinuierliches Signal bereitgestellt werden, welches kritische Situationen markiert und somit direkt die Reduktion des Datenmaterials auf relevante Events zulässt. Die mit dem Signal verknüpften Konfliktindikatoren und Objektdaten bieten die Möglichkeit der Überprüfung, als auch der Weiterverarbeitung der Daten für PIs.

Aufgrund der Seltenheit der kritischen Events im zu untersuchenden Datenmaterial kann davon ausgegangen werden, dass eine ausbleibende Detektion (FN, *False Negative*) gegenüber einer Fehldetektion (FP, *False Positive*) der schwerwiegendere Fehler ist. Das tatsächlich aufgetretene Event wird durch das FN-Ereignis nicht in die EBA einbezogen und ist somit für alle folgenden Verarbeitungsschritte nicht verfügbar. Eine FP-Klassifizierung markiert den Event jedoch fälschlicherweise, was ihn in die EBA einfließen lässt. Durch Nachbearbeitung, welche auch manuell per Sichtung geschehen kann, besteht die Möglichkeit die Fehlklassifizierung zu berichtigen. Bei manueller Nachbearbeitung sollte die FP-Rate in einem Maß bleiben, welche den Nutzen der automatisierten Eventdetektion nicht in Frage stellt. Wie dieses Maß definiert ist, hängt dabei vom Anwender ab und wird in dieser Arbeit nicht bestimmt.

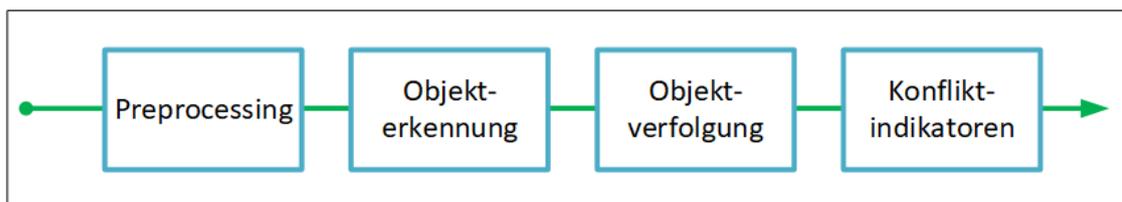
Im Folgenden werden die Kriterien an das Programm zusammengetragen:

- Eingangsdaten:
  - Videodaten
  - Kalibrationsdaten
- Ausgangsdaten:
  - Objektbezogene Kritikalität
  - Objektbezogene Konfliktindikatoren
- Prozess:
  - Datenfilterung
  - Erzeugen abgeleiteter Daten
  - Eventdetektion
- Eventdetektionsrate
  - Minimale False Negatives
  - Vertretbare False Positives

Zusätzlich zu den hier aufgestellten allgemeinen Kriterien existieren weitere Anforderungen an die spezifischen Teilprozesse. Diese werden in den jeweiligen Abschnitten dieses Kapitels behandelt.

## 3.3 Aufbau und Workflow

Der Prozess zur automatisierten Detektion von kritischen Situationen beinhaltet die in Kapitel 3.2 vorgestellten Teilschritte. Die Datenfilterung und das Erzeugen abgeleiteter Daten können grob in vier Einzelschritte gegliedert werden, welche in Abbildung 3.2 dargestellt sind. Nach dem *Preprocessing* der Inputdaten werden für jeden Zeitschritt mögliche Objekte und deren Bildkoordinaten sowie Disparität extrahiert. Eine zeitliche Koppelung dieser Objekte findet in der Objektverfolgung statt, welche unter anderem einen zeitlichen Verlauf der Weltkoordinaten des Objekts als sogenannten *track* liefert. Je nach Umsetzung können hier auch weitere Zustandsvariablen, wie z. B. die Objektgeschwindigkeit geschätzt werden. Aus diesen Objektzuständen werden die Konfliktindikatoren abgeleitet, welche der abschließenden Eventdetektion als Input dienen.

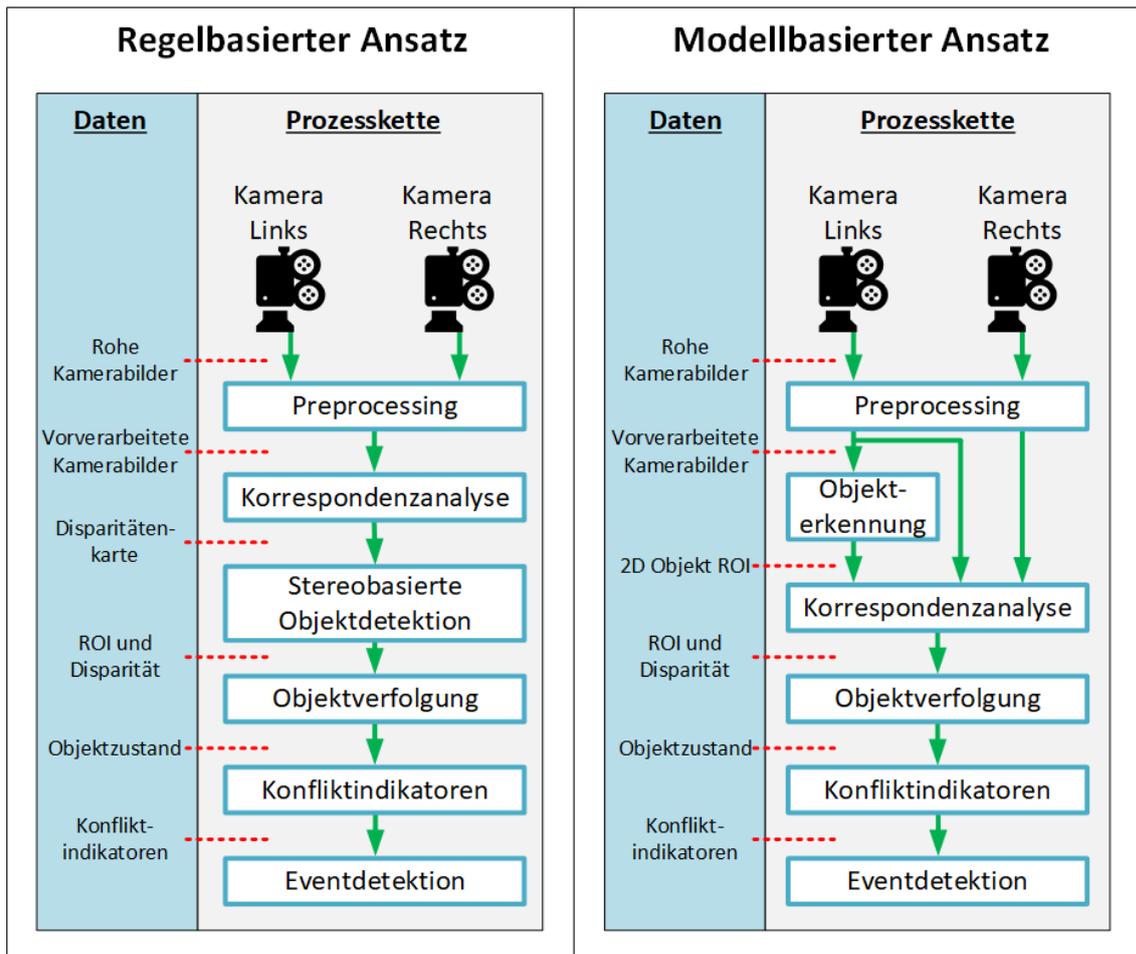


**Abbildung 3.2**

Der generalisierte Workflow, welcher von beiden Ansätzen durchlaufen wird

Diese vier Schritte sind in der vorliegenden Arbeit umgesetzt, wobei für die Objektdetektion zwei verschiedenen Ansätze zum Einsatz kommen. Zum einem ist dies ein regelbasierter Ansatz, der eine Disparitätenkarte auf vertikale Ebenen untersucht, um anhand verschiedener Annahmen Objekte zu extrahieren. Die zweite untersuchte Methode basiert auf maschinellem Lernen, wobei ein neuronales Netz zum Einsatz kommt. Beide Arten der Objektdetektion unterscheiden sich grundsätzlich hinsichtlich ihrer Leistungsmerkmale und den gestellten Anforderungen sowie in den einzelnen Teilschritten. Abbildung 3.3 stellt beide Prozessketten sowie den Datenfluss dar. Auf der linken Seite wird die regelbasierte Objektdetektion verwendet, auf der rechten Seite der Ansatz des maschinellen Lernens. Mit Hilfe der rot gestrichelten Linien werden die Ein- und Ausgangsdaten der einzelnen Teilschritte gekennzeichnet.

In der linken Prozesskette wird nach dem *Preprocessing* eine Disparitätenkarte durch die Korrespondenzanalyse erzeugt. Anhand dieser kann die stereobasierte Objektdetektion verschiedene ROIs aus dem Bild extrahieren, welche jeweils ein potenzielles Objekt beinhalten. Somit ist die Position der Objekte auf der Bildebene sowie deren Disparität bestimmt. Diese Daten bilden den Eingang der Objektverfolgung, welche mit Hilfe von Kalman-Filter die Weltkoordinaten und die Geschwindigkeit des Objekts über die Zeit schätzt. Wie im generalisierten Workflow in Abbildung 3.2 beschrieben, kann mit diesen Informationen die Berechnung der Konfliktindikatoren sowie die Eventdetektion von statten gehen.



**Abbildung 3.3**

Die Prozessketten der beiden verfolgten Ansätze. Links: Regelbasierte Objektdetektion, Rechts: Objekterkennung durch maschinelles Lernen

Der rechte Teil der Abbildung 3.3 verwendet das maschinelle Lernen zur Objekterkennung. Diese basiert auf den vorverarbeiteten Kamerabildern der linken Kamera und extrahiert daraus pixelgenaue ROIs der einzelnen im Bild vorhandenen Objekte. Eine Eigenschaft der auf Klassifikatoren basierten Objekterkennung ist das benötigte Training. Dies führt dazu, dass nur Objekte erkannt werden können, welche auch trainiert wurden. Um die ROIs in Kalman-Filter zur Objektverfolgung übergeben zu können, müssen diese mit der entsprechenden Disparität ergänzt werden. Hierfür muss wiederum eine Disparitätenkarte angefertigt werden, wobei jedoch nur die Bildregionen der detektierten Objekte von Belang sind. Die Folgeschritte entsprechen dem Workflow der linken Prozesskette.

In beiden Prozessketten spielt die Korrespondenzanalyse eine tragende Rolle, weshalb der Prozess in Kapitel 3.4 näher erläutert wird. Dort findet auch die Auswahl und Beschreibung des verwendeten Algorithmus zur Korrespondenzanalyse und Erstellung einer Disparitätenkarte statt. Die Anforderungen an die Objektdetektion und die darauf basierende Auswahl der Algorithmen wird in Kapitel 3.5 ausführlich behandelt. Die Objektverfolgung und Berechnung der Konfliktindikatoren werden in Kapitel 3.6 und 3.7 beschrieben.

## 3.4 Erstellung der Disparitätenkarte

Der Begriff der Disparität wurde erstmals in Kapitel 2.3 erläutert. Wird ein Punkt  $\mathbf{M}$  auf beiden Bildern des achsparallelen Stereosystems abgebildet, so sind diese resultierenden Punkte  $\mathbf{m}_1$  und  $\mathbf{m}_2$  aufgrund der Epipolargeometrie nur horizontal innerhalb derselben Bildzeile verschoben. Wird diese als Disparität bezeichnete Verschiebung für alle Pixel ermittelt, entsteht eine dicht besetzte (*dense*) Disparitätenkarte. Die Karte ist dünn (*sparse*) besetzt, wenn nur für einzelne Weltpunkte, im Bild oft repräsentiert durch möglichst eindeutige Merkmale, die Disparität eingetragen wird. Auch wenn das Korrespondenzproblem durch die Epipolargeometrie auf ein eindimensionales Problem reduziert wurde, weist die Erstellung einer Disparitätenkarte noch viele Schwierigkeiten auf. So können Weltpunkte in einer Perspektive verdeckt sein, wodurch keine korrespondierende Abbildung existiert. Weiterhin können Pixel in schwach texturierten Bildregionen oder aufgrund von Sensorrauschen einander nur schwer zugeordnet werden, was zu falschen oder fehlenden Disparitäten in der Karte führt.

In dieser Arbeit werden nur Algorithmen für die Erstellung dicht besetzter Disparitätenkarten berücksichtigt. Diese sind zwar aufwändiger zu berechnen, bieten jedoch durch die möglichst vollständige Abdeckung der beobachteten Szene einen erheblichen Mehrwert für die Detektion von Objekten und schränken die mögliche Algorithmenwahl nicht ein. [53] zeigt, dass dichte Disparitätenkarten einen Vorteil bei der Berechnung von Nickwinkel, Kamerahöhe und Straßenoberfläche bieten. Die Anforderungen an die Disparitätenkarte unterscheiden sich bei beiden vorgestellten Prozessketten. Der regelbasierte Ansatz zur Objektdetektion basiert direkt auf der Disparitätenkarte, womit die Qualität des Ansatzes direkt von dieser abhängt. Da die Qualität der Disparitätenkarte wiederum von der Güte der Kalibrierung, dem *Preprocessing*, aber auch von der beobachteten Szene selbst abhängt, muss die Korrespondenzanalyse möglichst robust sein. Werden zusammenhängende Bereiche der Disparitätenkarte falsch klassifiziert, z. B. aufgrund von fehlender Textur, so kann dies eine falsche Objekthypothese zur Folge haben. Andererseits führt exzessives Glätten der Karte zum möglichen Informationsverlust und folglich zu einem *False Negative* in der Objektdetektion.

In der rechten Prozesskette der Abbildung 3.3 basiert die Objekterkennung auf einem Kamerabild. Die extrahierten Objekte werden mit Informationen der Disparitätenkarte annotiert, um eine Transformation in das dreidimensionale Weltkoordinatensystem zu ermöglichen. Die Anforderungen an die Disparitätenkarte korrelieren hierbei mit den Anforderungen an die Objektdarstellung. Soll die Objektfläche bzw. -form in Weltkoordinaten modelliert werden, so erfordert dies eine entsprechende Güte der Disparitätenkarte. Die einfachste Form der Objektdarstellung ist ein Punkt, resultierend aus dem Zentrum der extrahierten ROI und einer zugeordneten repräsentativen Disparität. Hierfür muss die Disparitätenkarte innerhalb der ROI eine entsprechende Verteilung der Disparität aufweisen. Diese Objektdarstellung ist daher weniger anfällig für auftretendes Rauschen oder Lücken in der Karte als eine Modellierung der Objektfläche oder -form, weshalb sie für beide Prozessketten in dieser Arbeit verwendet wird.

### 3.4.1 Lokale und Globale Methoden

Die Berechnung der Disparitätenkarte anhand eines rektifizierten Bildpaares eines Stereokamerasystems kann nach [54] in zwei grundlegende Methoden unterteilt werden, lokale und globale Methoden. Beide Varianten unterscheiden sich bezüglich ihrer Annahmen bezogen auf die Disparitätenunterschiede benachbarter Pixel, das sogenannte Glattheitskriterium oder *smoothness assumption*.

Lokale Algorithmen verwenden Fenster endlicher Größe, mit deren Hilfe Bildausschnitte (*blocks*) entlang der Bildzeile extrahiert werden. Die Intensitäten der Ausschnitte des linken und rechten Bildes werden anhand einer Kostenfunktion verglichen, was meist *block matching* genannt wird. Die Position mit minimalen Kosten entspricht den korrespondierenden Abbildungen eines Weltpunkts, womit die Disparität berechnet werden kann. Die minimale Fenstergröße von einem Pixel erzeugt bei der Zuordnung Mehrdeutigkeiten. Durch die Verwendung größerer Fenster, der damit größeren Bildabdeckung und vermeintlich größeren Texturänderung, sollen diese reduziert werden [55] [56]. [57] zeigt diesen Effekt und das dadurch reduzierte Rauschen in der Disparitätenkarte. Die Annahme, dass innerhalb des Fensters eine konstante Disparität vorherrscht, führt in Abhängigkeit der Fenstergröße zu verwaschenen Konturen der dargestellten Objekte. Dies wird aufgrund der oft auftretenden Ausweitung der Objektgrenzen im Vordergrund auch als *foreground fattening* bezeichnet. Auf geneigten Flächen, wie z. B. einer Straßenebene aus der Perspektive einer vorwärtsgerichteten Kamera eines Fahrzeugs, wird diese Annahme ebenfalls verletzt, was zu vermehrten fehlerhaften Zuordnungen führt [57]. Erweiterungen des lokalen Ansatzes, z. B. durch adaptive Fenster, können diese Probleme lösen [56].

Treten in einer Szene schwer handhabbare Mehrdeutigkeiten auf, z. B. durch große, schwach texturierte Bildbereiche, so können selbst große lokale Fenster diese nicht auflösen. Globale Methoden können hier Abhilfe schaffen. Diese rechenintensiven Algorithmen basieren auf der Optimierung einer Energiefunktion, welche ein explizites Glattheitskriterium sowie einen datenspezifischen Term beinhaltet, der die Zuordnung der Abbildungen auf die beiden Eingangsbilder darstellt [55] [54]. Vergleicht man die Leistung von lokalen und globalen Methoden in Benchmarks, so dominieren nach [56] die lokalen Methoden in synthetischen oder Laborszenarien, während globale Methoden in realistischen Straßenszenen ein besseres Verhalten zeigen.

### 3.4.2 Algorithmenauswahl

Die Anforderungen an die Disparitätenkarte und somit an den berechnenden Algorithmus unterscheiden sich für beide, in dieser Arbeit umgesetzte Prozessketten. Bezogen auf die Unterscheidung zwischen globalen und lokalen Methoden scheint der globale Ansatz aufgrund der besseren Leistung in realistischen Szenen die richtige Wahl zu sein. Die hohe Komplexität dieses Ansatzes und die dadurch erforderliche Rechenzeit steht dem jedoch entgegen. Eine Methode, welche eine ähnliche Leistung wie globale Methoden bei geringerer Komplexität bietet, ist das *semi-global matching* [58]. Aufgrund der erwähnten Eigenschaften wurde dieser Ansatz in mehreren Veröffentlichungen und Echtzeitanwendungen in den Bereichen Robotik, ADAS und AD ausgewählt, wie z. B. [59], [60] und [61]. Das *semi-global matching* vereint lokale und globale Konzepte zu einem genauen, pixelbasierten Vergleich der Eingangsbilder, bei vergleichsweise geringer Komplexität und potenzieller Echtzeitfähigkeit. Die



**Abbildung 3.4**

Eine Szene aus dem „KITTF“-Datensatz [34] (oben). Es wird das Bild der linken Kamera und die daraus berechnete Disparitätenkarte (unten) dargestellt. Große Disparitäten (grün) entsprechen nahen Objekten, kleine Disparitäten (rot) weiter entfernten Objekten. Die minimal dargestellte Disparität entspricht ca. einer Entfernung von 12 m.

zu optimierende Energiefunktion wird nicht global, sondern entlang mehrerer eindimensionalen Pfade gerechnet [62].

Die in dieser Arbeit verwendete Implementierung der Bibliothek „OpenCV“ entspricht in weiten Teilen der Implementierung von [58]. Eine wichtige Änderung ist dabei die Verwendung von Fenstern anstatt einzelner Punkte. Die genauen Änderungen können der „OpenCV“-Dokumentation entnommen werden [63]. In Abbildung 3.4 ist beispielhaft eine Disparitätenkarte für eine reale Szene abgebildet. Dabei wurde die minimale Disparität auf einen Wert begrenzt, der einer Entfernung von 12 m entspricht. Nahe Objekte werden grün, entfernte Objekte rot dargestellt. In der rechten Bildhälfte kann eindeutig ein verdeckter Bereich durch die Person im Vordergrund erkannt werden, welcher sich als schwarzer Rand auf der linken Seite der Person darstellt. Dieser Bildbereich wird durch die Perspektive der rechten Kamera nicht erfasst. Weiterhin zeigen sich auf dem Boden regelmäßig fehlende oder falsch zugeordnete Disparitäten. Durch die verwendete Implementierung, welche auf Fenstern basiert, kann auch das *foreground fattening* wahrgenommen werden.

## 3.5 Objektdetektion

Beide in Abbildung 3.3 dargestellten Prozessketten unterscheiden sich durch die Objektdetektion bzw. Objekterkennung, welche aus verfügbaren Daten eines Zeitschritts Objekthypothesen extrahiert. Diese Hypothesen werden erst durch die Objektverfolgung zeitlich gekoppelt und somit die Existenz der Objekte bestätigt oder verworfen. Auch wenn sich die zwei verfolgten Ansätze grundlegend unterscheiden, müssen beide Methoden robust gegenüber den Eigenheiten der verwendeten NCS-Daten sein. Viele Algorithmen zur Objektdetektion auf Bilddaten oder Disparitätenkarten wurden für die Bereiche ADAS und AD entwickelt. Die Annahmen, welche für Pkw und Lkw gelten, treffen oft nicht auf die Verwendung mit Fahrrädern zu. Kapitel 3.5.1 definiert daher die Anforderungen an Algorithmen im Bereich NCS,

bevor in den Abschnitten 3.5.2 und 3.5.3 die Auswahl der Algorithmen anhand vorgestellter Literatur und den definierten Anforderungen beschrieben wird.

### 3.5.1 Kriterien zur Algorithmenwahl

Die NCS-Daten haben Eigenschaften, welche sie von den Daten aus klassischen Pkw- und Lkw-Anwendungen unterscheiden. Das Fahrverhalten eines Einspurfahrzeugs, wie eines Fahrrads, unterscheidet sich deutlich von dem eines Zweispurfahrzeugs. Der größte Unterschied ist die Neigung des Fahrzeugs bei der Kurvenfahrt. Die Wankwinkel von Pkw und Lkw unter Querschleunigung werden durch deren Fahrwerksauslegung definiert. Wankstabilisatoren helfen große Winkel zu vermeiden. Ein Einspurfahrzeug muss prinzipbedingt signifikante Wankwinkel für die Kurvenfahrt aufbringen. Abbildung 3.5 zeigt dies exemplarisch anhand einer aufgezeichneten Szene. Ein Abbiegevorgang mit angepasster Geschwindigkeit führt hier zu einem Wankwinkel von über  $15^\circ$ . Weiterhin besteht die Chance auf eine große Verdeckung des Sichtfelds bei geringem Abstand zu vorausfahrenden Objekten, siehe Abbildung 3.5 rechts. Dies ist keine NCS spezifische Eigenschaft, die Befestigung der Kamera am Lenker und der daraus resultierende minimale Abstand zum\*zur vorausfahrenden Verkehrsteilnehmer\*in verschärfen jedoch diese Problematik. Eine weitere Eigenschaft der NCS-Daten ist die Varianz der Sensorposition gegenüber der Fahrbahn. Durch die Verwendung von Probandenfahrrädern verschiedener Hersteller und Größen sowie durch eine potenziell veränderbare Orientierung des Sensors existiert keine feste Verbauposition. Manche auf Pkw und Lkw bezogene Algorithmen machen sich eine feste und bekannte Sensorposition und Orientierung zu Nutze, was im Falle einer NCS nicht möglich ist.

Ein wichtiges Kriterium für die Algorithmenwahl ist somit die Robustheit gegenüber den genannten Effekten in NCS-Daten. Dynamische Winkeländerungen gegenüber der Fahrbahn durch Wanken, aber auch statische Variabilität durch unterschiedliche Verbaupositionen der Sensorik dürfen keinen negativen Einfluss auf die Auswertung haben. Weiterhin muss der Algorithmus trotz großer, zeitlich beschränkter Verdeckungen des Sichtfelds verlässlich funktionieren.

Das wesentliche Kriterium zur Algorithmenwahl stellen die potenziell erkennbaren Objekte dar. Prinzipiell können alle nicht überfahrbaren Objekte zu einer kritischen Situation führen und sind somit relevant. Ab wann ein Objekt für ein Fahrrad überfahrbar ist, lässt sich jedoch nicht einfach definieren. Die persönliche Risikobereitschaft und das Können der Proband\*innen, aber auch der Fahrradtyp und andere Umgebungsvariablen spielen dabei eine Rolle. Die Objekthöhe kann nach Anwendungsfall definiert werden, wobei die Funktionsweise der verwendeten Algorithmen sowie der verwendete Sensor berücksichtigt werden müssen. Beliebig kleine Objekte können, selbst bei perfekter Kalibrierung und Korrespondenzanalyse, in der Tiefenkarte nicht von der Fahrbahnfläche abgehoben werden. Die Definition der Sensorparameter, die Algorithmenauswahl und die Festlegung der minimalen Objekthöhe gehen Hand in Hand, weshalb allen Punkten größte Sorgfalt beim Studiendesign geschenkt werden muss. Im Folgenden gilt die Annahme, dass alle Objekte mit einer Höhe kleiner als 20 cm überfahrbar sind. Wobei dieser Wert exemplarisch gewählt wurde und an der Grenze der Abbildbarkeit mit dem verwendeten Sensor liegt. Der regelbasierte Ansatz muss folglich in der Lage sein, nicht überfahrbare Objekte mit einer Höhe größer 20 cm zu detektieren.



**Abbildung 3.5**

Szenen aus aufgezeichneten NCS-Daten: Auftretende Wankwinkel bei Kurvenfahrt (links) und Verdeckung durch Kolonnenfahrt (rechts)

Für Algorithmen, welche auf Objektklassen trainiert wurden, wie die in Kapitel 3.5.3 vorgestellten, müssen andere Kriterien gelten. Diese Klassifikatoren können nur antrainierte Objekte erkennen, womit die Anforderung generische nicht überfahrbare Objekte zu erkennen per se nicht erfüllbar ist. Dem Anwender müssen daher die trainierten Objektklassen bewusst sein, diese müssen die aus der Forschungsfrage abgeleiteten Anforderungen erfüllen. Dies gilt ebenso für die Erkennungsraten, welche auf realen Verkehrsszenen basierenden Benchmarks verglichen werden können.

Die Objektdetektion gibt generell für jedes detektierte Objekt eine ROI aus. Mit der vorhandenen Disparitätenkarte wird daraus ein Punktmodell des Objekts berechnet, welches die Koordinaten des ROI Zentrums auf der Bildebene sowie einen entsprechenden Disparitätswert umfasst. Je nach Algorithmus wird die ROI in einer anderen Form ausgegeben, diese kann ein Polygonzug, aber auch ein Satz Pixel sein. Das Objekt als Punkt darzustellen ist für die prototypische Implementierung ausreichend. Erweiterungen um die Objektmaße bzw. die Oberflächenform sind möglich, werden hier aber nicht umgesetzt.

## 3.5.2 Regelbasierter Ansatz

Regelbasierte Expertensysteme vereinen Informationen über Fakten sowie Regeln zur Herleitung neuer Fakten in einer Wissensbasis. Basierend auf den Ausgangsdaten können die Regeln, welche sich als Wenn-Dann-Aussagen darstellen lassen, sukzessive angewendet werden [64]. Bezogen auf die Objektdetektion führt dies zu einer eindeutigen und nachvollziehbaren Verkettung von Abfragen, bis eine ROI aus den Daten extrahiert ist. Das Ziel ist dabei ohne Training eines Modells, sondern mit auf Annahmen bzw. auf Wissen basierter Datenverarbeitung, Objekte zu extrahieren. Die Objektdetektion soll so generisch wie möglich gestaltet sein und durch die Wissensbasis möglichst keine Vorabauswahl der detektierbaren Objekte treffen. Ebenso darf die Variabilität der Versuchsdaten, z. B. die veränderliche Sensorposition, keinen negativen Einfluss auf die Performanz darstellen.

### 3.5.2.1 Regelbasierte Methoden zur Objektdetektion

Nach [65] und [66] lässt sich die Erkennung von Fahrzeugen aus (Stereo-) Kameradaten in die Kategorien merkmalsbasiert, bewegungsbasiert und stereovisionsbasiert einteilen. Diese Kategorisierung kann auf die generische Objektdetektion übertragen

werden. Modellbasierte Verfahren werden aufgrund der vorhergenannten Einschränkungen bezüglich generischer Objekte an dieser Stelle nicht betrachtet. Hierfür sei auf Kapitel 3.5.3 verwiesen.

**Merkmalsbasierte Methoden:**

Einzelbilder einer Kamera werden hierbei nach visuellen Hinweisen durchsucht. Diese können beispielsweise Kanten, Symmetrie oder Objektschatten sein. Mehrere verschiedene Hinweise können verkettet werden. Der Schatten eines Objekts auf Asphalt kann durch dessen Intensität erkannt werden, da die dem Schatten zugehörigen Pixel wesentlich dunkler als die benachbarten Pixel außerhalb des Schattenbereichs sind [67]. Problematisch ist dabei jedoch, dass nach [68] nicht alle potenziellen Objekte einen deutlich sichtbaren Schattenbereich aufweisen, was die Detektionsrate beeinträchtigt. [69] verwendet neben dem Schatten die Textur, horizontale und vertikale Kanten sowie die Symmetrie der vertikalen Kanten zur Detektion von vorausfahrenden Fahrzeugen. Bei Motor- und Fahrrädern konnten diese Methoden aufgrund der weniger prominenten Kanten und deren Symmetrie keine überzeugenden Ergebnisse liefern [65]. Ebenso sind die in [52] vorgestellten Ansätze zur Detektion von Scheinwerfern oder Reifen gezielt auf Charakteristiken von Pkw und Lkw ausgelegt und lassen sich nicht für alle potenziell relevanten Objekte generalisieren. Sie können jedoch bei modellbasierten Verfahren zur Anwendung kommen.

**Bewegungsbasierte Methoden:**

Die bewegungsbasierten Methoden verwenden den optischen Fluss (*optical flow*), welcher sich aus zeitlich aufeinanderfolgenden Bildern einer Kamera berechnen lässt. Dabei werden korrespondierende Bildpunkte gesucht und deren Verschiebung auf der Bildebene ermittelt. Ähnlich der Disparitätenkarte lässt sich der optische Fluss für das gesamte Bild oder für bestimmte Bildpunkte ermitteln. Er lässt sich basierend auf Bildposition, Magnitude und Winkel in *cluster* gruppieren, woraus Objekthypothesen generiert werden können. Die Abhängigkeit des optischen Flusses von der Relativgeschwindigkeit zwischen Sensor und abgebildeten Weltpunkt führt zu niedrigen Erkennungsraten bei geringer Relativgeschwindigkeit. [70] zeigt anhand eines auf Schnellstraßen aufgezeichneten Datensatzes, eine Detektionsrate von überholenden Fahrzeugen von weniger als 70 % bei Relativgeschwindigkeiten von 5 bis 10 km/h. Steigen diese auf 10 bis 20 km/h erhöht sich die Detektionsrate auf über 90 %. [68] nennt weiterhin Probleme bei der Detektion von Bewegung in der Nähe des Fluchtpunkts sowie die fehlende Möglichkeit zur Detektion von spezifischen statischen Objekten.

**Stereovisionsbasierte Methoden:**

Werden die Tiefeninformationen eines Stereobildpaares für die Objektdetektion verwendet, spricht man von stereovisionsbasierten Methoden. Den Erläuterungen aus Kapitel 3.4 entsprechend, kann eine Disparitätenkarte dünn oder dicht besetzt sein. Die Extraktion von Objekten kann direkt auf der Disparitätenkarte oder den daraus abgeleiteten dreidimensionalen Punktwolken geschehen [71]. Letztere weisen eine mit der Entfernung abnehmende Dichte durch den nichtlinearen Zusammenhang zwischen Disparität und Tiefe auf. Eine Objektdetektion basierend auf einer Disparitätenkarte wird beispielsweise durch [72] gezeigt, welcher die sogenannte *v-disparity* berechnet und zur Schätzung der Bodenfläche sowie zur Objektdetektion nutzt. Eine

ebene Bodenfläche, welche sich parallel zu den Bildzeilen des Sensors erstreckt, stellt sich in der Disparitätenkarte durch konstante Werte entlang der Bildzeile dar. Entlang der Bildspalten nimmt die Disparität bis zum Horizont kontinuierlich ab. Durch eine Projektion der dreidimensionalen Szene auf eine Ebene können die Disparitäten der Bodenfläche extrahiert und somit die entsprechenden Pixel aus der Disparitätenkarte entfernt werden. Die resultierenden Pixel entsprechen Objektpunkten oberhalb des Bodens und eignen sich nach einem *clustering* für die Verwendung als Objekthypothesen. [73] zeigt die Extraktion von Objekthypothesen aus einer dreidimensionalen Punktwolke. Dabei werden für die Abbildungen einer dünn besetzten Disparitätenkarte die Punkte im Weltkoordinatensystem rekonstruiert. In Kombination mit dem optischen Fluss dieser Abbildungen kann für jeden Punkt die Position und Geschwindigkeit im Weltkoordinatensystem mit Hilfe eines Kalman-Filters berechnet werden. Die Verwendung des optischen Flusses in Verbindung mit Stereovision findet sich in weiteren Ansätzen wieder, wie z. B. [74].

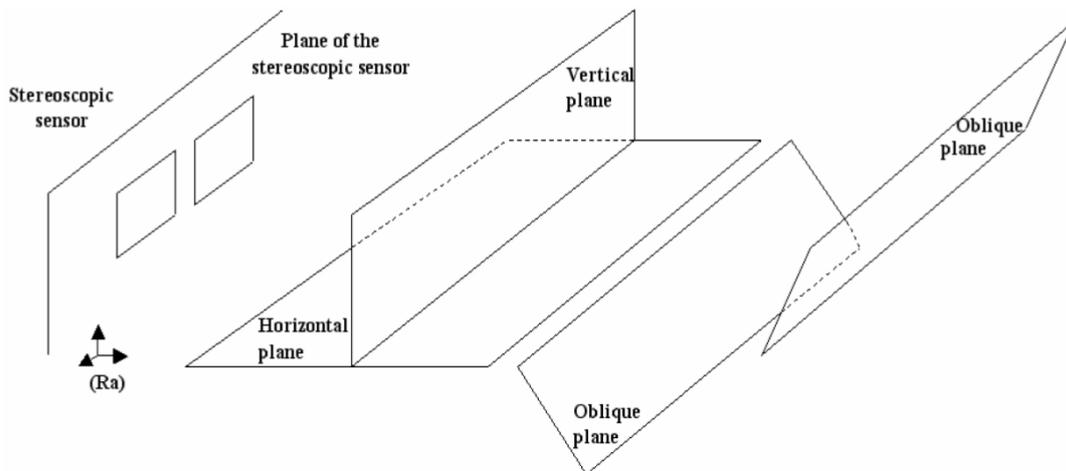
#### 3.5.2.2 Auswahl der Algorithmenklasse

Die zu verwendende Methode für die regelbasierte Objektdetektion kann anhand der Anforderungen aus Kapitel 3.5.1 ausgewählt werden. Aufgrund der Vielzahl von Annahmen eignen sich die merkmalsbasierten Methoden nur bedingt für eine generische Objektdetektion. Symmetriekriterien oder gar Detektoren für Reifen oder Scheinwerfer adressieren spezifische Objekteigenschaften, welche die Generalität als Voraussetzung missachten. Ansätze, wie z. B. die Detektion basierend auf dem Schatten des Objekts, stellen einerseits keine direkten Bedingungen an das Objekt bzw. dessen Ansicht wie die symmetriebasierten Hinweise, kranken andererseits an einer nicht allgemeingültigen Grundvoraussetzung. Die bewegungsbasierten Methoden scheiden als alleiniges Kriterium aufgrund der Limitationen bezüglich des nutzbaren Relativgeschwindigkeitsbereichs aus. Eine ergänzende Verwendung des optischen Flusses in Kombination mit anderen Ansätzen zeigt in der Literatur vielversprechende Ergebnisse. Merkmals- sowie bewegungsbasierte Methoden können trotz der Eigenbewegung des Sensorsystems auf einem Fahrradlenker prinzipiell angewendet werden. Spezifische Implementierungen, z. B. basierend auf der Symmetrie von Kanten, können Anpassungen bezüglich der Variation im Wankwinkel benötigen.

Die vorgestellten stereobasierten Methoden eignen sich für die generische Objektdetektion, da die Abbildung von Weltpunkten auf die Disparitätenkarte nicht von objektklassenspezifischen Eigenschaften abhängig ist. Das Kriterium ist hierbei einzig die valide Rekonstruktion des Weltpunkts, welche durch die Eigenbewegung des Stereosystems nicht negativ beeinflusst wird. Auf die Extraktion von Objekten aus der Disparitätenkarte oder einer abgeleiteten dreidimensionalen Punktwolke muss dies jedoch nicht zutreffen. In Abschnitt 3.5.2.3 werden einige stereobasierte Ansätze vorgestellt, um ein tieferes Verständnis für die Funktionsweise dieser Algorithmenklasse zu ermöglichen.

#### 3.5.2.3 Stereovisionsbasierte Objektdetektion

Die im folgenden Abschnitt beschriebenen Algorithmen zur Objektdetektion können nach der verwendeten Datenbasis in rein stereovisionsbasierte Ansätze und kombinierte Ansätze unterteilt werden. Letztere verwenden zusätzlich den optischen Fluss

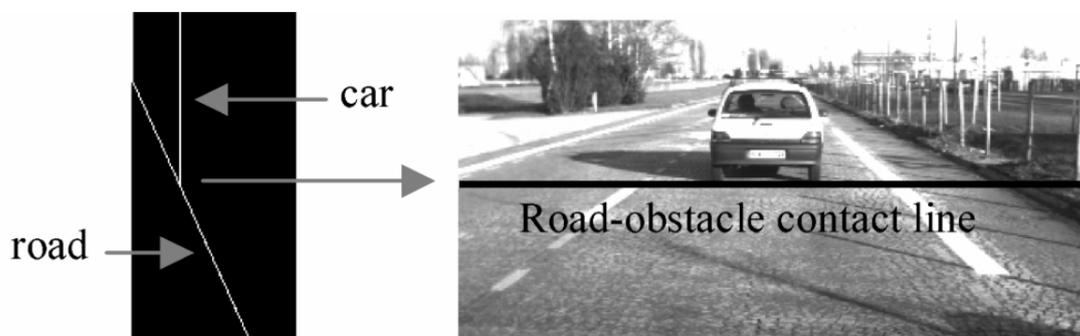


**Abbildung 3.6**

Horizontale, vertikale und schiefe Ebenen in einer dreidimensionalen Szene. Entnommen aus [72].

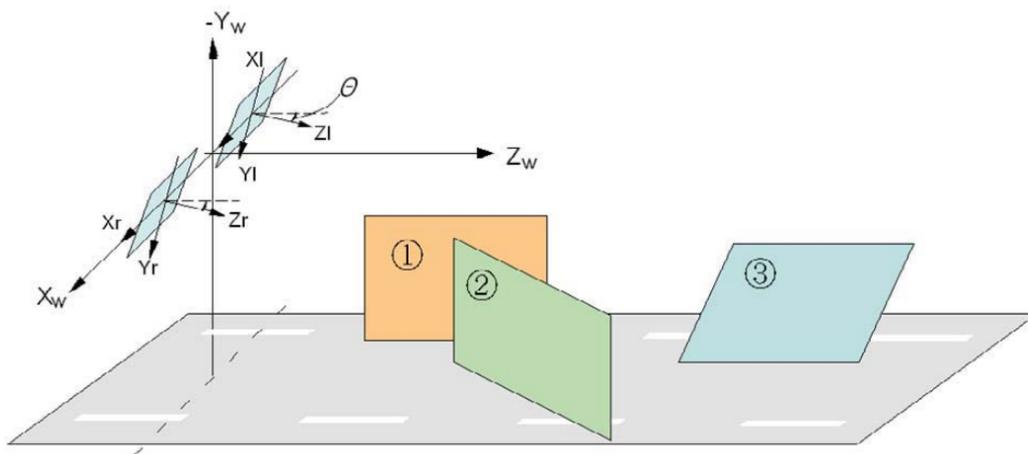
als Informationsquelle. Die stereovisionsbasierten Algorithmen lassen sich wiederum in verschiedene Ansätze unterteilen, wie z. B. die Projektionen der *v-disparity* und *u-disparity*, Bearbeitung der Disparitätenkarte mit klassischen Bildsegmentierungsalgorithmen oder die Extraktion und *clustering* von Stixeln. Ein Großteil der Ansätze greift dabei auf ein Ebenenmodell zurück, welches Objekte und Grund als Ebenen im Raum beschreibt.

In [72] wird gezeigt, dass dreidimensionale Szenen annähernd durch Projektionen in den zweidimensionalen Raum beschrieben werden können. Abbildung 3.6 zeigt horizontale, vertikale und schiefe Ebenen in einem Koordinatensystem  $R_a$ . Projiziert man diese Ebenen auf die YZ-Ebene des Kamerakoordinatensystems, so werden sie durch Geraden abgebildet. Unter der Annahme, dass ein Element der Szene durch eine dieser Ebenen in der Disparitätenkarte dargestellt werden kann, reduziert sich die Detektionsaufgabe für dieses Szenenelement auf die Extraktion von Linien aus der Projektion, z. B. durch RANSAC Algorithmen. In der Praxis wird die Projekti-



**Abbildung 3.7**

Darstellung der *v-disparity* (links) einer realen Szene (rechts). Die Straßenebene entspricht durch die veränderliche Tiefe einer schiefen Geraden. Der Pkw wird als vertikale Linie mit fast konstanter Tiefe abgebildet. Entnommen aus [72].



**Abbildung 3.8**

Schiefe Ebenen welche sich nicht in der *v-disparity* darstellen lassen. 1: Vertikale Ebenen parallel zur YZ-Ebene des Kamerakoordinatensystems, wie z. B. Seitenflächen von Fahrzeugen; 2: Schiefe vertikale Ebenen, ähnlich dem ersten Fall, mit geringen Winkel zur YZ-Ebene, wie z. B. einscherende Fahrzeuge; 3: Schiefe Ebenen, welche aufgrund einer Rotation des Kamerakoordinatensystems um die Z-Achse, nicht vertikal sind. Ein wankendes Fahrzeug kann diesen Effekt hervorrufen. Entnommen aus [77].

on durch die Akkumulation der Disparitäten entlang der Bildzeilen ausgeführt. Das resultierende Bild entspricht in der Höhe der ursprünglichen Disparitätenkarte, in der Breite dem erlaubten Disparitätenbereich. Abbildung 3.7 zeigt beispielhaft die *v-disparity* für eine reale Szene. Schiefe Ebenen, z. B. eine lateral abfallende Straßenoberfläche, führen zu einer Aufweitung des Entsprechenden Disparitätenbereichs und folglich zu einer Geraden in der *v-disparity* mit mehreren Pixeln Breite [75]. Die Breite ist dabei vom Winkel der abzubildenden Ebene zur Projektionsebene abhängig. Dieser Umstand erlaubt eine Schätzung des dynamischen Roll- und Nickwinkels des Stereosystems bezogen auf die Straßenoberfläche [76].

Nicht alle Objekte können in der *v-disparity* durch die in Abbildung 3.6 vorgestellten Ebenen dargestellt werden. [77] zeigt drei Fälle auf, welche nicht detektiert werden können, siehe Abbildung 3.8. Zur Detektion von Objekten mit entsprechenden Ebenenrepräsentationen kann eine weitere Projektion eingeführt werden. Die *u-disparity* akkumuliert die Disparitäten entlang der Bildspalten, wodurch alle annähernd vertikalen Ebenen als Geraden abgebildet werden. Annähernd horizontale Ebenen, wie die Straßenoberfläche, werden nicht abgebildet.

Die Darstellung von Szenenelementen durch Ebenen und deren Projektion in den zweidimensionalen Raum ist eine Grundannahme vieler Veröffentlichungen der stereobasierten Objektdetektion. [78] klassifiziert Pixel in Stereobildpaaren nach ihrer Umgebungsstruktur. Aus potenziell korrespondierenden vertikalen Strukturen in derselben Bildzeile werden mögliche Disparitäten berechnet. Das Korrespondenzproblem kann durch die Datenreduktion der Klassifikation oft nicht eindeutig gelöst werden. Aufgetragen in einem Histogramm für den gesamten Bildbereich, ergeben sich dennoch markante Spitzen für alle Objekte, welche sich durch vertikale Ebenen annähern lassen. Falsche Korrespondenzen zeigen sich durch ein Rauschen im Histogramm. [72] verwendet die *v-disparity* zur Detektion der Straßenoberfläche sowie zur Erkennung von Objekten. Dabei gelten die oben beschriebenen Limitationen

des Ansatzes. Weiterhin schätzt [76] mit Hilfe der *v-disparity* den Nick-, Wank- und Gierwinkel sowie ein lineares Straßenmodell. [79] extrahiert dreidimensionale Formen aus der Disparitätenkarte und bestimmt aus der *v-disparity* die Straßenebene. Anhand dieser werden durch Kriterien bezüglich der Disparität und des Neigungswinkels gegenüber der Straße Formen als Objekte klassifiziert. [80] verwendet adaptive Schwellenwerte und eine Normalisierung der Spalten der *v-disparity*, um die Extraktion der Straßenoberfläche robuster zu gestalten. Diese wird in der Folge zur Detektion des Freiraums verwendet. Eine Erweiterung der *v-disparity* zeigt [75]. Die Perspektive der Disparitätenkarte wird transformiert, wodurch eine virtuelle Disparitätenkarte auf Höhe der Straßenoberfläche unter Berücksichtigung des Nick- und Wankwinkels entsteht. Aus den bereits beschriebenen Projektionen der Disparitätenkarte kann in diesem Fall eine Änderung des Nick- und Wankwinkels sowie der Kamerahöhe über der Grundfläche direkt extrahiert werden. Die Objektdetektion kann ebenso mit den bereits vorgestellten Annahmen durchgeführt werden.

[81] verwendet eine Kombination aus *v-disparity* und *u-disparity*, um *Platooning* für einen Roboter umzusetzen. Auch [82] kombiniert diese Ansätze mit einem optimierten Vorgehen zur Linienextraktion aus den Projektionen, für die Erkennung der Straßenoberfläche und von Objekten. [83] extrahiert den Verkehrsraum als ROI für potenzielle Objekte, wodurch Regionen abseits der Straße von der Suche nach Objekten ausgeschlossen werden. Mit Hilfe der *u-disparity* und einem *connected component extraction algorithm*, welcher die Suche in der Projektion von Linien auf Polylinien erweitert, werden Objekte extrahiert. Durch die Erweiterung können diese durch mehrere projizierte Ebenen repräsentiert werden, z. B. durch die Front- und Seitenebene eines entgegenkommenden Fahrzeugs. Während aus der *u-disparity* direkt die Disparität und die Breite des Objekts entnommen werden kann, wird die Höhe aus der *v-disparity* extrahiert. Bei zwei Objekten gleicher Entfernung ist dies jedoch nicht eindeutig, weswegen die Objekthöhen aus der Disparitätenkarte mit Hilfe eines Schwellenwerts für die Disparitätenabweichung innerhalb eines Objekts bestimmt werden. Dieses Vorgehen kann wiederum zu Mehrfachdetektionen eines Objekts führen, welche über Höhenkriterien und Zusammenführung ähnlicher Objekte aufgelöst werden sollen.

[84] greift ebenfalls das Ebenenmodell auf und repräsentiert den urbanen Verkehrsraum durch ebene Rechtecke konstanter Breite, sogenannte Stixel. Ein Stixel wird durch dessen Tiefe und Höhe definiert, wobei die Basis des Rechtecks sich immer auf Höhe der Fahrbahn befindet. Der Verkehrsraum im Sichtfeld der Kamera wird in Freiraum und Objekt unterteilt. [85] erweitert diesen Ansatz ohne die Bedingung, dass ein Objekt Kontakt zur Fahrbahn haben muss. [86] gruppiert die Stixel mit Hilfe eines abgewandelten DBSCAN Algorithmus [87] zu Objekten.

[88] analysiert die *connexity* der Disparitätenkarte um fronto-parallele Ebenen zu extrahieren. Hierfür wird die Straßenebene aus der Disparitätenkarte entfernt, wobei angenommen wird, dass die Position des Stereosystems in Relation zur Straßenebene bekannt ist. Dynamische Änderungen des Nick- und Wankwinkels werden nicht berücksichtigt. Zusammenhängende Komponenten der Disparitätenkarte werden unter Berücksichtigung einer Tiefenbedingung mit Hilfe eines *flood-fill* Algorithmus extrahiert. Dabei gilt die Annahme des Ebenenmodells, welches für die *v-disparity* eingeführt wurde. Da diese nur im Idealfall gilt, muss eine Toleranz eingeführt werden, welche die Tiefenauflösung des Systems berücksichtigt. Die Extrakte, welche den geometrischen Anforderungen bezüglich Position im Bild, Breite und

Höhe genügen, werden als Objekthypothesen verwendet. Anhand der geometrischen Bedingungen kann eine Objektklasse spezifiziert werden, z. B. Personen oder Pkw. Lockert man die Bedingungen wird der Algorithmus für generische Objekte anwendbar. [89] verwendet einen zweistufigen Prozess zur Extraktion von Objekthypothesen für Fahrzeuge. Die erste Stufe extrahiert Objekthypothesen anhand eines Quadermodells. Hierfür wird die Disparitätenkarte durch *region growing* segmentiert, wobei die Disparitäten von benachbarten Zellen verglichen werden. Diese werden zu einem Segment vereint falls der Disparitätenunterschied unterhalb einer zu definierenden Schwelle liegt. Die Segmente werden bezüglich ihrer Höhe über der Straßenebene gefiltert. In der Folge werden die Segmente durch Ebenen approximiert, mit Hilfe einer Clusteranalyse gruppiert und daraus Quadermodelle erstellt. Die zweite Stufe der Objektextraktion besteht aus dem *u/v-disparity* Algorithmus, welcher bereits beschrieben wurde.

Die Arbeit von [90] zur Segmentierung von Szenen basiert auf dreidimensionalen Punktwolken und trifft keine Annahmen bzgl. der Projektion von Ebenen. Die Punkte der dreidimensionalen Szene werden jeweils zusammen mit ihren Farbinformationen zu einem normierten 6D Vektor kombiniert. Durch eine Clusteranalyse werden die Punkte der Szene entsprechend der Geometrie und Farbe gruppiert. Die resultierenden Segmente können wiederum als Objekthypothesen dienen. Der Einfluss von Szenengeometrie und Farbe der Punkte auf die Clusteranalyse kann durch einen wählbaren Faktor gewichtet werden.

[91] verwendet einen kombinierten Ansatz zur Positions- und Orientierungsschätzung von Fahrzeugen. Es wird mit horizontalem optischem Fluss und der Disparitätenkarte ein *sparse scene flow field* berechnet, welches in *cluster* gruppiert wird. Die Schätzaufgabe wird mit Hilfe des *iterative closest point* (ICP) Algorithmus absolviert. Dabei wird zuerst mittels Hauptkomponentenanalyse ein Startwert der Orientierung berechnet, um dann mittels ICP ein Cuboid an den *cluster* anzupassen und das Ergebnis zu optimieren. [92] nutzt einen kombinierten Ansatz zur Bestimmung der Position und Geschwindigkeit von Weltpunkten mit Hilfe von Kalman-Filter. Weiterhin wird die Eigenbewegung der Kamera, d. h. Translation und Rotation, basierend auf den Daten des Stereosystems geschätzt ohne Inertialsensorik zu verwenden.

Die Vielfalt der möglichen Algorithmen zur stereobasierten Objektdetektion reduziert sich durch die spezifischen Eigenschaften der NCS-Daten, die in Kapitel 3.5.1 beschrieben worden sind. Generell erfüllt das Ebenenmodell die Anforderung an eine generische Objektdetektion, wenn nicht überfahrbare Objekte verlässlich erkannt werden. Der *v-disparity* Ansatz, welcher auf der Erkennung der Straßenebene basiert und somit die Wankwinkel des Fahrrads berücksichtigen muss, benötigt zwangsläufig eine spezifische Anzahl an Weltpunkten auf der Straße zur robusten Ebenendetektion. Aufgrund von Kolonnenfahrten mit hoher Verdeckung des Sichtfelds kann diese Anforderung nicht immer erfüllt werden. Dies gilt ebenso für die Stixelrepräsentation der Szene. Die Objektdetektion anhand der *u-disparity* kann ohne die Extraktion der Straßenebene durchgeführt werden, wodurch die Verdeckung kein Problem darstellt. Da Objekte durch Ebenen senkrecht zur Straßenebene dargestellt werden, und somit annähernd parallel zur Bildebene der Kameras ausgerichtet sind, ist das Wanken des Fahrrads für die reine Objektdetektion unproblematisch. Die im Kapitel 3.5.2.4 dargelegten Untersuchungen bestätigen dies.

Die vorgestellten Algorithmen zur Objektdetektion anhand von Segmentierung der Disparitätenkarte werden ausgeschlossen. [88] verwendet die *v-disparity* zur Extraktion der Straßenebene, um diese als Vorverarbeitungsschritt aus der Disparitätenkarte zu entfernen. Wie bereits beschrieben können die Bedingungen für diesen Vorgang nicht durchgehend erfüllt werden. Der erste Schritt des zweistufigen Ansatzes aus [89] beinhaltet ein Quadermodell, welche die Anforderung an generische Objekte nicht erfüllt. Die zweite Stufe basiert wiederum auf der *u/v-disparity*, die in Teilen auf NCS-Daten anwendbar ist. Kombinierte Ansätze bieten vielversprechende Ergebnisse für die Detektion dynamischer Objekte, werden in dieser Arbeit aufgrund der erhöhten Komplexität durch die Berechnung des optischen Flusses und aufgrund der Einschränkung auf dynamische Objekte nicht verwendet.

Anhand dieser Bewertung der vorgestellten Algorithmen wird für den regelbasierten Ansatz der Objektdetektion in dieser Arbeit eine Variante der *u-disparity* implementiert. Die Robustheit gegenüber dem Wankverhalten des Fahrrads, die Detektion generischer Objekte sowie die intuitive Umsetzung sprechen für diese Auswahl. In Kapitel 3.5.2.4 wird die Implementierung im Detail vorgestellt.

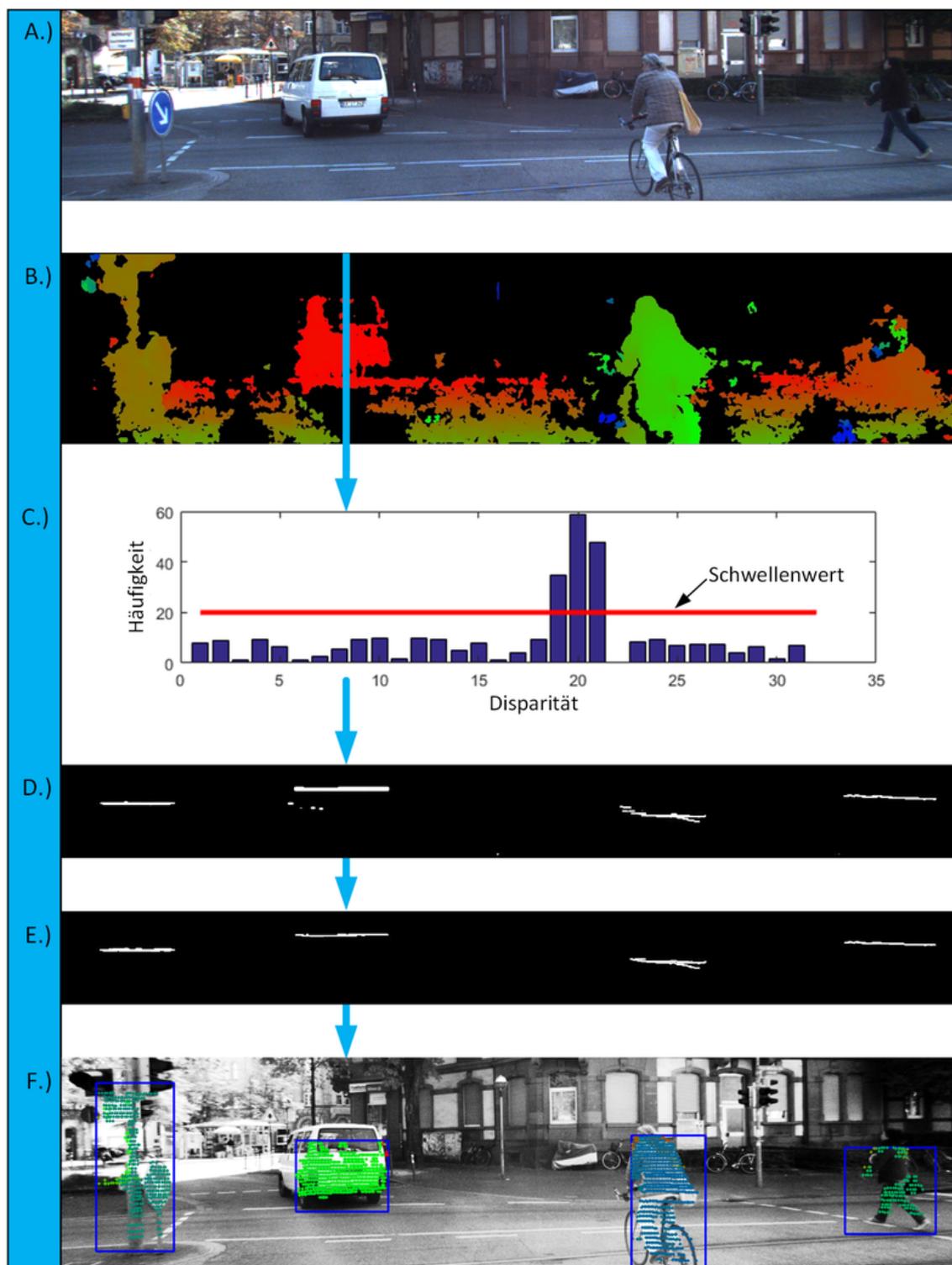
#### 3.5.2.4 Objektdetektion basierend auf U-Disparity

Das Ziel der Objektdetektion ist es, alle relevanten Objekte im Sichtfeld des Stereosystems als Punktmodell zu repräsentieren. Dieses Objektmodell wird durch einen Punkt auf der Bildebene des linken Kamerabildes sowie eine Disparität beschrieben. Die einzelnen Teilschritte der Objektdetektion sind in Abbildung 3.9 dargestellt.

Aus einem Stereobildpaar wird anhand des in Kapitel 3.4.2 ausgewählten Algorithmus eine Disparitätenkarte erstellt. Diese kann, bei vorhandenem Vorwissen bzw. anhand von Annahmen, in der Bildebene sowie im Disparitätenbereich begrenzt werden. Die Bildebene kann beispielsweise auf die beiden unteren Drittel des Bildes reduziert werden, falls darüber keine relevanten Objekte vermutet werden. Weiterhin kann der Wertebereich für gültige Disparitäten entsprechend der Annahmen im Studiendesign festgelegt werden. Beides reduziert den Berechnungsaufwand zur Erstellung der Disparitätenkarte.

Entlang der Bildspalten können nun die Disparitäten in einem Histogramm akkumuliert werden, um die Darstellung der *u-disparity* zu erhalten. Die Zeilenzahl entspricht dem Wertebereich der Disparität, die Spaltenzahl derjenigen des ursprünglichen Bildes. Obwohl die Disparitätenkarte durchaus Werte in Subpixelgenauigkeit enthalten kann, werden ganzzahlige Klassenbreiten des Histogramms bevorzugt, um einzelne Objekte über so wenige Klassen wie möglich zu verteilen. Abbildung 3.9, Abschnitt C zeigt ein hypothetisches Diagramm, welches der mit einem blauen Pfeil markierten Bildspalte entsprechen kann. Dabei zeigt sich das Objekt durch die absolute Häufigkeit der Disparitätenwerte 19 px bis 21 px. Alle weiteren Klassen entsprechen dem Rauschen der Disparitätenkarte entlang der Bildspalte. Die in Abschnitt D dargestellte *u-disparity* weist vier weiße Elemente auf, welche den markanten Objekten des ursprünglichen Bildes und der in der Tiefe auf 15 m begrenzte Disparitätenkarte entsprechen. Dabei entspricht die Breite der Elemente im Groben der Breite des jeweiligen Objekts. Diese sind von links nach rechts:

- Blaues Verkehrsschild mit Ampelpfahl
- Weißer Kleinbus



**Abbildung 3.9**

Darstellung der Prozesskette zur Objektdetektion mit Hilfe der  $u$ -disparity. A.) Bildinput aus dem Stereokamerasystem (hier ist nur das linke Bild dargestellt), entnommen aus [34]. B.) Erstellte Disparitätenkarte. C.) Beispiel eines Histogramms der Disparitäten entlang einer Bildspalte mit dargestelltem Schwellenwert zur Unterdrückung von Rauschen. D.) Die akkumulierten Histogramme entlang der Bildspalten ergeben die  $u$ -disparity. E.) Bereinigte Konturen der  $u$ -disparity. F.) Extrahierte ROIs der detektierten Objekte.

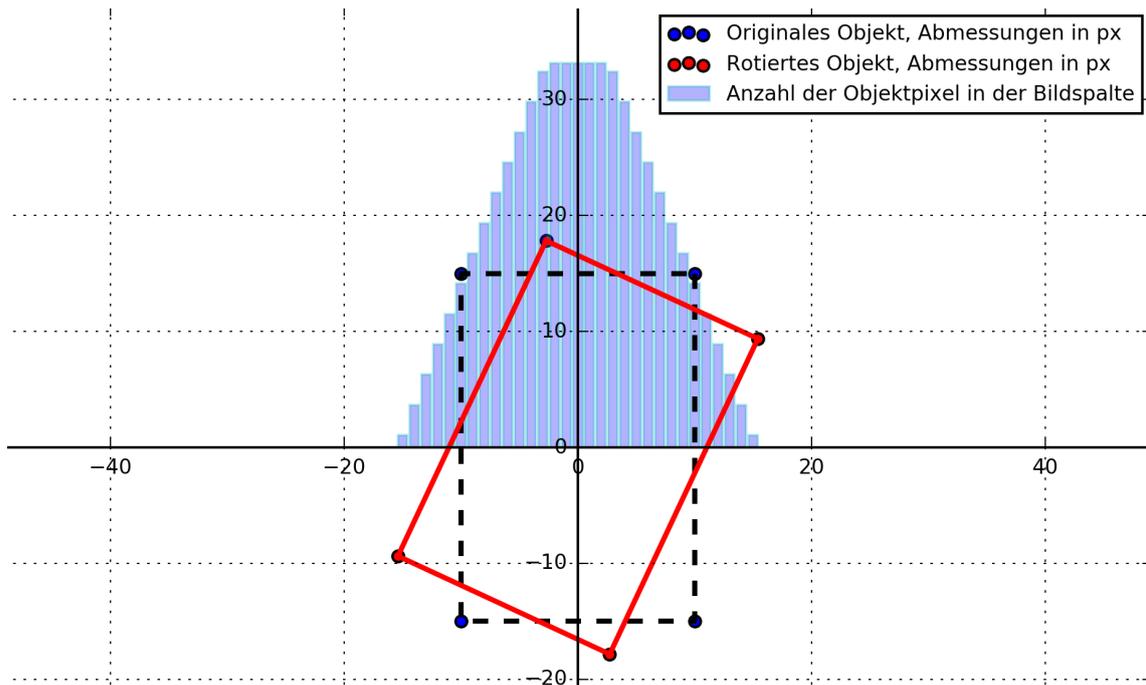
- Fahrradfahrer
- Fußgängerin

Um die *u-disparity* von Rauschen zu bereinigen und die gesamte Repräsentation zu vereinfachen wird mit Hilfe von Schwellenwerten ein Binärbild erzeugt. Die Wahl des Schwellenwerts gestaltet sich komplex. Es wird ein statischer Wert für die gesamte *u-disparity* definiert, ab welcher absoluten Häufigkeit eine Histogrammklasse als Objekt betrachtet wird. Dabei wird nicht berücksichtigt, dass entfernte Objekte auf der Bildebene eine geringere Anzahl an Bildzeilen belegen als nähere Objekte. Durch diesen nichtlinearen Effekt bildet ein statischer Schwellenwert keine Klassifizierung nach der Objekthöhe. Die Höhe  $\Delta Y$  eines Objekts belegt in Abhängigkeit der Disparität  $\delta$  und der Basislinie  $B$  die Höhe  $\Delta u$  in Pixel auf der Bildebene, siehe Formel 3.1. Somit kann der maximale Schwellenwert für ein Objekt berechnet werden, welches sich nur über eine Histogrammklasse erstreckt und keinerlei fehlerhafte Korrespondenzen oder andere Fehler der Disparitätenkarte innerhalb der ROI des Objekts aufweist.

$$\Delta u = \frac{\Delta Y \cdot \delta}{B} \quad (3.1)$$

Weiterhin weichen Objekte meist von der Annahme des Ebenenmodells ab, wodurch bei nahen Objekten eine größere Variation der Disparität auftritt als in der Ferne. Ein nahes Objekt verteilt sich, je nach Abweichung von der vertikalen Ebene, aufgrund des nichtlinearen Zusammenhangs von Entfernung und Disparität über mehr Histogrammklassen als ein weiter entferntes Objekt. Betrachtet man diesen Effekt isoliert, wäre die Häufigkeit eines nahen Objekts pro Klasse geringer als bei einem entfernten Objekt. Zusätzlich weist die Disparitätenkarte nicht für jeden Bildpunkt einen validen Disparitätswert auf. Falsche Korrespondenzen oder Verdeckungen beeinträchtigen die Disparitätenkarte und folglich die *u-disparity*. Um diese Effekte bei der Erstellung des Binärbilds zu berücksichtigen, müssen dynamische Schwellenwerte in Abhängigkeit der Disparität bzw. der Entfernung definiert werden. Weiterhin können offensichtlich fehlerhafte oder nichtexistierende Disparitätenwerte einer Bildspalte als Variable für die Schwellenwertberechnung herangezogen werden. An dieser Stelle wird auf die Formulierung eines solchen Schwellenwerts verzichtet.

Um die Objekte aus dem Binärbild zu extrahieren, werden die Konturen der zusammenhängenden Objektpixel aus der *u-disparity* extrahiert. Die Extrema einer Kontur in der Bildhöhe geben den Disparitätenbereich des Objekts an, in der Bildbreite spiegeln sie die Objektbreite innerhalb der Disparitätenkarte wider. Eine Vorfilterung der Objekthypothesen, z. B. anhand der Konturbreite oder Fläche, kann die Anzahl an Fehldetektionen reduzieren, wie in Abbildung 3.9, Abschnitt E dargestellt. Die tatsächlichen Koordinaten der ROI des jeweiligen Objekts auf der Bildebene lassen sich durch die ermittelten Extrema bestimmen. Die linke und rechte Grenze der ROI deckt sich mit den Koordinaten der Konturextrema in x-Richtung. Die obere und untere Begrenzung wird durch einen Abgleich des Disparitätenbereichs des Objekts mit der Disparitätenkarte gewonnen. Unter der Annahme, dass nicht mehrere Objekte desselben Disparitätenbereichs übereinander auf der Bildfläche abgebildet werden, können die Grenzen durch Schwellenwerte bestimmt werden. Es wird der Bereich der Disparitätenkarte extrahiert, welcher durch die linke und rechte ROI Grenze definiert ist. Die Bildzeilen des resultierenden Ausschnitts werden auf die relative Häufigkeit von validen Disparitäten untersucht. Als valide gelten



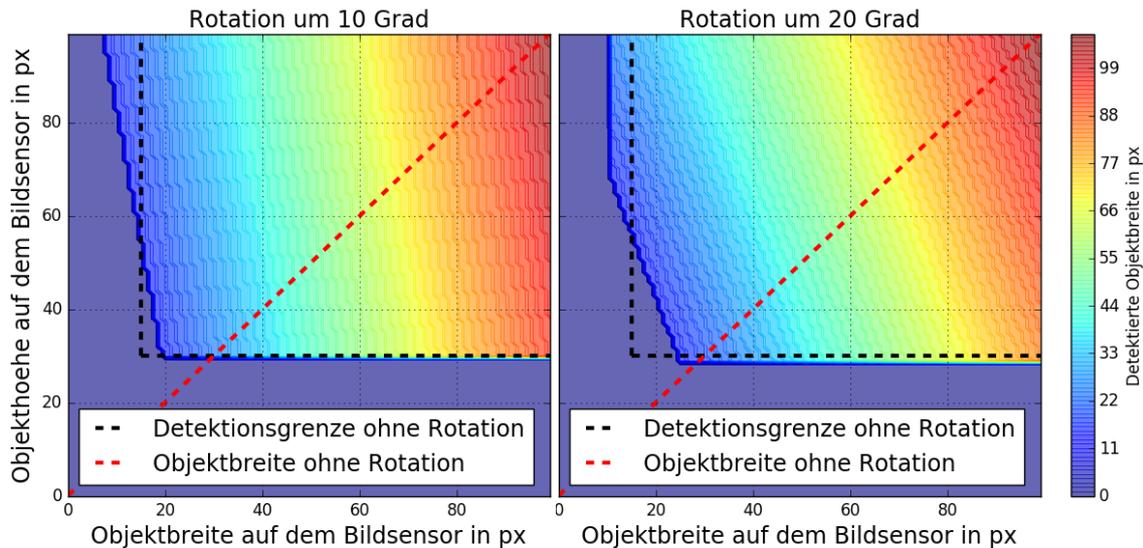
**Abbildung 3.10**

Ein Objekt der Dimension  $20 \times 30$  px (schwarzes Rechteck) wird um  $25^\circ$  rotiert (rotes Rechteck) um das Wanken des Versuchsträgers zu simulieren. Unter Annahme einer perfekten Disparitätenkarte und einer rein vertikalen Objektebene parallel zur Bildebene, weisen die Histogramme entlang der Bildspalten der *u-disparity* eine absolute Häufigkeit von 30 Elementen in der entsprechenden Klasse der Objektdisparität auf. Die Rotation ändert dies entsprechend der dargestellten Balken (blau). Die absolute Häufigkeit entspricht der Anzahl an Objektpixel pro Bildspalte.

alle Werte im Disparitätenbereich des Objekts. Die oberste und unterste Zeile, welche den Schwellenwert für valide Disparitäten überschreiten, werden als obere und untere Grenze der ROI definiert. Abbildung 3.9, Abschnitt F zeigt die ROIs der vier dargestellten Objekte sowie als Referenz LiDAR-Punkte innerhalb des validen Disparitätenbereichs der Objekte. Die Farbe der Punkte korrespondiert mit der Entfernung des Objekts. Als finales Objektmodell wird das Zentrum der ROI und der Medianwert der validen Disparitäten innerhalb der ROI definiert.

Da sich die NCS-Daten durch Wanken auszeichnen, muss die Robustheit dieser Umsetzung gegenüber der Rotation um die Z-Achse des Kamerakoordinatensystems untersucht werden, wobei angenommen wird, dass die Z-Achse parallel zum Boden verläuft. Objekte erfüllen bei diesem Gedankenexperiment die Annahmen des Ebenenmodells, d. h. der Disparitätenbereich von Objekten aus der *u-disparity* entspricht einem Disparitätenwert. Weiterhin wird die Disparitätenkarte als fehlerfrei, ohne Rauschen und ohne verdeckte Bereiche angenommen. Das Objekt wird hierbei als Rechteck der Dimension  $20 \times 30$  px modelliert.

Vor der Rotation entsprechen die Einträge in die Histogramme der Disparitäten der jeweiligen Bildspalte, für den Disparitätenwert welcher dem Objekt entspricht, einer absoluten Häufigkeit von 30. Dies resultiert aus der Höhe des Objekts von 30 px und den oben genannten Annahmen. Durch die Rotation erstreckt sich das Objekt über zusätzliche Bildspalten, die Anzahl der Pixel pro Bildspalte, welche auf das Objekt entfallen, sind nicht mehr konstant. Abbildung 3.10 zeigt die absolu-



**Abbildung 3.11**

Der Einfluss der Rotation um  $10^\circ$  (links) bzw.  $20^\circ$  (rechts) auf die detektierte Objektbreite. Der Schwellenwert für die Erstellung des Binärbilds beträgt 30 px, die minimale Konturbreite der Vorfilterung beträgt 15 px. Durch die Rotationen ändern sich die Detektionsgrenzen und die detektierten Objektbreiten.

te Häufigkeit der Klasseneinträge für die entsprechende Disparität des Objekts als blaue Balken, nachdem das Objekt um  $25^\circ$  rotiert wurde. Das Ergebnis der Objektdetektion ist von der Höhe des Schwellenwerts während der Bildung des Binärbilds abhängig. Ein Schwellenwert kleiner oder gleich 30 würde das originale Objekt detektieren, wobei durch die Schwellenwertbildung keine Fehler in den Abmessungen erzeugt werden. Ein Schwellenwert größer 30 klassifiziert das Objekt als Rauschen, wodurch es im Binärbild nicht dargestellt wird. Für das rotierte Objekt gilt dies nicht. Aufgrund der variablen Anzahl an Objektpixel pro Bildspalte über die Breite des Objekts führt die Erzeugung des Binärbilds zu einer reduzierten Objektbreite in Abhängigkeit von der Höhe des Schwellenwerts. Die zentralen Bildspalten weisen nach der Drehung eine höhere Anzahl an Objektpixel auf. Werden die gültigen Schwellenwerte des Falls ohne Rotation angewendet, so wird das Objekt in diesen Spalten als solches detektiert. Wird auf das Binärbild bzw. auf die enthaltenen Konturen eine Vorfilterung angewendet, welche auf der Objektbreite basiert, so kann die Rotation eines Objekts durch die reduzierte abgebildete Breite zu ausbleibenden Detektionen führen.

Abbildung 3.11 zeigt den Einfluss einer Rotation auf die detektierte Objektbreite für Objekte mit Breiten und Höhen bis zu 100 px. Dabei beträgt der Schwellenwert für die Erzeugung des Binärbilds 30 px. Weiterhin wird eine Filterung der Konturen angewandt, wobei die minimale Konturbreite 15 px beträgt. Der Detektionsbereich für die Objekthöhe und -breite ohne Rotation wird durch die schwarze Linie dargestellt. Alle Objekte mit Dimensionen innerhalb dieses Bereichs werden bei  $0^\circ$  Rotation und den gegebenen Schwellenwerten detektiert. Die rote Linie stellt die detektierte Breite von Objekten ohne Rotation dar, diese entspricht der tatsächlichen Objektbreite auf der Bildebene. Durch die Rotationen entsteht jeweils innerhalb der ursprünglichen Detektionsgrenze eine Zone, welche Objektdimensionen beschreibt, die unter Rotation nicht detektiert werden können. Die Größe dieser Zone steigt

mit dem Rotationswinkel. Die benötigte Mindestobjektbreite erhöht sich bei einer Rotation um  $10^\circ$  für die Minimalhöhe um 5 px, bei  $20^\circ$  um 10 px. Abbildung 3.11 zeigt auch, dass eine Rotation die Detektionsgrenze an anderen Stellen erweitert. Beide abgebildeten Fälle zeigen eine Detektion von Objekten mit einer Breite unter 15 px ab einer spezifischen Objekthöhe, was auf die oben beschriebene Erhöhung der Objektbreite auf dem Bildsensor unter Rotation zurückzuführen ist. Weiterhin zeigt sich im Fall der Rotation um  $20^\circ$  deutlich der ebenfalls bereits beschriebene Effekt, dass rotierte Objekte eine höhere Anzahl an Objektpixel in den zentralen Bildspalten aufweisen. Aus diesem Grund können Objekte mit einer Höhe kleiner des Schwellenwerts für die Binärbilderzeugung detektiert werden, wenn auch mit reduzierter Objektbreite. Für die Objekte wird außerdem die detektierte Breite dargestellt. Diese ändert sich mit dem Höhen-Breitenverhältnis des Objekts sowie dem Winkel. Bei einem Höhen-Breitenverhältnis kleiner eins reduziert sich die Objektbreite, bei einem Verhältnis größer eins erhöht sie sich.

Aufgrund der Verwendung des Punktmodells spielt die Breitenvariation in dieser Arbeit keine Rolle. Wird das Objektmodell um diese Dimensionen erweitert, muss die Änderung unter Wanken berücksichtigt werden. Weiterhin ist dabei zu bedenken, dass Algorithmen zur Erstellung der Disparitätenkarte, welche *block matching* Verfahren verwenden, die Objektgrenzen meist nicht pixelgenau abbilden [54]. Die Verschiebung der Detektionsgrenze unter Wanken reduziert die Möglichkeit kleine Objekte zu detektieren. Bei statischen Schwellenwerten ist der Effekt besonders für entfernte Objekte relevant, da die abgebildete Fläche des Objekts auf der Bildebene mit der Entfernung sinkt. Eine Anpassung der Schwellenwerte in Abhängigkeit des Wankwinkels kann den Effekt negieren, sorgt aber potenziell für vermehrte Falschdetektionen aufgrund des Rauschens in der Disparitätenkarte. Optimierungen des *u-disparity* Algorithmus, welche die Rotation der Disparitätenkarte berücksichtigen, können hierbei ein Ansatz sein. Werden keine Maßnahmen ergriffen, müssen die Limitationen bei der Systemauslegung berücksichtigt werden. Trotz der beschriebenen Limitationen ist die Objektdetektion basierend auf der *u-disparity* ein valider Ansatz, welcher die Anforderungen an die generische Objektdetektion erfüllt.

#### 3.5.3 Modellbasierte Objekterkennung

In Kapitel 3.5.2 wurden regelbasierte Ansätze vorgestellt, um generische Objekte aus Stereobildpaaren zu extrahieren. Da die Anforderung an die Generalität der Objekte von der modellbasierten Objekterkennung nicht zu erfüllen ist, wurde diese bisher ausgeschlossen. Die unüberschaubare Menge an Objektklassen macht die generische Objektbeschreibung aufgrund von Merkmalen quasi unmöglich [93]. Wird die Einschränkung gelockert und die zu detektierenden Objektklassen auf eine definierte Untermenge reduziert, können modellbasierte Verfahren eine Alternative zur regelbasierten Objektdetektion darstellen. Die Klasse des detektierten Objekts wird dabei als zusätzliche Information extrahiert.

[94] und [95] beschreiben die rapide Entwicklung dieses Felds während der letzten Jahrzehnte. Um die Jahrtausendwende wurden einfache Merkmale wie in [96] verwendet, um Objektmodelle zu erstellen und in Bildern wiederzufinden. [97] gibt einen Überblick über verschiedene Merkmale. Basierend auf diesen Arbeiten wurden vermehrt Methoden des maschinellen Lernens eingesetzt, wobei mit Hilfe von manuell generierten Merkmalsvektoren „*shallow learning*“-Modelle trainiert wurden.

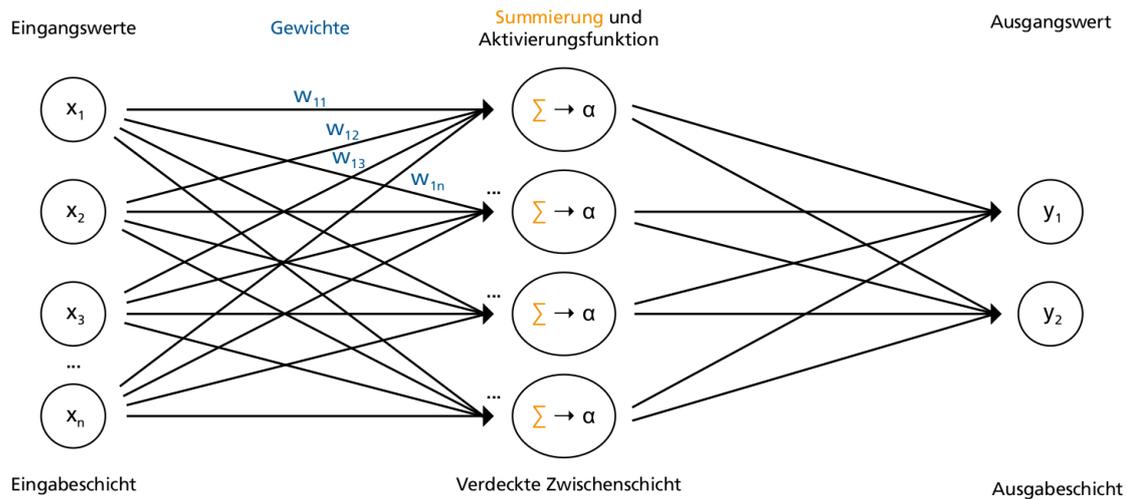
Der Fokus lag dabei auf dem *feature engineering*, d. h. dem Erzeugen von möglichst optimalen Merkmalsvektoren. [94] listet einige relevante Veröffentlichungen dieses Bereichs auf.

[98], [94] und [95] beschreiben die Veröffentlichung von [99] und [100] als einen Wendepunkt in dieser Domäne. Die Verwendung von *Convolutional Neural Networks (CNN)* zur Klassifikation und Lokalisierung von Objekten in den Bildern des „ImageNet“-Datensatzes [101] zeigten eine erhebliche Verbesserung zu den bisherigen Methoden, wodurch *Deep Learning* nicht nur in den Fokus der Forschung, sondern auch der Öffentlichkeit gelangte. Aufgrund dieser Errungenschaften wird in dieser Arbeit, neben der regelbasierten Objektdetektion, auch eine modellbasierte Variante verwendet, um die Vor- und Nachteile beider Methoden gegenüberzustellen.

### 3.5.3.1 Maschinelles Lernen und Deep Learning

Im maschinellen Lernen werden Modelle gebildet, welche ihre Wissensrepräsentation durch Training erwerben. Die aus Beispielen erworbene Erfahrung bezüglich einer spezifischen Domäne kann nachfolgend auf unbekannte, aber ähnliche Daten transferiert werden [102]. Es existieren verschiedene Varianten des Lernens. Diese sind das überwachte, das unüberwachte und das bestärkende Lernen. Hierbei unterscheiden sich neben den zu lösenden Problemklassen auch die Eingangs- und Ausgangsdaten der Modelle. Das überwachte Lernen verwendet Trainingsdaten in Kombination mit Labels, welche dem richtigen Ausgabewert bezogen auf den jeweiligen Input entsprechen. Diese Form des Lernens wird für Klassifizierungs- und Regressionsaufgaben benötigt. Unüberwachte Lernverfahren benötigen keine Labels, sondern nur die Eingangsdaten, um Muster im Datensatz zu erkennen und somit charakteristische Eigenschaften der Daten zu extrahieren. Diese Verfahren werden für das *clustering* oder die Dimensionsreduktion von Datensätzen verwendet. Bestärkende Lernverfahren nutzen Belohnungen als Feedback, um einem Agenten eine Handlungsstrategie anzulernen, beispielsweise das Steuern eines autonomen Fahrzeugs. Das Python Modul „Scikit-learn“ [103] beschreibt in seiner Dokumentation eine Vielzahl der verfügbaren Modelle des maschinellen Lernens, und bietet einen Überblick über deren spezifische Anwendungsfelder mit Hilfe eines nach Problemfeldern und Datenverfügbarkeit sortierten Flussdiagramms [104].

Künstliche neuronale Netze gehören zu den Standardmodellen des maschinellen Lernens. Sie finden ihren Ursprung im Versuch, die Informationsverarbeitung von biologischen neuronalen Netzen mathematisch abzubilden [105]. Die Netze können als Graphen dargestellt werden, welche aus einzelnen Neuronen bestehen, die durch gewichtete Kanten verbunden sind. Die topologischen Möglichkeiten sind vielfältig, jedoch werden meist die Neuronen in aufeinanderfolgende Schichten (*layer*) geordnet, siehe Abbildung 3.12. Dabei ist jedes Neuron einer Schicht mit allen Neuronen der vorherigen und folgenden Schicht durch Kanten verbunden, jedoch nicht mit Neuronen innerhalb einer Schicht. Die Dimension der Eingangsschicht entspricht der Größe des Merkmalsvektors, die Ausgangsschicht dem problemspezifischen Ausgangsvariablen des Modells. Dazwischen können beliebig viele verborgene Schichten mit einer beliebigen Anzahl Neuronen liegen. Ein einzelnes Neuron stellt sich mathematisch durch die Summe seiner gewichteten Eingänge sowie eines Bias dar, worauf eine nichtlineare Aktivierungsfunktion angewendet wird [106]. Die Gewichte der Signale stellen dabei den trainierbaren Teil eines Neurons und somit des gesamten Netzes dar.

**Abbildung 3.12**

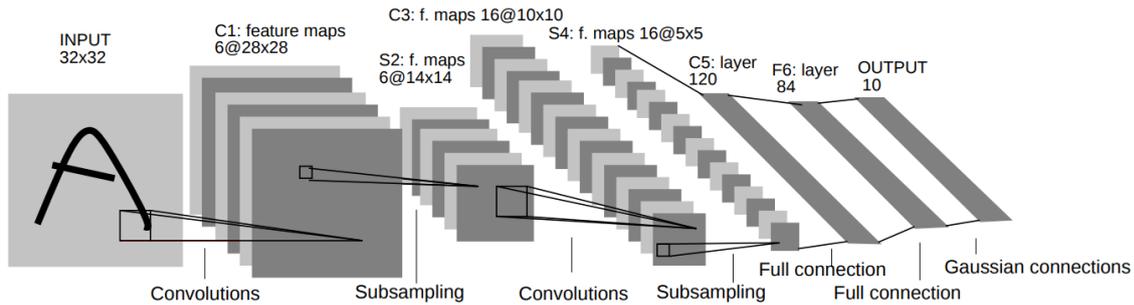
Schematische Darstellung eines künstlichen neuronalen Netzes. Entnommen aus [102].

Die Performanz eines neuronalen Netzes kann nach [100] durch größere Modelle gesteigert werden. Dies kann durch weitere Schichten und mehr Neuronen pro Schicht erreicht werden. Allerdings erhöht sich durch die steigende Komplexität des Modells die Gefahr der Überanpassung an den Trainingsdatensatz, was durch begrenzte Datenverfügbarkeit zusätzlich begünstigt wird. Weiterhin erhöht sich mit den Modellparametern auch der Rechenaufwand für das Training und die Anwendung des Modells. Die öffentliche Verfügbarkeit von großen Datensätzen in Kombination mit leistungsstarker Hardware ermöglichten die Fortschritte der letzten Jahre durch das *Deep Learning*, wobei sich die Modelle durch eine Vielzahl an Schichten und neuen Topologien auszeichnen [95].

Die einzelnen Pixel eines Bildes enthalten wenig Informationen. Ein Pixel im Verbund mit seiner näheren Umgebung kann hingegen vielfältige Informationen bieten. So können lokale Merkmale wie z. B. Kanten durch die Analyse der direkt benachbarten Bildpunkte eines Pixels extrahiert werden, wie in der klassischen Bildverarbeitung durch vielfältige Methoden gezeigt wird. Um diese räumlichen Zusammenhänge in neuronalen Netzen nutzbar zu machen, werden *Convolutional Neural Networks* gebildet. Hierbei wird die Eingangsschicht als zwei- bzw. dreidimensionale Matrix definiert und entspricht dem Eingangsbild in Graustufen (2D) oder einem beliebigen anderen Farbraum (3D). Die Architektur dieser Klasse von neuronalen Netzen besteht nach [106], [95] und [105] üblicherweise aus den folgenden Schichten:

### *Convolutional Layers*

Analog zu den rezeptiven Feldern des menschlichen Auges können lokale Gruppen von Neuronen gebündelt mit der folgenden Schicht verbunden werden, anstatt alle Neuronen zweier Schichten untereinander zu vernetzen. Dies geschieht mit der Hilfe von Faltungsmatrizen, auch Filter genannt, mit frei wählbaren Dimensionen in Breite und Höhe. Die Tiefe entspricht dabei der Eingangsschicht. Eine solche Matrix kann beispielsweise die Dimensionen  $(3 \times 3 \times 1)$  für Graustufenbilder, oder  $(3 \times 3 \times 3)$  für Bilder in einem dreidimensionalen Farbraum annehmen. Die Einträge der Faltungsmatrix entsprechen den trainierbaren Parametern. Wird diese Matrix per diskreter Faltung auf die Eingangsschicht angewandt, entsteht ein zweidimensionaler



**Abbildung 3.13**

*LeNet-5* als Beispiel für die Architektur eines CNN. Entnommen aus [107].

Output, auf dessen Elemente eine Aktivierungsfunktion angewendet wird. In einer *convolutional layer* können beliebig viele Filter auf die Eingangsschicht angewendet werden. Somit bestimmt die Anzahl der Filter die Tiefendimension der Schicht. Höhe und Breite werden durch die Randbedingungen der Faltung definiert, wie z. B. die Schrittweite oder das *padding* des Inputs. Wird beispielhaft eine Eingangsschicht mit den Dimensionen  $(7 \times 7 \times 3)$  und zwei Filter der Dimensionen  $(3 \times 3 \times 3)$  mit der Schrittweite 2 gefaltet, entsteht eine Schicht mit den Dimensionen  $(3 \times 3 \times 2)$ .

Die Parameter einer Faltungsmatrix spiegeln ein zu extrahierendes Merkmal aus einer lokalen Umgebung wider. Durch die Aneinanderreihung von *convolutional layers* wird die Komplexität der zu extrahierenden Merkmale erhöht. So können in den ersten Schichten einfache Merkmale wie Kanten detektiert werden, während tiefere Schichten komplexe Merkmale wie beispielsweise Kreise repräsentieren. Durch die Verwendung der Faltungsmatrizen wird die Anzahl der Parameter im Vergleich zu herkömmlichen neuronalen Netzen erheblich reduziert.

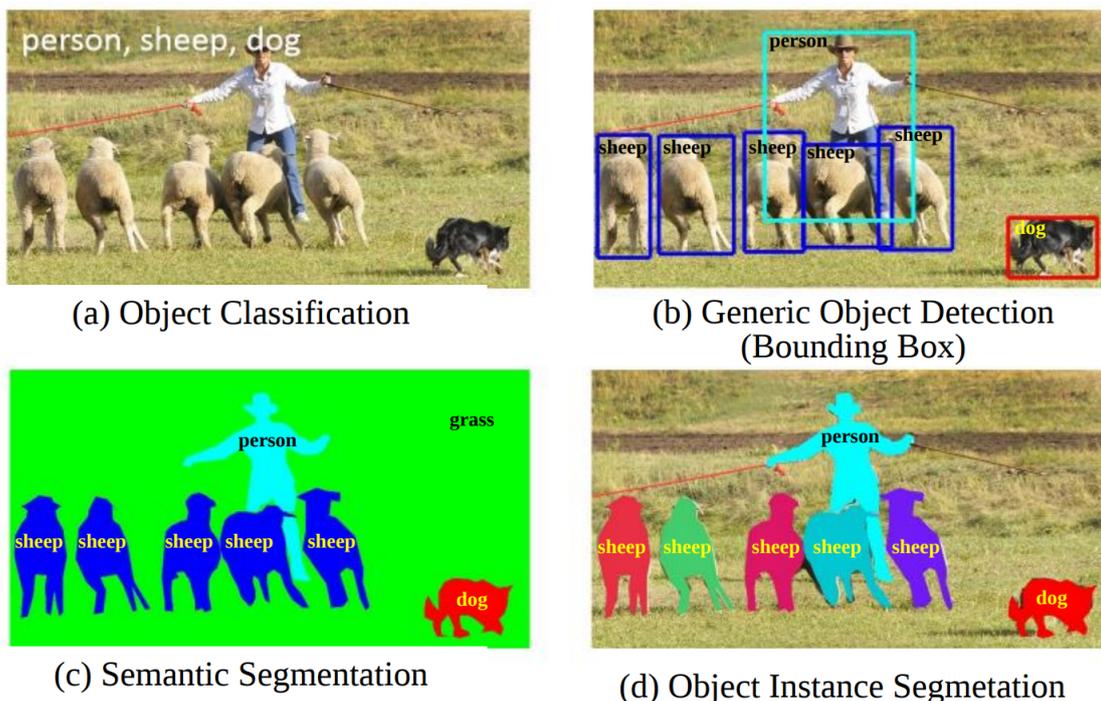
### *Pooling Layers*

Um die Dimensionen der Schichten zu Gunsten einer verringerten Parameterzahl und folglich geringerer Gefahr der Überanpassung reduzieren zu können, werden *pooling layer* verwendet. Diese enthalten keine trainierbaren Parameter. Es werden zweidimensionale Filter geringer Ausdehnung verwendet, auf deren Umgebung eine bestimmte Operation ausgeführt wird. Diese kann zum Beispiel das *max pooling* sein, welche das Maximum der Filterumgebung extrahiert. Durch die Filterdimension und die Schrittweite wird die Dimension der Ausgangsschicht definiert. Große Filterdimensionen und Schrittweiten sind jedoch zu vermeiden, um die in den Schichten repräsentierte Information nicht zu stark zu beschneiden.

### *Fully Connected Layers*

Diese Schichten entsprechen der herkömmlichen Architektur der neuronalen Netze. Der Unterschied zu den *convolutional layers* besteht darin, dass alle Neuronen zwischen zwei Schichten miteinander verbunden sind und dabei jede Kante ein Gewicht als trainierbarer Parameter darstellt.

Die *convolutional layers* werden aufgrund der beschriebenen Eigenschaften auf der Eingangsseite eines CNNs verwendet [106]. Eine oder mehrere solcher Schichten werden oft durch ein *pooling layer* ergänzt. Die Netze schließen mit einem oder



**Abbildung 3.14**

Abgrenzung der Objekterkennung von verwandten Aufgaben. Entnommen aus [98].

mehreren *fully connected layer* ab, wobei die finale Schicht den Output enthält und somit den herkömmlichen neuronalen Netzen gleicht. Als Beispiel kann hier eines der ersten CNNs, namentlich das Netz "LeNet-5" [107] dienen, welches zur Klassifizierung handschriftlicher Ziffern entwickelt wurde (siehe Abbildung 3.13). Auf den Input, ein  $(32 \times 32 \times 1)$  Grauwertbild, folgen alternierend jeweils zwei *convolutional* (Kürzel *C*) und *pooling layer*. Letztere werden in der Abbildung als *subsampling* mit dem Kürzel *S* bezeichnet. Die Faltungsmatrizen der *convolutional layer* haben beide Male die Dimension  $(5 \times 5)$ , wobei die erste Schicht sechs Filter und die zweite Schicht 16 Filter anwendet. Die *pooling layer* werden mit Filtern der Dimension  $(2 \times 2)$  betrieben. Auf diese Schichten folgen drei *fully connected layer* mit 120, 84 und 10 Neuronen, dies entspricht in der finalen Schicht jeweils einem Neuron für jede Ziffer von null bis neun.

#### 3.5.3.2 Stand der Technik

Zur Darstellung des aktuellen Forschungsstands unterscheidet [98] vier Anwendungsfälle, welche mit der modellbasierten Objekterkennung eng verbunden sind. Abbildung 3.14 zeigt die Unterschiede zwischen Objektklassifikation (*Object Classification*), Objekterkennung (*Object Detection*), semantischer Segmentierung (*Semantic Segmentation*) und Objektinstanzsegmentierung (*Object Instance Segmentation*) auf. Die Klassifikation liefert nur die Information, welche Objektklassen aus dem Satz der gelernten Kategorien im Eingangsbild vertreten sind, ohne dabei Informationen über die Objektposition auf der Bildebene zu geben. Die Objekterkennung vereint die Klassifikation mit der Objektdetektion, und gibt somit Objektposition und -klasse aus. Die Position wird durch eine ROI definiert. Eine Klassifikation eines jeden Pixels des Bildes nach den gelernten Klassenkategorien wird semantische Seg-

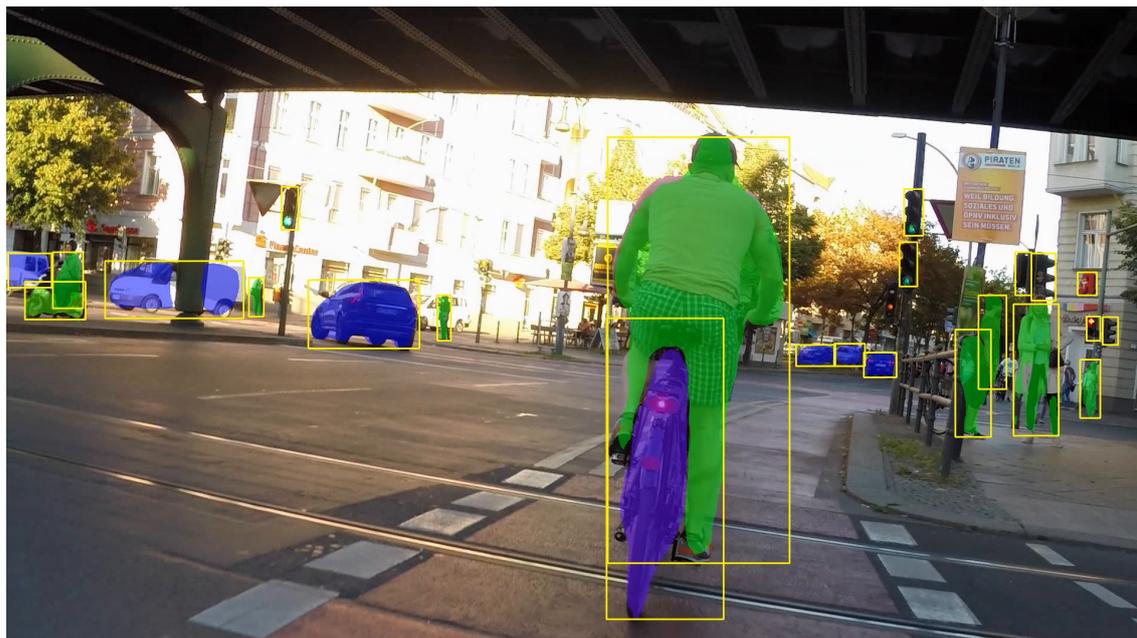
mentierung genannt. Dabei wird keine Information über einzelne Objektinstanzen extrahiert. In Abbildung 3.14c werden die Klassen durch unterschiedliche Farben dargestellt. Die blauen Pixel entsprechen der Klasse „Schaf“, wobei nicht zwischen den einzelnen Schafen unterschieden werden kann. Dies ist erst durch die Objektinstanzsegmentierung möglich, welche die einzelnen klassifizierten Pixel in Objektinstanzen gruppiert. Somit kann jedes Objekt pixelgenau vom Hintergrund oder anderen Objekten abgegrenzt werden. Wie beim menschlichen Gehirn kann auch in neuronalen Netzen gelerntes Wissen auf andere Aufgaben übertragen werden. So werden für die dargestellten Aufgaben oft trainierte Netze aus der Objektklassifikation verwendet [94]. Hierbei werden meist die trainierten *convolutional layer* weiterverwendet und dabei als *backbone network* bezeichnet.

In den letzten Jahren haben sich zwei Ansätze für die Objekterkennung entwickelt, der einstufige und der zweistufige Ansatz [94] [98]. Letzterer schlägt zuerst Bildregionen vor, welche potenziell Objekte enthalten. Die Aussage ist dabei unabhängig von der Objektklasse. Anhand dieser *region proposals* kann eine Objektklassifikation ausgeführt werden. Weiterhin wird mit Hilfe von Regression die ROI optimiert. In verschiedenen Veröffentlichungen wurden die einzelnen Teilschritte, welche aus eigenständig trainierbaren Modellen bestehen, optimiert und in ein einziges CNN verlagert, wodurch echtes „*end-to-end-learning*“ möglich wurde. Der erste Vertreter des zweistufigen Ansatzes ist das *Region Based CNN* (R-CNN) [108]. Im ersten Schritt werden dabei eine feste Anzahl an *region proposals* mit Hilfe des *selective search* Algorithmus [109] festgelegt. Für alle Objektvorschläge werden mit Hilfe eines CNN, welches auf [99] basiert, Merkmalsvektoren extrahiert. Diese werden wiederum in klassenspezifischen *Support Vector Machines* verarbeitet, wodurch die Objektklasse bestimmt und die Position und Größe der ROI optimiert wird. Dieser mehrstufige Ansatz ist jedoch ineffizient in der Anwendung, weshalb *Fast R-CNN* [110] und *Faster R-CNN* [111] entwickelt wurden, welche den Region Proposal Algorithmus in das CNN verlagern und somit erheblich effizienter arbeiten. Der einstufige Ansatz weist gegenüber dem zweistufigen Ansatz eine schlechtere Genauigkeit auf, arbeitet aber erheblich schneller, da keine *region proposal* Algorithmen verwendet werden. Eine der am meisten verbreiteten Veröffentlichungen ist [112]. Dabei werden multiple ROIs und Klassenwahrscheinlichkeiten in einem CNN prädiziert, welches als Regressionsproblem formuliert ist. Das Bild wird in ein Raster unterteilt, wobei jede Rasterzelle einen Satz ROIs enthält. Basierend auf jeder ROI wird der Versatz und die Größe der ROI zum Objekt sowie die Klassifizierung prädiziert. Als Backbone kommt dabei das Netz aus [113] zum Einsatz.

Die Übersichten von [98], [94] und [95] bieten einen hervorragenden Überblick über die Entwicklungen in diesem Bereich und geben Informationen über das Abschneiden der verschiedenen Modelle in den gängigen Benchmarks.

### 3.5.3.3 Mask R-CNN

In der Analyse von NCS-Daten existieren keine Echtzeitanforderungen an die Algorithmen. Generell ist eine höhere Detektionsrate oder ROI-Genauigkeit einer effizienteren Implementierung vorzuziehen. Daher können die einstufigen Ansätze ihren Vorteil in der NCS-Analyse nicht ausspielen. Schwerer wiegt jedoch ein weiteres Problem, welches auf den extrahierten ROIs basiert und beide bisher besprochenen Ansätze betrifft.



**Abbildung 3.15**

Objektinstanzsegmentierung mit *Mask R-CNN* in einer NCS-Szene. Die Masken werden farblich nach Klasse codiert dargestellt: Mensch (Grün), Pkw (Blau), Fahrrad (Lila). Zusätzlich werden die ROIs der Objekte in Gelb ausgegeben. Ampeln werden ohne Maske, aber mit ROI dargestellt.

Erdenkt man sich in einem Gedankenexperiment einen perfekten Objektdetektor, so liefert dieser eine genaue, rechteckige *Bounding Box* um jedes im Bild präsente Objekt, sofern es in den gelernten Objektklassen vertreten ist. Um die berechnete ROI in das in Kapitel 3.5.1 beschriebene Punktmodell zu überführen, muss das Zentrum der ROI sowie eine repräsentative Disparität für das Objekt berechnet werden. Im Gegensatz zur stereobasierten Objektdetektion liegen hier keine Disparitätenbereiche vor, welche den Bereich des gesuchten Disparitätenwerts des Punktmodells beschränken und dessen Bestimmung unterstützen. Werden in der Disparitätenkarte der ROI überproportional viele Pixel mit falschen Disparitäten angegeben, z. B. aufgrund von Verdeckung durch andere Objekte, können einfache Annahmen zur Bestimmung der repräsentativen Disparität des Objekts versagen. Wird beispielsweise der Medianwert aller Disparitäten der ROI verwendet, reicht eine ROI mit weniger als 50 % Flächenanteil des relevanten Objekts, um eine falsche Disparität zu berechnen.

Die einfachste Lösung des beschriebenen Problems liegt in der Verwendung eines alternativen Konzepts. Abbildung 3.14 zeigt neben der Objektklassifizierung, Objekterkennung und semantischen Segmentierung auch die Objektinstanzsegmentierung als ähnliche Problemstellung auf. Diese Klasse an Modellen liefert keine rechteckigen ROIs für jede Objektinstanz, sondern bündelt diese pixelweise. Die Disparitätenkarte eines Bilds kann mit Hilfe dieser instanzbasiert gelabelten Pixel auf die Disparitäten des jeweiligen Objekts reduziert werden, wodurch einfache Methoden wie der Medianwert valide Ergebnisse liefern.

*Mask R-CNN* [114] ist eine Weiterentwicklung des bereits vorgestellten *Faster R-CNN*, und erweitert dessen Netz um einen Zweig zur Schätzung der Segmentierungsmaske. Somit wird neben der Objektklasse und dessen ROI auch die Segmen-

tierungsmaske für jede Instanz bestimmt. Das Punktmodell kann mit Hilfe beider Ausgaben, der ROI sowie der Maske, befüllt werden. Die Position des Objekts auf der Bildebene wird durch das Zentrum der ROI bestimmt. Die Disparität des Objekts berechnet sich durch die Anwendung der Segmentierungsmaske auf die Disparitätenkarte. Die verbleibenden Werte werden um offensichtlich invalide Disparitäten bereinigt und der Medianwert bestimmt.

Das Verhalten der Objekterkennung unter Wanken ist von den Trainingsdaten bzw. der Augmentierung der Trainingsdaten abhängig. Sind rotierte Objekte in den Trainingsdaten enthalten, oder werden sie für das Training künstlich rotiert, so kann das Netz auch dies erlernen und folglich auch um den Wankwinkel rotierte Objekte erkennen.

In dieser Arbeit wird eine Implementierung von [115] für das „Tensorflow“-Framework verwendet, welche auf Basis des „COCO“-Datensatzes [116] trainiert wurde. Der Datensatz besteht aus 330 000 Bildern und enthält 80 Objektklassen, darunter viele für NCS relevante Verkehrsteilnehmer\*innen oder alltägliche Objekte. Das extensive Training mittels dieses Datensatzes lässt eine prototypische Anwendung für den vorgestellten Anwendungsfall zu. Anpassungen der Trainingsdaten, trainierten Objektklassen bzw. der trainierten Gewichte wurden im Sinne einer prototypischen Implementierung nicht vorgenommen. Eine Bereitstellung eines eigenen Datensatzes und ein darauf basiertes Training ist aufgrund des zeitlichen Aufwands nicht möglich. Abbildung 3.15 zeigt eine Szene aus den aufgenommenen NCS-Daten, wobei die ROI sowie die Segmentierungsmaske für alle detektierten Objektinstanzen, ausgenommen Ampeln, dargestellt werden.

Abgesehen von den in diesem Kapitel dargestellten Einschränkungen, kann die modellbasierte Objekterkennung die stereobasierte Objektdetektion ersetzen. Beide Varianten können das Punktmodell eines Objekts befüllen, welches in der Objektverfolgung benötigt wird, siehe Kapitel 3.6.

## 3.6 Objektverfolgung

Die aus den Bildpaaren extrahierten und zu Punktmodellen erweiterten Objekte müssen für eine effektive Weiterverarbeitung zeitlich gekoppelt werden. Dies bedeutet, dass die Objekte aufeinanderfolgender Bildpaare einander zugeordnet werden müssen. Aus den zugeordneten Objektmessungen kann der zeitliche Verlauf des Objektzustands bestimmt werden, wobei sich dieser mit der Zeit aufgrund der Bewegung der Objekte sowie der Eigenbewegung des Sensors im Raum verändert. Diese Aufgabe wird als Objektverfolgung oder Objekttracking bezeichnet, wobei die Aufgabe aus den Teilbereichen Datenzuordnung (*Data Association*) und Zustandsschätzung besteht [117]. Wird zu jedem Messzeitpunkt ein Objekt wiederholt detektiert, ohne dass FPs auftreten, so ist die Datenzuordnung trivial. Die Existenz falsch positiver und falsch negativer Messungen sowie die Tatsache, dass einzelne Objektinstanzen sensorspezifisch mehrere Objektdetektionen hervorrufen können, erschweren das *Tracking* mehrerer Objekte. Die sogenannte Mehrobjektverfolgung (*Multiple Object Tracking, MOT*) muss diese Probleme in der Datenzuordnung lösen. Werden aufgrund von Mehrfachdetektionen oder FPs fälschlicherweise nichtexistente Objekte angenommen und verfolgt, so müssen diese Instanzen z. B. mit Hilfe einer geschätzten Existenzwahrscheinlichkeit klassifiziert und verworfen werden [51].

### 3.6.1 Stand der Technik

Aktuelle Implementierungen der Mehrobjektverfolgung basieren nach [51] und [118] meist auf rekursiven Bayes-Filtern. Es werden Kalman-Filter bzw. bei nichtlinearen Prozess- oder Messgleichungen auch dessen Erweiterungen *Extended Kalman-Filter (EKF)* und *Unscented Kalman-Filter (UKF)* verwendet. Für die gleichzeitige Verfolgung mehrerer Objekte wird pro Objekt eine Filterinstanz angelegt. Die Gesamtheit des *Trackings* wird somit als Multi-Instanzen-Filter beschrieben. Durch mögliche Mehrdeutigkeiten der Datenassoziation, welche die Detektionen den einzelnen Filtern zuordnet, wird die Objektverwaltung benötigt.

#### 3.6.1.1 Das Kalman-Filter

Die Aufgabe des von [119] und [120] beschriebenen Kalman-Filters besteht in der Objektverfolgung aus der Zustandsschätzung der im Sichtfeld befindlichen Objekte. Der Zustand  $\mathbf{x}_k$  eines Objekts zum diskreten Zeitpunkt  $t_k$  lässt sich durch dessen Position sowie weiteren Zustandsgrößen wie Orientierung, Geschwindigkeit oder Drehrate beschreiben. Eine Bedingung für die Verwendung des Kalman-Filters ist die Darstellung eines Messmodells zur Beobachtung des Zustands anhand von Messungen, sowie ein Prozessmodell zur Beschreibung der Dynamik des beobachteten Systems.

Ist eine Reihe von Messungen  $Z$  zu diskreten Zeitpunkten gegeben, so beschreibt Gleichung 3.3 das Messmodell mit der Beobachtungsmatrix  $\mathbf{H}_k$  und dem normalverteilten Messrauschen  $\omega_k$  mit Mittelwert null. Die Kovarianzmatrix der Messung  $\mathbf{R}_k$  ergibt sich nach Gleichung 3.6. Das Prozessmodell, welches den Zustand des Objekts präzisieren soll, wird durch Gleichung 3.4 beschrieben. Hierbei stellt  $\mathbf{F}_k$  die Übergangsmatrix von Zustand  $\mathbf{x}_k$  nach  $\mathbf{x}_{k+1}$  dar, und beschreibt somit die Dynamik des Objekts. Der stochastische Modellfehler des Prozessmodells wird wiederum durch das normalverteilte, mittelwertfreie Rauschen  $\nu_k$  abgebildet. Die Kovarianzmatrix des Prozessrauschens  $\mathbf{Q}_k$  ergibt sich aus dem Erwartungswert nach Gleichung 3.5. Da auf die Objekte keine bekannten externen Steuergrößen  $\mathbf{u}_k$  wirken, entfällt der Term  $\mathbf{G}_k \cdot \mathbf{u}_k$ , wobei  $\mathbf{G}_k$  die Steuermatrix darstellt.

$$Z = \{\mathbf{z}_1, \dots, \mathbf{z}_k\} \quad (3.2)$$

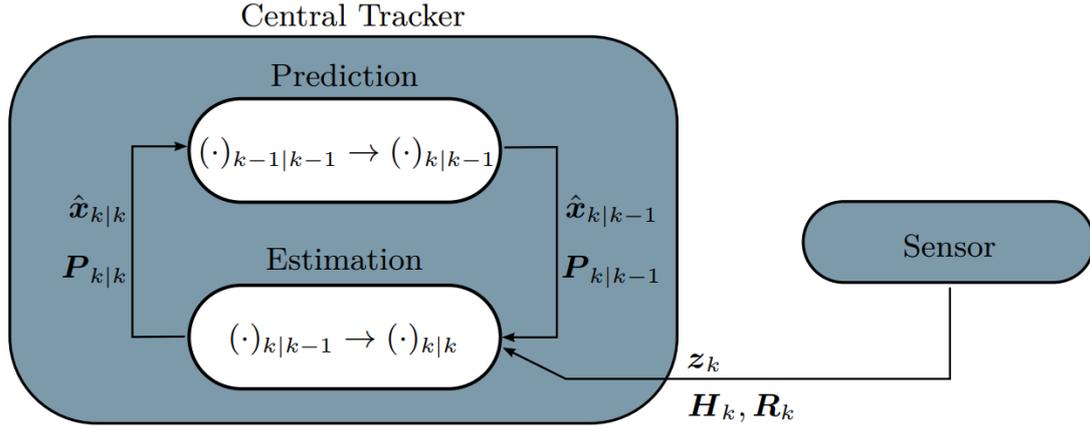
$$\mathbf{z}_k = \mathbf{H}_k \cdot \mathbf{x}_k + \omega_k \quad (3.3)$$

$$\mathbf{x}_{k+1} = \mathbf{F}_k \cdot \mathbf{x}_k + \mathbf{G}_{k+1} \cdot \mathbf{u}_{k+1} + \nu_k \quad (3.4)$$

$$\mathbf{Q}_k = \mathbf{E}\{\nu_k \nu_k^T\} \quad (3.5)$$

$$\mathbf{R}_k = \mathbf{E}\{\omega_k \omega_k^T\} \quad (3.6)$$

In Abbildung 3.16 wird der Ablauf des Kalman-Filters dargestellt, welcher aus den zwei iterativ zu wiederholenden Schritten, der Prädiktion und der Korrektur, besteht. Beide Schritte werden für jeden Zeitpunkt  $t_k$  durchgeführt, für den eine Messung  $\mathbf{z}_k$  vorliegt. Der Zustand eines Objekts zum Zeitpunkt  $t_{k-1}$  wird durch dessen Mittelwert  $\hat{\mathbf{x}}_{k-1|k-1}$  und Kovarianzmatrix  $\mathbf{P}_{k-1|k-1}$  beschrieben. Mit Hilfe

**Abbildung 3.16**

Prädiktions- (*Prediction*) und Korrekturschritt (*Estimation*) des Kalman-Filters. Entnommen aus [118]

des Prozessmodells nach Gleichungen 3.7 und 3.8 wird der Zustand zum Zeitpunkt  $t_k$  prädiziert. Unter Zuhilfenahme des Messmodells kann der Korrekturschritt ausgeführt werden. Hierbei wird der prädizierte Zustand mit Mittelwert  $\hat{\mathbf{x}}_{k|k-1}$  und Kovarianzmatrix  $\mathbf{P}_{k|k-1}$  in den Messraum transformiert und die Innovation  $\gamma_k$  und deren Kovarianzmatrix  $\mathbf{S}_k$  nach Gleichungen 3.9, 3.10 und 3.11 berechnet. Unter Verwendung der Filterverstärkung  $\mathbf{K}_k$  nach Gleichung 3.12 kann die korrigierte Schätzung des Zustands mit Gleichungen 3.13 und 3.14 bestimmt werden.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1} \cdot \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_k \cdot \mathbf{u}_k \quad (3.7)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \cdot \mathbf{P}_{k-1|k-1} \cdot \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (3.8)$$

$$\hat{\mathbf{z}}_{k|k-1} = \mathbf{H}_k \cdot \hat{\mathbf{x}}_{k|k-1} \quad (3.9)$$

$$\gamma_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \quad (3.10)$$

$$\mathbf{S}_k = \mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T + \mathbf{R}_k \quad (3.11)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^T \cdot \mathbf{S}_k^{-1} \quad (3.12)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \cdot \gamma_k \quad (3.13)$$

$$\mathbf{P}_{k|k} = \hat{\mathbf{P}}_{k|k-1} - \mathbf{K}_k \cdot \mathbf{S}_k \cdot \mathbf{K}_k^T \quad (3.14)$$

### 3.6.1.2 Das Erweiterte Kalman-Filter

Das beschriebene Kalman-Filter setzt ein lineares Prozess- und Messmodell voraus, was für das Messmodell einer Stereokamera nicht erfüllt ist. Für nichtlineare Modelle können die Gleichungen 3.7 und 3.9 durch die Ausdrücke 3.15 und 3.16 ersetzt werden, wobei  $f$  und  $h$  nichtlineare Abbildungen darstellen. Für die Berechnung der Kovarianzen in Gleichung 3.14 und 3.8 werden die nichtlinearen Abbildungen durch eine Taylor-Entwicklung um den jeweiligen Entwicklungspunkt linearisiert [121]. Die Matrizen  $\mathbf{F}_{k-1}$  bzw.  $\mathbf{H}_k$  können somit durch die Jacobi-Matrizen aus Gleichungen 3.17 und 3.18 ersetzt werden. Diese Anpassungen werden durch den Begriff Erweitertes Kalman-Filter (*EKF*) beschrieben.

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \quad (3.15)$$

$$\hat{\mathbf{z}}_{k|k-1} = h(\hat{x}_{k|k-1}) \quad (3.16)$$

$$\mathbf{F}_{k-1}^J = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k-1|k-1}, u_k} \quad (3.17)$$

$$\mathbf{H}_k^J = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_{k|k-1}} \quad (3.18)$$

### 3.6.1.3 Datenassoziation

Die potenzielle Präsenz mehrerer Objekte im Sichtfeld der Sensorik verlangt nach einem *MOT*-Ansatz. Zur Verwendung des Multi-Instanzen-Filters müssen die Messungen eines Zeitschritts den jeweiligen Filterinstanzen zugeordnet werden. Durch Verdeckungen oder FNs können Messungen für einzelne Filter ausbleiben, FPs können wiederum zusätzliche Mehrdeutigkeiten hervorrufen. Die Methoden zur Datenzuordnung können in harte und weiche Zuordnungsverfahren unterschieden werden.

Die harten Zuordnungsverfahren teilen jedem Filter eine Messung zu. Ein Abstandsmaß, wie die Mahalanobis-Distanz, kann verwendet werden um mit Hilfe von Nächster-Nachbar-Methoden (*Nearest Neighbor, NN*) eine Zuordnung zu finden. Lokale Methoden ordnen dabei jedem Filter die Messung mit minimalen Distanzmaß zu, wobei einzelne Messungen auch mehrfach zugeordnet werden können. Globale Methoden optimieren die Zuordnungen bezüglich einer Kostenfunktion, wodurch keine Mehrfachzuordnung möglich ist [51]. In Szenarien mit dicht beieinanderliegenden Objekten führen die harten Zuordnungsverfahren oft zu fehlerhaften Zuordnungen, wodurch Fehler in der Objektverfolgung entstehen. Dabei können sich die Identitäten zweier Objekte vertauschen oder die Verfolgung eines Objekts unterbrochen werden. In Szenarien mit gut zu trennenden Objekten funktionieren die harten Zuordnungsverfahren hinreichend gut [118].

Um fehlerhafte, harte Entscheidungen zur Datenassoziation zu vermeiden, können weiche Zuordnungsverfahren verwendet werden. Die probabilistische Datenassoziation (PDA) löst den Fall, dass mehrere Messungen einem Objekt zugeordnet werden können. Dabei wird jede Messung gewichtet, wodurch deren Einfluss in den Korrekturschritt des Kalman-Filters eingeht. Folglich muss sich nicht für eine potenziell falsche Messung entschieden werden. Um unwahrscheinliche Zuordnungen

ausschließen zu können, werden *Gating*-Verfahren mit Hilfe der Mahalanobis-Distanz eingesetzt [121]. Eine Beschreibung der PDA sowie deren Erweiterung auf den *MOT*-Fall ist in [118] zu finden.

#### 3.6.1.4 Objektverwaltung

Treten bisher unbekannte, detektierbare Objekte in das Sichtfeld eines Sensors ein, muss eine neue Filterinstanz für dieses Objekt angelegt werden. Ebenso müssen Objekte außerhalb des Sichtfelds gelöscht werden. Gleiches gilt für Objekte, welche aus FPs resultieren und folglich keine weiteren Messungen mehr zugewiesen bekommen. In der Praxis treten heuristische und probabilistische Methoden zur Objektverwaltung auf [121]. Erstere zeichnen sich durch ihre Einfachheit aus. Neue Objekte werden durch mehrfache Datenzuordnung über einen willkürlich zu definierenden Zeitraum bestätigt. Bei ausbleibender Zuordnung werden Objekte wiederum verworfen. Methoden, welche auf der Existenzwahrscheinlichkeit eines Objekts beruhen, bieten erweiterte Möglichkeiten falls mehrere Applikationen auf den Ergebnissen der Objektverfolgung basieren und unterschiedliche Anforderungen an die Objektverwaltung bzw. die FPs haben [122].

### 3.6.2 Umsetzung

Die Anforderungen an die Objektverfolgung für die Auswertung von NCS-Daten bezüglich kritischer Situationen unterscheiden sich von jenen aus dem AD- bzw. ADAS-Bereich. Letztere dienen der Beschreibung eines Umfeldmodells, welches zur Realisierung von teilweise sicherheitskritischen Fahrfunktionen dient. Andere Verkehrsteilnehmer\*innen müssen frühzeitig erkannt und verlässlich verfolgt werden, um Fehlalarme und kurzfristige Eingriffe mit entsprechenden Beschleunigungsspitzen und Komfortverlust zu vermeiden. Die Detektion von kritischen Situationen im Radverkehr beschränkt sich hingegen auf eine reduzierte Darstellung des Umfelds. Es gilt dabei die Annahme, dass die Verfolgung von Objekten in einem zu spezifizierenden Nahbereich ausreicht, um jene Situationen zu detektieren, welche zwar als kritisch zu bewerten sind, jedoch keinen kinematischen Trigger der Eigenbewegung auslösen (vgl. Kapitel 1.1.3). Dieser Nahbereich muss vom gewählten Umfeldsensor, hier der Stereokamera, abgedeckt werden.

Diese reduzierten Anforderungen spiegeln sich in der Umsetzung der Objektverfolgung wider. Eine heuristische Objektverwaltung wird dabei mit einem harten Zuordnungsverfahren gekoppelt, um ein einfaches, aber robustes System zu entwerfen. Ein Objekt gilt in dieser Arbeit als valide, sobald es über 30 Zeitschritte verfolgt wurde. Wird für ein Objekt über zehn konsekutive Zeitschritte keine Assoziation hergestellt, wird es aus der aktiven Objektliste gelöscht und die Verfolgung wird beendet. Die Datenzuordnung geschieht über eine NN-Methode. In einem lokalen *Gate* um den prädizierten Zustand des Objekts wird die Messung mit der geringsten Mahalanobis-Distanz zugeordnet. Das *Gate* basiert ebenfalls auf der Mahalanobis-Distanz und repräsentiert eine elliptische Region um den prädizierten Zustand, welcher in den Messraum transformiert wurde [121]. Messungen innerhalb dieser Region kommen für eine Datenzuordnung in Frage, Mehrfachzuordnungen sind dabei erlaubt. Entsprechend der oben benannten Vor- und Nachteile der gewählten Varianten kann diese Wahl die Anforderungen der NCS-Auswertung erfüllen.

Die einzelnen Objektinstanzen werden durch EKFs umgesetzt. Als Prozessmodell dient dabei ein lineares *Constant Velocity (CV)*-Modell, welches die Dynamik aller im Nahbereich auftretenden Objekte abbilden soll und dabei nur deren Translation berücksichtigt. Eine Herleitung des Modells findet sich in [123]. Auf objektklassenspezifische Prozessmodelle wurde an dieser Stelle aus Gründen der Vergleichbarkeit zwischen den verschiedenen Ansätzen der Objekterkennung verzichtet. Der Zustandsvektor besteht aus den kartesischen Weltkoordinaten des Objekts sowie den daraus abgeleiteten Geschwindigkeiten, siehe Gleichung 3.19. Die zugehörige  $6 \times 6$  Übergangsmatrix  $\mathbf{F}$  lässt sich durch Untermatrizen beschreiben.  $\mathbf{I}_3$  stellt dabei die  $3 \times 3$  Einheitsmatrix und  $\mathbf{Z}_3$  die  $3 \times 3$  Nullmatrix dar. Die Zeitdifferenz  $\Delta t$  beschreibt den zu präzifizierenden Zeitschritt von  $t_k$  nach  $t_{k+1}$ . Das Prozessrauschen  $\mathbf{Q}$  ergibt sich nach [124] aus Gleichung 3.21.

$$\mathbf{x} = [X, Y, Z, V_x, V_y, V_z] \quad (3.19)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 \cdot \Delta t \\ \mathbf{Z}_3 & \mathbf{I}_3 \end{bmatrix} \quad (3.20)$$

$$\mathbf{Q} = \mathbf{U} \cdot \mathbf{U}^T \cdot \nu, \quad \text{mit } \mathbf{U} = \left[ \frac{\Delta t^2}{2}, \frac{\Delta t^2}{2}, \frac{\Delta t^2}{2}, \Delta t, \Delta t, \Delta t \right]^T \quad (3.21)$$

Die nichtlineare Abbildung des Zustands- in den Messraum wird durch  $\mathbf{h}$  beschrieben, siehe Gleichung 3.22. Die Herleitung ergibt sich aus Gleichung 2.12. Die linearisierte Abbildung, wie in Kapitel 3.6.1.2 beschrieben, berechnet sich nach Gleichung 3.23.

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \frac{X \cdot f}{Z} + c_u \\ \frac{Y \cdot f}{Z} + c_v \\ \frac{f \cdot B}{Z} \end{bmatrix} \quad (3.22)$$

$$\mathbf{H}^J = \begin{bmatrix} \frac{f}{Z} & 0 & \frac{-X \cdot f}{Z^2} & 0 & 0 & 0 \\ 0 & \frac{f}{Z} & \frac{-Y \cdot f}{Z^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{-f \cdot B}{Z^2} & 0 & 0 & 0 \end{bmatrix} \quad (3.23)$$

Für das Sensormodell einer Stereokamera nimmt [125] unabhängige Fehler für die Bildkoordinaten und die Disparität an, wobei die Größe des Fehlers jeweils von den verwendeten Algorithmen zur Erstellung der Disparitätenkarte und der Objektdetektion abhängen. [59] nennt verschiedene Veröffentlichungen, welche für das hier verwendete *semi-global matching* eine Genauigkeit von bis zu 0,25 px nennen, äußert jedoch Bedenken in Bezug auf das Gesamtsystem und erwartet einen größeren Wert. Aufgrund der unterschiedlichen verwendeten Algorithmen wird das Messrauschen als Optimierungsparameter definiert, die einzelnen Elemente dabei jedoch als unabhängig angenommen. Die erwähnten 0,2px sollen dabei als untere Grenze für den Disparitätenfehler dienen.

## 3.7 Konfliktindikatoren

Der finale Schritt des generalisierten Workflows aus Abbildung 3.2 ist die Berechnung der Konfliktindikatoren. Die Berechnung basiert auf den Objektinstanzen, welche durch die Objektverfolgung generiert und getracked werden. Für jede Instanz und jeden diskreten Zeitpunkt  $t_k$  müssen Indikatoren berechnet werden, welche die Kritikalität des möglichen Konflikts beschreiben. Sie bilden die Basis für die Eventdetektion.

Im Rahmen der stationären Beobachtung des Verkehrsraums werden häufig Konfliktindikatoren verwendet. [126] nutzt Daten der automatisierten Verkehrszählung auf Autobahnen, um anhand der TTC-Verteilung der passierenden Fahrzeuge auf das Unfallrisiko zu schließen. Die TTC wird dabei als Zeitdifferenz bis zu einer Kollision zweier Objekte definiert, welche sich auf Kollisionskurs befinden und ihre aktuelle Geschwindigkeit und Bewegungsrichtung beibehalten. Ein Ergebnis der Studie ist, dass die TTC zwar nicht vom Wetter abhängt, jedoch von der Verkehrsdichte. Eine höhere Dichte führt dabei zu niedrigeren Werten der TTC und folglich zu erhöhtem Unfallrisiko. [11] verwendet neben der TTC noch einen weiteren Indikator. Die *Post Encroachment Time* (PET) beschreibt die Zeitdifferenz, zwischen dem Verlassen der ersten und der Ankunft der zweiten Konfliktpartei in einer lokalen Konfliktzone, in welcher sich die Trajektorien schneiden oder auf ein zu definierendes Mindestmaß annähern. [127] verwendet u. a. ebenfalls TTC und PET als Indikatoren. Dabei wurde nachgewiesen, dass die TTC die Kritikalität aufgrund der Extrapolation oft überschätzt, während die PET Situationen unterschätzt. Der Kollisionskurs als Bedingung für die TTC vernachlässigt gefährliche Annäherungen ohne direkte Kollisionsgefahr als kritischen Event. Insgesamt wurde die PET als zuverlässigster Indikator klassifiziert. In dieser Arbeit werden aufgrund der beschriebenen Erfahrungen ebenfalls die TTC und PET als Konfliktindikatoren implementiert. Beide Indikatoren werden dabei für jeden Zeitschritt der verfolgten Objekte berechnet, anstatt einmal für den jeweiligen Konflikt. Dies soll potenzielle Änderungen der Bewegungsrichtung und Geschwindigkeit widerspiegeln.

Da die Objekte im Sensorkoordinatensystem dargestellt werden, der Sensor wiederum am Lenker des Fahrrads befestigt und in Fahrtrichtung ausgerichtet ist, gilt für den Fall eines Kollisionskurses für die TTC Gleichung 3.24. Dabei ist  $Z$  die Entfernung des Objekts in Z-Richtung,  $V_z$  die Geschwindigkeit des Objekts in Z-Richtung und  $d_z$  die Distanz zwischen Kamerakoordinatensystem und Fahrradfront. Eine negative TTC beschreibt ein sich entfernendes Objekt, dessen Bewegungsrichtung folglich nicht zu einer Kollision führen kann. Ob sich ein Objekt mit positiver TTC auf Kollisionskurs befindet, lässt sich wiederum über dessen X-Position zum Zeitpunkt  $t(Z = d_z)$  bestimmen. Unter der Bedingung konstanter Geschwindigkeit und Kurs berechnet sich die Position in X-Richtung zum Zeitpunkt  $t(Z = d_z)$  aus Gleichung 3.25, wobei  $X_0$  die die Position in X-Richtung zum Zeitpunkt der Berechnung darstellt. Bei einer definierten eigenen Breite von  $b_{ego}$  und einer Objektbreite von  $b_{obj}$  besteht ein Kollisionskurs, wenn Bedingung 3.26 erfüllt ist. In der gewählten Implementierung werden Objekte als Punktmodell dargestellt, deren Abmaße nicht gemessen oder geschätzt werden. Als Alternative kann jedoch ein fester Wert für  $b_{obj}$  gewählt werden. Ist die Objektklasse bekannt, kann dies klassenspezifisch geschehen. Ähnlich verhält es sich mit der tatsächlichen Ausdehnung des Objekts in Z-Richtung. Durch die Darstellung als Punktmodell werden Fehler in der tatsächli-

chen Entfernung zum naheliegendsten Berührungspunkt des Objekts implementiert. Annahmen über diesen Fehler können nur schwer getroffen werden, da dieser von der Objektform und der Güte der Disparitätenkarte abhängig ist.

$$TTC = -\frac{Z - d_z}{V_z} \quad (3.24)$$

$$X_{Kollision} = X_0 + V_x \cdot TTC \quad (3.25)$$

$$|X_{Kollision}| \leq \frac{b_{ego}}{2} + \frac{b_{obj}}{2} \quad (3.26)$$

Die Bestimmung der PET unterteilt sich in die folgenden vier Schritte:

1. Berechnung der Konfliktzone
2. Zeitdifferenz  $\Delta t_{obj}$  bis das Objekt die Konfliktzone verlässt
3. Zeitdifferenz  $\Delta t_{ego}$  bis das eigene Fahrrad die Konfliktzone erreicht
4. Berechnung der PET.

Aufgrund der Kameraausrichtung gilt für eine punktförmige Konfliktzone ohne Ausdehnung bei kreuzenden Trajektorien zwingend die Bedingung 3.27. Es wird davon ausgegangen, dass die Orientierung der Kamera bzw. des Lenkers die aktuelle Fahrtrichtung beschreibt. Kreuzen sich die Trajektorien nicht, sondern nähern sich nur an, so muss eine Ausdehnung für die Konfliktzone definiert werden, welche wiederum durch die eigenen Abmessungen sowie die Objektbreite beeinflusst wird. Auch für kreuzende Trajektorien kann eine Ausdehnung sinnvoll sein, um die Objektabmessungen zu berücksichtigen.  $X_{Konfliktzone}$  besitzt dann einen endlichen, positiven Wert. Die Zeitdifferenz  $\Delta t_{obj}$  bis das Objekt die Konfliktzone verlässt, ergibt sich aus Gleichung 3.28. Hierbei ist zu beachten, dass  $\Delta t_{obj}$  nur unter der Bedingung 3.29 definiert ist, da andernfalls keine Annäherung des Objekts an die Konfliktzone stattfindet.

$$X_{Konfliktzone} = 0 \quad (3.27)$$

$$\Delta t_{obj} = \frac{\Delta X_{Konfliktzone}}{V_x} = \frac{|X| + X_{Konfliktzone}}{|V_x|} \quad (3.28)$$

$$sign(X) \neq sign(V_x) \quad (3.29)$$

$$Z_{Konfliktzone} = Z + V_z \cdot \Delta t_{obj} \quad (3.30)$$

$$\Delta t_{ego} = \frac{Z_{Konfliktzone} - d_z}{V_{ego}} \quad (3.31)$$

$$PET = \Delta t_{ego} - \Delta t_{obj} \quad (3.32)$$

Die Zeitdifferenz  $\Delta t_{ego}$ , welche die Ankunft des eigenen Fahrrads in der Konfliktzone beschreibt, ist durch die Z-Koordinate der Konfliktzone  $Z_{Konfliktzone}$ , die

Distanz zwischen Kamerakoordinatensystem und Fahrradfront  $d_z$  sowie der Eigengeschwindigkeit  $V_{ego}$  definiert. Erstere ergibt sich aus Gleichung 3.30, mit der Objektposition  $Z$  und -geschwindigkeit  $V_z$  in  $Z$ -Richtung sowie der aus Gleichung 3.28 bekannten Zeit  $\Delta t_{obj}$ . Eine Ausdehnung der Konfliktzone in  $Z$ -Richtung kann ebenfalls berücksichtigt werden. Die Eigengeschwindigkeit kann durch spezifische Sensorik gemessen werden, wie z. B. durch ein klassisches Fahrradtachometer. Weiterhin kann die Eigengeschwindigkeit auch aus der Stereokamera mittels Verfahren der visuellen Odometrie bestimmt werden. Die PET ergibt sich aus der Differenz der beiden errechneten Zeitspannen nach Gleichung 3.32.

Zur eigentlichen Detektion der Events müssen für die Konfliktindikatoren Schwellenwerte definiert werden, welche bei Unter- bzw. Überschreitung eine Szene als Event markiert. Diese Schwellenwerte sind von der eigentlichen Forschungsfrage abhängig. Weiterhin können sie von weiteren Faktoren abhängig sein, wie z. B. Objektklasse, der Verkehrsdichte oder weiteren Umweltbedingungen, weswegen sie an dieser Stelle nicht definiert werden.

## 3.8 Zusammenfassung des Kapitels

Die Entwicklung eines Programms zur automatisierten Auswertung von NCS-Daten umfasst eine Vielzahl an Schritten, welche in diesem Kapitel beschrieben sind. Nach der Vorstellung bisheriger Arbeiten und der Definition der Anforderungen in Abschnitt 3.1 und 3.2, wurde ein generalisierter Workflow zur Bewältigung der Aufgaben entwickelt und in Abbildung 3.2 dargestellt. Der als *Preprocessing* deklarierte Teil beinhaltet die Erstellung der Disparitätenkarte aus Stereobildern, wofür in Kapitel 3.4 der Ansatz des *semi-global matching* als Mittelweg zwischen den globalen und lokalen Methoden gewählt wurde.

Nach dem Stand der Technik kann einer der essenziellen Teile des Workflows, die Objekterkennung bzw. -detektion, durch zwei grundverschiedene Algorithmenklassen vorgenommen werden, welche in Abschnitt 3.5 beschrieben sind. Für den regelbasierten Ansatz wurde die *u-disparity* als zielführend für den spezifischen Anwendungsbereich definiert. Die generische Objektdetektion kann theoretisch alle als annähernd vertikale Ebenen darstellbare Objekte in der Disparitätenkarte detektieren, wobei die Objektklasse nicht relevant ist. Als Gegenpol kann eine modellbasierte Methode nur gelernte Klassen erkennen, gibt somit jedoch weitere Informationen in die Prozesskette ein, welche im weiteren Verlauf des Workflows genutzt werden können. Die Objektinstanzsegmentierung mit *Mask R-CNN* liefert pixelgenaue Objektkonturen, wodurch die Objekttiefe ohne Mehrdeutigkeiten aus der Disparitätenkarte extrahiert werden kann.

Beide Ansätze erzeugen für jedes detektierte Objekt ein Punktmodell, welches durch die Objektverfolgung verarbeitet wird. Dieser Teil der Prozesskette, beschrieben in Abschnitt 3.6, stellt die zeitliche Kopplung der Einzelmessungen dar und erzeugt einen *Track* für jedes detektierte Objekt. Durch das generalisierte Punktmodell kann die Objektverfolgung für beide Varianten der Objektdetektion bzw. -erkennung angewendet werden. Die *Tracks* enthalten den zeitlichen Verlauf der Position des Objekts sowie dessen Geschwindigkeit in Sensorkoordinaten. Im finalen Schritt des Workflows (Kapitel 3.7) werden die Konfliktindikatoren TTC und PET anhand der *Tracks* ermittelt.

Der beschriebene Workflow und seine Komponenten werden im folgenden Kapitel 4 analysiert und bewertet. Hierfür werden verschiedene Testszenarien entwickelt, und die Ergebnisse sowie verschiedenen Einflüsse auf ebendiese dargestellt.

---

# Kapitel 4

## Funktionsanalyse und Bewertung

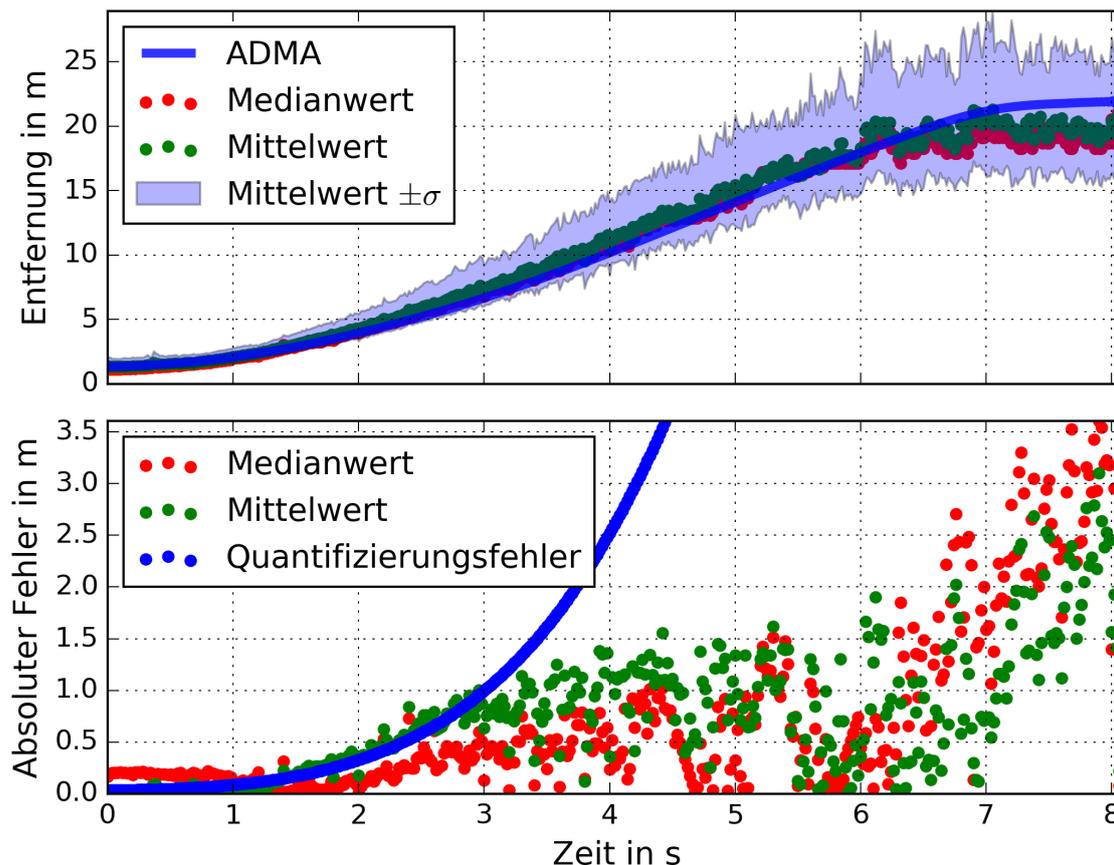
Das Programm zur Erkennung kritischer Situationen, welches in Kapitel 3 beschrieben ist, wird nun auf seine Funktion geprüft und bewertet. Einerseits werden hierbei einzelne Elemente der Prozesskette betrachtet, andererseits wird das Gesamtsystem geprüft. Die beiden umgesetzten Varianten, welche sich durch die Objekterkennung bzw. -detektion unterscheiden, werden dabei gegenübergestellt.

In Abschnitt 4.1 wird die Genauigkeit der Positionsbestimmung bei einem bekannten Objekt bestimmt. Der Fokus richtet sich somit auf die Performanz der 3D-Rekonstruktion in Kombination mit der in Abbildung 2.7 dargestellten Prozesskette. Kapitel 4.2 bewertet die beiden Ansätze der Objektdetektion bzw. -erkennung im realen Straßenverkehr. Die Funktionalität des Gesamtsystems wird in Abschnitt 4.3 analysiert. Anhand von Aufzeichnungen aus dem realen Straßenverkehr kann dies nur im qualitativen Sinn geschehen, da keine Referenzdaten zum Vergleich zur Verfügung stehen. Um dennoch quantitative Aussagen über das System treffen zu können, werden konstruierte Szenarien in einem kontrollierten Umfeld definiert und aufgezeichnet. Die damit verfügbaren Referenzdaten lassen eine direkte Auswertung des Systemverhaltens und somit einen Vergleich der beiden Varianten zu. Zusätzlich werden Beispielszenen aus dem Realverkehr qualitativ analysiert, um einen Einblick in das Systemverhalten in realen Szenarien zu ermöglichen.

### 4.1 Genauigkeit der Positionsbestimmung bei bekannten Objekten

Die gesamte Prozesskette der Vorverarbeitung der Daten, wie sie in Kapitel 2.6 beschrieben ist, beinhaltet die Entzerrung der Kamerabilder sowie deren Rektifikation, wofür die jeweiligen Parameter aus der Kamerakalibration erlangt werden müssen. Auf Basis der vorverarbeiteten Bilddaten kann die Korrespondenzanalyse und schließlich die 3D-Rekonstruktion stattfinden. Um die Performanz dieser Prozesskette zu prüfen, wird ein bekanntes Ziel mit Hilfe des Stereokamerasystems vermessen.

Hierfür wird auf einer geraden Strecke ein Pkw aus dem Stand beschleunigt und wieder bis zum Stillstand abgebremst. Das statisch positionierte Kamerasystem filmt die Szene von hinten, der Pkw fährt also vom Kamerasystem weg. Als Vergleich zu den Stereomessungen wird ein Trägheitsnavigationssystem (INS) des Typs „GeneSys ADMA“ eingesetzt. Dieses System berechnet u.a. die Position des Fahrzeugs



**Abbildung 4.1**

Positionsbestimmung bei bekannten Objekten.

Oben: Entfernung des Fahrzeugs über die Zeit. Dargestellt sind die Grundwahrheit des ADMA-Systems (blau), sowie die stereobasierte Entfernung auf Basis des Medianwerts (rot) und des Mittelwerts (grün) aller validen Disparitäten. Weiterhin ist der Entfernungsbereich dargestellt, welcher der Standardabweichung der validen Disparitäten entspricht (hellblau).

Unten: Die absoluten Fehler der stereobasierten Entfernungen (rot - Median; grün - Mittelwert) werden dem Quantifizierungsfehler nach [43] (blau) gegenübergestellt.

bezogen auf ein ortsfestes Koordinatensystem und kann aufgrund der hohen Messgenauigkeit als Grundwahrheit dienen. Start- und Endposition des Fahrzeughecks im Kamerakoordinatensystem wurden mittels Laserentfernungsmesser bestimmt.

Im Vorfeld des Versuchs wurde das Kamerasystem entsprechend den Angaben in Kapitel 2.4 kalibriert. Für den Versuch wurde das Kamerasystem von Tabelle 2.2 abweichend konfiguriert. Während Brennweite und Auflösung beibehalten wurden, beträgt die Basislänge 0,034 m, ist also um den Faktor 3,3 kleiner als das vorgestellte Standardsystem. Durch diese Änderung lässt sich der entstehende Tiefenfehler bei kleiner Basislänge, siehe Kapitel 2.5, gut darstellen.

Die Detektion des Fahrzeugs wird mit dem in Kapitel 3.5.3.3 vorgestellten modellbasierten Verfahren umgesetzt. Die richtige Maskierung des Fahrzeugs auf der Bildebene wird dabei manuell überprüft, um etwaige systematische Fehler durch die Objektdetektion ausschließen zu können. Für jedes Bild der Messung werden der Medianwert sowie der Mittelwert aller auf die Maske fallenden Disparitätenwerte berechnet und anhand dessen die Entfernung zur Kamera bestimmt. Abbildung 4.1

zeigt das Ergebnis eines Versuchs. In der oberen Darstellung stellt die blaue Linie die Grundwahrheit der Entfernung dar, gemessen durch das INS. Die stereobasierten Entfernungen, basierend auf dem Mittelwert bzw. des Medians, werden in grün und rot dargestellt. Weiterhin wird der Bereich der Entfernung abgebildet, welcher der Standardabweichung der Disparität einer jeden Messung entspricht. Die untere Darstellung zeigt den absoluten Fehler der beiden stereobasierten Methoden im Vergleich zur Grundwahrheit. Weiterhin wird der Quantifizierungsfehler nach [43] dargestellt, welcher den Entfernungsfehler bei einer um 1 px fehlerhaften Disparität beschreibt.

Wie erwartet steigt der Fehler mit der Entfernung. Im Vergleich zum dargestellten Quantifizierungsfehler ist der Fehler der Messungen geringer, was auf eine Abweichung von weniger als 1 px von der berechneten Disparität zum Erwartungswert schließen lässt. Einzig die auf dem Median basierte Entfernung weist bei niedrigen Entfernungen unterhalb von 1,5 m, im Zeitbereich von 0 s bis 1 s, eine größere Abweichung des Disparitätenfehlers von mehreren Pixeln auf. Durch den nichtlinearen Zusammenhang zwischen Disparität und Entfernung ist der resultierende Fehler mit einem Wert kleiner 25 cm gering. Der RMSE (*Root Mean Square Error*, Wurzel der mittleren Fehlerquadratsumme) beträgt 0,65 px mit einer Standardabweichung des absoluten Fehlers von 0,42 px.

Der Fehler des Medians in der ersten Sekunde der Messung ist auf die Abbildung des Pkw in den Bildern der Stereokamera zurückzuführen. Zu Beginn der Messung ist das Fahrzeug nah am Kamerasystem, so dass es nur von den Heckleuchten aufwärts abgebildet wird. Der Rest des Hecks ist außerhalb des Bildbereichs. Durch das Stufenheck des Versuchsfahrzeugs ist ein Teil des detektierten Fahrzeugs weiter von der Kamera entfernt als das Heck, dessen Entfernung durch die Grundwahrheit des INS Systems angegeben wird. Da für die schräge Heckscheibe nur wenige valide Disparitäten berechnet werden, verzerren diese das Ergebnis nur gering. Der Medianwert gibt wie erwartet eine geringere Entfernung aus, welche jedoch stärker von der Referenz abweicht. Scheinbar liegt an dieser Stelle ein systematischer Fehler vor, welcher mit 0,2 m jedoch nicht kritisch ist. Mit zunehmender Entfernung des Pkw wird ein größerer Teil des Hecks sichtbar und der Fehler nimmt ab. Wie beschrieben ist der Fehler durch den nichtlinearen Zusammenhang von Disparität und Entfernung von geringem Ausmaß bezogen auf die Weltkoordinaten. Im Entfernungsbereich ab 17 m steigt der Fehler für beide Berechnungsvarianten stark an. Der Quantifizierungsfehler liegt in diesem Bereich bei Werten größer 7 m, somit reicht eine kleine Abweichung der Disparität, um Fehler in der gezeigten Größenordnung zu generieren.

Prinzipiell ist der Fehler über den gezeigten Messbereich, trotz der reduzierten Basislänge, als gut einzuschätzen. Ein Disparitätenfehler unter 1 px spricht hierbei für eine erfolgreiche Kalibration und *Preprocessing*. Der Median ist das Mittel der Wahl zur Berechnung der Disparität des Punktmodells eines Objekts. Abgesehen von den erklärten Effekten bei niedrigen Entfernungen sowie den schlechteren Werten im Bereich hoher Quantifizierungsfehler, werden durch den Median geringere Fehler als bei der Verwendung des Mittelwerts erzielt. Da sich der Quantifizierungsfehler mit steigender Basislänge reduziert und für die weiteren Versuche die bereits definierte größere Basislänge verwendet wird, stellt der Median insgesamt die bessere Wahl dar.

### 4.2 Erkennung von Objekten im realen Verkehr

Als Grundvoraussetzung für die Detektion eines Konflikts muss der\*die Konfliktpartner\*in wahrgenommen werden. Erst die Detektion eines Objekts ermöglicht dessen Verfolgung und somit die Berechnung von Konfliktindikatoren. Die beiden Ansätze zur Objekterkennung bzw. -detektion verwenden nach Kapitel 3.5 unterschiedliche Eingangsdaten, ihre Funktionsweise ist kaum vergleichbar und auch die Ausgangsdaten unterscheiden sich. Folglich kann auch die Detektions- bzw. Erkennungsrate der Ansätze nicht verglichen werden. Dennoch sollen die Ansätze für sich bewertet werden.

Für die regelbasierten Objekterkennung wird in Kapitel 4.2.1 ein Konzept zur Evaluation anhand von Referenzdaten vorgestellt, welches die generische Art der Objektdetektion berücksichtigt. Der Fokus liegt dabei auf Genauigkeit und Sensitivität des Ansatzes in Abhängigkeit der verschiedenen Parameter. Dabei wird auch geklärt, ob FNs und FPs von ihrer Lage im Bildraum, ihrer tatsächlichen Entfernung oder anderen Parametern abhängen.

Für den modellbasierten Ansatz existieren in der Literatur bereits vielfältige Analysen bezüglich der Leistungsfähigkeit für verschiedene Anwendungsszenarien. Kapitel 4.2.2 gibt einen Überblick über die verwendeten Metriken zur Evaluation und listet verschiedenen Quellen und Datensätze auf, für die bereits Analysen existieren. Weiterhin wird ein kurzer Vergleich zwischen dem regelbasierten und dem modellbasierten Ansatz bezüglich der zu erwartenden Güte der Objektentfernung, -breite und -höhe gezogen.

#### 4.2.1 Regelbasierter Ansatz

Um die Detektionsrate des regelbasierten Ansatzes bestimmen zu können, ist ein Referenzmodell des Umfelds im Sichtfeld des Stereokamerasystems nötig. Die daraus extrahierten Objekte müssen denselben Anforderungen entsprechen wie der implementierte Ansatz, welcher in Kapitel 3.5.2 beschrieben wurde. Die Objekte müssen somit generischer Natur sein, was bedeutet, dass die Objekte nicht auf eine endliche Anzahl an Klassen limitiert sein dürfen.

#### **Evaluation mit korrespondierenden LiDAR-Daten**

Da eine solche Abbildung der Welt abseits von Simulationen schwer zu erlangen ist, wird anhand des „KITTI“-Datensatzes [34] eine alternative Form der Referenz geschaffen. Neben Kameradaten bietet der Datensatz eine hochauflösende LiDAR-Punktwolke passend zu jedem Kamerabild sowie alle benötigten Kalibrationsdaten, um die Punktwolke in den Bildraum der Kameras zu transformieren. Aufgrund des Messprinzips des LiDAR-Sensors entspricht die in den Bildraum transformierte Punktwolke einer dünn besetzten Tiefenkarte mit hoher Genauigkeit. Die Tiefenkarte kann anhand der Kalibrationsparameter des Kamerapaares als Disparitätenkarte dargestellt werden, wodurch die Berechnung der *u-disparity* ermöglicht wird. Durch morphologisches Schließen kann die geringere Auflösung der Punktwolke im Vergleich zur Kamera ausgeglichen werden.

Die als *u-disparity* dargestellte LiDAR-Punktwolke kann als Referenz für die korrespondierende Darstellung durch die Stereokamera verwendet werden. Eine Extraktion von Objekten kann für beide Varianten auf die gleiche Weise geschehen, was



**Abbildung 4.2**

Berechnung der TP und FP des regelbasierten Ansatzes unter Verwendung der LiDAR-Punktwolke des „KITTI“-Datensatzes [34]. Können stereobasierte Konturen in der LiDAR *u-disparity* wiedergefunden werden, so gelten sie als TP (grün markiert), andernfalls als FP (rot markiert). Die maximal berücksichtigte Entfernung für Objekte beträgt hierbei 15 m.

der Vergleichbarkeit dient. Dabei darf jedoch nicht vergessen werden, dass für beide Varianten, die in Kapitel 3.5.2 getroffenen Annahmen für detektierbare Objekte gelten.

Das Ziel des Vergleichs ist die Genauigkeit und Sensitivität der Objektdetektion unter den genannten Einschränkungen und für verschiedene Parametrierungen zu untersuchen. Hierfür müssen die Werte für TP, FP und FN bestimmt werden. Zur Ermittlung der *true* und *false positives* wird für jede Kontur in der stereobasierten *u-disparity* ein Pendant aus der LiDAR-basierten *u-disparity* gesucht, welches eine ähnliche Disparität und eine horizontale Überlappung mit der stereobasierten Kontur aufweist. Dies ist beispielhaft in Abbildung 4.2 dargestellt. Die Ähnlichkeit der Disparität beschränkt sich hierbei auf den Bereich  $\delta \pm 1$  px, um Ungenauigkeiten in der Kalibration oder der Synchronisation der Sensoren zu berücksichtigen. Zu beachten ist weiterhin, dass die Konturdarstellungen nach einer minimalen Objektbreite in Pixel vorgefiltert werden, während die Darstellungen der *u-disparity* keinem Filter unterliegen. Da die LiDAR-Punktwolke keine zur Disparitätenkarte vergleichbaren Regionen falscher Tiefe aufweist, stellen detektierte Elemente beliebiger Breite definitiv ein reales Objekt dar. Ein gefundenes Pendant entspricht dabei einer richtig positiven Detektion, ein fehlendes Pendant einer falsch positiven Detektion. Falsch negative, also ausbleibende Detektionen, können durch das umgekehrte Verfahren detektiert werden, wie es in Abbildung 4.3 dargestellt ist. Kann für eine Kontur aus der LiDAR-basierten *u-disparity* kein korrespondierendes Element in den stereobasierten Daten gefunden werden, entspricht dies einem FN. Einzig die Bestimmung der richtig negativen Detektionen kann auf diese Weise nicht evaluiert werden, was aber hingenommen werden kann.



**Abbildung 4.3**

Berechnung der FN des regelbasierten Ansatzes unter Verwendung der LiDAR-Punktwolke des „KITTI“-Datensatzes [34]. In den LiDAR-Konturen kann ein Objekt erkannt werden, welches sich nicht in der stereobasierten *u-disparity* wiederfinden lässt (orange markiert). Die maximal berücksichtigte Entfernung für Objekte beträgt hierbei 15 m.

Für die Evaluation gelten verschiedene Randbedingungen. Die berücksichtigte Entfernung von Objekten wird auf mindestens 4 m beschränkt, da die Sensorkonfiguration im Nahfeld einen großen Disparitätenbereich abdeckt, was zu erheblichen Rechenaufwand führt. Weiterhin sind die Kameras über 2,5 m von der Fahrzeugfront entfernt montiert. Die maximale Entfernung wird auf 14 m begrenzt, da einerseits nur eine Auswertung für nahe Objekte relevant für den Anwendungsfall ist, andererseits die Anzahl der verwertbaren LiDAR-Punkte pro Flächeneinheit mit der Entfernung sinkt. Am linken und rechten Rand wird eine Randzone definiert. Detektierte Objekte, gleich ob aus LiDAR- oder Stereodaten, werden nur für die Evaluation berücksichtigt, wenn das Zentrum ihrer ROI nicht in der Randzone liegt. Somit werden negative Effekte am Rande der Disparitätenkarte negiert.

Die Ergebnisse der Objektdetektion sind, abgesehen von der Kalibration des Sensorsystems und der Vorverarbeitung der Daten, von zwei Parametern abhängig. Diese sind der Schwellenwert zur Bildung der *u-disparity* sowie die Mindestbreite der extrahierten Konturen aus ebendieser. Der erste Parameter beschreibt, über wie viele Zeilen sich ein Objekt in der Disparitätenkarte erstrecken muss, um als solches detektiert zu werden (siehe Kapitel 3.5.2.4). Der zweite Parameter beschreibt die Mindestbreite des Objekts in Bildkoordinaten, wobei die tatsächliche Breite in Weltkoordinaten von der jeweiligen Disparität abhängig ist. Die Wahl der beiden Parameter sind vom jeweiligen Sensorset, dessen Kalibration und der Vorverarbeitung abhängig, wodurch die verwendeten „KITTI“-Daten nicht zur Identifikation für einen eigenen Datensatz dienen können. Die resultierenden Effekte bei einer Parametervariation sind jedoch übertragbar. Bei der Anwendung der Objektdetektion auf die LiDAR-Punktwolke werden Anpassungen für den *u-disparity* Schwellenwert notwendig, da die abgeleitete Disparitätenkarte dünn besetzt ist, jedoch keine fehlerhaften Bereiche wie in der stereobasierten Variante zu erwarten sind. In der Theorie stellen

zwei übereinanderliegende Punkte gleicher Höhe bereits eine vertikale Fläche in der Punktwolke, und somit ein Objekt dar. Um etwaige Toleranzen auszugleichen kann der Schwellwert auf wenige Pixel, jedoch größer zwei Pixel gesetzt werden. Die Mindestbreite der Objekte sollte sich zwischen stereo- und LiDAR-basierter Methode nicht unterscheiden, um die Vergleichbarkeit der Methoden sicherzustellen.

### **Grenzen der Evaluationsmethodik**

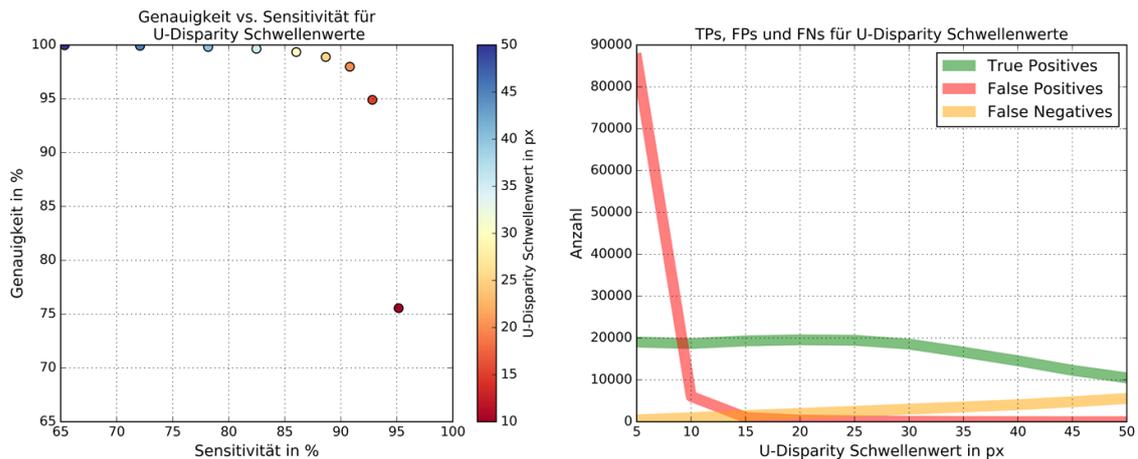
Die aufgezeigte Methodik zur Funktionsanalyse der regelbasierten Objektdetektion weist einige Limitationen auf. Der verwendete Datensatz ist nicht mit dem in dieser Arbeit definierten Sensor, sondern einem anderen Stereokamerasystem aufgezeichnet. Während die grundsätzliche Messmethode gleich bleibt, ändern sich die Sensorparameter. Auch die Kalibration ist nicht selbst durchgeführt, sondern direkt aus dem Datensatz übernommen. Die berechneten Metriken beziehen sich somit nicht auf die vorgestellte Prozesskette von Rohdaten bis hin zur Objektdetektion, sondern nur auf den Algorithmus zur Berechnung der Disparitätenkarte und der Objektdetektion selbst. Weiterhin ist das System nicht auf einem Fahrrad montiert, wodurch die besondere Dynamik von Fahrrädern, wie in Kapitel 3.5.1 beschrieben, in den verwendeten Daten nicht abgebildet ist. Die extrahierten Objekte aus der Punktwolke des LiDAR sind außerdem nur bedingt als Grundwahrheit zu werten, da die Anwendung der *u-disparity* die darstellbaren Objekte entsprechend eingrenzt. Zusätzlich kann durch die unterschiedlichen Verbaupositionen von Kamerasystem und LiDAR ein Objekt im Auswertebereich nur für eines der beiden Sensorsysteme verdeckt sein, wodurch eine Detektion nicht für beide Systeme möglich ist.

Aufgrund der zum Teil verrauschten Disparitätenkarten sowie der Ausdehnung von Objekten über einen großen Disparitätenbereich, werden Einzelobjekte oft durch mehrere kleine Objekte unterschiedlicher Tiefe dargestellt. Durch die Auswertelogik können mehrere stereobasierte Objekte einem LiDAR-basierten Objekt zugeordnet werden, wodurch für ein reales Objekt mehrere TPs entstehen. Dies führt zu verzerrten Werten für Genauigkeit und Sensitivität. Relativ gesehen können die Metriken jedoch für Parameterstudien verwendet werden. Die Anzahl der FNs, welche in Kapitel 3.2 als eine kritische Leistungskennzahl des Auswertesystems definiert wurde, unterliegt nicht dem genannten Problem und spiegelt somit das reale Verhalten wider.

### **Einfluss des Schwellenwerts der *u-disparity***

Die Performanz der Objektdetektion ist stark von dem gewählten Schwellenwert der *u-disparity* abhängig. Abbildung 4.4 Links stellt diesen Zusammenhang anhand der Genauigkeit und Sensitivität in Abhängigkeit des Schwellenwerts dar. Dabei wurde für die LiDAR-basierte *u-disparity* ein Schwellenwert von 3 px gewählt, um alle Objekte zu detektieren, welche einer vertikalen Ebene entsprechen. Die minimale Konturbreite entspricht für beide Sensorvarianten 20 px.

Die Genauigkeit erreicht für Schwellenwerte größer 25 px einen Wert über 98 %, was durch die starke Filterung bedingt ist. Bereiche gleicher Disparität, welche sich über wenige Zeilen erstrecken, werden nicht als Objekt detektiert, was für Bereiche fehlerhafter Korrespondenz durchaus erwünscht ist. Daraus resultiert die sinkende Anzahl an FPs mit steigendem Schwellenwert, siehe Abbildung 4.4 rechts. Durch große Schwellenwerte werden jedoch auch Objekte geringer Höhe aussortiert, was sich durch die sinkende Gesamtzahl der TPs zeigt. Da ein fester Schwellenwert einer



**Abbildung 4.4**

Links: Genauigkeit und Sensitivität in Abhängigkeit des *u-disparity* Schwelldwerts. Je größer der Schwelldwert gewählt wird, desto besser ist die Genauigkeit der Objektdetektion, bei abnehmender Sensitivität.

Rechts: Detektierte *True Positives*, *False Positives* und *False Negatives* in Abhängigkeit des *u-disparity* Schwelldwerts. FPs und TPs sinken mit steigendem Schwelldwert, während FNs steigen.

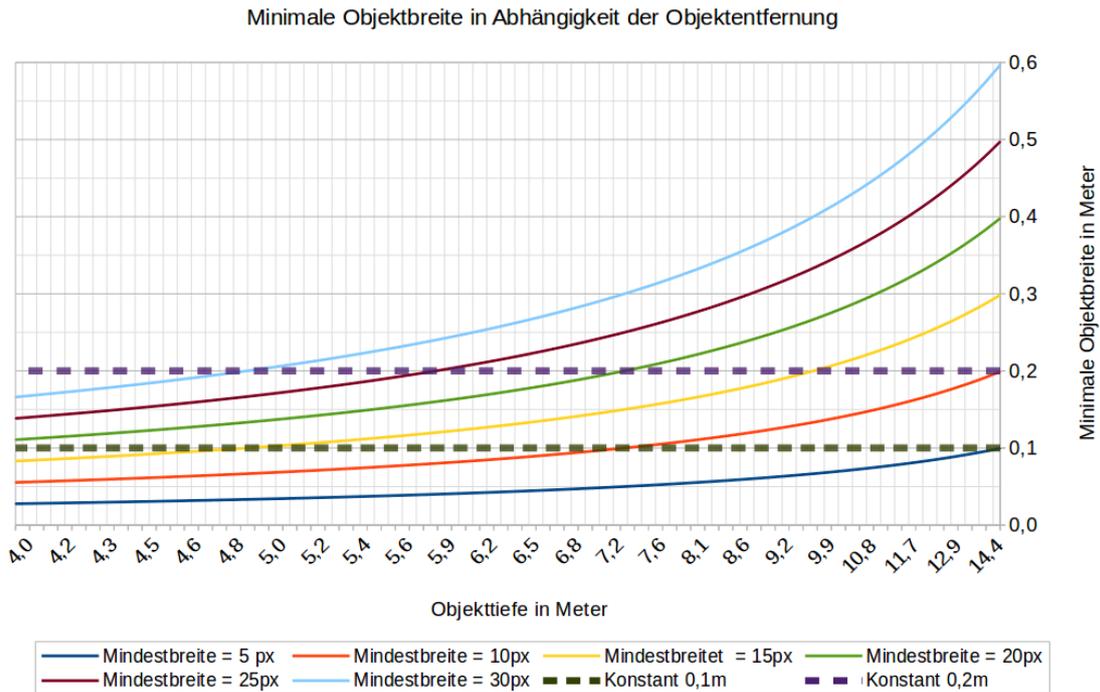
von der Disparität abhängigen Mindesthöhe entspricht, muss dieser Sachverhalt bei der Definition der Sensorparameter berücksichtigt werden, um die Anforderungen an die Mindesthöhe der zu detektierenden Objekte nicht zu verletzen.

Die sinkende Anzahl an FNs und dadurch steigende Sensitivität mit abnehmendem Schwelldwert ist durch die geringe Minimalhöhe der Objekte bedingt. Aufgrund des geringen Schwelldwerts für die LiDAR-basierten Objekte, werden solche mit geringen Objekthöhen detektiert. Werden niedrige Schwelldwerte auf die stereobasierte Objektdetektion angewendet, sinkt ebenfalls die Mindesthöhe und es wird ein größerer Anteil der LiDAR-basierten Objekte auch durch die Stereokamera detektiert. Eine Reduktion der FNs ist die Folge, jedoch zum Preis der sinkenden Genauigkeit.

Der Zielkonflikt kann anhand der Anforderungen an die Auswertung, welche in Kapitel 3 definiert wurden, und der projektspezifischen Mindesthöhe von Objekten aufgelöst werden. Weist der kleinstmögliche Schwelldwert, welcher die Anforderungen an die Objekthöhe erfüllt, eine zu schlechte Genauigkeit auf, um eine automatisierte Auswertung durchzuführen, muss das Sensorset bzw. müssen die Sensorparameter angepasst werden. Im vorliegenden Fall erfüllt ein Schwelldwert von 15 px die Anforderungen an die Mindesthöhe, wobei die Genauigkeit einen Wert von 95 % und die Sensitivität einen Wert von 93 % erreicht.

### Einfluss der Mindestbreite der extrahierten Kontur

Die aus der *u-disparity* extrahierten Konturen können nach ihrer Breite gefiltert werden. Um fehlerhafte Regionen in der Disparitätenkarte nicht fälschlicherweise als Objekt zu deklarieren, kann zusätzlich zum Schwelldwert der *u-disparity* eine Mindestbreite der Kontur sinnvoll sein, um die Anzahl der FPs zu reduzieren. Objekte, deren Darstellung in der *u-disparity* kleiner als dieser Schwelldwert sind, werden ebenfalls gefiltert, was zu einem FN führt. Ein fester Schwelldwert im Bildraum

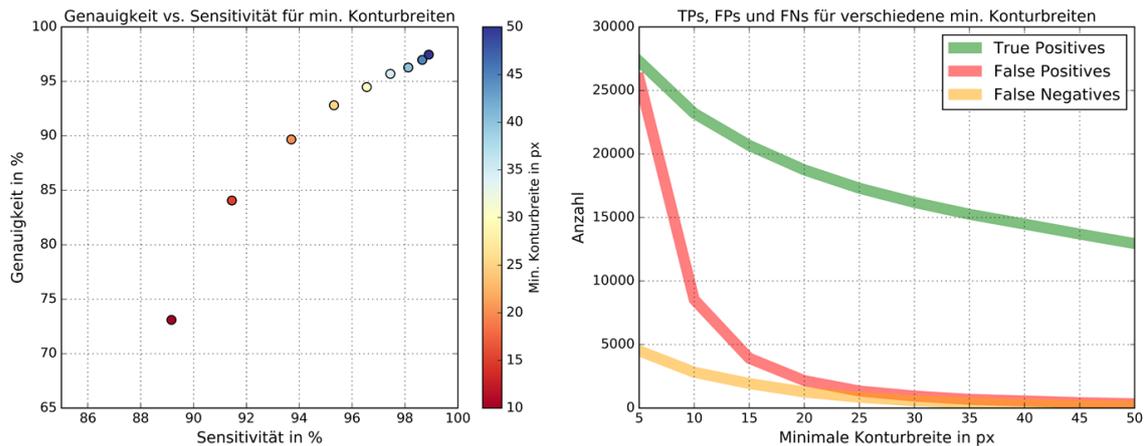
**Abbildung 4.5**

Ein fester Schwellenwert für die minimale Konturbreite im Bildraum führt zu einer nicht-linearen, tiefenabhängigen minimalen Objektbreite in Weltkoordinaten. Die Projektion ist dabei von den Sensorparametern abhängig. Dargestellt ist der Sachverhalt für die prozessierten „KITTF“-Daten [34] und einem Schwellenwert der  $u$ -disparity von 20 px.

entspricht einer variablen Breite in Weltkoordinaten (Abbildung 4.5), wobei die Projektion wie schon bei der Objekthöhe von den Sensorparametern abhängig ist. Die zu detektierende Mindestbreite, welche im jeweiligen Studiendesign festgelegt ist, muss somit bei der Auswahl der Sensorparameter und des Schwellenwerts berücksichtigt werden.

Durch die Anwendung einer Mindestbreite auf die extrahierten Konturen der stereo- und LiDAR-basierten  $u$ -disparity verändert sich, wie bereits bei der Variation des Schwellenwerts für die  $u$ -disparity, die Anzahl der zu detektierenden Objekte einer spezifischen Szene. Der Schwellenwert für die Objektbreite ist hierbei für beide Sensortechnologien gleich. Für die stereobasierte  $u$ -disparity wurde außerdem der bereits ermittelte Wert von 15 px verwendet, während die LiDAR-basierte Variante mit einem Schwellenwert von 3 px gefiltert wurde.

Abbildung 4.6 Rechts stellt die FPs, FNs und TPs der analysierten Daten für verschiedene Mindestbreiten dar. Eine geringe Mindestbreite von bis zu 15 px führt zu einer überproportionalen Anzahl an FPs. Wie zuvor, bei der Variation des Schwellenwerts der  $u$ -disparity, nimmt die Genauigkeit mit steigendem Parameterwert zu. Die Begründung bleibt dabei ebenfalls gleich, da eine größere Mindestbreite kleine fehlerhaften Regionen der Disparitätenkarte nicht als Objekt zulässt. Die absolute Menge an TPs nimmt ebenfalls wie erwartet ab. Die hohe Anzahl der FNs bei geringen Mindestbreiten resultieren aus niedrigen Objekten, welche aus der LiDAR-Punktwolke als Objekt extrahiert werden. Diese werden ebenfalls durch Erhöhen des Schwellenwerts gefiltert, wodurch die Sensitivität steigt.



**Abbildung 4.6**

Links: Genauigkeit und Sensitivität in Abhängigkeit der min. Konturbreite. Je größer die Mindestbreite gewählt wird, desto besser ist die Performanz der Objektdetektion.

Rechts: Detektierte *True Positives*, *False Positives* und *False Negatives* in Abhängigkeit der minimalen Konturbreite. Die Gesamtzahl der drei Metriken sinken mit steigender minimaler Konturbreite.

Wie bereits beim Schwellenwert der *u-disparity* beschrieben, ist bei der Parameterwahl für die Mindestbreite einer Kontur ebenfalls ein Zielkonflikt zu lösen. Im Gegensatz zur Objekthöhe, welche als überfahrbar definiert und somit das Objekt als irrelevant klassifiziert werden kann, ist keine ungefährliche Objektbreite bestimmbar. Die Wahl eines Schwellenwerts dient somit rein der Robustheit des Detektionsalgorithmus. Dennoch muss im Studiendesign und eventuell anhand von Vorstudien definiert werden, welche Objektbreite als irrelevant klassifiziert werden kann und welche Einflüsse dies auf das Studienergebnis hat. Insbesondere, da schmale Objekte wie Begrenzungspfosten oder Laternen durchaus eine Gefahr für Radfahrer\*innen sein können und allein die Existenz solcher das Verhalten potenziell beeinflussen kann. Für das dargestellte Sensorsystem bietet sich beispielsweise eine Mindestbreite von 20 px an, wodurch die Genauigkeit einen Wert von ca. 89,7% und die Sensitivität ca. 93,7% erreicht, während bei 10 m Entfernung eine Objektbreite von 27 cm abbildbar ist. Eine Anpassung der Sensorparameter kann als weiterer Schritt dienen, um die Randbedingungen dem Studiendesign anzupassen.

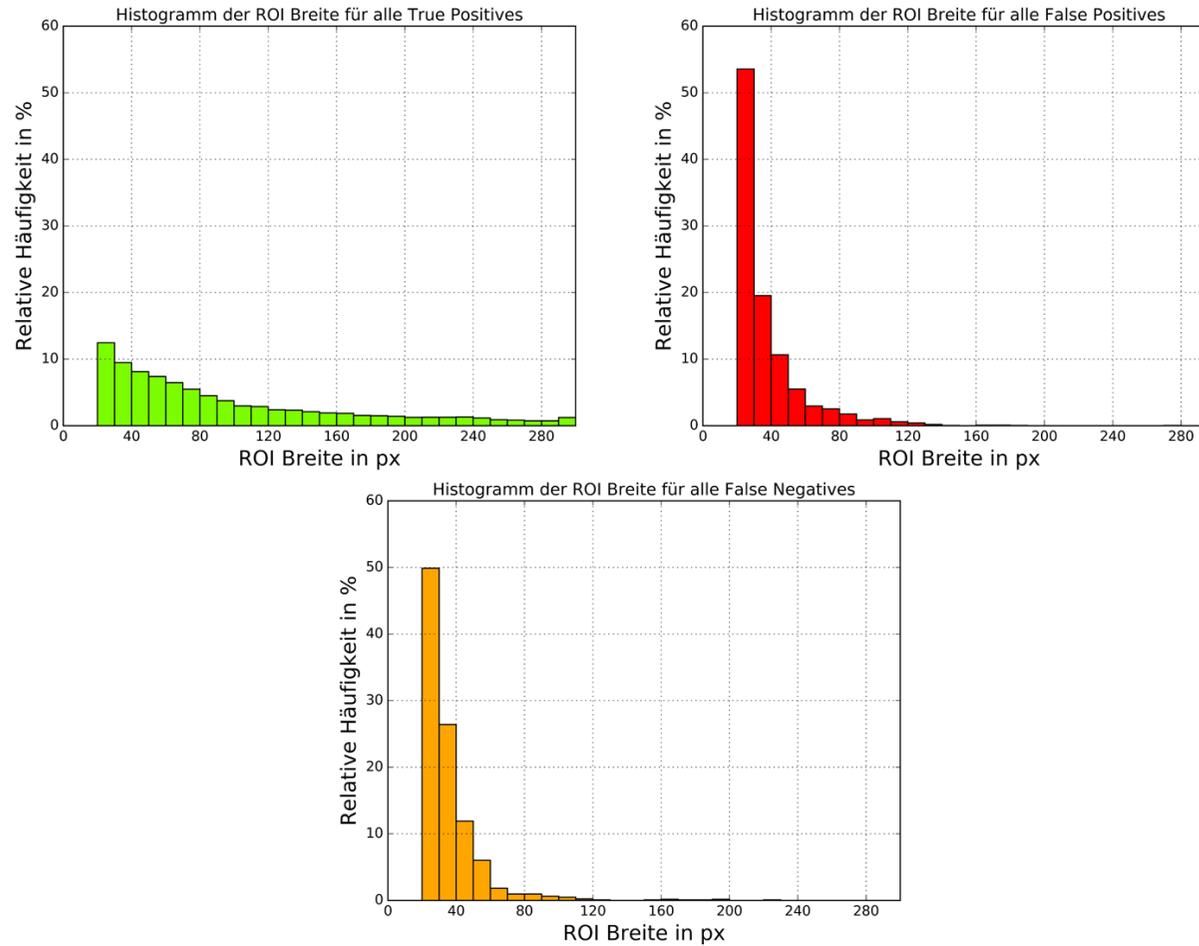
### Verteilung der FNs und FPs mit den gewählten Parametern

In der Folge werden die gewählten Parameter angewendet. In Abbildung 4.7 zeigt sich, dass schmale Objekte in den TPs häufiger vertreten sind als breite, was unter anderem an der Aufspaltung von Objekten über große Disparitätenbereiche liegt. FPs und FNs verhalten sich ähnlich, weisen allerdings ab 120 px Breite keine Einträge mehr auf. Dies lässt die Folgerung zu, dass breite Objekte erkannt werden, wenn auch durch viele kleinere Objekte dargestellt. Das große FPs nicht detektiert werden zeigt, dass keine großen fehlerhaften Bereiche in der Disparitätenkarte existieren. Über 70% der FPs sind schmaler als 40 px, wobei die Objekte unter 20 px durch die Mindestbreite herausgefiltert wurden. Bei den FNs gilt dies für über 75% Anteil an den kleinen Objekten bis 40 px Breite. Dem gegenüber stehen 20% der TPs in diesem Bereich. FPs und FNs sind für schmale Objekte also überproportional vertreten.

Die *Heatmaps* in Abbildung 4.8 zeigen die Anzahl der TPs, FPs und FNs in einem 2D Histogramm, wobei die Bildbreite über die Entfernung dargestellt ist. Bei der TP *Heatmap* erkennt man einen vertikalen, breiten Bereich, welcher die Straße repräsentiert. Dieser Bereich ist aufgrund des Rechtsverkehrs leicht von der Bildmitte verschoben. In den Bereichen abseits der zentralen Zone werden weitaus mehr Objekte wahrgenommen. Dies ist neben der Anzahl an Fußgängern auch auf die Randbebauung und andere statische Objekte zurückzuführen. Die 80 px breite Randzone links und rechts des Bilds enthält keine Objekte, da diese bei der Auswertung unterdrückt wurden, um keine Randeffekte der Disparitätenkarte in die Analyse einfließen zu lassen. Dies reduziert zwar das Sichtfeld des Kamerasystems, erlaubt jedoch eine Bewertung der Objekterkennung ohne Effekte anderer Algorithmen berücksichtigen zu müssen, welche die Ergebnisse verzerren. Dieses Vorgehen ist auch für die Auswertung von NCS ratsam.

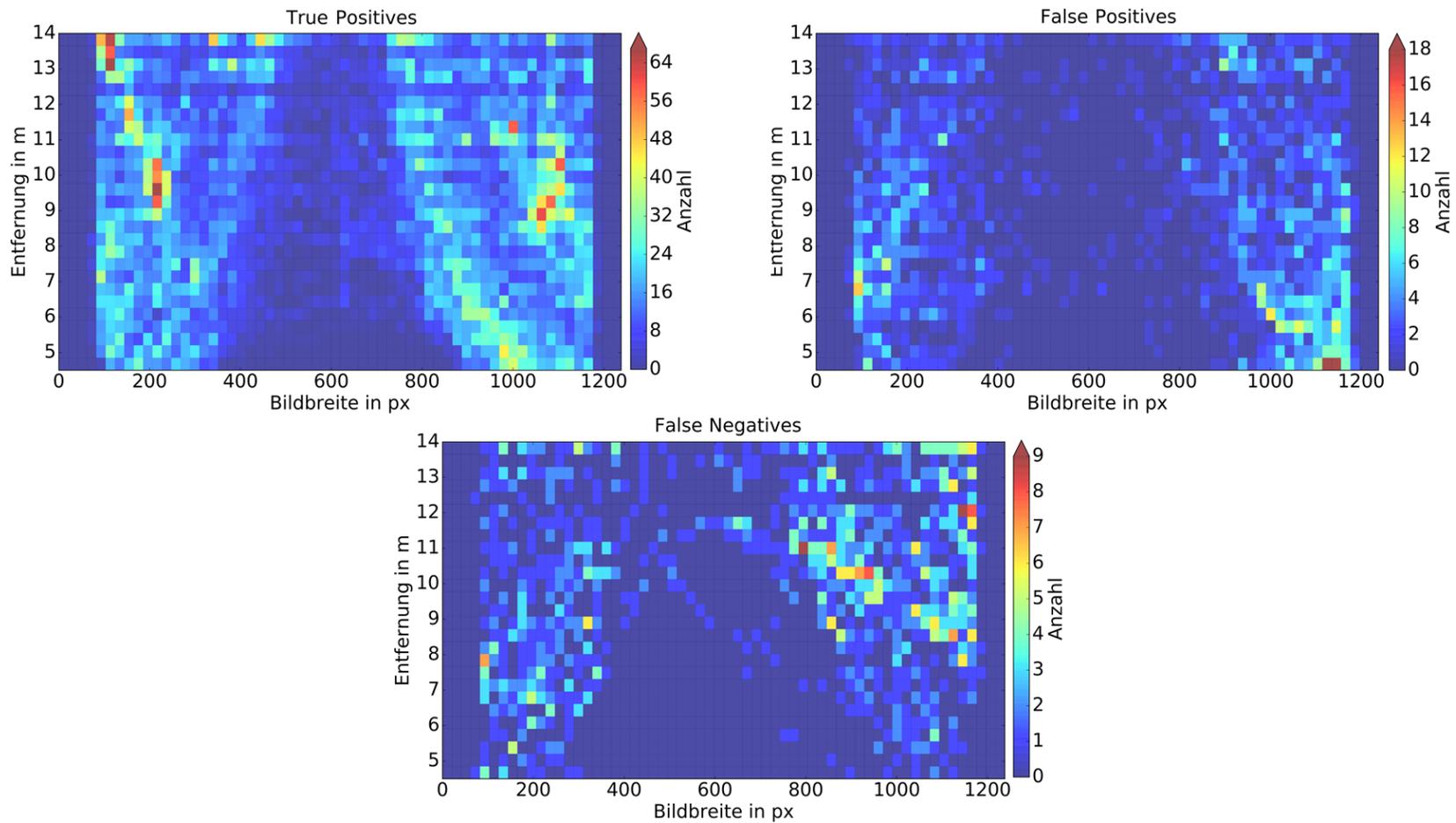
Die FPs treten vermehrt in Randbereichen niedriger Entfernung auf. Eine fehlende Entzerrung der LiDAR Punktwolke nach der Eigenbewegung des Fahrzeugs kann als Ursache ausgeschlossen werden. Aufgrund des beweglichen LiDAR Scanners, dessen Ausrichtung in Längsrichtung des Fahrzeugs mit den Kameras synchronisiert wurde, weisen die Punkte abseits der Längsachse einen Fehler auf, welcher durch die Eigenbewegung und die limitierte Rotationsgeschwindigkeit des LiDARs bedingt ist. Wären die FPs ein Resultat der fehlenden Entzerrung, müssten auch erhöhte FNs in den angrenzenden Bereichen auftreten, da das durch die Kameras detektierte Objekt entsprechend seiner Bewegung und der Eigenbewegung des Sensorträgers an einer anderen Stelle im Sichtfeld dargestellt sein muss als das zeitlich versetzt aufgezeichnete Objekt der LiDAR Punktwolke. Durch den zeitlichen Versatz können Objekte für beide Sensortechnologien auch neu in den Sichtbereich eintreten oder austreten. Ein signifikantes Auftreten von FNs und FPs im Randbereich ist jedoch nicht auszumachen. Vielmehr sind die FPs im Nahbereich auf Rauschen in der Disparitätenkarte zurückzuführen. Weiterhin können über das gesamte Sichtfeld FPs und FNs durch Verdeckungen einzelner Sensoren erwartet werden. So können beispielsweise durch das LiDAR Objekte detektiert werden, welche aufgrund von Verdeckung nicht auf beiden Kamerabildern abgebildet und somit nicht detektiert werden können. Die FNs konzentrieren sich auf Bereiche hoher TP-Dichte, was die These der Verdeckungen unterstützt, da für die Stereokamera das Objekt in beiden Kameras sichtbar sein muss. Weiterhin ist die Anzahl der FNs auf den geringen Schwellenwert zur LiDAR-basierten Berechnung der *u-disparity* zurückzuführen, wenn davon ausgegangen werden kann, dass in Bereichen hoher Objektdichte auch viele Objekte geringer Höhe auftreten.

Da die FNs, welche als kritischer Leistungsindikator für die Objektdetektion definiert wurden, keinen offensichtlichen systematischen Fehler in ihrer Verteilung über das Sichtfeld aufweisen und zusätzlich durch Anpassung der Schwellenwerte bzw. der Sensorparameter weiter reduziert werden können, erfüllt die regelbasierte Objektdetektion die in Kapitel 3.2 geforderten Eigenschaften. Die geforderte vertretbare Anzahl an FPs kann ebenso beeinflusst werden, allerdings steht die Optimierung der FNs im Vordergrund. Eine Vorstudie zur Definition der Sensorparameter und der entsprechenden Schwellenwerte bietet sich an, um zusätzlich die für das Studiendesign spezifische Anforderung an die FPs zu prüfen.



**Abbildung 4.7**

Verteilung der *True Positives*, *False Positives* und *False Negatives* über die Objektbreite. Hier verwendete Parameter sind für die Mindestbreite 20 px, ein *u-disparity* Schwellenwert von 15 px für stereobasierte und 3 px für LiDAR-basierte Objekte.



**Abbildung 4.8**

Heatmap der *True Positives*, *False Positives* und *False Negatives* über die Bildbreite und Entfernung. Hier verwendete Parameter sind für die Mindestbreite 20 px, ein *u-disparity* Schwellenwert von 15 px für stereobasierte und 3 px für LiDAR-basierte Objekte.

### 4.2.2 Modellbasierter Ansatz

Im Gegensatz zum regelbasierten Ansatz, welcher in Kapitel 4.2.1 evaluiert wird, muss für den modellbasierten Ansatz keine generische Referenz aus anderen Sensordaten geschaffen werden. Da die Objektinstanzsegmentierung auf überwachtem Lernen basiert, ist die Anwendbarkeit auf spezifische Klassen beschränkt. Zur Überprüfung der Performanz eines CNNs kann vor dem Training ein Teil der Daten als Validationsdatensatz definiert werden. Dieser wird nicht zum Trainieren des Netzes verwendet, kann aber als Basis zur Ermittlung der Performanz herangezogen werden. Üblicherweise halten offizielle Wettbewerbe oder Benchmarks, welche meist anhand eines spezifischen Datensatzes durchgeführt werden, diesen Validationsdatensatz unter Verschluss. Zusätzlich zu den Daten werden Evaluationsmetriken definiert, nach welchen die CNNs geprüft werden.

Das in dieser Arbeit verwendete Netz *Mask R-CNN* [114] wurde von den ursprünglichen Autor\*innen auf Basis des „COCO“-Datensatzes [116] trainiert und geprüft, wobei auch die dort definierten Evaluationsmetriken zur Anwendung kommen. Einzelteile der Netzarchitektur können ausgetauscht werden, um die Abhängigkeit der Leistungsfähigkeit von diesen zu untersuchen. Zu den Elementen gehört der *backbone* (siehe Kapitel 3.5.3.2), der für die Extraktion der Merkmale aus dem Bild zuständig ist, sowie der *head*, welcher als Abschluss auf den *backbone* folgt und die ROI-Erkennung sowie die Prädiktion der Maske in die einzelnen ROIs bewerkstelligt.

Soll die Performanz eines modellbasierten Objektdetektors beschrieben werden, wird meist die *mean average precision* (mAP) als Metrik eingesetzt. Diese vereint die Leistung des Modells für die Klassifikationsaufgabe als auch die Lokalisierung des Objekts. Eine ausführliche Beschreibung der Metrik ist unter [128] zu finden. Zur Bewertung der Klassifikation wird die Fläche unterhalb der Genauigkeits-Sensitivitätskurve für alle Objektklassen berechnet und gemittelt, was die *average precision* (AP) ergibt. Der veränderliche Parameter ist dabei der Konfidenzschwellenwert, welcher überschritten werden muss, um eine gültige Klassifikation des Modells zu erhalten. Klassifikationen mit geringerer Konfidenz werden verworfen. Die Güte der Lokalisierung lässt sich über die *intersection over union* (IoU) berechnen. Diese vergleicht die ROI oder Segmentierungsmaske eines Objekts mit der gegebenen Grundwahrheit des Objekts. Dabei wird die überschneidende Fläche in der Bildebene aus dem lokalisierten Objekt und der Grundwahrheit durch deren vereinigte Fläche geteilt. Folglich kann als weiterer Parameter ein Schwellenwert für die IoU eingeführt werden, welcher überschritten werden muss, um eine valide Objektdetektion zu erhalten. Der mAP-Wert berechnet sich aus dem Mittelwert der AP über mehrere IoU-Schwellenwerte.

Die mit dem „COCO“-Datensatz gelieferten Metriken, mit welchen auch das Modell *Mask R-CNN* evaluiert wird, verwendet mehrere Varianten der mAP, wobei in der „COCO“-Notation die beschriebene Metrik AP genannt wird, was nicht mit der oben beschriebenen AP verwechselt werden darf. Die IoU-Werte werden dabei anhand der erzeugten Maske der Objektinstanzsegmentierung berechnet. Die hauptsächlich verwendete Metrik ist der mAP-Wert über das Intervall von IoU-Schwellenwerten  $I_{IoU}$ , mit  $I_{IoU} = [0,5 : 0,05 : 0,95]$ . Weiterhin wird der mAP-Wert gesondert für IoU-Schwellenwerte von 0,5 und 0,75 berechnet. Zusätzlich wird der mAP-Wert anhand der Objektgröße auf der Bildebene in kleine, mittlere und große Objekte gruppiert. Die Details können unter [116] eingesehen werden. Offensichtlich sind die

Kennzahlen nicht mit denen aus Kapitel 4.2.1 vergleichbar, was durch das grundlegende Vorgehen und der verfügbaren Datenbasis bzw. der fehlenden Grundwahrheit für generische Objekte begründet ist.

Auf eine detaillierte Auflistung der Kennzahlen wird an dieser Stelle aufgrund der nicht gegebenen Vergleichbarkeit zu den Ergebnissen der regelbasierten Objektdetektion in Kapitel 4.2.1 verzichtet, und auf die ursprüngliche Veröffentlichung [114] sowie die „COCO“- [116], „KITTI“- [34] und „Cityscapes“-Benchmarks [129] verwiesen. Dort wird aufgezeigt wie die Leistung des Modells mit verschiedenen *backbones* und *heads* skaliert, wodurch auch in Grenzen eine Anpassung an die verfügbaren Hardware-Ressourcen möglich ist. Weiterhin wird gezeigt, dass das Modell den Stand der Technik zum Zeitpunkt der Publikation in allen Belangen übertrifft. Die Autor\*innen beschreiben ebenfalls erhebliche Fortschritte bei der Evaluation anhand des anspruchsvollen „Cityscape“-Datensatzes [129], welcher sich auf urbane Verkehrsszenen konzentriert. Die in dieser Arbeit verwendete Implementierung basiert auf *Feature Pyramid Network* (FPN) und einem ResNet101 *backbone*. Diese Kombination wurde ebenfalls von den Autor\*innen evaluiert.

Mit der Veröffentlichung von [130] wurde erstmals die menschliche Leistungsfähigkeit in der Bildklassifikation im „ImageNet“ Klassifikationswettbewerb [131] übertroffen und seither stetig verbessert. Bei einem auf den Anwendungsbereich maßgeschneiderten Trainingsdatensatz kann also von einer ausreichenden Klassifikationsgüte ausgegangen werden, insbesondere im Vergleich zur manuellen Auswertung von NCS-Daten. Die Güte der Lokalisierung in der Bildebene hat im vorliegenden Anwendungsfall direkte Auswirkungen auf die Objektbreite und -höhe. Im Vergleich zur Abschätzung dieser Werte anhand der *u-disparity* können jedoch Verbesserungen erwartet werden, da der regelbasierte Ansatz von Rauschen und *foreground fattening* in der Disparitätenkarte beeinflusst wird. Ersteres ist u.a. von der Kalibrationsgüte abhängig. Durch die Objektmaske lassen sich genaue Disparitäten über die Objektkontur ermitteln, was die Berechnung der Objekttiefe robuster gestaltet, da bei geringen Abweichungen von der tatsächlichen Objektfläche keine Disparitäten des Hintergrunds oder anderer Objekte durch komplexe Algorithmen oder Annahmen gefiltert werden müssen, wie in Kapitel 3.5.3.3 dargelegt wurde.

Die Tatsache das die modellbasierte Objekterkennung auf trainierte Klassen begrenzt ist, mag zuerst als konzeptuelles Problem erscheinen. Für die Objektverfolgung wird die Datenassoziation jedoch erheblich erleichtert wenn Klasseninformationen verfügbar sind, da Zuordnungen nur innerhalb einer Klasse erlaubt werden. Weiterhin bietet die Unabhängigkeit der Objekterkennung von der Disparitätenkarte die Möglichkeit, letztere nur bei Bedarf bzw. nur für relevante Bildregionen zu berechnen, was bei Bedarf zur Laufzeitoptimierung des Analysetools verwendet werden kann.

## 4.3 Funktionsanalyse - Grundwahrheit und realer Verkehr

Bei der Analyse des Gesamtsystems stellt die Verfügbarkeit der Grundwahrheit eine große Hürde dar. Um diese für Szenarien im realen Verkehr zu generieren sind komplementäre Sensoren hoher Güte mit speziellen Eigenschaften notwendig, um die Objekte der Grundwahrheit nach den Vorgaben der Objektdetektion extrahieren zu

können. Weiterhin müssen diese Extraktionen gelabelt und zeitlich gekoppelt werden, um das Ergebnis des *Trackings* prüfen zu können. Dieser Vorgang ist somit nicht nur technisch eine Herausforderung, sondern bedarf auch einer manuellen Bearbeitung erheblichen Aufwands. Für die Variante mit regelbasierter Objektdetektion ist der Vorgang eine besonders große Hürde, da die zu detektierenden Objekte nicht auf Klassen beschränkt sind, sondern sich über ihre Form bzw. Objekthöhe und Breite definieren.

Eine Fusion aus Kameradaten mit LiDAR-Punktwolken kann je nach Spezifikation der Sensoren und Anforderungen der zu testenden Algorithmen zielführend sein. Die Kameradaten erlauben ein manuelles *Labeling* und eine zeitliche Kopplung der Objekte, die LiDAR-Punktwolke beinhaltet hingegen Informationen über die Objektform und Position in Weltkoordinaten. Ein *Clustering* der Punktwolke kann die Grundwahrheit für die generische Objekterkennung darstellen. Somit können beide Prozessketten unabhängig von der verwendeten Objekterkennung geprüft werden. Der Aufbau an einem Messfahrzeug gestaltet sich aufgrund der Komplexität des Systems und den Sensormaßen jedoch als schwer umsetzbar. Weiterhin ist ein solcher Messaufbau mit hohen Kosten verbunden, welche jedoch durch die benötigte Arbeitsleistung zum Aufzeichnen des notwendigen Datenvolumens und dessen manueller Nachbearbeitung noch übertroffen werden.

Um dieses Dilemma zu lösen werden in Kapitel 4.3.1 künstlich generierte Szenarien erzeugt, welche durch Ausstattung der Konfliktbeteiligten mit passender Sensorik eine Grundwahrheit bereitstellen. Diese Szenarien prüfen dabei primär die Güte der Konfliktindikatoren ab, lassen jedoch auch Schlüsse auf die Leistungsfähigkeit einzelner Teilkomponenten der Prozesskette zu. Um dennoch die Funktionalität der Prozessketten im realen Verkehr zu prüfen, werden ausgewählte Szenarien in Kapitel 4.3.2 qualitativ analysiert.

### 4.3.1 Funktionsanalyse mit Hilfe von konstruierten Szenarien

Für die beiden verwendeten Konfliktindikatoren TTC und PET werden insgesamt drei Versuchsszenarien definiert. Zwei davon fallen auf die Prüfung der TTC, wobei die Stereokamera stationär befestigt ist und ein Objekt mit verschiedenen Trajektorien auf das Kamerasystem zufährt. Für das Versuchsszenario zum Prüfen der PET wird die Stereokamera auf einem beweglichen Versuchsträger, einem Pkw, befestigt, während das Zielobjekt sich ebenfalls bewegt.

Das zu analysierende Objekt ist jeweils ein Crashtest-Dummy auf einem Fahrrad. Das Objekt wird dabei durch eine geführte Aufhängung bewegt, wodurch sich eine Solltrajektorie umsetzen lässt. Das Fahrrad des Objekts ist mit einem Unfalldatenschreiber (UDS) bestückt, welcher anhand eines Accelerometers und eines Raddrehzahlmessers die gefahrene Entfernung und Geschwindigkeit entlang der Solltrajektorie bestimmt. Start und Endposition des Objekts werden bezüglich eines ortsfesten Kontrollpunkts mit einem Laserentfernungsmesser gemessen. Der Pkw, welcher als Träger der Stereokamera für den PET-Versuch verwendet wird, ist mit einem Correvit-Sensor ausgestattet. Dieser misst über die Verschiebung des Bodens die zurückgelegte Entfernung in Längsrichtung, woraus ebenfalls die Geschwindigkeit abgeleitet wird. Auch die Start- und Endposition des Pkw wird mit einem Laserentfernungsmesser vermessen. Weiterhin ist der Pkw über einen Seilzug und mehrere Umlenkrollen mit der Aufhängung des Zielobjekts verbunden, wodurch die

Bewegung von Kameraträger und Zielobjekt für den PET-Versuch gekoppelt werden können. Um das Kamerasystem mit den Messungen des Zielobjekts und des Correvits zu synchronisieren, werden überfahrbare Trigger verwendet, welche eine LED im Sichtfeld des Kamerasystems auslösen. Die Einzelheiten zu den drei Versuchsszenarien werden in den folgenden Abschnitten erläutert.

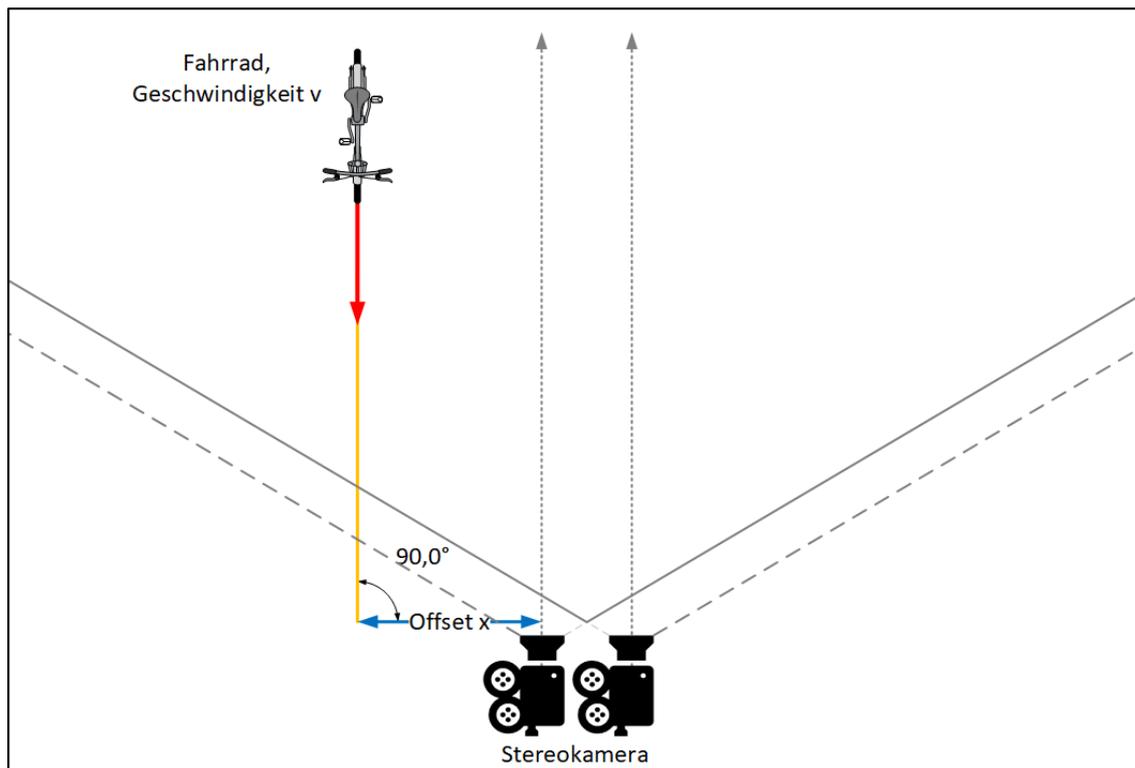
Für die Auswertung der Versuchsszenarien gelten je nach Methode verschiedene Bedingungen. Generell werden die Ansätze wie in Kapitel 3 beschrieben verwendet, die Parametrisierung unterscheidet sich jedoch zwischen den Ansätzen für die Berechnung der Disparitätenkarte und die Objektverfolgung. Weiterhin müssen einige spezifische Parameter für beide Ansätze gesetzt werden. Hierfür wurden wenige unterschiedliche Szenarien analysiert und eine funktionierende Parametrisierung gefunden. Auf eine weitere Parameterstudie wird an dieser Stelle verzichtet und auf die Ergebnisse aus den vorangehenden Kapiteln verwiesen.

Für den regelbasierten Ansatz wird eine möglichst dichte Disparitätenkarte mit wenig Rauschen angestrebt, um die getroffenen Annahmen nicht zu verletzen. Bei der Berechnung der Disparität einer ROI aus dem modellbasierten Ansatz wird weniger Wert auf die Dichte oder geringes Rauschen, als auf Korrektheit der Werte innerhalb der ROI gelegt. Die Schwellenwerte für die *u-disparity* und die minimale Konturbreite der regelbasierten Objektdetektion werden jeweils auf 25 px gesetzt. Die modellbasierte Objektdetektion beschränkt sich für die Validationsversuche auf Personen als einzige relevante Klasse. Durch die unterschiedlichen Charakteristiken der Objektdetektion der beiden Ansätze wird das Messrauschen für die Objektverfolgung unterschiedlich definiert. Ein weiterer Unterschied ist die Initialisierung der *Tracks* während der Objektverfolgung. Durch die geringe Tiefenauflösung der Stereokamera bei großen Entfernungen, können einzelne Objekte in diesem Bereich nur schlecht durch den regelbasierten Ansatz extrahiert werden. Um diesen Sachverhalt Sorge zu tragen, wird für den regelbasierten Ansatz die Initialisierung eines *Tracks* nur mit Objektdetektionen erlaubt, welche eine Entfernung von weniger als 12 m aufweisen. Die Initialisierung eines *Tracks* des modellbasierten Ansatzes unterliegt keiner solchen Einschränkung, da die Objektdetektion in diesem Fall nicht von der Disparitätenkarte und deren Auflösung abhängig ist. Alle verwendeten Parameter sind im Anhang A.1 gelistet.

#### 4.3.1.1 Beschreibung der Versuche zur Bestimmung der TTC

Um die Berechnung der TTC und somit auch das Gesamtsystem zu prüfen, werden zwei unterschiedliche Szenarien getestet. Im ersten, dargestellt in Abbildung 4.9, fährt das Versuchsobjekt frontal auf das Kamerasystem zu. Die in Gelb dargestellte Trajektorie des Objekts weist einen Versatz  $x$  zum Kamerakoordinatensystem auf. Weiterhin werden qualitativ die Sichtfelder der beiden Kameras (grau gestrichelt) und deren Überlappung (grau) dargestellt. Existiert ein Kollisionskurs, so muss das System eine TTC berechnen. Diese wird der TTC, welche aus den Messdaten des UDS berechnet wird, gegenübergestellt. Der laterale Versatz  $x$  der Stereokamera zur Trajektorie des Zielobjekts erlaubt es zu prüfen, ob das System eine knappe Vorbeifahrt ohne Kollisionskurs erkennt. Es werden jeweils fünf Versuche mit einem *Offset* von 0 m bis 0,6 m in Schritten von 0,2 m analysiert.

Im zweiten Szenario wird ein Winkel  $\alpha$  zwischen Trajektorie und Kamerasystem variiert, wie in Abbildung 4.10 dargestellt. Der Winkel nimmt Werte zwischen  $10^\circ$  und  $30^\circ$  an, und wird in Schritten von  $10^\circ$  variiert. Im Gegensatz zum ersten



**Abbildung 4.9**

Validationsversuch zum Prüfen der TTC. Es wird ein lateraler Versatz (blau) der Stereokamera zur Trajektorie (gelb) des Zielobjekts variiert.

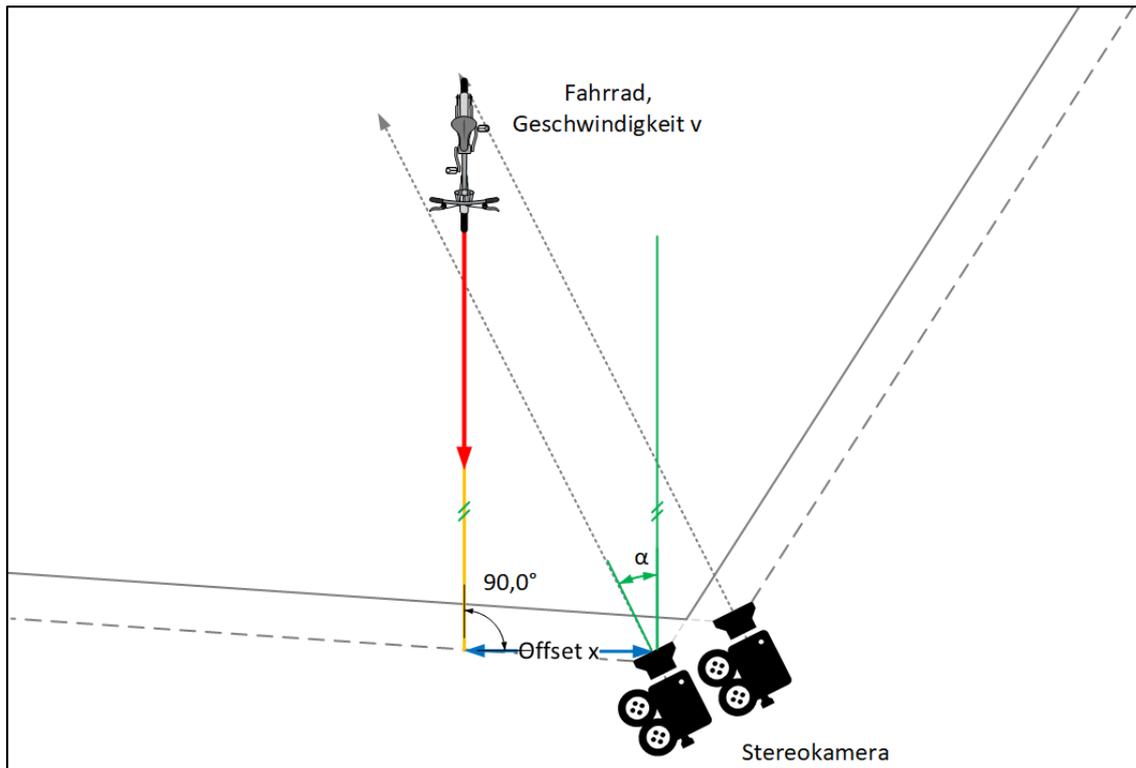
Versuchsszenario spielt bei der Berechnung der TTC auch die Geschwindigkeit in lateraler Richtung eine Rolle. Weiterhin wird sich das Zielobjekt in Abhängigkeit von  $\alpha$  näher am Rand des Bilds beider Kameras befinden. Da in diesem Bereich größere Verzerrungen in den Rohbildern auftreten als im zentralen Bildbereich, kann der Versuch zeigen, ob dies die Güte der Messungen und der abgeleiteten Konfliktindikatoren beeinflusst. Um das Verhalten bei sinkender Objektüberlappung zu evaluieren, wird zusätzlich ein *Offset* von 0 m und 0,4 m geprüft. In beiden Versuchsszenarien können nicht nur die berechneten Konfliktindikatoren, sondern auch die berechneten Positionen in Weltkoordinaten sowie die Objektgeschwindigkeit mit den UDS Messungen verglichen werden.

#### 4.3.1.2 Auswertung TTC

Bevor die einzelnen Versuche ausgewertet werden ist es essenziell zu wissen, inwiefern die gemessenen Signale des UDS als Grundwahrheit verwendet werden können. Nach dieser Analyse werden die Ergebnisse der beiden Szenarien hinsichtlich verschiedener Aspekte diskutiert, um die Gültigkeit der TTC-Berechnung anhand der beiden Ansätze bewerten zu können.

#### Verwendung der UDS-Daten als Grundwahrheit

Für die Auswertung der Versuche werden Vergleichswerte benötigt, welche aus den aufgezeichneten Daten des UDS berechnet werden können. Der UDS der Firma „VDO Kienzle GmbH“, Bauart 1.3, wird hierbei mit einer Abtastrate von 25 Hz betrieben

**Abbildung 4.10**

Validationsversuch zum Prüfen der TTC. Es wird der Winkel zwischen Kamera und Objekttrajektorie sowie der laterale *Offset* variiert.

und gibt in der verwendeten Konfiguration Signale für die Quer- und Längsbeschleunigung sowie die Längsgeschwindigkeit aus. Durch einfache Integration kann somit der zurückgelegte Weg und daraus die TTC als Vergleichswert berechnet werden.

Da das System keine Informationen über die Genauigkeit der Signale liefert, müssen diese ermittelt werden. Aufgrund fehlender Informationen über die Funktionsweise des Systems wird angenommen, dass die ausgegebene Geschwindigkeit aus der Integration der bereinigten Beschleunigungssignale berechnet wird, wobei die Beschleunigungssignale um deren im Stillstand berechneten Bias bereinigt sind. Für die Stillstandserkennung wird das externe Signal der Raddrehzahl verwendet. Eine Sichtung der ausgegebenen Beschleunigungssignale zeigt keine Längsbeschleunigung im Stillstand, trotz einer nicht exakt horizontalen Lage, was die Annahme bestätigt. Aufgrund der Bereinigung um den Bias und der kurzen Dauer der einzelnen Versuche, wodurch der Bias als konstant angenommen werden kann, lässt sich der Fehler des berechneten Geschwindigkeits- und Wegsignals als stochastischer Prozess, einen sogenannten *random walk*, beschreiben [132]. Andere Fehleranteile des Beschleunigungssensors, wie z. B. Skalenfehler, werden als nicht relevant angenommen. Ein normalverteiltes weißes Rauschen nach Formel 4.1 überlagert dabei das Beschleunigungssignal, was bei Integration zum *random walk* führt. Die Varianz des Geschwindigkeitssignals ergibt sich nach Gleichung 4.2, die Varianz des Wegsignals nach Formel 4.3, wobei  $f_s$  die Abtastrate des Sensors darstellt.

$$\epsilon(t) \sim \mathcal{N}(0, \sigma_a^2) \quad (4.1)$$

$$\sigma_v(t)^2 = \frac{1}{f_s} \cdot t \cdot \sigma_a^2 \quad (4.2)$$

$$\sigma_s(t)^2 \approx \frac{1}{3} \cdot \frac{1}{f_s} \cdot t^3 \cdot \sigma_a^2 \quad (4.3)$$

Da das UDS keine Langzeitmessungen über mehrere Stunden unterstützt [133], kann  $\sigma_a$  nicht über die Berechnung der Allan Varianz bestimmt werden. Zur Verfügung stehen jedoch 20 Messungen entlang einer geraden Trajektorie, deren Start- und Endpunkt mit einem Laserentfernungsmesser vermessen sind, wodurch die gefahrene, normalverteilte Entfernung  $s_{GT}$  bekannt ist. Die Messgenauigkeit des Sensors ist mit  $\pm 2,0$  mm im Vergleich zu den Entfernungsmessungen mit dem UDS zu vernachlässigen. Alle Messungen sind mit unterschiedlichen Geschwindigkeitsprofilen  $v(t)$  eingefahren, was auch zu unterschiedlichen Gesamtzeiten  $t_{end}$  führt. Der Erwartungswert für  $s(t_{end})$  entspricht somit  $s_{GT}$ . Für jeden Versuch  $i \in [1; 20]$  gilt Gleichung 4.4 woraus sich  $\sigma_a$  ableiten lässt. Die Ergebnisse sind in Tabelle 4.1 dargestellt.

$$\sigma_{s_i}(t_{end_i})^2 = \frac{1}{3} \cdot \frac{1}{f_s} \cdot t_{end_i}^3 \cdot \sigma_a^2 = (s_i(t_{end_i}) - s_{GT})^2 \quad (4.4)$$

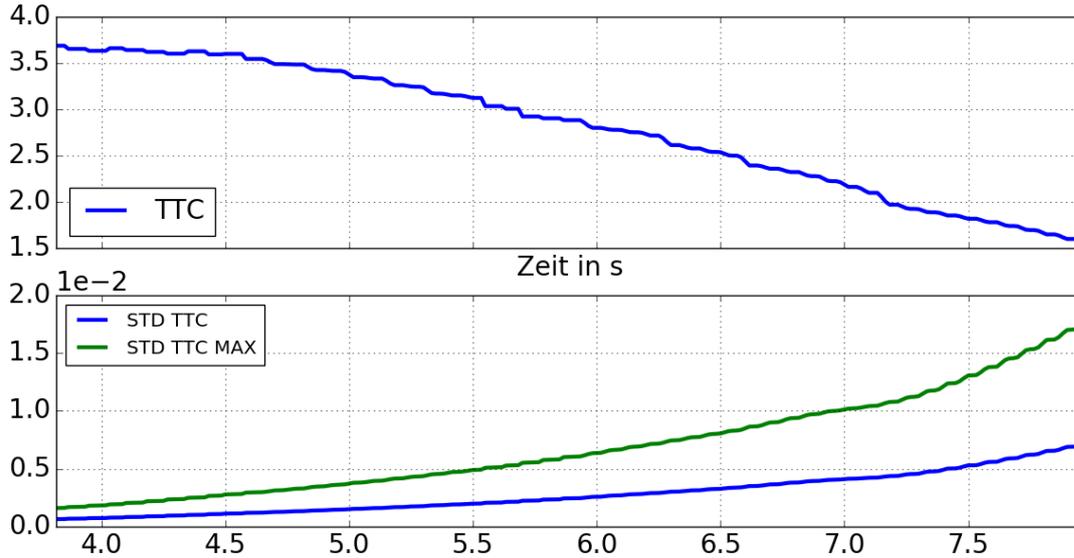
**Tabelle 4.1**Ermittlung von  $\sigma_a$ 

Parameter	Wert	Einheit
Mittelwert	$1,5 \cdot 10^{-3}$	m/s <sup>2</sup>
Maximum	$3,7 \cdot 10^{-3}$	m/s <sup>2</sup>
Standardabweichung	$1,1 \cdot 10^{-3}$	m/s <sup>2</sup>

Mit den Gleichungen 4.2 und 4.3 kann anhand der Gaußschen Fehlerfortpflanzung für unkorrelierte Größen die Varianz der TTC berechnet werden. In der Formel der TTC nach Gleichung 3.24 wird die Ausdehnung des eigenen Fahrzeugs  $d_Z = 0$  m definiert. Die Entfernung  $Z$  der Kamera von dem verfolgten Objekt wird über die per Laserentfernungsmesser vermessene Startposition und den vom UDS berechneten zurückgelegten Weg ermittelt. Unter Vernachlässigung der Messunsicherheit des Laserentfernungsmessers und Substitution des Zählers mit  $S_{Obj} = -Z - d_z$  ergibt sich für die TTC aus UDS-Daten Gleichung 4.5, wobei die Unsicherheit von  $S_{Obj}$  dem beschriebenen  $\sigma_s$  aus Gleichung 4.3 entspricht.

$$TTC_{UDS} = \frac{S_{Obj}}{V_{Obj}} \quad (4.5)$$

Nach der Gaußschen Fehlerfortpflanzung ergibt sich Gleichung 4.6 für die Varianz der UDS-basierten TTC. Nach Einsetzen der Varianzen für den zurückgelegten Weg und die Geschwindigkeit und Berechnen der partiellen Ableitungen ergibt sich Gleichung 4.7. In Abbildung 4.11 wird die für einen Versuch berechnete TTC und deren Standardabweichung dargestellt. Die Berechnung dieser basiert für die blaue


**Abbildung 4.11**

Exemplarische TTC und Standardabweichung aus UDS-Daten für einen Validationsversuch. Die blaue Kurve wurde anhand des Mittelwerts von  $\sigma_a$  aus Tabelle 4.1 berechnet. Die grüne Kurve basiert auf dem Maximum von  $\sigma_a$ .

Kurve auf dem Mittelwert von  $\sigma_a$ , während die grüne Kurve auf dem Maximalwert basiert. Dieser Versuch ist repräsentativ für alle Validationsversuche, da sich die Dauer der Versuche kaum verändert und somit keine nennenswerte Änderung der Standardabweichung der TTC zu erwarten ist. Die Abbildung zeigt, dass die Standardabweichung der TTC zum Ende des Versuchs auf 0,0075 s angewachsen ist. Für den Maximalwert von  $\sigma_a$  beträgt der Wert 0,018 s. Beide Werte sind erheblich kleiner als die Unsicherheiten welche in der vorliegenden Untersuchung durch die Berechnung der TTC auf Basis der Stereokamera zu erwarten sind, weshalb die UDS-basierten Werte unter den getroffenen Annahmen für die Validationsversuche als Grundwahrheit angenommen werden können.

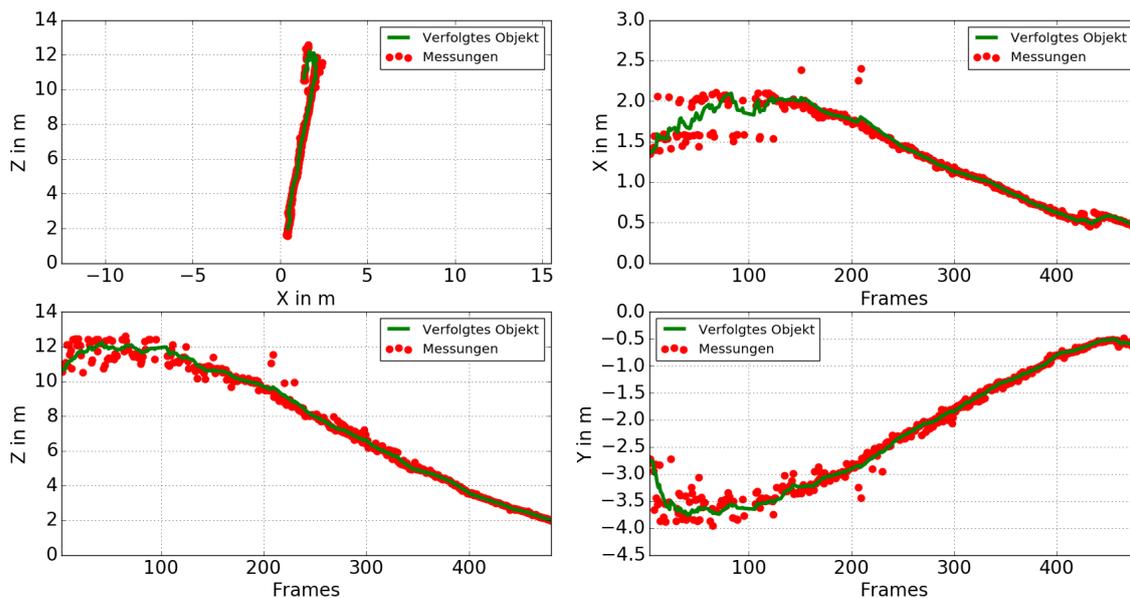
$$\sigma_{TTC}(t)^2 = \left( \frac{\delta TTC_{UDS}}{\delta S_{Obj}} \right)^2 \cdot \sigma_s(t)^2 + \left( \frac{\delta TTC_{UDS}}{\delta V_{Obj}} \right)^2 \cdot \sigma_v(t)^2 \quad (4.6)$$

$$\sigma_{TTC}(t)^2 = \frac{1}{f_S} \cdot t \cdot \sigma_a^2 \cdot \left( \frac{t^2}{3 \cdot V_{Obj}(t)^2} + \frac{S_{Obj}(t)^2}{V_{Obj}(t)^4} \right) \quad (4.7)$$

### Ergebnisse des ersten Szenarios

Im Folgenden werden die Ergebnisse des ersten der 20 Versuche des ersten Szenarios vorgestellt, wobei kein lateraler Versatz zwischen Kamerasystem und Solltrajektorie auftritt. Die Einzelergebnisse der weiteren Versuche können dem Anhang A.2 entnommen werden. Im Anschluss werden die Gesamtergebnisse des Szenarios dargestellt und erläutert.

Anhand der Trajektorie des Objekts sowie der assoziierten Einzelmessungen lassen sich Aussagen über die Performanz der beiden Ansätze der Objektdetektion sowie der Objektverfolgung treffen. Eine Darstellung der Trajektorie in der horizontalen Ebene und die einzelnen Positionen in X-, Y- und Z-Richtung im Zeitverlauf sind für den regelbasierten Ansatz in Abbildung 4.12 und den modellbasierten



**Abbildung 4.12**

Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC-Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

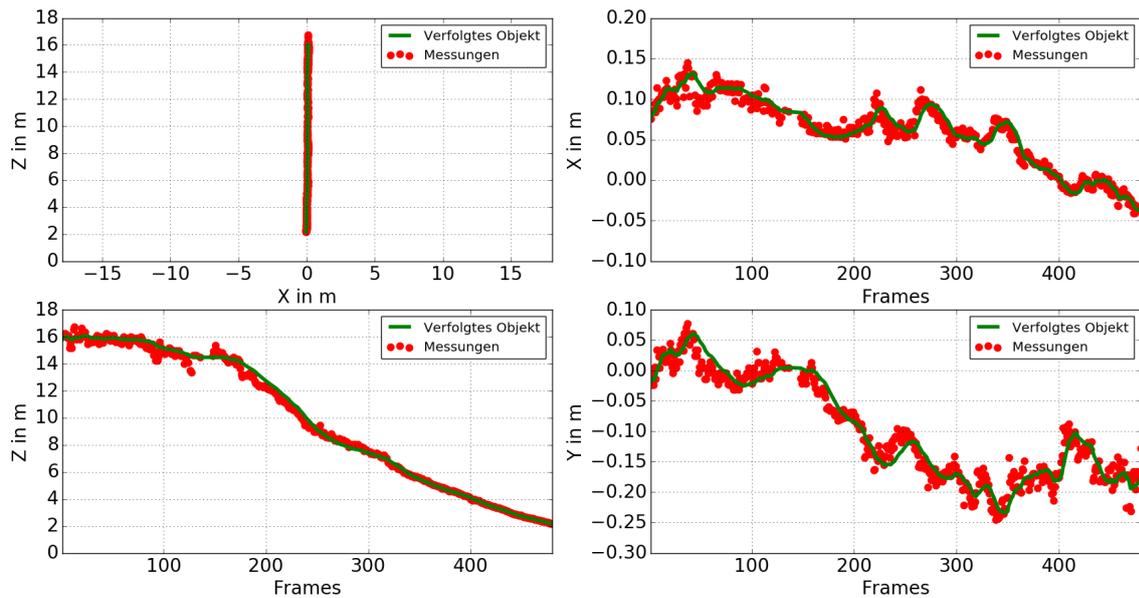
Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.

Ansatz in Abbildung 4.13 dargestellt. Die grünen Linien stellen dabei das kalmangefilterte Signal und die roten Punkte die assoziierten Messungen dar. Für den regelbasierten Ansatz ist eine starke Streuung der Messungen mit steigender Entfernung in allen drei Dimensionen zu erkennen, was sich durch die sinkende Auflösung der Disparität ergibt. Folglich können einzelne Objekte nicht mehr genau genug voneinander getrennt werden, sie verschmelzen zu einigen wenigen Objekten in der Ferne. Die daraus extrahierten Punktmodelle weisen somit ein fehlerhaftes Objektzentrum und eine potenziell falsche Disparität auf. Die Vorgabe einer maximalen Entfernung von 12 m zur Initialisierung des *Tracks* löst das Problem nicht vollständig, wie der Bereich der Trajektorie bis ca. 120 Frames zeigt. Quasi willkürlich geformte, verschmolzene Objekte führen in diesem Bereich zu einer steigenden Entfernung sowie zu einem steigenden vertikalen und horizontalen Versatz, welcher über den gesamten *Track* große Werte abseits der Erwartungen aufweist.

Für den modellbasierten Ansatz ist kein solcher Versatz auszumachen. Die Abweichung beträgt für die maximale Entfernung von 16 m in lateraler Richtung ca. 0,1 m und in der Vertikalen  $-0,2$  m, was einem Fehler von  $0,7^\circ$  in der Vertikalen entspricht. Dieser Fehler kann auf die Ausrichtung der Kamera zurückgeführt werden, da ähnliche Werte für den modellbasierten Ansatz in den Auswertungen der weiteren Versuche ohne *Offset* auftreten. Die Streuung der Messwerte fällt für große Entfernungen erheblich geringer aus. In horizontaler und vertikaler Richtung ist dies auf die Stabilität der Ergebnisse der modellbasierten Objekterkennung zurückzuführen. Die geringe Streuung in der Tiefe resultiert aus der akkuraten Begrenzung des

**Abbildung 4.13**

Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC-Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

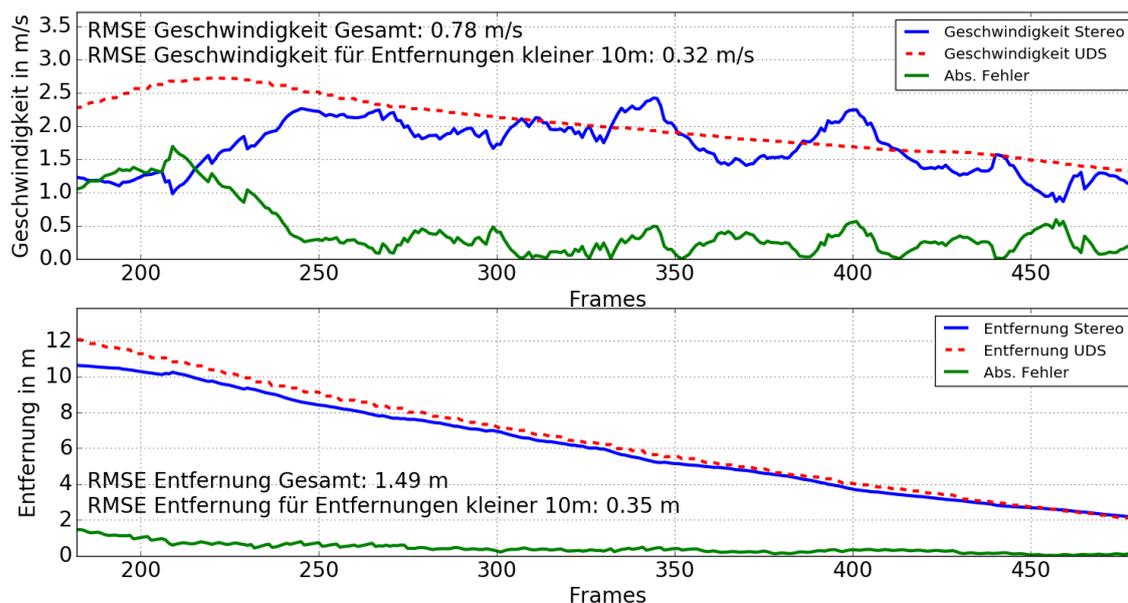
Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.

Objekts durch die Objektsegmentierung und der darüber extrahierten Tiefe aus der Disparitätenkarte. Die Trajektorien beider Ansätze zeigen keine große Sensitivität auf einzelne Ausreißer unter denn Messungen, bilden die Dynamik des Versuchs jedoch gut ab.

Die absolute Geschwindigkeit sowie die Entfernung des Objekts können mit der Grundwahrheit aus dem UDS verglichen werden. Abbildung 4.14 zeigt dies für den regelbasierten Ansatz, Abbildung 4.15 für den modellbasierten, wobei nur der Bereich mit einer erwarteten Entfernung kleiner 12 m dargestellt wird. Dabei stellt die rote Linie jeweils die Grundwahrheit dar, die grüne Linie zeigt den absoluten Fehler zwischen dieser und dem gefilterten Signal in Blau. Für den regelbasierten Ansatz weist das Geschwindigkeitssignal für Entfernungen über 10 m einen vergleichsweise großen Fehler mit Werten über 1,0 m/s auf. Im Nahbereich, definiert durch eine Entfernung kleiner 10 m, beträgt der RMSE 0,3 m/s und ist somit erheblich reduziert. Für den modellbasierten Ansatz sinkt der Fehler ebenfalls sichtbar mit abnehmender Entfernung, beträgt jedoch über den gesamten *Track* ca. 0,6 m/s bzw. 0,5 m/s im Nahbereich. Der RMSE der absoluten Entfernung beträgt im Nahbereich 0,35 m für den regelbasierten Ansatz, während der modellbasierte einen Wert von 0,22 m aufweist. Der Fehler des regelbasierten Ansatzes weist für diesen Versuch eine schlechtere Leistung für die absolute Entfernung, nicht aber für die absolute Geschwindigkeit im Nahbereich auf. Entscheidend für die Berechnung der TTC sind jedoch die Geschwindigkeit sowie die Position in Z-Richtung. Für die Berechnung

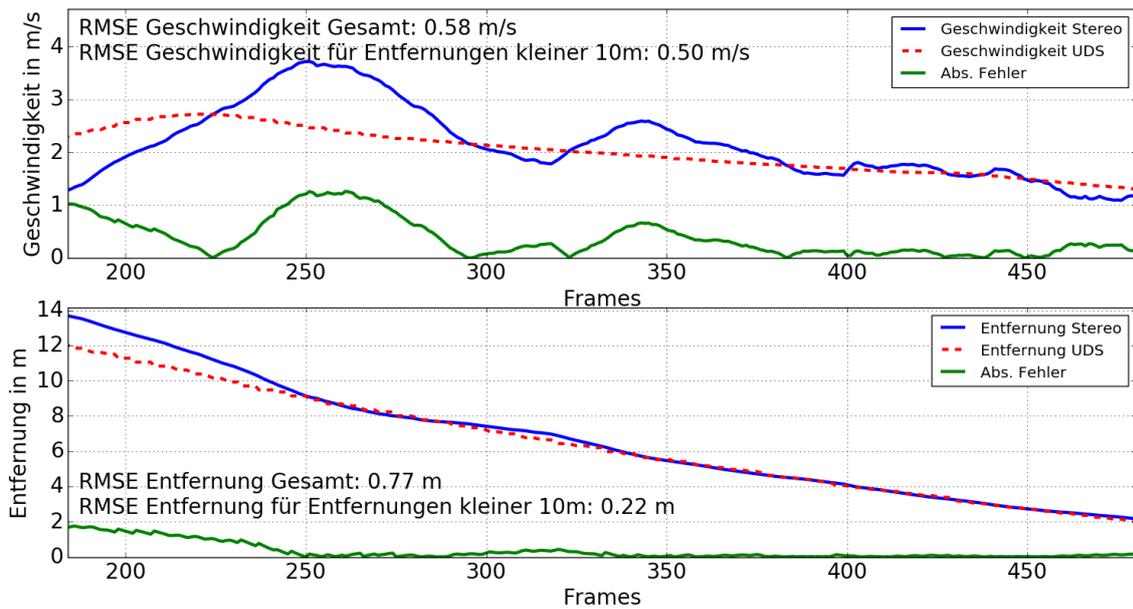
**Abbildung 4.14**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC-Validationsversuche im Szenario 1. Die UDS-Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

des Kollisionskurses spielt zusätzlich die laterale Position und Geschwindigkeit eine Rolle.

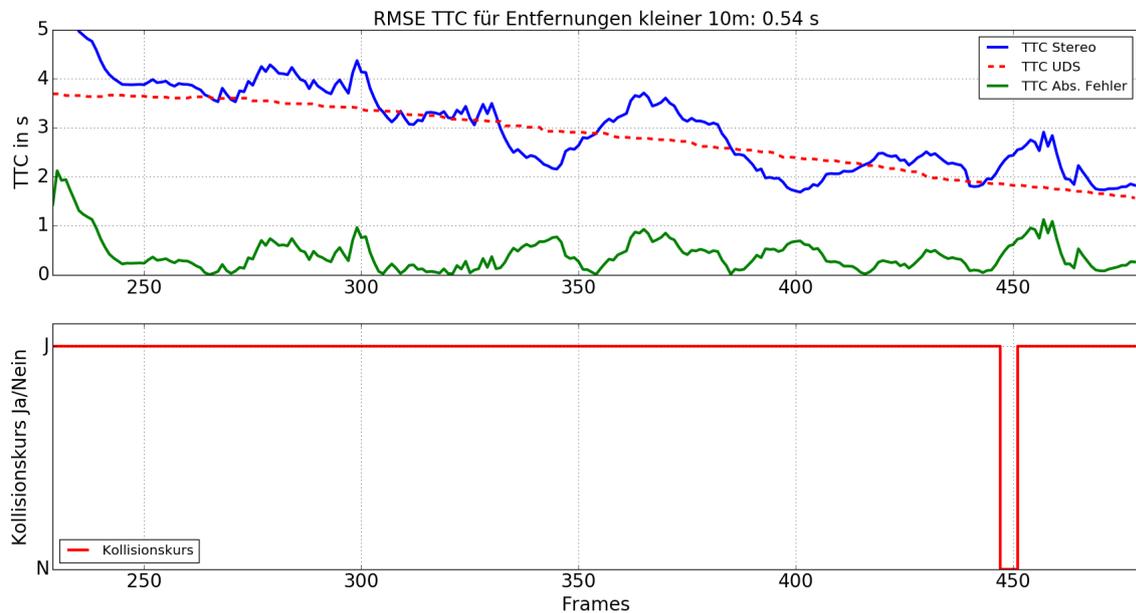
Die berechnete TTC der UDS-Signale sowie der gefilterten Signale aus dem Stereosystem werden für beide Ansätze nur für den Nahbereich angegeben. Die Abbildungen 4.16 und 4.17 bilden diese für beide Ansätze ab. Der regelbasierte Ansatz weist mit 0,54 s einen ähnlichen RMSE wie der modellbasierte Ansatz auf, dessen Fehler 0,56 s beträgt. Ein kleiner Unterschied ist auch für die Genauigkeit der Detektion des Kollisionskurses auszumachen. Unter der Annahme, dass das Ego-Fahrzeug sowie der\*die Konfliktpartner\*in eine Breite von 1 m aufweisen, ist der Kollisionskurs gegeben, wenn beide Objekte zum prädizierten Kollisionszeitpunkt eine laterale Überlappung aufweisen. Dies ist für den vorliegenden Versuch per Definition zu jedem Zeitpunkt gegeben. Dennoch kann mit den Daten des regelbasierten Ansatzes nur für 98 % der Zeit ein Kollisionskurs erkannt werden. In der Trajektorie ist ein Knick bei ca. 3 m Entfernung auszumachen, welcher zu einer kurzzeitigen lateralen Bewegung in X-Richtung führt. Folglich ist für wenige Frames die Kollisionsbedingung nicht erfüllt. Der Versatz spiegelt sich auch in den Messungen der einzelnen Frames wieder. Anhand der Daten des modellbasierten Ansatzes kann durchgehend der richtige Kurs erkannt werden. Die genauere Bedatung des Punktmodells spiegelt sich somit in allen relevanten Signalen wider und bringt insbesondere für die beiden wichtigsten Signale, die TTC und die Detektion des Kollisionskurses, erhebliche Vorteile.

Die Beobachtungen bezüglich der Genauigkeit der richtungsbezogenen Positions- und Geschwindigkeitssignale, der abgeleiteten TTC und der Kollisionskursdetektion spiegeln sich zumeist auch in den Auswertungen über alle 20 Versuche wider. Tabelle 4.2 zeigt die Fehler der Entfernungsberechnung des ersten TTC-Szenarios für beide Ansätze, Tabelle 4.3 die Fehler der Geschwindigkeitsberechnung. Es werden



**Abbildung 4.15**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC-Validationsversuche im Szenario 1. Die UDS-Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

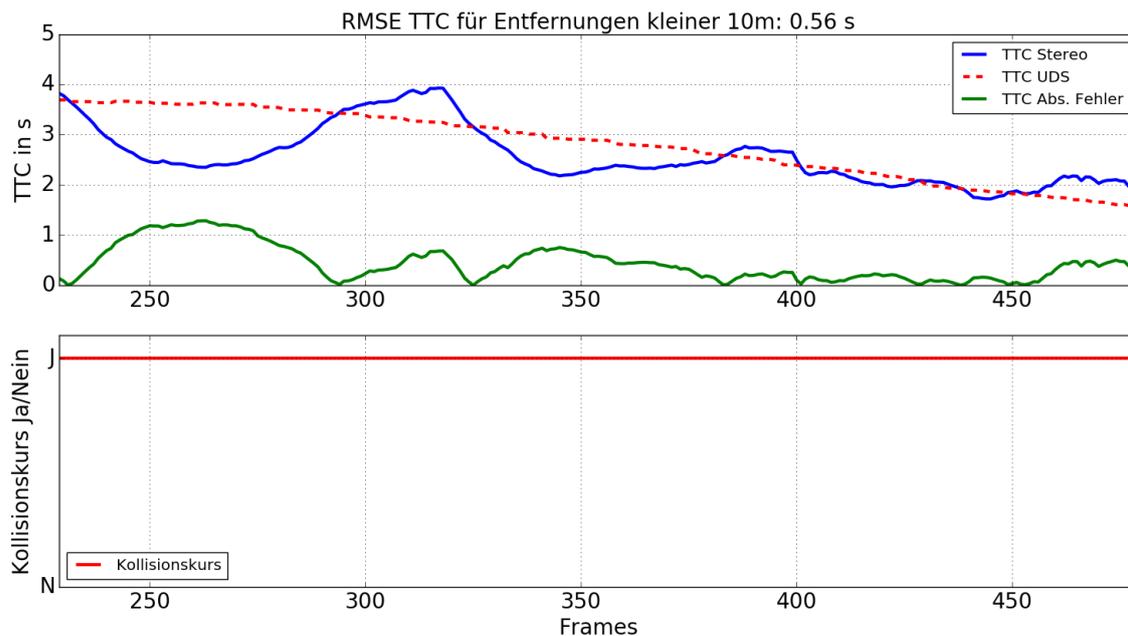


**Abbildung 4.16**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC-Validationsversuche im Szenario 1. Für die TTC werden die UDS-Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

**Abbildung 4.17**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC-Validationsversuche im Szenario 1. Für die TTC werden die UDS-Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün). Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

Mittelwert und Standardabweichung des RMSE über alle Versuche des Szenarios dargestellt. Weiterhin werden die Werte für den Nahbereich mit Entfernungen kleiner 10 m gesondert gelistet. Über alle Versuche zeigt der modellbasierte Ansatz einen signifikant geringeren Fehler für beide Signale, unabhängig davon ob der Nahbereich oder der gesamte *Track* ausgewertet wird. Die Standardabweichung aller RMSE-Werte ist bei Verwendung des regelbasierten Ansatzes für alle Signale erheblich größer.

Dies zeigt sich ebenso in den Ergebnissen der TTC. Tabelle 4.4 zeigt, dass der RMSE der TTC für den regelbasierten Ansatz mit 3,22 s um mehr als Faktor sieben größer ist als für den modellbasierten Ansatz. Weiterhin streut der Fehler stark, was eine Anwendung zur Eventdetektion bereits ausschließt. Der Fehler der modellbasierten Variante, mit einem Fehler von 0,43 s, weist nur eine geringe Streuung auf.

**Tabelle 4.2**

Fehler der Entfernungsberechnung des ersten TTC-Szenarios für beide Ansätze.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Mittlerer RMSE Entfernung Gesamt	2,19	0,52	m
STD RMSE Entfernung Gesamt	0,76	0,27	m
Mittlerer RMSE Entfernung Nahbereich	0,52	0,16	m
STD RMSE Entfernung Nahbereich	0,21	0,03	m

**Tabelle 4.3**

Fehler der Geschwindigkeitsberechnung des ersten TTC-Szenarios für beide Ansätze.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Mittlerer RMSE Geschw. Gesamt	1,04	0,49	m/s
STD RMSE Geschw. Gesamt	0,28	0,06	m/s
Mittlerer RMSE Geschw. Nahbereich	0,55	0,27	m/s
STD RMSE Geschw. Nahbereich	0,20	0,07	m/s

**Tabelle 4.4**

Fehler der TTC-Berechnung des ersten TTC-Szenarios für beide Ansätze.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Mittlerer RMSE TTC Nahbereich	3,22	0,43	s
STD RMSE TTC Nahbereich	5,31	0,11	s

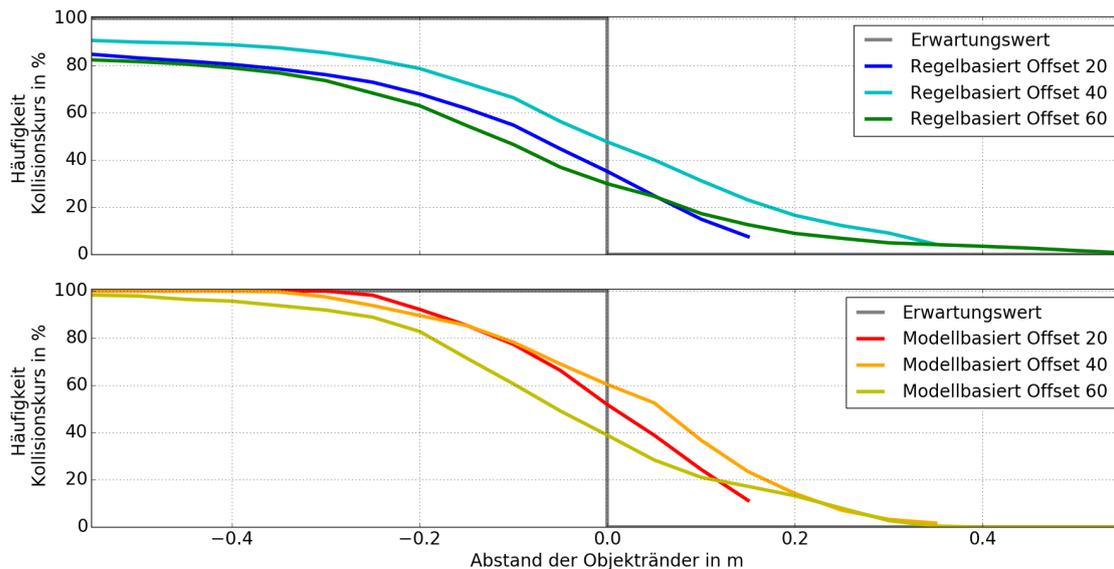
Eine Optimierung von Sensorparametern und algorithmenspezifischen Parametern kann hierbei zu einer weiteren Verbesserung führen. Die wichtigste Metrik zur Bewertung der Anwendbarkeit in der Eventdetektion stellt jedoch die Genauigkeit der Kollisionskursdetektion dar, welche in Tabelle 4.5 dargestellt ist. Diese Genauigkeit beträgt für den oben beschriebenen Fall des 1 m breiten Objekts und Ego-Fahrzeugs 89 % im Nahbereich und 56 % gesamt für den regelbasierten Ansatz. Die modellbasierte Variante weist wiederum erheblich bessere Werte auf. Mit einer korrekten Detektionsrate von 73 % gesamt und 99 % im Nahbereich ist dieser Ansatz valide, da die Fehldetektionen sich immer nur über kurze Zeiträume eines *Tracks* erstrecken.

Da die Kollisionskursdetektion ein essenzieller Teil der Eventdetektion ist, wird diese im Folgenden auf ihre Sensitivität bezüglich der Objektabstände untersucht. Hierfür werden verschiedene Objektbreiten definiert, um unabhängig vom *Offset* des Versuchs eine definierte Überlappung der Objekte bzw. einen definierten Abstand zu erhalten. Versuche ohne *Offset* wurden von der Auswertung aufgrund des von der Objektbreite unabhängigen Kollisionskurses ausgeschlossen. Die Abstände der Objekte wurden in Schritten von 0,05 m verändert, wobei Abstände mit Werten gleich

**Tabelle 4.5**

Genauigkeit der Kollisionskursdetektion unter der Annahme von 1 m Objektbreite und Ego-Breite.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Genauigkeit Gesamt	56	73	%
STD Gesamt	7	7	%
Genauigkeit Nahbereich	89	99	%
STD Nahbereich	10	3	%



**Abbildung 4.18**

Sensitivität der Kollisionsdetektion für verschiedene Objektdistanzen, untersucht für Versuche mit *Offsets* größer null. Der Erwartungswert ist in grau dargestellt.

Oben: Regelbasierter Ansatz.

Unten: Modellbasierter Ansatz.

oder größer des *Offsets* der Trajektorie nicht definiert sind. Abbildung 4.18 zeigt die Ergebnisse für beide Ansätze. Dabei wird die relative Häufigkeit eines detektierten Kollisionskurses über der Objektüberlappung bzw. deren Abstand dargestellt. In Grau ist der Erwartungswert eingezeichnet, welcher für negative Abstände der Objekttränder einer Kollisionsdetektion entspricht. Positive Abstände entsprechen einer Vorbeifahrt, wodurch keine Detektion erwartet wird.

Für den regelbasierten Ansatz wird selbst für große Überlappungen der Objekte keine Kollisionsdetektion jenseits der 90 % erreicht. Die Detektionsrate sinkt für niedrige Überlappungen annähernd linear, und erreicht bei einer Objektüberlappung von 0 m für die Versuche mit 0,2 m, 0,4 m und 0,6 m *Offset* eine Detektionsrate von ca. 35 %, 50 % und 30 %. Für die *Offsets* 0,2 m und 0,4 m sinkt die Detektionsrate annähernd linear auf ca. 10 % für ihre jeweils minimal darstellbare Objektdistanz, bei einem *Offset* von 0,6 m tendiert die Detektionsrate gegen null. Für den modellbasierten Ansatz ergeben sich ähnliche Verläufe der Kollisionsdetektion. Bei allen Trajektorienoffsets bleibt die Detektionsrate bei einer Überlappung größer 0,2 m auf über 80 %, für Objektabstände größer 0,2 m auf unter 20 %. Letzteres trifft auch auf den regelbasierten Ansatz zu. Der modellbasierte Ansatz weist insgesamt ein genaueres Verhalten bezogen auf die Objektüberlappung bzw. -abstände auf.

Anhand der Analyse des ersten Szenarios kann bereits eine Aussage über die Güte der beiden Ansätze getroffen werden. Durch die regelbasierte Objekterkennung werden entfernte Objekte ähnlicher Tiefe mit geringen lateralen Abständen oft als einzelne Objekte zusammengefasst. Dies führt zu fehlerhaften Trajektorien, was anhand des Beispiels in Abbildung 4.12 ersichtlich wird. Eine unmittelbare Folge ist die Beschränkung der maximalen Tiefe von Objekten zur Initialisierung der Objektverfolgung für den regelbasierten Ansatz. Gemittelt über alle Versuche des ersten Szenarios weist dieser Ansatz signifikant größere Fehler für alle geprüften Parameter auf. Die TTC erscheint aufgrund des großen Fehlers und dessen Streuung als nicht

nutzbar. Auch die Kollisionsdetektion erreicht nicht das Niveau des modellbasierten Ansatzes, welcher sich in allen Bereichen durch geringe Fehler auszeichnet. Insbesondere bei geringen Objektabständen oder -überlappungen, wie sie im Fahrradverkehr vermehrt vorkommen, weist der modellbasierte Ansatz eine bessere Detektionsrate auf.

### Ergebnisse des zweiten Szenarios

Im zweiten Szenario zur Bewertung der TTC werden, wie in Kapitel 4.3.1.1 beschrieben, die Trajektorien um einen Winkel  $\alpha$  zur Kamerahauptachse verdreht und mit keinem *Offset*, bzw. mit 0,4 m *Offset* zum Kamerakoordinatensystem beaufschlagt. Ohne *Offset* können unter dem maximalen Winkel von  $30^\circ$  keine Objekte mehr detektiert werden, da der gewählte horizontale Öffnungswinkel durch die Entzerrung der Bilder beschnitten wird. Nur die Versuche mit *Offset* können den maximalen Winkel abbilden. Für jede Variation wurden drei Versuche durchgeführt.

Um den Einfluss des Winkels zu bewerten, wird jeweils mit dem regelbasierten sowie dem modellbasierten Ansatz der RMSE der TTC sowie die Genauigkeit der Kollisionsdetektion überprüft, da diese Parameter indirekt die Güte der berechneten Position und Geschwindigkeit des Objekts widerspiegeln. Die Anzahl der Versuche für jedes Paar aus *Offset* und Winkel ist zwar gering, eine Tendenz bezüglich des Einflusses der Parameter lässt sich dabei jedoch ableiten.

**Tabelle 4.6**

Genauigkeit der TTC im Nahbereich für das zweite TTC-Szenario.

Beschreibung	$\alpha$	Regelbasiert		Modellbasiert		Einheit
<i>Offset</i>	-	0	0,4	0	0,4	m
Mittlerer RMSE	$10^\circ$	0,87	2,29	0,57	0,89	s
STD	$10^\circ$	0,19	2,25	0,13	0,17	s
Mittlerer RMSE	$20^\circ$	0,68	1,82	0,67	0,63	s
STD	$20^\circ$	0,12	0,87	0,12	0,17	s
Mittlerer RMSE	$30^\circ$	-	25,75	-	1,26	s
STD	$30^\circ$	-	22,05	-	0,31	s

Die in Tabelle 4.6 dargestellte TTC zeigt für beide Ansätze und Winkel von  $10^\circ$  und  $20^\circ$  geringe Änderungen der TTC, welche sich in einem Wertebereich befinden wie er aus den vorherigen Versuchen des ersten Szenarios zu erwarten ist. Einzig die Ergebnisse des regelbasierten Ansatzes ohne *Offset* zeigen einen geringeren Fehler als erwartet und liegen dabei fast auf dem Niveau des modellbasierten Ansatzes. Aufgrund der geringen Versuchszahl lassen sich hieraus noch keine Schlüsse ableiten. Für einen Winkel von  $30^\circ$  ergeben sich jedoch erheblich schlechtere RMSE Werte. Das zu verfolgende Objekt befindet sich in diesem Fall über einen Großteil des Versuchs am Bildrand der aufgezeichneten Kamerabilder. Dieser Teil des Bilds ist vor dem Entzerren am stärksten von der radialen Verzerrung betroffen. Ein ansteigender Fehler für Objekte am Bildrand lässt hierbei auf eine fehlerhafte Modellierung

**Tabelle 4.7**

Genauigkeit der Kollisionskursdetektion im Nahbereich für das zweite TTC-Szenario.

Beschreibung	$\alpha$	Regelbasiert		Modellbasiert		Einheit
<i>Offset</i>	-	0	0,4	0	0,4	m
Genauigkeit	10°	95	88	100	97	%
STD	10°	4	2	0	4	%
Genauigkeit	20°	79	58	100	89	%
STD	20°	3	6	0	0	%
Genauigkeit	30°	-	35	-	64	%
STD	30°	-	9	-	7	%

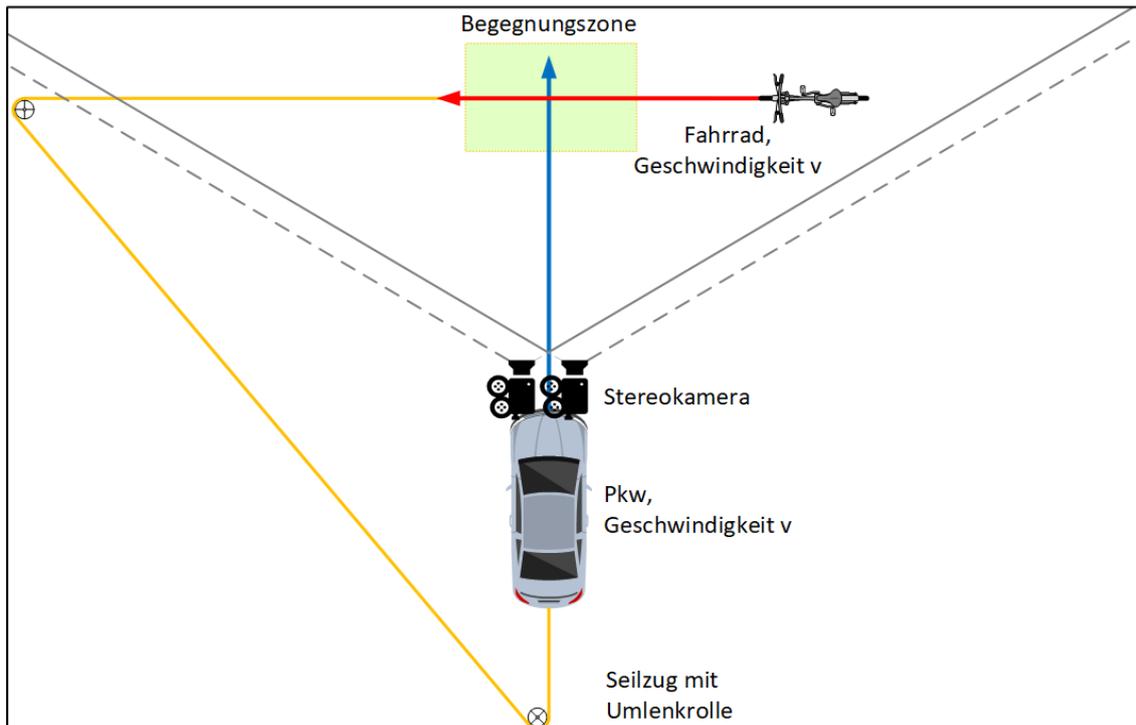
der Verzerrung in diesen Bildregionen schließen. Der Anstieg des Fehlers ist dabei besonders für den regelbasierten Ansatz beträchtlich.

Die Genauigkeit der Kollisionsdetektion, abgebildet in Tabelle 4.7, stellt sich für die Versuche ähnlich dar. Für Winkel von 10° und 20° ergibt sich für den modellbasierten Ansatz keine oder nur eine geringe Verschlechterung der Genauigkeit mit steigendem Winkel. Im regelbasierten Fall verschlechtert sich die Kollisionsdetektion mit steigendem Winkel signifikant. Für ein  $\alpha$  von 30° zeigen beide Ansätze eine reduzierte Genauigkeit, wobei der modellbasierte Ansatz mit 64 % Genauigkeit besser funktioniert als der regelbasierte Ansatz mit 35 % Genauigkeit.

Generell zeigt die Auswertung des Szenarios, wie anfällig beide Algorithmen auf eine ungenügende Entzerrung der Linsenverzeichnung im Randbereich sind. Eine Anpassung des Modells auf die verwendete Kamera sowie ein Test der Entzerrung sind für die Anwendung im NCS-Bereich auf jeden Fall notwendig, da Objekte nahe dem Bildrand im Realverkehr keine Ausnahme bilden. Weiterhin werden die Probleme der fehlenden Abgrenzung zwischen nahen Objekten bei Verwendung der regelbasierten Objektdetektion bestätigt.

#### 4.3.1.3 Beschreibung der Versuche zur Bestimmung der PET

Um die Berechnung der PET zu prüfen, muss ein Szenario konstruiert werden, in welchem die Trajektorien des Zielobjekts und des Kamerafahrzeugs sich schneiden oder zumindest nahekommen. Dabei darf es jedoch nicht zu einer Kollision kommen, sondern die Fahrzeuge müssen sich knapp verpassen. Um dies umzusetzen wird ein Pkw, das Kamerafahrzeug, mit Hilfe einer Seilwinde und Umlenkrollen am Zielobjekt befestigt. Abbildung 4.19 zeigt den Versuchsaufbau. Die Solltrajektorie des Zielobjekts, in Rot dargestellt, und die Seillänge (gelb) werden dabei so gewählt, dass sich die beiden Trajektorien in einer Begegnungszone schneiden, die Objekte jedoch nicht kollidieren. Wie bereits in den Validationsversuchen zur TTC wird anhand der am Pkw und Zielobjekt angebrachten Sensorik ein Vergleichswert zur stereobasierten PET berechnet und verglichen. Insgesamt wurden sechs Versuche für das dargestellte Szenario aufgezeichnet, wovon ein Versuch aufgrund eines Abbruchs der Datenaufzeichnung nicht ausgewertet werden kann.

**Abbildung 4.19**

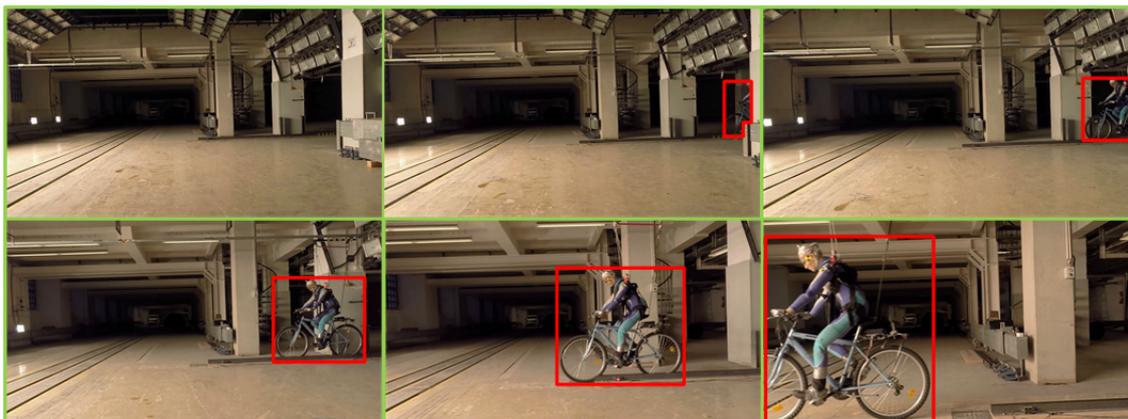
Validationsversuch zum Prüfen der PET. Die Trajektorien des Pkw und des Fahrrads kreuzen sich in der Begegnungszone.

#### 4.3.1.4 Auswertung PET

Analog zur Auswertung der vorangegangenen Versuche zur Bewertung der berechneten TTC wird die Sensorik an beiden verwendeten Fahrzeugen als Grundwahrheit herangezogen. Wie bereits gezeigt wurde, ist der Fehler der Daten des am Fahrrad befestigten UDS gegenüber dem Fehler der stereokamerabasierten Messungen vernachlässigbar. Für die Versuche zur Auswertung der PET wird dieselbe Strecke in ähnlichen Zeiträumen absolviert, weshalb die Auswertung übertragbar ist. Für den verwendeten Correvit der Firma „Corrsys-Datron“ wird eine Messunsicherheit von 0,2% spezifiziert, woraus sich ableiten lässt, dass die verwendete Sensorik für die vorliegenden Versuche ebenfalls als Grundwahrheit verwendet werden kann. Auf eine detaillierte Herleitung, analog zu Kapitel 4.3.1.2, wird an dieser Stelle aufgrund der Ähnlichkeiten verzichtet.

Wie bereits in der Auswertung zur TTC wird für beide Ansätze jeweils die Trajektorie, die absolute Entfernung und Geschwindigkeit sowie der Konfliktindikator berechnet. Abgesehen von der Trajektorie werden alle berechneten Signale mit der Grundwahrheit verglichen. Zusätzlich wird die Grundbedingung zur Berechnung der PET dargestellt, welche durch Gleichung 3.29 beschrieben ist. Im Vergleich zu den vorangegangenen Versuchen stellen sich die Ergebnisse aufgrund der nun bewegten Stereokamera verändert dar, wobei sie ebenfalls im Kamerakoordinatensystem dargestellt werden.

Die Versuchsumgebung und -umsetzung weisen einige Besonderheiten auf, welche in Abbildung 4.20 ersichtlich werden. Die Bildsequenz zeigt Daten der Stereokamera, welche aus einem der Versuche extrahiert wurden. Das zu detektierende Objekt ist zur besseren Sichtbarkeit rot umrandet. Das Objekt ist zu Beginn des Versuchs



**Abbildung 4.20**

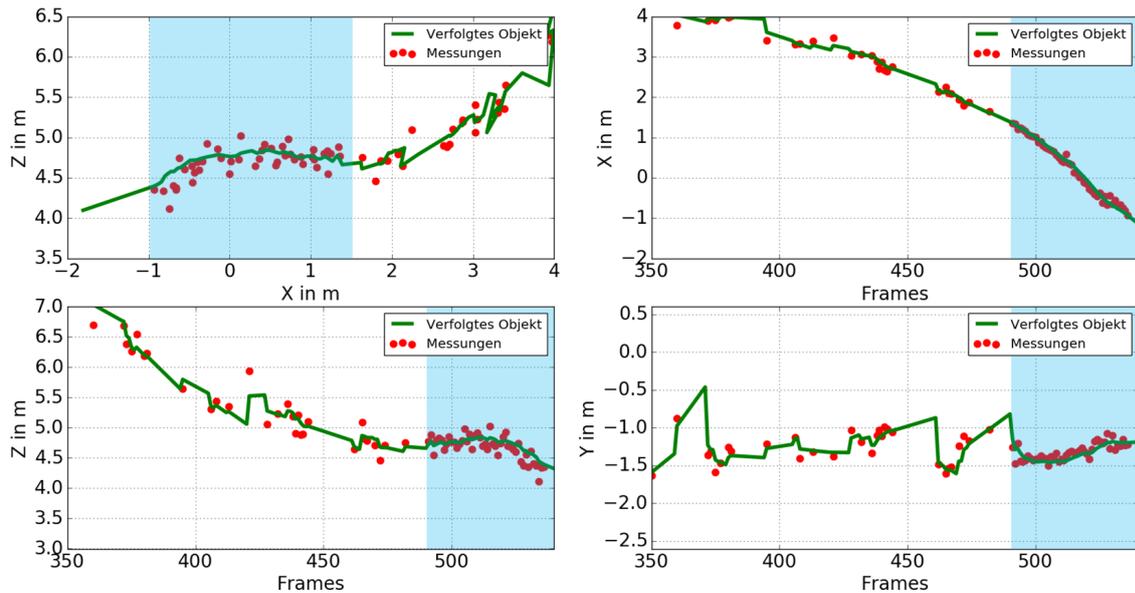
Bildsequenz eines PET-Validationsversuchs. Die Sequenz startet oben links und endet unten rechts. Das zu detektierende Objekt wurde durch rote Umrandungen in den einzelnen Bildern kenntlich gemacht.

nicht im Sichtfeld, sondern tritt erst im weiteren Verlauf hinter einer Säule hervor. Durch die Eigenbewegung des Pkw, und folglich des Kamerasystems, befindet sich das Objekt für einen Großteil des weiteren Versuchverlaufs am rechten Bildrand, bevor sich die Trajektorie der Kamera und des Objekts kurz vor Versuchsende kreuzen. Zusätzlich befinden sich im rechten Bildbereich weitere Säulen in geringer Entfernung zur Objekttrajektorie. Diese Bedingungen sind bei einer Anwendung im Realverkehr realistisch. Spät sichtbare Objekte und dicht besetzter Raum, z. B. durch andere Verkehrsteilnehmer\*innen oder statische Objekte, stellen im urbanen Umfeld keine Ausnahme dar.

Die beschriebenen Versuchsbedingungen und die bisherigen Erkenntnisse aus der Auswertung der TTC lassen auf die Leistung der Algorithmen in dem vorliegenden Versuchsszenario schließen. Da sich das Objekt über einen signifikanten Teil der Versuchsdauer nah am Bildrand befindet, wird das Ergebnis durch die Kalibrationsgüte beeinflusst. Das zweite Szenario der TTC-Versuche hat eine ungenügende Entzerrung im Randbereich des Bildes aufgedeckt, wodurch für diese Bildbereiche erhöhte Fehler der berechneten Signale auftreten. Die verschiedenen Säulen in der Nähe der Objekttrajektorie spiegeln sich in der Performanz des regelbasierten Ansatzes wider, welcher durch die Auflösung der Disparitätenkarte limitiert ist und somit unterschiedliche Objekte fälschlicherweise zu einer Instanz zusammenfasst. Das resultierende fehlerhafte Punktmodell kann in der Objektverfolgung zu offensichtlichen Fehlern im Objektzustand führen, oder aufgrund des *Gatings* zu einer ausbleibenden Assoziation. Folglich wird für den modellbasierten Ansatz eine bessere Leistung erwartet, welche dennoch am Bildrand erhebliche Fehler aufweist.

Im Folgenden wird exemplarisch ein Versuch ausgewertet und in die Ergebnisse in den entsprechenden, aus der TTC-Auswertung bekannten Abbildungen dargestellt. Weiterhin werden die Ergebnisse der gesamten Versuchsreihe zusammenfassend in Tabellen dargelegt. Die Einzelergebnisse der weiteren Versuche können dem Anhang A.2 entnommen werden.

Die in Abbildung 4.21 dargestellte Trajektorie des Objekts sowie die assoziierten Messungen bestätigen die Vermutung, dass durch die regelbasierte Objektdetektion fehlerhafte Punktmodelle generiert werden. Die Trajektorie weist über alle Di-



**Abbildung 4.21**

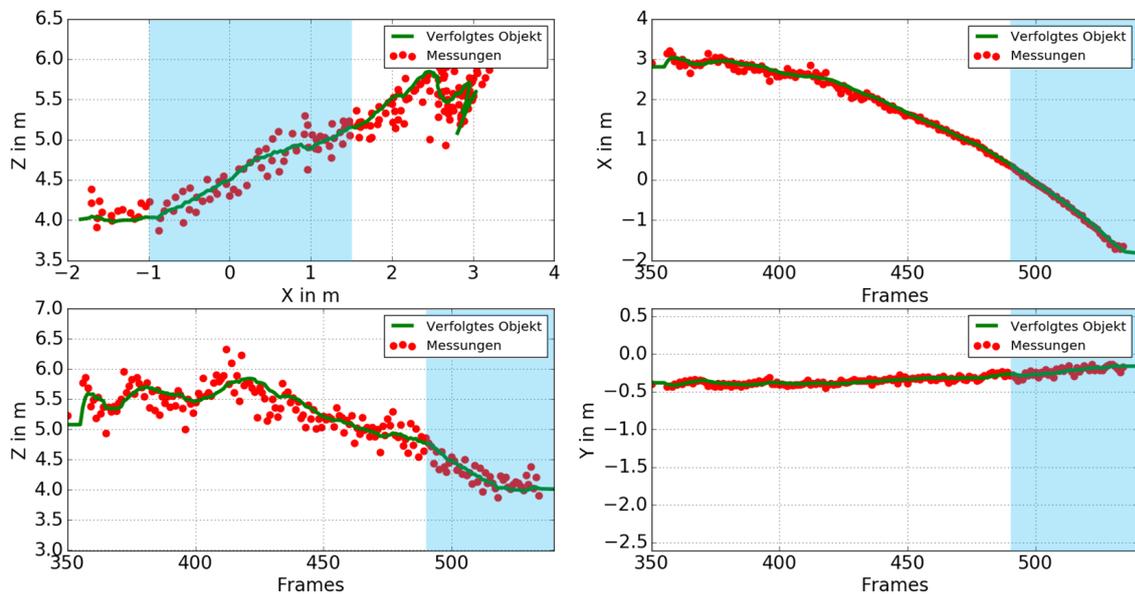
Trajektorie des Objekts eines PET-Versuchs, berechnet mit dem regelbasierten Ansatz. Signale des *Trackers* werden als grüne Linie, verwendete Messwerte als roter Punkt eingezeichnet. Der zentrale Bildbereich bzw. korrespondierende Frames sind blau markiert.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung 4.22**

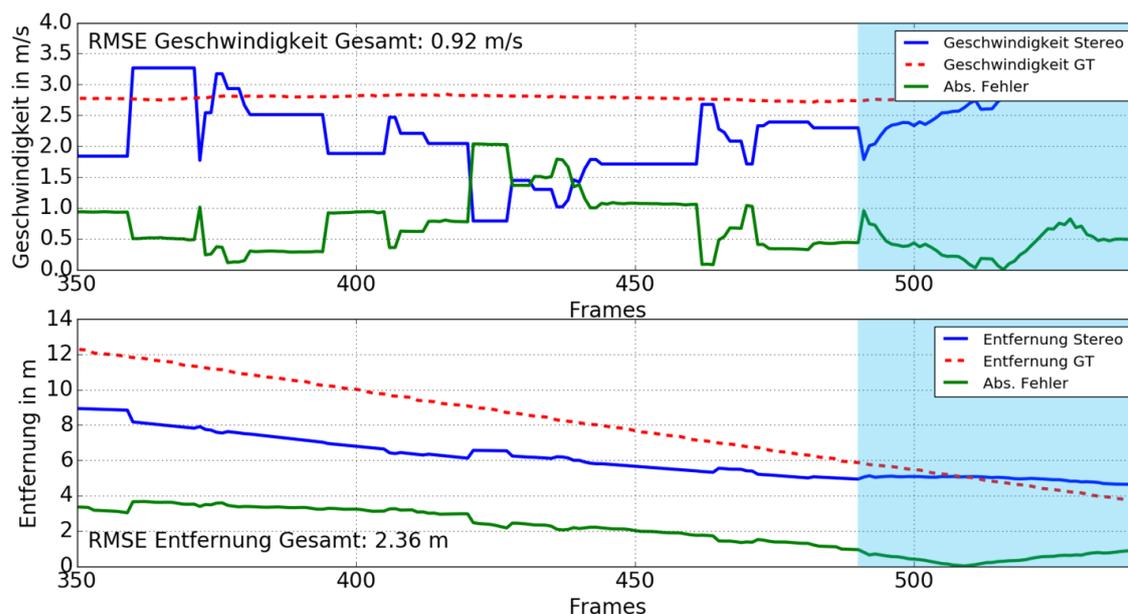
Trajektorie des Objekts eines PET-Versuchs, berechnet mit dem modellbasierten Ansatz. Signale des *Trackers* werden als grüne Linie, verwendete Messwerte als roter Punkt eingezeichnet. Der zentrale Bildbereich bzw. korrespondierende Frames sind blau markiert.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung 4.23**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts eines PET-Versuchs berechnet mit dem regelbasierten Ansatz. Die Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün). Die Frames des zentralen Bildbereichs sind blau markiert.

mensionen offensichtliche Ausreißer auf. Weiterhin wird nur eine geringe Anzahl an Messungen assoziiert, nicht wie im Optimalfall erwartet eine Messung pro Frame. Auffällig ist die höhere Anzahl an Assoziationen im Intervall  $X \in [-1; 1,5]$ , bzw. dem korrespondierenden Frameintervall  $\text{Frame} \in [490; 540]$ , welche dem zentralen Bildbereich mit geringerer Verzerrung in den Rohdaten entsprechen. Die Trajektorie des modellbasierten Ansatzes in Abbildung 4.22 zeigt einen glatteren Verlauf, mit offensichtlichen Ausreißern bei geringen Framezahlen, insbesondere die Entfernung betreffend. Im Gegensatz zum regelbasierten Ansatz entspricht die Anzahl assoziierter Messungen den Erwartungen. Die Position des Objekts unterscheidet sich bei beiden Ansätzen, insbesondere für geringe Framezahlen, erheblich.

Werden die absolute Geschwindigkeit und Entfernung des Objekts im Kamerakoordinatensystem mit der Grundwahrheit verglichen, so weisen beide Ansätze erhebliche Fehler auf. Abbildung 4.23 zeigt die Ergebnisse für den regelbasierten Ansatz, während die modellbasierte Methode in Abbildung 4.24 dargestellt ist. Für den regelbasierten Ansatz beträgt der RMSE der absoluten Geschwindigkeit 0,92 m/s und der RMSE der Objektentfernung 3,26 m. Der modellbasierte Ansatz weist einen höheren Fehler auf, wobei der RMSE der Geschwindigkeit 1,50 m/s und der Entfernung 3,05 m beträgt. Der Fehler für beide Signale, Geschwindigkeit und Entfernung, erreicht erst im oben erwähnten Framebereich die aus den TTC-Versuchen bekannten Größenbereiche. Dieses Verhalten lässt sich wiederum mit der Kalibrationsproblematik erklären. Erwähnenswert ist, dass für den ausgewerteten Versuch der Fehler des regelbasierten Ansatzes für beide Signale geringer ausfällt als für den modellbasierten Ansatz.

Basierend auf der bisher beschriebenen Leistung der Ansätze bezüglich der berechneten Positions- und Geschwindigkeitssignale sind entsprechend große Fehler in der PET zu erwarten. Abbildung 4.25 stellt die PET und deren Fehler bezogen auf

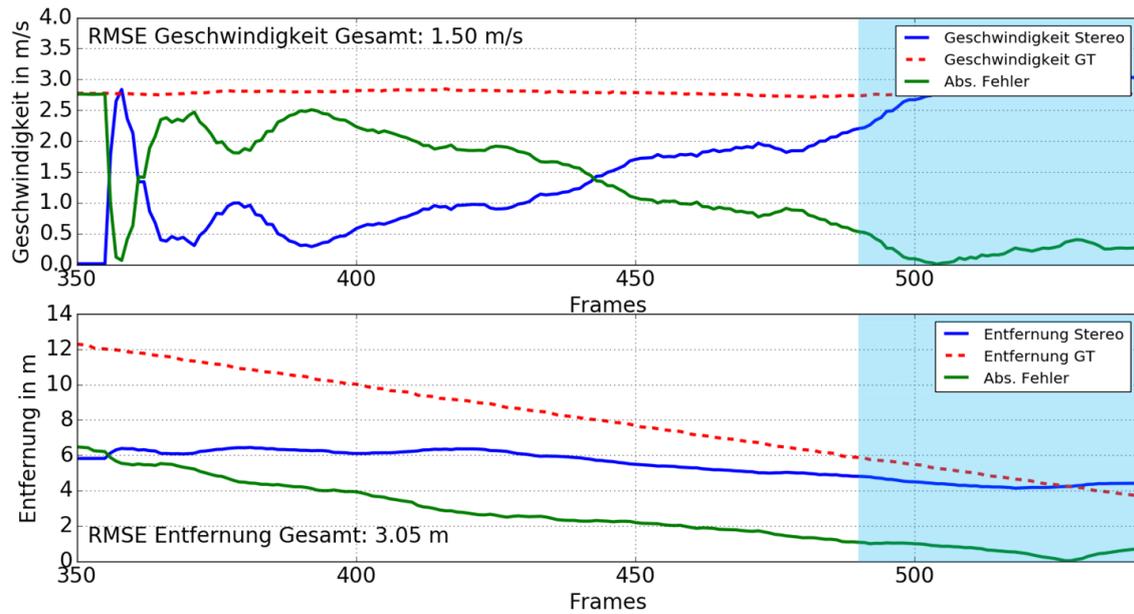


Abbildung 4.24

Geschwindigkeit (oben) und Entfernung (unten) des Objekts eines PET-Versuchs berechnet mit dem modellbasierten Ansatz. Die Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün). Die Frames des zentralen Bildbereichs sind blau markiert.

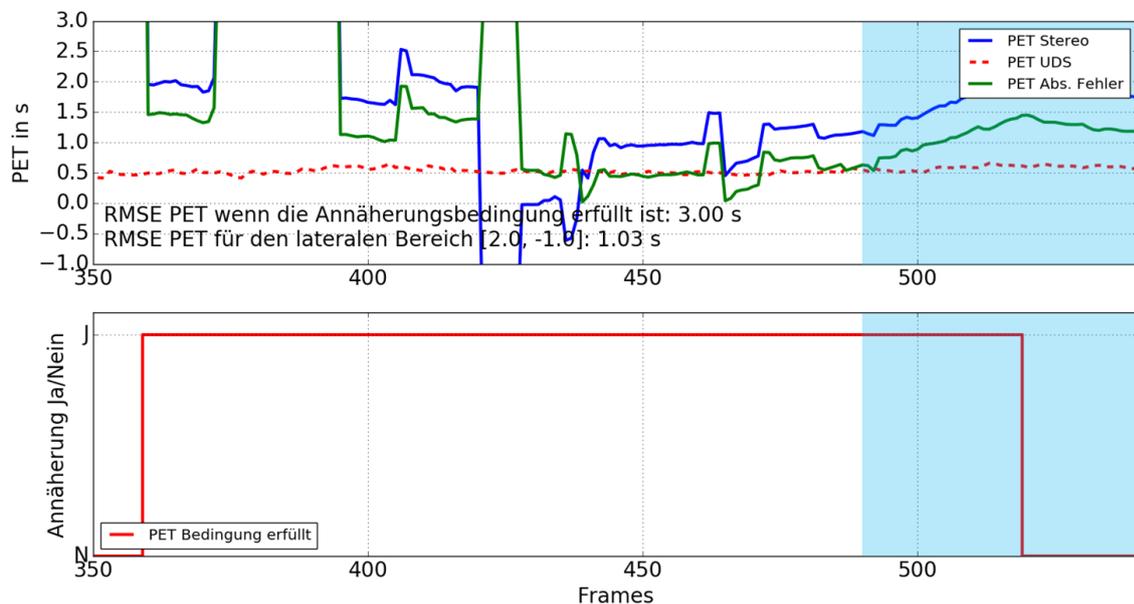
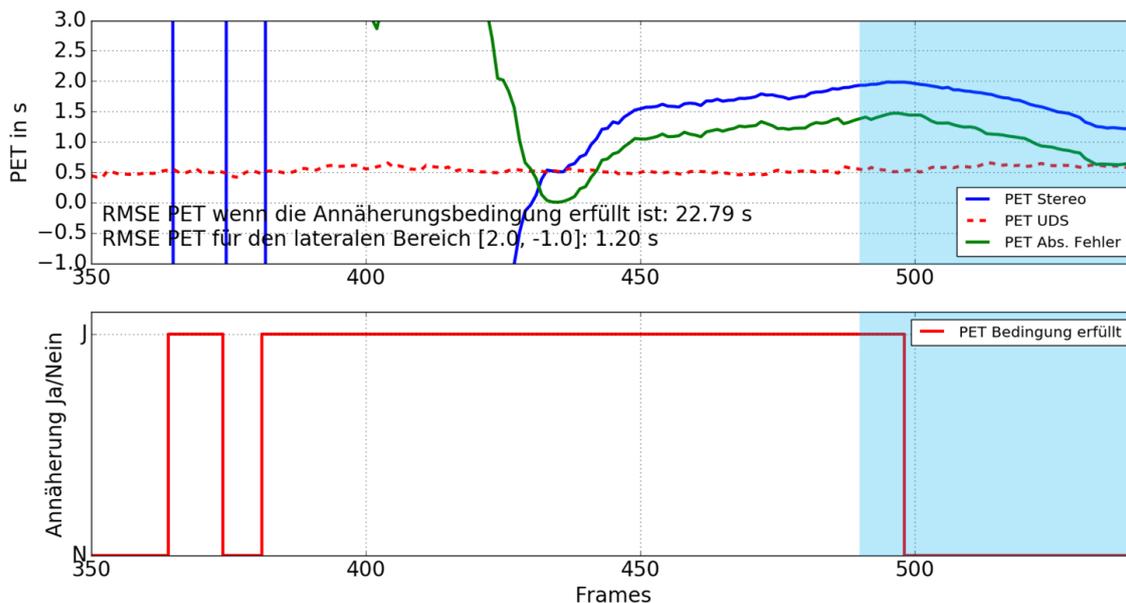


Abbildung 4.25

PET und Grundbedingung des Objekts berechnet mit dem regelbasierten Ansatz. Für die PET wird die Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün). Die Frames des zentralen Bildbereichs sind blau markiert.

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts. Aufgrund des Versuchsdesigns ist die Bedingung erfüllt solange sich die X-Werte im positiven Bereich befinden, was jedoch nicht durchgehend detektiert wird.

**Abbildung 4.26**

PET und Grundbedingung des Objekts berechnet mit dem modellbasierten Ansatz. Für die PET wird die Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün). Die Frames des zentralen Bildbereichs sind blau markiert.

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts. Aufgrund des Versuchsdesigns ist die Bedingung erfüllt solange sich die X-Werte im positiven Bereich befinden, was jedoch nicht durchgehend detektiert wird.

die Grundwahrheit für den regelbasierten Ansatz dar. Weiterhin wird gezeigt, für welche Zeitbereiche des Versuchs die Grundbedingung für die Berechnung der PET erfüllt ist. Abbildung 4.26 zeigt die Daten für den modellbasierten Ansatz. Für beide Ansätze ist die Bedingung nach Gleichung 3.29 ca. ab Frame 360 erfüllt und bleibt wie erwartet bis zum Kreuzen der Trajektorien bestehen. Während die Fehler innerhalb dieses Bereichs für den regelbasierten Ansatz mit 3,0s und den modellbasierten Ansatz mit 22,8s groß ausfallen, relativiert sich die Aussage, wenn die Berechnung auf einen zentralen Bildbereich limitiert wird. Für das Intervall  $X \in [-1,0; 2,0]$ , in welchem sich das Objekt nicht mehr unmittelbar am Bildrand befindet, ergibt sich ein reduzierter Fehler von 1,03s für den regelbasierten Ansatz und 1,20s für den modellbasierten Ansatz.

Die dargestellten Ergebnisse lassen sich zu großen Teilen auf die gesamte Versuchsreihe übertragen. Tabelle 4.8 zeigt den mittleren prozentualen Anteil an Kamerabildern, für welche eine Assoziation einer Messung zum ausgewerteten *Track* erfolgt ist, bezogen auf die Länge des *Tracks*. Die Ergebnisse zeigen eine hohe Assoziationsrate von 92% für die modellbasierte Variante, während mit der regelbasierten Objektdetektion 34% der Frames eines *Tracks* eine Assoziation aufweisen. Zusammen mit der geringen Streuung der Ergebnisse bestätigt dies die aufgestellte These der ungenügenden regelbasierten Objektdetektion, welche zum Verwerfen der Messung durch das *Gating* führt.

Die Werte des mittleren RMSE für die absolute Entfernung und Geschwindigkeit, welche in den Tabellen 4.9 und 4.10 aufgezeigt werden, bestätigen ebenfalls die Beobachtungen des oben dargestellten Versuchs für die gesamte Versuchsreihe.

**Tabelle 4.8**Anteil an Frames mit erfolgter Assoziation einer Messung zum *Track*.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Mittelwert Anteil Assoziationen	34	92	%
STD Anteil Assoziationen	9	6	%

**Tabelle 4.9**

Fehler der Entfernungsberechnung des PET-Szenarios für beide Ansätze.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Mittlerer RMSE Entfernung Gesamt	2,19	3,28	m
STD RMSE Entfernung Gesamt	0,58	0,54	m

Beide Werte weisen, bezogen auf die Gesamtversuchsdauer, einen im Vergleich zu den TTC-Versuchen großen Fehler auf, welcher nicht durch einzelne Ausreißer entsteht. Der offensichtlich systematische Fehler bestätigt einmal mehr die Annahme der Kalibrationsprobleme am Bildrand.

Dass die Leistung der Entfernungs- und Geschwindigkeitsbestimmung im Zentrum des Bildes erheblich zunimmt, zeigt die Auswertung bezüglich der berechneten PET in Tabelle 4.11. Der Mittelwert des RMSE der PET beträgt, über die gesamte Versuchsdauer betrachtet, 4,57 s für den regelbasierten und 105,09 s für den modellbasierten Ansatz. Der große Unterschied lässt sich auf die Sensitivität des Algorithmus bezüglich fehlerhafter Positionen und Geschwindigkeiten zurückführen, wobei die Entfernung gerade zu Beginn der Versuche für die modellbasierte Variante erheblich größere Fehler aufweist. Zu beachten ist dabei die Standardabweichung, welche ebenfalls groß ausfällt, aufgrund der geringen Versuchszahl jedoch keine starke Aussagekraft besitzt. Der mittlere RMSE zeigt sich stark reduziert, wenn die Auswertung auf den zentralen Bereich des Bildes beschränkt wird. Einerseits wird somit die Kalibrationsproblematik umgangen, andererseits befindet sich das Objekt in einem geringeren Abstand zur Kamera als zu Beginn des Versuchs, wodurch sich die Performanz der Entfernungsmessung generell und des regelbasierten Ansatzes aufgrund des nichtlinearen Zusammenhangs zwischen Disparität und Entfernung verbessert. Beide Ansätze weisen in diesem Bereich ähnliche Fehler mit Werten um 1 s mit ebenfalls vergleichbarer Standardabweichung auf, was die prinzipielle Funktionalität der Bestimmung der PET bestätigt. Beide Ansätze detektieren die Grundbedingung für die Berechnung der PET mit über 80 % Genauigkeit, wie in Tabelle 4.12 aufgelistet

**Tabelle 4.10**

Fehler der Geschwindigkeitsberechnung des PET-Szenarios für beide Ansätze.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Mittlerer RMSE Geschw. Gesamt	1,38	1,30	m/s
STD RMSE Geschw. Gesamt	0,48	0,11	m/s

**Tabelle 4.11**

Fehler der PET-Berechnung für beide Ansätze und einer quadratischen Begegnungszone mit 1 m Kantenlänge. Es werden die Werte für den Gesamtversuch und für die zentrale Bildregion mit lateralen Positionen zwischen  $-1$  m und  $2$  m angegeben.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Mittlerer RMSE PET Gesamt	4,57	105,09	s
STD RMSE PET Gesamt	3,91	117,65	s
Mittlerer RMSE PET Zentral	1,05	0,93	s
STD RMSE PET Zentral	0,35	0,27	s

**Tabelle 4.12**

Genauigkeit der PET-Bedingungsdetektion.

Beschreibung	Regelbasiert	Modellbasiert	Einheit
Genauigkeit Gesamt	84	82	%
STD Gesamt	3	11	%

wird. Dabei ist die Standardabweichung des regelbasierten Ansatzes mit 3 % erheblich geringer als der Wert des modellbasierten Ansatzes, welcher 11 % beträgt. Die Bereiche der ausbleibenden Detektion der Bedingung befinden sich ausschließlich zu Beginn des *Tracks*, welcher mit den bereits beschriebenen Problemen behaftet ist.

Zusammenfassend zeigen beide Ansätze eine ungenügende Leistung, wenn der Gesamtverlauf aller Versuche betrachtet wird. Die Gründe liegen dabei hauptsächlich in der fehlerhaften Kalibration im Randbereich sowie der stark fehlerbehafteten regelbasierten Objekterkennung durch weitere Objekte im Sichtfeld. Die Folge sind große Fehler in der Positions- und Geschwindigkeitsberechnung, welche sich im Fehler der PET widerspiegeln. Weiterhin weist der regelbasierte Ansatz nur geringe Assoziationsraten auf, was die Interpretation der Ergebnisse bestätigt. Eine Folge der ausbleibenden Assoziationen mit gültigen Messungen ist die reduzierte Möglichkeit zur Abbildung der Bewegungsdynamik des Objekts bzw. der Eigenbewegung der Kamera relativ zum Objekt, da diese Zeiträume durch das Prozessmodell der Objektverfolgung abgebildet werden müssen. Wird die Auswertung auf den zentralen Bildbereich begrenzt, weisen die mittleren Fehler der PET Werte im Bereich einer Sekunde auf, was in der zu erwartenden Größenordnung liegt. Generell ist somit die Funktionsfähigkeit der PET-Berechnung bewiesen. Die aufgezeigte Anfälligkeit der Algorithmen bieten Raum zur Optimierung der Kalibrationsroutine. Ob eine PET-Berechnung im Randbereich mit vergleichsweise großen Fehlern bereits ein Ausschlusskriterium für die Anwendung in einer realen NCS darstellt, bleibt dem Anwender überlassen. Da die Detektion im zentralen Bildbereich eine adäquate Leistung zeigt, können kritische Situationen, welche der Grundbedingung der PET-Berechnung genügen, detektiert werden, solange die Beteiligten nicht frühzeitig auf die Situation reagieren. Folglich können keine Events detektiert werden, welche schon in frühen Entstehungsstadien durch entsprechende Reaktionen entschärft werden. Je



**Abbildung 4.27**

Szenario: Fußgänger\*innen. Abgebildet sind vier Frames des aufgezeichneten Szenarios, zeitlich geordnet von links nach rechts. Das Szenario zeichnet sich durch mehrere abgestellte Fahrräder auf der linken Seite sowie mehrere Fußgänger\*innen auf der rechten Seite aus.

nach Ziel der Studie kann dies ein Ausschlusskriterium für beide Ansätze bedeuten, wenn die aufgedeckten Fehlerquellen vorliegen.

### 4.3.2 Reale Szenarien im Straßenverkehr

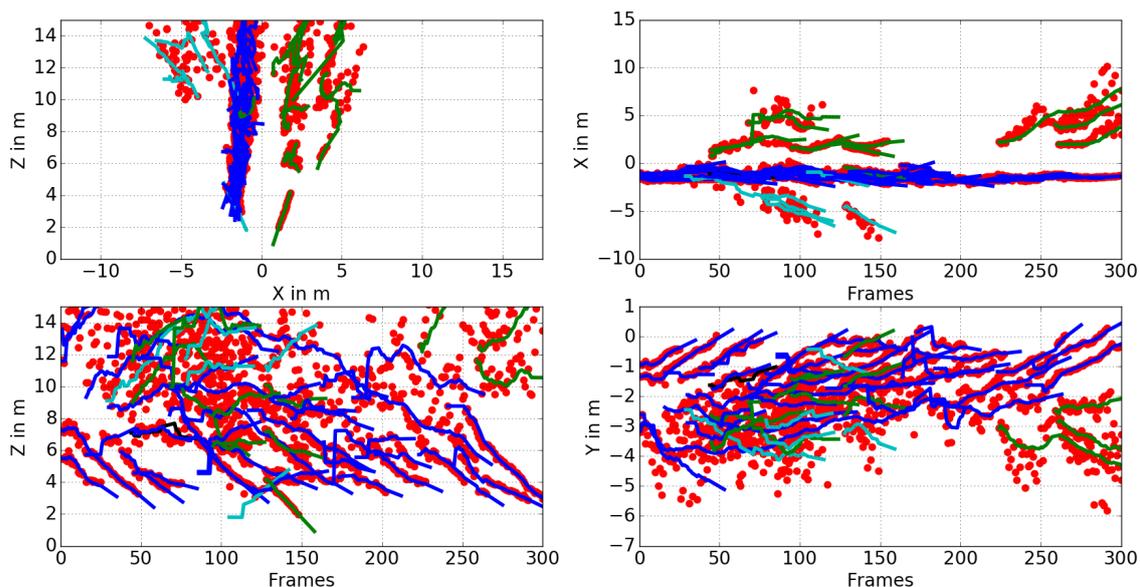
Nachdem in Kapitel 4.3.1 die Funktionalität des Programms anhand kontrollierter Szenarien mit verfügbarer Grundwahrheit analysiert wurde, soll im folgenden Abschnitt die Funktionalität mit Szenen aus dem realen Straßenverkehr geprüft werden. Wie bereits beschrieben existiert für diese Szenarien keine Grundwahrheit, weshalb die Funktionalität nur qualitativ mit Hilfe des zu sichtenden Videomaterials geschieht. Für die Analyse werden vier repräsentative Szenarien aus dem Berliner Straßenverkehr ausgewählt, welche die Möglichkeiten und Grenzen des Programms aufzeigen. Um einen Überblick über das Szenario zu bieten, wird jeweils eine Sequenz von vier Kamerabildern abgebildet, welche grob die erfassten Objekte und ihre Bewegungsrichtungen aufzeigen sollen.

Für jedes Szenario werden die bereits bekannten Darstellungen der Objekttrajektorien und die ermittelten TTC- sowie PET-Signale dargestellt, falls entsprechende Werte berechnet werden. Letztere werden nur für die Zeitbereiche abgebildet, in welchen die jeweilige Grundbedingung erfüllt ist. Da sich beide Bedingungen nicht gegenseitig ausschließen, kann für ein Objekt im selben Zeitbereich eine gültige TTC sowie PET vorliegen. Weiterhin gilt, dass die Bedingungen jeweils für mindesten zehn aufeinanderfolgende Frames gültig sein müssen, um berücksichtigt zu werden.

Da im modellbasierten Ansatz die Objekte klassifiziert werden, erfolgt die Assoziation zu einem *Track* klassenbasiert. Dies erlaubt nicht nur eine effektive Zuordnung von Messungen zu den jeweiligen *Tracks* einer Klasse, sondern auch die farbliche Kodierung der Ergebnisse. Entsprechend sind in den dargestellten Abbildungen des modellbasierten Ansatzes alle Trajektorien und TTC- bzw. PET-Signale nach Klasse farblich geordnet. Für den regelbasierten Ansatz ist dies nicht möglich, weshalb die Farben der einzelnen Objekte willkürlich gewählt wurden, allerdings weist jedes Objekt für alle Darstellungen eines Szenarios dieselbe Farbe auf.

#### Szenario: Fußgänger\*innen

Das erste Szenario, abgebildet in der Bildfolge in Abbildung 4.27, zeichnet sich durch eine große Anzahl abgestellter Fahrräder aus. Weiterhin gehen einige Fußgänger\*innen mit und entgegen der Fahrtrichtung des Versuchsfahrzeugs auf dem Gehweg, welcher sich auf der rechten Bildseite befindet. Die seitlichen Abstände zu den abgestellten Fahrrädern und manchen Personen sind gering und betragen teilweise weniger als 1 m.

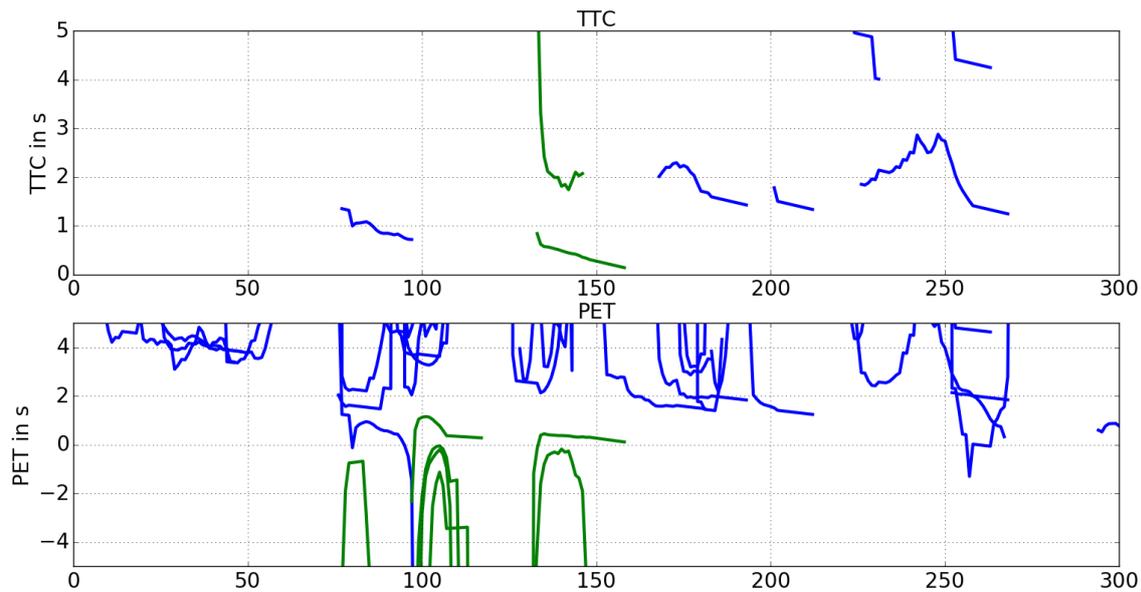


**Abbildung 4.28**

Szenario: Fußgänger\*innen. Trajektorie des modellbasierten Ansatzes. Dargestellte Klassen: Person (grün), Fahrrad (blau), Pkw (cyan), Motorrad (schwarz). Rote Punkte stellen die assoziierten Messungen dar.

Die verfolgten Trajektorien des modellbasierten Ansatzes werden in Abbildung 4.28 dargestellt. Im positiven Bereich der X-Koordinate, was der rechten Bildseite entspricht, befinden sich einige gehende Personen (grün), während links von der Trajektorie des Versuchsfahrrads über die gesamte Strecke Fahrräder (blau) abgebildet sind. In einer Entfernung über 10 m können am linken Bildrand Pkw (cyan) detektiert und verfolgt werden. Aufgrund der in Kapitel 4.3.1 erwähnten Unzulänglichkeiten bezüglich der Entzerrung am Bildrand, werden Trajektorien der Objekte in dieser Bildregion verzerrt dargestellt. Die Objekte zentral im Bild, d. h. die Fahrräder in mittlerer Entfernung sowie Personen mit geringem lateralem *Offset*, weisen die zu erwartenden Trajektorien auf. Diese können anhand des Videomaterials grob geprüft werden. Allerdings werden wiederholt *Tracks* von Personen aufgrund ausbleibender Assoziationen verloren. Die Sichtung der Innovationen des Kalman-Filters zeigt, dass das verwendete Bewegungsmodell der Objekte im Kamerakoordinatensystem die Schwankungen durch die Pendelbewegung des Fahrrads nicht immer abbilden kann. Eine Anpassung des Prozessrauschens kann Abhilfe schaffen, wenn das Modell nicht angepasst werden kann oder soll, wodurch der *Track* stärker den Messungen folgt und auch die Rate der Assoziationen erhöht wird. Dies wurde getestet, wird jedoch hier nicht weiter dargestellt und analysiert.

Die lateralen Entfernungen der Objekte, am besten sichtbar in der Darstellung der X-Koordinate über die Frames, entsprechen den Erwartungen, wobei die Werte von ca. 0,8 m – 1,0 m zu den Fahrrädern im Nahbereich nach visueller Prüfung realistisch sind. In größerer Entfernung streuen die Werte aufgrund der Pendelbewegung des Fahrrads. Die zeitlichen Verläufe der Objektentfernung zeigen für die abgestellten Fahrräder im Bereich mittlerer Entfernungen wie zu erwarten ähnliche Verläufe. Bei großen Entfernungen sind diese durch die Auflösung der Disparitätenkarte unrealistisch. Für geringe Entfernungen gibt es zwischen den Objekten geringe Abweichungen, welche wiederum auf die Kalibrationsproblematik am Bildrand zu-



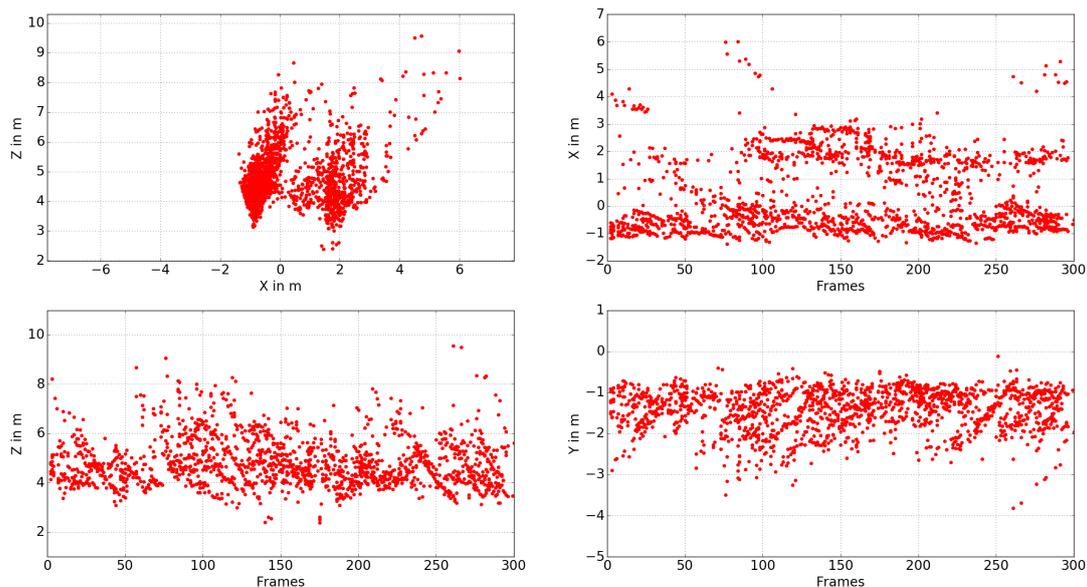
**Abbildung 4.29**

Szenario: Fußgänger\*innen. TTC- und PET-Signale des modellbasierten Ansatzes. Dargestellte Klassen: Person (grün), Fahrrad (blau).

rückzuführen sind. Die Höhe der Objekte, d. h. die Position in Y-Richtung, wird invertiert dargestellt und resultiert aus der Neigung der Kamera bezüglich der Fahrbahnoberfläche.

Bei genauer Betrachtung der farblich kodierten Trajektorien fallen zwei Klassifikationsfehler auf. Unter den Fahrrädern auf der linken Seite wird eine Person erkannt, welche in der Realität nicht vorhanden ist. Hier wurde bis zu einer Entfernung von ca. 9 m über einige Frames der orange Abfalleimer, welcher an einer Laterne befestigt ist, fälschlicherweise als Person klassifiziert. Bei reduzierter Entfernung tritt dieser Fehler nicht weiter auf, weshalb die Trajektorie nicht weiter fortgeführt wird. Eine weitere, schwer zu erkennende Fehlklassifikation eines Fahrrads tritt ebenfalls nur kurz auf, wobei dieses als Motorrad klassifiziert wurde.

Die Signale der TTC und PET spiegeln die bereits geschilderten Resultate wider und werden in Abbildung 4.29 dargestellt. Aufgrund der Pendelbewegung und der geringen Abstände zu den Fahrrädern sowie zu einigen Personen, werden einzeln Werte der TTC berechnet, welche sich in den Bereich unter 2 s erstrecken. Insbesondere die knappe Vorbeifahrt an einer Person im Framebereich von 130 bis 160 erscheint hierbei als relevant. Dies ist auf die definierte Objektbreite und die eigene Breite von jeweils 1 m zurückzuführen, wodurch in diesem Fall tatsächlich ein Kollisionskurs vorliegen würde. Die wahre Breite ist jedoch geringer, weshalb in der Realität offensichtlich keine Kollision vorliegt. Die dargestellte TTC betrifft hier nur einen kurzen Zeitbereich, da das Objekt von den bereits erwähnten Trackverlust betroffen ist. Wäre dies nicht der Fall, könnte der Verlauf der TTC über einen längeren Zeitraum nachvollzogen werden. Dennoch wird der kritische Abschnitt im Nahbereich detektiert. Es wird eine zweite passierte Person detektiert, welche ungefähr 2 m in Z-Richtung weiter entfernt zur ersten Person, ebenfalls nah an der Trajektorie des Versuchsfahrzeugs entlangläuft. Diese weist im selben Framebereich eine TTC von ca 2 s auf, bei weiterer Annäherung wird jedoch kein Kollisionskurs mehr detektiert,

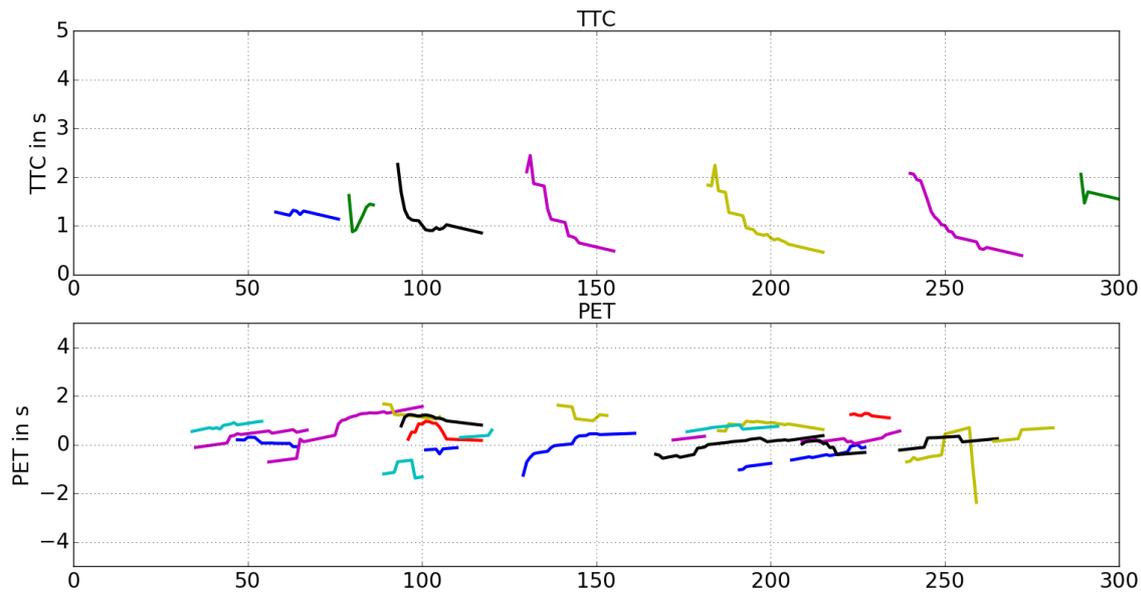
**Abbildung 4.30**

Szenario: Fußgänger\*innen. Alle Messungen des regelbasierten Ansatzes werden über das Szenario akkumuliert dargestellt. Jeder rote Punkt entspricht einem detektierten Objekt.

was offensichtlich an der größeren lateralen Entfernung von wenigen Dezimetern liegt.

Die Bedingung zur Berechnung einer PET wird von vielen Objekten erfüllt. Betrachtet man das zyklische Auftreten der Fahrräder in der Darstellung der PET, scheint dieser Effekt ebenfalls durch die Pendelbewegung bedingt. Durch diese Bewegung erhalten alle Trajektorien einen zusätzlichen Geschwindigkeitsanteil in X-Richtung, welcher um den Wert null oszilliert. Folglich ist je nach lateraler Position des Objekts die Bedingung zyklisch erfüllt. Es kann ein Schwellenwert eingeführt werden, um die Auswirkungen dieser Schwankung zu unterdrücken. Tests haben gezeigt, dass ein Schwellenwert der lateralen Geschwindigkeit von 1,5 m/s bei der Berechnung der Grundbedingung diese Effekte gänzlich unterdrückt. Nachteilig wirkt sich aus, dass Objekttrajektorien mit spitzem Winkel zur Trajektorie des Versuchsfahrrads häufig ebenfalls geringe laterale Geschwindigkeitsanteile aufweisen. Diese Fälle könnten in Abhängigkeit des gewählten Schwellenwerts nicht mehr als Event detektiert werden. Die beiden passierten Personen, welche als Kollisionspartner\*in detektiert wurden, erfüllen auch die PET Bedingung. Beide erzeugen PET-Werte nahe Null, was logisch erscheint, da die prädierten Trajektorien bei ausbleibender Kollision kurz nacheinander die Konfliktzone verlassen würden.

Mit dem regelbasierten Ansatz können keine einfach interpretierbaren Ergebnisse für das Szenario berechnet werden. Die regelbasierte Objektdetektion erzeugt, wie in Abbildung 4.30 dargestellt, viele Messungen. Die Tatsache, dass Objekte mit sichtbarer Ausdehnung in Z-Richtung dazu neigen, als mehrere Objekte detektiert zu werden, erhöht die Anzahl zusätzlich. Die Detektionen passen prinzipiell gut zum beschriebenen Szenario, es können eindeutig Häufungen in den Bereichen der Fahrräder und Fußgänger\*innen beobachtet werden. Die Messungen weisen jedoch eine erhebliche Streuung in der Position auf. Dieser Effekt konnte teilweise schon in den Versuchen aus Kapitel 4.3.1 beobachtet werden und tritt in der realen Anwendung verstärkt auf. Die Assoziation kann im regelbasierten Fall nicht nach Objektklassen



**Abbildung 4.31**

Szenario: Fußgänger\*innen. TTC- und PET-Signale des regelbasierten Ansatzes. Da keine Informationen über die Objektklassen vorliegen, werden die Daten der einzelnen Objekte mit zufällig ausgewählten Farben markiert. Die PET-Signale wurde bezüglich der Pendelbewegung des Versuchsfahrrads bereinigt.

getrennt werden, wodurch für eröffnete *Tracks* eine Vielzahl möglicher Messungen in Frage kommen. Aufgrund der bereits beschriebenen Unzulänglichkeiten des Bewegungsmodells erhöht sich die Rate falscher Assoziationen, und somit auch der Fehler in Trajektorien, TTC und PET der Objekte. Da jede nicht zugeordnete Messung einen weiteren *Track* eröffnet, existieren weitaus mehr verfolgte Objekte als im modellbasierten Fall und in der Realität existieren. Die Darstellung der resultierenden Trajektorien ist aufgrund dieser Tatsachen unübersichtlich und generiert keinen Mehrwert, weshalb darauf verzichtet wird. Für die Darstellung der PET wird zur Steigerung der Übersichtlichkeit der oben beschriebene Schwellenwert auf die Geschwindigkeit in X-Richtung implementiert. Dadurch können die aus der Pendelbewegung resultierenden Einträge der PET ignoriert werden. Abbildung 4.31 zeigt die TTC und PET für diesen Fall.

Sowohl TTC und PET weisen erheblich mehr als die erwarteten Einträge auf. Dies resultiert aus der angesprochenen Streuung der Objektposition sowie der ungenauen Bestimmung der ROI, welche wiederum die Position des Punktmodells der Detektion beeinflusst. Weiterhin führen falsche Assoziationen zu fehlerhaften Trajektorien und lösen somit ebenfalls die Bedingungen für die Konfliktindikatoren aus. Objekte, welche im modellbasierten Fall knapp passiert werden und folglich die Bedingung für die TTC nicht erfüllen, werden nun fälschlicherweise als Kollisionspartner\*in detektiert. Ebenso werden häufig Trajektorien detektiert, welche die Begegnungszone der PET durchqueren. In Anbetracht der vorliegenden Probleme des regelbasierten Ansatzes in realen Szenarien sowie der erheblichen Fehler des Ansatzes in den konstruierten Szenarien des Kapitels 4.3.1, wird auf eine weitere Auswertung der Szenarien verzichtet. Die folgenden Analysen beziehen sich nur auf den modellbasierten Ansatz.



**Abbildung 4.32**

Szenario: Kreuzender Radfahrer. Abgebildet sind vier Frames des aufgezeichneten Szenarios, zeitlich geordnet von links nach rechts. Das Szenario zeichnet sich durch einen von links nach rechts kreuzendem Radfahrer aus.

### **Szenario: Kreuzender Radfahrer**

Das zweite auszuwertende Szenario besteht aus einem von links nach rechts kreuzendem Fahrradfahrer. Abbildung 4.32 zeigt das Szenario als Bildsequenz. Abgesehen von einem Pkw am rechten Bildrand, welcher nur für wenige Frames sichtbar ist, und dem angesprochenen Fahrradfahrer, befinden sich keine weiteren Objekte im zu evaluierenden Sichtfeld. Durch die geneigte Verbauposition des Kamerasystems am Versuchsfahrrad befindet sich der Fahrradfahrer den Großteil des Szenarios am oberen Bildrand.

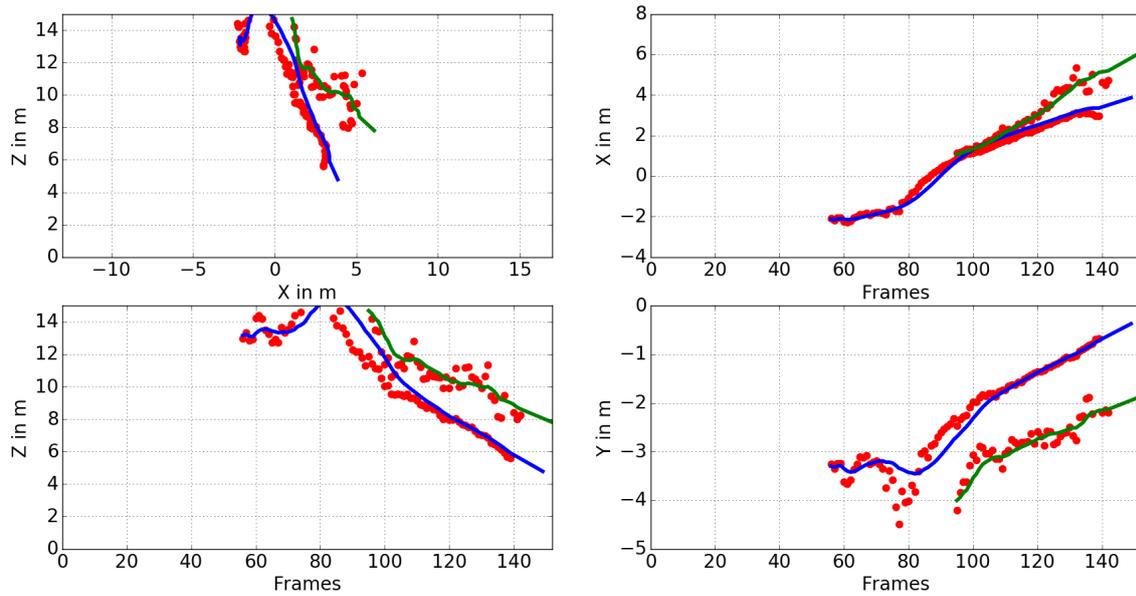
Die von der Kamera erfassten Objekte und die zugehörigen Trajektorien werden in Abbildung 4.33 dargestellt. Der zu Beginn sichtbare Pkw wird aufgrund der geringen Anzahl an konsekutiven Detektionen nicht als valider *Track* klassifiziert und folglich nicht dargestellt. Das Fahrrad sowie der Fahrer werden hingegen wie erwartet verfolgt und abgebildet. Es existiert eine Differenz zwischen beiden *Tracks*, vor allem bezüglich der Entfernung, welche in der Realität natürlich nicht auftritt. Beide Objekte befinden sich am oberen Bildrand, die Person ist dabei näher am Rand als das Fahrrad, und werden somit durch die Kalibrationsprobleme beeinflusst. Während die Trajektorie des Fahrrads glatt und realistisch aussieht, ist die stark springende Trajektorie der Person offensichtlich fehlerbehaftet.

Da sich kein verfolgtes Objekt auf einem Kollisionskurs befindet, wird keine TTC berechnet. Für das verfolgte Fahrrad wird jedoch eine PET ca. ab Frame 65 berechnet, wie in Abbildung 4.34 zu sehen ist. Ab dem Überschreiten des Nullpunkts in X-Richtung ist die Grundbedingung nicht mehr erfüllt. Die Werte des Signals erscheinen in Anbetracht der vorliegenden Videodaten valide. Für die detektierte Person wird keine PET ausgegeben, da der *Track* erst im positiven X-Bereich initialisiert wird und somit die Grundbedingung nicht erfüllt ist. Eine Ausrichtung der Kamera mit weniger Neigung zur Straßenoberfläche würde zu einer früheren, weniger fehlerbehafteten Detektion und Verfolgung der Person führen.

### **Szenario: Überholendes Fahrrad**

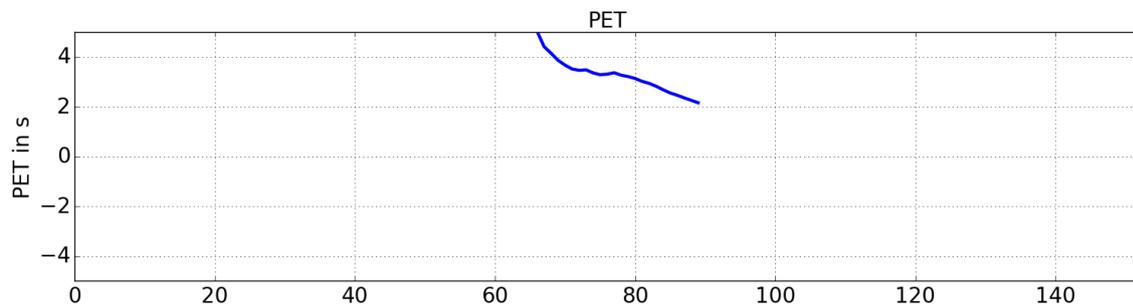
Im dritten Szenario wird ein im Stadtverkehr häufig auftretendes Überholmanöver analysiert. Abbildung 4.35 zeigt die zugehörige Bildfolge. Das Versuchsfahrrad fährt auf einem Schutzstreifen auf der Fahrbahn und wird dabei von einem Fahrrad links überholt. Direkt rechts neben dem Schutzstreifen, abgetrennt durch einen Bordstein, befinden sich ausgewiesene Parkplätze für Pkw, wovon einer im Sichtfeld belegt ist. Die Straße weist in Fahrtrichtung des Versuchsfahrrads zwei Spuren auf, wobei sich zu Beginn des Szenarios ein Pkw auf der linken Spur befindet.

Die Darstellung der Trajektorien in Abbildung 4.36 zeigt den Pkw auf der linken Spur mit einer Entfernung von 12 m bis 14 m. Dieser befindet sich am Bildrand



**Abbildung 4.33**

Szenario: Kreuzender Radfahrer. Trajektorie des modellbasierten Ansatzes. Dargestellte Klassen: Person (grün), Fahrrad (blau). Rote Punkte stellen die assoziierten Messungen dar.



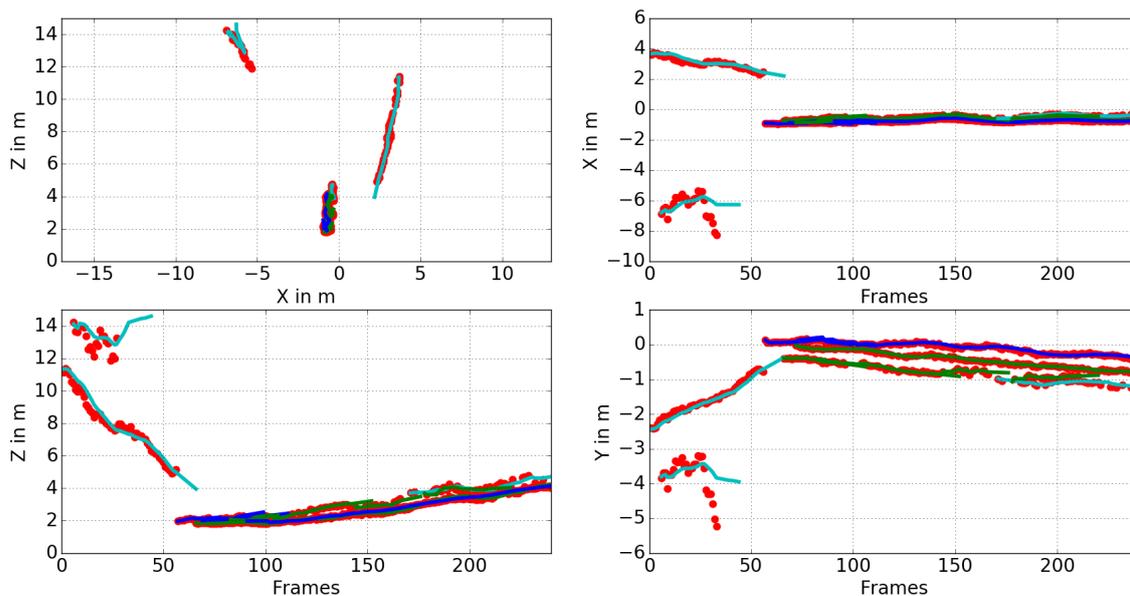
**Abbildung 4.34**

Szenario: Kreuzender Radfahrer. PET-Signale des modellbasierten Ansatzes. Dargestellte Klasse: Fahrrad (blau).



**Abbildung 4.35**

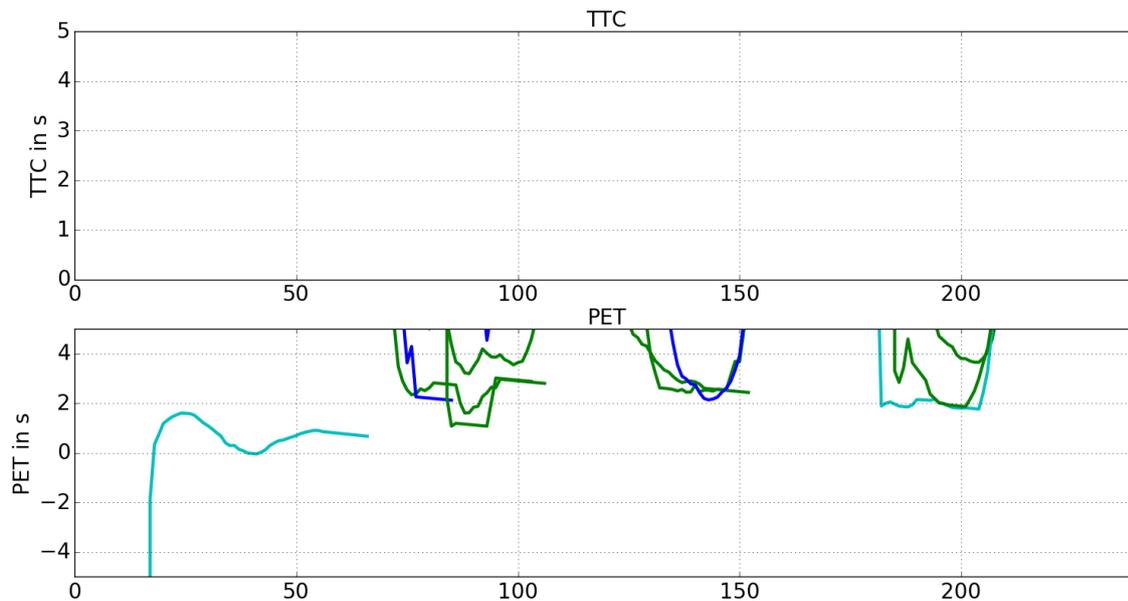
Szenario: Überholendes Fahrrad. Abgebildet sind vier Frames des aufgezeichneten Szenarios, zeitlich geordnet von links nach rechts. Das Szenario zeichnet sich durch ein links überholendes Fahrrad mit geringem Abstand aus.

**Abbildung 4.36**

Szenario: Überholendes Fahrrad. Trajektorie des modellbasierten Ansatzes. Dargestellte Klassen: Person (grün), Fahrrad (blau), Pkw (cyan). Rote Punkte stellen die assoziierten Messungen dar.

und weist daher zusätzlich zu den Einschränkungen durch die Entfernungsauflösung des Kamerasystems einen Fehler durch die Verzerrung am Bildrand auf, was insbesondere durch den zeitlichen Verlauf der Entfernung in  $Z$ -Richtung ersichtlich ist. Der zweite erkannte und verfolgte Pkw befindet sich auf der rechten Bildseite und kann dem parkenden Fahrzeug zugeordnet werden. Die laterale Entfernung des Pkw nimmt ab, da durch die Vorbeifahrt die Seite des Pkw sichtbar wird und einen kontinuierlich steigenden Anteil an der extrahierten Objektmaske einnimmt. Das Zentrum des Objekts, und dadurch die Position des *Tracks*, wandert somit vom Zentrum der Rückansicht zum Zentrum der Seitenansicht, was ca. 1 m *Offset* entspricht.

Der überholende Radfahrer ist durch zwei Objektklassen, der Person und dem Fahrrad, als Trajektorie abgebildet. Es ist ein lateraler *Offset* von ca. 1 m dargestellt, was in etwa dem Abstand im dargestellten Videomaterial entspricht. Die zu erwartende Pendelbewegung ist erkennbar. Die *Tracks* beginnen bei einer Entfernung in  $Z$ -Richtung von ca. 2 m und setzen sich bis zum Ende des Szenarios fort. Anhand der Darstellung der Höhe des Objekts ist leicht erkennbar, dass das Fahrrad und die Person jeweils mehrere parallel verlaufende *Tracks* erzeugen, wobei diejenigen der Person auch eine unterschiedliche Höhe aufweisen. Die Ursache lässt sich in den Bildbereichen der Person bzw. des Fahrrads finden, welche über Teile des Szenarios in mehrere sichtbare Bildflächen geteilt werden. Die Person trägt eine große Umhängetasche, welche den oberen Bereich des Torsos von den Beinen trennt. Ebenso wird das Fahrrad bei der Seitenansicht zu Beginn des *Tracks* durch die Beine der Person in zwei Bildbereiche getrennt. Die Objekterkennung behandelt diese Bereiche als unterschiedliche Messungen, wodurch auch unterschiedliche *Tracks* eröffnet werden. Prinzipiell ist dies kein Problem, solange die Objekte die nahezu gleiche Positionen und Geschwindigkeiten aufweisen, um gleichwertige Konfliktindikatoren zu erzeugen.



**Abbildung 4.37**

Szenario: Überholendes Fahrrad. PET-Signale des modellbasierten Ansatzes. Dargestellte Klassen: Person (grün), Fahrrad (blau), Pkw (cyan).

Ab Frame 170 zeigt sich ein weiterer *Track* im Bereich des Fahrradfahrers, der jedoch als Pkw klassifiziert ist. In diesem Fall ist dies jedoch keine Fehlklassifikation. Über einige Frames kann zwischen Oberkörper und rechten Arm des Fahrradfahrers, bzw. kurz darauf auch rechts neben dem Arm, der Pkw auf der linken Spur erkannt werden. Dieser wird von der Objekterkennung detektiert. Durch das *foreground fattening* der Disparitätenkarte, siehe Kapitel 3.4.2, weist in diesem kleinen Bildbereich die Disparität den Wert des Fahrradfahrers auf. Folglich wird eine Messung der Klasse Pkw mit der Position des Radfahrers erzeugt, woraus wiederum der entsprechende *Track* generiert wird.

Für das gesamte Szenario werden keine Kollisionskurse detektiert und entsprechend keine TTC-Signale berechnet. Abbildung 4.37 zeigt die berechneten PET-Signale. Der parkende Pkw weist aufgrund des sinkenden lateralen *Offsets* eine PET auf, welche jedoch als FP gewertet werden muss. Die Grundbedingung wird nur durch die Pendelbewegung des Versuchsfahrrads und, in diesem Fall maßgeblich, durch die verzerrte Trajektorie erfüllt. Der Fahrradfahrer, d. h. alle *Tracks* welche ihren Ursprung im Fahrrad bzw. der Person selbst haben, erfüllt durch die Pendelbewegung die Bedingung zyklisch. Durch die vorgegebene Gesamtbreite des Objekts und des Versuchsfahrrads von 1 m, und der Ausdehnung der Begegnungszone von ebenfalls 1 m, befindet sich das überholende Fahrrad an der Grenze ebendieser. Eine Berechnung der PET ist somit gültig. Bei Einführung des oben vorgeschlagenen Schwellenwerts für die Geschwindigkeit in X-Richtung besteht die Chance, dass der hier vorliegende Fall nicht detektiert wird.

## 4.4 Bewertung der Ergebnisse

Innerhalb des Kapitels 4 werden einzelne Teile der Prozesskette sowie die gesamte Funktionalität auf unterschiedliche Weisen analysiert. Kapitel 4.1 zeigt die Genau-

igkeit der Entfernungsberechnung gegenüber einer Referenz auf, wobei der Fokus auf der Vorverarbeitungskette und der Kalibration liegt. Der RMSE der Disparität weist einen Wert von 0,65 px auf, was als ausreichend zu bewerten ist. Allerdings wird während der Versuche nur ein zentral im Bild befindliches Objekt detektiert, wodurch Fehler der Entzerrung im Randbereich des Bilds nicht erfasst werden.

In Kapitel 4.2 wird eine Aussage über die Objekterkennung beider Ansätze im realen Verkehr getroffen. Die regelbasierte Methode wird anhand der „KITTI“-Daten analysiert. Der Datensatz liefert Kalibrationsdaten, wodurch die Vorverarbeitungskette nicht Gegenstand der Untersuchung ist. Der im Datensatz verfügbare LiDAR-Scanner wird als Referenzsensor für eine Tiefenkarte verwendet. Über eine Umrechnung anhand der Kamerakalibrationsdaten kann eine Disparitätenkarte als Referenz berechnet werden, welche ebenfalls die Anwendung des regelbasierten Ansatzes mit spezifischen Parametern zulässt. Ein Abgleich der Objekte aus beiden Quellen, Stereokamera und LiDAR, lässt eine Bewertung unter realen Bedingungen zu. Weiterhin können verschiedene Parameter auf ihre Sensitivität geprüft werden. Mit den gewählten Parametern erreicht der Ansatz für Genauigkeit und Sensitivität ca. 90 % bzw. 94 %. Die modellbasierte Objekterkennung hat sich in den verschiedenen Benchmarks als Stand der Technik etabliert, und übertrifft die menschlichen Fähigkeiten zur Objekterkennung. Die Ergebnisse beider Ansätze sind aufgrund ihrer unterschiedlichen Natur nicht vergleichbar, basierend auf den vorliegenden Daten sind beide grundsätzlich für die Objektdetektion geeignet.

Eine Analyse der Funktionsfähigkeit der gesamten Prozesskette wird in Kapitel 4.3.1 vorgestellt. Anhand konstruierter Szenarien zeigen beide Ansätze ihre Vor- und Nachteile, wobei durch Referenzsensorik eine Fehlerberechnung ermöglicht wird. Der modellbasierte Ansatz erzielt bezüglich der TTC erheblich bessere Ergebnisse als der regelbasierte Ansatz. Letzterer weist zu große Fehler auf, um eine automatisierte Auswertung von NCS-Daten umsetzen zu können. Die Assoziationsrate fällt im PET-Versuch für den regelbasierten Ansatz sehr gering aus, was zu problematischem Verhalten bei höherer Objektdynamik führen kann. Weiterhin wird ein kritisches Verhalten am Bildrand durch eine fehlerhafte Entzerrung festgestellt, was beide Ansätze betrifft. Im zentralen Bildbereich, welcher von diesem Fehler nicht merklich betroffen ist, entsprechen die PET Werte beider Ansätze jedoch den Erwartungen, da sich der Fehler bezogen auf die Grundwahrheit in einem ähnlichen Wertebereich wie der Fehler der TTC befindet. Werden die Ansätze mit realen Versuchsdaten aus dem Straßenverkehr geprüft, zeigt sich, dass der regelbasierte Ansatz nicht anwendbar ist. Fehlassoziationen und weitere spezifische Eigenschaften der regelbasierten Objektdetektion führen zu vielen, und vor allem offensichtlich falschen *Tracks*. Der modellbasierte Ansatz zeigt hingegen gute Ergebnisse, weist jedoch die aus Kapitel 4.3.1 bekannten Limitationen auf.

Die Einzelergebnisse lassen eine Bewertung der Prozesskette insgesamt, und der beiden Ansätze im speziellen zu. Weiterhin ergeben sich notwendige Anforderungen und mögliche Erweiterungen, welche zur Verwendung in einer realen NCS unabdingbar sind. Die Prüfung der Vorverarbeitung und der Kalibration des Systems hat oberste Priorität. Die Probleme durch eine fehlerbehaftete Entzerrung im Randbereich sind für eine reale Studie nicht hinnehmbar, aber auch vermeidbar.

Weiterhin ist trotz der positiv bewerteten Ergebnisse der reinen regelbasierten Objektdetektion der Ansatz im Realverkehr nicht verwendbar. Die hohe Streuung der Positionen sowie die Eigenschaft, je nach Entfernung und Objektansicht die Ob-

jekte in vielfache Detektionen aufzuspalten oder mehrere Objekte in eine Messung zu vereinen, führt zu vielfältigen Problemen. Die fehlende Möglichkeit zur Klassifizierung erhöht die Komplexität des *Trackings* und die Fehleranfälligkeit. Ein Beheben dieser Probleme kann nur mit erheblichen algorithmischen Anpassungen geschehen.

Die modellbasierte Objekterkennung funktioniert erheblich besser, es können weder in den konstruierten Szenarien noch in den realen Verkehrsdaten kritische Fehler entdeckt werden. Fehlklassifikationen kommen in den Daten zum Teil vor, sind jedoch unkritisch, da sie in größerer Entfernung stattfinden und bei Annäherung korrigiert werden. Die Forderung einer generischen Objekterkennung wird damit jedoch nicht erfüllt. Während des Studiendesigns einer NCS muss somit genau abgewägt werden, welche Objektklassen relevant sind.

Die Eigenbewegung des Versuchsfahrrads ist eine Fehlerquelle, welche erst in den realen Verkehrsdaten in Erscheinung tritt. Durch die Pendelbewegung werden Objekttrajektorien beeinflusst, was sich in den Konfliktindikatoren niederschlägt. Weiterhin bildet das Prozessmodell des Kalman-Filters diese Bewegung nicht passend ab, was teilweise zu verlorenen *Tracks* führt. Eine Anpassung des Prozessrauschens, des Modells oder gar eine Bereinigung der detektierten Objektpositionen um die Pendelbewegung sind mögliche Abhilfen.

Das implementierte Punktmodell umfasst per Definition nicht die Abmessungen der Objekte, was eine feste Vorgabe der Objektbreite zur Berechnung der Konfliktindikatoren erfordert. Diese kann im modellbasierten Fall klassenbasiert geschehen, was bereits ein Fortschritt zum regelbasierten Ansatz darstellt. Die große Varianz der Abmessungen von Objekten sorgt hierbei jedoch auch für erhebliche Unsicherheit. Eine bessere Variante ist das Schätzen der Objektbreite im Kalman-Filter, wobei das Punktmodell erweitert werden muss. Der modellbasierte Ansatz lässt dies ohne Anpassungen zu, da die Grenzen des Objekts pixelgenau bekannt sind. Der regelbasierte Ansatz, welcher die Objektgrenzen aus der Disparitätenkarte ermittelt, ist hierfür weniger geeignet. Die Unschärfe der Objektgrenzen durch eine fensterbasierte Berechnung der Disparitätenkarte sowie durch die schwellenwertbasierte Ermittlung der *u-disparity* ist schwer abschätzbar. Die bereits erörterten Fehler durch das Zusammenfassen von Objekten oder deren Aufteilung in mehrere Objekte kommt erschwerend hinzu. Eine Erweiterung der Schätzung der Abmessungen wäre eine Repräsentation der Detektionen als dreidimensionales Modell. Hierdurch können Trajektorienfehler durch die Änderung der Objektansicht, wie geschehen im letzten Szenario der realen Verkehrsdaten, vermieden werden.

Die möglichen Erweiterungen bzw. Anpassungen der einzelnen Prozesse werden in Kapitel 5 diskutiert, wobei die bereits erwähnten Punkte ausführlich erläutert und durch weitere Ansätze ergänzt werden.



---

# Kapitel 5

## Mögliche algorithmische Anpassungen und Erweiterungen

Das entwickelte Programm entspricht, zumindest für den modellbasierten Ansatz, den in Kapitel 3.2 definierten Anforderungen. Dies wird in Kapitel 4 ausführlich dargelegt, wobei auch einige notwendige Anpassungen oder Erweiterungen angesprochen werden, um die Funktionalität zu optimieren. Diese und weitere relevante Themen werden nun im Detail besprochen. Die Möglichkeit zur Verbesserung und stetigen Optimierung der Kamerakalibration wird in Kapitel 5.1 beschrieben. Die alternative Berechnung von Disparitäten- oder Tiefenkarten mit Hilfe von maschinellem Lernen ist im Fokus von Kapitel 5.2. Die Erweiterung des Messmodells vom Punktmodell zu einer Objektrepräsentation, welche dessen Ausdehnung in der Breite bzw. generell im Raum berücksichtigt, wird in Kapitel 5.3 dargestellt. Eine weitere Anpassung des Messmodells ist die Berücksichtigung von Geschwindigkeitssignalen, erläutert in Kapitel 5.4. Abschließend wird in Kapitel 5.5 die Schätzung der Eigengeschwindigkeit behandelt, welche zur Bereinigung der Messungen um die Pendelbewegung des Versuchsfahrrads verwendet werden kann.

### 5.1 Online-Kalibration

In der Auswertung der verschiedenen Versuche in Kapitel 4 ist die Kalibration des Kamerasystems das dominante Problem. Diese führt besonders im Randbereich der Bilder zu fehlerhaften Detektionen und den daraus abgeleiteten Signalen. Eine fehlerhafte Kalibration zu Beginn einer NCS kann die Auswertung der Datensätze des jeweiligen Sensors negativ beeinflussen und gar zur Unbrauchbarkeit der Daten führen. Ein weiteres Problem stellt bei Langzeitversuchen die Dekalibration des Kamerasystems dar, welche sich durch Vibrationen, Temperaturänderungen oder sonstige äußere Einflüsse einstellt [134]. Selbst eine perfekte initiale Kalibration ist mit großer Wahrscheinlichkeit nach einiger Zeit wertlos.

Als Abhilfe können Methoden verwendet werden, welche das System ohne spezifisches Kalibrierobjekt kalibrieren und die Parameter im Versuchsverlauf überwachen bzw. optimieren. Eine Methode, welche anhand eines willkürlichen Bilds die intrinsischen Parameter einer Kamera berechnet wird in [135] beschrieben, wobei ein neuronales Netz verwendet wird. Durch das sphärische Kameramodell ist der Ansatz für vielfältige Kameratypen verwendbar, welche nicht durch das hier verwendete Lochkameramodell mit radialer und tangentialer Verzerrung nach [38] abgebildet werden

können. Die Autoren zeigen, dass andere Methoden, welche ein Kalibrierobjekt verwenden, bessere Ergebnisse erzielen. Der Vorteil der Parameterschätzung anhand eines beliebigen Frames ist für NCS jedoch durchaus wertvoll.

Ein weiterer Ansatz wird in [134] vorgestellt. Hierbei werden die extrinsischen und intrinsischen Parameter eines Stereokamerasystems *online* geschätzt. Dabei kann mit willkürlichen Anfangswerten oder einer initialen Kalibrierung gestartet werden, während die Parameter mit jedem Frame weiter optimiert werden. Einzig die Basislänge des Stereokamerasystems ist nicht beobachtbar und kann somit nicht geschätzt werden. Wie im Analysetool verwendeten Ansatz findet das Lochkammermodell Anwendung, allerdings wird nur die radiale Verzerrung modelliert. Auf die tangentielle Verzerrung wird aufgrund des geringen Einflusses auf das Ergebnis verzichtet. Der auf Bündelblockausgleichung basierende Prozess kann nur bei einer vorliegenden Bewegung der Kamera umgesetzt werden, im Stand findet also keine Optimierung statt. Es wurde anhand von zwei unterschiedlichen Anwendungsfällen die Performanz gegenüber anderen Kalibrationsverfahren getestet. Zum einen wurden Disparitätenkarten erstellt, aber auch visuelle Odometrie für Szenarien mit Grundwahrheit berechnet. In beiden Fällen können Verbesserungen durch die vorgestellte Methode erreicht werden. Der Vorteil für die Anwendung in NCS liegt aber nicht nur in der Güte der Kalibration, sondern in der laufenden Optimierung ebendieser, um die stetige Dekalibration des Systems während der beliebig langen Versuchsdauer zu verhindern.

## 5.2 Tiefen- und Disparitätenkarten durch maschinelles Lernen

Die intensive Verwendung des maschinellen Lernens hält auch in der Berechnung von Tiefen- und Disparitätenkarten Einzug. Der im Analysetool verwendete Ansatz wurde gewählt, da dieser bereits erfolgreich in den Bereichen Robotik und ADAS eingesetzt wird. Zum Zeitpunkt dieser Entscheidung waren die in im nachfolgenden Abschnitt vorgestellten Veröffentlichungen nicht publiziert, weshalb sie nicht in Betracht gezogen werden konnten. Dennoch müssen diese Algorithmen für eine zukünftige Umsetzung evaluiert werden.

Wie in Kapitel 2 erwähnt existieren monokulare Tiefenkriterien, welche zur Bestimmung von Tiefeninformationen verwendet werden können. Folglich ist eine Stereokamera nicht zwingend notwendig, aber die naheliegende Lösung. In [33] wird beschrieben, dass bei monokularer Tiefenbestimmung Mehrdeutigkeiten aufgelöst werden müssen. Monokulare Tiefenkriterien, wie z. B. Objektgrößen oder Fluchtlinien, benötigen ein Szenenverständnis, um sie zu einer Tiefenkarte kombinieren zu können. Das Ergebnis ist eine relative Tiefenkarte, welche die Relationen zwischen einzelnen Objekten darstellt. Die globale Skalierung, d. h. die tatsächliche Tiefe der Objekte, bleibt jedoch weiterhin ungelöst. Ohne diese können keine sinnvollen Konfliktindikatoren berechnet werden ohne weitere Anpassungen vorzunehmen. Ist die Skalierung zeitlich stabil, wie in [136] beschrieben, und sind die skalierte Eigengeschwindigkeit sowie die skalierten Abmaße des Versuchsfahrrads bekannt, so können alle Konfliktindikatoren auch anhand der relativen Tiefenkarten bestimmt werden. Ein wertvoller Aspekt der Arbeit in [136] ist, dass das Modell auch mit einer unkalibrierten Kamera verwendet werden kann. Der Ansatz nutzt unüberwachtes Lernen

anhand von monokularen Videos ohne weitere Tiefeninformation, was einen großen Vorteil bezüglich der Verfügbarkeit von Trainingsdaten darstellt.

Neben den monokularen Ansätzen können auch rektifizierte Stereopaare als Eingang für Modelle zur Bestimmung von dichten Disparitätenkarten verwendet werden, analog zu den vorgestellten Methoden in Kapitel 3.4. Nach [137] werden durch maschinelles Lernen die Probleme der klassischen Ansätze entschärft. Diese sind Fehler bei texturschwachen oder spiegelnden Oberflächen sowie dünnen Strukturen oder repetitiven Mustern. Die Modelle verwenden hierbei Kontextinformationen. Die Windschutzscheibe eines Pkw dient als anschauliches Beispiel. Diese spiegelt in vielen Szenarien stark, wodurch die berechneten Disparitäten aus klassischen Methoden oft Fehler aufweisen. Durch semantische Informationen kann ein entsprechender Ansatz des maschinellen Lernens dies vermeiden. Anhand des „KITTI 2012“-Benchmarks [138], mit komplexen Szenarien aus dem realen Straßenverkehr, wird gezeigt das der mittlere Fehler über das gesamte Bild inklusive okkludierter Bereiche 0,7 px bei einer Dichte von 100 % beträgt. Hierbei sind viele der oben beschriebenen problematischen Flächen Teil des evaluierten Datensatzes. Im selben Benchmark erreicht der in Kapitel 3.4.2 ausgewählte Algorithmus eine mittlere Dichte von 86,5 % bei einem mittleren Disparitätenfehler von 2,0 px. Es existieren weitere veröffentlichte Modelle, wie z. B. [139], die ebenfalls rektifizierte Bildpaare verwenden und ähnliche Fehler bei teilweise stark unterschiedlichen Laufzeiten aufweisen. Die Ansätze des überwachten Lernens benötigen zum Training jedoch nicht nur rektifizierte Bildpaare, sondern auch die jeweilige Disparitätenkarte als Grundwahrheit. Das Erstellen solcher Datensätze bedarf einen hohen Aufwand an Sensorik und an Nachprozessierung, weshalb nur wenige existieren.

Die Verwendung von den beschriebenen Ansätzen kann, je nach Anforderung und Datenverfügbarkeit, einen erheblichen Vorteil gegenüber den klassischen Ansätzen darstellen. Kann die geforderte Funktionalität des Programms für einen spezifischen Anwendungsfall einer NCS auch mit den monokularen Ansätzen umgesetzt werden, so ist die Verwendbarkeit mit unkalibrierten Kameras ein großer Schritt in der Reduzierung des Aufwands zur Umsetzung der Studie.

## 5.3 Erweiterung des Messmodells um die Objektdimensionen

Das entwickelte Programm verwendet als Objektdarstellung ein Punktmodell. Per Definition beinhaltet dieses keine Objektdimensionen, weshalb für die Detektion von Kollisionskursen oder die Berechnung, wann ein Objekt die Begegnungszone verlässt, eine Objektbreite vorgegeben werden sollte. Für die modellbasierte Objekterkennung können klassenspezifische Objektbreiten definiert werden, der regelbasierte Ansatz lässt dies nicht zu. Prinzipiell funktionieren die Algorithmen auch ohne diese Vorgabe, die Ergebnisse werden dabei jedoch verfälscht. Ebenso trägt auch eine falsche Objektbreite zu fehlerhaften Ergebnissen bei. Folglich sollte zur Optimierung der Genauigkeit der Berechnungen die Objektbreite oder gar die gesamten dreidimensionalen Abmessungen des Objekts bestimmt werden.

Die Berechnung der Abmessungen schafft zusätzlich Abhilfe bei der Verzerrung von Trajektorien durch einen veränderlichen Blickwinkel auf ein Objekt, wie es in Kapitel 4.3.2 im Szenario des überholenden Fahrrads bei einem parkenden Pkw

beobachtet werden kann, welcher passiert wurde. Das Zentrum der beobachtbaren Oberfläche eines Objekts wird als Position des Punktmodells gewählt. Bei einer Ansicht von hinten, wie in Abbildung 4.35 im ersten Bild sichtbar, befindet sich die ermittelte Position des Pkw relativ zentral auf der Heckansicht. Wird der Pkw passiert, so ändert sich die Perspektive auf das Fahrzeug kontinuierlich und die Fahrzeugseite nimmt einen steigenden Anteil der beobachtbaren Oberfläche des Pkw ein. Damit ändert sich die Position des Punktmodells im Bildraum bezogen auf das gesamte Fahrzeug, da sich der Punkt im Zentrum der beobachteten Fläche befindet. Weiterhin ändert sich der Median der Disparität mit der Perspektive, da die Seitenfläche des Pkw sich in die Tiefe erstreckt und somit die Disparität in Richtung Fahrzeugfront abnimmt. Beide Effekte führen zu einer fehlerhaften Darstellung der Trajektorie.

Im folgenden Abschnitt soll gezeigt werden, wie die Breite des Objekts in das Messmodell aufgenommen werden kann. Im darauffolgenden Abschnitt werden Möglichkeiten zur Bestimmung von dreidimensionalen Abmessungen von Objekten und deren Darstellung durch *3D bounding boxes* behandelt.

### Erweiterung um die Objektbreite

Für jedes detektierte Objekt, unabhängig vom gewählten Ansatz, kann eine beobachtete Objektbreite ausgegeben werden. Für den regelbasierten Ansatz ergeben die linken und rechten Extremwerte der Konturen, welche aus der *u-disparity* extrahiert werden, die Objektgrenzen in der Bildebene. Analog können für Objektmasken des modellbasierten Ansatzes die Extremwerte bestimmt werden. Unter Verwendung der Reprojektionsmatrix  $\mathbf{Q}_R$  aus Gleichung 2.12 können die entsprechenden Punkte in Kamerakoordinaten berechnet und daraus die Objektbreite bestimmt werden. Da die Objekthöhe aktuell keinen Mehrwert für die Berechnung der Konfliktindikatoren hat, wird an dieser Stelle auf eine Berechnung verzichtet.

Um die gemessenen Objektbreiten zeitlich zu koppeln, macht es Sinn, die Breite in den Zustandsvektor aufzunehmen. Der Messvektor, der bisher nur die Position des Punktmodells in der Bildebene beinhaltet, kann direkt um die Breite erweitert werden. Die Beobachtungsmatrix  $\mathbf{H}$  sowie die Übergangsmatrix  $\mathbf{F}$  des Prozessmodells müssen entsprechend angepasst werden, wobei die Annahme einer konstanten Objektbreite zulässig ist. Perspektivische Änderungen der Objektbreite können durch das Prozessrauschen abgebildet werden.

### Berechnung einer 3D-Objektrepräsentation

Wie bereits beschrieben löst die Erweiterung des Objektmodells um die beobachtete Objektbreite nicht die fehlerhaften Trajektorien aufgrund der Perspektivenänderung. Hierfür müssen die Abmessungen des Objekts bzw. eines Ersatzmodells, wie z. B. eines Quaders, sowie dessen Orientierung bekannt sein. Damit lässt die detektierte Oberfläche und deren Disparitäten eine Berechnung des tatsächlichen, geometrischen Objektzentrums zu. Dieses kann wie bisher als Punkt verfolgt werden. Als zusätzliche Elemente des Zustandsvektors werden die Abmessungen des Ersatzmodells sowie dessen Orientierung geführt.

Betrachtet man die am besten bewerteten Einträge des „KITTI“-Benchmarks für 3D-Objektdetektion [138], zeigt sich die Dominanz der LiDAR-basierten Algorithmen. Auch [140] nennt die Dominanz dieses Sensortyps, welcher auch in Fusion mit anderen Sensoren wie Monokameras verwendet wird. Weiterhin wird erläutert,

dass insbesondere die monokulare 3D-Objektdetektion aktuell aufgrund der fehlenden Tiefeninformationen keine vergleichbare Leistung zeigen kann. Als Alternative gelten Ansätze basierend auf Stereobildern, welche sich von der Leistungsfähigkeit im Mittelfeld ansiedeln.

Da für die Inferenz der Modelle, d. h. für die Anwendung eines trainierten Modells, auf dem Versuchsfahrrad keine LiDAR-Daten zur Verfügung stehen, können die entsprechenden Ansätze für diesen Anwendungsfall nicht verwendet werden. Modelle, welche mit Mono- oder Stereokameras sowie LiDAR-Daten trainiert werden, für die Inferenz jedoch nur die Kamera benötigen, sind prinzipiell anwendbar. Dabei muss allerdings die Verfügbarkeit entsprechender Trainingsdatensätze berücksichtigt werden.

Ein auf Einzelbilder aus Monokameras basierender Ansatz wird in [141] vorgestellt. Die Gesamtarchitektur kann dabei in drei Bereiche gruppiert werden. Ein Netz für zweidimensionale Objektrepräsentation im Bildraum wird parallel durch ein Netz zur Schätzung einer dicht besetzten Disparitätenkarte ergänzt. Eine daraus ermittelte Punktwolke bildet mit den zweidimensionalen Objektvorschlägen die Basis für ein weiteres Netz zur Bestimmung der dreidimensionalen Abmessungen, Orientierung und Position der klassifizierten Objekte. Das Untermodell zur Schätzung der Disparitätenkarte ist dabei ersetzbar, z. B. durch stereobasierte Ansätze. Ein Test mit einem solchen zeigt erhebliche Verbesserungen der Ergebnisse, was die bereits erwähnten Defizite der monokularen Ansätze bestätigt. Stereobasierte Ansätze, wie *Stereo R-CNN* [142], weisen im Vergleich bessere Ergebnisse auf. *Stereo R-CNN* stellt dabei eine Erweiterung von *Faster R-CNN* für Stereobildpaare dar, wobei die in beiden Bildern detektierten Objekte direkt assoziiert werden. Aus diesen Zwischenergebnissen werden in weiteren Schichten des Netzes die dreidimensionalen Objektrepräsentationen ermittelt.

Pseudo-LiDAR beschreibt einen Ansatz, welcher aus Kamera- und Tiefendaten eine Punktwolke generiert, die einer LiDAR Punktwolke ähnelt [143] [144]. Als Quelle für die Tiefeninformation können stereobasierte Disparitätenkarten, aber auch Informationen aus monokularen Methoden verwendet werden. Mit Hilfe dieser Informationen kann jeder Pixel eines Kamerabildes in Weltkoordinaten projiziert werden. Durch die vergleichbaren Eigenschaften der Punktwolken sind Algorithmen, welche auf LiDAR-Daten entwickelt wurden, indirekt auch für Kameras nutzbar. [143] [144] erzielen mit diesem Ansatz im „KITTI“-Benchmark [138] bessere Ergebnisse als reine mono- bzw. stereobasierte Algorithmen. Es ist jedoch zu beachten, dass Fehler in der Tiefeninformation direkte Auswirkungen auf die Qualität der Punktwolke des Pseudo-LiDARs und somit der Objektrepräsentationen haben. Die erforderliche Güte der Kamerakalibration muss also für stereobasierte sowie Pseudo-LiDAR Ansätze gewährleistet sein.

Aufgrund der Vielzahl an Veröffentlichungen in dieser Domäne sowie dem für viele Veröffentlichungen offengelegten Quellcode, ist die Erweiterung um eine dreidimensionale Objektrepräsentation ratsam. Eine parallele Auswertung mit dem Ansatz zur Erweiterung des Messmodells um die Objektbreite kann als erster Validierungsschritt der Ergebnisse verwendet werden.

## 5.4 Erweiterung des Messmodells um Geschwindigkeitssignale

Das Messmodell der Objekte kann nicht nur um die Objektdimensionen oder Orientierung erweitert werden, sondern auch um Geschwindigkeitsanteile. Bezüglich der Initialisierung von *Tracks* sowie der Genauigkeit von Objektgeschwindigkeiten kann dies ein Vorteil sein. Insbesondere, da die Geschwindigkeitskomponenten der Objekte eine wichtige Rolle in der Berechnung der Konfliktindikatoren spielen. Unter der Bedingung, dass die Messungen auf Einzelbildern bzw. Stereopaaren basieren, ist keine direkte Messung von Geschwindigkeiten möglich. Die Verwendung zeitlich aufeinander folgender Bilder oder Stereopaare ermöglicht dies jedoch. Die Translation einzelner Pixel in der Bildebene werden durch den optischen Fluss beschrieben [37]. Eine direkte Ableitung der dreidimensionalen Geschwindigkeitskomponenten von Objekten ist aufgrund der fehlenden Tiefeninformationen nicht möglich. Das Pendant des optischen Flusses im dreidimensionalen Raum stellt der Szenenfluss (*scene flow*) dar, welcher aus dem optischen Fluss und Tiefeninformationen berechnet werden kann [145]. Das resultierende Vektorfeld beinhaltet Informationen über die geometrische Zusammensetzung der Szene sowie die darin befindlichen Einzelobjekte, welche als Starrkörper angenommen werden können. Die resultierende Geschwindigkeit ist dabei eine Überlagerung der Eigenbewegung des Fahrzeugs, dargestellt als Bewegung des statischen Hintergrunds und der Geschwindigkeiten der detektierten Objekte.

Ein Datensatz mit Grundwahrheit zum Trainieren und Bewerten von Modellen wird von [146] vorgestellt. Bei der Analyse der gelisteten Methoden zeigt sich meist eine hohe Laufzeit zur Berechnung des Ergebnisses für ein Bildpaar. Dabei werden oft mehrere Minuten benötigt, was eine Anwendung selbst in der vorliegenden *Offline*-Prozessierung quasi ausschließt. Für die zum Zeitpunkt der Veröffentlichung beste Methode im genannten Benchmark trifft dies nicht zu. Der Ansatz aus [147] benötigt für eine Bildpaar 0,75 s, wobei alle Teilschritte einberechnet sind. Als Datenbasis werden konsekutive Stereobildpaare benötigt, anhand derer drei Zwischenergebnisse berechnet werden. Dies sind die Disparitätenkarte, eine semantische Segmentierung des Bildraums und der optische Fluss, welche jeweils durch neuronale Netze berechnet werden. Im vorgestellten Analysetool werden für den modellbasierten Ansatz bereits zwei der drei Ergebnisse berechnet, nämlich die Disparitätenkarte sowie die semantische Segmentierung durch *Mask R-CNN*. Einzig der optische Fluss muss als weiteres Element berechnet werden. Durch die Verknüpfung der Zwischenergebnisse kann eine dreidimensionale Bewegung für jede Objektinstanz sowie den Hintergrund ausgegeben werden. Die Berechnung des Szenenflusses ist anfällig für große Translationen von Objekten bzw. des Eigenfahrzeugs, was in der Evaluation der erwähnten Daten teilweise beobachtet werden kann. Die verwendeten Kameras zeichnen Bilder mit einer Frequenz von 10 Hz auf, was nur einem Sechstel bzw. einem Drittel der Aufzeichnungsrate der hier verwendeten Kameras entspricht. Folglich sollten diese Effekte vermindert auftreten. Fehler aufgrund von texturlosen Bereichen im Bild, Verdeckungen etc. verhalten sich ähnlich, da der Datensatz ebenfalls den realen Straßenverkehr in verschiedenen Umgebungen widerspiegelt.

Da ein beträchtlicher Teil der benötigten Berechnungsschritte zur Ermittlung des Szenenflusses nach [147] bereits in das Programm implementiert sind, spricht nichts gegen die Aufnahme der Objektgeschwindigkeiten in das Messmodell. Die

angegebene Laufzeit pro Bildpaar kann durch Parallelisierung weiter optimiert werden, wodurch keine Nachteile in dieser Hinsicht zu erwarten sind. Die Vorteile in der Initialisierung der *Tracks* und der Genauigkeit der Geschwindigkeitssignale, welche sich in den Konfliktindikatoren widerspiegelt, sprechen ebenso für eine Anwendung.

## 5.5 Schätzung der Eigenbewegung zur Sensordatenbereinigung

Die Eigengeschwindigkeit des Versuchsfahrrads bzw. des Kamerasystems ist ein essenzielles Signal zur Auswertung von NCS. Im vorliegenden Analysetool wird sie zur Bestimmung der PET benötigt, um die Dauer bis zum Erreichen der Begegnungszone zu berechnen. Es können jedoch weitere Aussagen basierend auf der Geschwindigkeit der Proband\*innen getroffen werden, wie z. B. die Geschwindigkeitsverteilung bezogen auf die genutzte Infrastruktur oder die Häufigkeit und Schwere kritischer Situationen in Abhängigkeit der Geschwindigkeit. Generell ist eine Schätzung der Eigenbewegung mit hoher Güte ein wichtiger Aspekt der Auswertung.

Eine einfache Methode die Eigengeschwindigkeit zu erlangen, ist die Messung der Raddrehzahl, z. B. durch einen am Rad montierten Magneten in Verbindung mit einem Hall-Sensor, oder alternativ die Verwendung von GNSS-Daten. Letztere Option ist gerade im urbanen Raum fehleranfällig aufgrund von Mehrwegeausbreitungseffekten, d. h. der Reflektion der Satellitensignale an Objekten und deren Empfang auf verschiedenen Wegen. Eine reduzierte Satellitensichtbarkeit durch Bebauung oder Vegetation trägt ebenfalls zu fehlerhaften Messungen bei. Weiterhin spiegeln beide Varianten nicht die benötigte Dynamik des Fahrrads wider, um die in Kapitel 4.3 beschriebenen Pendelbewegungen zu detektieren. Die Verwendung eines INS, welches GNSS-Daten mit den Beschleunigungssignalen und Drehraten einer inertialen Messeinheit fusioniert, kann für diesen Zweck ebenfalls ausgeschlossen werden. Diese Systeme sind zwar in der Lage die Pendelbewegung zu detektieren, sind preislich jedoch nicht attraktiv und benötigen vergleichsweise viel Bauraum. Da die Pendelbewegung des Lenkers gemessen werden muss, dieser ist die Montagestelle für das Kamerasystem, müsste auch das INS dort installiert sein, was für die meisten Systeme ohne auffällige Anbauten kaum möglich ist.

Um die Bewegung des Kamerasystems gegenüber der statischen Umgebung zu ermitteln bietet es sich daher an, die Kameradaten selbst zu verwenden. Wie bei der Berechnung der Objektmessungen und -geschwindigkeit müssen hierbei Tiefeninformationen verwendet werden, um die richtige Skalierung der Signale zu erreichen. Der in Kapitel 5.3 vorgestellte Ansatz zur Berechnung des Szenenflusses, ein dreidimensionales Geschwindigkeitsfeld, berechnet die Eigenbewegung als die inverse Bewegung des Hintergrunds. Der Hintergrund in der Bildebene ist dabei durch die semantische Segmentierung bekannt. Ist dieser Ansatz implementiert, so wird die Eigenbewegung der Kamera automatisch berechnet. Dabei besitzt die Kamera als Starrkörper sechs Freiheitsgrade, jeweils drei translatorische und drei rotatorische. Die Berechnung der Eigenbewegung ist somit unabhängig vom Versuchsträger, ein Wissen über das verwendete Fahrrad wird nicht benötigt. Es existieren weitere stereobasierte Methoden, welche im „KITTI“-Benchmark [138] evaluiert werden. Da die Eigenbewegung aus dem in Kapitel 5.3 vorgestellten Modell nicht anhand dieses Benchmarks evaluiert wurde, ist ein Vergleich schwierig. Es werden in [147] ein

translatorischer Fehler von 0,09 m und ein rotatorischer Fehler von  $0,24^\circ$  pro 10 m zurückgelegten Weg angegeben. Die Evaluationsmetriken des „KITTI“-Benchmarks unterscheiden sich von diesen, da die Fehler über Strecken verschiedener Länge evaluiert und gemittelt werden, wobei die Mindestlänge 100 m beträgt. Die stereobasierte Methode, welche zum Zeitpunkt der Veröffentlichung die geringsten Fehler aufweist, ist [148]. Die Fehler des Ansatzes liegen in der Translation bei 0,65 % und der Rotation bei  $0,0014^\circ$ .

Um den Einfluss der erwähnten Pendelbewegung auf Objekttrajektorien, deren Geschwindigkeiten und vor allem auf die Konfliktindikatoren zu minimieren, können vielfältige Ansätze implementiert werden. Eine anschauliche Form ist die Verwendung eines lokalen, ortsfesten Koordinatensystems, in dem die Pose des Versuchsfahrrads, bzw. des Kamerasystems abgebildet wird. Die Pose, also Position und Orientierung der Kamera, ist durch die berechnete Eigenbewegung bekannt, wenn zu einem Zeitpunkt  $t_0$  die aktuelle Pose als der Ursprung des ortsfesten Koordinatensystems gewählt wird. Die im Kamerakoordinatensystem detektierten Objekte können anhand der Eigenbewegung in das ortsfeste Koordinatensystem transformiert werden. Das Kalman-Filter ist weiterhin mit den transformierten Objektmodellen verwendbar. Einzig die Berechnung der Konfliktindikatoren muss angepasst werden, da z. B. die Definition des Kollisionskurses und der TTC für eine Berechnung im Kamerakoordinatensystem gilt. Um den Einfluss der Pendelbewegung zu unterdrücken müssen diese im ortsfesten Koordinatensystem berechnet werden. Ein weiterer Vorteil durch die Darstellung der Objekte in diesem Koordinatensystem ist, dass die Prozessmodelle der Objekte die Pendelbewegung nicht mehr abbilden müssen. Das verwendete CV-Modell wird die Bewegung der Objekte entsprechend besser darstellen. Dies gilt ebenso für weitere in der Literatur bekannte Modelle [149], welche in Abhängigkeit der Objektklasse implementiert werden können.

Die Berechnung der Eigenbewegung ist ein essenzieller Teil der Auswertung einer NCS, und kann stereokamerabasiert umgesetzt werden. Dies entspricht den Vorgaben einer naturalistischen Studie, welche so wenig auffällige Sensorik wie möglich fordert. Die Performanz der aktuell verfügbaren, kamerabasierten Eigenbewegungsschätzung ist wahrscheinlich für die vorliegenden Zwecke ausreichend, eine Vorstudie kann dies klären. Da der vorgestellte Ansatz zur Berechnung des Szenenflusses bereits die Berechnung der Eigenbewegung beinhaltet, sind bei dessen Verwendung keine zusätzlichen Schritte notwendig. Die Definition eines ortsfesten Koordinatensystems, in dem ein *Track* dargestellt und dessen Konfliktindikatoren berechnet werden, ist eine Möglichkeit um die Einflüsse der als problematisch erkannten Pendelbewegungen zu bereinigen. Dieser Schritt optimiert nicht nur die Rate falsch-positiv detektierter kritischer Situationen, sondern stellt auch die Verwendbarkeit des Prozessmodells sicher. Die Umsetzung hat somit voraussichtlich einen großen Einfluss auf die Güte der gesamten Auswertung.

---

## Kapitel 6

# Weitere Analysemöglichkeiten für NCS-Daten

Der Fokus dieser Arbeit liegt auf der Detektion von Konflikten anhand berechneter Konfliktindikatoren, denn diese spielen eine erhebliche Rolle bei NCS mit Augenmerk auf Verkehrssicherheit. Für die ganzheitliche Analyse von NCS-Daten und die erweiterte Bewertung von kritischen Situationen müssen jedoch häufig weitere Informationen aus den Versuchsdaten extrahiert werden. Aus diesem Grund wurde in Kapitel 1.2 die Frage nach sinnvollen Erweiterungen des Analysetools in die Forschungsfragen aufgenommen. Dieses Kapitel behandelt potenzielle Erweiterungen und zeigt dabei die Möglichkeiten und Grenzen der automatisierten Kodierung von Daten und bereits detektierten kritischen Situationen auf. Dabei zeigt sich, dass selbst eine nicht vollständige erweiterte Beschreibung der detektierten kritischen Situationen einen Mehrwert gegenüber der rein manuellen Auswertung aufweist.

Das in [14] beschriebene Schema zur Kodierung von Fahrverhalten liefert ein ausführliches Beispiel, welche Informationen bei einer manuellen Auswertung extrahiert werden können. Dieses umfasst Informationen zur angebotenen und genutzten Infrastruktur sowie deren regelwidriger Nutzung. Weiterhin werden Kreuzungen, Einmündungen und Einfahrten in einen Kreisverkehr kodiert. Anstiege und Gefälle werden ebenfalls extrahiert. Ist eine freie Fahrt ohne Hindernisse möglich, d. h. beeinträchtigen keine stehenden oder bewegten Verkehrsteilnehmer\*innen, Gegenstände oder die Infrastruktur die Fahrt für mindestens zehn Sekunden, wird dies ebenfalls dokumentiert. Diese Informationen können durch Erweiterungen des Auswertungstools auch automatisiert bereitgestellt werden.

Zusätzlich werden in [14] die manuell extrahierten, kritischen Situationen mit weiteren Informationen verknüpft. Diese beinhalten neben dem Typ des Ereignisses, wie z. B. Konflikt mit oder ohne Fahrerreaktion, auch Informationen zur Art des Ereignisses. Dabei werden die Konflikte nach verschiedenen Szenarien klassifiziert, wie z. B. Abbiege- oder Kreuzungskonflikt. Auch Konflikte mit ruhendem und Längsverkehr bilden eigene Konfliktarten. Weitere Informationen, wie die Objektklasse des Beteiligten, dessen genutzte Infrastruktur sowie das Verhalten und die Reaktion der beiden Konfliktpartner\*innen vor und während des Konflikts werden gelistet. Dabei werden nicht alle Informationen durch spezifische Kodierungen abgebildet, sondern auch Freitext verwendet. Manche der kodierten Informationen basieren auf einer subjektiven Interpretation des Auswertungspersonals, was zu möglichen Verzerrungen der resultierenden Daten führen kann. Insbesondere Fehlinterpretationen, welche

zu einer fehlerhaften Klassifizierung der Situation führen, z. B. bezüglich der Fahrerreaktion, können das Gesamtergebnis verfälschen. Sind verlässliche automatisierte Methoden verfügbar, kann die auf das Auswertungspersonal bezogene Verzerrung der kodierten Daten umgangen werden.

Betrachtet man die detaillierte Kodierung der kritischen Situationen, so ist offensichtlich, dass dieser Detailgrad nicht gänzlich durch eine automatisierte Auswertung erreicht werden kann. Vor allem die im Freitext beinhalteten Informationen sind kaum zu detektieren. Informationen wie die fehlerhafte Blickführung der Konfliktpartner\*innen beim Abbiegen oder unachtsames Überqueren der Straße durch Fußgänger\*innen können schwer oder gar nicht automatisiert erfasst werden. Dies reduziert den Nutzen der vorliegenden Arbeit jedoch nicht. In Kapitel 1.2 wurde aufgezeigt, dass allein die automatisierte Detektion von Events einen erheblichen Vorteil gegenüber der rein manuellen Auswertung darstellt. Das nach Events gefilterte Datenvolumen beträgt nur noch einen Bruchteil der ursprünglichen Menge, wodurch der Aufwand einer händischen Sichtung und Kodierung des Materials stark reduziert wird. In Kapitel 1.2 wurde am Beispiel der Auswertung in [14] eine Reduktion der auszuwertenden Daten um den Faktor 1000 errechnet, wobei die Annahme getroffen wurde, dass jede detektierte kritische Situation ein manuell auszuwertendes Zeitfenster von 30 Sekunden ergibt. Dieser Faktor ist nicht auf andere Studien übertragbar. Für die in [23] und [24] präsentierten Daten, welche sich durch eine erheblich geringere Gesamtdauer auszeichnen, ergeben sich Faktoren von ca. 200 bzw. 600. Die Reduktion des Datenvolumens ist also immer noch erheblich. Ist für die Studie ein Detailgrad der Kodierung notwendig, der nicht durch die automatisierte Auswertung bereitgestellt werden kann, so besteht durch die Verwendung von Kameras weiterhin die Möglichkeit zur manuellen Analyse. Gerade die Interpretation des Verhaltens von anderen Verkehrsteilnehmer\*innen kann so umgesetzt werden. Die Reduktion des Datenvolumens ermöglicht diese Form der Auswertung auch für große Datensätze.

Es werden in diesem Kapitel Wege aufgezeigt, wie ein großer Teil der vorgestellten Informationen automatisiert bereitgestellt werden kann, um den Prozess der Datenauswertung weiter zu verkürzen und effizienter zu gestalten. In Kapitel 6.1 werden Methoden zur Bestimmung der verwendeten Infrastruktur vorgestellt. Weiterhin werden Methoden aufgezeigt, welche die verfügbare Infrastruktur detektieren und im Raum darstellen, wodurch eine Zuordnung von Objekten zu den einzelnen Instanzen möglich ist. Eine weitere wichtige Informationsquelle im Verkehrsraum sind Verkehrsschilder und Ampeln, deren Detektion in 6.2 erläutert wird. Eine zusätzliche Informationsquelle bietet der sogenannte Freiraum, welcher die unmittelbar befahrbare Umgebung abbildet. Eine detaillierte Erläuterung zu diesem Konzept findet sich in Kapitel 6.3. Weitere extrahierbare Informationen werden in Kapitel 6.4 zusammengefasst.

### 6.1 Fahrradinfrastrukturerkennung

Das Wissen über die genutzte und verfügbare Infrastruktur während eines detektierten Events, aber auch während der gesamten Versuchsdauer, lässt verschiedene Analysemöglichkeiten zu. Zum Beispiel kann die Anzahl der kritischen Situationen während der Nutzung eines Infrastrukturtyps auf die gefahrenen Kilometer normiert werden, was eine Aussage über die Effektivität der einzelnen Infrastrukturtypen be-

zogen auf die Fahrradsicherheit zulässt. Die Informationen über die genutzte und verfügbare Infrastruktur ermöglichen viele weitere Analysen. Sie können Aufschluss über fehlende oder schlecht geplante Fahrradinfrastruktur bieten oder auch Regelverstöße der Fahrradfahrer\*innen direkt aufdecken. Werden sicherheitskritische Elemente wie Kreuzungen, Einmündungen oder Kreisverkehre detektiert, lässt sich diese Auswertung erheblich erweitern.

Diese Informationen lassen sich bei einer manuellen Auswertung von Videodaten einfach extrahieren. Das Auswertungspersonal kann Verkehrsschilder, Straßenmarkierungen und die Trajektorien anderer Verkehrsteilnehmer\*innen interpretieren und das Videomaterial entsprechend kodieren. Der Zeitaufwand ist jedoch sehr hoch. Werden die Informationen nur für die automatisiert detektierten Events manuell extrahiert, so können keine Rückschlüsse über die gesamte Infrastrukturnutzung getroffen werden. Folglich ist es wichtig in Abhängigkeit der Studienziele zu definieren, welche Informationen nur für die kritischen Situationen erhoben werden müssen und welche kontinuierlich extrahiert werden sollen. So kann es beispielsweise Sinn machen kontinuierlich die verwendete Infrastruktur zu extrahieren, aber nur für kritische Situationen weitere Informationen, wie verfügbare Infrastruktur, Regelverstöße, Kreuzungen oder Einmündung u. s. w. manuell zu kodieren, insbesondere vor dem Hintergrund der teilweise komplexen Regelungen für spezifische Wegtypen. Dies erlaubt eine Analyse der gesamten Streckenanteile des Versuchs bezüglich der verwendeten Infrastruktur, ermöglicht aber auch eine detaillierte Analyse der Entstehung von kritischen Situationen durch die erweiterten Daten der manuellen Sichtung.

Theoretisch ist es möglich anhand von GNSS-Daten und einer geeigneten Karte die genutzte und vorhandene Infrastruktur zu extrahieren. In der Praxis gestaltet sich dies schwierig. Einerseits weisen GNSS-basierte Positionsdaten ohne weitere Korrektursignale nicht die benötigte Genauigkeit auf, um das Fahrrad bezogen auf die Infrastruktur lokalisieren zu können. Bei einem exemplarischen 1,5 m breiten Radweg und einer Positionsgenauigkeit größer  $\pm 1,0$  m kann sich das Fahrrad auch auf der parallel verlaufenden Straße oder dem Fußweg befinden. Die GNSS-basierten Positionsdaten weisen in urbanen Räumen oft Fehler aufgrund der Mehrwegeausbreitung der Satellitensignale auf, was eine Extraktion der Karteninformationen verwehrt. Entsprechende Karten sind mit genauen Positionsangaben der verfügbaren Infrastruktur nicht zwingend verfügbar bzw. aktuell.

Eine bessere Alternative zu dem GNSS-basierten Ansatz ist somit die, an der manuellen Sichtung angelehnte, Extraktion der Information aus den Videodaten. Durch maschinelles Lernen können Modelle trainiert werden, welche die visuellen Eigenschaften der verschiedenen Infrastrukturtypen erfassen und folglich auf die aufgezeichneten Versuchsdaten angewendet werden können. Zum Trainieren werden entsprechende Datensätze benötigt, welche die zu erkennenden Klassen abdecken. Betrachtet man die Vielfalt an möglichen Infrastrukturklassen in [14], zeigt sich, dass nicht alle davon durch einfach visuelle Eigenschaften voneinander trennbar sind. So sind beispielsweise gemeinsame Rad- und Gehwege durch ein entsprechendes Schild gekennzeichnet. Ist dieses jedoch nicht im Bild, lässt sich dieser Infrastrukturtyp nicht von einem reinen Gehweg unterscheiden. Für die Auswertung kann dies Folgen haben, denn das Befahren eines Gehwegs ist in Deutschland nach Straßenverkehrsordnung für erwachsene Fahrradfahrer\*innen nur gestattet, wenn ein Kind unter acht Jahren begleitet wird. Ein gemeinsamer Rad- und Gehweg ist für Erwachsene

immer freigegeben, aber nicht benutzungspflichtig. Bei der Entwicklung eines Modells zur automatisierten Detektion sollten diese schwer interpretierbaren Randfälle berücksichtigt werden, indem die einzelnen Klassen nach ihrer Abgrenzung durch visuelle Eigenschaften definiert werden. Eine Analyse der Daten bezüglich komplexerer Klassendefinitionen kann nur manuell erfolgen.

Offen zugängliche Datensätze zum Trainieren der Modelle sind schwer zu erlangen, insbesondere mit einer spezifischen Ausrichtung auf Fahrradinfrastruktur. Datensätze wie der in [129] präsentierte „Cityscapes“-Datensatz, welcher auf Pixelebene gelabelt ist, unterscheiden horizontale Flächen nur grob nach ihrer Kategorie. Im vorliegenden Beispiel sind dies die Klassen „Straße“, „Gehweg“, „Parkflächen“ und „Gleise“. [129] listet hierfür weitere vergleichbare Datensätze auf. Nicht frei zugängliche Datensätze, welche ebenfalls für das Training verwendet werden können, sind ausgewertete NCS-Daten. Da die Daten aus [14] zumindest für das Versuchsfahrrad nach den vorgestellten Kategorien kodiert sind, könnten diese für das Training bestimmter Modelle verwendet werden. Da hierbei jedoch keine Kodierung auf Pixelebene oder für einzelne Instanzen im Bildraum stattgefunden hat, ist die Anwendbarkeit eingeschränkt.

Um die Verwendung von bereits kodierten NCS-Daten zum Training eines Infrastrukturklassifikators zu prüfen, wurde eine Machbarkeitsstudie umgesetzt [150]. Aus drei Stunden Videomaterial aus dem Berliner Stadtverkehr werden jeweils zwei Bilder pro Sekunde extrahiert und nach der vom Versuchsfahrrad verwendeten Infrastruktur kodiert. Die gewählten Klassen sind hierbei „Straße“, „Schutzstreifen“ und „Radweg“. Die Daten werden im Verhältnis 80/20 in Trainings- und Validationsdaten geteilt und nur die untere Bildhälfte verwendet, da dort die visuellen Hinweise bezüglich des Infrastrukturtyps zu erwarten sind. Ein weiterer Datensatz wird zum Testen des Modells verwendet, wobei keine Überschneidung zu der Route des Trainingsdatensatzes auftritt, um die Übertragbarkeit des Trainings zu prüfen. Für den Testdatensatz sind Routen mit Unregelmäßigkeiten gewählt worden. Diese sind z. B. Baustellen mit alternativer Straßenmarkierung, Tramgleise und stark befahrene Verkehrsräume mit hoher Dichte an Pkw, Fahrrädern und Personen. All diese Unregelmäßigkeiten sind im Trainingsdatensatz nicht vorhanden. Das in [151] vorgestellte CNN zur Objekterkennung wurde für die Anzahl Klassen angepasst und verwendet, um dem jeweiligen Bild die befahrene Infrastruktur zuzuordnen. Durch das Training mit den vergleichsweise geringen Datenmengen kann eine Genauigkeit von ca. 90 % im Testdatensatz erreicht werden. Dabei passieren die meisten Fehlklassifikationen bei uneindeutigen Fahrbahnmarkierungen sowie bei Fahrradwegen. Dies lässt sich durch die große Varianz im Erscheinungsbild der Berliner Radwege erklären, welche neben der fehlenden Erweiterung und Ausbesserung der Radwege auch ein Relikt aus den Zeiten der Teilung in Ost und West ist. Dennoch zeigt diese Machbarkeitsstudie, dass mit einem Datensatz, welcher diese Varianz abbildet, durchaus zielführende Ergebnisse erreicht werden können. Eine Filterung über mehrere Frames kann die Genauigkeit noch verbessern. Mit diesem Konzept kann allerdings weder eine komplexere Strukturierung der Klassen nach dem Konzept von [14] umgesetzt werden, noch die verfügbare neben der verwendeten Infrastruktur ermittelt werden. Inwiefern somit ein Nutzen für eine spezifische NCS entsteht, muss in der Entwurfsphase festgelegt werden.

Der in [152] vorgestellte stereobasierte Algorithmus dient als ein Beispiel, welches die Ergebnisse der Machbarkeitsstudie um die vorhandene Infrastruktur erweitert.

Für das automatisierte Fahren als Anwendungsfall wird eine Darstellung der eigenen Spur und der benachbarten Spuren aus der Vogelperspektive geliefert. Die verschiedenen Spuren werden als die eigene, parallele Spuren oder Gegenverkehrsspuren klassifiziert. Hierfür werden mehrere Prozessschritte durchlaufen. Zuerst wird eine semantische Segmentierung der Szene vorgenommen, welche neben der Spurgeometrie auch die Semantik sowie eine Pseudowahrscheinlichkeit der Klassifizierung liefert. Mit Hilfe einer berechneten Disparitätenkarte wird eine 3D-Punktewolke erzeugt, welche die Basis für eine auf die Bodenebene des Fahrzeugs projizierte *Gridmap* ist. Diese unterteilt den Raum in einzelne Zellen. Die *Gridmap* kann über die Zeit akkumuliert werden, wofür jedoch eine ausreichend genaue Eigenbewegungsschätzung benötigt wird. Für das Training des Modells wurde der „Cityscapes“-Datensatz um die erwähnten Unterklassen der Fahrbahn erweitert, wobei an Kreuzungen abbiegende Spuren ebenfalls separat kodiert wurden. Alle Objektklassen, die in irgendeiner Form semantische Informationen über die Spuren beinhalten, wurden beibehalten, z. B. Verkehrsschilder, Ampeln oder Fahrzeuge. Andere statische Objekte wurden zu einer Klasse *static* zusammengefasst. Die präsentierten Ergebnisse zeigen, dass das Modell selbst in uneindeutigen Szenarien die Spuren gut erkennt. Die oben erwähnten Unregelmäßigkeiten, wie z. B. Baustellen, werden durch diese Umsetzung besser behandelt. Das vorgestellte Konzept ist für eine NCS nicht direkt anwendbar, da kein entsprechender Trainingsdatensatz mit semantischen Labels auf Pixelebene zur Verfügung steht, der die benötigten Infrastrukturtypen kodiert. Das Konzept an sich ist jedoch vielversprechend. Durch die Verwendung eines Stereokamerasystems ist die Grundvoraussetzung erfüllt. Ist das Modell auf die benötigte Infrastruktur trainiert, kann mit einer noch zu bestimmenden Sicherheit die verfügbare und verwendete Infrastruktur ermittelt werden. Durch die Repräsentation der Spuren im dreidimensionalen Raum können die detektierten Konfliktpartner\*innen den einzelnen Infrastrukturinstanzen zugeordnet werden. Erweist sich die Detektion von Kreuzungsbereichen, Einmündungen und Kreisverkehren ebenfalls als performant im Rahmen der Erwartungen, so ist das Problem der Infrastrukturerkennung gelöst.

Ob die Detektion der genutzten und verfügbaren Infrastruktur für eine NCS benötigt wird, ist vom jeweiligen Fokus der Studie abhängig. Müssen diese Daten nur für die detektierten Events erhoben werden, so ist eine manuelle Kodierung durchaus möglich. Die Reduktion der aufgezeichneten Daten auf einzelne Events lässt den Aufwand handhabbar werden. Eine durchgehende Kodierung des Datensatzes muss automatisiert geschehen. Für die genutzte Infrastruktur scheint dies möglich. Die vorgestellte Machbarkeitsstudie zeigt mit geringen Mitteln bereits erste Erfolge. Eine Optimierung durch Erweiterung der Trainingsdaten, welche theoretisch von bereits absolvierten NCS übernommen werden können, ist hierbei der Schlüssel zum Erfolg. Eine automatisierte Erfassung der verfügbaren Infrastruktur ist aufgrund der benötigten Form von Trainingsdaten zur semantischen Segmentierung der Szene aktuell kaum umsetzbar, obwohl es vielversprechende Ansätze aus dem Bereich des autonomen Fahrens gibt. Datensätze mit den benötigten Infrastrukturklassen für NCS sind nicht verfügbar und der Aufwand der Kodierung zum Erstellen eigener Datensätze ist zu groß. Generell muss bei einer automatisierten Detektion von einer gröberen Unterteilung der Infrastrukturtypen in Vergleich zu der vorgestellten Kodierung von [14] ausgegangen werden. Wie stark dies den Nutzen einschränkt ist wiederum von der spezifischen NCS abhängig.

## 6.2 Verkehrszeichen- und Ampelphasenerkennung

Die Erkennung von Verkehrszeichen und Ampeln ist prinzipiell nicht von der Erkennung anderer Objekte zu unterscheiden, weshalb die in Kapitel 3.5.3 beschriebenen Methoden auch hier Anwendung finden. Dies zeigt sich in der Abbildung 3.15, in welcher Ampeln mit ROIs markiert wurden. Für die Verkehrsampel- bzw. Ampelphasenerkennung und die Verkehrszeichenerkennung ergeben sich jedoch spezielle Anforderungen an die Objektklassen und somit auch an die Trainingsdaten. In der beschriebenen Implementierung von *Mask R-CNN*, welche mit dem „COCO“-Datensatz [116] trainiert wurde, wird zwischen unterschiedlichen Ampeln und deren Phase nicht unterschieden. Für die automatisierte Auswertung in NCS muss eine differenzierte Beschreibung verwendet werden. Nicht nur die vier Phasen der Ampel, sondern auch die Art der Ampel spielt eine Rolle, weshalb zwischen Fußgängerampeln, Fahrradampeln, Abbiegeampeln und normalen Ampeln unterschieden werden muss.

Es existieren verschieden Datensätze, welche auf die Erkennung von Verkehrsampeln ausgerichtet sind, wovon aber nicht alle die genannten Anforderungen erfüllen. So beinhaltet das in [153] beschriebene „Bosch Small Traffic Lights Dataset“ zwar die verschiedenen Ampelphasen in komplexen Szenen, aber nicht die Unterscheidung der verschiedenen Typen. Das „DriveU Traffic Light Dataset“ [154] beinhaltet Daten aus elf deutschen Städten und berücksichtigt die unterschiedlichen Typen und Phasen. Der Datensatz bietet auch Informationen über die Orientierung der Ampeln, d. h. unter welchem Winkel die Frontfläche beobachtet wird, was potenziell hilfreich sein kann, um diese der Spur des Versuchsfahrzeugs zuzuordnen. Beide Datensätze können prinzipiell für das Training von modellbasierten Algorithmen verwendet werden, aber nur der Datensatz in [154] beinhaltet die für NCS relevante Unterscheidung zwischen den Ampeltypen.

Ähnliches gilt für die Verkehrszeichenerkennung. Der verwendete Trainingsdatensatz muss die in der Region der NCS verwendeten Zeichen beinhalten. Hierbei gibt es eine Vielzahl an verschiedenen regionalen Datensätzen. Für Deutschland bietet sich die Verwendung des Datensatzes aus dem „German Traffic Sign Detection Benchmark“ [155] an. Internationale Datensätze sind im Netz ebenfalls frei verfügbar. Werden neue Verkehrszeichen eingeführt, deren Detektion für die Studie einen Mehrwert bietet, so müssen diese im Training berücksichtigt werden. Folglich muss der verwendete Datensatz um die entsprechenden Klassen erweitert werden.

Werden die Objekte erkannt, kann analog zur Detektion von Konfliktpartner\*innen die Position in Weltkoordinaten berechnet werden. Ist eine Repräsentation der Infrastruktur der unmittelbaren Umgebung in Weltkoordinaten verfügbar, wie in Kapitel 6.1 beschrieben, so können die Verkehrszeichen der jeweiligen Spur zugeordnet werden. Dadurch wird in der Theorie eine Kopplung von Konfliktpartner\*innen, Infrastruktur und Verkehrsregelung möglich. Eine Implementierung, welche diese Kopplung umsetzt, ist aktuell nicht bekannt.

## 6.3 Freiraumerkennung

Im Bereich des autonomen Fahrens ist der Freiraum als der befahrbare Raum vor dem Egofahrzeug definiert, in welchem keine Hindernisse zu Kollisionen führen [156]. Folglich wird der Raum, der von statischen Objekten wie Schildern, Zäunen, parkenden Pkw oder sonstiger Bebauung, aber auch dynamischen Objekten wie fahrenden Pkw, Fußgänger\*innen oder Fahrrädern eingenommen wird, nicht als frei deklariert. Überfahrbare Objekte, also auf der Fahrbahn liegende Objekte geringer Höhe, werden als Freiraum deklariert. Sie können aber eine Sonderstellung für die Pfadplanung im autonomen Fahren einnehmen, da evtl. die Geschwindigkeit verringert werden muss oder alternative Pfade bevorzugt werden. Die Überschneidung des Freiraums mit der genutzten Infrastruktur gibt eine Information über nötige Spur- bzw. Infrastrukturwechsel.

Für die Auswertung einer NCS ist der Freiraum in mehrfacher Hinsicht relevant. So können darüber Wechsel der Infrastruktur erklärt werden, aber auch Änderungen im Fahrerverhalten. Wird eine Umgebung durch viel Freiraum charakterisiert, welcher bei der gewählten Trajektorie keine Passagen mit geringem Abstand zu statischen oder dynamischen Objekten beinhaltet, so kann der\*die Proband\*in die Geschwindigkeit entsprechend des eigenen Sicherheitsempfindens höher wählen, als bei geringen Abständen mit wenig nutzbarem Freiraum. Da im vorgestellten Auswertungstool relevante Objekttypen bereits detektiert und analysiert werden, muss für die Bestimmung des Freiraums noch eine Repräsentation des restlichen, potenziell nicht nach Objekttyp klassifizierten Raumes gefunden werden. Diese trennt in Kombination mit den detektierten Objekten den freien vom besetzten Raum ab.

Die in Kapitel 3.5.3.2 beschriebene Projektion einer dreidimensionalen Szene auf eine Ebene wird auch für die Berechnung des Freiraums angewendet. In [157] wird die Verwendung der *v-disparity* zur Bestimmung der Straßenebene verwendet, die *u-disparity* findet Anwendung in der generischen Objektdetektion. Alle Pixel im Bildraum mit valider Disparität werden somit der Straße oder einem Objekt zugeordnet. Dieser Klassifizierung entsprechend kann der Bildraum, und über die Disparität auch der dreidimensionale Raum, in Freiraum und besetzten Raum unterteilt werden.

Ein weit verbreiteter Ansatz ist die Verwendung von Belegungsgittern. Hierbei wird das Umfeld auf den Boden projiziert und in Zellen spezifischer Auflösung geteilt. Anhand von Umfeldsensorik können diese Belegungsgitter befüllt werden. Ein stereobasierter Ansatz wird von [156] vorgestellt. Hierfür wird eine Disparitätenkarte erstellt und die in den dreidimensionalen Raum projizierten Pixel des Bildraums, unter Verwendung einer Funktion zur Berechnung der Belegungswahrscheinlichkeit einer Zelle, in das Belegungsgitter eingetragen. Die Belegungsgitter einzelner Stereobildpaare können über die Eigenbewegung zeitlich gekoppelt werden. Mit Hilfe des RANSAC-Algorithmus werden Messungen, welche zu einem dynamischen Objekt gehören, detektiert und für die Verwendung im Belegungsgitter ausgeschlossen. Ein Schwellenwert für die Belegungswahrscheinlichkeit lässt die Klassifizierung der einzelnen Zellen in freie und belegte Zellen zu. Alle Zellen zwischen dem Fahrzeug und der ersten belegten Zelle werden im Verbund als Freiraum definiert.

Werden in der Auswertung einer NCS bereits Algorithmen wie [152] zur Detektion der verfügbaren Infrastruktur verwendet, so werden bereits Teile der Prozesskette zur Berechnung des Freiraums berechnet. In diesem Fall wird eine semantisch ko-

dierte 3D-Punktwolke berechnet und in eine *Gridmap* eingepflegt. Die *Gridmap* und das Belegungsgitter sind sich dabei ähnlich, wobei letzteres eine Erweiterung um die Belegungswahrscheinlichkeit darstellt. Die Extraktion des Freiraums kann folglich mit wenig Berechnungsaufwand in den bestehenden Algorithmus eingepflegt werden. Ähnlich verhält es sich mit der Verwendung der in Kapitel 3 vorgestellten regelbasierten Objektdetektion, welche die *u-disparity* verwendet. Eine Berechnung des Freiraums nach [157] ist somit auch hier zum Teil abgeschlossen.

### 6.4 Weitere Anwendungen

Neben den bereits vorgestellten Erweiterungen der automatisierten Analyse der NCS-Daten existiert eine Vielzahl weiterer Möglichkeiten, welche potenziell Mehrwert generieren können. Da dies jedoch vom Fokus der eigentlichen Studie abhängig ist, können an dieser Stelle nicht alle Optionen gelistet werden. Dennoch werden in diesem Abschnitt einige interessante Möglichkeiten präsentiert.

Eine in mehrfacher Hinsicht interessante Information zur Auswertung von NCS-Daten ist das Wetter. So kann Regen zu Tropfen auf den Kameralinsen führen, was zu Verzerrungen des Bildes führt. Die Auswirkungen auf die automatisierte Auswertung kann dabei verschiedene Formen annehmen. So können z. B. kontinuierlich FPs in der Objekterkennung entstehen oder große Fehler in den Disparitätenkarten. Informationen über Regenwetter oder eine nasse Fahrbahn lassen eine Analyse und mögliche Behebung dieses Problems zu. Weiterhin hat das Wetter und die daraus resultierenden Fahrbahneigenschaften einen potenziellen Einfluss auf das Fahrerverhalten der Proband\*innen. Wird automatisiert eine Änderung des Fahrerverhaltens detektiert, so sollte das Wetter in die Bewertung einfließen. Wetterdaten werden von verschiedenen Anbietern im Internet bereitgestellt. Die Auswertung der NCS-Daten geschieht nicht direkt im Versuchsfahrrad, sondern zentralisiert und zeitlich versetzt auf Hardware der ausführenden Institution. Folglich können auch im Internet verfügbare Wetterdatenbanken verwendet werden, ohne zusätzliche Hardware zum Abrufen der Informationen am Fahrrad zu verbauen. Allerdings wird hierfür eine globale Position des Versuchsfahrrads benötigt, wodurch das Sensorset um einen GNSS Empfänger erweitert werden muss. Wetterdatenbanken bieten aber nicht zwingend Informationen über den aktuellen, lokalen Zustand des Wetters und der Fahrbahn während der Aufzeichnung. Folglich ist eine kamerabasierte Wettererkennung zusätzlich zu den Wetterdaten oder alleinstehend als Detektor eine sinnvolle Erweiterung. Eine auf ADAS ausgerichtete Implementierung wird von [158] vorgestellt. Dabei werden aus Einzelbildern histogrammbasierte Merkmale extrahiert. Als Klassifikator verwendet eine *Support Vector Machine* die Merkmale, um zwischen klarem Wetter sowie leichten und starken Regen zu unterscheiden. Eine Implementierung auf Basis eines CNNs stellt [159] vor, wobei die Wetterklassen um „Schneefall“, „Nebel“ und „bewölkt“ erweitert wurden. Eine Unterscheidung zwischen starkem und schwachem Regen findet nicht statt. In der Literatur finden sich weitere Ansätze, wovon nicht alle für bewegte Kameras ausgelegt sind.

Bezüglich der Analyse des Fahrerverhaltens ist nicht primär das Wetter, sondern der resultierende Reibwert für den Kontakt zwischen Reifen und Fahrbahn ausschlaggebend. Daher kann es Sinn machen direkt Informationen über den Reibwert anhand der Videodaten zu ermitteln. Nach [160] ist der Haupteinflussfaktor auf den Reibwert der Zustand der Straßenoberfläche bzw. die Zwischenschicht zwischen Rei-

fen und Fahrbahnbelag, weshalb bildbasierte Verfahren zumeist diese zu detektieren versuchen. In [161] wird hierfür ein CNN verwendet, welches die Zustände „trocken“, „nass“, „Matsch“ und „Schnee“ bzw. „Eis“ erkennt. Ein regelbasierter Algorithmus klassifiziert daraufhin den Reibwert als „niedrig“, „mittel“ oder „hoch“. Ein ähnliches Vorgehen zeigt [162], wobei drei Oberflächentypen, Asphalt, Kopfsteinpflaster und Schotter erkannt werden können. Diesen werden zusätzlich die Zustände „nass“ und „trocken“ zugewiesen. Der Reibwert wird in Abhängigkeit des Radschlupfs und der klassifizierten Fahrbahn aus Versuchsdaten extrahiert, um die Funktion eines Antiblockiersystems zu optimieren. Der tatsächliche Reibwert ist für eine automatisierte Auswertung von NCS-Daten allerdings nicht relevant, Informationen über die wahrnehmbaren Einflussfaktoren jedoch schon. Daher kann die Klassifikation des Fahrbahnzustands einen Mehrwert für die Interpretation des Fahrverhaltens darstellen.

Wie bereits beschrieben, können sich Regentropfen auf den Linsen des Kamerasystems ansammeln und die aufgezeichneten Bilder verzerren. Die Auswirkungen auf die ausgewerteten Daten können dabei schwerwiegend sein. In [163] wird dargelegt, dass die durch Regentropfen hervorgerufene Verzerrung nicht einem Rauschen gleicht, sondern das Licht sich entsprechend der Form des Tropfens bricht. Folglich kann dieses Verhalten modelliert werden und das Bild ohne den Einfluss der Tropfen rekonstruiert werden, wobei hierfür Methoden des maschinellen Lernens verwendet werden. Inwiefern dieser Vorgang die Leistungsfähigkeit des vorgestellten Analyse-tools beeinflusst ist schwer abzuschätzen. Allerdings ist bereits die Detektion von Regentropfen ein Mehrwert, wodurch die Versuchsdaten für weitere Untersuchungen oder sogar eine manuelle Auswertung markiert werden können. Ob wasserabweisende Beschichtungen der Linsen einen ähnlichen Mehrwert bieten ist fraglich, da auch geringe Mengen Flüssigkeit zu Verzerrungen führen und die Beschichtung keine sofortige Selbstreinigung garantieren kann. Eine funktionierende Detektion und entsprechende manuelle Auswertung dieser Fälle ist demnach besser als eine unsichere Lösung durch Beschichtungen. Ob eine Kombination beider Möglichkeiten zielführend ist, können Vorversuche klären. Werden durch die Beschichtung jedoch die Verzerrungseigenschaften der Regentropfen verändert, so muss das Modell neu trainiert werden.

Es sind weitere Anpassungen für die Aufbereitung der Videodaten vor der Anwendung der Prozesskette denkbar, wobei die Optionen mit Bedacht gewählt werden müssen. So bieten aktuelle Actionkameras oder Drittanbietersoftware oft Möglichkeiten zur Bildstabilisation oder eine Bereinigung des Neigungswinkels der Kamera bezüglich des Horizonts. Gerade ersteres ist verlockend, da die Kameras in Abhängigkeit des Untergrunds teilweise starken Vibrationen ausgesetzt sind. Kameraintern werden Drehraten und gegebenenfalls Beschleunigungen gemessen, um die Einzelbilder bezüglich ihres Versatzes ausrichten zu können. Auch der optische Fluss kann hierbei verwendet werden. Dieser Vorgang verändert die Ausrichtung der Kamerabilder von zwei oder mehreren Kameras zueinander, wodurch die Systeme nicht mehr zueinander kalibriert sind. Weiterhin passen die internen Kalibrierungsparameter nicht zu den neu ausgerichteten Bildern, da das Verzerrungsmodell auf das ursprüngliche Bild parametrisiert wurde. Die definierte Prozesskette ist nicht mehr nutzbar. Die mechanische Alternative, eine kardanische Lagerung (*gimbal*) des Kamerasystems, ist von der Funktionalität geeignet, um den Neigungswinkel auszugleichen und das Bild zu stabilisieren. Die Prozesskette ist damit auch nutzbar, allerdings ist

aufgrund der Größe des Systems und der zusätzlichen Kosten eine Anwendung im NCS-Bereich nicht ratsam.

---

# Kapitel 7

## Fazit

Die vorliegende Arbeit, welche in den Jahren von 2014 bis 2020 bearbeitet wurde, beschreibt die Möglichkeiten und Grenzen einer automatisierten Auswertung von NCS-Daten anhand einer prototypischen Implementierung eines Auswertungsprogramms. Da die bisher verwendete manuelle Auswertung in Bezug auf den benötigten zeitlichen Aufwand und die dadurch entstehenden Kosten das Versuchsdesign stark limitiert, ist eine Automatisierung von Teilen der Auswertung nötig. Mit einem Fokus auf Verkehrssicherheit wird hierfür das Umfeld des Versuchsfahrrads erfasst und interpretiert, um daraus Informationen über relevante Events zu erlangen. Die Markierung dieser Events reduziert das Datenvolumen für die manuelle Auswertung stark, wodurch diese vertretbar wird. Die Eventdetektion liefert zusätzliche abgeleitete Daten, welche zur weiteren Auswertung der NCS verwendet werden können. Die Methodik naturalistischer Versuche vermeidet die exzessive Verwendung von Sensorik, um die Datenerfassung so unaufdringlich wie möglich zu gestalten. Daraus ergeben sich strikte Anforderungen an das verwendete Sensorset und die Auswertung der Daten. Die Beschränkung auf ein Stereokamerasystem erlaubt die manuelle Interpretation der Versuchsdaten sowie eine automatisierte Eventdetektion bei niedrigen Anschaffungskosten. Eine Verwendung redundanter Sensorik wird aufgrund der Anforderungen der naturalistischen Methodik ausgeschlossen. Im Kontrast dazu stehen die Bedingungen für eine erfolgreiche Anwendung der automatisierten Auswertung, welche auf einer minimalen Anzahl ausbleibender Eventdetektionen basiert. Unter Verwendung des aktuellen Stands der Technik aus dem Bereich des maschinellen Sehens werden Konfliktindikatoren zur Detektion kritischer Situationen in NCS-Daten berechnet. Die prototypische Implementierung erfüllt die gestellten Anforderungen an die Eventdetektion und bietet eine Basis zur weiteren manuellen Interpretation der NCS-Daten, mit erheblich reduziertem zeitlichem Aufwand.

Das folgende Kapitel 7.1 fasst die Ergebnisse der Arbeit zusammen und beantwortet die Forschungsfragen. Abschließend wird in Kapitel 7.2 ein Fazit der Arbeit gezogen.

### 7.1 Beantwortung der Forschungsfragen

Das übergeordnete Ziel dieser Arbeit ist eine auf dem Fahrkontext basierende Eventdetektion. Diese erweitert die automatisierte fahrerreaktionsbasierte Eventdetektion und stellt somit einen komplementären Beitrag dar. Die Reduktion des auszuwertenden Datenvolumens durch die markierten Events sowie die Bereitstellung von ob-

jektiven, messbaren Kriterien stellt eine erhebliche Optimierung des Auswertungsprozesses dar. Um die Möglichkeiten, Grenzen und die Umsetzung für eine reale NCS zu beschreiben, wurden verschiedene Forschungsfragen definiert, welche nun beantwortet werden.

### **1. Wie kann ein Stereokamerasystem für NCS aussehen und wie beeinflusst die Auslegung die Ergebnisse?**

Zur Abbildung des Umfelds wurde ein Stereokamerasystem als der zu verwendende Sensor in dieser Arbeit definiert. Die Informationsdichte des Kamerabilds, die vergleichsweise niedrigen Kosten und die Möglichkeit die Daten auch manuell zu interpretieren waren hierfür die ausschlaggebenden Gründe. Ein Stereokamerasystem weist neben dem Sichtfeld drei Hauptparameter auf. Diese sind Brennweite, Basislänge und Auflösung. Basierend auf den Grundlagen der Stereovision wurden die Auswirkungen der Parametervariation des Sensors auf den Tiefenfehler, die Blindzone und den Disparitätsbereich dargelegt. Dieses Wissen wird zwingend für die Sensorauswahl bzw. Parameterdefinition bei der Umsetzung einer NCS mit Stereokameras benötigt. Ebenso ist die Prozesskette beschrieben worden, um aus den Rohdaten der Kamera eine dreidimensionale Szene rekonstruieren zu können. Dabei spielen insbesondere die Kamerakalibration sowie die Entzerrung und Rektifikation der Bilder eine bedeutende Rolle.

### **2. Wie lauten die Anforderungen an ein Auswertungstool?**

Um Anforderungen an ein Programm zur Detektion kritischer Situationen in NCS-Daten stellen zu können, wurden Studien betrachtet, die in Daten aus statischen Verkehrskameras automatisiert Events detektieren. Der Anwendungsbereich NCS musste zusätzlich von den Bereichen ADAS und AD abgetrennt werden, welche üblicherweise Redundanz in den Sensordaten aufweisen und sich somit auf Fusionskonzepte fokussieren. Mit diesen Informationen und den Erkenntnissen aus dem „FESTA V“, einer Prozessdefinition für Feldstudien, konnten die Anforderungen an das Auswertungstool definiert werden.

Als Eingangsdaten werden die rohen Videodaten sowie die benötigten Kalibrationsdaten verwendet. Der Verarbeitungsprozess beinhaltet eine Filterung der Daten, umgesetzt als die beschriebene Vorverarbeitungskette. Darauf folgt die Berechnung der Konfliktindikatoren als abgeleitete Daten. Diese können anhand von Schwellenwerten in die Kritikalität des Konflikts umgerechnet werden, wobei die Definition der Schwellenwerte Teil des jeweiligen Entwurfs der NCS ist. Es wurden beide resultierende Informationen, Konfliktindikatoren und Kritikalität, als Ausgangsdaten des Programms definiert. Für die Umsetzung einer validen Eventdetektion muss sichergestellt werden, dass nur eine minimale Anzahl an FNs auftritt, um das Gesamtergebnis einer NCS nicht zu verfälschen. Weiterhin reduzieren FPs den Nutzen der Eventdetektion, da die folgende manuelle Auswertung des Events ergebnislos bleibt. Die Festlegung einer gültigen FN- bzw. FP-Rate obliegt dabei dem Anwender und unterscheidet sich je nach NCS.

### **3. Wie kann eine Implementierung auf Basis des aktuellen Stands der Technik im Bereich maschinelles Sehen gestaltet werden?**

Anhand der definierten Anforderungen und Schritte der Auswertung konnte ein generalisierter Workflow definiert werden, welcher das Programm beschreibt. Dieser

besteht aus Datenvorverarbeitung, Objekterkennung, Objektverfolgung und der Berechnung der Konfliktindikatoren. Diese Abfolge der Teilprozesse orientiert sich zum einen an den Implementierungen für die Auswertung statischer Kameradaten sowie an den bekannten Umsetzungen im ADAS- und AD-Bereich.

Für jeden Teil dieses Workflows wurde der Stand der Technik sondiert und Algorithmen ausgewählt, welche die definierten Anforderungen unter den gegebenen Randbedingungen einer NCS erfüllen. Einige Fortschritte aus den einzelnen Bereichen, welche erst im fortgeschrittenem Bearbeitungszeitraum publiziert wurden, konnten nicht mehr berücksichtigt werden. Dies betrifft z. B. die dreidimensionale Objektrepräsentation durch Pseudo-LiDAR oder die Schätzung der Eigenbewegung durch den Szenenfluss. Diese Arbeiten wurden als potenzielle Anpassungen und Erweiterungen des Analyseprogramms vorgestellt. Da die Objekterkennung bzw. Objektdetektion den kritischen Baustein dieses Auswertungstools darstellt, wurden zwei unterschiedliche Varianten umgesetzt. Die regelbasierte Objektdetektion extrahiert generische Objekte aus der Disparitätenkarte und baut auf Annahmen zu Form, Lage und Größe der Objekte im Raum. Die alternative Variante basiert auf maschinellem Lernen, erkennt zuvor trainierte Objektklassen in den Kamerabildern und segmentiert diese pixelgenau in der Bildebene. Beide Varianten weisen dieselben Schnittstellen auf und konnten dadurch in das Programm implementiert werden, ohne Änderungen an den anderen Komponenten vornehmen zu müssen. Die aus den Bilddaten extrahierten Objekte werden durch ein erweitertes Kalman-Filter verfolgt, wodurch die einzelnen Objektdetektionen zeitlich gekoppelt werden. Die resultierenden *Tracks* bilden die Basis für die Berechnung der Konfliktindikatoren, welche anhand der Position und Geschwindigkeit der Objekte in Weltkoordinaten berechnet werden können.

Um den Erkenntnissen der Funktionsanalyse Rechnung zu tragen, wurden Erweiterungen bzw. Anpassungen von einigen Elementen des umgesetzten Workflows vorgestellt. Dies betrifft neben dem Kalibrationsprozess auch die Berechnung von Disparitätenkarten und der Eigenbewegung des Versuchsfahrrads bzw. der Kamera. Die Repräsentation der Objektdetektionen sollte von einem Punktmodell zu einem räumlichen Modell erweitert werden, welches auch Informationen der Objektgeschwindigkeit beinhalten kann. Die Erweiterungen machen stark von maschinellem Lernen gebrauch.

#### **4. Wie gut ist die Erkennungsrate relevanter Objekte?**

Die Detektion von Objekten ist für die beschriebene Anwendung ein zentrales Element. Nur detektierte Objekte können verfolgt und ein potenzieller Konflikt somit erkannt werden. Die verwendeten Ansätze unterscheiden sich stark, weshalb sie nicht direkt miteinander verglichen werden können.

Der regelbasierte Ansatz detektiert generische Objekte über eine Projektion der Disparitätenkarte auf eine Ebene. Unter der Annahme, Objekte durch annähernd vertikale Flächen repräsentieren zu können, werden aus dieser Projektion Objekte anhand von Schwellenwerten für die minimale Objekthöhe und -breite im Bildraum extrahiert. Für einen öffentlichen Datensatz mit realen Verkehrsszenen wurde eine Grundwahrheit der Objektdetektion mit Hilfe von Daten eines LiDAR-Scanners erstellt und mit den Ergebnissen des regelbasierten Algorithmus verglichen. Dabei wurde auch der Einfluss der Schwellenwerte auf die Ergebnisse analysiert. Beide Werte dienen als Filter zum Unterdrücken des Rauschens in der Disparitätenkarte,

was zur Folge hat, dass eine große Mindesthöhe und -breite zu hohen Genauigkeiten in der Objekterkennung führt, dafür aber viele relevante Objekte nicht detektiert werden. Die tatsächlichen Schwellenwerte müssen für jede NCS definiert werden und sind von den Mindesthöhen und -breiten der zu erfassenden Objekte abhängig. Werden für die gewählten Werte keine verwendbare Genauigkeit und Sensitivität erreicht, müssen die Sensorparameter angepasst werden. Da dies vor der eigentlichen Studie geschehen muss, bietet sich bei der Verwendung dieses Algorithmus eine Vorstudie an. Für den ausgewerteten Datensatz wurde eine Genauigkeit von ca. 90 % und eine Sensitivität von ca. 94 % erzielt. Für den regelbasierten Ansatz werden zwei ungewünschte Effekte beobachtet. Es werden Objekte mit geringen Disparitäten und geringen lateralen Abständen oft als ein Objekt extrahiert. Weiterhin werden Objekte, welche sich aufgrund der Perspektive über einen größeren Disparitätenbereich erstrecken, oft in mehrere Objekte mit unterschiedlicher Entfernung aufgespalten. Beide Effekte führen zu potenziellen Fehlern in der Objektverfolgung und folglich den Konfliktindikatoren. Dies zeigte sich bei der Analyse der realen Verkehrsdaten, wodurch eine weitere Verwendung des Algorithmus ausgeschlossen wurde.

Der modellbasierte Ansatz kann keine generischen Objekte detektieren, sondern muss auf die zu erkennenden Objekte trainiert werden. Somit müssen relevante Objekte im Voraus definiert werden und ein entsprechender Trainingsdatensatz vorhanden sein. Mit Hilfe von überwachtem Lernen kann der Algorithmus einzelne Objektinstanzen pixelgenau in der Bildebene segmentieren, wodurch die Abmaße und Disparität des Objekts gut zu bestimmen sind. Die Klassifikation der Objekte erleichtert die Assoziation zu vorhandenen *Tracks*, wodurch im Vergleich zum regelbasierten Ansatz eine Verbesserung im Gesamtergebnis der Konfliktindikatorenberechnung erreicht wurde. Die Genauigkeit der Objekterkennung bzw. der Instanzsegmentierung des gewählten Ansatzes *Mask R-CNN* wurde von den Autor\*innen des Algorithmus anhand verschiedener komplexer Datensätze in Verkehrsszenarien getestet. Die Ergebnisse sind aufgrund der unterschiedlichen Art des Algorithmus nicht mit denen des regelbasierten Ansatzes vergleichbar. Gerade im Nahbereich ist die Zuordnung einzelner Pixel zu den verschiedenen Objektinstanzen genau, was im „Cityscapes“-Benchmark gezeigt wird. Die mAP über mehrere IoU Werte wird für den Benchmark mit ca. 32 % angegeben, wobei die maximale Entfernung zur Auswertung nicht beschränkt wird. Bei großen Entfernungen und entsprechend kleinen Objekten ergeben sich Überlappungen zwischen Objekten derselben Klasse, welche vom Modell nicht pixelgenau aufgelöst werden. Dies zeigt sich bei einer Beschränkung der maximalen Entfernung auf 100 m bzw. 50 m, wobei eine mAP von 46 % bzw. 49 % erreicht wird. In der Anwendung auf Daten aus dem realen Verkehrsgeschehen zeigten sich keine gravierenden Fehldetektionen. Kurzzeitige fehlerhafte Klassifizierungen von weiter entfernten Objekten wurden bei weiterer Annäherung richtig klassifiziert. Folglich ist der modellbasierte Ansatz das Mittel der Wahl was die Erkennungsrate und die Möglichkeiten zur Weiterverarbeitung der Daten betrifft.

### **5. Wie gut ist die Genauigkeit und Aussagekraft von abgeleiteten Messdaten?**

Die Konfliktindikatoren werden in dieser Arbeit als abgeleitete Messdaten bezeichnet. Die Genauigkeit dieser Daten basiert auf der Güte der Vorverarbeitungskette, der Disparitätenkarte, der Objekterkennung und der daraus abgeleiteten Objektrepräsentation. Zusätzlich müssen das Mess- und Prozessmodell des Kalman-Filters

zum vorliegenden Anwendungsfall passen. Anhand von konstruierten Szenarien und komplementärer Sensorik als Grundwahrheit wurde die Berechnung der Konfliktindikatoren TTC und PET geprüft.

Für die Berechnung der TTC wurden zwei Szenarien umgesetzt, welche zusammengefasst ähnlich Ergebnisse liefern. Der regelbasierte Ansatz zeigt im ersten Szenario für den Nahbereich erhebliche Fehler von mehreren Sekunden bei einer großen Streuung, während der mittlere RMSE des modellbasierten Ansatzes mit  $(0,4 \pm 0,1)$  s geringer ausfällt. Aufgrund des bereits beschriebenen Verhaltens der regelbasierten Objektdetektion bei größeren Entfernungen, können nur Objekte mit einer Entfernung unter 12 m ausgewertet werden, während der modellbasierte Ansatz auch bei Entfernungen über 16 m noch wie erwartet funktioniert. Die Kollisionskursdetektion zeigt für den modellbasierten Ansatz ebenfalls eine bessere Leistung, insbesondere bei geringen Überdeckungen der Objekte bzw. bei geringen Abständen. Für das zweite Szenario zeigt sich ein ähnliches Verhalten, wobei für den regelbasierten Ansatz die TTC und Kollisionskursdetektion insbesondere bei großen Winkeln der Objekttrajektorie zum Kamerasystem starke Fehler aufweisen. Hierbei zeigt sich auch der Einfluss einer ungenügenden Entzerrung der Bilder in der Vorverarbeitungskette.

Die Entzerrungsproblematik ist auch in den Versuchen zur Auswertung der PET präsent, weshalb beide Ansätze große Fehler aufweisen solange sich das Objekt nah am Bildrand befindet. Im zentralen Bildbereich wird für den regelbasierten Ansatz ein mittlerer Fehler von ca.  $(1,1 \pm 0,4)$  s erreicht, während der modellbasierte Ansatz einen geringeren Wert von  $(0,9 \pm 0,3)$  s aufweist. Ein großer Unterschied zeigt sich in der Assoziationsrate des Kalman-Filters, der im regelbasierten Fall nur in 34 % der Zeitschritte ein Objekt assoziiert, während dieser Wert für den modellbasierten Ansatz 92 % beträgt. Dies bestätigt die bereits beschriebenen Probleme der regelbasierten Objekterkennung. Unabhängig vom Ansatz gilt, dass die Berechnung PET auch von der Güte der Eigenbewegung abhängig ist. Fehler im Geschwindigkeitssignal schlagen sich direkt im Konfliktindikator nieder, weshalb dies in der Anwendung berücksichtigt werden muss.

Die beschriebenen Ergebnisse zeigen, dass der regelbasierte Ansatz aufgrund ungenügender Genauigkeit und der dadurch fehlenden Aussagekraft bezüglich der Kritikalität von Konflikten nicht verwendet werden kann. Im Gegensatz dazu zeigt der modellbasierte Ansatz die geforderte Leistung, wobei die dargelegten Fehler auch immer direkt an die Sensorparameter gekoppelt sind. Die Leistung kann weiter optimiert werden, indem das Analysetool erweitert wird. Es wurden einige Ansätze zur Optimierung aufgezeigt, z. B. bezüglich der Kalibration oder der Darstellung des Objektmodells, welche sich in der Güte der Konfliktindikatoren niederschlagen werden.

## **6. Wie zuverlässig ist das Auswertungstool bezogen auf kritische Situationen und Konflikte?**

Wird das Analysetool auf tatsächliche NCS-Daten im realen Verkehr angewendet, zeigt sich ob das Vorgehen zielführend ist und lässt eine Bewertung der Zuverlässigkeit zu. Da für diese Daten keine Grundwahrheit existiert, kann die Analyse nur bezüglich qualitativer Merkmale geschehen. Eine manuelle Sichtung der Daten gibt Aufschluss über die Performanz. Für diese Analyse wird nur der modellbasierte

Ansatz betrachtet, da die regelbasierte Variante aufgrund der Vielzahl an bereits beschriebenen Problemen nicht verwendbar ist.

Es wurden drei Szenarien betrachtet, welche typische Szenen im Straßenverkehr abdecken und somit Informationen bezüglich der Zuverlässigkeit der Konfliktindikatoren liefern. Dabei konnten keine kritischen Fehldetektionen oder ausbleibende Detektionen von Objekten beobachtet werden. Vereinzelt fehlerhafte Klassifizierungen fanden in großen Abständen statt bzw. sind durch perspektivische Effekte zu erklären, wobei kein Fall zu einer ausbleibenden Detektion von kritischen Situationen geführt hätte.

Die Eigenbewegung des Fahrrads, welches ein charakteristisches Wanken aufweist, wurde als Störquelle für die Konfliktindikatoren identifiziert. Durch das Wanken werden die berechneten Objekttrajektorien beeinflusst, was zu Fehlern in den Konfliktindikatoren führt. Zusätzlich kann das gewählte Prozessmodell des Kalman-Filters diese Bewegung nicht abbilden, wodurch es teilweise zu verlorenen *Tracks* kommt. Als Abhilfe bietet sich eine Berechnung der Eigenbewegung des Kamerasystems an, welche alle sechs Freiheitsgrade bestimmt. Entsprechende Algorithmen wurden vorgestellt. Mit dieser Information können Objekte in einem ortsfesten Koordinatensystem dargestellt und verfolgt werden, wodurch der Einfluss des Wankens umgangen wird. Als Folge wird die Assoziation von Objekten zu bestehenden *Tracks* optimiert und verlorene *Tracks* reduziert. Die Konfliktindikatoren werden nicht mehr durch das Wanken beeinflusst, was bezüglich der Eventdetektion die Genauigkeit erhöht.

Als weitere Limitation wurden perspektivische Fehler bei der Berechnung des Punktmodells zur Objektdarstellung erkannt. Die Reduktion eines dreidimensionalen Objekts auf einen Punkt im Raum, indem das Zentrum des auf der Bildebene dargestellten Objekts berechnet und in das Weltkoordinatensystem reprojiziert wird, führt bei einer Änderung der Perspektive auf das Objekt zu Fehlern. Diese Fehler werden in der Trajektorie des Objekts und somit in den Konfliktindikatoren abgebildet. Als Abhilfe sollten Objekte nicht auf ein Punktmodell reduziert werden, sondern mittels dreidimensionaler Modelle repräsentiert werden. Es wurden mehrere Algorithmen vorgestellt, welche dies auf Basis von Monokamera- und Stereokameradaten umsetzen. Die tatsächliche Berechnung des geometrischen Objektzentrums und die Ausdehnung des Objekts bildet zum einen die Basis, um die perspektivischen Fehler zu umgehen, andererseits liefern die Objektdimensionen erweiterte Möglichkeiten zur Berechnung der Grundbedingungen für die Konfliktindikatoren.

Da die modellbasierte Objekterkennung nur a priori definierte Objektklassen erkennen kann, werden auch nur Konfliktindikatoren für diese Objekte bestimmt. Im Gegensatz dazu stand die Anforderung, generische Objekte zu erkennen, um auch kritische Situationen im Zusammenhang mit statischen Objekten zu detektieren, wie z. B. knapp passierte Poller oder ähnliche Objekte. Natürlich können auch solche Objekte in einen Datensatz vorhanden sein und aktiv erkannt werden, die schier endlose Vielfalt der statischen Umgebung erlaubt dies in der Praxis jedoch nicht. Eine bessere Alternative ist der vorgestellte Ansatz, statische Objekte bzw. deren Abwesenheit in einer Freiraumerkennung abzubilden. Ein auf Basis von Stereobildern berechnetes Belegungsgitter kann das Umfeld abbilden und bietet die Möglichkeit zur Detektion kritischer Engstellen. In Kombination mit den detektierten dynamischen Objekten und den objektbezogenen Konfliktindikatoren schließt sich die Lücke zur generischen Objektdetektion. Somit können einerseits Konflikte mit

Verkehrsteilnehmer\*innen detektiert werden, andererseits kritische Situationen im Zusammenhang mit der statischen Umgebung erkannt werden. Zusätzlich existieren die aus der Literatur bekannten fahrerreaktionsbasierten Kriterien zur Eventdetektion, zu welchen diese Arbeit einen komplementären Beitrag leistet.

Betrachtet man das umgesetzte Programm auf Basis des modellbasierten Ansatzes, werden nicht alle Anforderungen an die zuverlässige Detektion von kritischen Situationen erfüllt, da nur Konflikte detektiert werden. Erst eine Erweiterung um die Berechnung des Freiraums lässt eine erweiterte Detektion von konfliktfreien kritischen Situationen zu. Die Detektion der Konflikte erfüllt alle Anforderungen, wenn die beschriebenen Erweiterungen um das dreidimensionale Objektmodell und der Berechnung der Kamerabewegung implementiert werden. Ohne diese können Fehldetektionen in den aufgezeigten Fällen nicht ausgeschlossen werden. Für die analysierten Szenarien ergaben sich keine ausbleibenden Detektionen oder zu wenig kritische Konfliktindikatoren. Allerdings wurden vermehrt FPs durch das Wanken des Fahrrads sowie durch die perspektivischen Fehler der Objektrepräsentation detektiert.

### **7. Welche Aspekte der manuellen Sichtung kann die automatisierte Auswertung nicht leisten?**

Die automatisierte Auswertung kann nicht den Detailgrad einer manuellen Auswertung bieten, selbst mit allen dargestellten potenziellen Erweiterungen des Programms. Die Kodierschemata von NCS-Studien zeigen zum Teil Details auf, welche nur anhand von spezifischen Domänenwissen aus den Daten extrahiert werden können. Die Klassifizierung der Vorhersehbarkeit des Verhaltens von Konfliktpartner\*innen oder dessen Regelkonformität sind hierfür gute Beispiele. Wichtig ist dabei, dass diese Details nicht für eine wirksame Eventdetektion benötigt werden. Durch die Reduktion der auszuwertenden Daten auf wenige Events ist eine manuelle Sichtung machbar. Der Detailgrad einer solchen Auswertung bleibt also prinzipiell erhalten, allerdings mit erheblich reduziertem zeitlichem Aufwand. Es entsteht durch die automatisierte Eventdetektion kein Nachteil.

Sollen aus dem Gesamtvolumen der Daten, also nicht beschränkt auf detektierte Events, spezifische Merkmale extrahiert werden, welche diesem nicht erreichbaren Detailgrad entsprechen, so ist eine automatisierte Auswertung nicht möglich. Eine Auflistung der genutzten und verfügbaren Infrastruktur wäre ein solches Beispiel, wenn diese in allen Facetten dargestellt werden soll. Eine Unterscheidung zwischen Gehweg und gemeinsamen Geh- und Radweg ist z. B. nicht umsetzbar. Beschränken sich die möglichen Klassen jedoch auf automatisiert detektierbare Optionen, ist auch hier eine automatisierte Auswertung umsetzbar.

### **8. Welche sinnvollen Erweiterungen des Auswertungstools sind denkbar?**

Aus den Videodaten können neben den Events noch weitere Informationen extrahiert werden, wie anhand des Kodierschemas einer NCS dargelegt wurde. Es wurden Methoden und deren Grenzen aufgezeigt, welche entsprechende Informationen automatisiert extrahieren können. Dies betrifft neben der verwendeten und verfügbaren Infrastruktur auch Verkehrszeichen und -ampeln. Zusätzlich können Informationen zum Freiraum um das Versuchsfahrrad wertvolle Informationen bezüglich des statischen Umfeldes liefern, welches durch die modellbasierte Objektdetektion nicht

zwingend abgebildet wird. Die Analyse der Videodaten bezüglich des Wetters und der Straßenbeschaffenheit kann ebenfalls einen Mehrwert bei der Interpretation des Fahrerverhaltens bieten. Zur Optimierung der Funktionalität des Auswertungstools wurden weitere mögliche Erweiterungsoptionen vorgestellt. Diese beinhalten unter anderem die Berechnung der Disparitätenkarte anhand von Methoden des maschinellen Lernens sowie Methoden zur Online-Kalibration. In Abhängigkeit des Studienziels sind Erweiterungen möglich, werden jedoch hier aufgrund der unzähligen Möglichkeiten nicht gelistet.

## 7.2 Fazit

Die vorliegende Arbeit hat gezeigt, dass eine automatisierte Auswertung sinnvoll und für viele Daten möglich ist. Für die Detektion von relevanten Events wurde dies anhand eines implementierten Programms gezeigt. Automatisierte Eventdetektion in Kombination mit einer manuellen Auswertung dieser Events vereint die Vorteile beider Varianten, löst dabei die Problematik des Zeitaufwands und liefert objektive Konfliktindikatoren. Allerdings muss schon beim Design einer NCS vieles beachtet werden, da die Leistungsfähigkeit der verwendeten Algorithmen auch vom Design der Messtechnik abhängig ist. Folglich erhöht sich die Vorlaufzeit einer NCS, wobei die investierte Zeit durch die Vorteile der automatisierten Auswertung gut investiert ist. Generell können viele Informationen aus den NCS-Daten automatisch extrahiert werden, die Grenzen zeigen sich im Detailgrad. Dieser muss ebenfalls in der Vorbereitung einer Studie diskutiert werden, um Kosten und Nutzen der detaillierten manuellen Auswertung einer weniger detaillierten automatisierten Auswertung gegenüberzustellen.

Durch die Forschung im Bereich AD, Robotik sowie vielen anderen Feldern, werden ständig neue und teilweise bahnbrechende Entwicklungen präsentiert. Viele davon stützen sich auf die Methoden des maschinellen Lernens. Es bietet sich an die aktuellen Entwicklungen in den bekannten und neuen Benchmarks im Auge zu behalten. Viele Autoren veröffentlichen den Quellcode der entwickelten Algorithmen unter Lizenzen, welche eine nichtkommerzielle Nutzung erlauben. Die Übernahme in ein modular entworfenes Auswertungsprogramm, wie das in dieser Arbeit beschriebene, gestaltet sich somit einfach und lädt zum Experimentieren ein. Neben Algorithmen werden auch immer mehr Datensätze offen zugänglich gemacht, welche zum Training von Modellen verwendet werden können.

Die Wichtigkeit von verfügbaren Datensätzen darf nicht unterschätzt werden. Die Fortschritte der datengetriebenen Forschung hängen davon ab, was die Veröffentlichung des „KITTI“-Datensatzes und die darauf basierenden Fortschritte in vielen Bereichen der Umfelderfassung und -interpretation eindrucksvoll bewiesen haben. NCS können hiervon profitieren, aber auch selbst einen erheblichen Beitrag leisten, wenn die kodierten Datensätze für die Forschung frei verfügbar gemacht werden. Ein entsprechendes Konzept zur Nachnutzung der aufgezeichneten und kodierten Daten sollte in Zukunft bei keiner NCS fehlen, auch wenn ein nicht unerheblicher Aufwand entsteht, vor allem bezüglich des Datenschutzes der abgebildeten Individuen.

# Literaturverzeichnis

- [1] INFAS INSTITUT FÜR ANGEWANDTE SOZIALWISSENSCHAFT GMBH: Mobilität in Deutschland - MiD Kurzreport. (2018). [http://www.mobilitaet-in-deutschland.de/pdf/infas\\_Mobilitaet\\_in\\_Deutschland\\_2017\\_Kurzreport\\_DS.pdf](http://www.mobilitaet-in-deutschland.de/pdf/infas_Mobilitaet_in_Deutschland_2017_Kurzreport_DS.pdf). – Aufgerufen am 24.10.2018
- [2] BUNDESMINISTERIUM FÜR VERKEHR, BAU UND STADTENTWICKLUNG: Nationaler Radverkehrsplan 2020 - Den Radverkehr gemeinsam Entwickeln. (2012). <https://nationaler-radverkehrsplan.de/de/node/5742>. – Aufgerufen am 22.10.2018
- [3] SINUS MARKT- UND SOZIALFORSCHUNG GMBH: Fahrrad-Monitor Deutschland 2019. (2019). [https://www.bmvi.de/SharedDocs/DE/Anlage/K/fahrradmonitor-2019-ergebnisse.pdf?\\_\\_blob=publicationFile](https://www.bmvi.de/SharedDocs/DE/Anlage/K/fahrradmonitor-2019-ergebnisse.pdf?__blob=publicationFile). – Aufgerufen am 21.06.2020
- [4] ZWEIRAD-INDUSTRIE-VERBAND: *Zahlen-Daten-Fakten zum Fahrradmarkt in Deutschland 2017*. [https://www.ziv-zweirad.de/fileadmin/redakteure/Downloads/Marktdaten/PK-2018\\_13-03-2018\\_Praesentation.pdf](https://www.ziv-zweirad.de/fileadmin/redakteure/Downloads/Marktdaten/PK-2018_13-03-2018_Praesentation.pdf). – Aufgerufen am 24.10.2018
- [5] (DESTATIS), Statistisches B.: Pressemitteilung Nr. 23. (2020). [https://www.destatis.de/DE/Presse/Pressemitteilungen/Zahl-der-Woche/2020/PD20\\_23\\_p002.html](https://www.destatis.de/DE/Presse/Pressemitteilungen/Zahl-der-Woche/2020/PD20_23_p002.html). – Aufgerufen am 21.06.2020
- [6] CZEH, Alexander: Pop-up-Infrastruktur in Berlin. (2020). <https://experimobilitaet.de/?p=1139>. – Aufgerufen am 21.06.2020
- [7] STATPLAN - INGENIEURBÜRO FÜR VERKEHRSERHEBUNGEN, -STATISTIK UND -PLANUNG: Fahrradverkehr-Pegelzählungen Berlin Jahresbericht 2016. (2017). [https://www.berlin.de/senuvk/verkehr/lenkung/vlb/download/bericht\\_radverkehr\\_2016.pdf](https://www.berlin.de/senuvk/verkehr/lenkung/vlb/download/bericht_radverkehr_2016.pdf). – Aufgerufen am 31.10.2018
- [8] SCHLEINITZ, K. ; PETZOLDT, T. ; FRANKE-BARTHOLDT, L. ; KREMS, J.F. ; GEHLERT, T.: The German Naturalistic Cycling Study – Comparing cycling speed of riders of different e-bikes and conventional bicycles. In: *Safety Science* (2015). <http://dx.doi.org/https://doi.org/10.1016/j.ssci.2015.07.027>. – DOI <https://doi.org/10.1016/j.ssci.2015.07.027>
- [9] ALRUTZ, D.: *Pedelecs im Alltag - Was bedeutet der Boom für die (inter-)kommunale Planung?*. – Präsentiert auf dem 5. Workshop Radverkehrsstrate-

- gie Metropolregion Hannover Braunschweig Göttingen Wolfsburg am 1. März 2013 in Hannover. Aufgerufen am 29.10.2018.
- [10] STATISTISCHES BUNDESAMT (DESTATIS): Unfallentwicklung auf deutschen Straßen 2017. (2018). [https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2018/verkehrsunfaelle\\_2017/Pressebrochure\\_unfallentwicklung.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2018/verkehrsunfaelle_2017/Pressebrochure_unfallentwicklung.pdf?__blob=publicationFile). – Aufgerufen am 06.11.2018
- [11] SHIRAZI, Mohammad S. ; MORRIS, Brendan T.: Investigation of Safety Analysis Methods Using Computer Vision Techniques. In: *Journal of Electronic Imaging* 26 (2017), Nr. 5
- [12] SCHAGEN, Ingrid van ; SAGBERG, Fridulv: The Potential Benefits of Naturalistic Driving for Road Safety Research: Theoretical and Empirical Considerations and Challenges for the Future. In: *Procedia - Social and Behavioral Sciences* 48 (2012), 692 - 701. <http://dx.doi.org/https://doi.org/10.1016/j.sbspro.2012.06.1047>. – DOI <https://doi.org/10.1016/j.sbspro.2012.06.1047>. – ISSN 1877-0428. – Transport Research Arena 2012
- [13] FESTA CONSORTIUM: FESTA Handbook, Version 7. (2017). <http://fot-net.eu/Documents/festa-handbook-version-7/>. – Aufgerufen am 12.10.2018
- [14] SCHLEINITZ, Katja ; FRANKE-BARTHOLDT, Luise ; PETZOLDT, Tibor ; SCHWANITZ, Stefan ; GEHLERT, Tina ; KUEHN, Matthias: *Pedelec-Naturalistic Cycling Study*. Gesamtverband der Deutschen Versicherungswirtschaft e. V., 2014. – ISBN 978-3-939163-50-3
- [15] DOZZA, M. ; WERNEKE, J. ; FERNANDEZ, A.: Piloting the Naturalistic Methodology on Bicycles. In: *Proceedings, International Cycling Safety Conference 2012*, 2012
- [16] MOHANTY, S ; LEE, A ; CARVALHO, T ; DIAS, L ; LOVEGROVE, G: A Global Review of Current Instrumented Probe Bicycle (IPB) Technology and Research. In: *International Cycling Safety Conference. 2014.*, 2014
- [17] GELLERMAN, H ; SVANBERG, E ; BARNARD, Y: Data Sharing Framework for Naturalistic Driving Data. In: *ITS World Congress 2015 Proceedings Leeds*, 2015
- [18] KASSIM, Ali ; ISMAIL, Karim ; HASSAN, Yasser: Automated measuring of cyclist – motor vehicle post encroachment time at signalized intersections. In: *Canadian Journal of Civil Engineering* 41 (2014), Nr. 7, S. 605–614. <http://dx.doi.org/10.1139/cjce-2013-0565>. – DOI 10.1139/cjce-2013-0565
- [19] GUO, Feng ; KLAUER, Sheila G. ; MCGILL, Michael T. ; DINGUS, Thomas A.: Evaluating the Relationship Between Near-Crashes and Crashes : Can Near-Crashes Serve as a Surrogate Safety Metric for Crashes?, U.S. Department of Transportation, National Highway Traffic Safety Administration, 2010

- 
- [20] PERKINS, Joseph L. Stuart R Harris H. Stuart R Harris: *Highway Research Record*. Bd. 225: *Traffic Conflict Characteristics - Accident Potential at Intersections*. Highway Research Board, Washington D.C., 1968 [https://books.google.de/books?id=\\_kyVGwAACAAJ](https://books.google.de/books?id=_kyVGwAACAAJ)
- [21] DOZZA, Marco ; FERNANDEZ, Andre: Understanding Bicycle Dynamics and Cyclist Behavior From Naturalistic Field Data. In: *IEEE Trans. Intelligent Transportation Systems* 15 (2014), Nr. 1, S. 376–384
- [22] DOZZA, Marco ; GONZÁLEZ, Nieves P. n.: Recognising safety critical events: can automatic video processing improve naturalistic data analyses? In: *Accident; analysis and prevention* 60 (2013), S. 298–304
- [23] DOZZA, Marco ; WERNEKE, Julia ; MACKENZIE, Michael: eBikeSAFE: A Naturalistic Cycling Study to Understand how Electrical Bicycles Change Cycling Behaviour and Influence Safety. In: *Proceedings, International Cycling Safety Conference 2013*, 2013, S. 20–21
- [24] JOHNSON, Marilyn ; CHONG, Derek ; CARROLL, Justin ; KATZ, Rod ; OXLEY, Jennie ; CHARLTON, Judith: Naturalistic cycling study: Identifying risk factors for cyclists in the Australian capital territory. In: *Monash University Accident Research Centre - Report 322* (2014). <https://www.monash.edu/muarc/archive/our-publications/reports/muarc322>
- [25] LIETZ, H ; PETZOLDT, T ; HENNING, M ; HAUPT, J ; WANIELIK, G ; KREMS, J ; MOSEBACH, H ; SCHOMERUS, J ; BAUMANN, M ; NOYER, U: Methodische und technische Aspekte einer Naturalistic Driving Study. In: *FAT-Schriftenreihe* (2011), Nr. 229
- [26] SAUNIER, N. ; SAYED, T. ; LIM, C.: Probabilistic Collision Prediction for Vision-Based Automated Road Safety Analysis. In: *2007 IEEE Intelligent Transportation Systems Conference, 2007*. – ISSN 2153–0009, S. 872–878
- [27] SHAPIRO, L.G. ; STOCKMAN, G.C.: *Computer Vision*. Prentice Hall, 2001. – ISBN 9780130307965
- [28] *Kapitel Visuelle Informationsverarbeitung*. In: MÜSSELER, Jochen: *Allgemeine Psychologie; 3. Auflage*. Berlin : Springer, 2017, S. 13–49
- [29] JIANG, Xiaoyi ; BUNKE, Horst: *Dreidimensionales Computersehen: Gewinnung und Analyse von Tiefenbildern*. Springer-Verlag, Berlin und Heidelberg, 1997
- [30] HARTLEY, R. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004
- [31] POLLEFEYS, Marc: *Visual 3D Modeling from Images*. <https://www.cs.unc.edu/~marc/tutorial/index.html>. – Aufgerufen am 25.05.2020
- [32] SCHREER, Oliver: *Stereoanalyse und Bildsynthese*. Berlin [u.a.] : Springer, 2005. – ISBN 354023439X 9783540234395
-

- [33] EIGEN, David ; PUHRSCH, Christian ; FERGUS, Rob: Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In: *CoRR* abs/1406.2283 (2014)
- [34] GEIGER, Andreas ; LENZ, Philip ; STILLER, Christoph ; URTASUN, Raquel: Vision meets Robotics: The KITTI Dataset. In: *International Journal of Robotics Research (IJRR)* (2013)
- [35] FU, Huan ; GONG, Mingming ; WANG, Chaohui ; BATMANGHELICH, Kayhan ; TAO, Dacheng: Deep Ordinal Regression Network for Monocular Depth Estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, S. 2002–2011
- [36] BRADSKI, Gary ; KAEHLER, Adrian: *Learning OpenCV, [Computer Vision with OpenCV Library ; software that sees]*. 1. ed. O'Reilly Media, 2008. – ISBN 0–596–51613–4
- [37] SZELISKI, Richard: *Computer Vision: Algorithms and Applications*. Springer Verlag, 2010
- [38] BROWN, Duane C.: Decentering Distortion of Lenses. In: *Photogrammetric Engineering and Remote Sensing* (1966)
- [39] QI, Wang ; LI, Fu ; ZHENZHONG, Liu: Review on Camera Calibration, 2010, S. 3354 – 3358
- [40] LUBER, Andreas: *Ein generisches Abbildungsmodell für Stereokamerasysteme*, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Diss., 2015. <http://dx.doi.org/http://dx.doi.org/10.18452/17114>. – DOI <http://dx.doi.org/10.18452/17114>
- [41] ZHANG, Z.: A Flexible New Technique for Camera Calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 11, S. 1330–1334. <http://dx.doi.org/10.1109/34.888718>. – DOI 10.1109/34.888718. – ISSN 0162–8828
- [42] BRADSKI, G.: The OpenCV Library. In: *Dr. Dobb's Journal of Software Tools* (2000)
- [43] FERNÁNDEZ-LLORCA, David ; SOTELO, Miguel-Angel ; IGNACIO, Parra ; OCAÑA, Manuel ; BERGASA, Luis: Error Analysis in a Stereo Vision-Based Pedestrian Detection Sensor for Collision Avoidance Applications. In: *Sensors* 10 (2010). <http://dx.doi.org/10.3390/s100403741>. – DOI 10.3390/s100403741
- [44] DANG, Thao: *Kontinuierliche Selbstkalibrierung von Stereokameras*, Diss., 2007. <http://dx.doi.org/10.5445/KSP/1000006635>. – DOI 10.5445/KSP/1000006635
- [45] WITTENBURG, Peter ; BRUGMAN, Hennie ; RUSSEL, Albert ; KLASSMANN, Alexander ; SLOETJES, Han: ELAN: a Professional Framework for Multimodality Research. In: *LREC* Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands, 2006

- 
- [46] MIRANDA-MORENO, Luis: Flexible, Mobile Video Camera System and Open Source Video Analysis Software for Road Safety and Behavioral Analysis. In: *Transportation Research Record Journal of the Transportation Research Board* (2013), 01
- [47] MESSELODI, Stefano ; MODENA, Carla M. ; ZANIN, Michele: A computer vision system for the detection and classification of vehicles at urban road intersections. In: *Pattern Analysis and Applications* 8 (2005), Nr. 1, S. 17–31. <http://dx.doi.org/10.1007/s10044-004-0239-9>. – DOI 10.1007/s10044-004-0239-9. – ISSN 1433-755X
- [48] ZANGENEHPOUR, Sohail ; ROMANCYSHYN, Taras ; MIRANDA-MORENO, Luis F. ; SAUNIER, Nicolas: Video-Based Automatic Counting For Short-Term Bicycle Data Collection in a Variety of Environments. In: *94th Transportation Research Board Annual Meeting*, 2015
- [49] ST-AUBIN, Paul ; SAUNIER, Nicolas ; MIRANDA-MORENO, Luis: Large-Scale Automated Proactive Road Safety Analysis Using Video Data. In: *Transportation Research Part C: Emerging Technologies* 58 (2015). <http://dx.doi.org/10.1016/j.trc.2015.04.007>. – DOI 10.1016/j.trc.2015.04.007
- [50] AFGHARI, Amir P. ; ISMAIL, Karim ; SAUNIER, Nicolas ; SHARMA, Abhisar ; MIRANDA-MORENO, Luis: Pedestrian-cyclist Interactions at Bus Stops along Segregated Bike Paths: A Case Study of Montreal. In: *93rd Annual Meeting of Transportation Research Board. Washington DC*, 2014
- [51] WINNER, Hermann (Hrsg.) ; HAKULI, Stephan (Hrsg.) ; LOTZ, Felix (Hrsg.) ; SINGER, Christina (Hrsg.): *Handbuch Fahrerassistenzsysteme*. 3. Springer Vieweg, 2015 (ATZ/MTZ-Fachbuch). <http://dx.doi.org/10.1007/978-3-658-05734-3>. <http://dx.doi.org/10.1007/978-3-658-05734-3>. – ISBN 978-3-658-05733-6
- [52] SPETH, J.: *Videobasierte modellgestützte Objekterkennung für Fahrerassistenzsysteme*. Cuvillier, 2010 (Audi-Dissertationsreihe). – ISBN 9783869553177
- [53] KELLER, Christoph G. ; LLORCA, David F. ; GAVRILA, Darius M.: Dense Stereo-Based ROI Generation for Pedestrian Detection. In: DENZLER, Joachim (Hrsg.) ; NOTNI, Gunther (Hrsg.) ; SÜSSE, Herbert (Hrsg.): *Pattern Recognition*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2009. – ISBN 978-3-642-03798-6, S. 81–90
- [54] SCHARSTEIN, Daniel ; SZELISKI, Richard: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. In: *International journal of computer vision* 47 (2002), Nr. 1-3, S. 7–42
- [55] UNGER, Christian: *Contributions to Stereo Vision*, Technical University Munich, Diss., 2013. <https://nbn-resolving.org/urn:nbn:de:bvb:91-diss-20130617-1136933-0-0>
- [56] BLEYER, Michael ; BREITENEDER, Christian: Stereo Matching - State-of-the-Art and Research Challenges. In: *Advanced Topics in Computer Vision*. Springer, 2013
-

- [57] EINECKE, N. ; EGGERT, J.: A Multi-Block-Matching Approach for Stereo. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, S. 585–592
- [58] HIRSCHMÜLLER, Heiko: Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*. Washington, DC, USA : IEEE Computer Society, 2005 (CVPR '05). – ISBN 0–7695–2372–2, 807–814
- [59] PFEIFFER, David: *The Stixel World*, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Diss., 2012. <http://dx.doi.org/http://dx.doi.org/10.18452/16576>. – DOI <http://dx.doi.org/10.18452/16576>
- [60] GÖRNER, Martin ; CHILIAN, Annett ; HIRSCHMÜLLER, Heiko: *Towards an Autonomous Walking Robot for Planetary Surfaces*, 2010
- [61] FRANKE, U. ; PFEIFFER, D. ; RABE, C. ; KNOEPEL, C. ; ENZWEILER, M. ; STEIN, F. ; HERRTWICH, R. G.: Making Bertha See. In: *2013 IEEE International Conference on Computer Vision Workshops*, 2013, S. 214–221
- [62] HIRSCHMÜLLER, Heiko: Semi-Global Matching: Motivation, Development and Applications Photogrammetric Week '11, 2011, S. 173–184
- [63] OPENCV.ORG: *OpenCV - cv::StereoSGBM Class Reference*. [https://docs.opencv.org/3.2.0/d2/d85/classcv\\_1\\_1StereoSGBM.html](https://docs.opencv.org/3.2.0/d2/d85/classcv_1_1StereoSGBM.html). – Aufgerufen am 02.03.2019
- [64] MAINZER, K.: *Künstliche Intelligenz – Wann übernehmen die Maschinen?* Springer Berlin Heidelberg, 2019 (Technik im Fokus). – ISBN 9783662580455
- [65] MUKHTAR, A. ; XIA, L. ; TANG, T. B.: Vehicle Detection Techniques for Collision Avoidance Systems: A Review. In: *IEEE Transactions on Intelligent Transportation Systems* 16 (2015), Nr. 5, S. 2318–2338. <http://dx.doi.org/10.1109/TITS.2015.2409109>. – DOI 10.1109/TITS.2015.2409109. – ISSN 1524–9050
- [66] SIVARAMAN, S. ; TRIVEDI, M. M.: Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. In: *IEEE Transactions on Intelligent Transportation Systems* 14 (2013), Nr. 4, S. 1773–1795. <http://dx.doi.org/10.1109/TITS.2013.2266661>. – DOI 10.1109/TITS.2013.2266661. – ISSN 1524–9050
- [67] TZOMAKAS, Christos ; SEELEN, Werner von: Vehicle Detection in Traffic Scenes Using Shadows. (1998). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.45.3234&rep=rep1&type=pdf>
- [68] CATALÁ PRAT, Alvaro: *Sensordatenfusion und Bildverarbeitung zur Objekt- und Gefahrenerkennung*, Technische Universität Carolo-Wilhelmina zu Braunschweig, Diss., 2011

- 
- [69] LI-SHENG, J. ; BAI-YUAN, G. ; RONG-BEN, W. ; LIE, G. ; YI-BING, Z. ; LIN-HUI, L.: Preceding Vehicle Detection Based on Multi-characteristics Fusion. In: *2006 IEEE International Conference on Vehicular Electronics and Safety*, 2006, S. 356–360
- [70] CAO, Yanpeng ; RENFREW, Alasdair ; COOK, Peter: Comprehensive Vehicle Motion Analysis using Optical Flow Optimization Based on Pulse-Coupled Neural Network. In: *IFAC Proceedings Volumes 41* (2008), Nr. 2, 158 - 163. <http://dx.doi.org/https://doi.org/10.3182/20080706-5-KR-1001.00027>. – DOI <https://doi.org/10.3182/20080706-5-KR-1001.00027>. – ISSN 1474–6670. – 17th IFAC World Congress
- [71] PERROLLAZ, M. ; YODER, J. ; NEGRE, A. ; SPALANZANI, A. ; LAUGIER, C.: A Visibility-Based Approach for Occupancy Grid Computation in Disparity Space. In: *IEEE Transactions on Intelligent Transportation Systems* 13 (2012), Nr. 3, S. 1383–1393. <http://dx.doi.org/10.1109/TITS.2012.2188393>. – DOI 10.1109/TITS.2012.2188393. – ISSN 1524–9050
- [72] LABAYRADE, Raphael ; AUBERT, Didier: In-Vehicle Obstacles Detection and Characterization by Stereovision. In: *in 'Proceedings the 1st International Workshop on In-Vehicle Cognitive Computer Vision Systems*, 2003, S. 13–19
- [73] RABE, Clemens ; FRANKE, U. ; GEHRIG, Stefan D.: Fast Detection of Moving Objects in Complex Scenarios. In: *2007 IEEE Intelligent Vehicles Symposium* (2007), S. 398–403
- [74] LENZ, Philip ; ZIEGLER, Julius ; GEIGER, Andreas ; ROSER, Martin: Sparse Scene Flow Segmentation for Moving Object Detection in Urban Environments. In: *Intelligent Vehicles Symposium (IV)*, 2011
- [75] SUGANUMA, N.: Clustering and tracking of obstacles from Virtual Disparity Image. In: *2009 IEEE Intelligent Vehicles Symposium*, 2009. – ISSN 1931–0587, S. 111–116
- [76] LABAYRADE, Raphael ; AUBERT, Didier: A Single Framework for Vehicle Roll, Pitch, Yaw Estimation and Obstacles Detection by Stereovision. In: *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)* IEEE, 2003, S. 31–36
- [77] HU, Zhencheng ; LAMOSA, Francisco ; UCHIMURA, Keiichi: A Complete U-V-Disparity Study for Stereovision Based 3D Driving Environment Analysis. In: *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*. Washington, DC, USA : IEEE Computer Society, 2005 (3DIM '05), 204–211
- [78] FRANKE, U. ; KUTZBACH, I.: Fast Stereo Based Object Detection for Stop & Go Traffic. In: *Proceedings of Conference on Intelligent Vehicles*, 1996, S. 339–344
- [79] TOULMINET, G. ; BERTOZZI, M. ; MOUSSET, S. ; BENSRAHAIR, A. ; BROGGI, A.: Vehicle Detection by Means of Stereo Vision-Based Obstacles Features Extraction and Monocular Pattern Analysis. In: *IEEE Transactions*
-

- on Image Processing* 15 (2006), Nr. 8, S. 2364–2375. <http://dx.doi.org/10.1109/TIP.2006.875174>. – DOI 10.1109/TIP.2006.875174. – ISSN 1057–7149
- [80] SOQUET, N. ; AUBERT, D. ; HAUTIERE, N.: Road Segmentation Supervised by an Extended V-Disparity Algorithm for Autonomous Navigation. In: *2007 IEEE Intelligent Vehicles Symposium*, 2007. – ISSN 1931–0587, S. 160–165
- [81] IRKI, Z. ; OUSSAR, A. ; HAMDI, M. ; SEDDI, F.: A Fuzzy UV-disparity Based Approach for Following a Leader Mobile Robot. In: *2015 International Conference on Advanced Robotics (ICAR)*, 2015, S. 170–175
- [82] BENACER, I. ; HAMISSI, A. ; KHOUAS, A.: A Novel Stereovision Algorithm for Obstacles Detection Based on U-V-Disparity Approach. In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, S. 369–372
- [83] WANG, B. ; FLOREZ, S. A. R. ; FRÉMONT, V.: Multiple Obstacle Detection and Tracking using Stereo Vision: Application and Analysis. In: *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, 2014, S. 1074–1079
- [84] BADINO, Hernán ; FRANKE, Uwe ; PFEIFFER, David: The Stixel World - A Compact Medium Level Representation of the 3D-World. In: *Joint Pattern Recognition Symposium* Springer, 2009, S. 51–60
- [85] PFEIFFER, David ; FRANKE, Uwe: Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In: *Proceedings of the British Machine Vision Conference*, BMVA Press, 2011, S. 51.1–51.12
- [86] MUFFERT, M. ; MILBICH, T. ; PFEIFFER, D. ; FRANKE, U.: May I enter the Roundabout? A Time-To-Contact Computation Based on Stereo-Vision. In: *2012 IEEE Intelligent Vehicles Symposium*, 2012. – ISSN 1931–0587, S. 565–570
- [87] ESTER, Martin ; KRIEGEL, Hans-Peter ; SANDER, Jörg ; XU, Xiaowei: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, AAAI Press, 1996, S. 226–231
- [88] VEIT, Thomas: Connexity based fronto-parallel plane detection for stereovision obstacle segmentation. In: *IEEE Int. Conf. on Robotics and Automation, Workshop on Safe Navigation in Open and Dynamic Environments: Applications to Autonomous Vehicles. 2009*, 2009
- [89] KORMANN, Benjamin ; NEVE, Antje ; KLINKER, Gudrun ; STECHELE, Walter u. a.: Stereo Vision based Vehicle Detection. In: *VISAPP (2)*, 2010, S. 431–438
- [90] DAL MUTTO, Carlo ; DOMINIO, Fabio ; ZANUTTIGH, Pietro ; MATTOCCIA, Stefano: Stereo Vision and Scene Segmentation. In: *Current Advancements in Stereo Vision*. IntechOpen, 2012
- [91] BARROIS, B. ; HRISTOVA, S. ; WOHLER, C. ; KUMMERT, F. ; HERMES, C.: 3D Pose Estimation of Vehicles Using a Stereo Camera. In: *2009 IEEE Intelligent Vehicles Symposium*, 2009, S. 267–272

- 
- [92] BADINO, Hernán ; FRANKE, Uwe ; RABE, Clemens ; GEHRIG, Stefan: Stereo-Vision Based Detection of Moving Objects under Strong Camera Motion. In: *International Conference on Computer Vision Theory and Applications*, 2006, S. 253–260
- [93] BACHMANN, Alexander: *Dichte Objektsegmentierung in Stereobildfolgen*, Diss., 2010. <http://dx.doi.org/10.5445/KSP/1000019278>. – DOI 10.5445/KSP/1000019278
- [94] AGARWAL, Shivang ; TERRAIL, Jean Ogier D. ; JURIE, Frédéric: Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks. In: *CoRR* abs/1809.03193 (2018)
- [95] ZHAO, Zhong-Qiu ; ZHENG, Peng ; XU, Shou-Tao ; WU, Xindong: Object Detection With Deep Learning: A Review. In: *IEEE Transactions on Neural Networks and Learning Systems* (2019), 1–21. <http://dx.doi.org/10.1109/tnnls.2018.2876865>. – DOI 10.1109/tnnls.2018.2876865. – ISSN 2162–2388
- [96] LOWE, D. G.: Object Recognition from Local Scale-Invariant Features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision* Bd. 2, 1999, S. 1150–1157 vol.2
- [97] MIKOLAJCZYK, K. ; SCHMID, C.: A Performance Evaluation of Local Descriptors. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), Nr. 10, S. 1615–1630. <http://dx.doi.org/10.1109/TPAMI.2005.188>. – DOI 10.1109/TPAMI.2005.188. – ISSN 0162–8828
- [98] LIU, Li ; OUYANG, Wanli ; WANG, Xiaogang ; FIEGUTH, Paul ; LIU, Xinwang ; PIETIKÄINEN, Matti: Deep Learning for Generic Object Detection: A Survey. In: *CoRR* (2018). <http://arxiv.org/abs/1809.02165>
- [99] KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; HINTON, Geoffrey E.: ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems*, S. 2012
- [100] SERMANET, Pierre ; EIGEN, David ; ZHANG, Xiang ; MATHIEU, Michaël ; FERGUS, Robert ; LECUN, Yann: OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In: *International Conference on Learning Representations (ICLR2014)*, CBLIS, April 2014, 2014
- [101] DENG, J. ; DONG, W. ; SOCHER, R. ; LI, L.-J. ; LI, K. ; FEI-FEI, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*, 2009
- [102] DÖBEL, I. ; LEIS, M. ; MOLINA VOGELSANG, M. ; NEUSTROEV, D. ; PETZKA, H. ; RIEMER, A. ; RÜPING, S. ; VOSS, A. ; WEGELE, M. ; WELZ, J.: Maschinelles Lernen. Eine Analyse zu Kompetenzen, Forschung und Anwendung. (2018). [https://www.bigdata.fraunhofer.de/content/dam/bigdata/de/documents/Publicationen/Fraunhofer\\_Studie\\_ML\\_201809.pdf](https://www.bigdata.fraunhofer.de/content/dam/bigdata/de/documents/Publicationen/Fraunhofer_Studie_ML_201809.pdf)
- [103] PEDREGOSA, F. ; VAROQUAUX, G. ; GRAMFORT, A. ; MICHEL, V. ; THIRION, B. ; GRISEL, O. ; BLONDEL, M. ; PRETTENHOFER, P. ; WEISS, R. ; DUBOURG, V. ; VANDERPLAS, J. ; PASSOS, A. ; COURNAPEAU, D. ; BRUCHER, M. ;
-

- PERROT, M. ; DUCHESNAY, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830
- [104] SCIKIT-LEARN.ORG: *Choosing the right estimator*. [https://scikit-learn.org/stable/tutorial/machine\\_learning\\_map/index.html](https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html). – Aufgerufen am 12.05.2019
- [105] BISHOP, Christopher M.: *Pattern recognition and machine learning*. Springer, 2006
- [106] KARPATY, Andreij: *CS231n: Convolutional Neural Networks for Visual Recognition*. <http://cs231n.stanford.edu/syllabus.html>. Version: 2019. – Aufgerufen am 21.05.2019
- [107] LECUN, Y. ; BOTTOU, L. ; BENGIO, Y. ; HAFFNER, P.: Gradient-Based Learning Applied to Document Recognition. In: *Proceedings of the IEEE* 86 (1998), Nr. 11, S. 2278–2324. <http://dx.doi.org/10.1109/5.726791>. – DOI 10.1109/5.726791. – ISSN 0018–9219
- [108] GIRSHICK, Ross B. ; DONAHUE, Jeff ; DARRELL, Trevor ; MALIK, Jitendra: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *CoRR* abs/1311.2524 (2013). <http://arxiv.org/abs/1311.2524>
- [109] UIJLINGS, Jasper R. R. ; VAN DE SANDE, Koen E. A. ; GEVERS, Theo ; SMEULDERS, Arnold W. M.: Selective Search for Object Recognition. In: *International Journal of Computer Vision* 104 (2013), Nr. 2, S. 154–171
- [110] GIRSHICK, Ross B.: Fast R-CNN. In: *CoRR* abs/1504.08083 (2015). <http://arxiv.org/abs/1504.08083>
- [111] REN, Shaoqing ; HE, Kaiming ; GIRSHICK, Ross B. ; SUN, Jian: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *CoRR* abs/1506.01497 (2015). <http://arxiv.org/abs/1506.01497>
- [112] REDMON, Joseph ; DIVVALA, Santosh K. ; GIRSHICK, Ross B. ; FARHADI, Ali: You Only Look Once: Unified, Real-Time Object Detection. In: *CoRR* abs/1506.02640 (2015). <http://arxiv.org/abs/1506.02640>
- [113] SZEGEDY, Christian ; LIU, Wei ; JIA, Yangqing ; SERMANET, Pierre ; REED, Scott E. ; ANGUELOV, Dragomir ; ERHAN, Dumitru ; VANHOUCHE, Vincent ; RABINOVICH, Andrew: Going Deeper with Convolutions. In: *CoRR* abs/1409.4842 (2014). <http://arxiv.org/abs/1409.4842>
- [114] HE, Kaiming ; GKIOXARI, Georgia ; DOLLÁR, Piotr ; GIRSHICK, Ross B.: Mask R-CNN. In: *CoRR* abs/1703.06870 (2017). <http://arxiv.org/abs/1703.06870>
- [115] ABDULLA, Waleed: *Mask R-CNN for Object Detection and Instance Segmentation on Keras and TensorFlow*. [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN), 2017. – Aufgerufen am 12.02.2018

- 
- [116] LIN, Tsung-Yi ; MAIRE, Michael ; BELONGIE, Serge J. ; BOURDEV, Lubomir D. ; GIRSHICK, Ross B. ; HAYS, James ; PERONA, Pietro ; RAMANAN, Deva ; DOLLÁR, Piotr ; ZITNICK, C. L.: Microsoft COCO: Common Objects in Context. In: *CoRR* abs/1405.0312 (2014). <http://arxiv.org/abs/1405.0312>
- [117] LUO, Wenhan ; ZHAO, Xiaowei ; KIM, Tae-Kyun: Multiple Object Tracking: A Review. In: *CoRR* abs/1409.7618 (2014). <http://arxiv.org/abs/1409.7618>
- [118] WILKING, Benjamin: *Generic Sensor Data Fusion in Information Space and a New Approach to Processing Dense Sensor Data*, Universität Ulm, Diss., 2018. <http://dx.doi.org/10.18725/oparu-5388>. – DOI 10.18725/oparu-5388
- [119] HO, Y. ; LEE, R.: A Bayesian Approach to Problems in Stochastic Estimation and Control. In: *IEEE Transactions on Automatic Control* 9 (1964), Nr. 4, S. 333–339. <http://dx.doi.org/10.1109/TAC.1964.1105763>. – DOI 10.1109/TAC.1964.1105763. – ISSN 0018–9286
- [120] KALMAN, Rudolph E.: A New Approach to Linear Filtering and Prediction Problems. In: *Journal of basic Engineering* 82 (1960), Nr. 1, S. 35–45
- [121] WESTENBERGER, Antje: *Simultane Zustands- und Existenzschätzung mit chronologisch ungeordneten Sensordaten für die Fahrzeugumfelderfassung*, Universität Ulm, Diss., 2015. <http://dx.doi.org/10.18725/oparu-3216>. – DOI 10.18725/oparu-3216
- [122] MUNZ, Michael: *Generisches Sensorfusionsframework zur gleichzeitigen Zustands- und Existenzschätzung für die Fahrzeugumfeldererkennung*, Universität Ulm, Diss., 2011. <http://dx.doi.org/10.18725/oparu-1763>. – DOI 10.18725/oparu-1763
- [123] KOHLER, M.: *Using the Kalman Filter to Track Human Interactive Motion: Modelling and Initialization of the Kalman Filter for Translational Motion*. Dekanat Informatik, Univ., 1997 (Forschungsberichte des Fachbereichs Informatik der Universität Dortmund)
- [124] RONG LI, X. ; JILKOV, V. P.: Survey of Maneuvering Target Tracking. Part I. Dynamic Models. In: *IEEE Transactions on Aerospace and Electronic Systems* 39 (2003), Nr. 4, S. 1333–1364. <http://dx.doi.org/10.1109/TAES.2003.1261132>. – DOI 10.1109/TAES.2003.1261132. – ISSN 0018–9251
- [125] PERROLLAZ, M. ; SPALANZANI, A. ; AUBERT, D.: Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection. In: *2010 IEEE Intelligent Vehicles Symposium*, 2010. – ISSN 1931–0587, S. 313–318
- [126] PHAM, Minh-Hai ; DE MOUZON, O. ; CHUNG, E. ; DUMONT, A.-G.: Applicability of road safety indicators to assess driving risks under Swiss road conditions. In: *Proceedings of the 7th STRC Swiss Transport Research* (2007)
- [127] ISMAIL, Karim ; SAYED, Tarek ; SAUNIER, Nicolas ; LIM, Clark: Automated Analysis of Pedestrian-Vehicle Conflicts Using Video Data. In: *Transportation research record* 2140 (2009), Nr. 1, S. 44–54
-

- [128] ARLEN, Timothy: *Understanding the mAP Evaluation Metric for Object Detection*. <https://medium.com/@timothycarlen/understanding-the-map-evaluation-metric-for-object-detection-a07fe6962cf3>. Version: 2018. – Aufgerufen am 4.11.19
- [129] CORDTS, Marius ; OMRAN, Mohamed ; RAMOS, Sebastian ; REHFELD, Timo ; ENZWEILER, Markus ; BENENSON, Rodrigo ; FRANKE, Uwe ; ROTH, Stefan ; SCHIELE, Bernt: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: *CoRR* abs/1604.01685 (2016). <http://arxiv.org/abs/1604.01685>
- [130] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: *CoRR* abs/1502.01852 (2015). <http://arxiv.org/abs/1502.01852>
- [131] RUSSAKOVSKY, Olga ; DENG, Jia ; SU, Hao ; KRAUSE, Jonathan ; SATHEESH, Sanjeev ; MA, Sean ; HUANG, Zhiheng ; KARPATHY, Andrej ; KHOSLA, Aditya ; BERNSTEIN, Michael ; BERG, Alexander C. ; FEI-FEI, Li: ImageNet Large Scale Visual Recognition Challenge. In: *International Journal of Computer Vision (IJCV)* 115 (2015), Nr. 3, S. 211–252. <http://dx.doi.org/10.1007/s11263-015-0816-y>. – DOI 10.1007/s11263-015-0816-y
- [132] WOODMAN, Oliver J.: An Introduction to Inertial Navigation. (2007), Nr. UCAM-CL-TR-696. <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>
- [133] HARMS, Dirk-Antonio: *Unfalldatenspeicher (UDS) als möglicher Beitrag zur Verkehrssicherheit im Meinungsbild Jugendlicher und Heranwachsender*, Diss., 2003. [https://publikationsserver.tu-braunschweig.de/receive/dbbs\\_mods\\_00001480](https://publikationsserver.tu-braunschweig.de/receive/dbbs_mods_00001480)
- [134] REHDER, E. ; KINZIG, C. ; BENDER, P. ; LAUER, M.: Online Stereo Camera Calibration From Scratch. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, S. 1694–1699
- [135] BOGDAN, Oleksandr ; ECKSTEIN, Viktor ; RAMEAU, François ; BAZIN, Jean C.: DeepCalib: A Deep Learning Approach for Automatic Intrinsic Calibration of Wide Field-of-View Cameras. In: *European Conference on Visual Media Production, London, 2018*
- [136] MAHJOURIAN, Reza ; WICKE, Martin ; ANGELOVA, Anelia: Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In: *CoRR* abs/1802.05522 (2018). <http://arxiv.org/abs/1802.05522>
- [137] KENDALL, Alex ; MARTIROSYAN, Hayk ; DASGUPTA, Saumitro ; HENRY, Peter ; KENNEDY, Ryan ; BACHRACH, Abraham ; BRY, Adam: End-to-End Learning of Geometry and Context for Deep Stereo Regression. In: *CoRR* abs/1703.04309 (2017). <http://arxiv.org/abs/1703.04309>

- [138] GEIGER, Andreas ; LENZ, Philip ; URTASUN, Raquel: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012
- [139] ZBONTAR, Jure ; LECUN, Yann: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. In: *CoRR* abs/1510.05970 (2015). <http://arxiv.org/abs/1510.05970>
- [140] SIMONELLI, Andrea ; BULÒ, Samuel R. ; PORZI, Lorenzo ; LÓPEZ-ANTEQUERA, Manuel ; KONTSCHIEDER, Peter: Disentangling Monocular 3D Object Detection. In: *CoRR* abs/1905.12365 (2019). <http://arxiv.org/abs/1905.12365>
- [141] XU, B. ; CHEN, Z.: Multi-level Fusion Based 3D Object Detection from Monocular Images. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, S. 2345–2353
- [142] LI, Peiliang ; CHEN, Xiaozhi ; SHEN, Shaojie: Stereo R-CNN based 3D Object Detection for Autonomous Driving. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, S. 7644–7652
- [143] WANG, Yan ; CHAO, Wei-Lun ; GARG, Divyansh ; HARIHARAN, Bharath ; CAMPBELL, Mark ; WEINBERGER, Kilian Q.: Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), S. 8437–8445
- [144] YOU, Yurong ; WANG, Yan ; CHAO, Wei-Lun ; GARG, Divyansh ; PLEISS, Geoff ; HARIHARAN, Bharath ; CAMPBELL, Mark ; WEINBERGER, Kilian Q.: Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. In: *International Conference on Learning Representations*, 2020
- [145] SCHUSTER, René ; BAILER, Christian ; WASENMÜLLER, Oliver ; STRICKER, Didier: Combining Stereo Disparity and Optical Flow for Basic Scene Flow. In: *Commercial Vehicle Technology 2018*. Springer, 2018, S. 90–101
- [146] MENZE, Moritz ; GEIGER, Andreas: Object Scene Flow for Autonomous Vehicles. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015
- [147] MA, Wei-Chiu ; WANG, Shenlong ; HU, Rui ; XIONG, Yuwen ; URTASUN, Raquel: Deep Rigid Instance Scene Flow. In: *CoRR* abs/1904.08913 (2019). <http://arxiv.org/abs/1904.08913>
- [148] CVIŠIĆ, Igor ; ĆESIĆ, Josip ; MARKOVIĆ, Ivan ; PETROVIĆ, Ivan: SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. In: *Journal of field robotics* 35 (2018), Nr. 4, S. 578–595
- [149] SCHUBERT, R. ; RICHTER, E. ; WANIELIK, G.: Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking. In: *2008 11th International Conference on Information Fusion*, 2008, S. 1–6

- [150] RIPPEL, Benjamin ; GRUENER, Jan ; MARKER, Stefanie: Using Convolutional Neural Networks for Cycling Infrastructure Classification. In: *International Cycling Safety Conference, Davis* (2017)
- [151] SIMONYAN, Karen ; ZISSERMAN, Andrew: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *CoRR* abs/1409.1556 (2015)
- [152] MEYER, A. ; SALSCHIEDER, N. O. ; ORZECZOWSKI, P. F. ; STILLER, C.: Deep Semantic Lane Segmentation for Mapless Driving. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, S. 869–875
- [153] BEHRENDT, Karsten ; NOVAK, Libor: A Deep Learning Approach to Traffic Lights: Detection, Tracking, and Classification. In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on IEEE*
- [154] FREGIN, A. ; MÜLLER, J. ; KREEL, U. ; DIETMAYER, K.: The DriveU Traffic Light Dataset: Introduction and Comparison with Existing Datasets. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018. – ISSN 2577–087X, S. 3376–3383
- [155] HOUBEN, Sebastian ; STALLKAMP, Johannes ; SALMEN, Jan ; SCHLIPSING, Marc ; IGEL, Christian: Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. In: *International Joint Conference on Neural Networks*, 2013
- [156] SAHDEV, Raghavender: Free Space Estimation using Occupancy Grids and Dynamic Object Detection. In: *CoRR* abs/1708.04989 (2017). <http://arxiv.org/abs/1708.04989>
- [157] SOQUET, Nicolas ; PERROLLAZ, Mathias ; LABAYRADE, Raphaël ; AUBERT, Didier: Free Space Estimation for Autonomous Navigation. In: *ICVS 2007*, 2007
- [158] ROSER, M. ; MOOSMANN, F.: Classification of Weather Situations on Single Color Images. In: *2008 IEEE Intelligent Vehicles Symposium*, 2008, S. 798–803
- [159] VILLARREAL GUERRA, J. C. ; KHANAM, Z. ; EHSAN, S. ; STOLKIN, R. ; MCDONALD-MAIER, K.: Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks. In: *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2018, S. 305–310
- [160] NOLTE, Marcus ; KISTER, Nikita ; MAURER, Markus: Assessment of Deep Convolutional Neural Networks for Road Surface Classification. In: *CoRR* abs/1804.08872 (2018). <http://arxiv.org/abs/1804.08872>
- [161] ROYCHOWDHURY, S. ; ZHAO, M. ; WALLIN, A. ; OHLSSON, N. ; JONASSON, M.: Machine Learning Models for Road Surface and Friction Estimation using Front-Camera Images. In: *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, S. 1–8

- [162] ŠABANOVIČ, Eldar ; ŽURĀULIS, Vidas ; PRENTKOVSKIS, Olegas ; SKRICKIJ, Viktor: Identification of Road-Surface Type Using Deep Neural Networks for Friction Coefficient Estimation. In: *Sensors* 20 (2020), S. 612. <http://dx.doi.org/10.3390/s20030612>. – DOI 10.3390/s20030612
- [163] PORAV, Horia ; BRULS, Tom ; NEWMAN, Paul: I Can See Clearly Now : Image Restoration via De-Raining. In: *CoRR* abs/1901.00893 (2019). <http://arxiv.org/abs/1901.00893>



# Abbildungsverzeichnis

2.1	Das menschliche Auge. . . . .	20
2.2	Die perspektivische Transformation des Lochkameramodells . . . . .	22
2.3	Die allgemeine Epipolareometrie. . . . .	23
2.4	Das achsparallele Stereosystem . . . . .	24
2.5	Die Verzerrung der Kameraabbildung. . . . .	26
2.6	Tiefenfehler des verwendeten Stereokamerasystems. . . . .	28
2.7	Darstellung der Prozesskette: von Kamerarohdaten zu 3D-Punkten. . . . .	29
3.1	„FESTA V“ . . . . .	33
3.2	Der generalisierte Workflow . . . . .	36
3.3	Die Prozessketten der beiden verfolgten Ansätze. . . . .	37
3.4	Ein Kamerabild und die zugehörige Disparitätenkarte. . . . .	40
3.5	Szenen aus aufgezeichneten NCS-Daten. . . . .	42
3.6	Ebenen einer dreidimensionalen Szene . . . . .	45
3.7	<i>v-disparity</i> mit Straßenebene und Fahrzeug . . . . .	45
3.8	Nicht in der <i>v-disparity</i> darstellbare Ebenen . . . . .	46
3.9	Prozess der Objektdetektion anhand der <i>u-disparity</i> . . . . .	50
3.10	Auswirkung des Wankens auf ein Objekt in der <i>u-disparity</i> . . . . .	52
3.11	Einfluss einer Rotation auf die detektierte Objektbreite . . . . .	53
3.12	Schematische Darstellung eines künstlichen neuronalen Netzes . . . . .	56
3.13	<i>LeNet-5</i> als Beispiel für die Architektur eines CNN . . . . .	57
3.14	Abgrenzung der Objekterkennung von verwandten Aufgaben . . . . .	58
3.15	Objektinstanzsegmentierung mit <i>Mask R-CNN</i> . . . . .	60
3.16	Das Kalman-Filter . . . . .	63
4.1	Positionsbestimmung bei bekannten Objekten. . . . .	72
4.2	Berechnung der TP und FP des regelbasierten Ansatzes . . . . .	75
4.3	Berechnung der FN des regelbasierten Ansatzes . . . . .	76
4.4	Performanz der Objektdetektion in Abhängigkeit des <i>u-disparity</i> Schwellenwerts . . . . .	78
4.5	Schwellenwert der minimalen Objektbreite in Bild- und Weltkoordinaten . . . . .	79
4.6	Performanz der Objektdetektion in Abhängigkeit der minimalen Konturbreite . . . . .	80
4.7	Verteilung der <i>True Positives</i> , <i>False Positives</i> und <i>False Negatives</i> über die Objektbreite . . . . .	82
4.8	<i>Heatmap</i> der <i>True Positives</i> , <i>False Positives</i> und <i>False Negatives</i> über die Bildbreite und Entfernung . . . . .	83
4.9	TTC-Validationsversuch mit Variation des lateralen Kameraversatzes. . . . .	88

---

4.10	TTC-Validationsversuch mit Variation der Winkelausrichtung von Kamera und Trajektorie. . . . .	89
4.11	TTC und Standardabweichung aus UDS-Daten. . . . .	91
4.12	Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz; erster Versuch der TTC-Validationsversuche, Szenario 1. . . . .	92
4.13	Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz; erster Versuch der TTC-Validationsversuche, Szenario 1. . . . .	93
4.14	Geschwindigkeit und Entfernung des Objekts berechnet mit dem regelbasierten Ansatz; erster Versuch der TTC-Validationsversuche, Szenario 1. . . . .	94
4.15	Geschwindigkeit und Entfernung des Objekts berechnet mit dem modellbasierten Ansatz; erster Versuch der TTC-Validationsversuche, Szenario 1. . . . .	95
4.16	TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz; erster Versuch der TTC-Validationsversuche, Szenario 1. . . . .	95
4.17	TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz; erster Versuch der TTC-Validationsversuche, Szenario 1. . . . .	96
4.18	Sensitivität der Kollisionsdetektion für verschiedene Objektdistanzen. . . . .	98
4.19	PET-Validationsversuch. . . . .	101
4.20	PET-Validationsversuch Bildsequenz. . . . .	102
4.21	Trajektorie des Objekts eines PET-Versuchs, berechnet mit dem regelbasierten Ansatz. . . . .	103
4.22	Trajektorie des Objekts eines PET-Versuchs, berechnet mit dem modellbasierten Ansatz. . . . .	103
4.23	Geschwindigkeit und Entfernung des Objekts eines PET-Versuchs berechnet mit dem regelbasierten Ansatz. . . . .	104
4.24	Geschwindigkeit und Entfernung des Objekts eines PET-Versuchs berechnet mit dem modellbasierten Ansatz. . . . .	105
4.25	PET und Grundbedingung des Objekts berechnet mit dem regelbasierten Ansatz. . . . .	105
4.26	PET und Grundbedingung des Objekts berechnet mit dem modellbasierten Ansatz. . . . .	106
4.27	Szenario: Fußgänger*innen. . . . .	109
4.28	Szenario: Fußgänger*innen. Trajektorien des modellbasierten Ansatzes. . . . .	110
4.29	Szenario: Fußgänger*innen. TTC- und PET-Signale des modellbasierten Ansatzes. . . . .	111
4.30	Szenario: Fußgänger*innen. Messungen des regelbasierten Ansatzes. . . . .	112
4.31	Szenario: Fußgänger*innen. TTC- und PET-Signale des regelbasierten Ansatzes. . . . .	113
4.32	Szenario: Kreuzender Radfahrer. . . . .	114
4.33	Szenario: Kreuzender Radfahrer. Trajektorien des modellbasierten Ansatzes. . . . .	115
4.34	Szenario: Kreuzender Radfahrer. PET-Signale des modellbasierten Ansatzes. . . . .	115
4.35	Szenario: Überholendes Fahrrad. . . . .	115

4.36 Szenario: Überholendes Fahrrad. Trajektorien des modellbasierten Ansatzes. . . . . 116

4.37 Szenario: Überholendes Fahrrad. PET-Signale des modellbasierten Ansatzes. . . . . 117



# Abkürzungsverzeichnis

<b>Abk.</b>	<b>Abkürzung</b>
ACC	<i>Adaptive Cruise Control</i>
AD	<i>Autonomous Driving</i>
ADAS	<i>Advanced Driver Assistance Systems</i>
AEB	<i>Automatic Emergency Braking</i>
AP	<i>Average Precision</i>
CNN	<i>Convolutional Neural Network</i>
COCO	<i>Common Objects in Context</i>
CV	<i>Constant Velocity</i>
DAS	Datenaufzeichnungssystem
EBA	<i>Event Based Analysis</i>
EKF	<i>Extended Kalman-Filter</i>
FN	<i>False Negative</i>
FOT	<i>Field Operational Test</i>
FP	<i>False Positive</i>
GNSS	Globales Navigationssystem
ICP	<i>Iterative Closest Point</i>
INS	Trägheitsnavigationssystem
IoU	<i>Intersection Over Union</i>
KITTI	<i>Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago (TTI-C)</i>
LDW	<i>Lane Departure Warning</i>
LiDAR	<i>Light Detection and Ranging</i>
mAP	<i>Mean Average Precision</i>
MOT	<i>Multiple Object Tracking</i>
NCS	<i>Naturalistic Cycling Study</i>

NDS	<i>Naturalistic Driving Study</i>
NN	<i>Nearest Neighbor</i>
PDA	Probabilistische Datenassoziation
PET	<i>Post Encroachment Time</i>
PI	<i>Performance Indicator</i>
RaDAR	<i>Radio Detection and Ranging</i>
R-CNN	<i>Region Based Convolutional Neural Network</i>
RANSAC	<i>Random Sample Consensus</i>
RMSE	<i>Root Mean Square Error</i>
ROI	<i>Region of Interest</i>
TTC	<i>Time To Collision</i>
UDS	Unfalldatenschreiber
UKF	<i>Unscented Kalman-Filter</i>

# Symbolverzeichnis

## Grundlagen der Stereovision

Symbol	Einheit	Bezeichnung
$a_u$	px	Nach der internen Transformation skalierte horizontale Brennweite
$a_v$	px	Nach der internen Transformation skalierte vertikale Brennweite
<b>A</b>	diverse	Intrinsische Matrix
$B$	m	Basislänge
$c$	-	Kamerahauptpunkt
$C$	-	Optisches Zentrum (Brennpunkt)
$\delta$	px	Disparität
<b>D</b>	diverse	Extrinsische Matrix
<b>e</b>	-	Epipol
<b>E</b>	-	Essential-Matrix
$f$	m	Brennweite
$\gamma$	-	Scherungsfaktor der internen Transformation
<b>I</b>	-	Epipolarlinie
$I$	-	Bildebene
$k_u$	px/m	Horizontaler Skalierungsfaktor der internen Transformation
$k_v$	px/m	Vertikaler Skalierungsfaktor der internen Transformation
$\tilde{\mathbf{m}}' = [U, V, S]$	m	Punkt auf der Bildebene in homogenen Koordinaten
$\mathbf{m}' = [x, y]$	m	Punkt auf der Bildebene in kartesischen Koordinaten
$\tilde{\mathbf{m}}$	px	Punkt auf der Bildebene in diskreten homogenen Koordinaten

$\mathbf{m} = [u, v]$	px	Punkt auf der Bildebene in diskreten kartesischen Koordinaten
$\mathbf{M} = [X, Y, Z]^T$	-	Punkt im dreidimensionalen Raum
$\tilde{\mathbf{M}}_{\mathbf{w}} = [X_w, Y_w, Z_w, 1]$	m	Punkt im dreidimensionalen projektiven Raum, dargestellt im Weltkoordinatensystem in homogenen Koordinaten
$\tilde{\mathbf{M}}_{\mathbf{c}} = [X_c, Y_c, Z_c, 1]$	m	Punkt im dreidimensionalen projektiven Raum, dargestellt im Kamerakoordinatensystem in homogenen Koordinaten
$O$	-	Weltkoordinatensystem
$\mathbf{P}'$	-	Perspektivische Projektionsmatrix
$\mathbf{P}$	diverse	Allgemeine Projektionsmatrix
$\mathcal{P}^2$	-	Zweidimensionaler projektiver Raum
$\mathcal{P}^3$	-	Dreidimensionaler projektiver Raum
$\pi$	-	Epipolarebene
$\mathbf{Q}_{\mathbf{R}}$	diverse	Reprojektionsmatrix
$\mathbf{R}$	-	Rotationsmatrix
$\mathbf{R}^T$	-	Transponierte einer beliebigen Matrix $\mathbf{R}$
$S$	-	Skalierungsfaktor der homogenen Bildkoordinaten
$\mathbf{t}$	m	Translationsvektor
$[\mathbf{t}]_{\times}$	-	Matrixschreibweise des Kreuzprodukts mit $\mathbf{t}$
$u_0$	px	Horizontale Bildkoordinate des Kamerahauptpunkts
$v_0$	px	Vertikale Bildkoordinate des Kamerahauptpunkts
$W$	-	Skalierungsfaktor der homogenen Koordinaten im dreidimensionalen Raum
$Z_C$	-	Optische Achse

## Kalman-Filter

Symbol	Einheit	Bezeichnung
$\Delta t$	s	Zu prädizierende Zeitdifferenz
$f(\hat{x}_{k-1 k-1}, u_k)$	diverse	Nichtlineare Abbildung des Zustandsübergangs
$F_{k-1}^J$	diverse	Jacobi-Matrix der nichtlinearen Abbildung des Zustandsübergangs
$F_k$	diverse	Zustandsübergangsmatrix
$\gamma_k$	px	Innovation der Messung zum Zeitpunkt $t_k$
$G_k$	-	Steuermatrix
$h(\hat{x}_{k k-1})$	diverse	Nichtlineare Abbildung des Messmodells
$H_k$	diverse	Beobachtungsmatrix
$H_k^J$	diverse	Jacobi-Matrix der nichtlinearen Abbildung der Messmodells
$I_3$	-	$3 \times 3$ Einheitsmatrix
$K_k$	diverse	Filterverstärkung
$\nu_k$	m/s <sup>2</sup>	Prozessrauschen
$\omega_k$	px	Messrauschen
$P_{k k-1}$	diverse	Kovarianzmatrix des Systemzustands, prädiziert von $t_{k-1}$ nach $t_k$ ( <i>Prior</i> )
$P_{k k}$	diverse	Kovarianzmatrix des Systemzustands nach Korrektur mit der Messung $z_k$ ( <i>Posterior</i> )
$Q_k$	diverse	Kovarianzmatrix des Prozessrauschens
$R_k$	px <sup>2</sup>	Kovarianzmatrix des Messrauschens
$S_k$	px <sup>2</sup>	Kovarianzmatrix der Innovation
$t_k$	s	Diskreter Zeitpunkt
$u_k$	-	Steuergröße
$V_{x/y/z}$	m/s	Objektgeschwindigkeit in X- / Y- bzw. Z-Richtung
$x_k$	diverse	Systemzustand zum Zeitpunkt $t_k$
$\hat{x}_{k k-1}$	diverse	Mittelwert des Systemzustands, prädiziert von $t_{k-1}$ nach $t_k$ ( <i>Prior</i> )

$\hat{\mathbf{x}}_{\mathbf{k} \mathbf{k}}$	diverse	Mittelwert des Systemzustands nach Korrektur mit der Messung $z_k$ ( <i>Posterior</i> )
$\hat{\mathbf{z}}_{\mathbf{k} \mathbf{k}-1}$	px	In den Messraum transformierter Mittelwert des Systemzustands $\hat{\mathbf{x}}_{\mathbf{k} \mathbf{k}-1}$
$\mathbf{Z}_3$	-	$3 \times 3$ Nullmatrix
$Z = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$	px	Messreihe zu diskreten Zeitpunkten

### Konfliktindikatoren

Symbol	Einheit	Bezeichnung
$b_{ego}$	m	Breite des Versuchsfahrrads
$b_{obj}$	m	Breite des Objekts
$\Delta t_{ego}$	s	Zeitdifferenz bis das eigene Fahrrad die Konfliktzone erreicht
$\Delta t_{obj}$	s	Zeitdifferenz bis das Objekt die Konfliktzone verlässt
$\Delta X_{Konfliktzone}$	m	Abstand des Objekts vom Konfliktzonenrand
$d_z$	m	Distanz zwischen Kamerakoordinatensystem und Fahrradfront
$PET$	s	<i>Post Encroachment Time</i>
$TTC$	s	<i>Time To Collision</i>
$V_{ego}$	m/s	Eigengeschwindigkeit des Versuchsfahrrads
$X_0$	m	Position des Objekts in X-Richtung zum Zeitpunkt der Berechnung der TTC
$X_{Konfliktzone}$	m	X-Koordinate des rechten Konfliktzonenrands bzw. halbe Zonenbreite
$Z_{Konfliktzone}$	m	Z-Koordinate der Konfliktzone

## UDS-Daten

Symbol	Einheit	Bezeichnung
$\epsilon(t)$	m/s <sup>2</sup>	Weißes Rauschen welches das Beschleunigungssignal des UDS überlagert
$f_s$	Hz	Abtastrate des UDS
$\mathcal{N}(0, \sigma_a^2)$	-	Normalverteilung mit Erwartungswert 0 und Varianz $\sigma_a^2$
$s_{GT}$	m	Grundwahrheit der gefahrenen Distanz eines Versuchs
$\sigma_a^2$	m <sup>2</sup> /s <sup>4</sup>	Varianz des Beschleunigungssignals
$\sigma_s(t)^2$	m <sup>2</sup>	Varianz des berechneten Wegsignals des UDS
$\sigma_v(t)^2$	m <sup>2</sup> /s <sup>2</sup>	Varianz des berechneten Geschwindigkeitssignals des UDS
$S_{Obj}$	m	Zurückzulegende Distanz bis zur Kollision
$s(t)$	m	Berechnetes Wegsignal des UDS
$t_{end}$	s	Endzeitpunkt eines Versuchs
$t$	s	Zeitsignal des UDS
$TTC_{UDS}$	s	Aus UDS-Daten berechnete TTC
$V_{Obj}$	m/s	Momentane Objektgeschwindigkeit nach UDS
$v(t)$	m/s	Berechnetes Geschwindigkeitssignal des UDS

## Sonstiges

Symbol	Einheit	Bezeichnung
$\alpha$	°	Winkel zwischen Objekttrajektorie und Kamerahauptachse
$\Delta u_{px}$	px	Objekthöhe in Bildkoordinaten
$\Delta Y$	m	Objekthöhe in Weltkoordinaten
$I_{IoU}$	-	Intervall von IoU-Schwellenwerten
$x$	m	Lateraler Versatz der Objekttrajektorie zum Kamerakoordinatensystem



# Tabellenverzeichnis

1.1	Datenmengen und Anzahl kritischer Situationen verschiedener NCS . . . . .	14
2.1	Auswirkungen der Parameter eines Stereokamerasystems. . . . .	27
2.2	Designparameter des verwendeten achsparallelen Stereosystems . . . . .	28
4.1	Ermittlung von $\sigma_a$ . . . . .	90
4.2	Fehler der Entfernungsberechnung des ersten TTC-Szenarios für beide Ansätze. . . . .	96
4.3	Fehler der Geschwindigkeitsberechnung des ersten TTC-Szenarios für beide Ansätze. . . . .	97
4.4	Fehler der TTC-Berechnung des ersten TTC-Szenarios für beide Ansätze. . . . .	97
4.5	Genauigkeit der Kollisionskursdetektion unter der Annahme von 1 m Objektbreite und Ego-Breite. . . . .	97
4.6	Genauigkeit der TTC im Nahbereich für das zweite TTC-Szenario. . . . .	99
4.7	Genauigkeit der Kollisionskursdetektion im Nahbereich für das zweite TTC-Szenario. . . . .	100
4.8	Anteil an Frames mit erfolgter Assoziation einer Messung zum <i>Track</i> . . . . .	107
4.9	Fehler der Entfernungsberechnung des PET-Szenarios für beide Ansätze. . . . .	107
4.10	Fehler der Geschwindigkeitsberechnung des PET-Szenarios für beide Ansätze. . . . .	107
4.11	Fehler der PET-Berechnung für beide Ansätze und einer quadratischen Begegnungszone mit 1 m Kantenlänge. . . . .	108
4.12	Genauigkeit der PET-Bedingungsdetektion. . . . .	108



# Anhang

## Inhaltsverzeichnis Anhang

1. Verwendete Parameter
2. Ergebnisse der Einzelversuche
  - 2.1 Ergebnisse der Einzelversuche des ersten TTC Szenarios
  - 2.2 Ergebnisse der Einzelversuche des zweiten TTC Szenarios
  - 2.3 Ergebnisse der Einzelversuche der PET Validationsversuche



## Verwendete Parameter

**Tabelle A.1**

Parametrisierung der Korrespondenzanalyse für den regelbasierten Ansatz. Die Beschreibung der einzelnen Parameter des verwendeten Algorithmus ist in [63] einsehbar.

Beschreibung	Wert	Einheit
minDisparity	0	px
numDisparities	112	px
blockSize	5	px
uniquenessRatio	4	%
speckleWindowSize	3	px
speckleRange	1	px
disp12MaxDiff	2	px
P1	70	-
P2	3594	-
mode	STEREO_SGBM_MODE_HH	-

**Tabelle A.2**

Parametrisierung der Korrespondenzanalyse für den modellbasierten Ansatz. Die Beschreibung der einzelnen Parameter des verwendeten Algorithmus ist in [63] einsehbar.

Beschreibung	Wert	Einheit
minDisparity	0	px
numDisparities	112	px
blockSize	3	px
uniquenessRatio	3	%
speckleWindowSize	15	px
speckleRange	3	px
disp12MaxDiff	1	px
P1	180	-
P2	3594	-
mode	STEREO_SGBM_MODE_HH	-

**Tabelle A.3**

Parametrisierung der regelbasierten Objektdetektion.

Beschreibung	Wert	Einheit
Minimale Konturbreite	25	px
<i>u-disparity</i> Schwellenwert	25	px

**Tabelle A.4**

Parametrisierung der Objektverfolgung für den regelbasierten Ansatz.

Beschreibung	Wert	Einheit
Prozessrauschen	$x = 2,2$ $y = 1,1$ $z = 2,2$	$m/s^2$
Messrauschen	$x = 180$ $y = 124$ $\delta = 0,76$	px
Max. Entfernung bei Initialisierung	12	m
Erlaubte konsekutive Frames ohne Assoziation	30	-
Minimale Tracklänge für Auswertung	100	-

**Tabelle A.5**

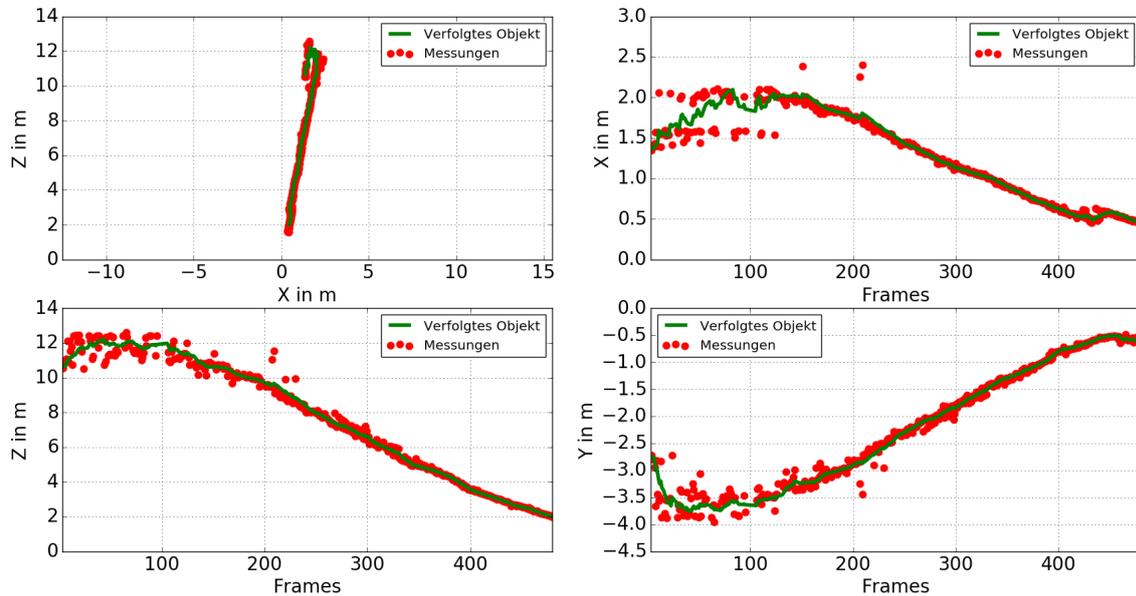
Parametrisierung der Objektverfolgung für den modellbasierten Ansatz.

Beschreibung	Wert	Einheit
Prozessrauschen	$x = 1,7$ $y = 0,85$ $z = 1,7$	m/s <sup>2</sup>
Messrauschen	$x = 180$ $y = 124$ $\delta = 0,23$	px
Max. Entfernung bei Initialisierung	16	m
Erlaubte konsekutive Frames ohne Assoziation	30	-
Minimale Tracklänge für Auswertung	100	-



## Ergebnisse der Einzelversuche

### Ergebnisse der Einzelversuche des ersten Szenarios der TTC Validationsversuche



**Abbildung A.1**

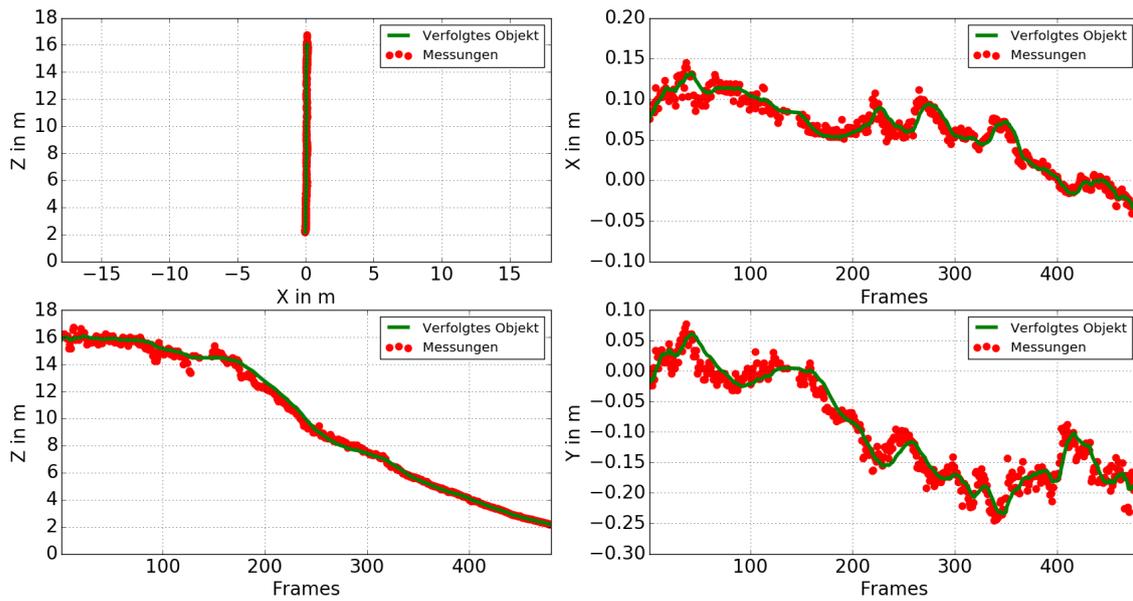
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.2**

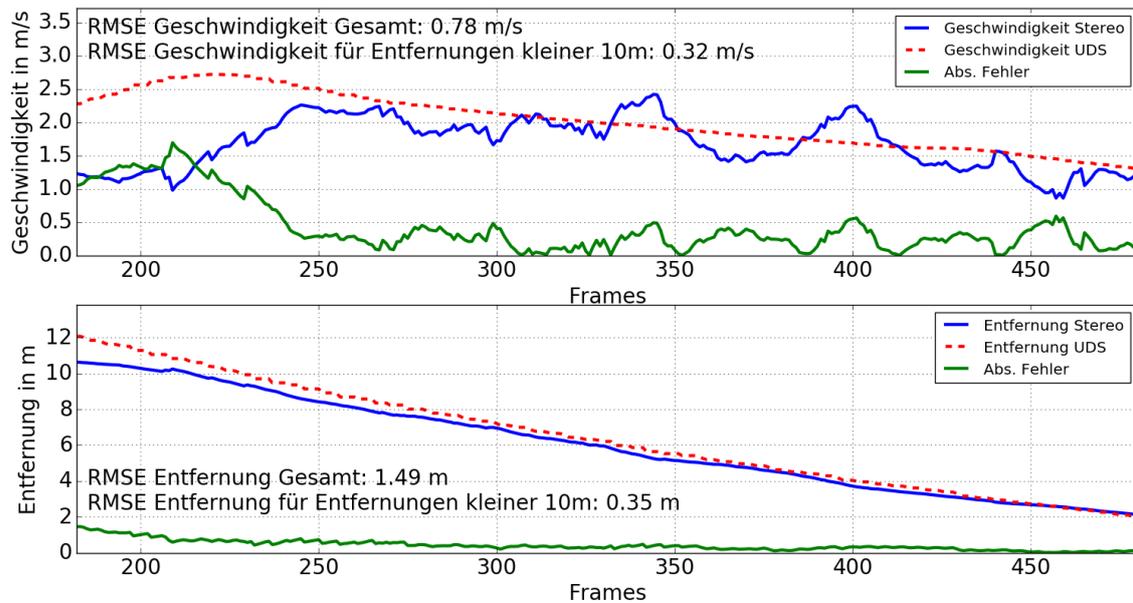
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

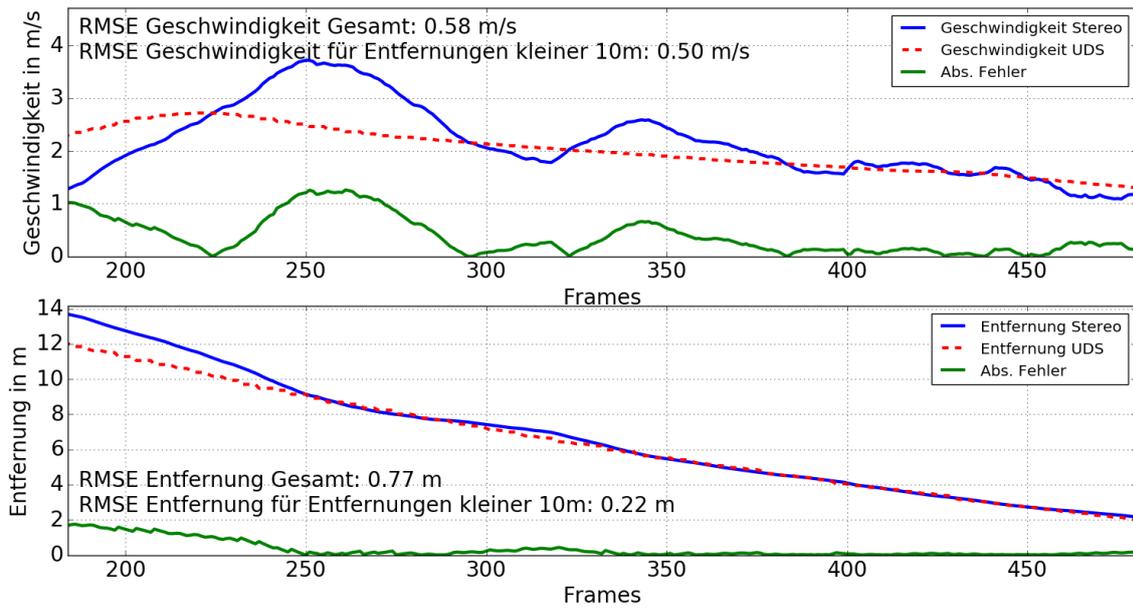
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



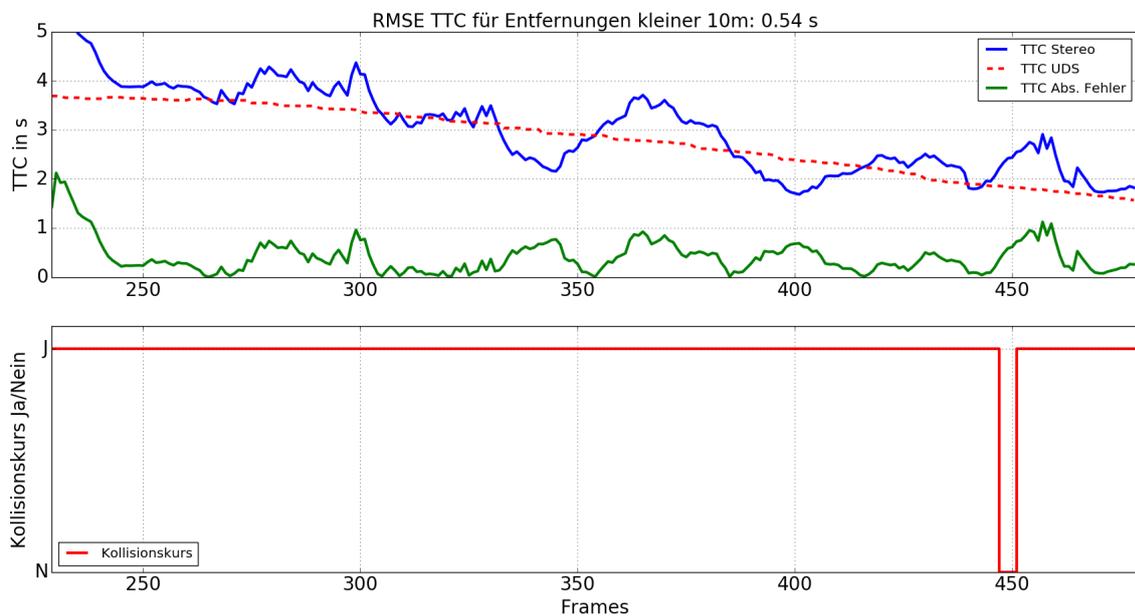
**Abbildung A.3**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.4**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

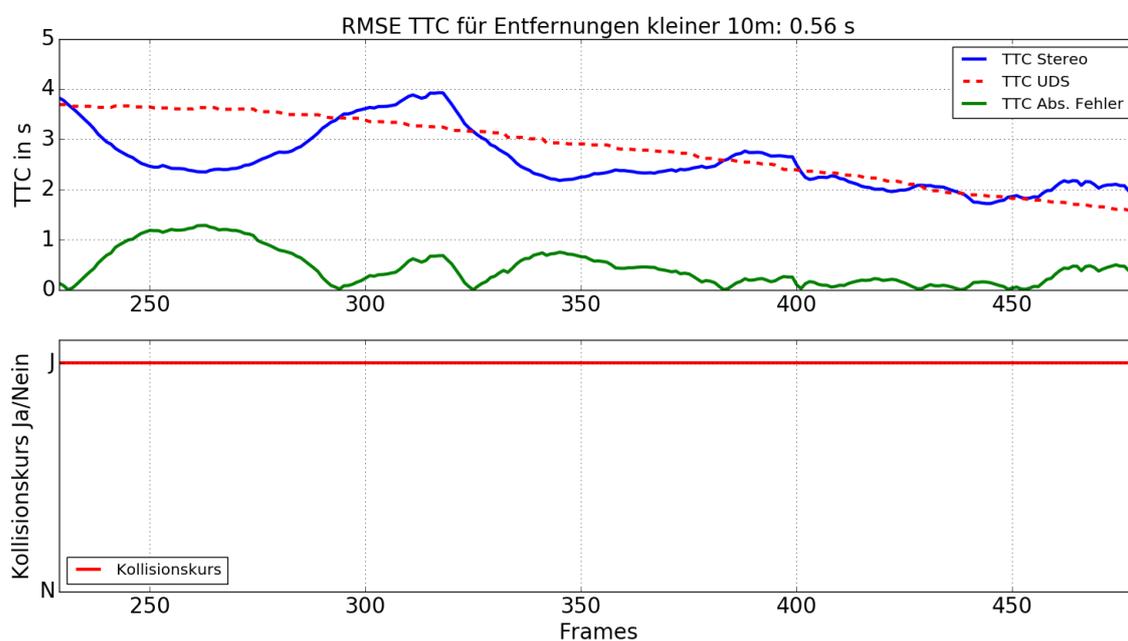


**Abbildung A.5**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

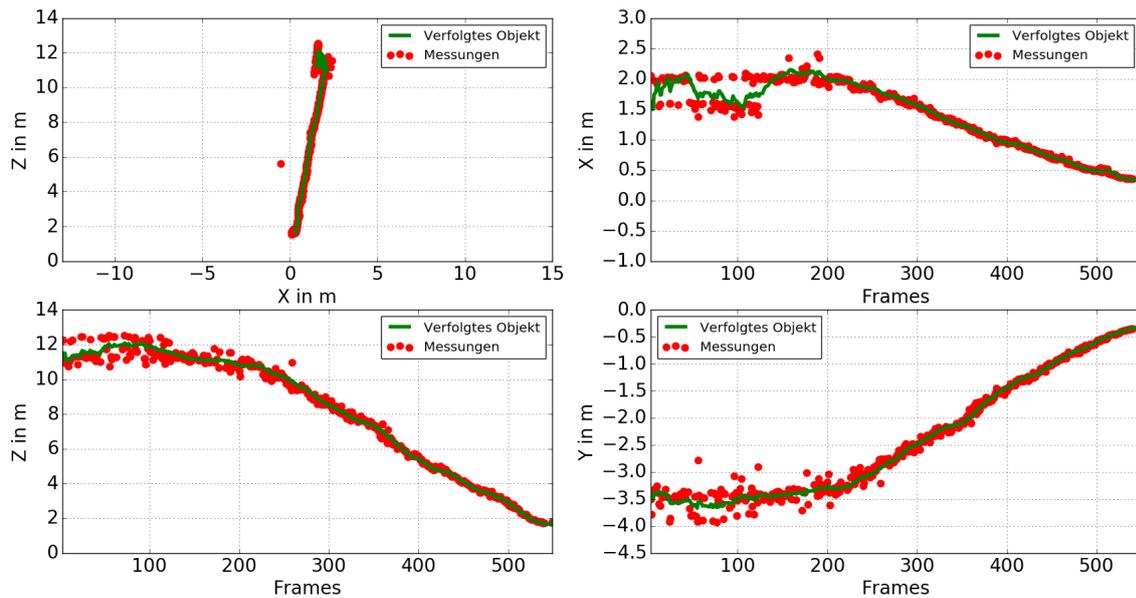


**Abbildung A.6**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.7**

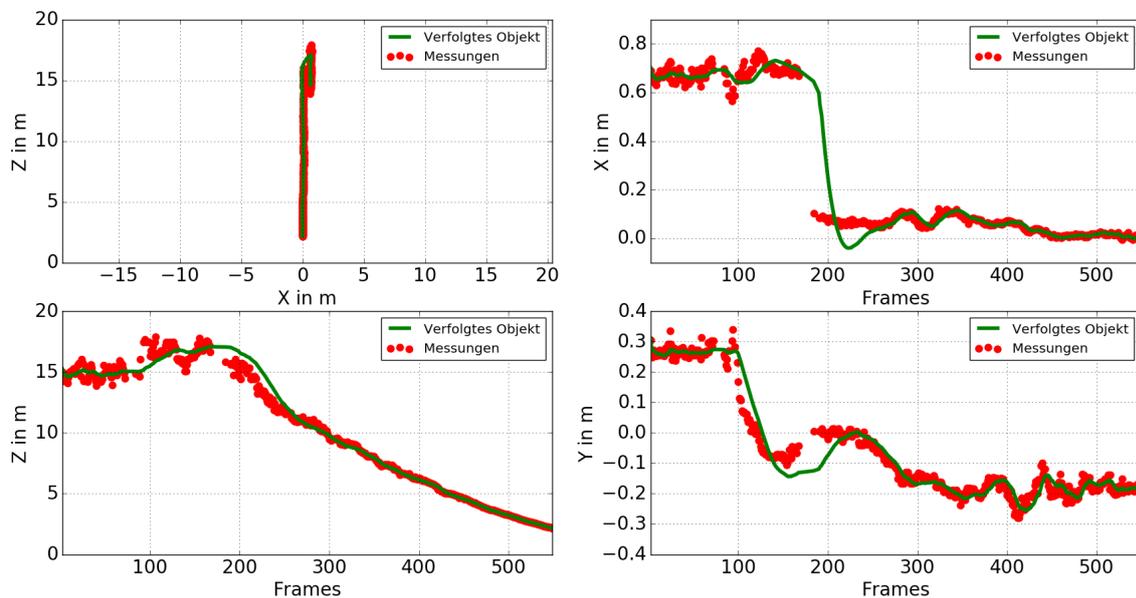
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.8**

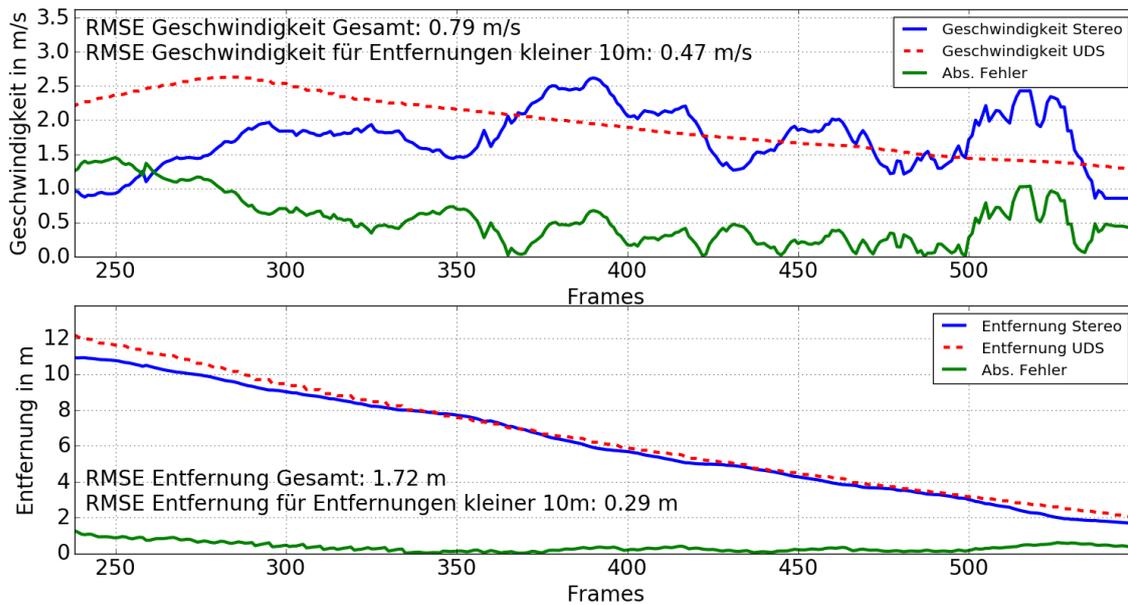
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

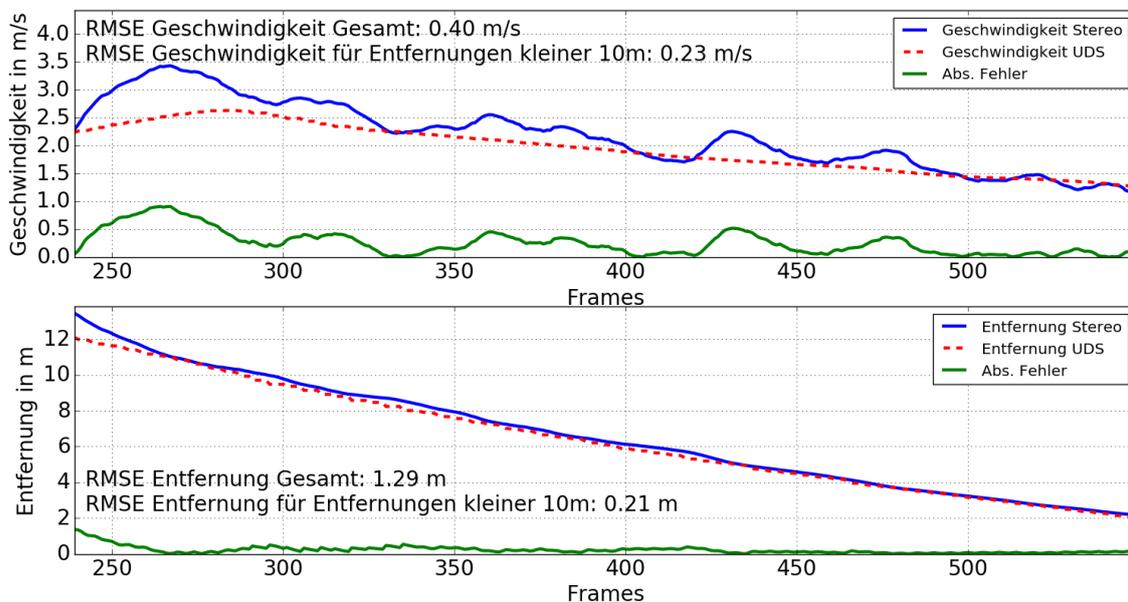
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



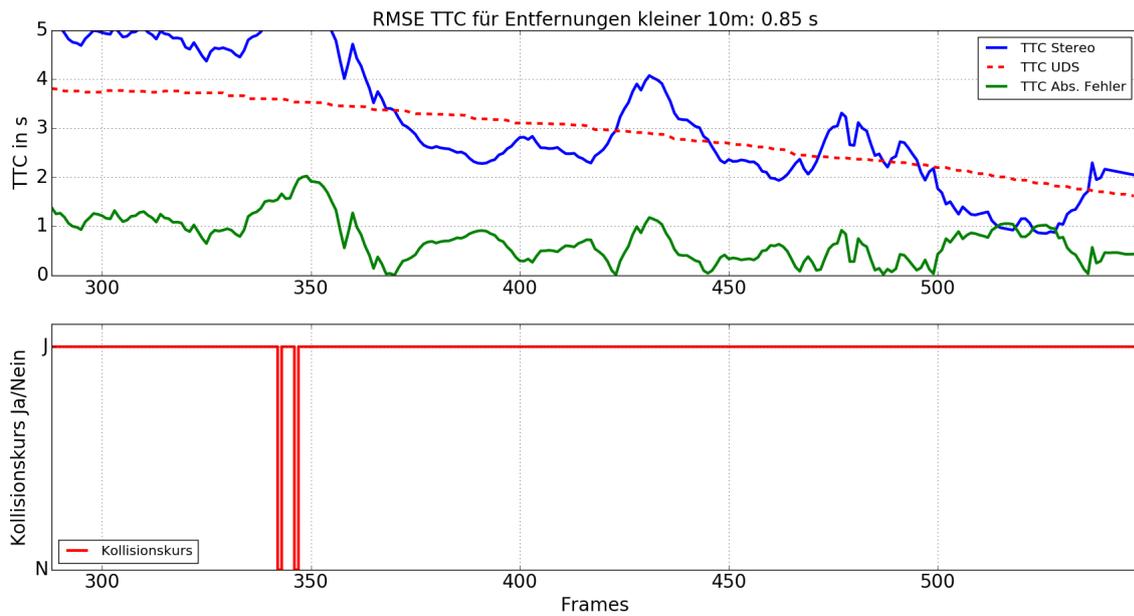
**Abbildung A.9**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.10**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

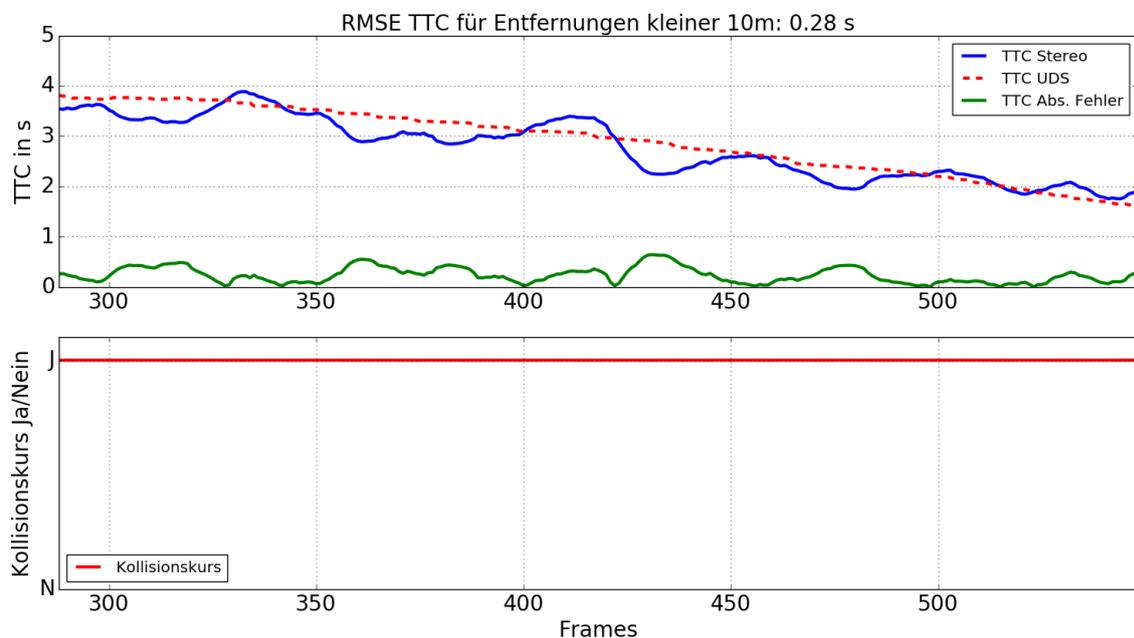


**Abbildung A.11**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

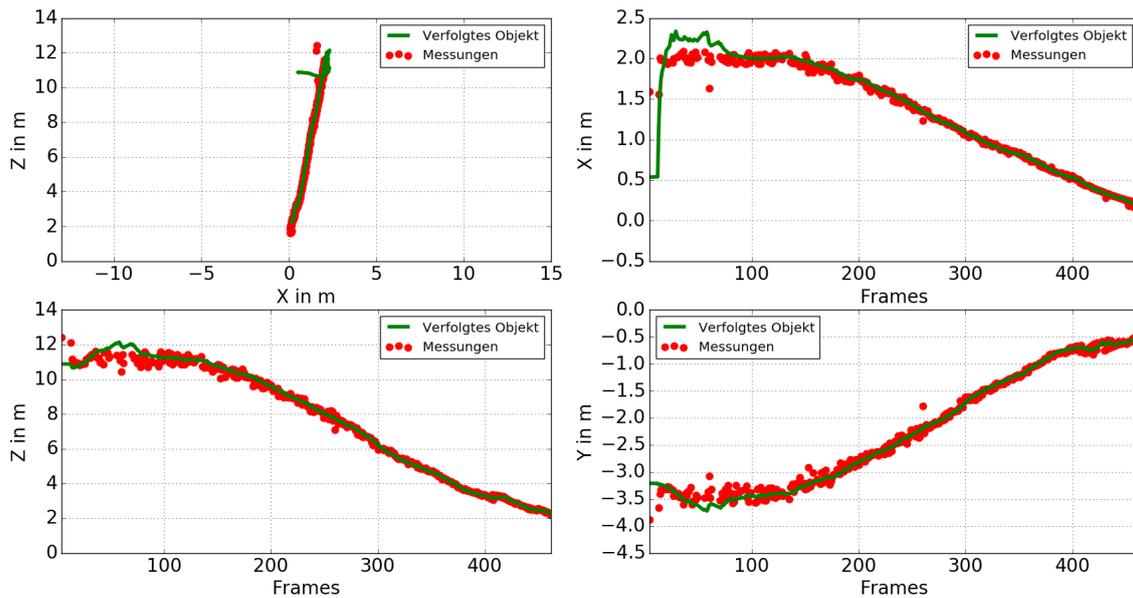


**Abbildung A.12**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.13**

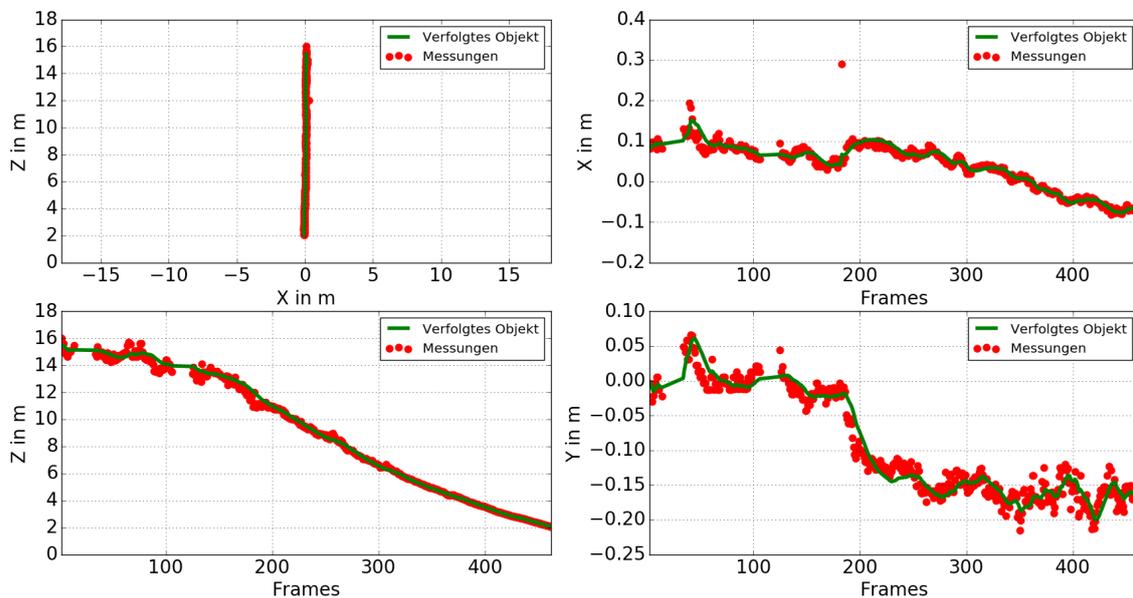
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.14**

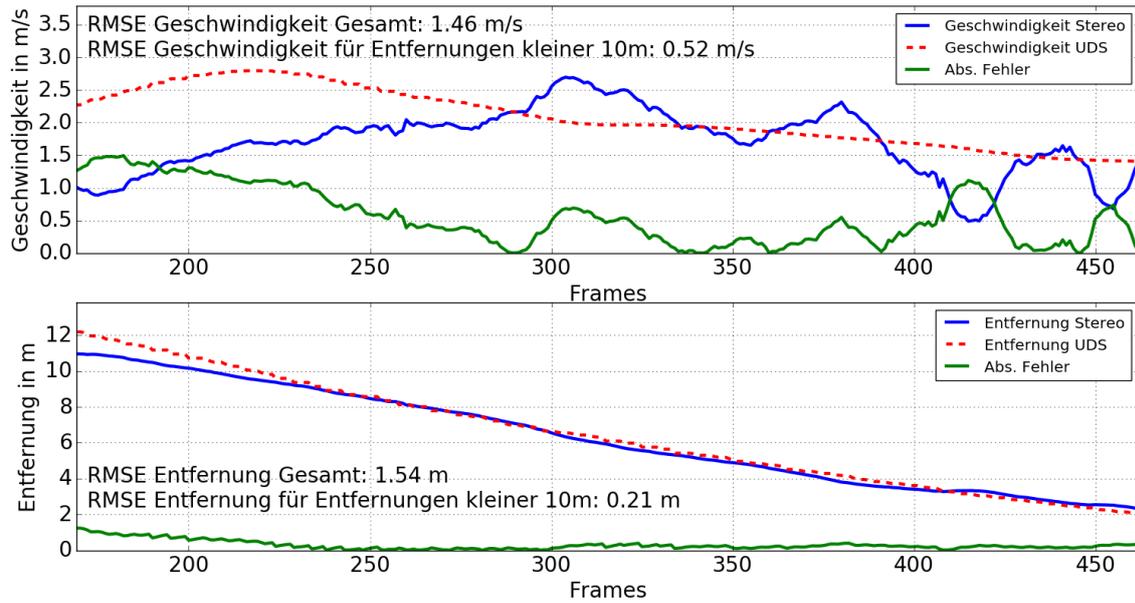
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

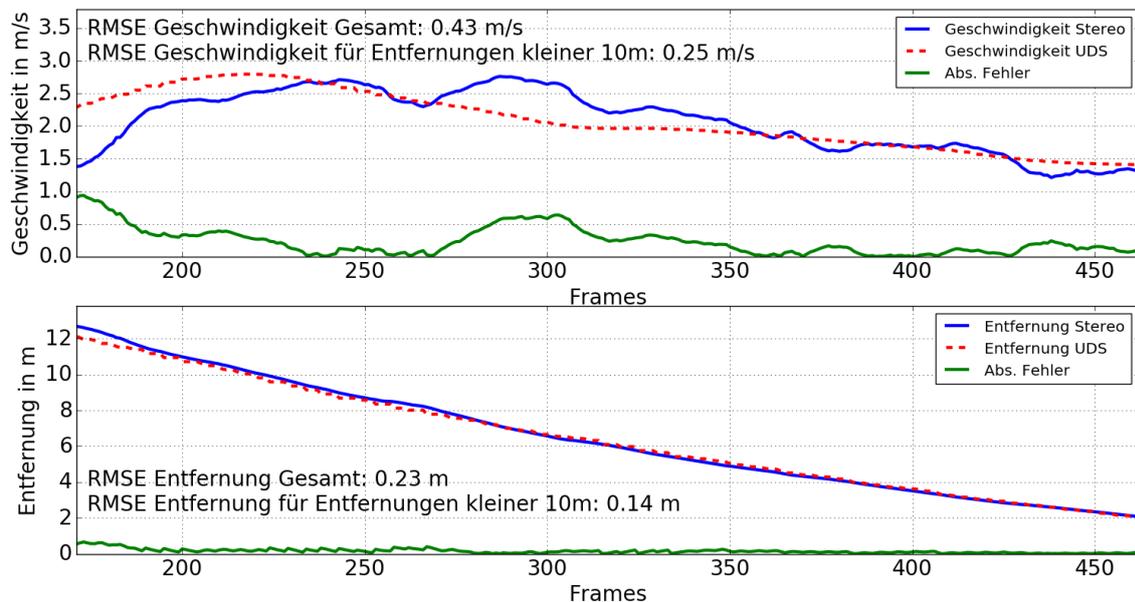
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



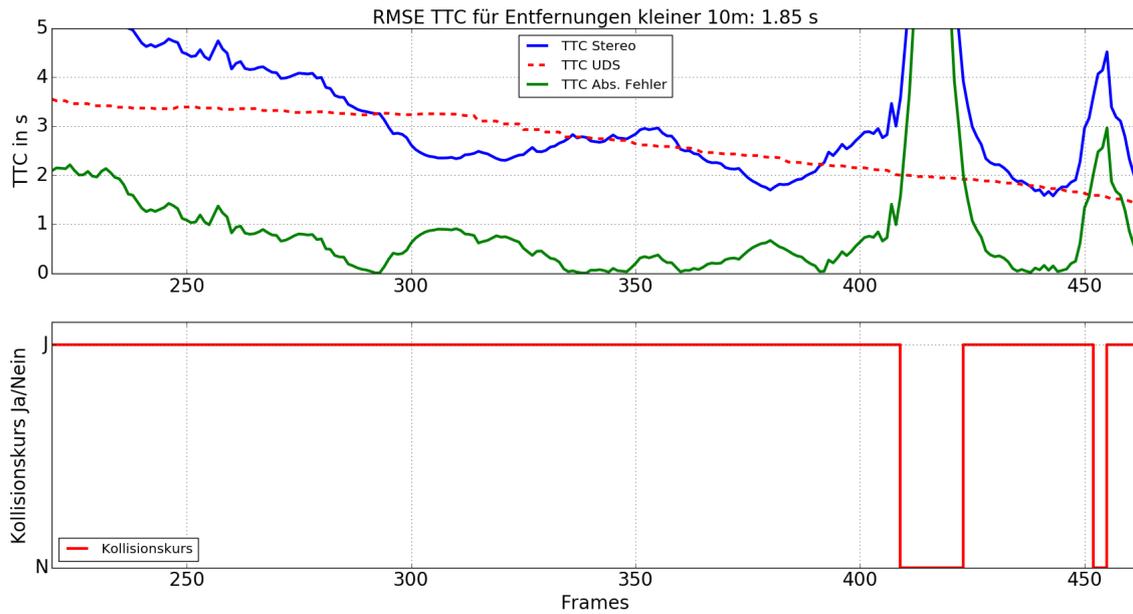
**Abbildung A.15**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.16**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

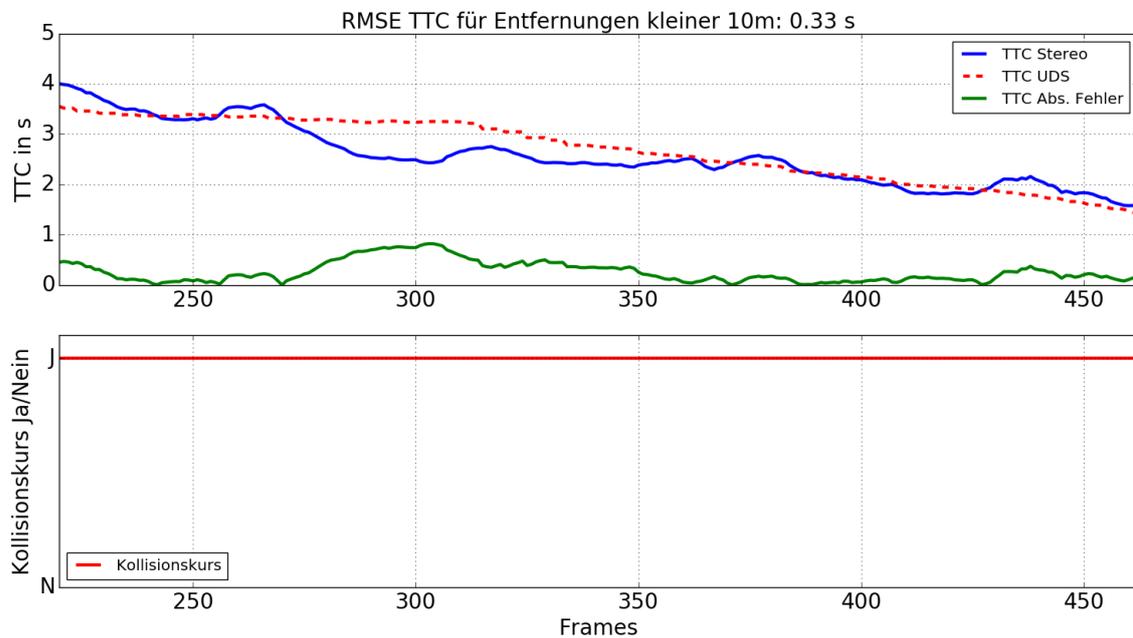


**Abbildung A.17**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

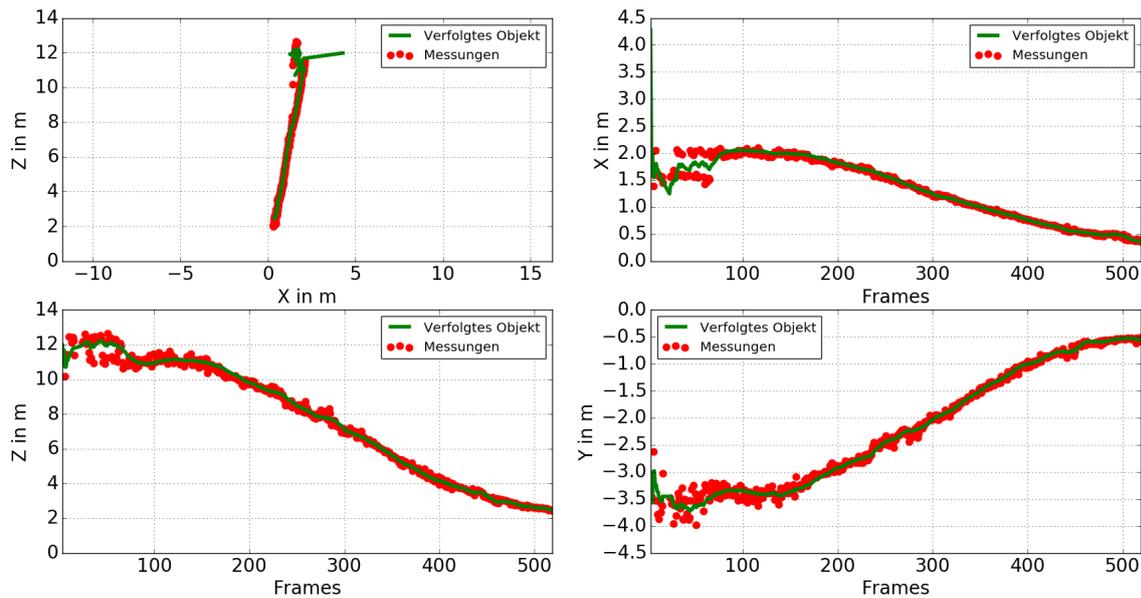


**Abbildung A.18**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.19**

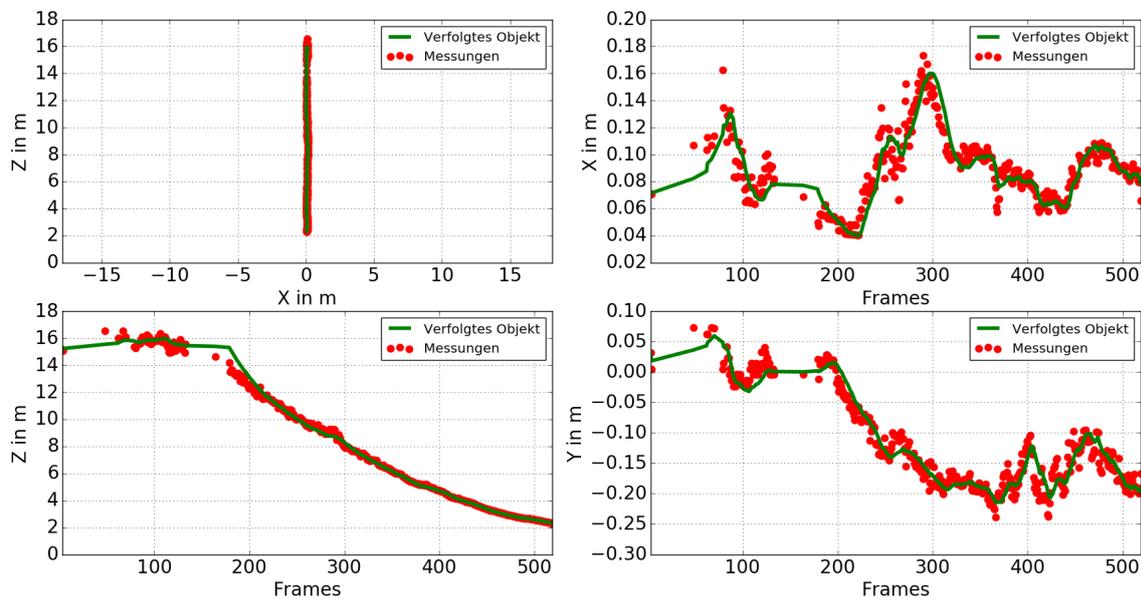
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.20**

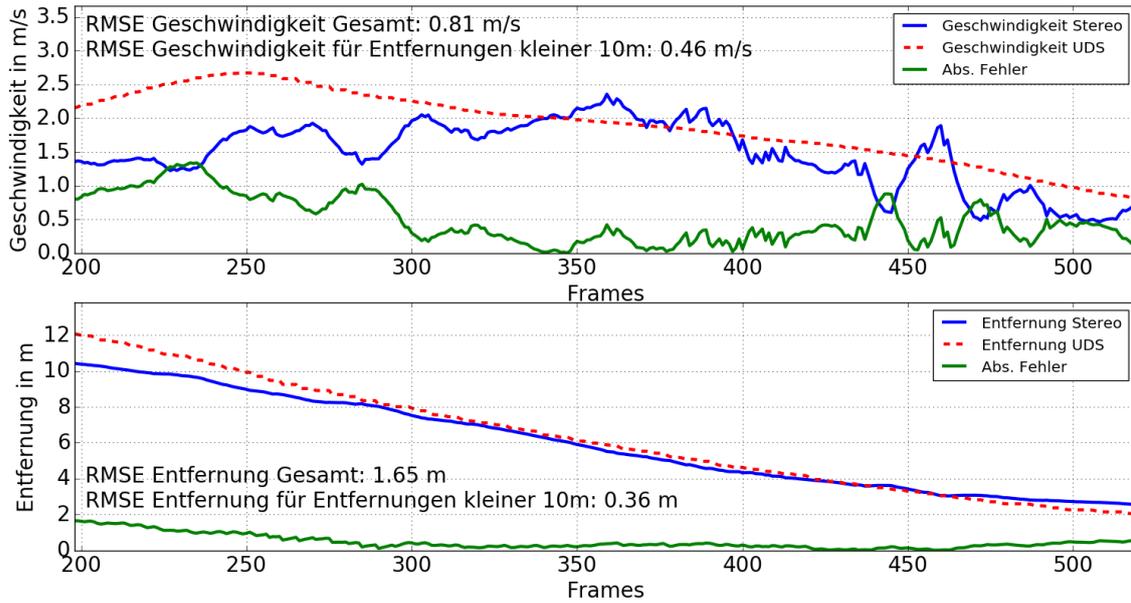
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

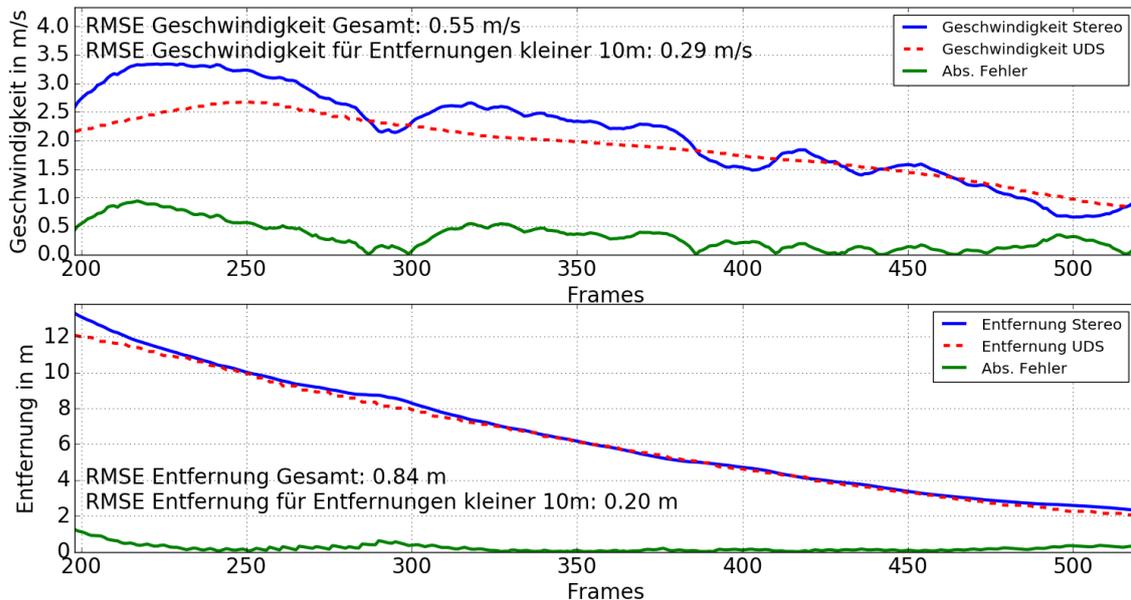
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



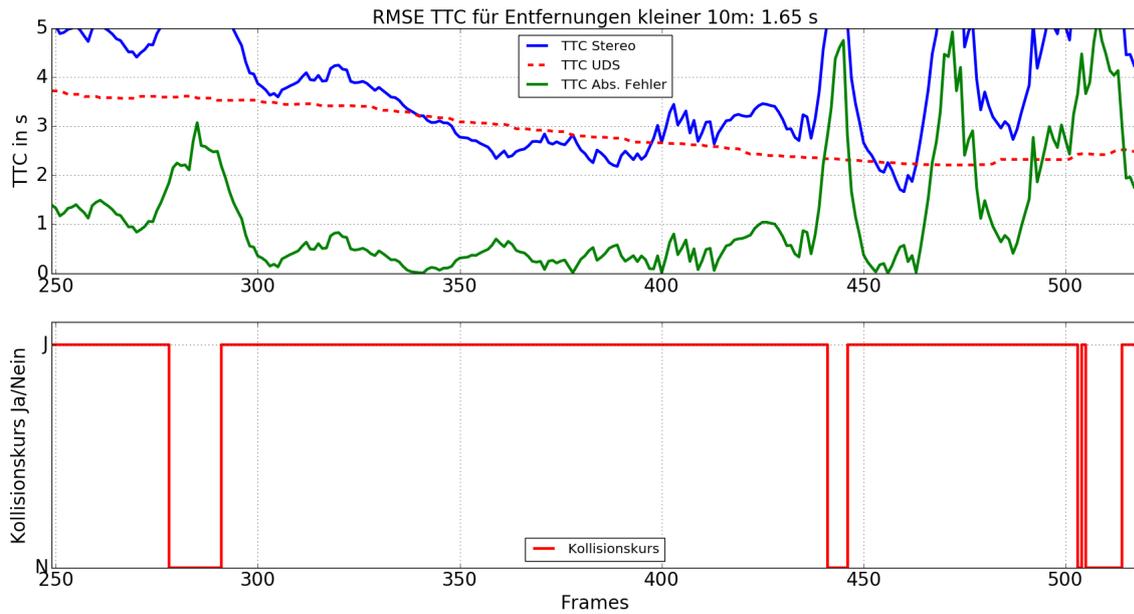
**Abbildung A.21**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.22**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

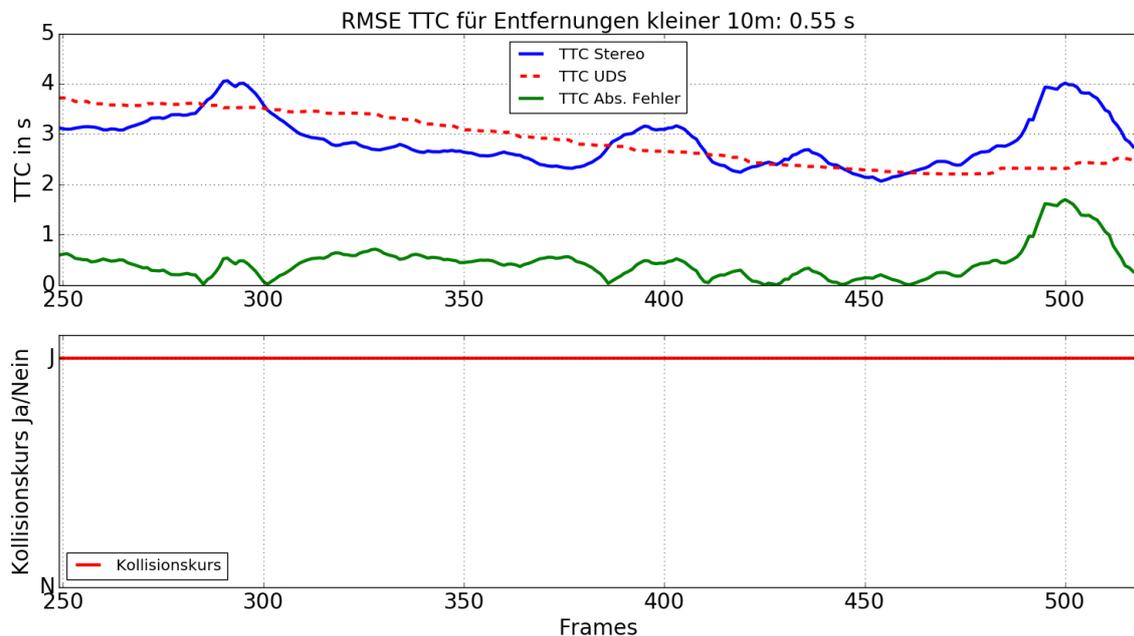


**Abbildung A.23**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

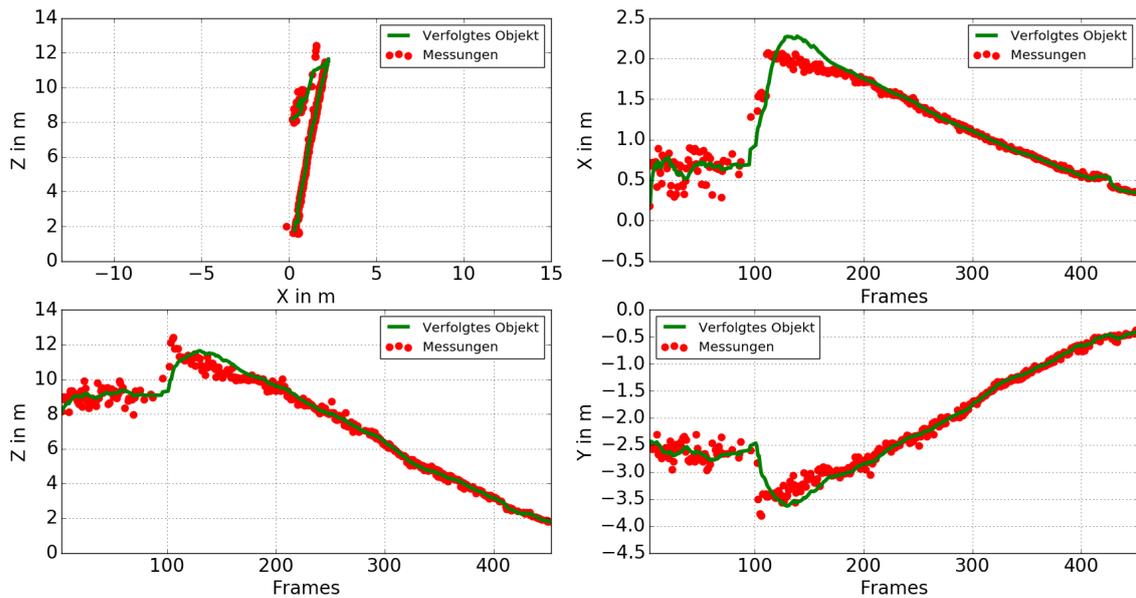


**Abbildung A.24**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.25**

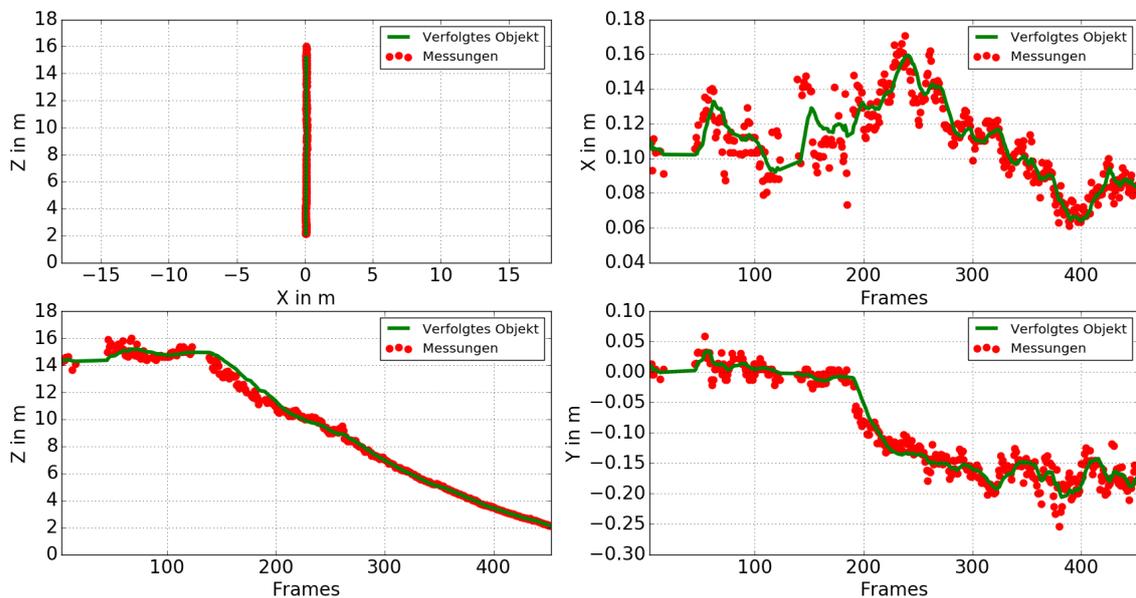
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.26**

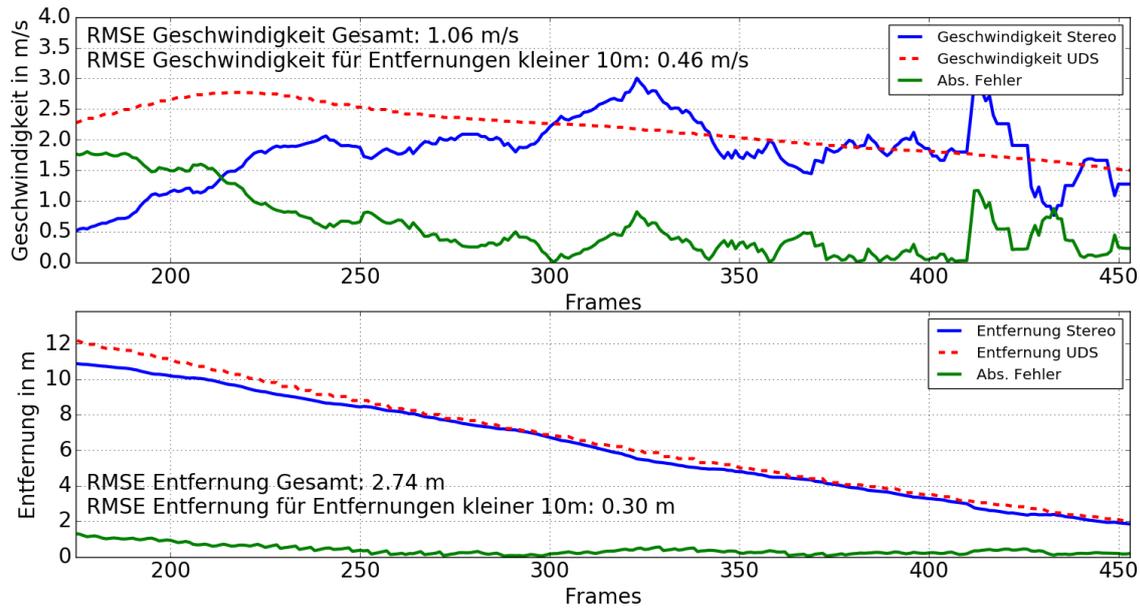
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 1. Es liegt kein *Offset* vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

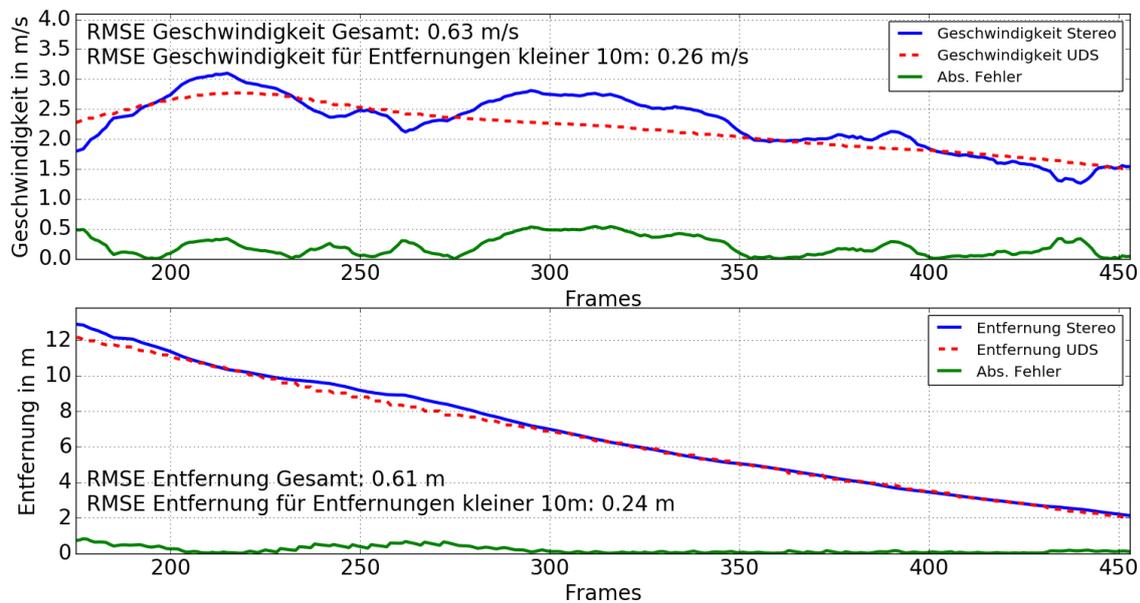
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



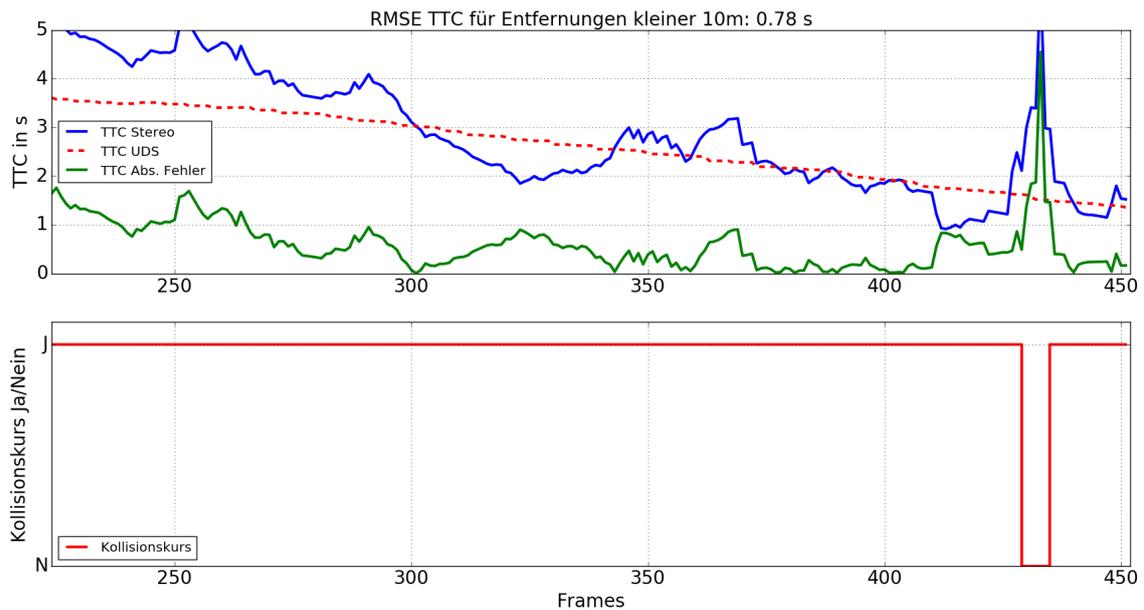
**Abbildung A.27**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.28**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

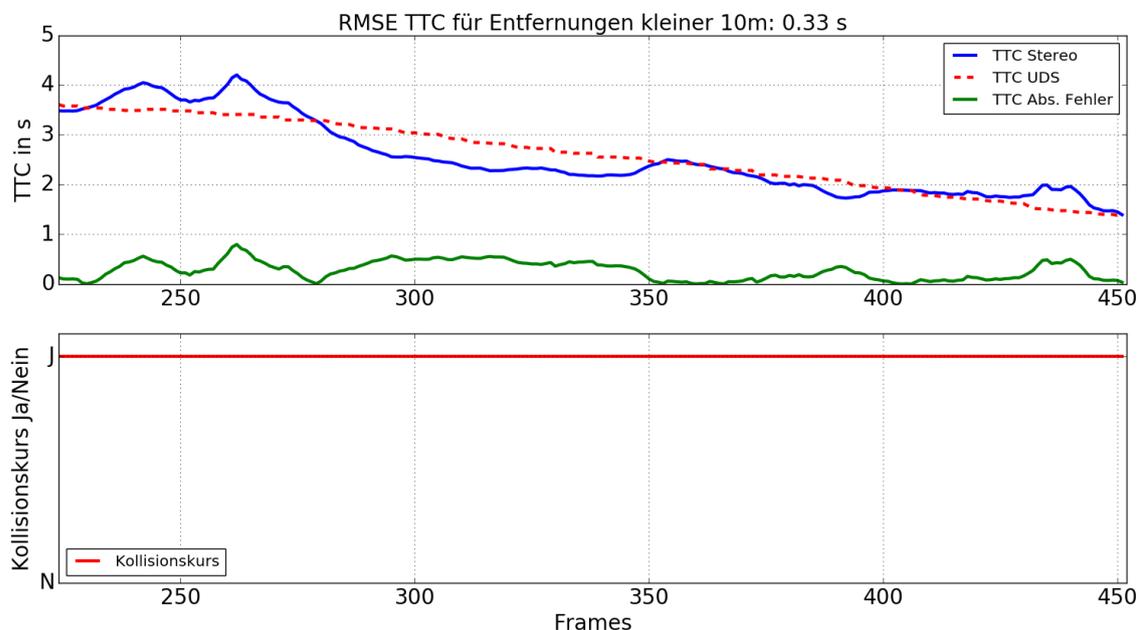


**Abbildung A.29**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

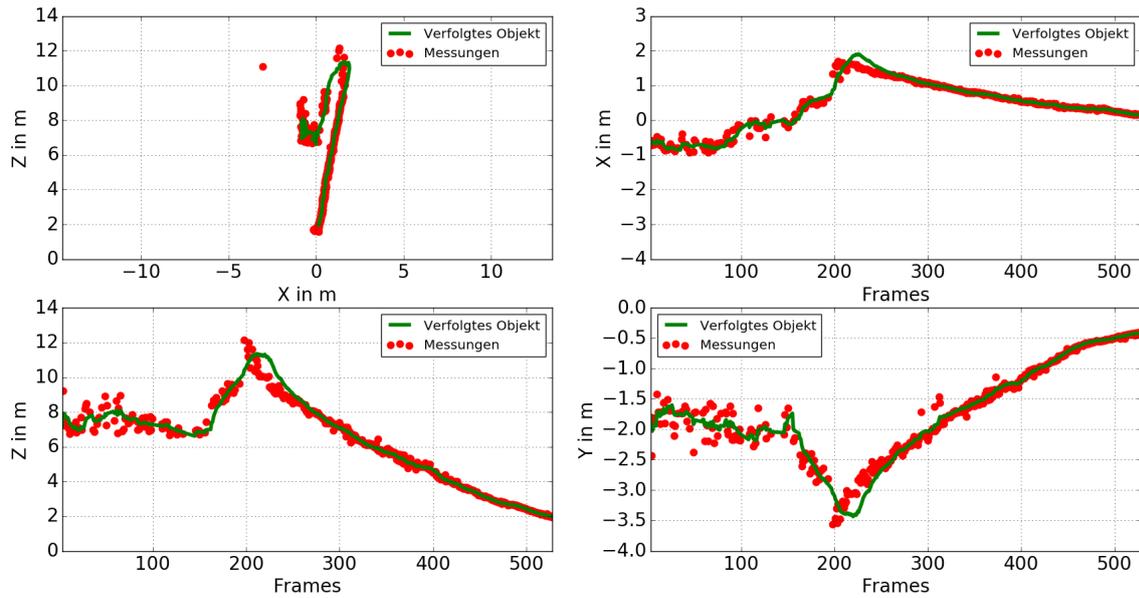


**Abbildung A.30**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



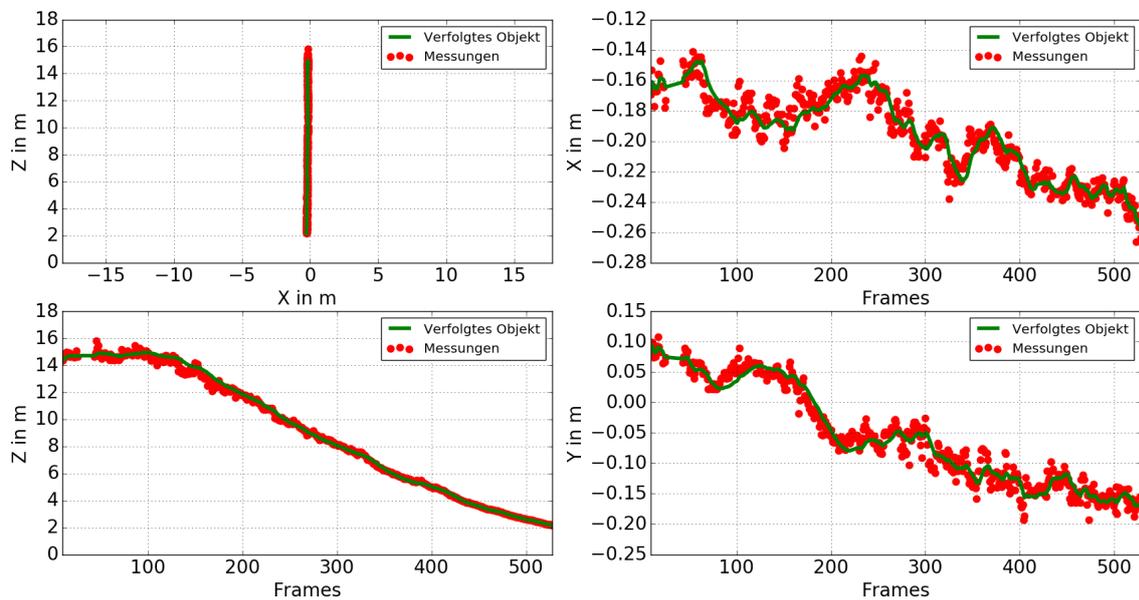
**Abbildung A.31**

Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet. Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.32**

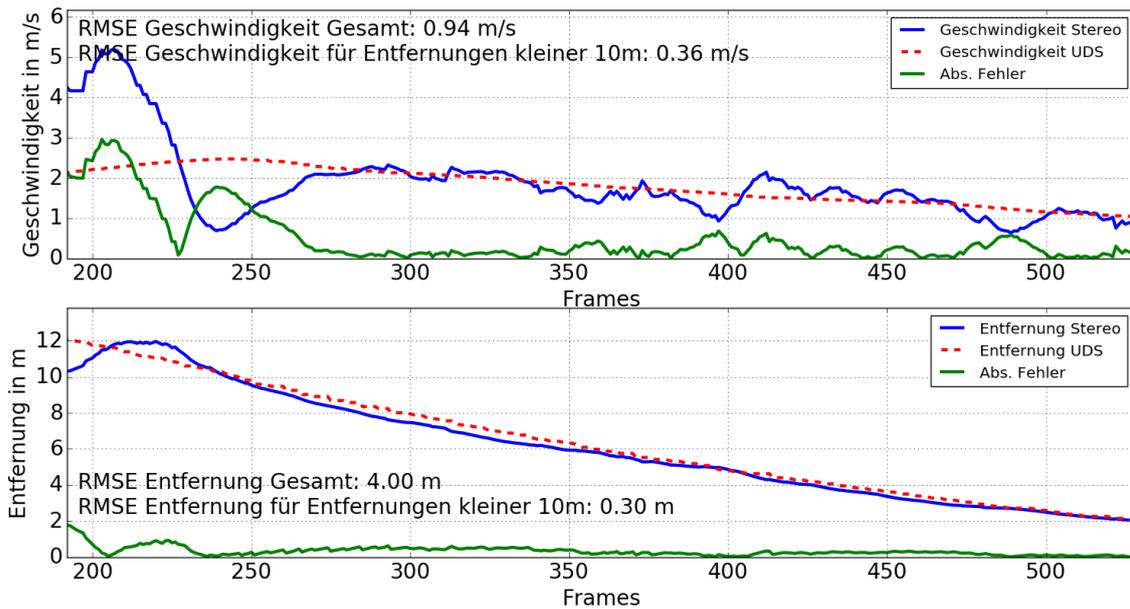
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

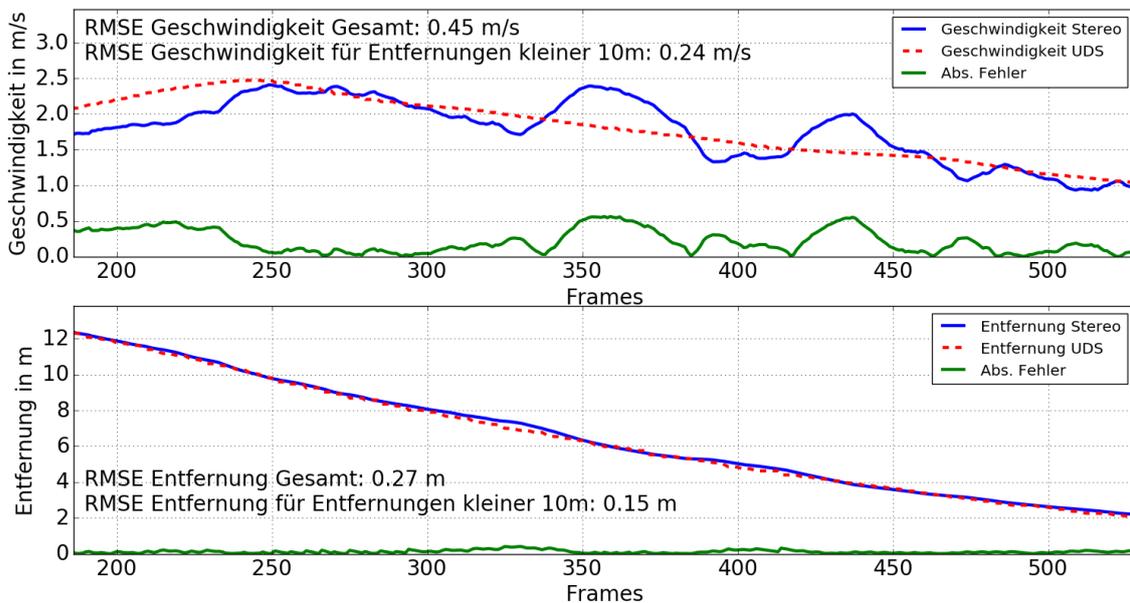
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



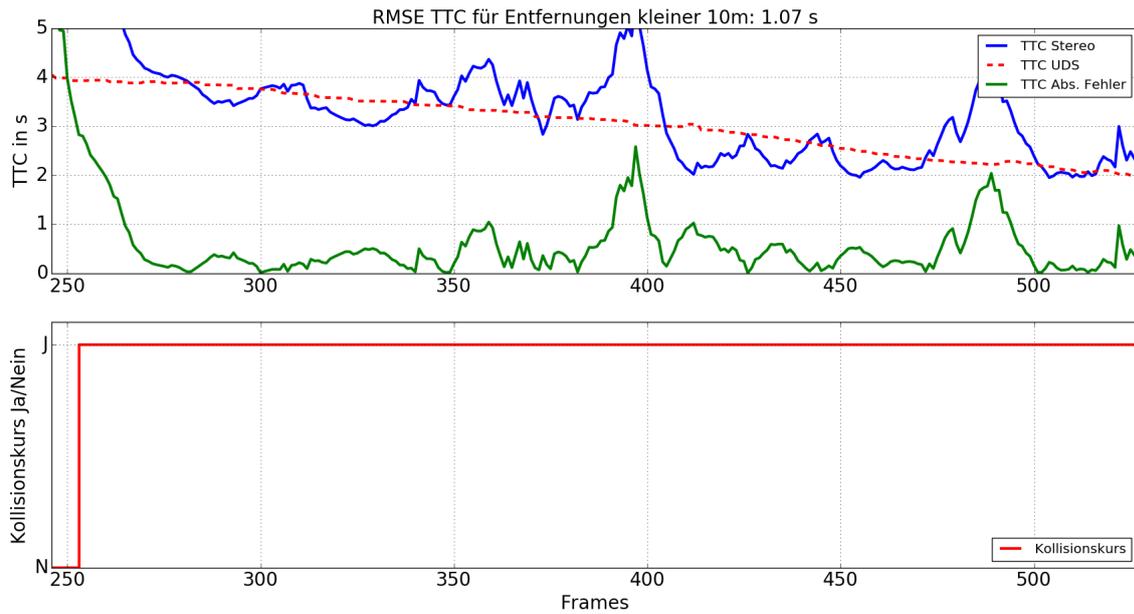
**Abbildung A.33**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.34**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

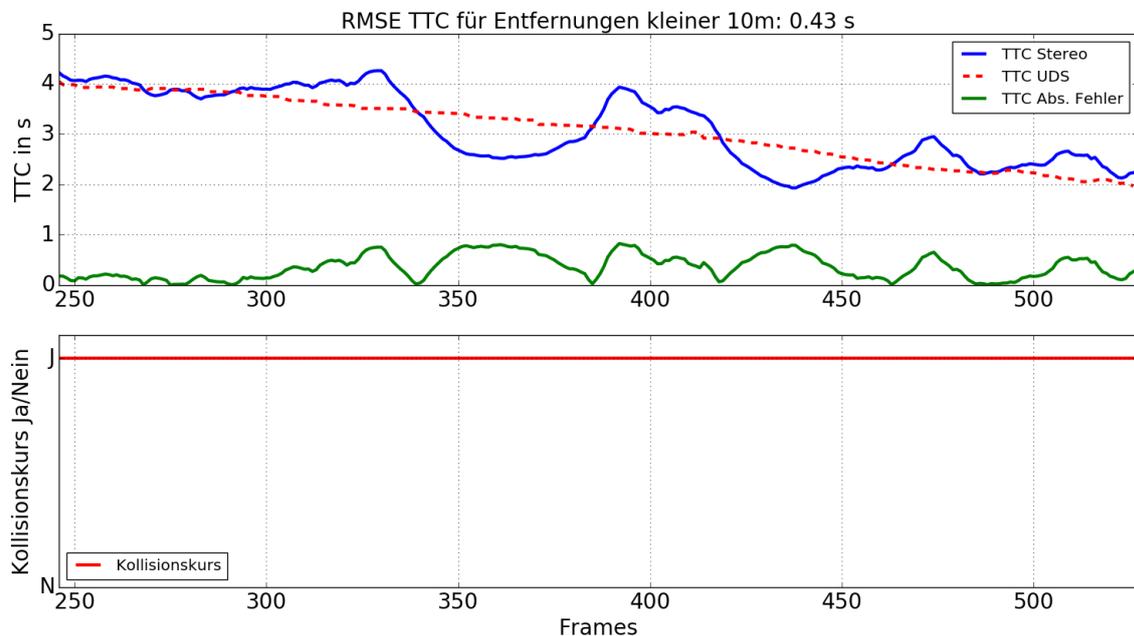


**Abbildung A.35**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

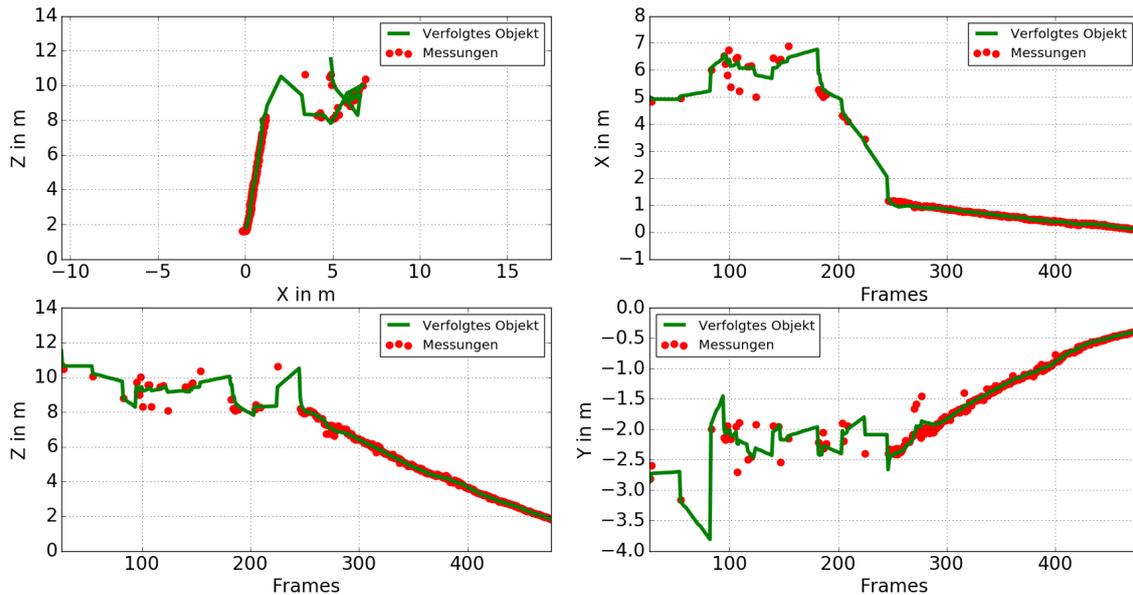


**Abbildung A.36**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

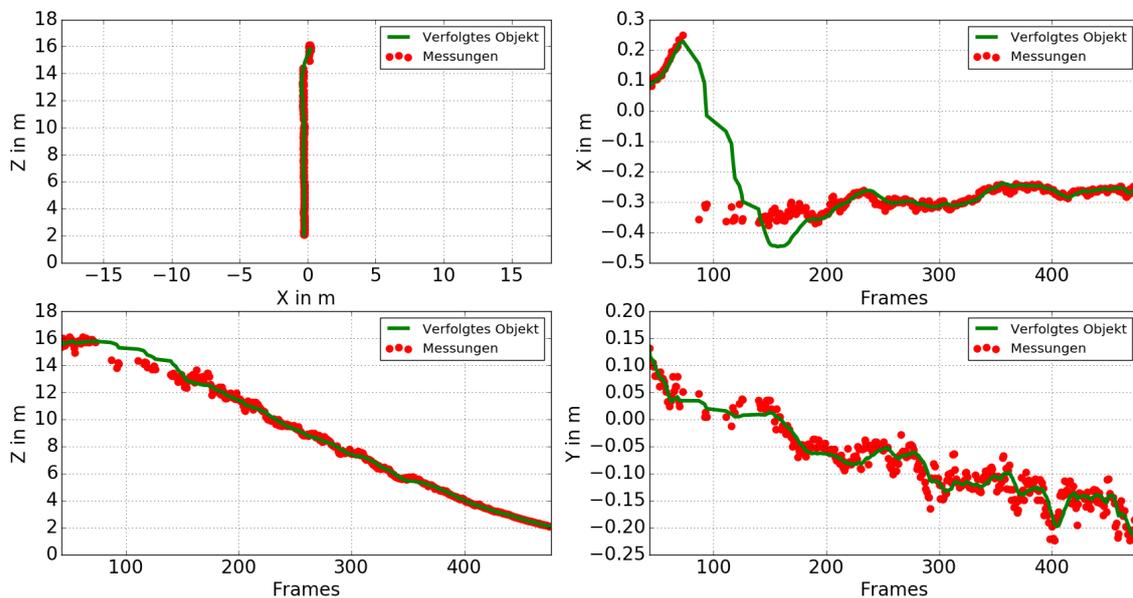
Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



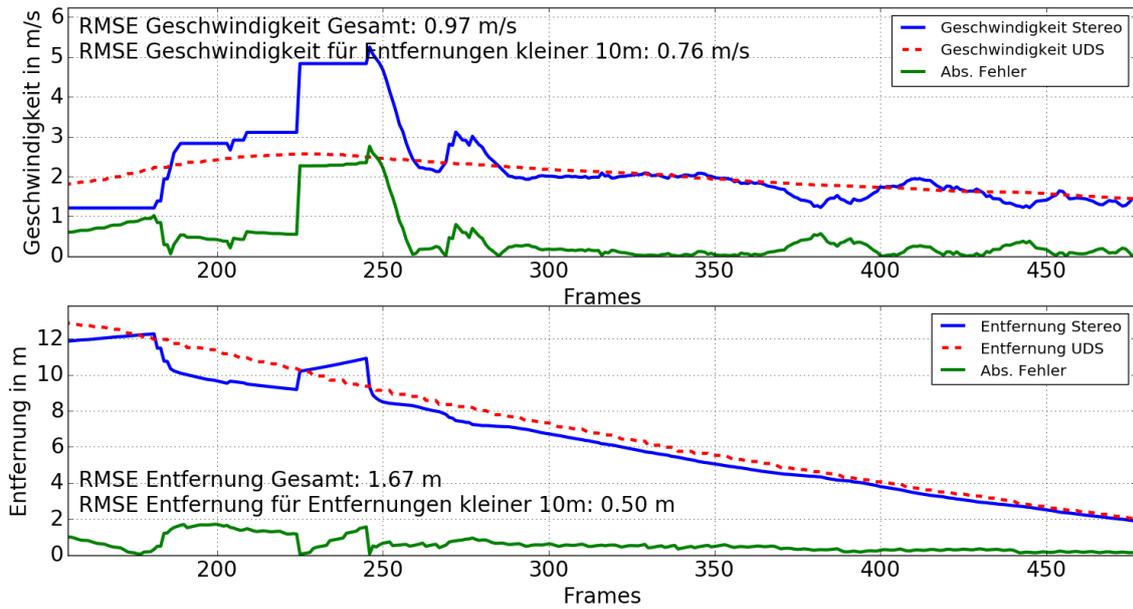
**Abbildung A.37**

Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.  
 Links Oben: Trajektorie in der horizontalen X-Z-Ebene.  
 Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.  
 Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.  
 Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



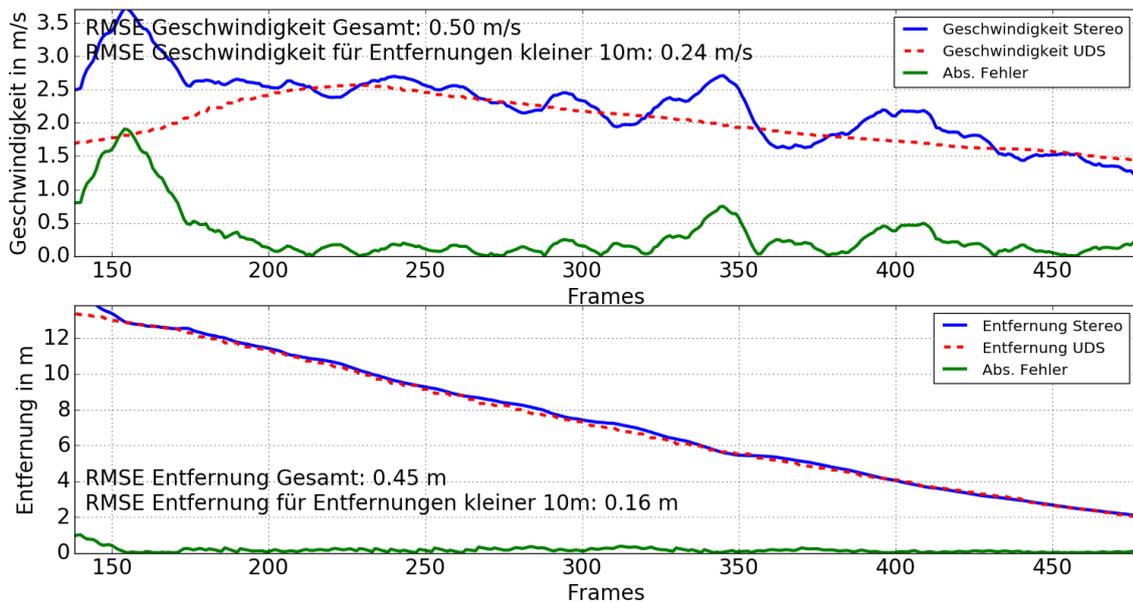
**Abbildung A.38**

Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.  
 Links Oben: Trajektorie in der horizontalen X-Z-Ebene.  
 Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.  
 Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.  
 Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



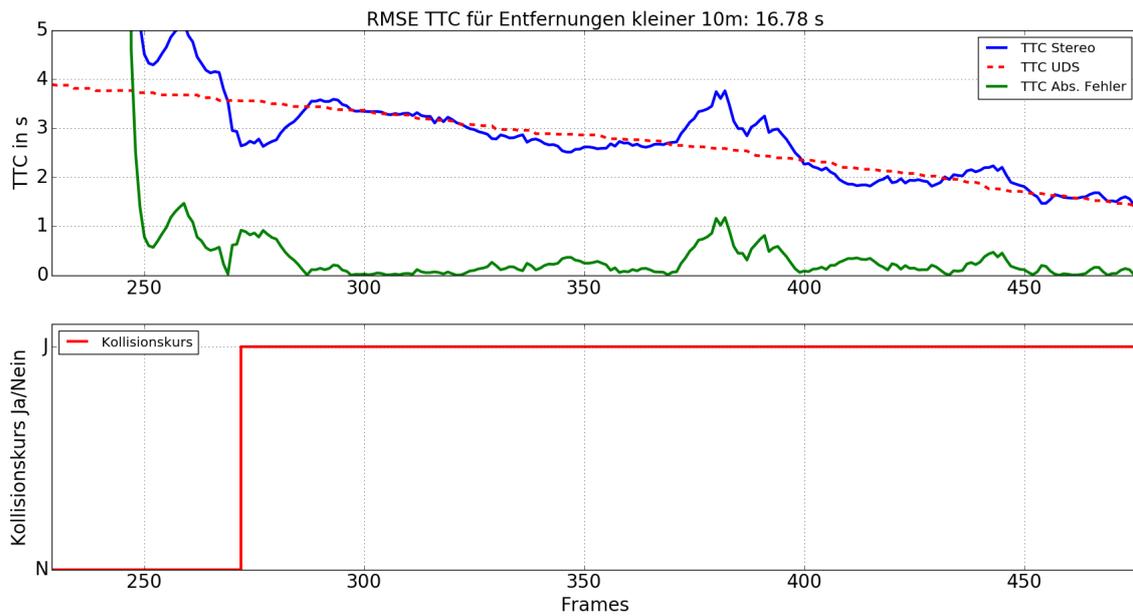
**Abbildung A.39**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.40**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

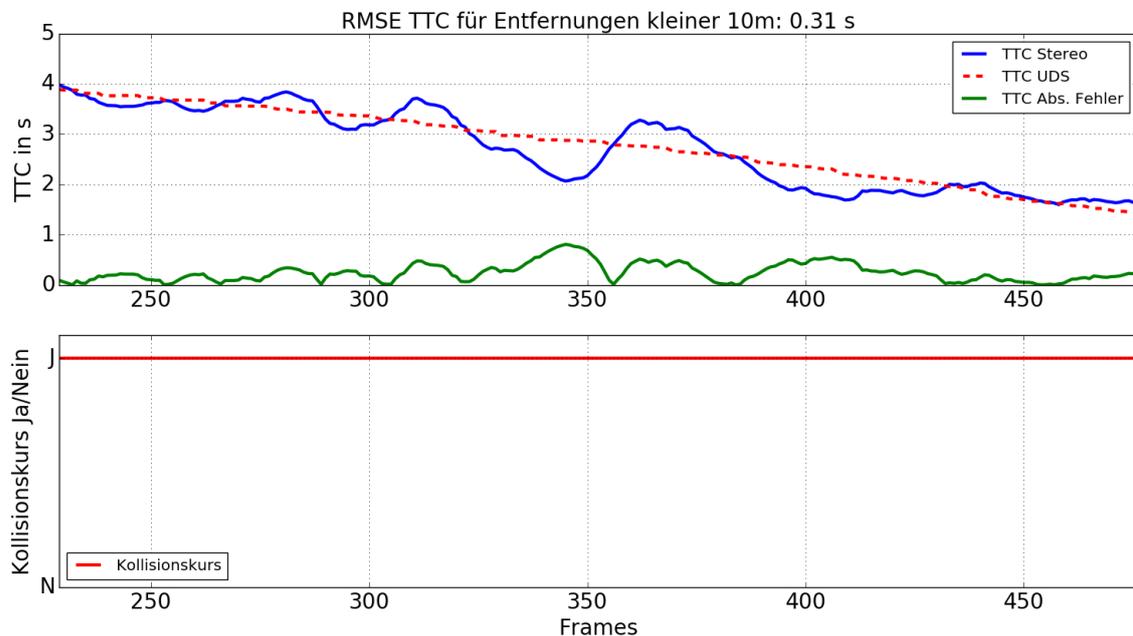


**Abbildung A.41**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

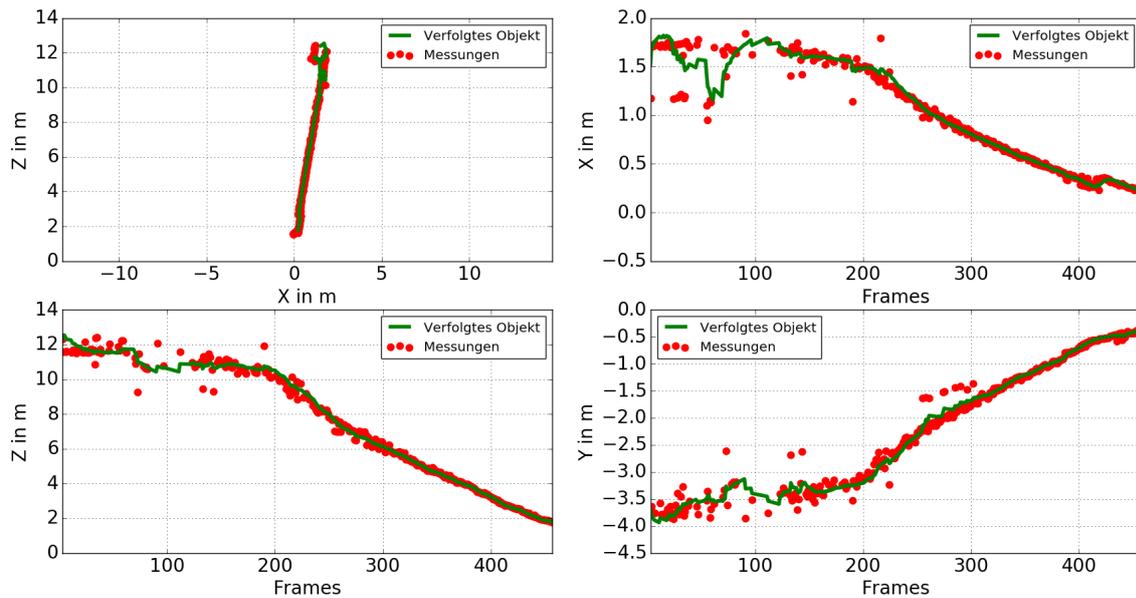


**Abbildung A.42**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



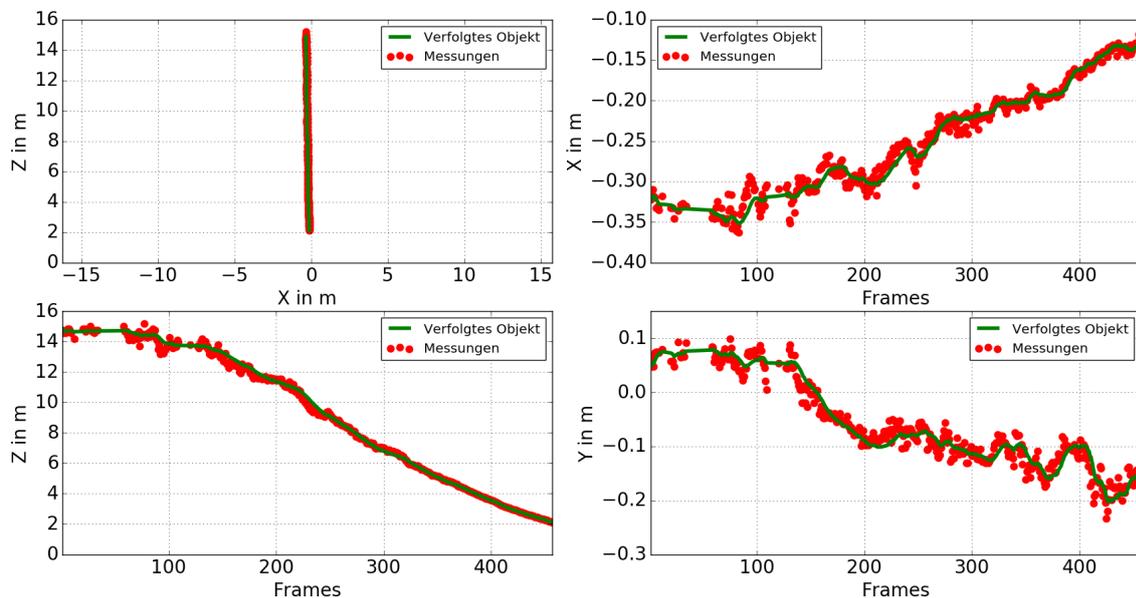
**Abbildung A.43**

Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet. Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



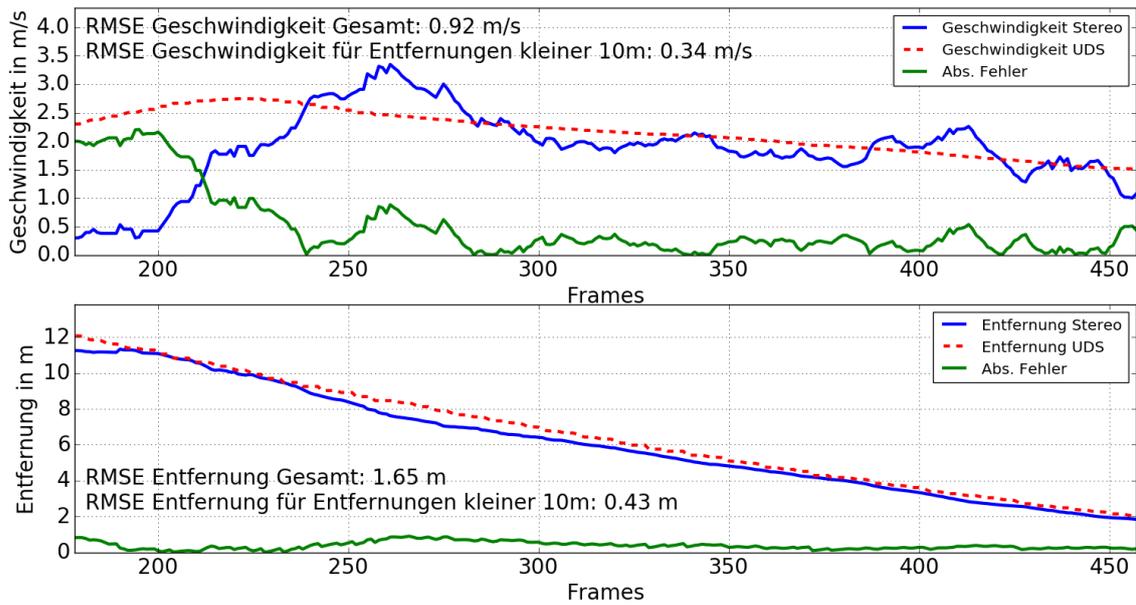
**Abbildung A.44**

Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet. Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

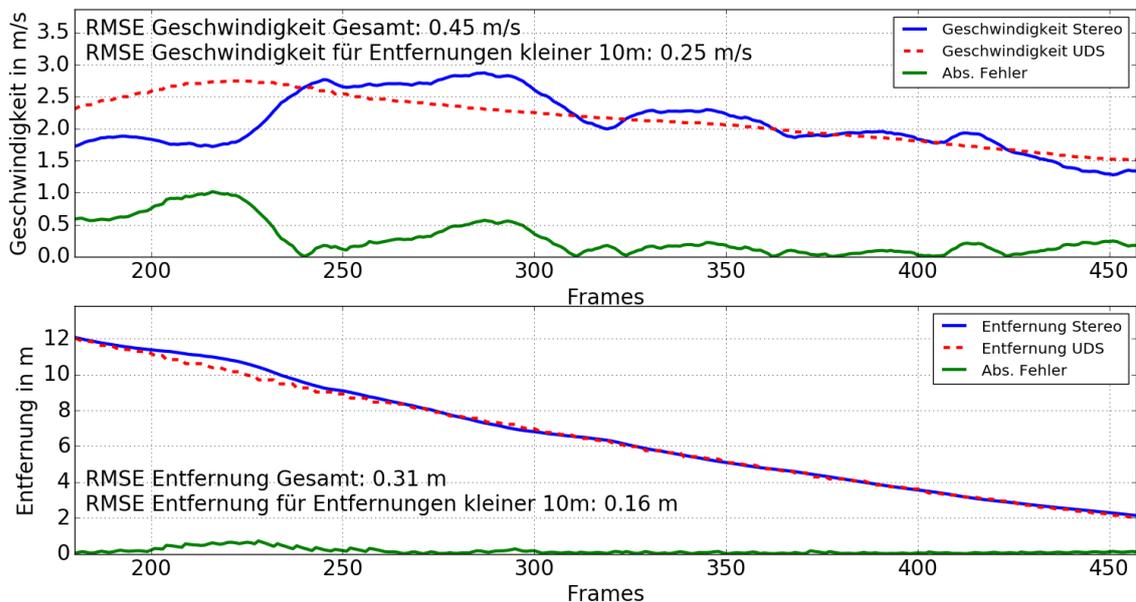
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



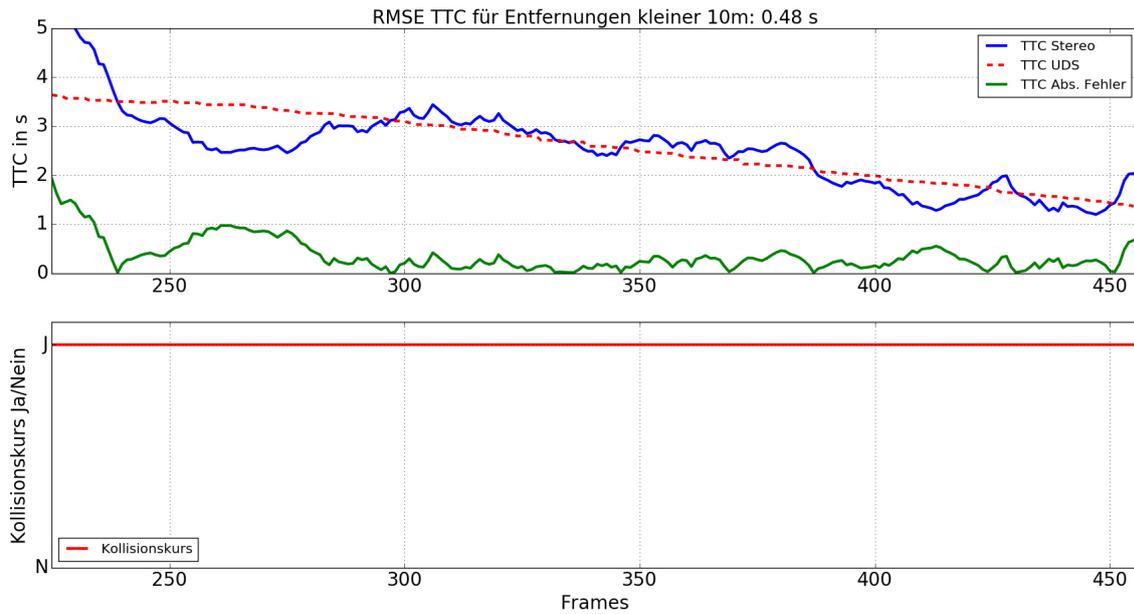
**Abbildung A.45**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.46**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

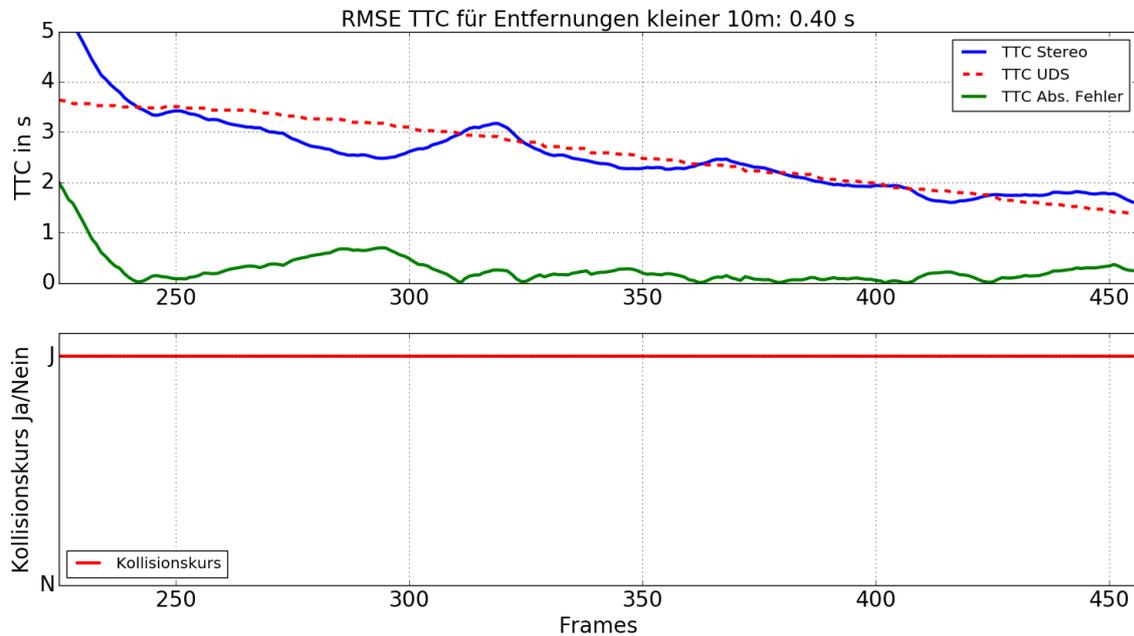


**Abbildung A.47**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

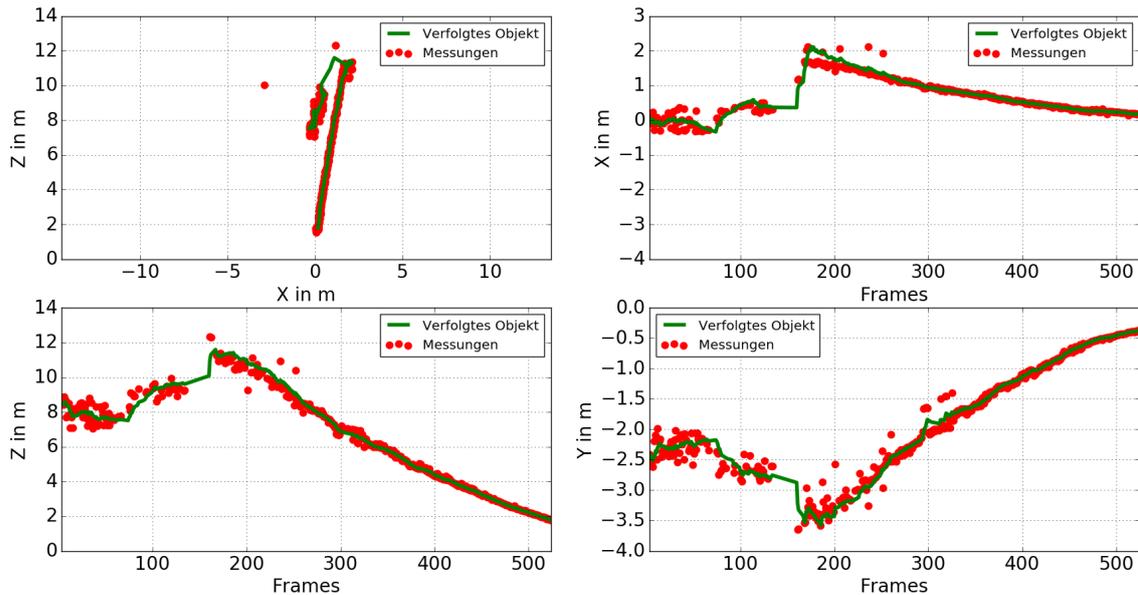


**Abbildung A.48**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

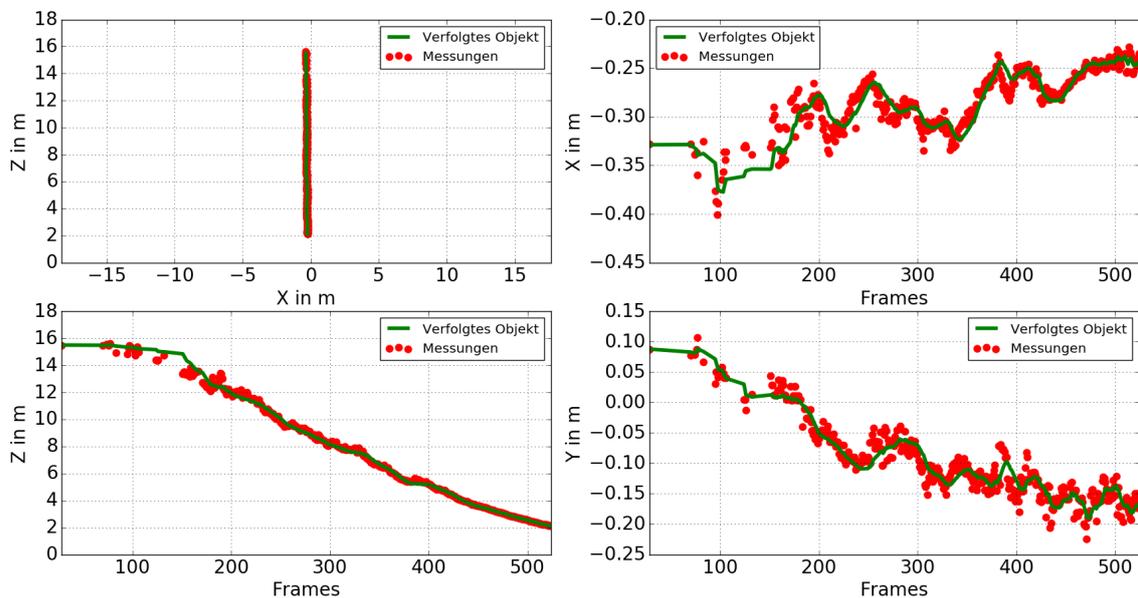
Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



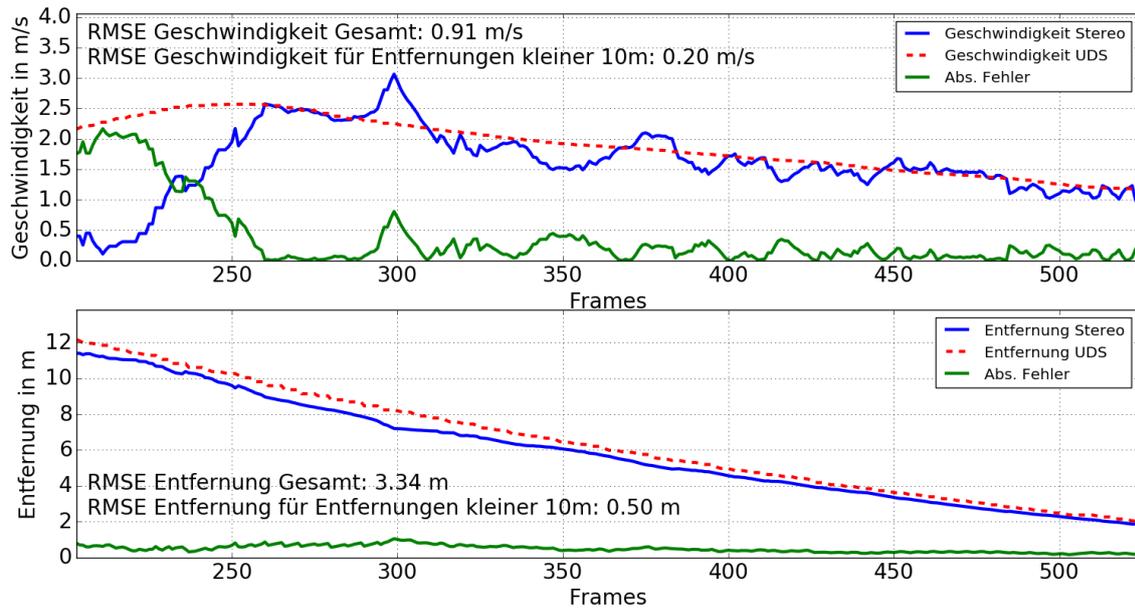
**Abbildung A.49**

Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.  
 Links Oben: Trajektorie in der horizontalen X-Z-Ebene.  
 Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.  
 Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.  
 Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



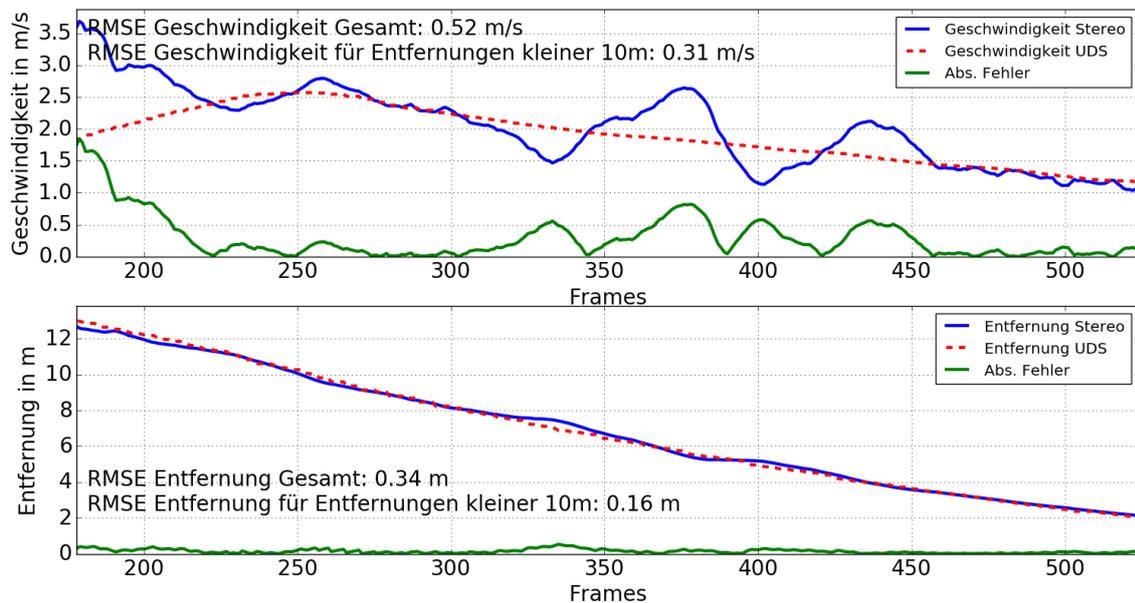
**Abbildung A.50**

Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.  
 Links Oben: Trajektorie in der horizontalen X-Z-Ebene.  
 Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.  
 Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.  
 Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



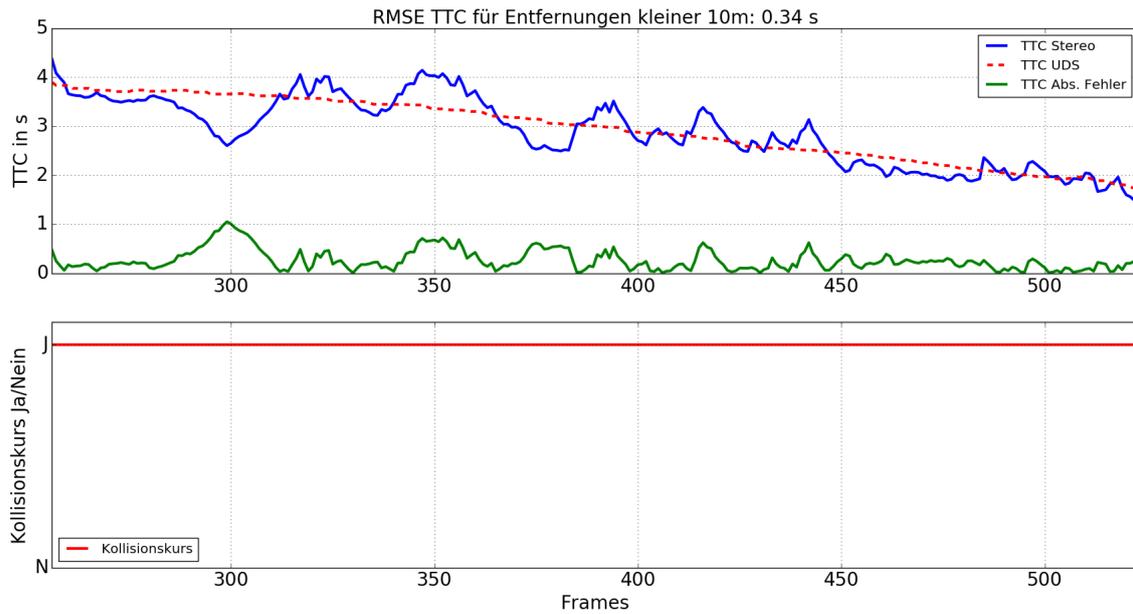
**Abbildung A.51**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.52**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

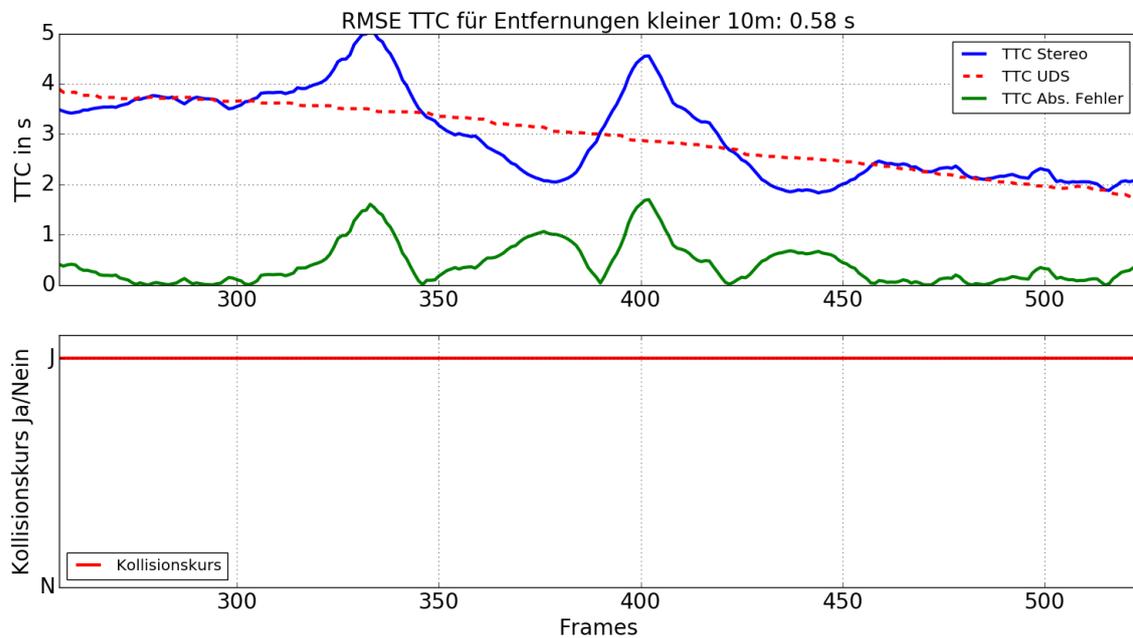


**Abbildung A.53**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

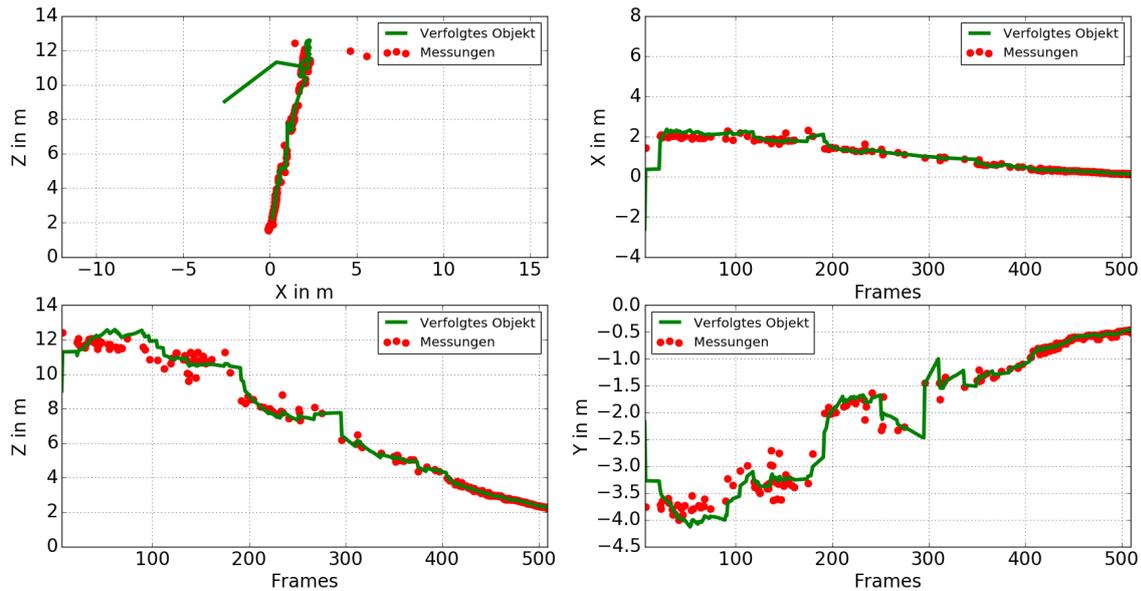


**Abbildung A.54**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



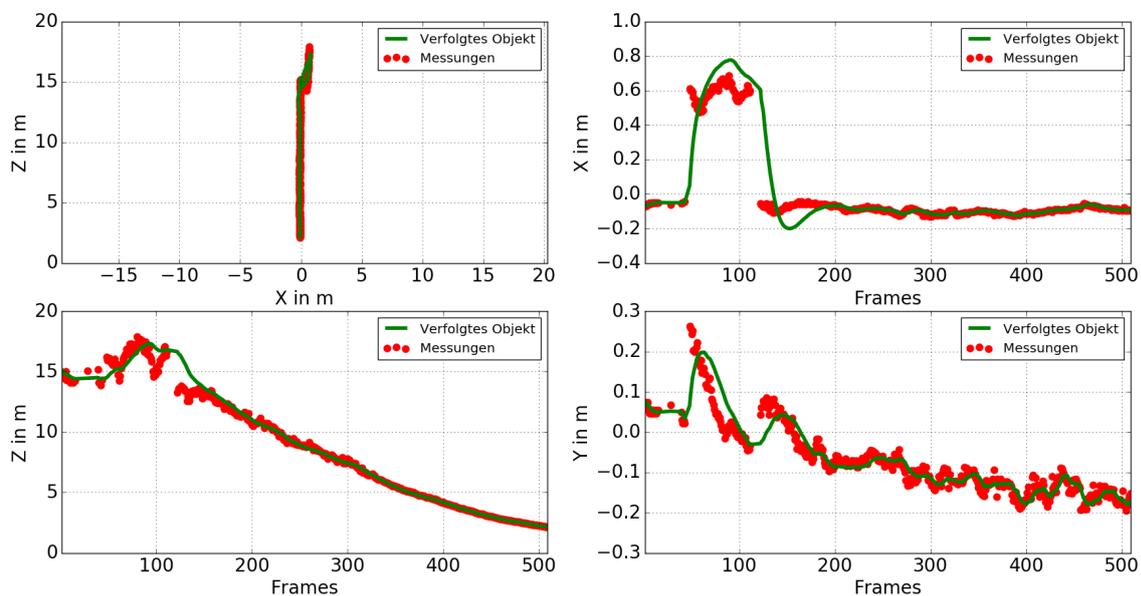
**Abbildung A.55**

Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet. Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.56**

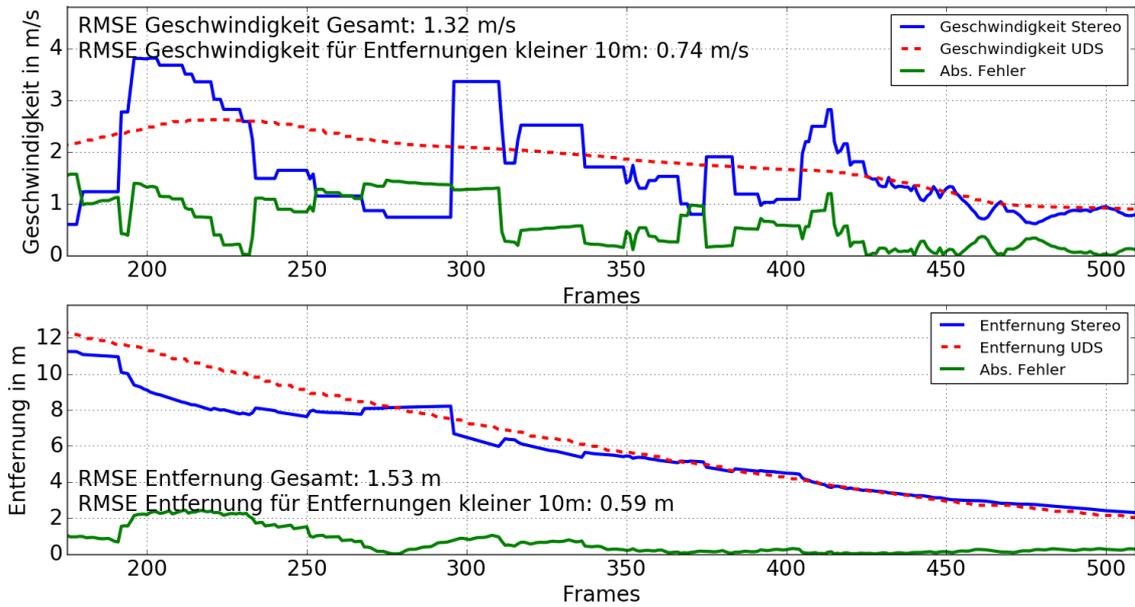
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,2m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

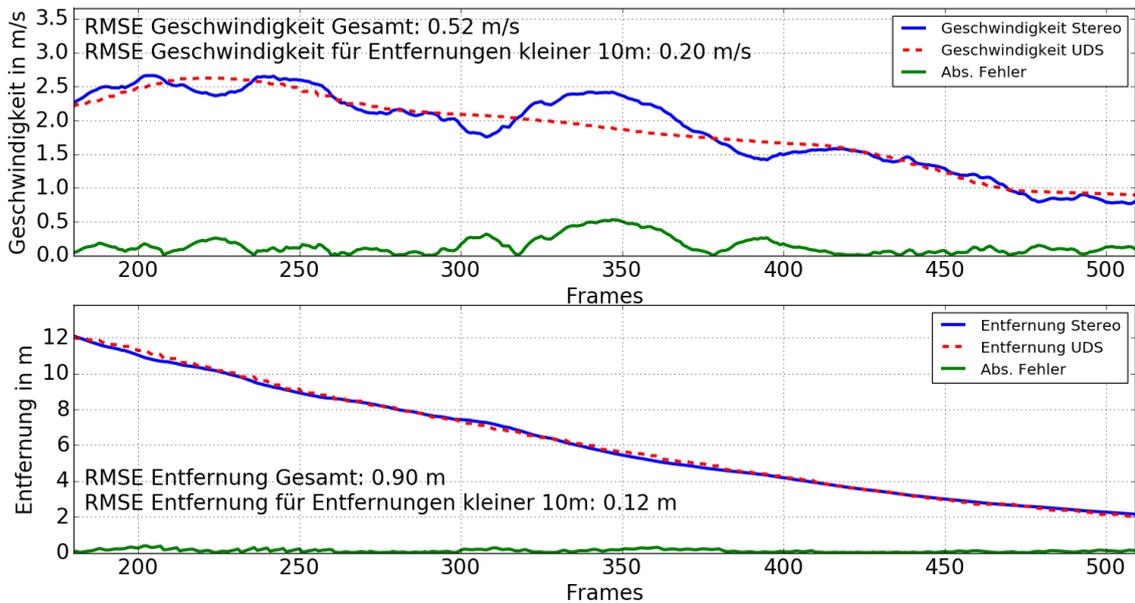
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



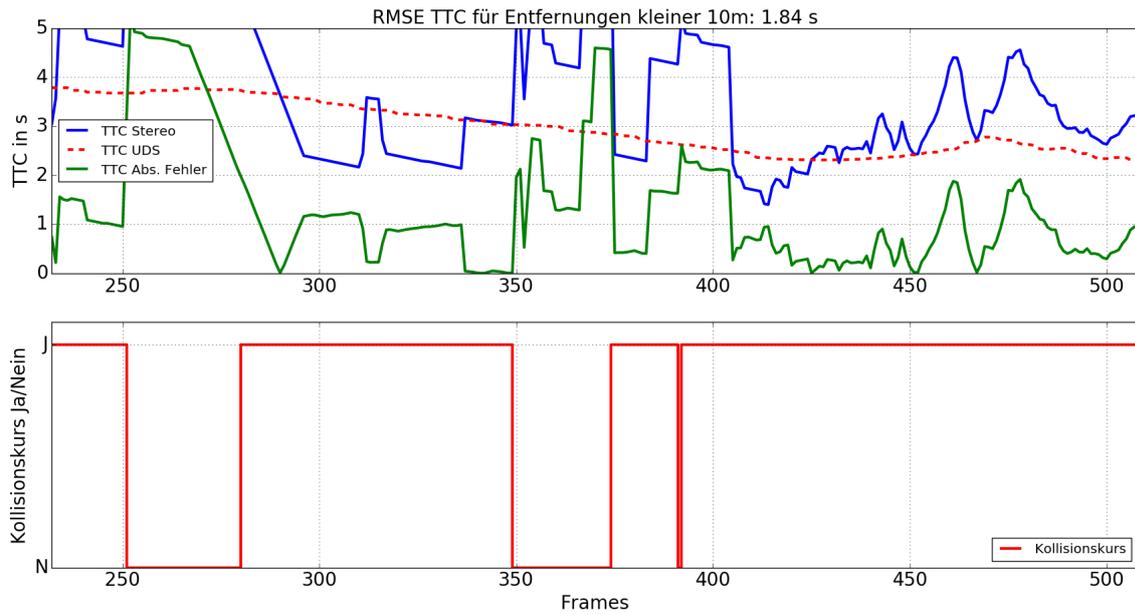
**Abbildung A.57**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.58**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

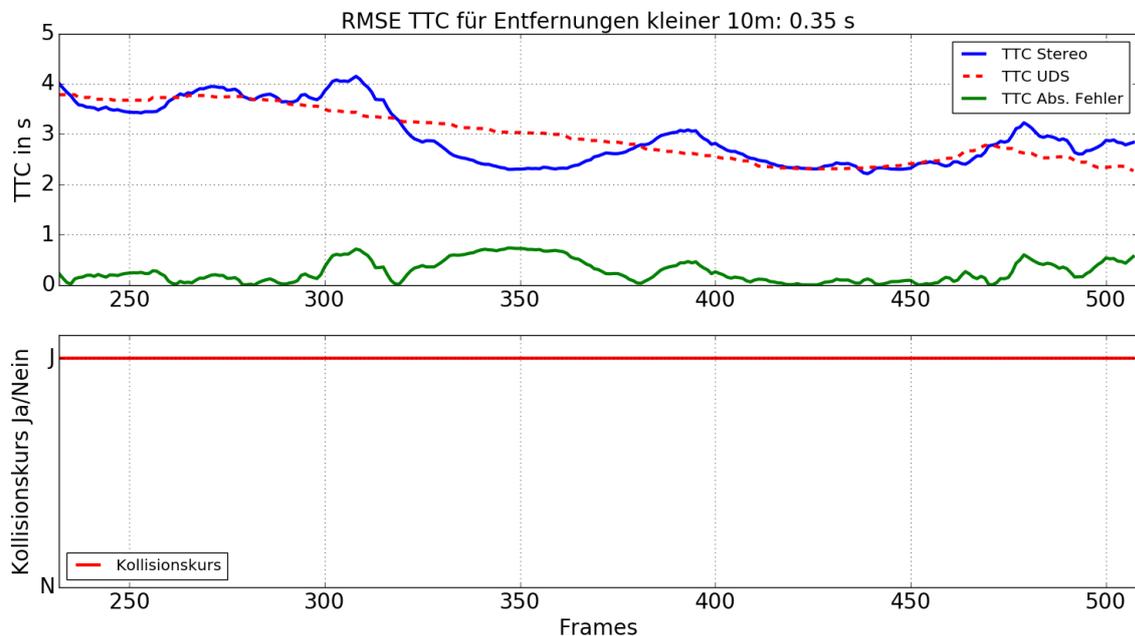


**Abbildung A.59**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

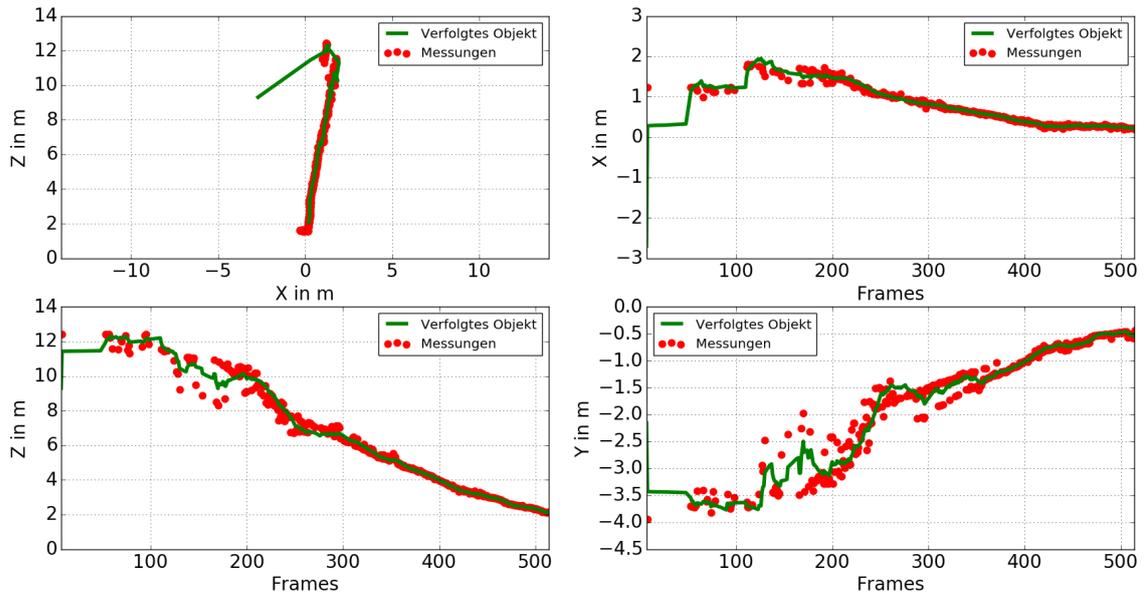


**Abbildung A.60**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.61**

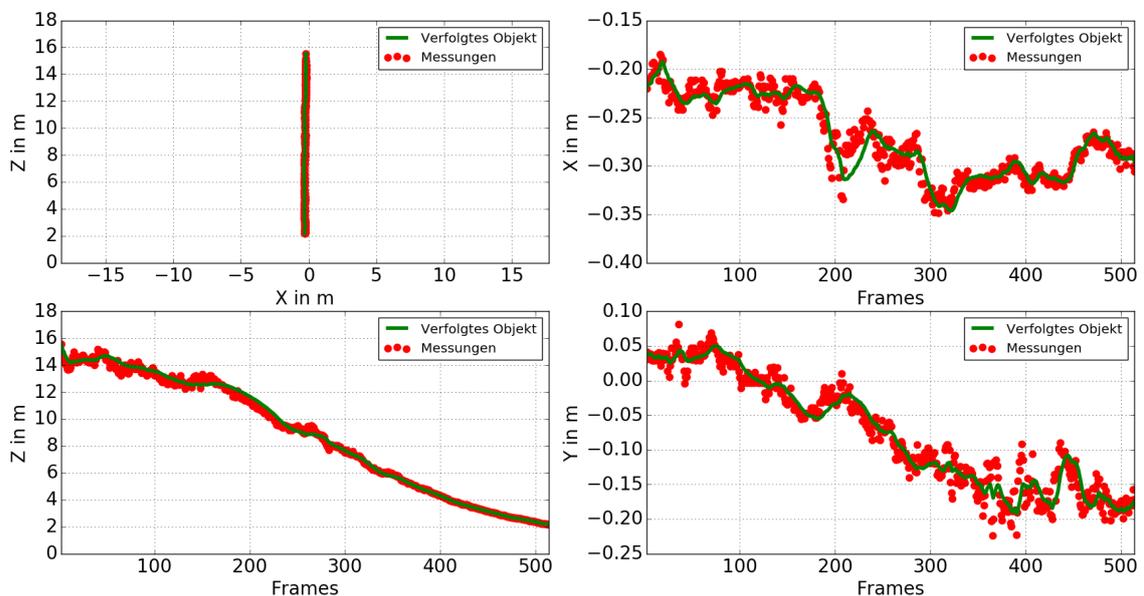
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.62**

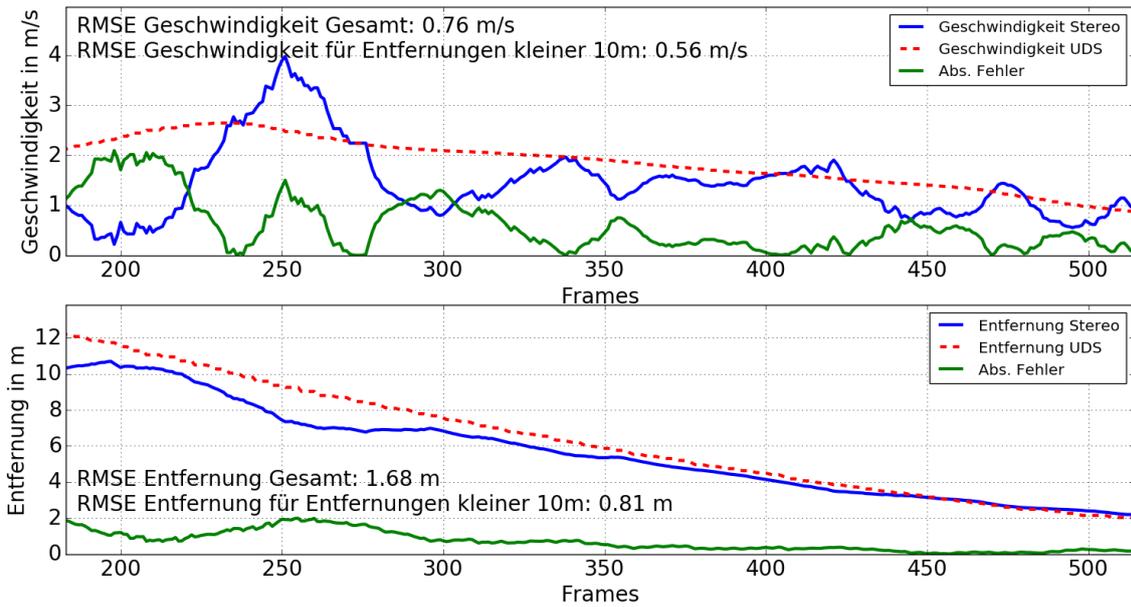
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

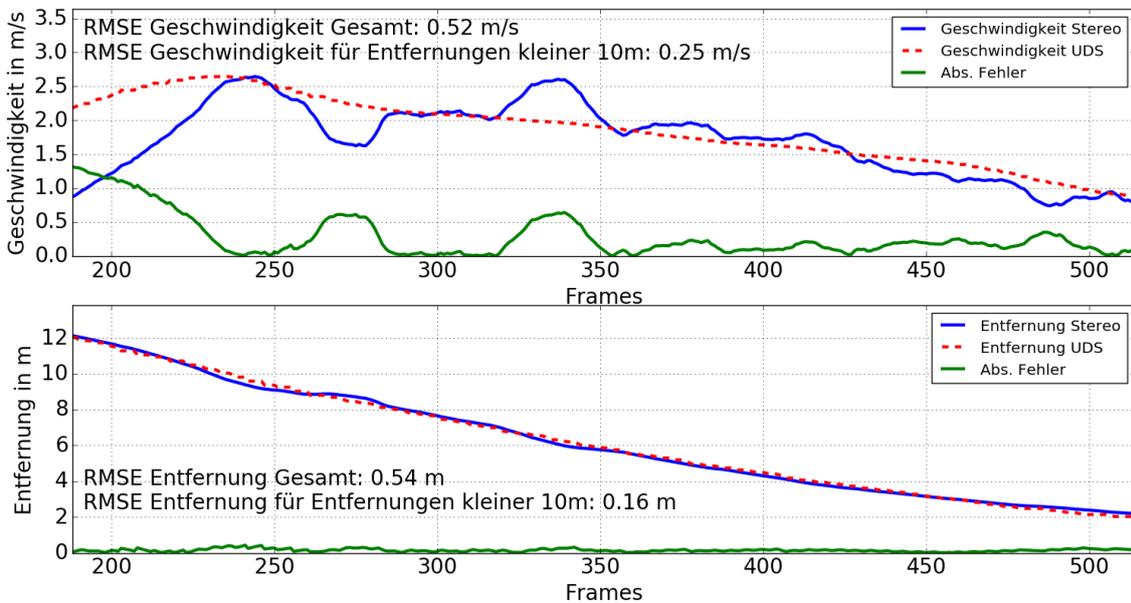
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



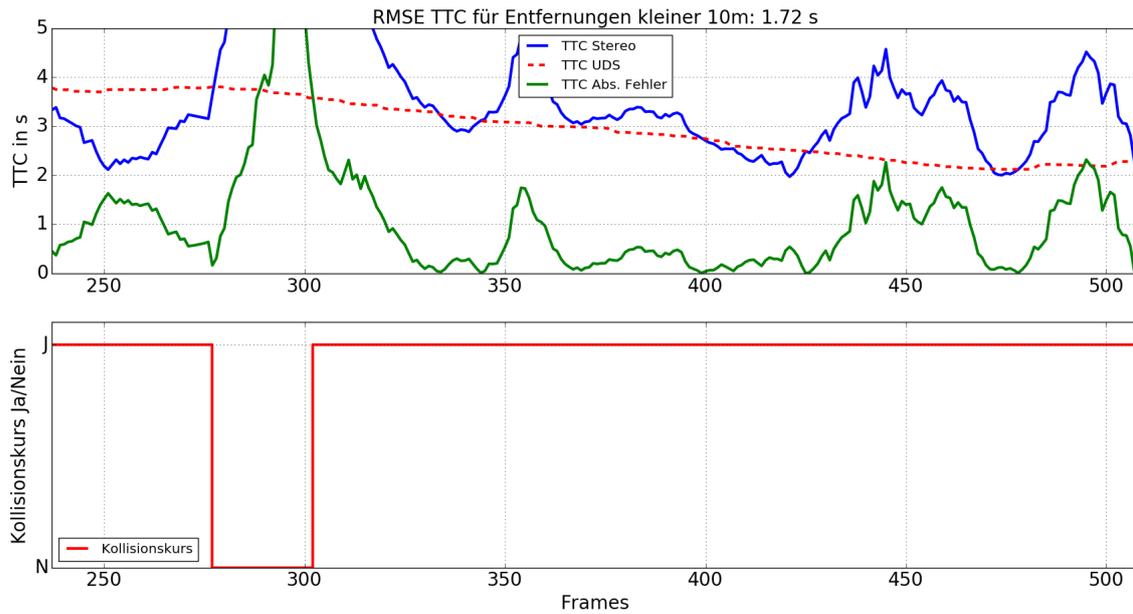
**Abbildung A.63**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.64**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

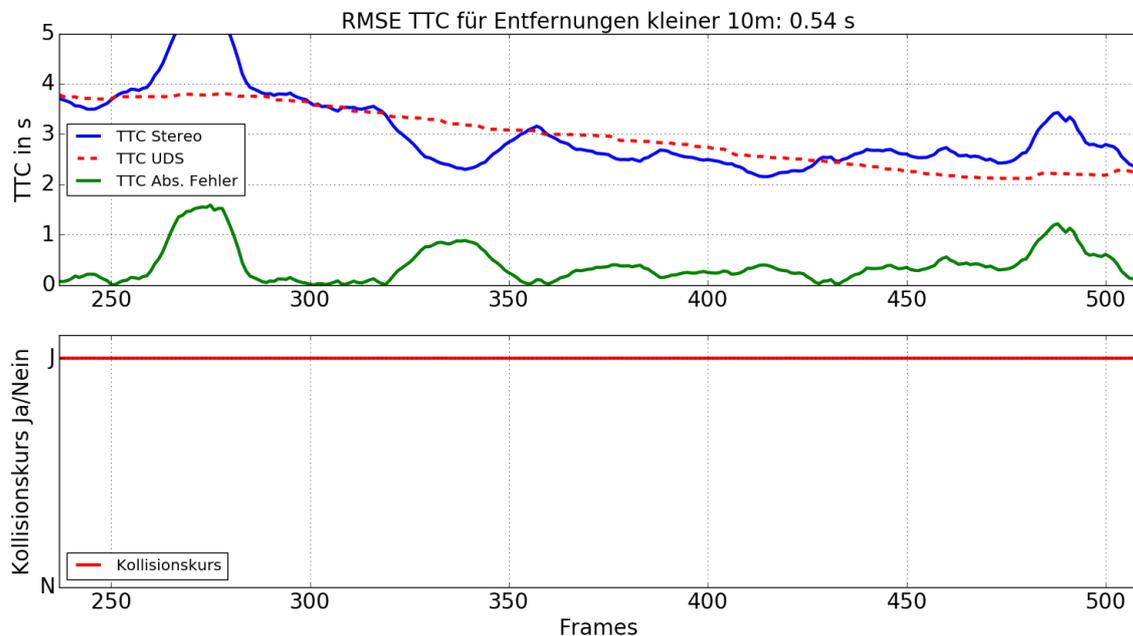


**Abbildung A.65**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

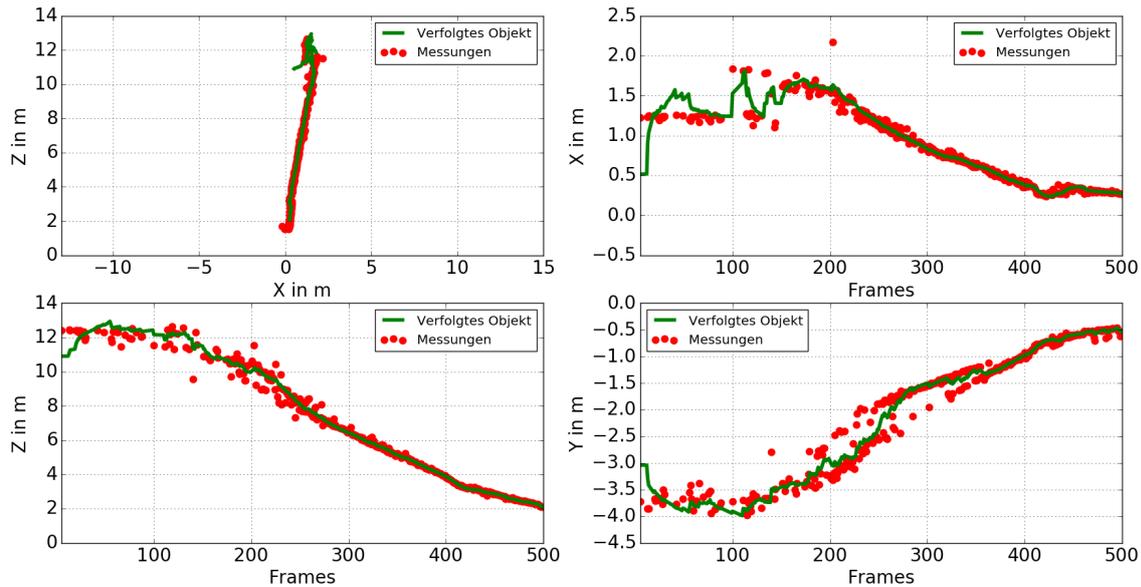


**Abbildung A.66**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

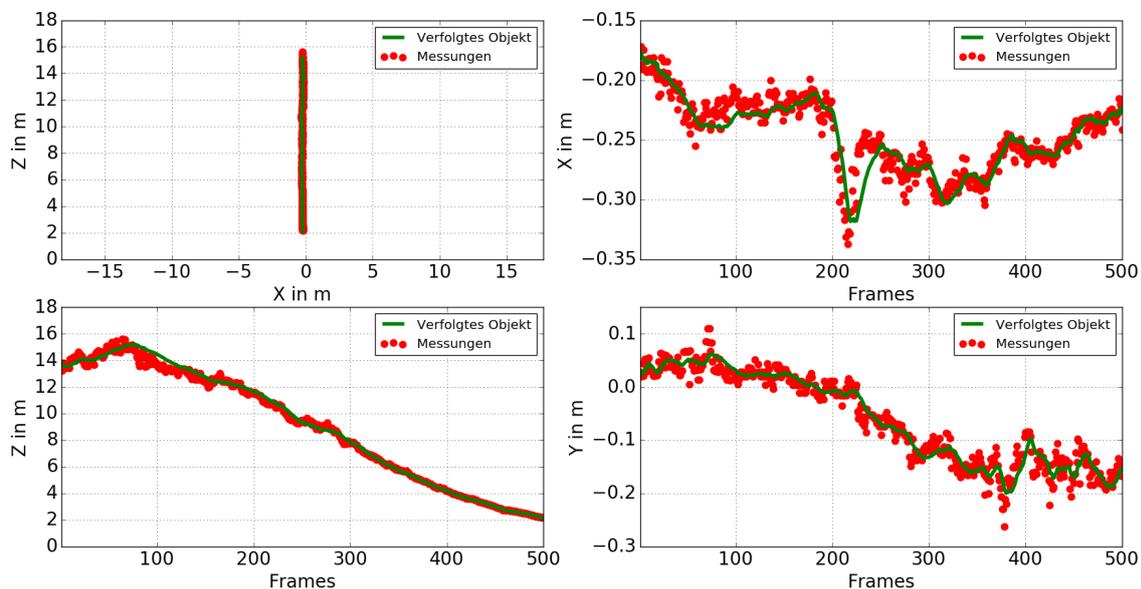
Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



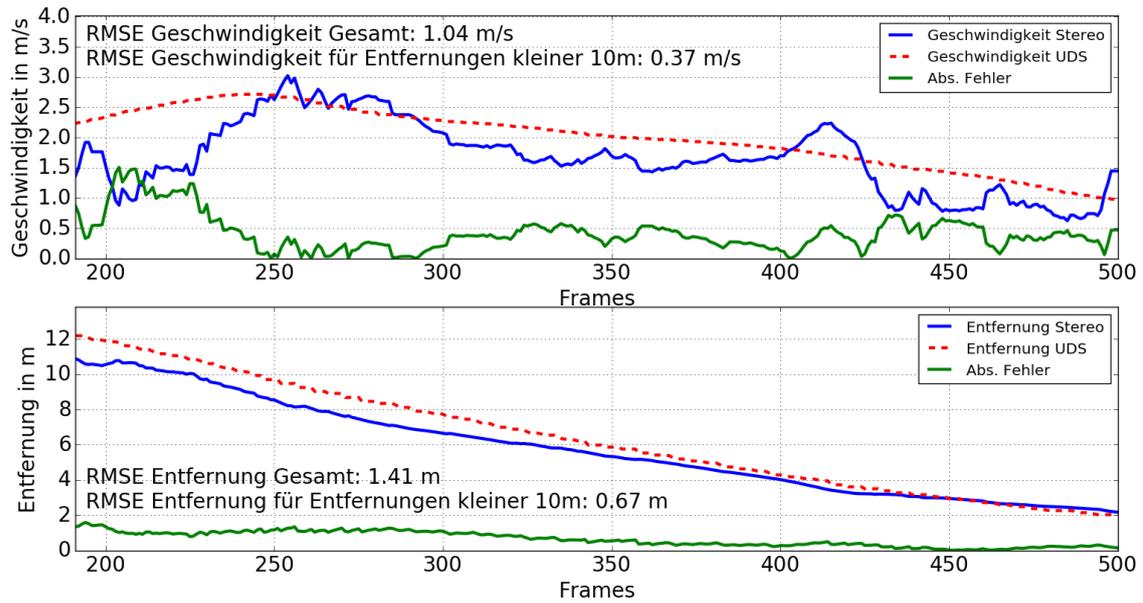
**Abbildung A.67**

Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.  
 Links Oben: Trajektorie in der horizontalen X-Z-Ebene.  
 Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.  
 Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.  
 Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



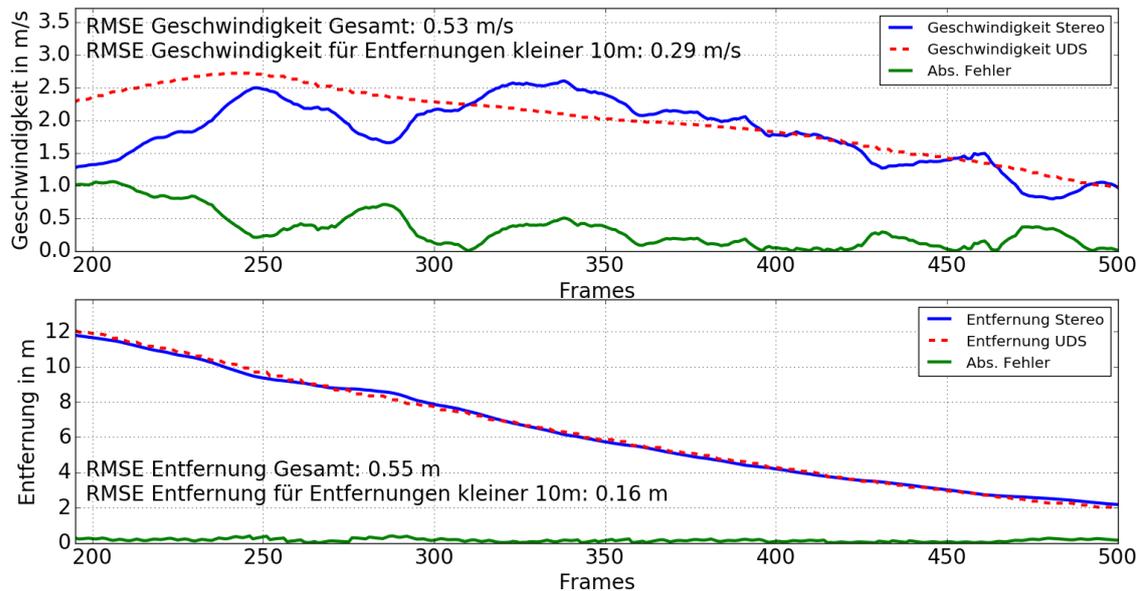
**Abbildung A.68**

Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.  
 Links Oben: Trajektorie in der horizontalen X-Z-Ebene.  
 Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.  
 Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.  
 Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



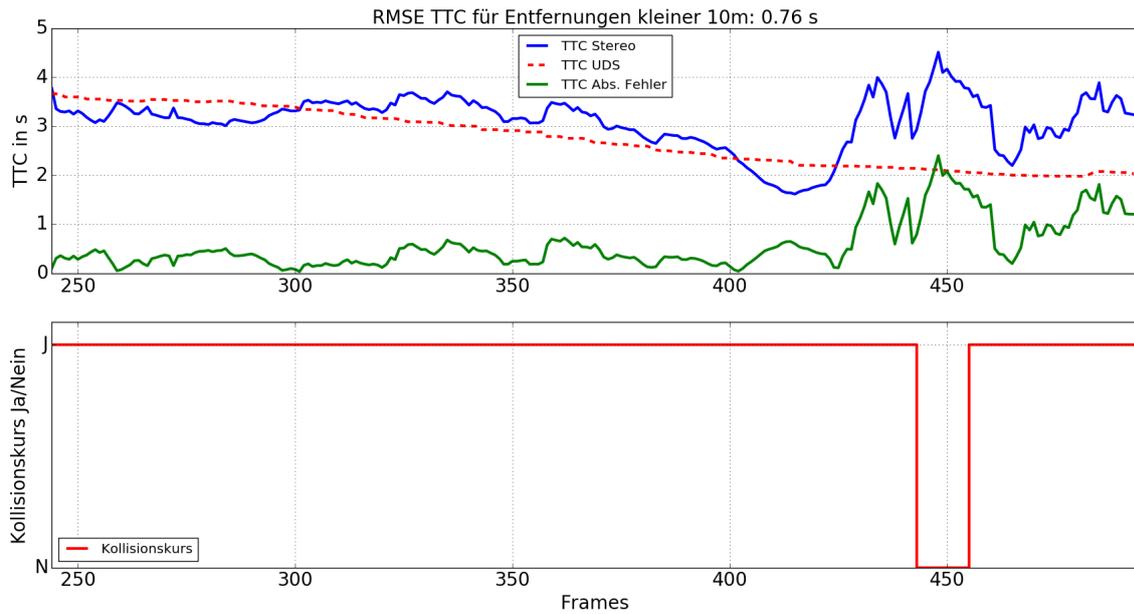
**Abbildung A.69**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.70**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

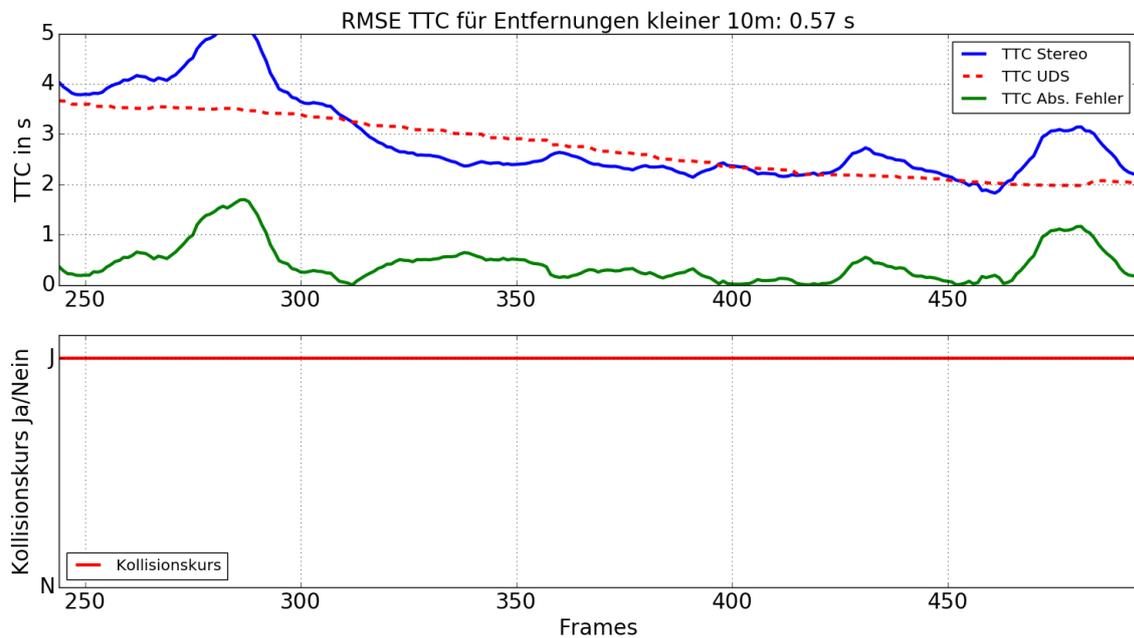


**Abbildung A.71**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

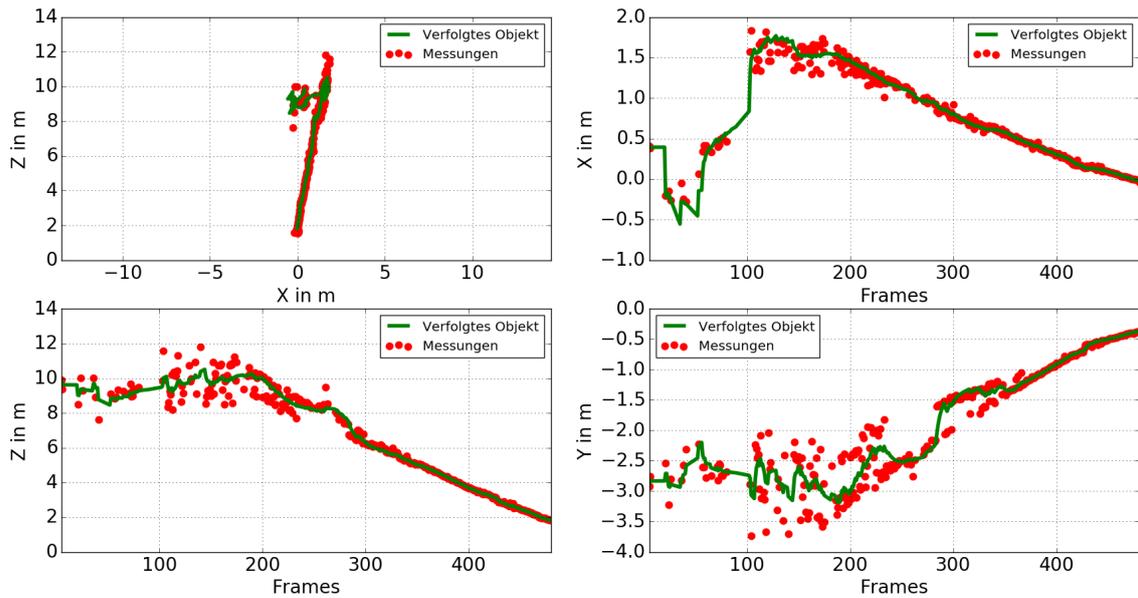


**Abbildung A.72**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.73**

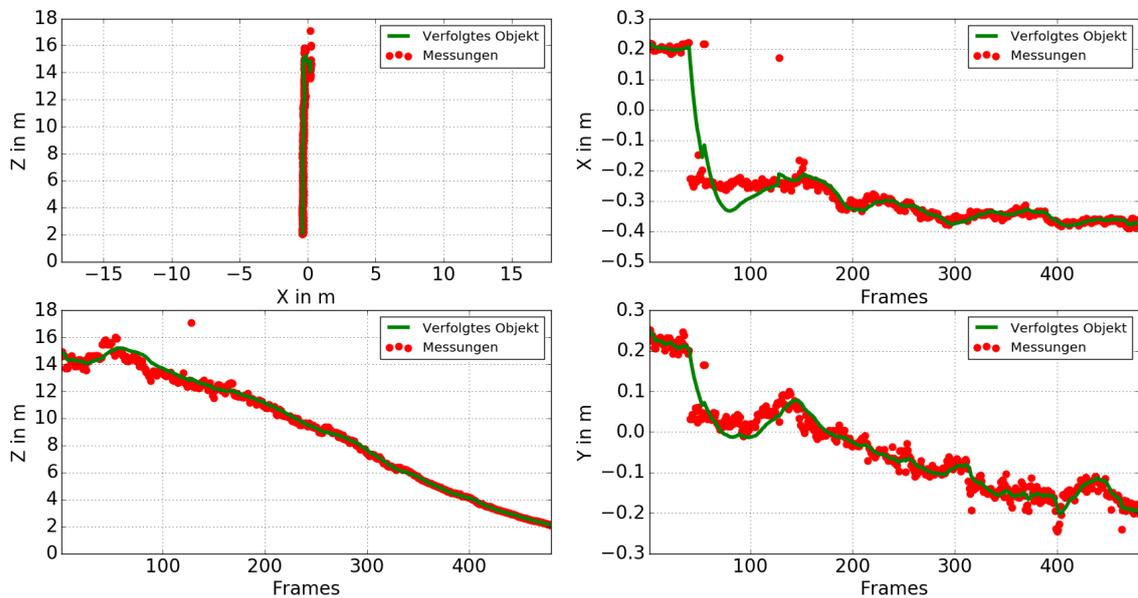
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des Trackers wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.74**

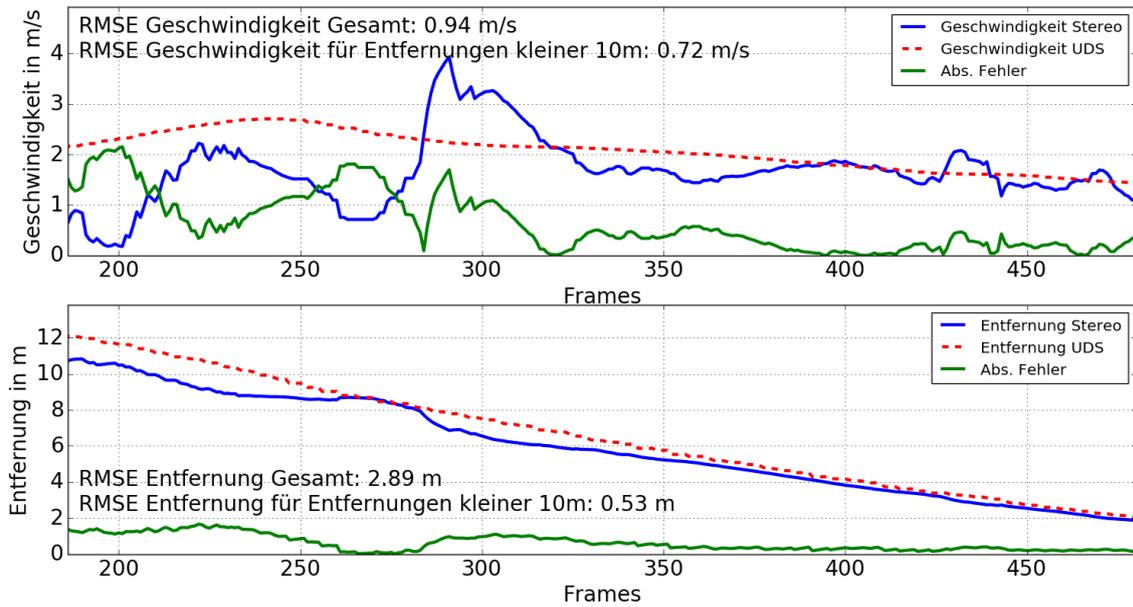
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des Trackers wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

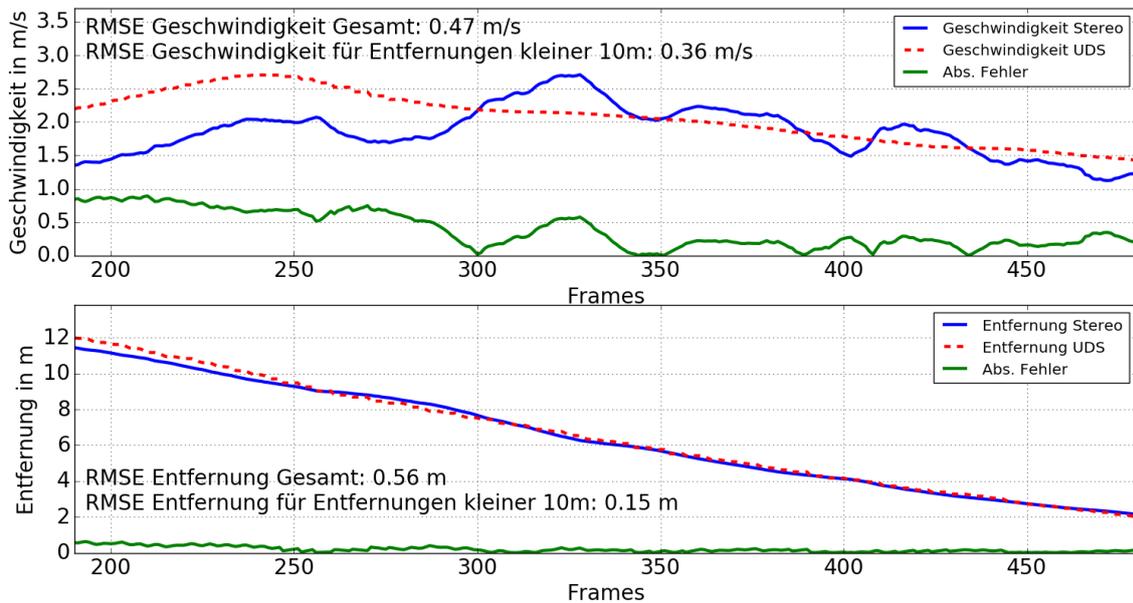
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



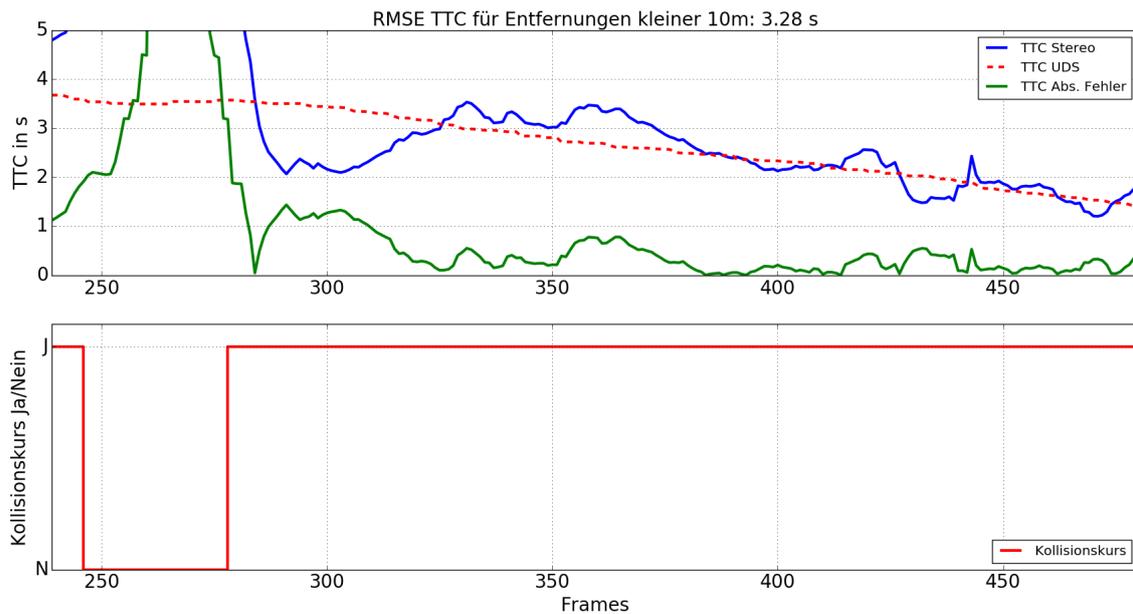
**Abbildung A.75**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.76**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

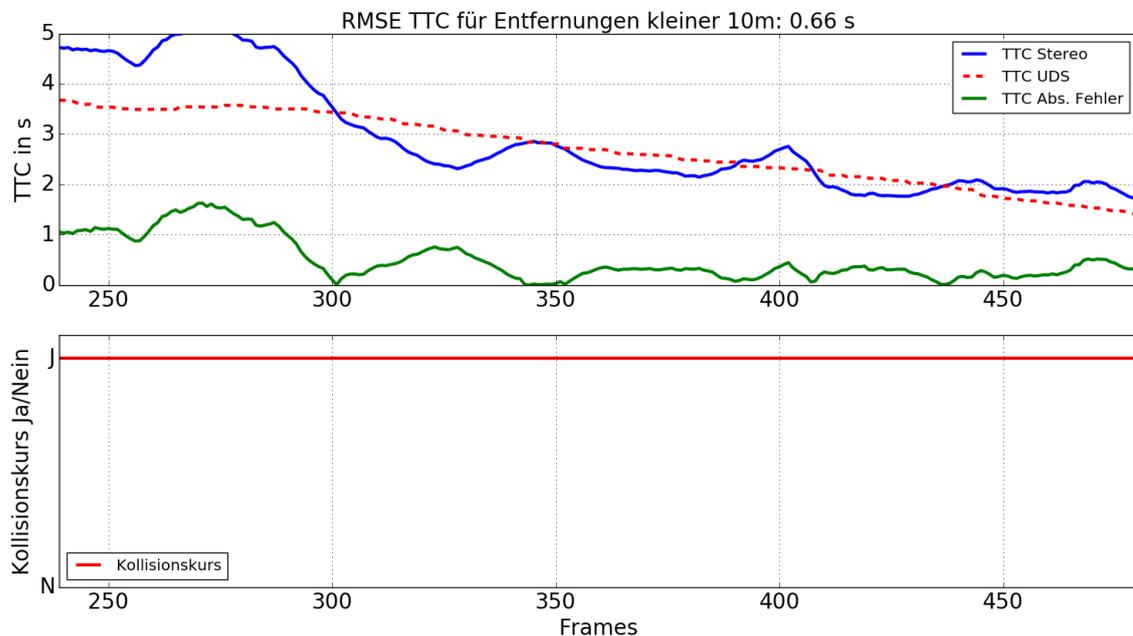


**Abbildung A.77**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

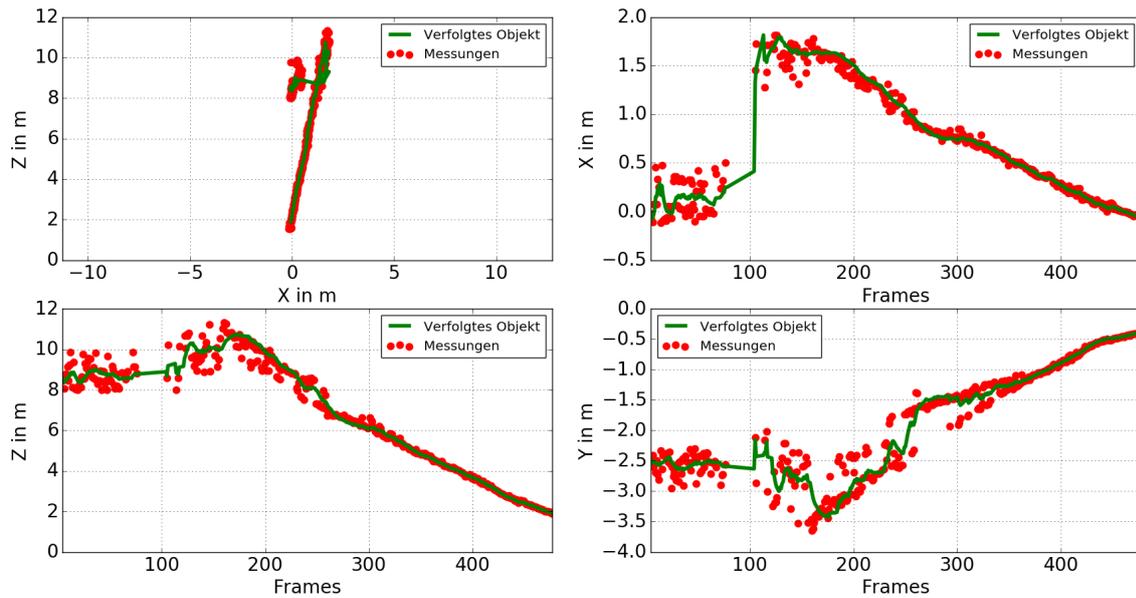


**Abbildung A.78**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.79**

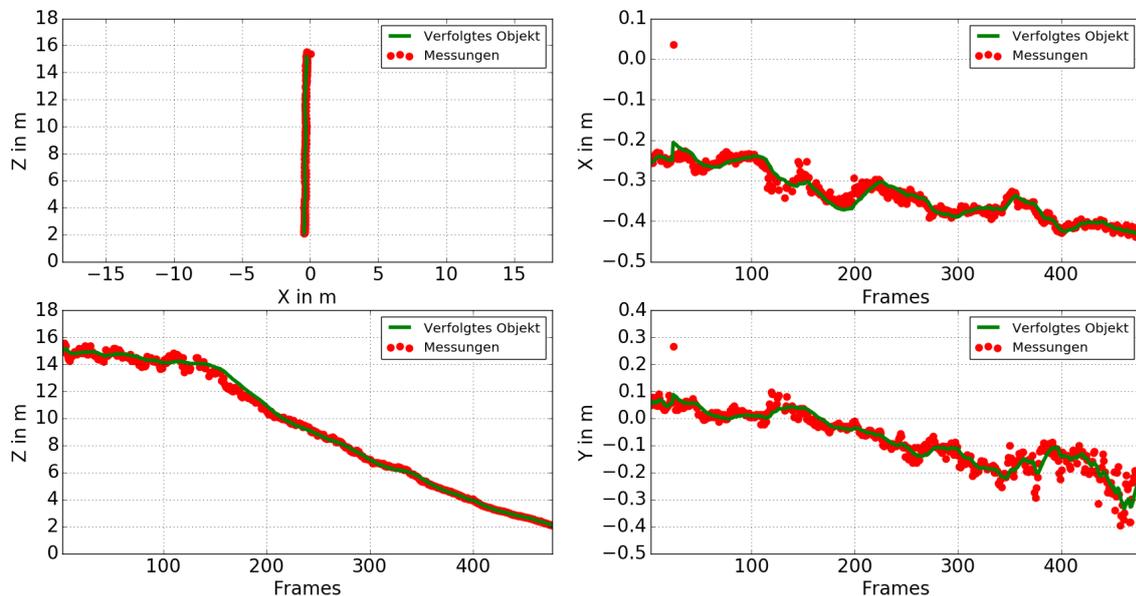
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.80**

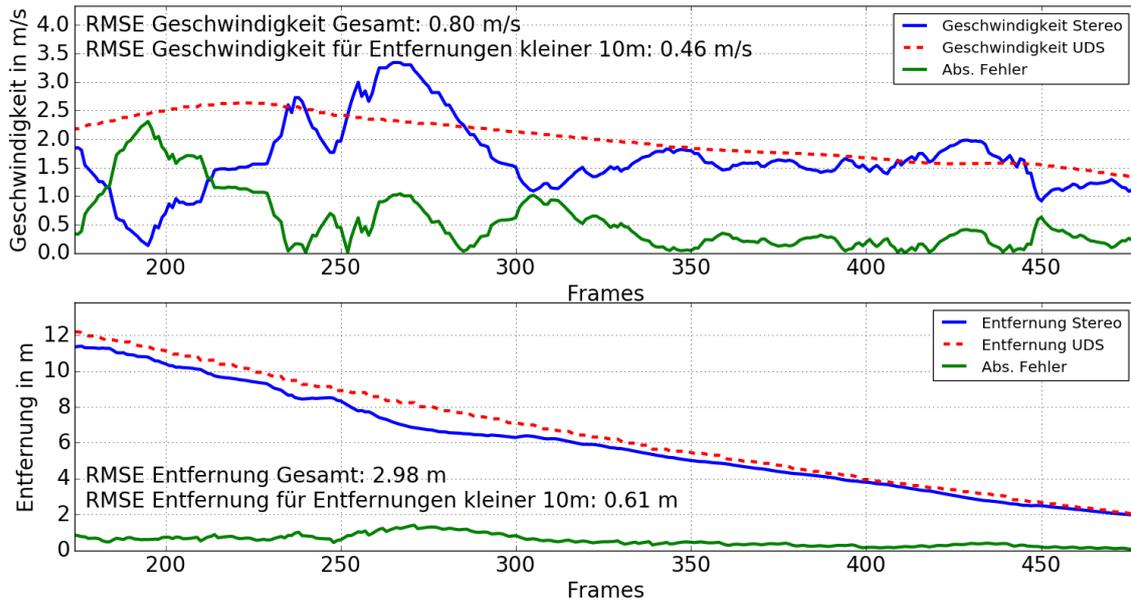
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

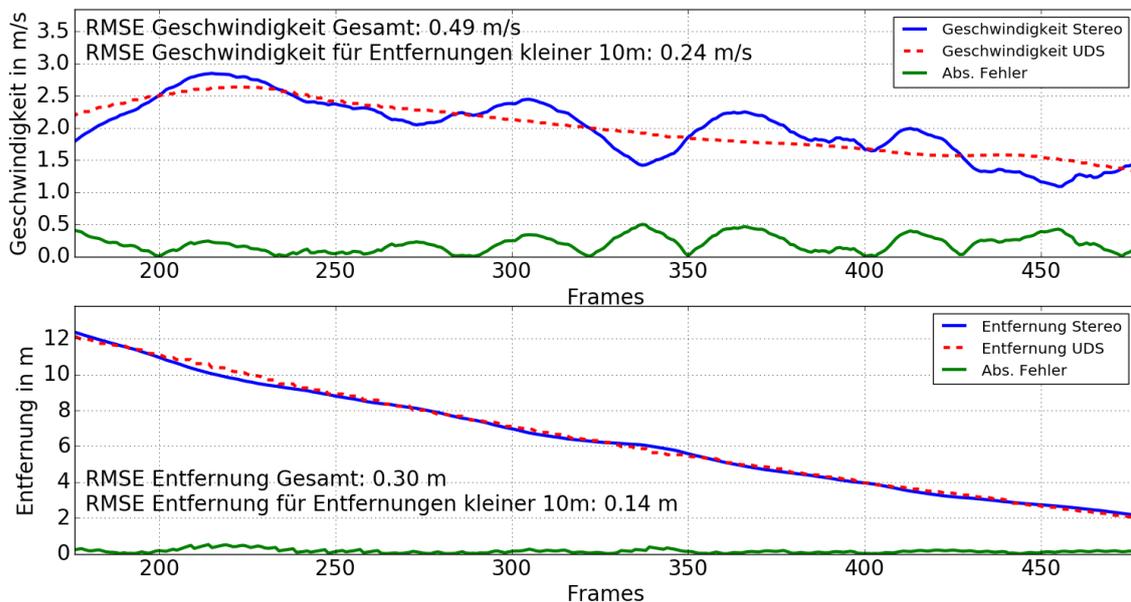
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



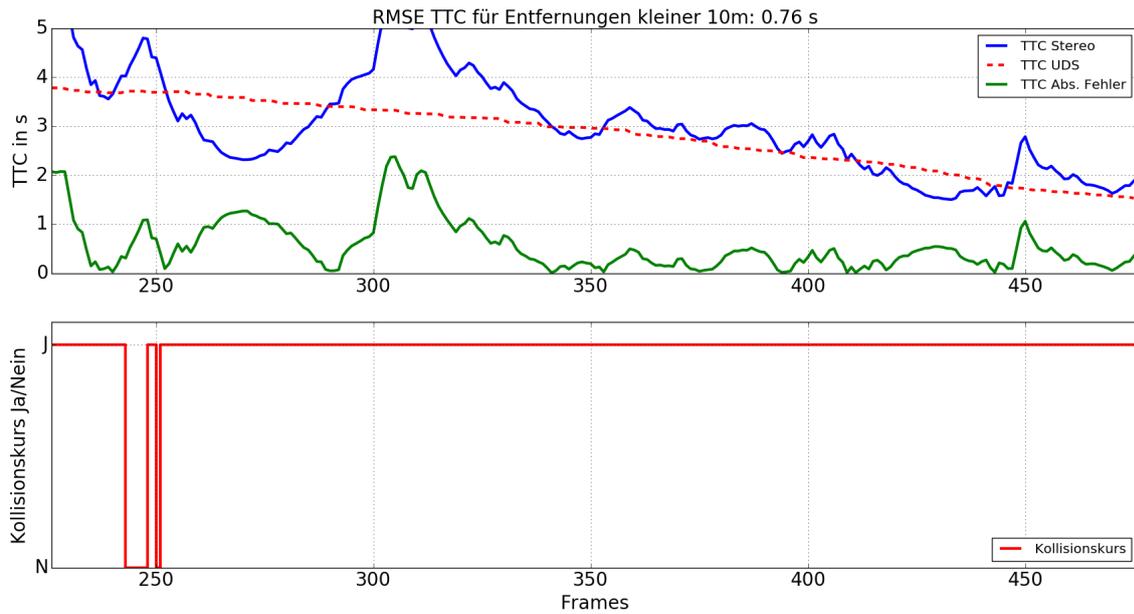
**Abbildung A.81**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.82**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

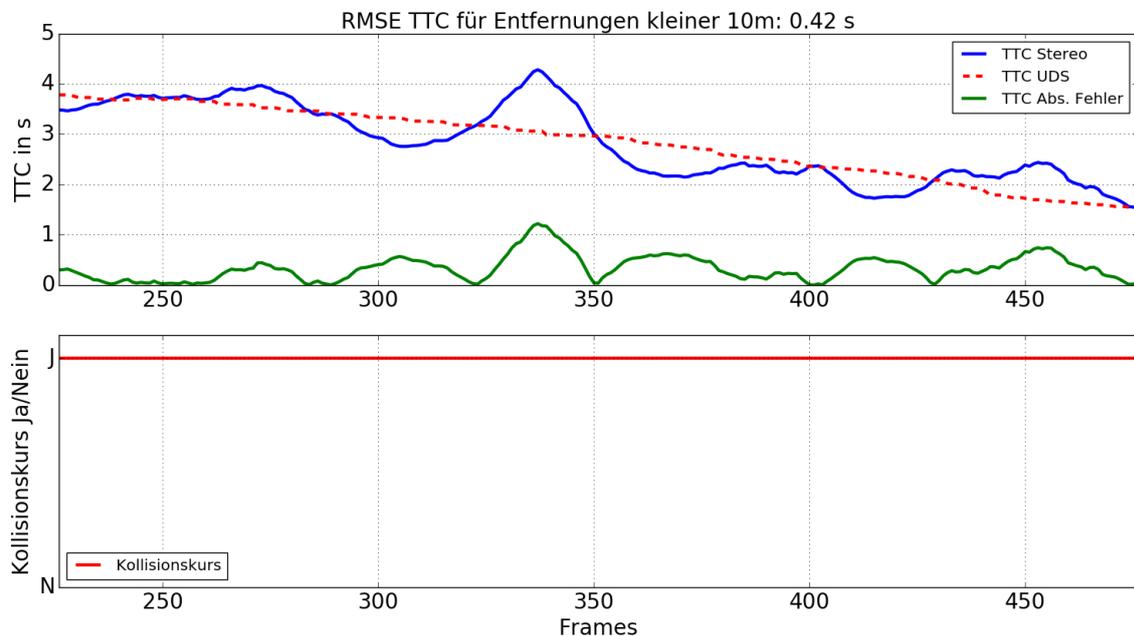


**Abbildung A.83**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

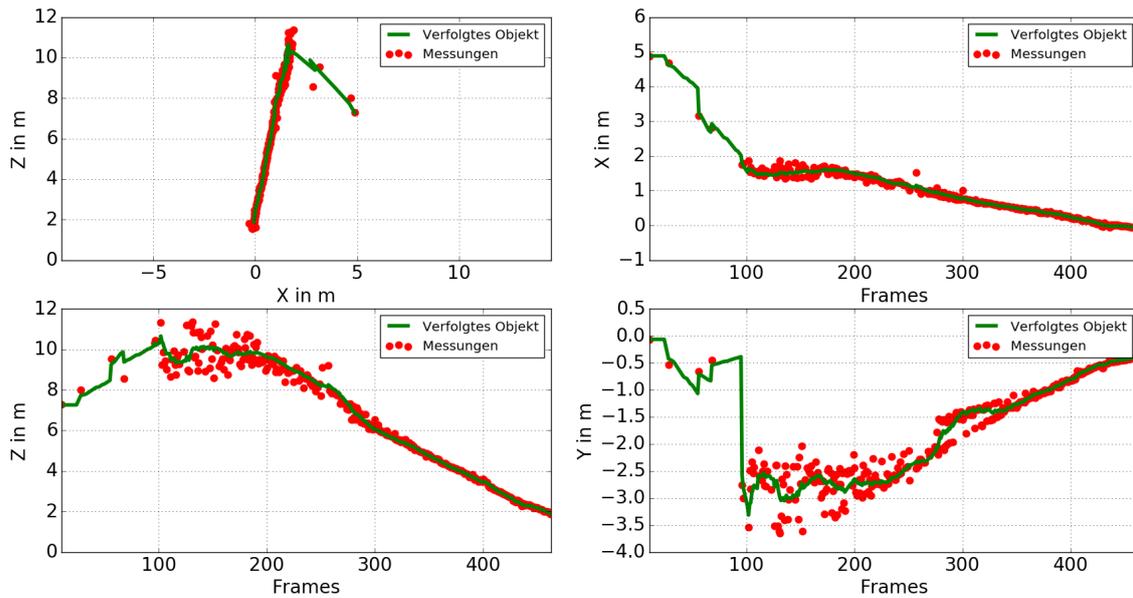


**Abbildung A.84**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.85**

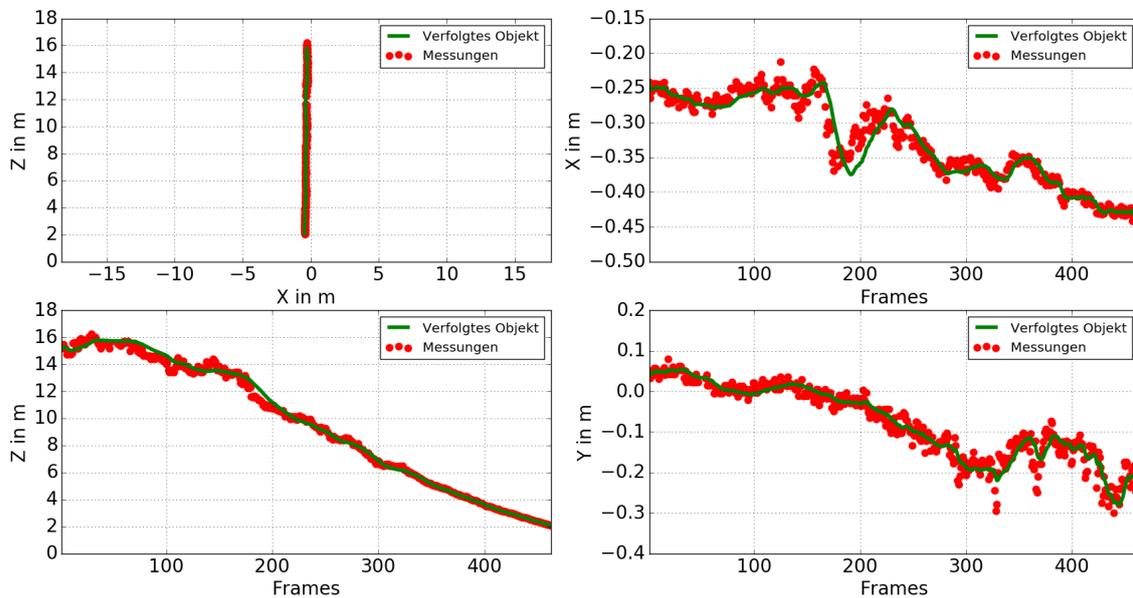
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.86**

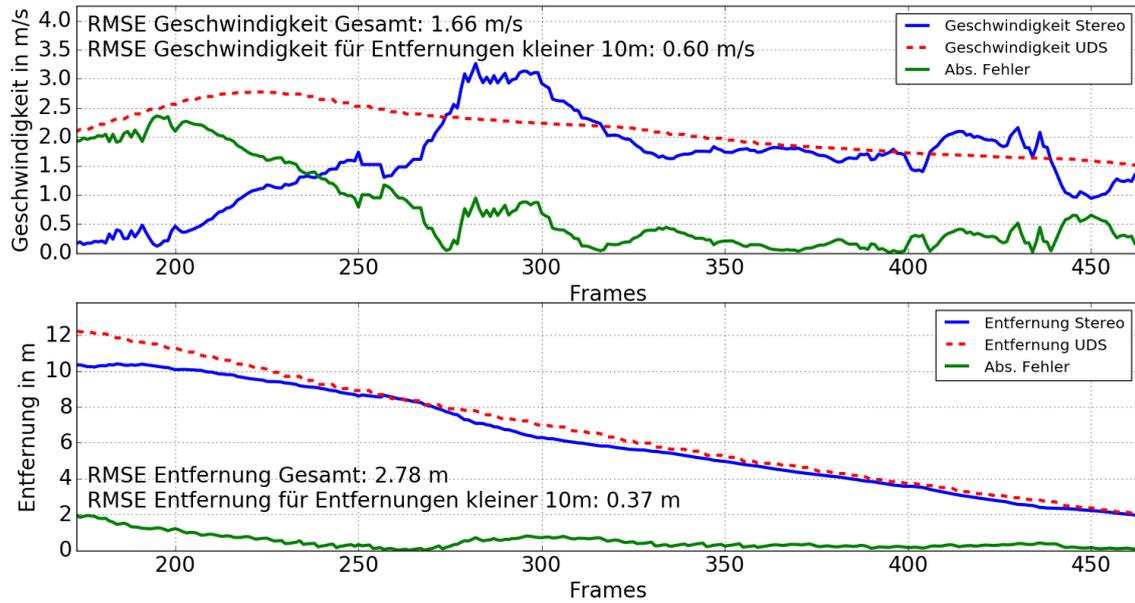
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,4m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

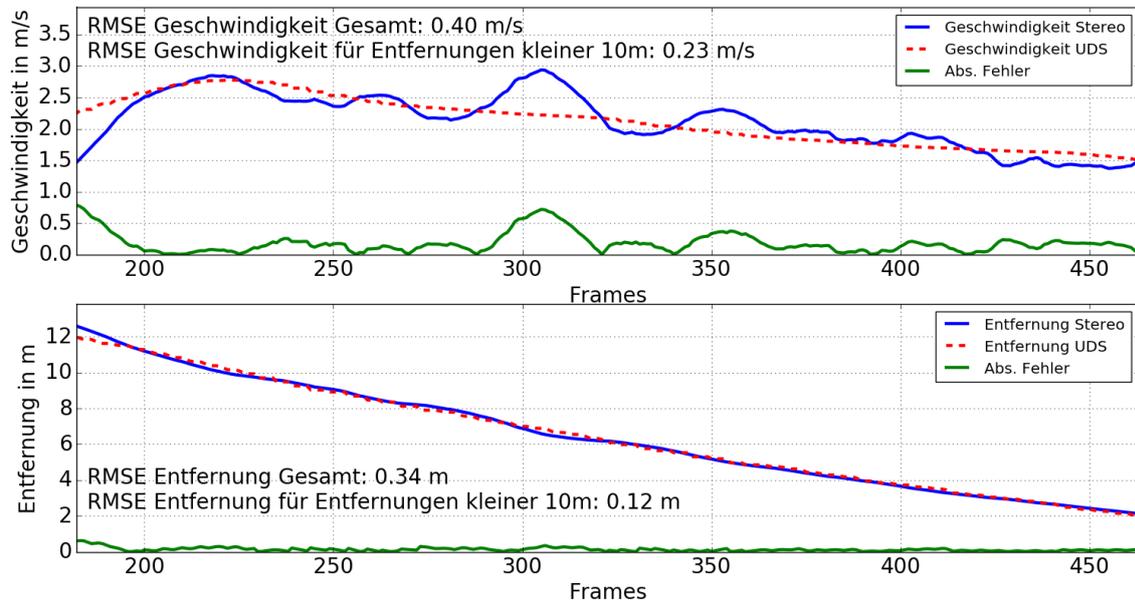
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



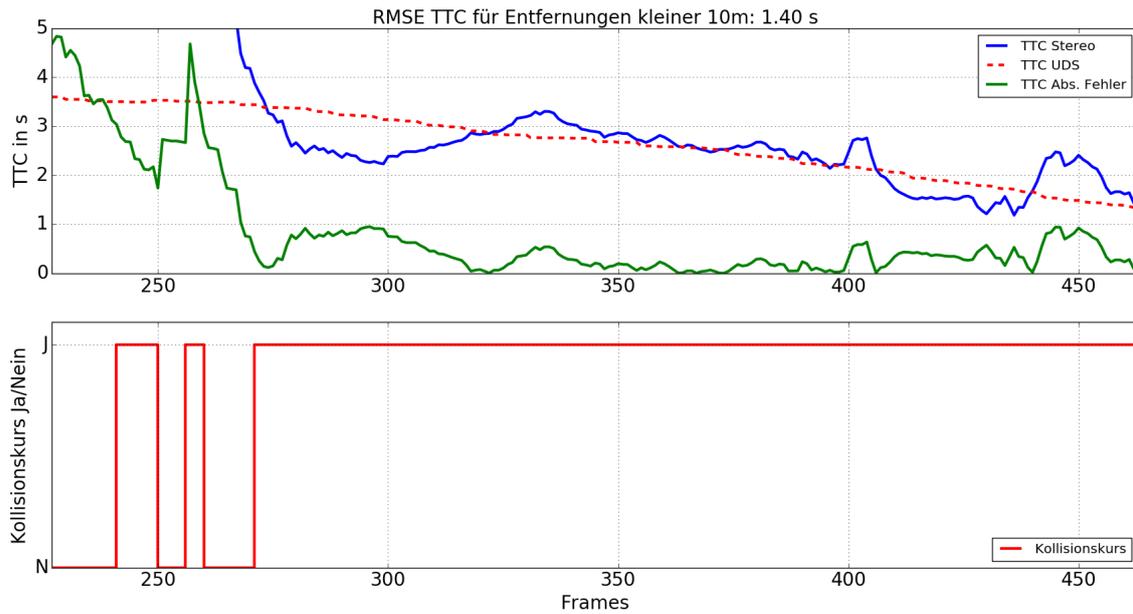
**Abbildung A.87**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.88**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

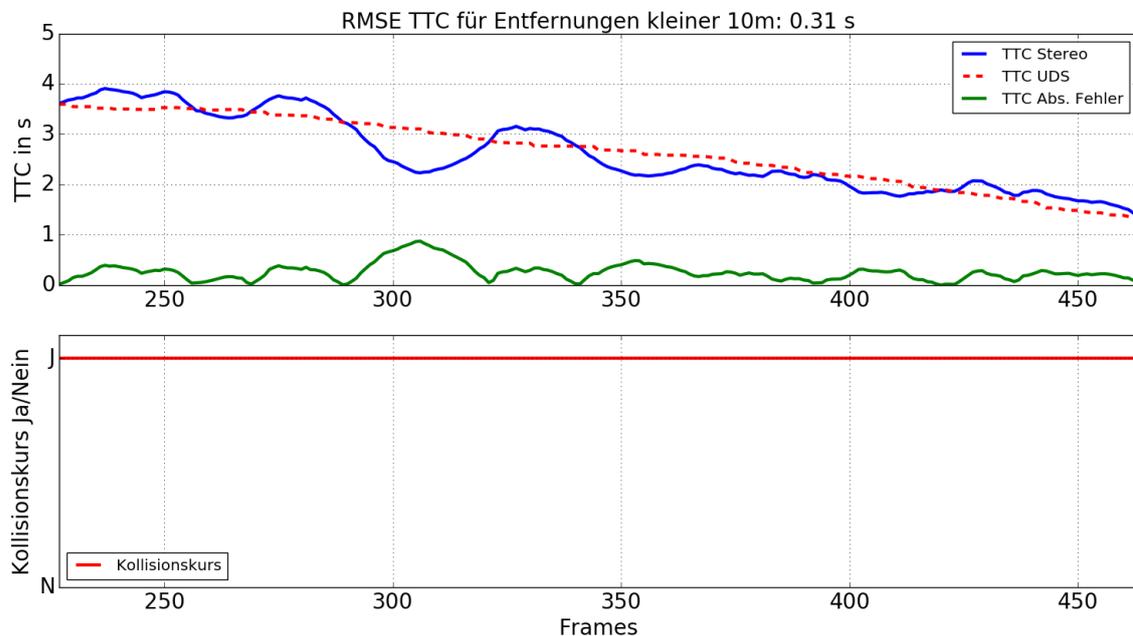


**Abbildung A.89**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

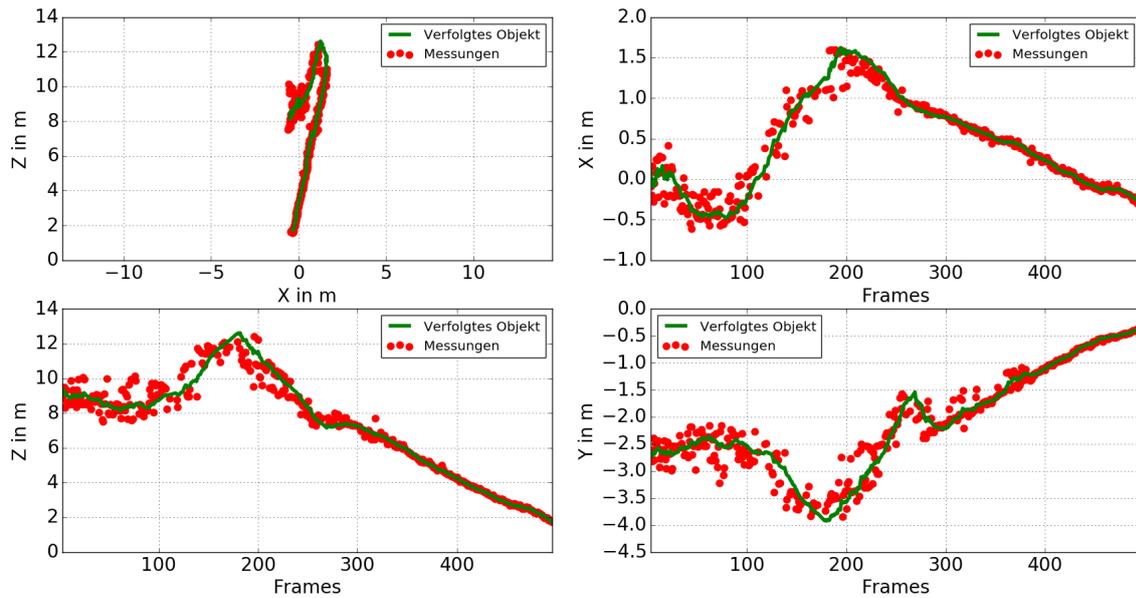


**Abbildung A.90**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.91**

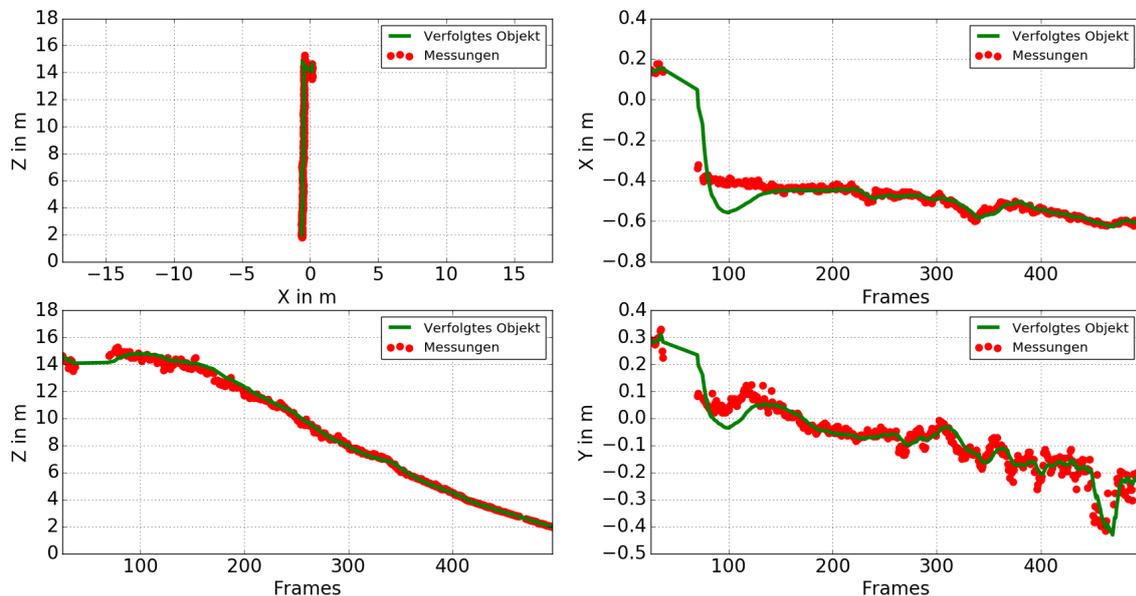
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den sechzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.92**

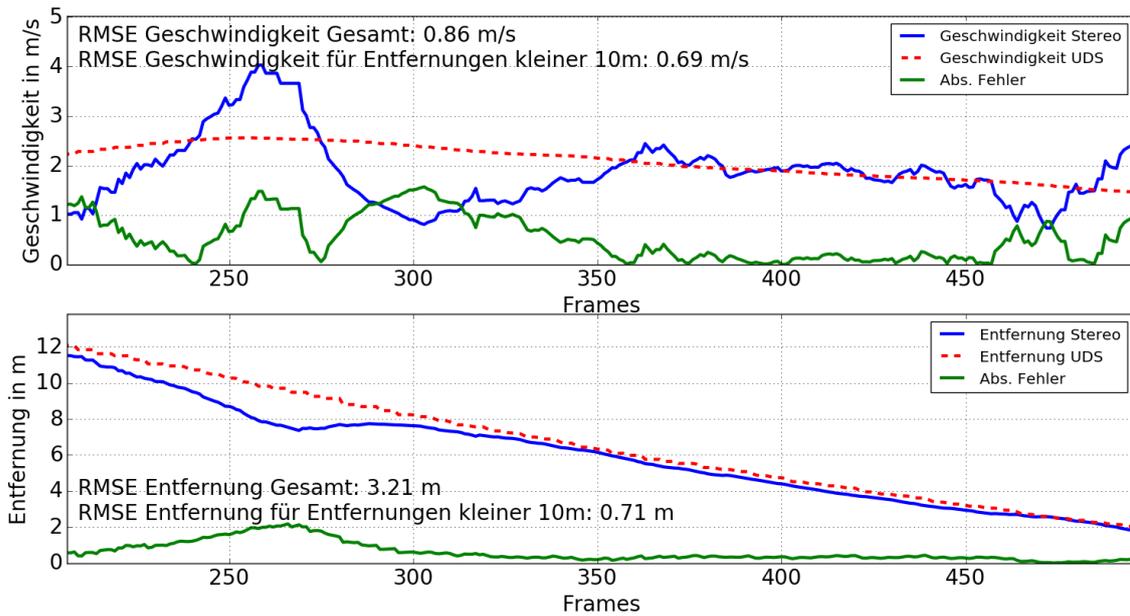
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den sechzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

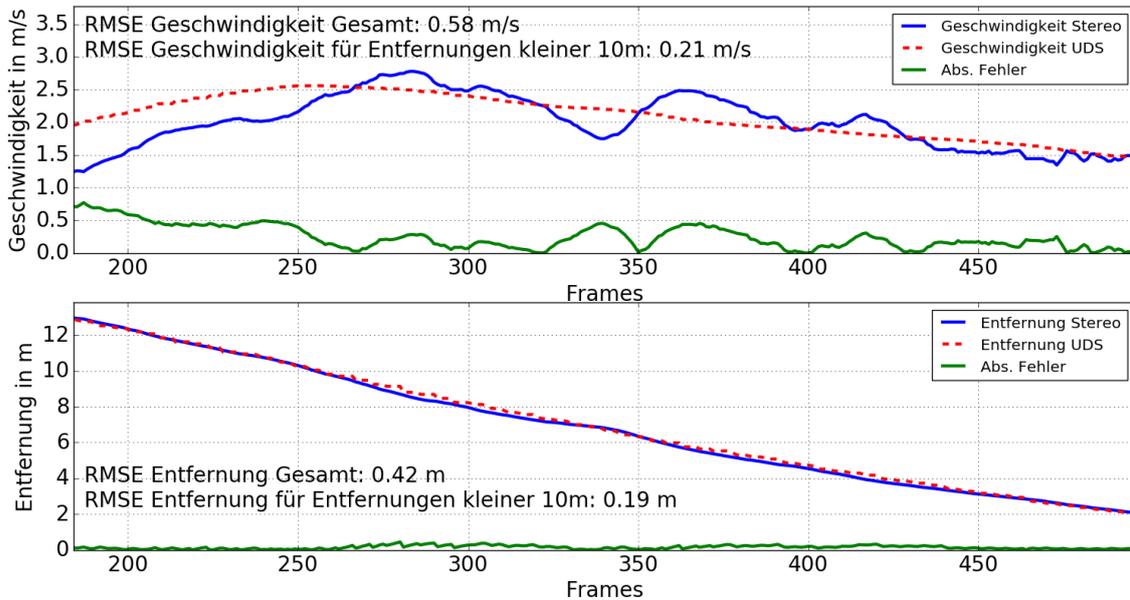
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



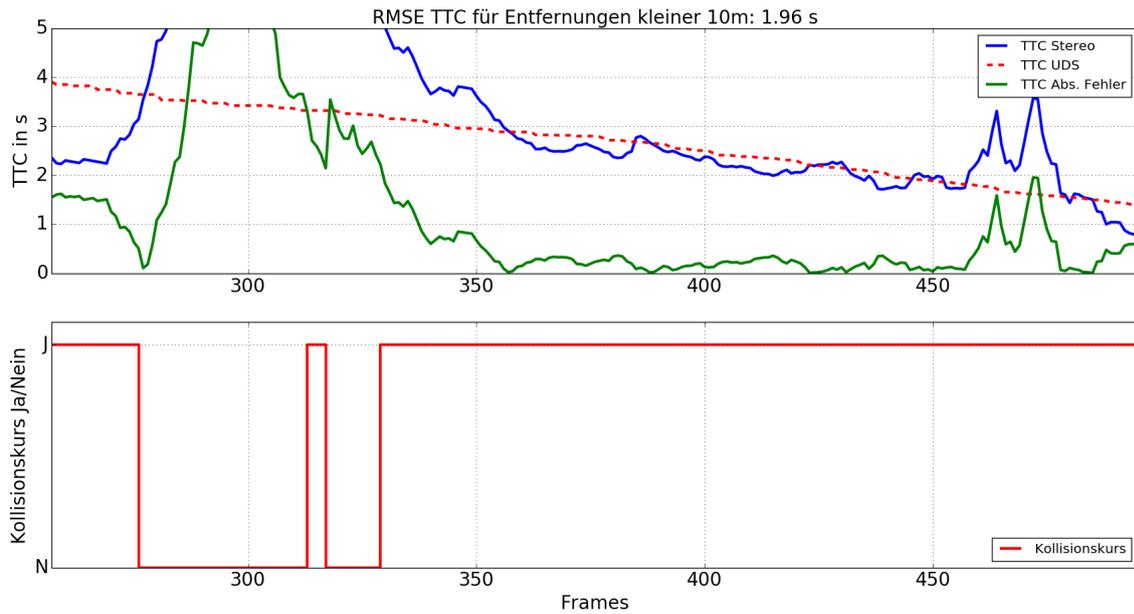
**Abbildung A.93**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den sechzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.94**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den sechzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

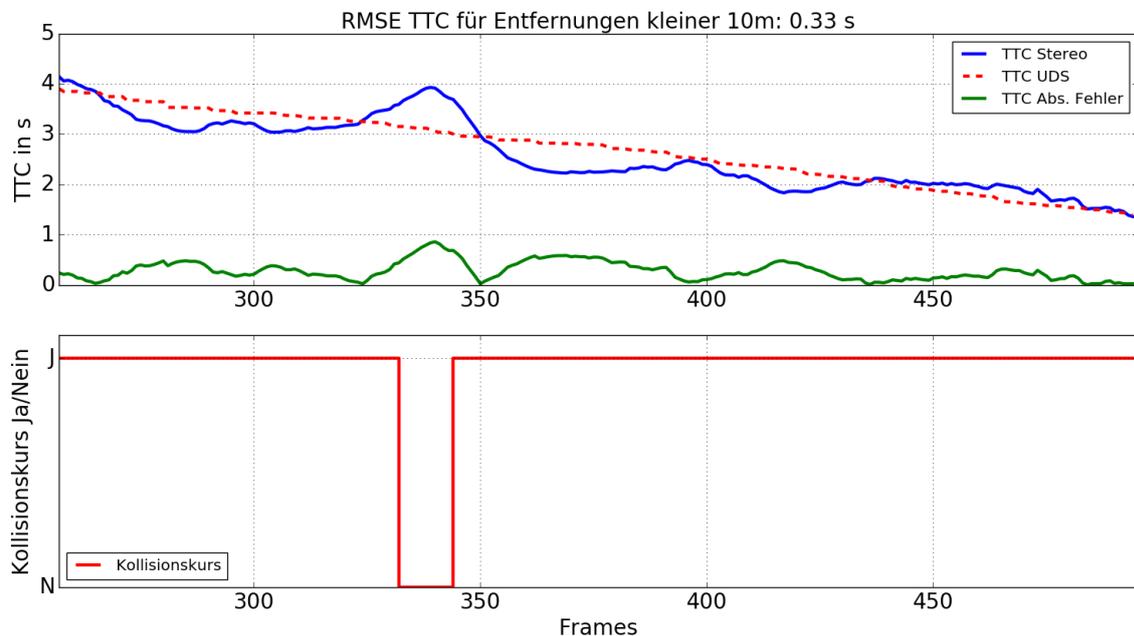


**Abbildung A.95**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den sechzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

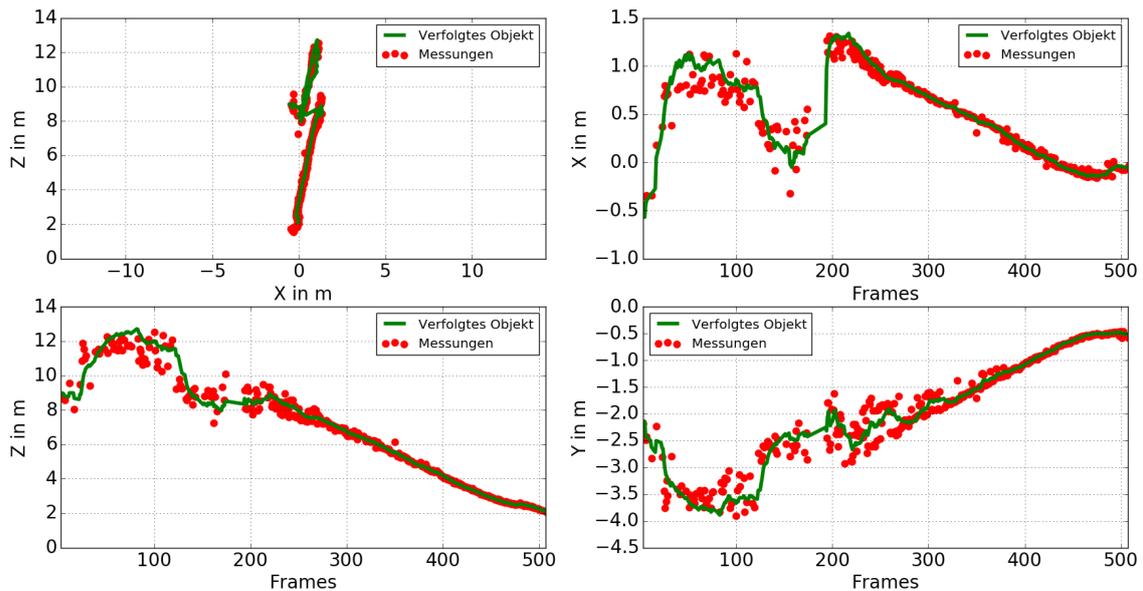


**Abbildung A.96**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den sechzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.97**

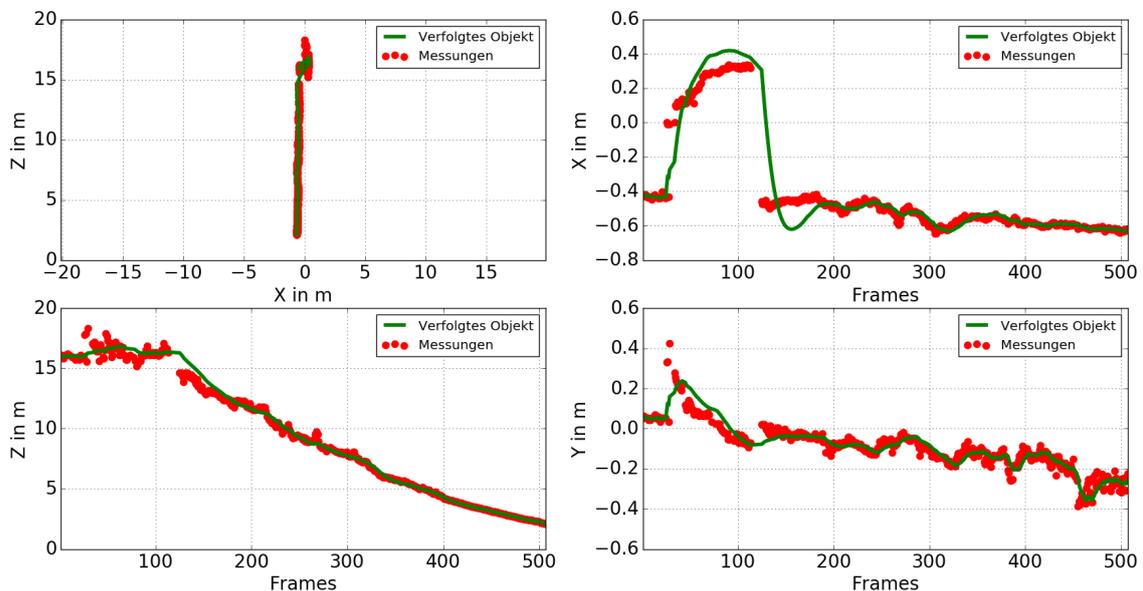
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den siebzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.98**

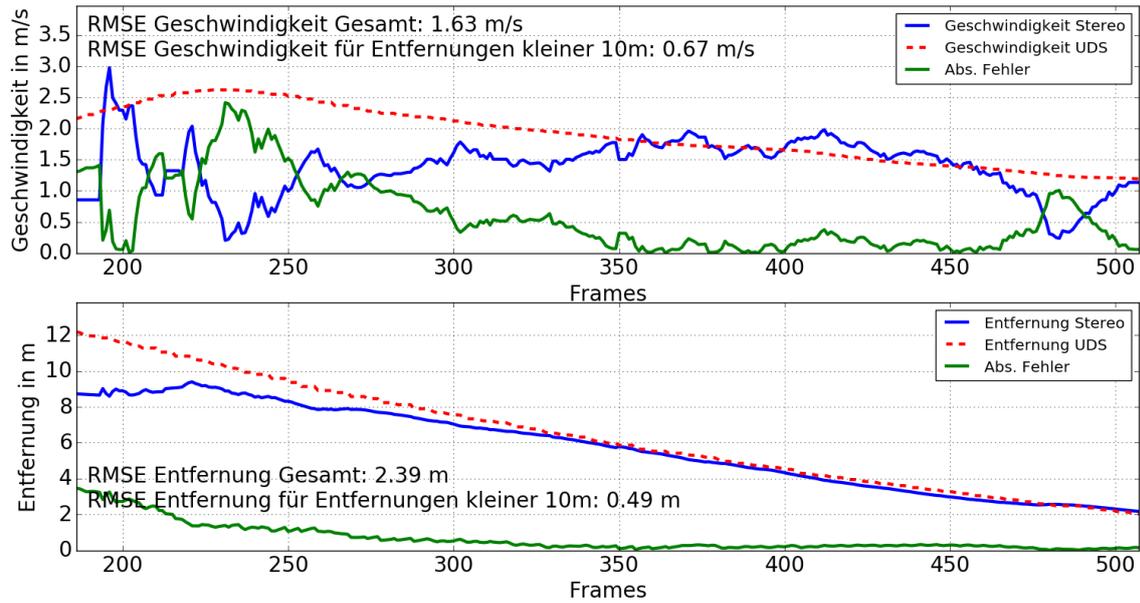
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den siebzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

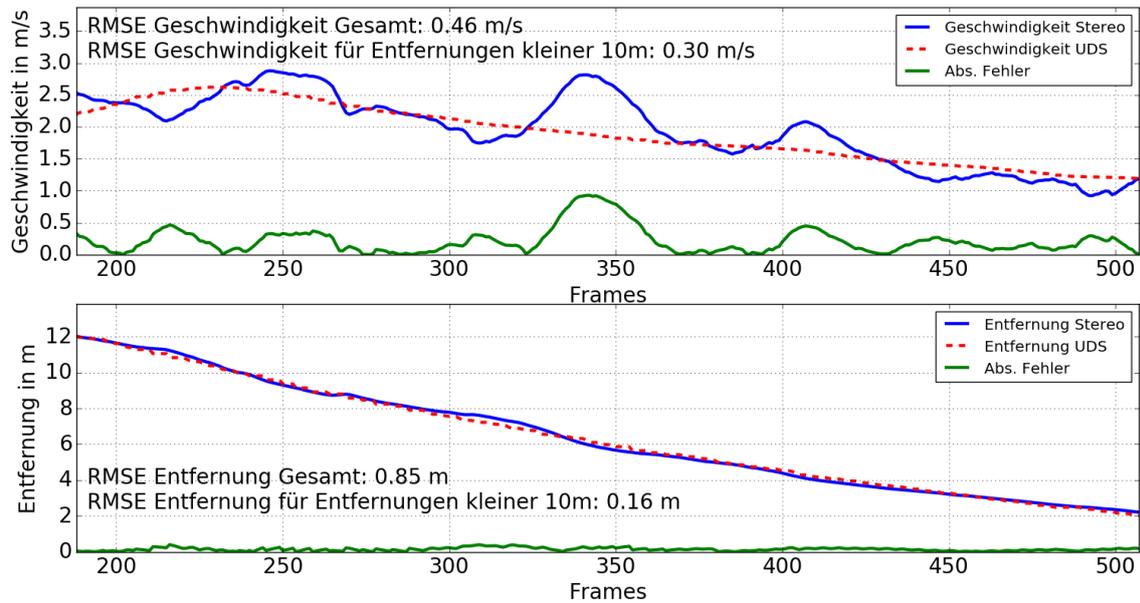
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



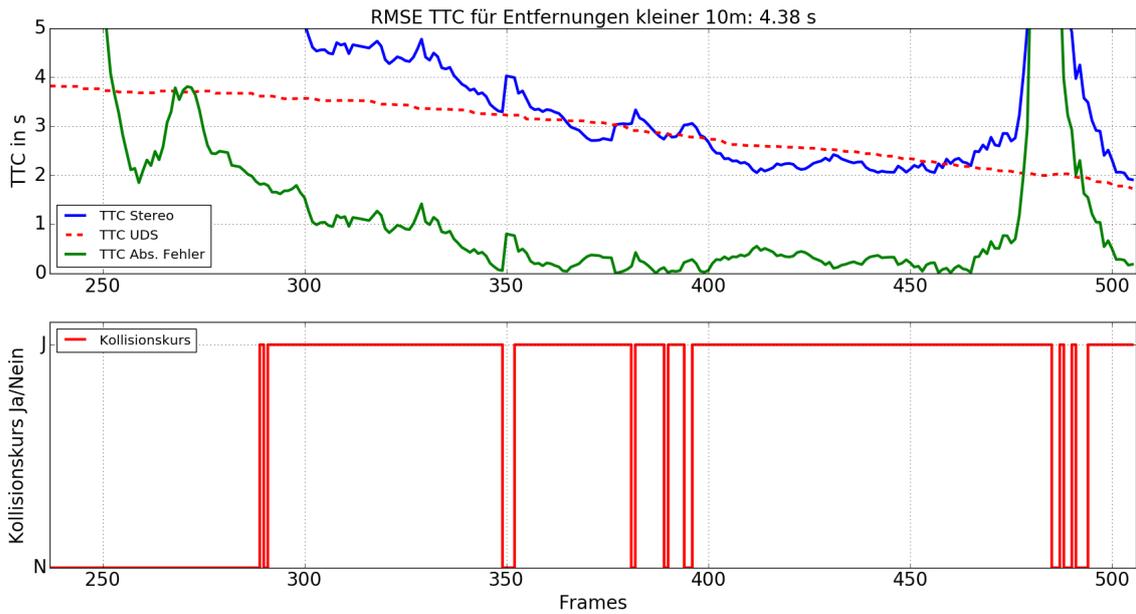
**Abbildung A.99**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den siebzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.100**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den siebzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

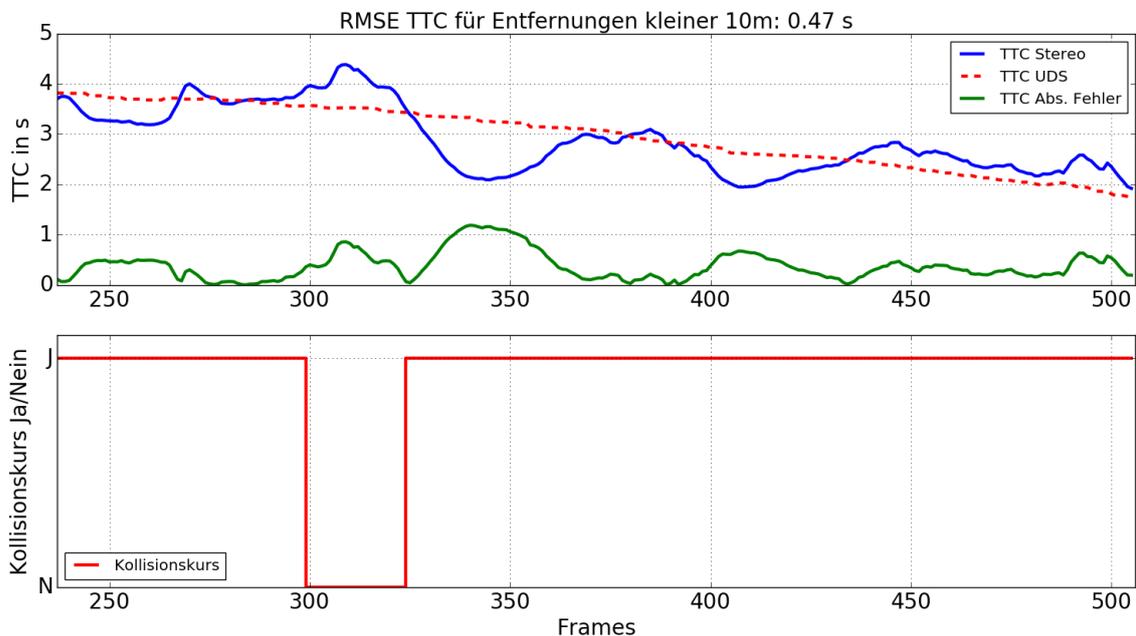


**Abbildung A.101**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den siebzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

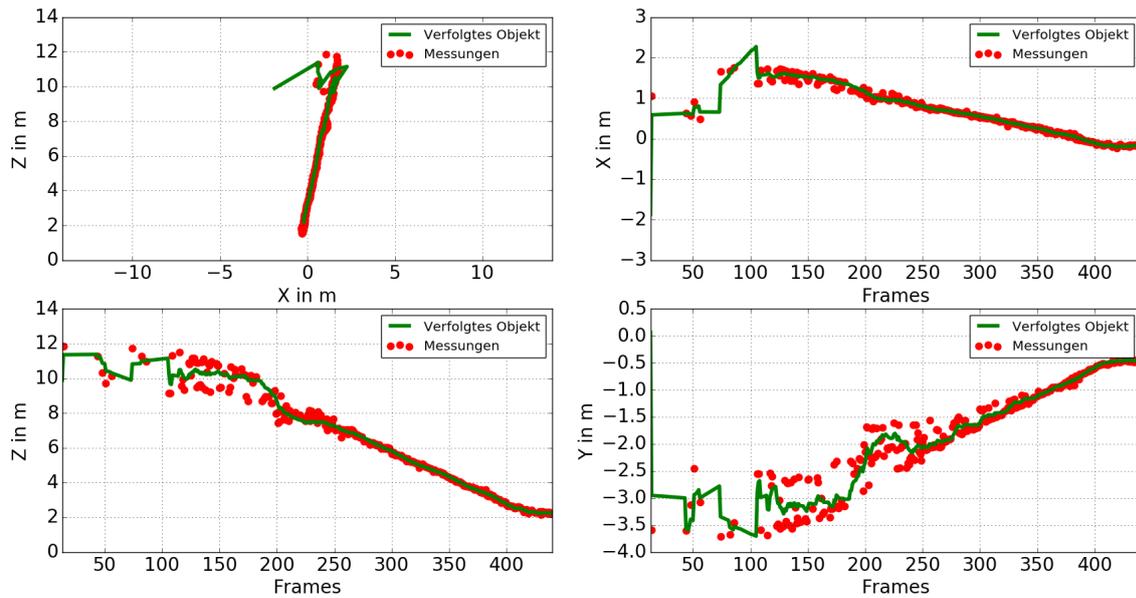


**Abbildung A.102**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den siebzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.103**

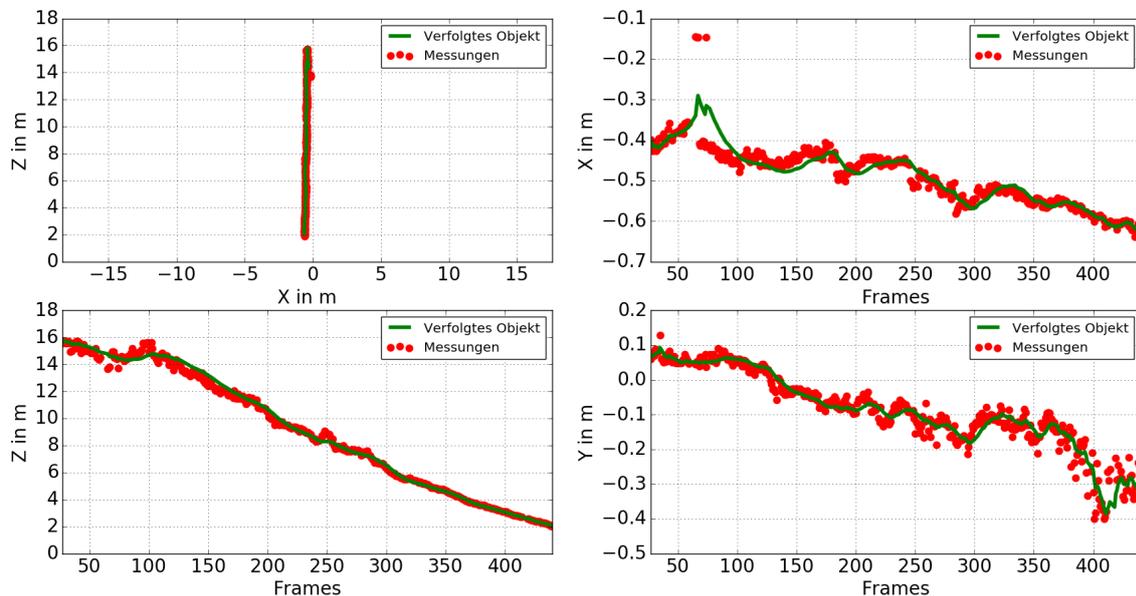
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den achzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.104**

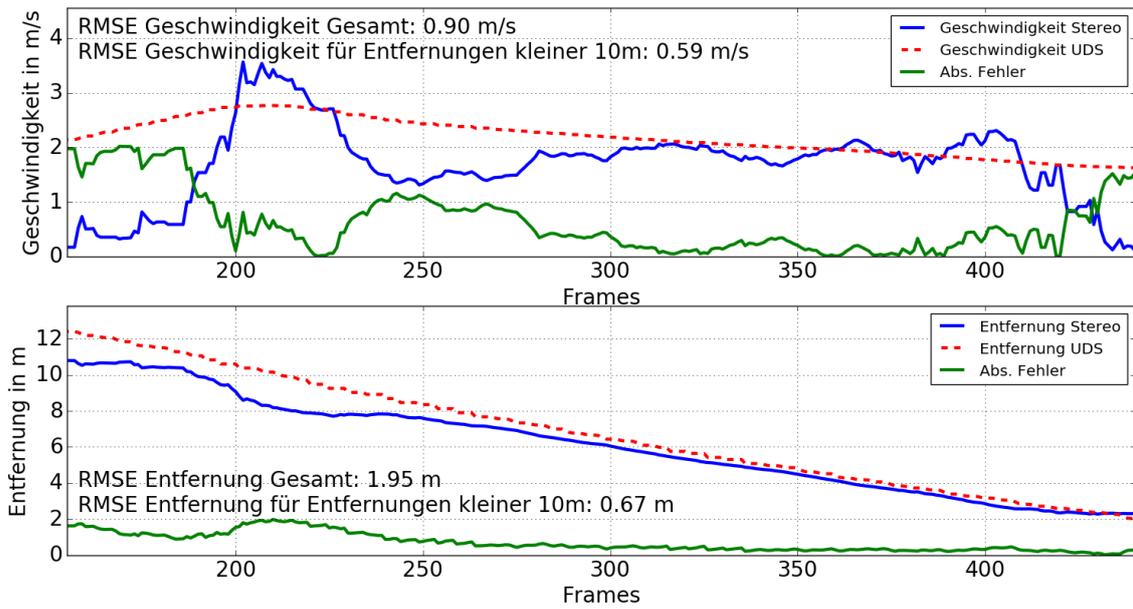
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den achzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

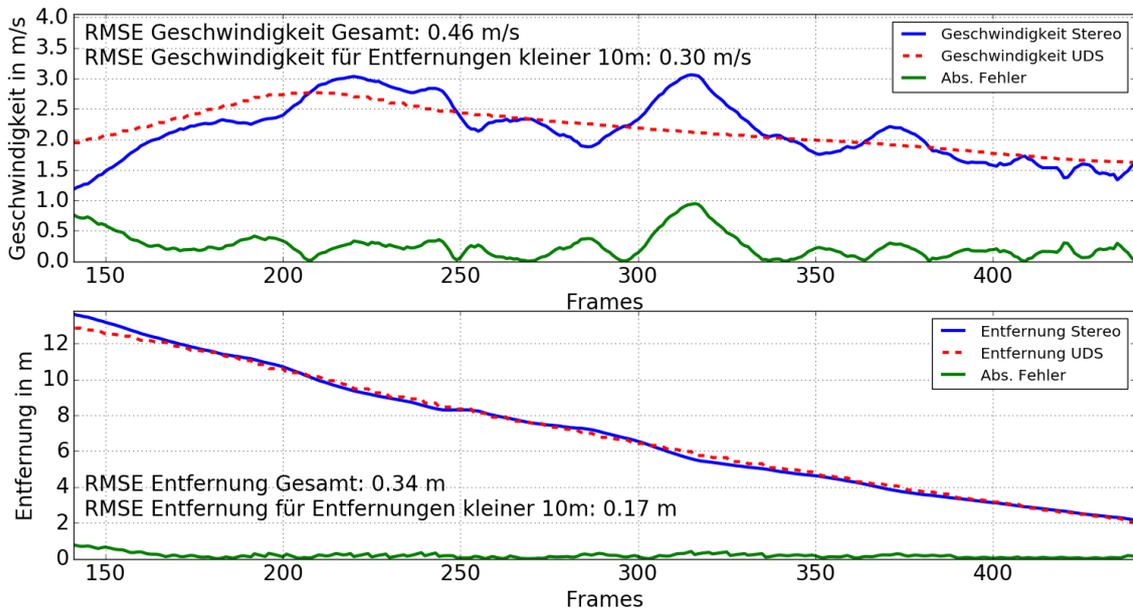
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



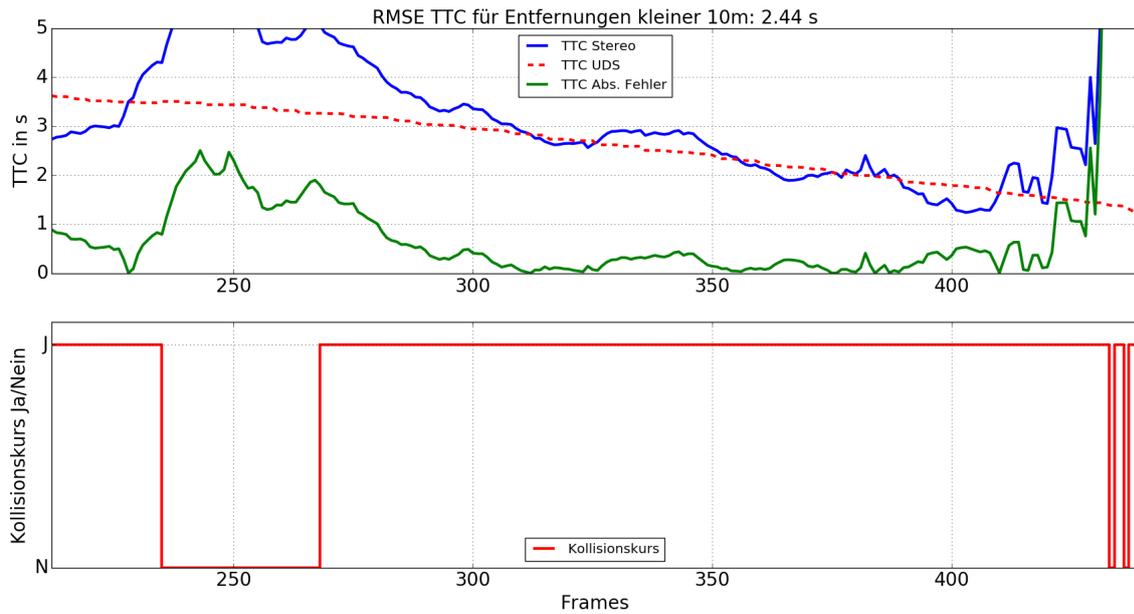
**Abbildung A.105**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den achzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.106**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den achzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

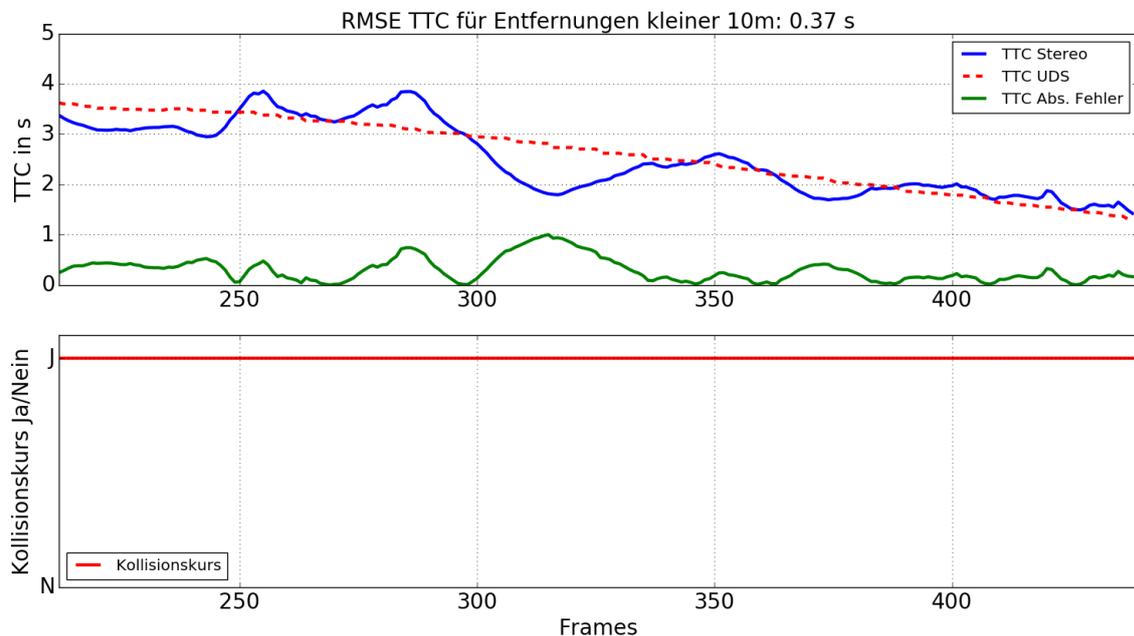


**Abbildung A.107**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den achtzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

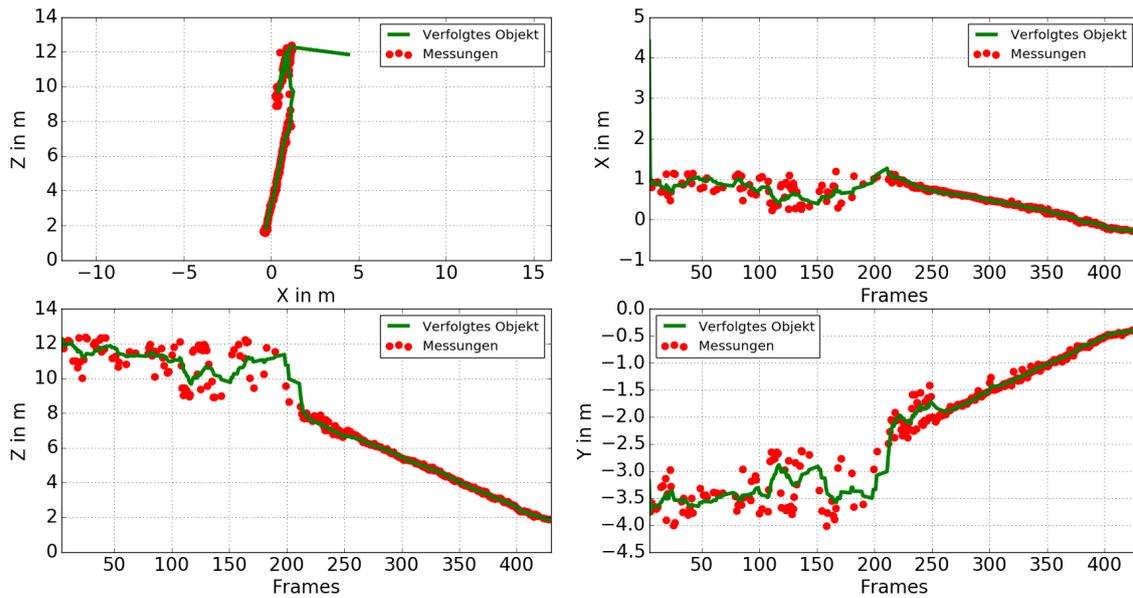


**Abbildung A.108**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den achtzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.109**

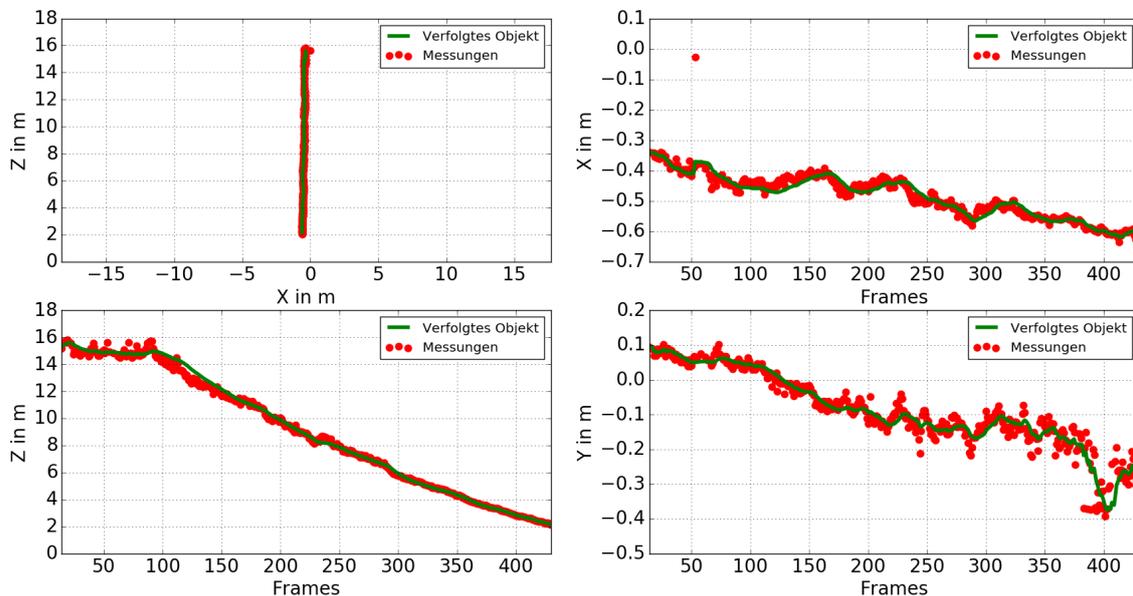
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den neunzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.110**

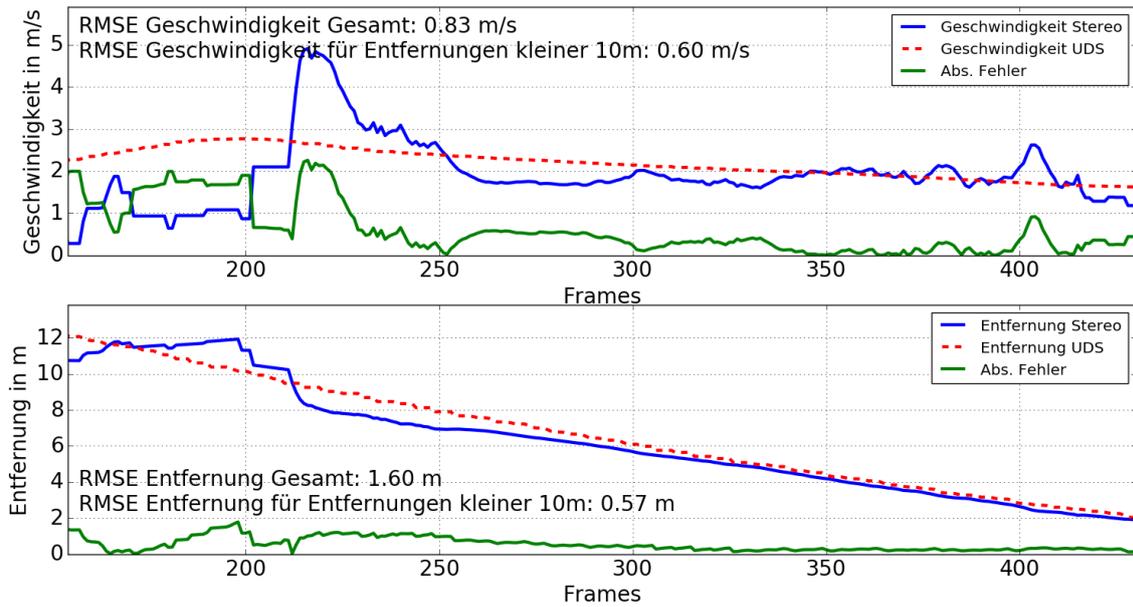
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den neunzehnten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

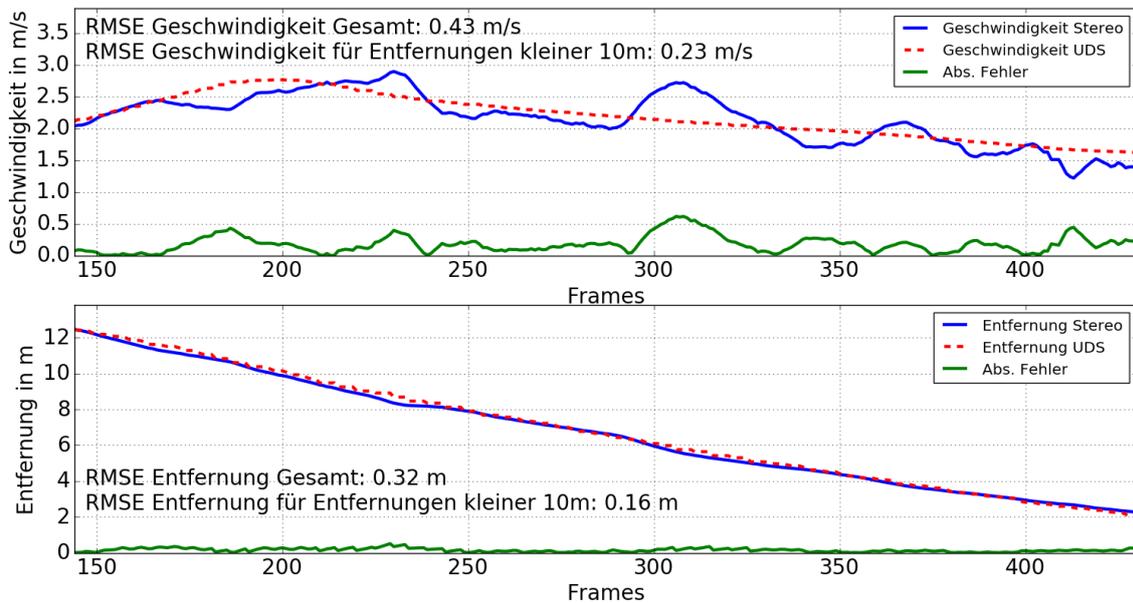
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



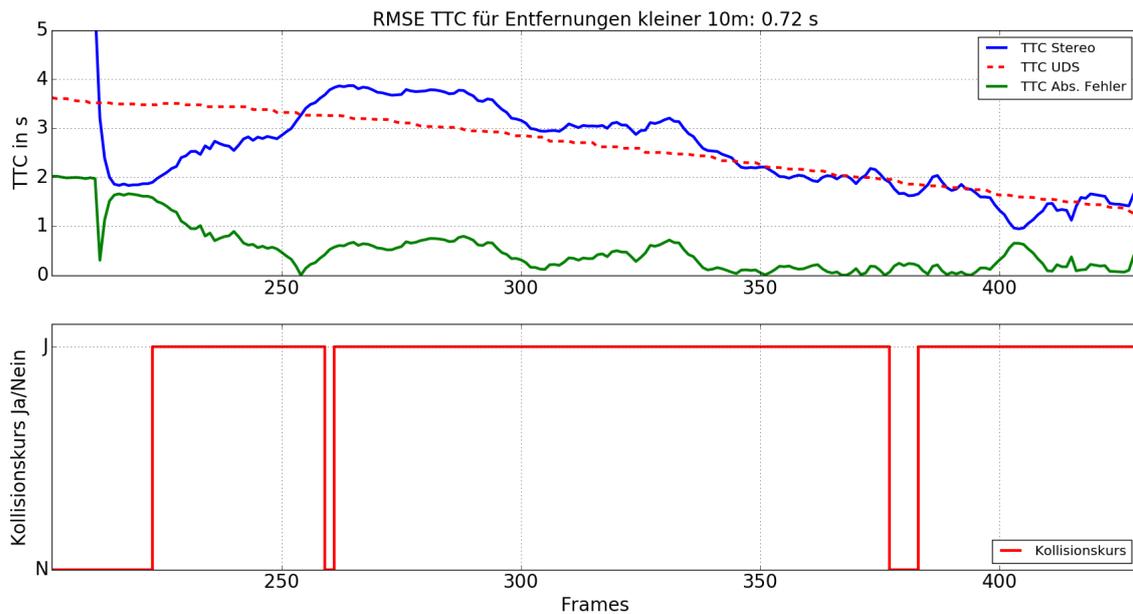
**Abbildung A.111**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den neunzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.112**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den neunzehnten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

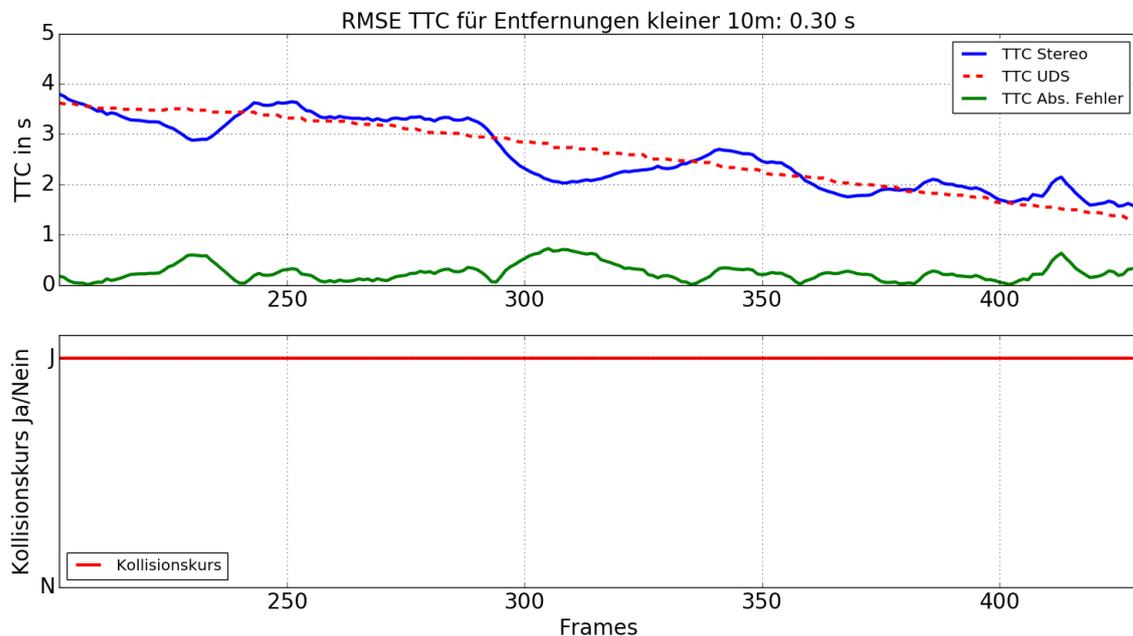


**Abbildung A.113**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den neunzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

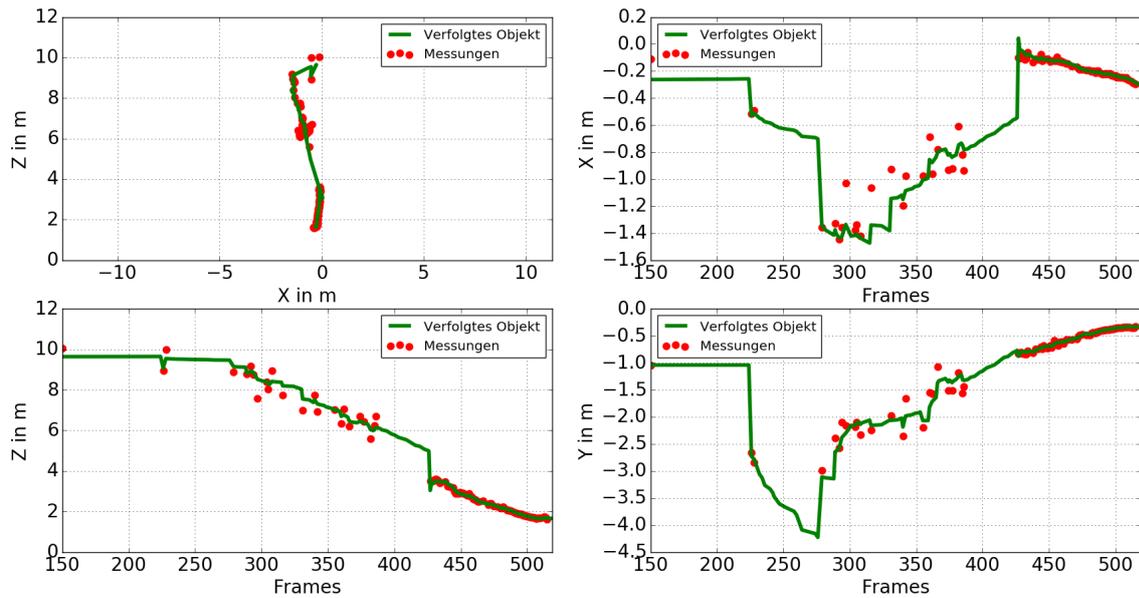


**Abbildung A.114**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den neunzehnten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.115**

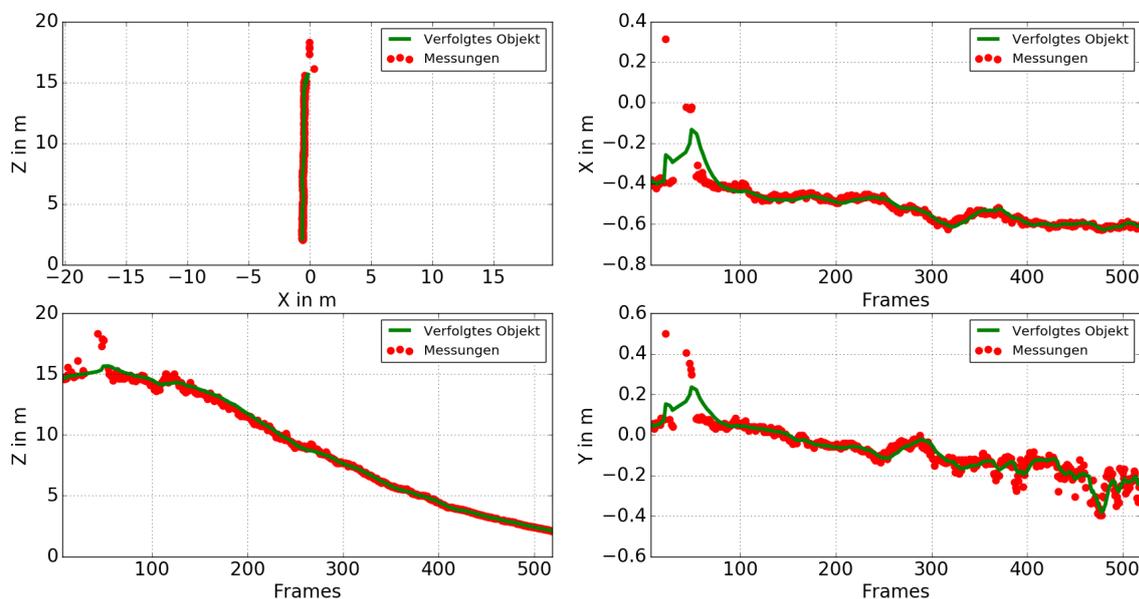
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den zwanzigsten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.116**

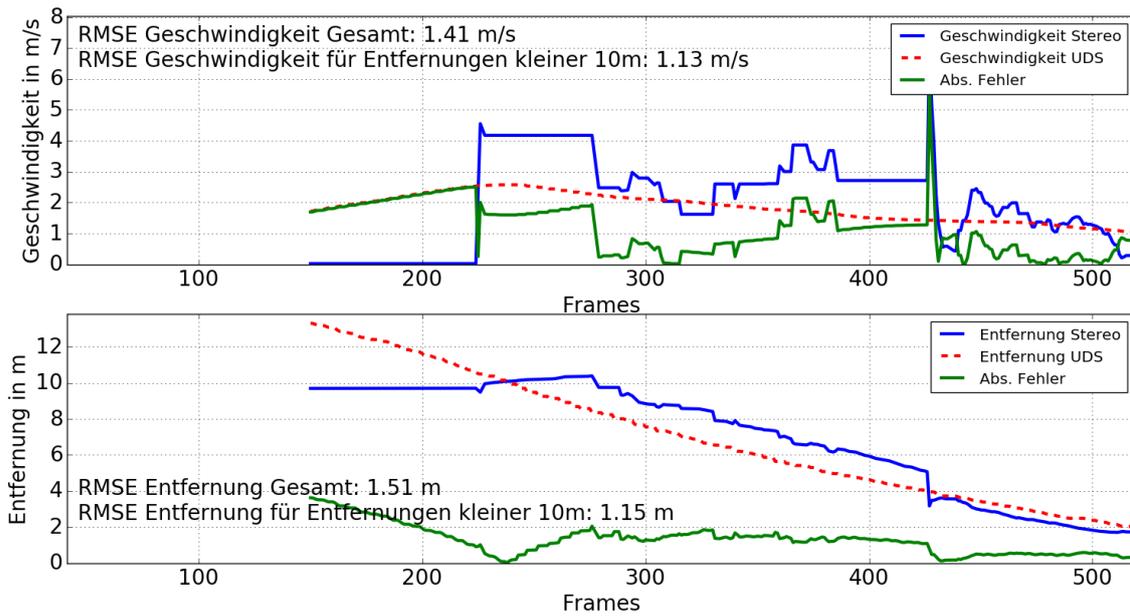
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den zwanzigsten Versuch der TTC Validationsversuche im Szenario 1. Es liegt ein *Offset* von 0,6 m vor. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

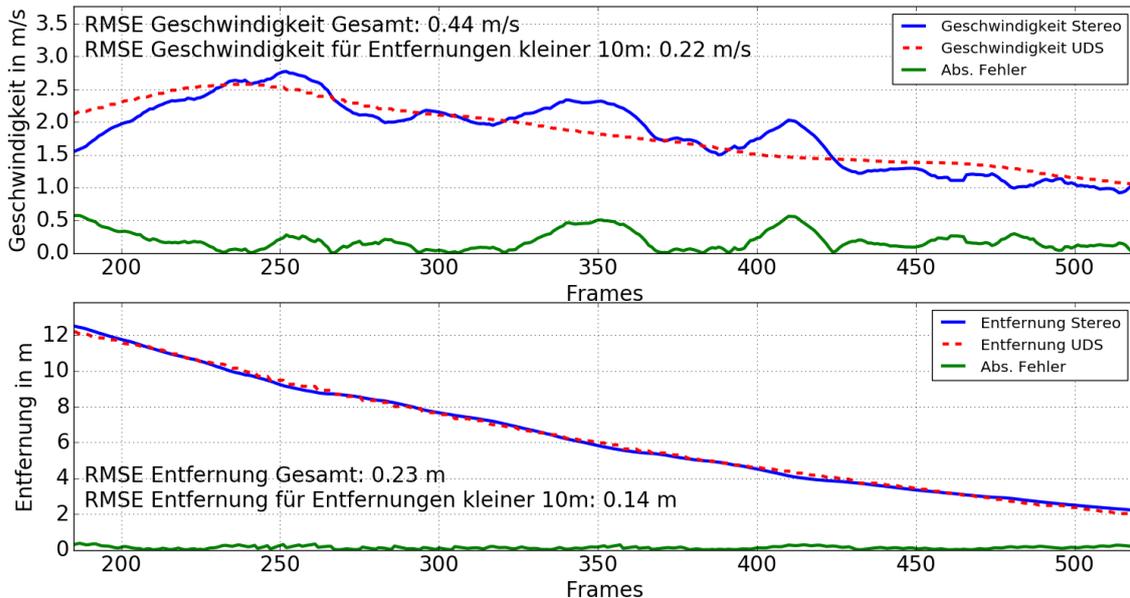
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



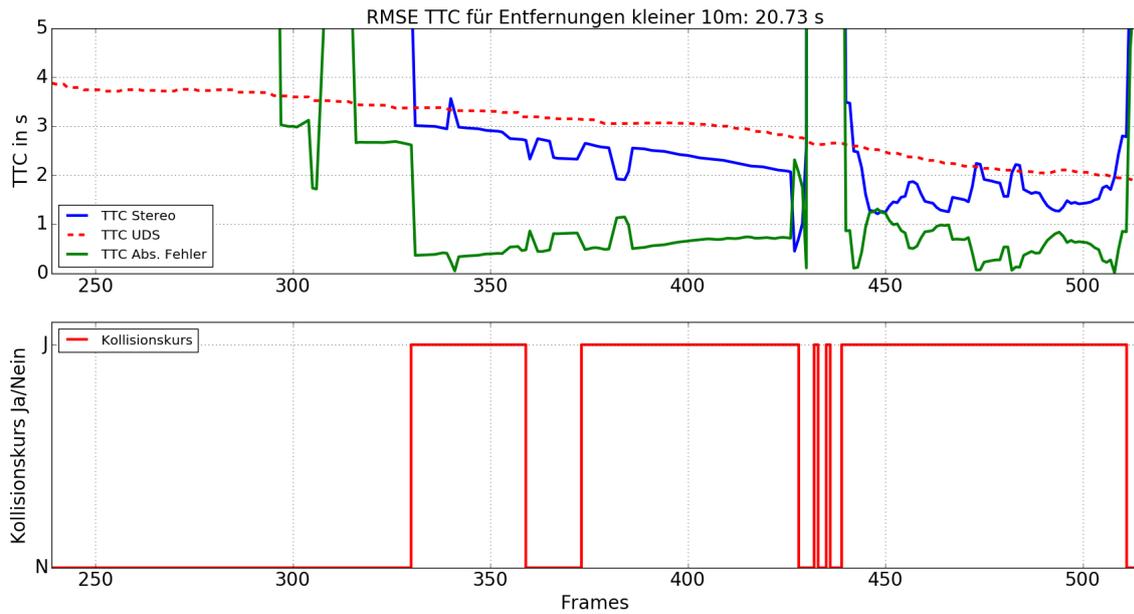
**Abbildung A.117**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den zwanzigsten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.118**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den zwanzigsten Versuch der TTC Validationsversuche im Szenario 1. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

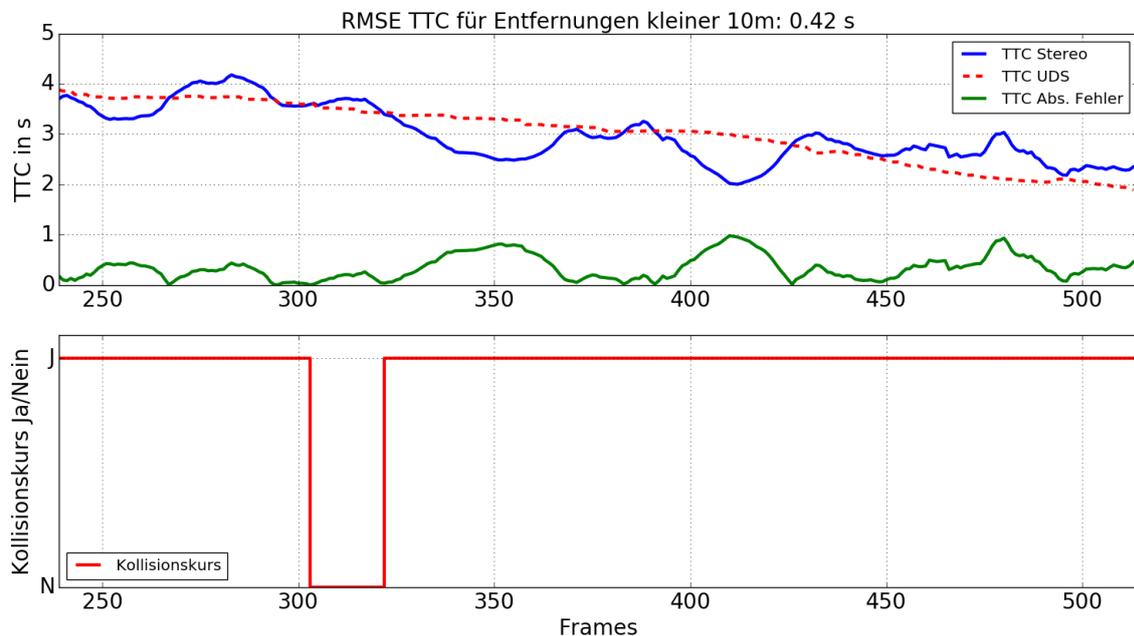


**Abbildung A.119**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den zwanzigsten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



**Abbildung A.120**

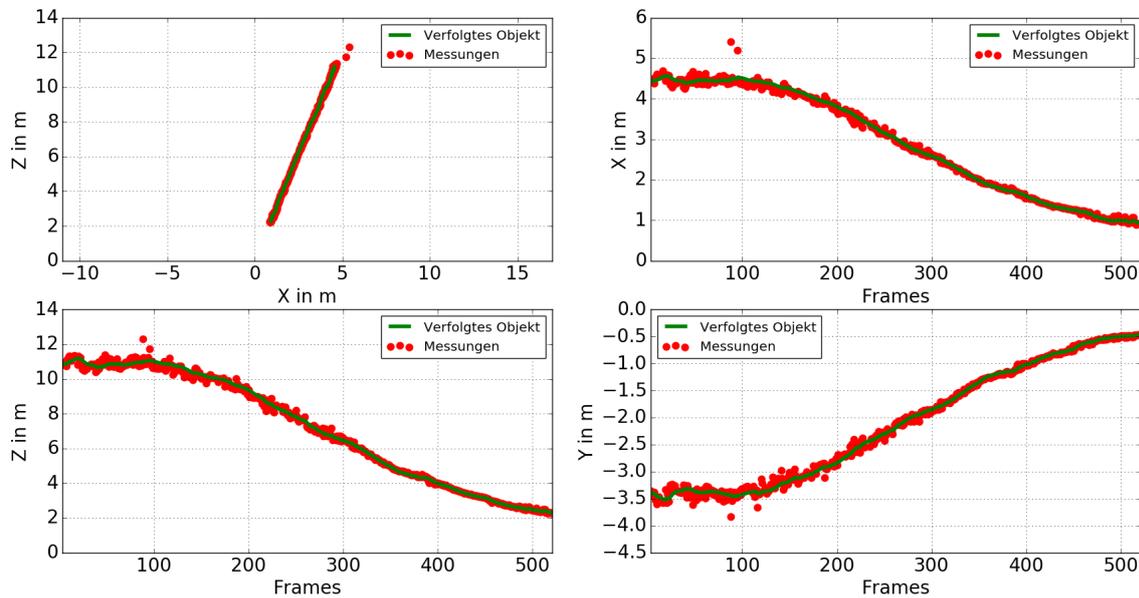
TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den zwanzigsten Versuch der TTC Validationsversuche im Szenario 1. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



## Ergebnisse der Einzelversuche des zweiten Szenarios der TTC Validationsversuche



**Abbildung A.121**

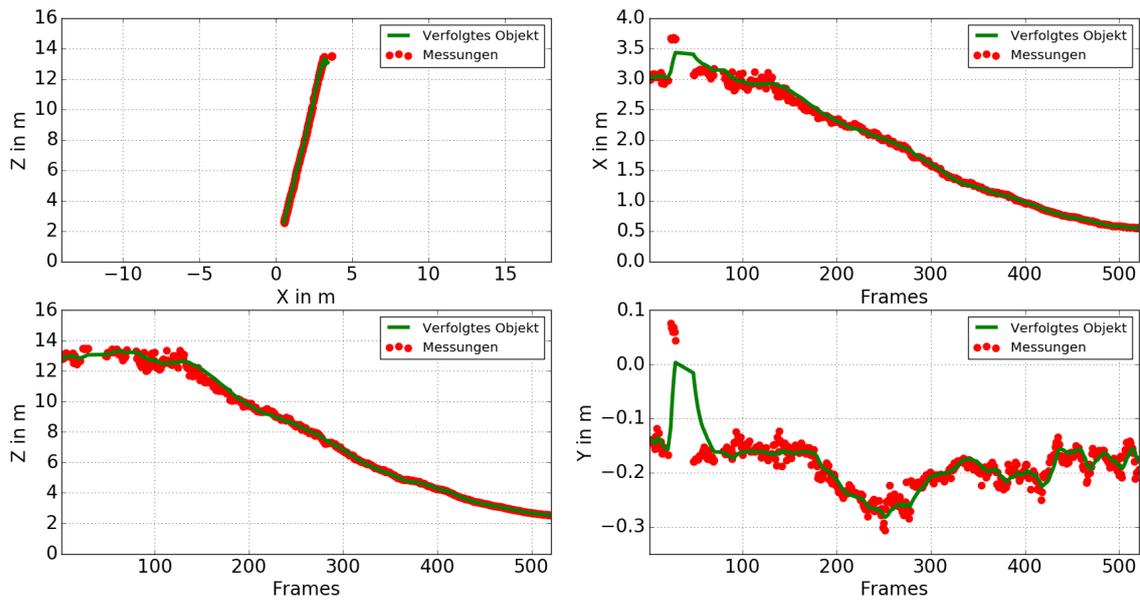
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.122**

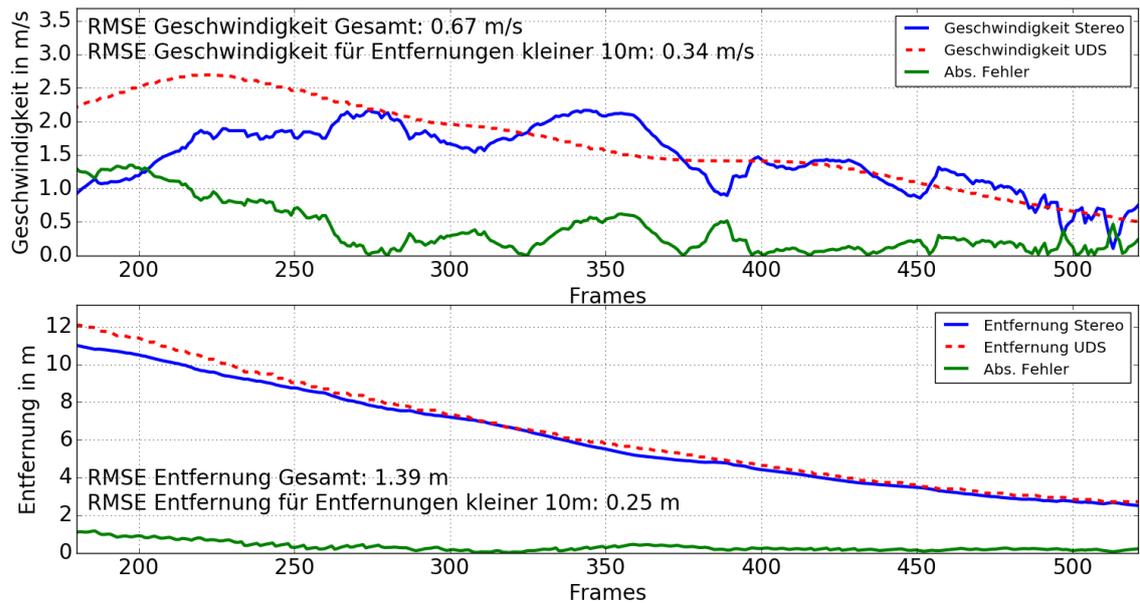
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

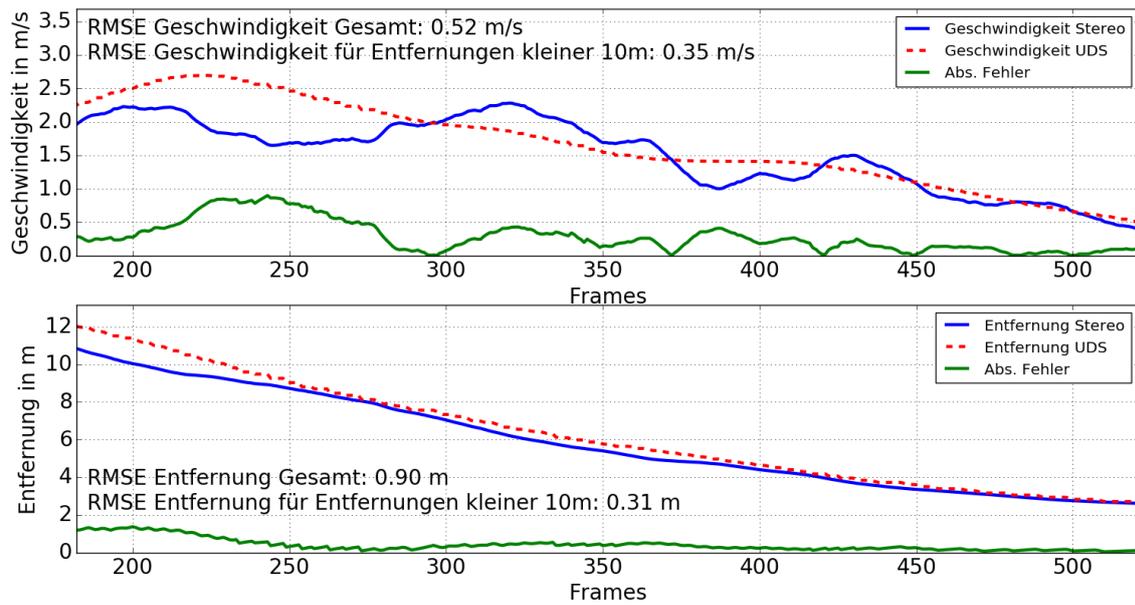
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



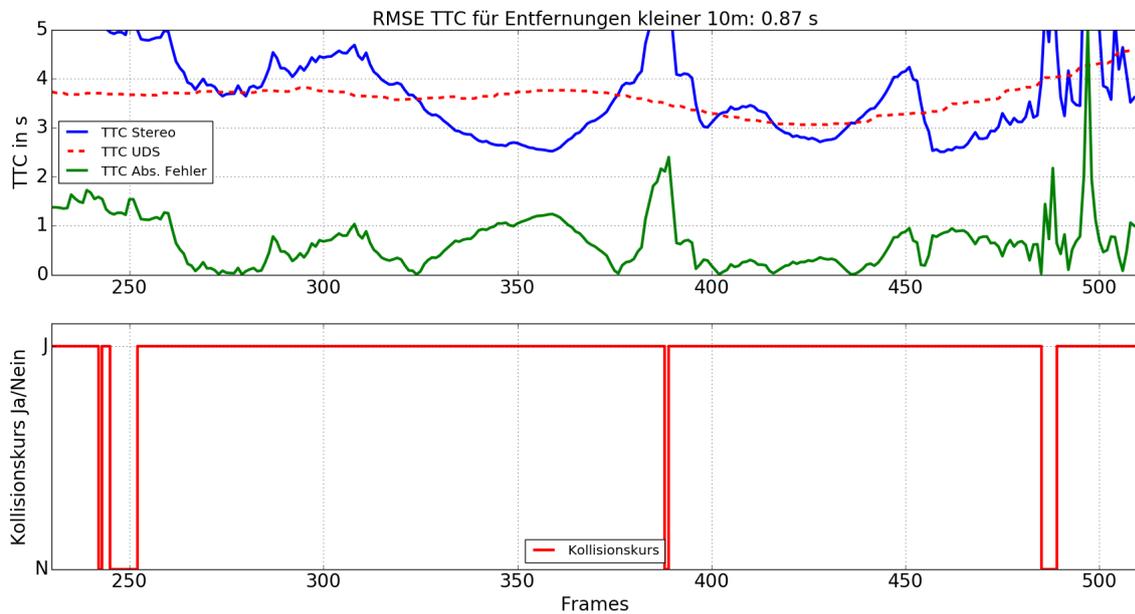
**Abbildung A.123**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.124**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

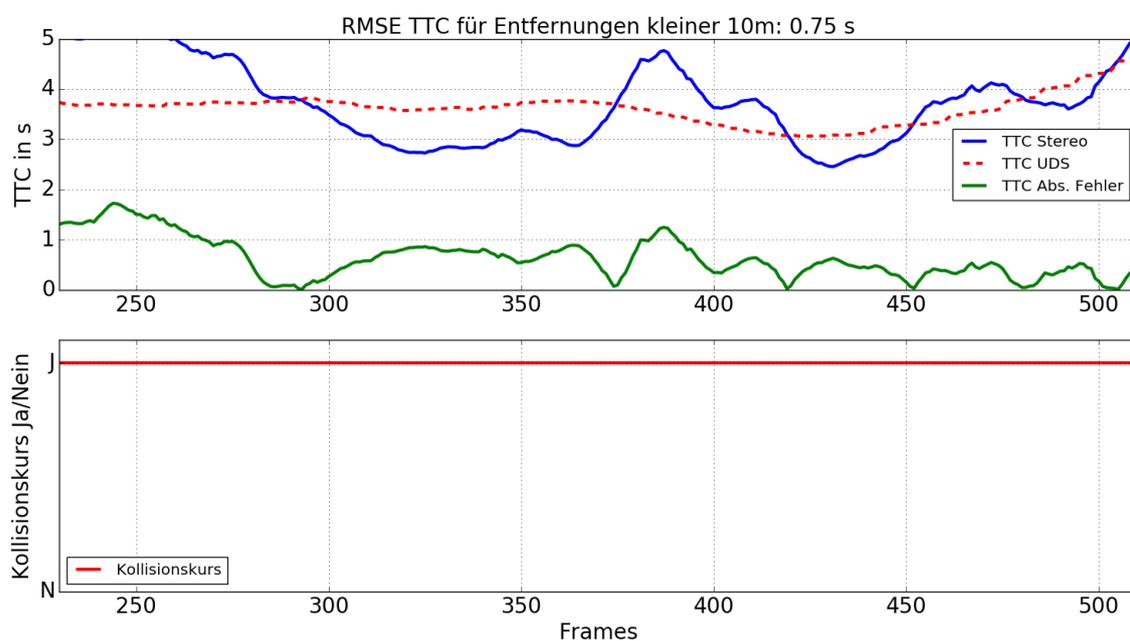


**Abbildung A.125**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

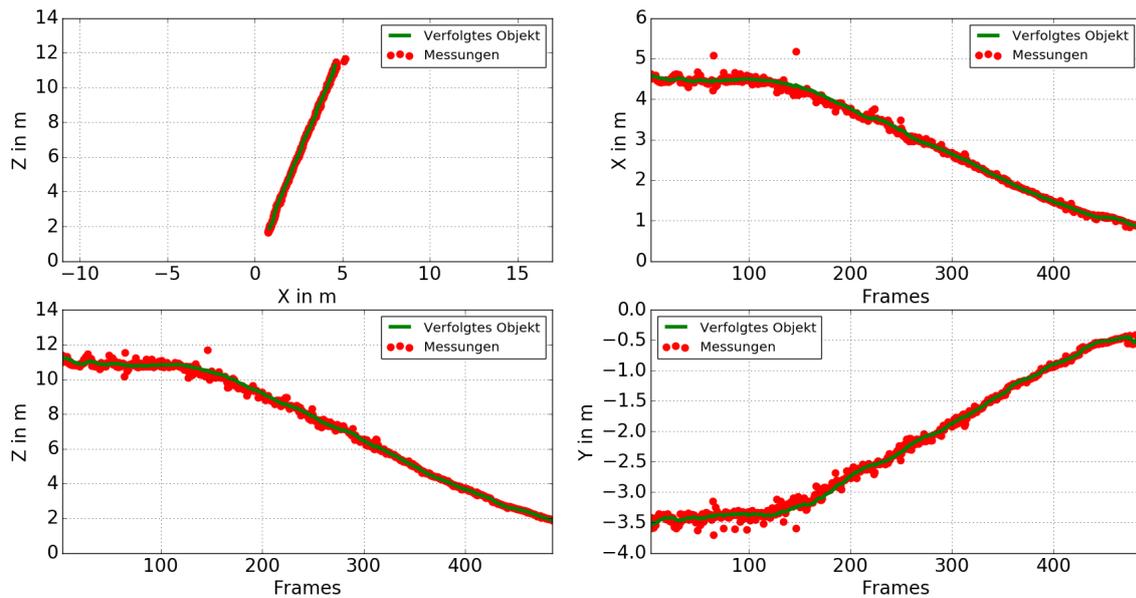


**Abbildung A.126**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den ersten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.127**

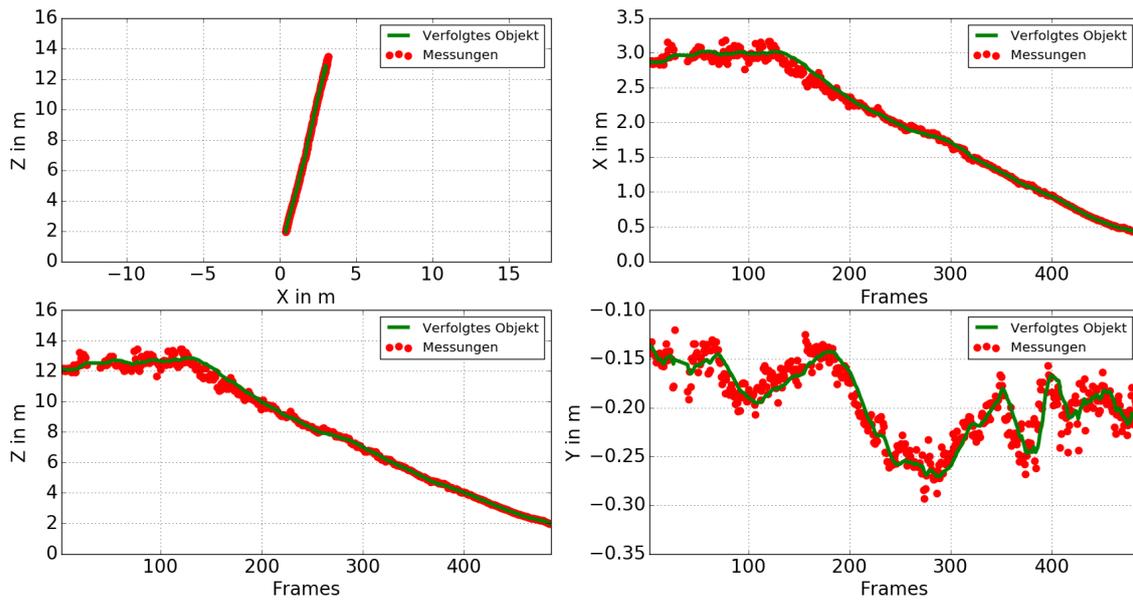
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.128**

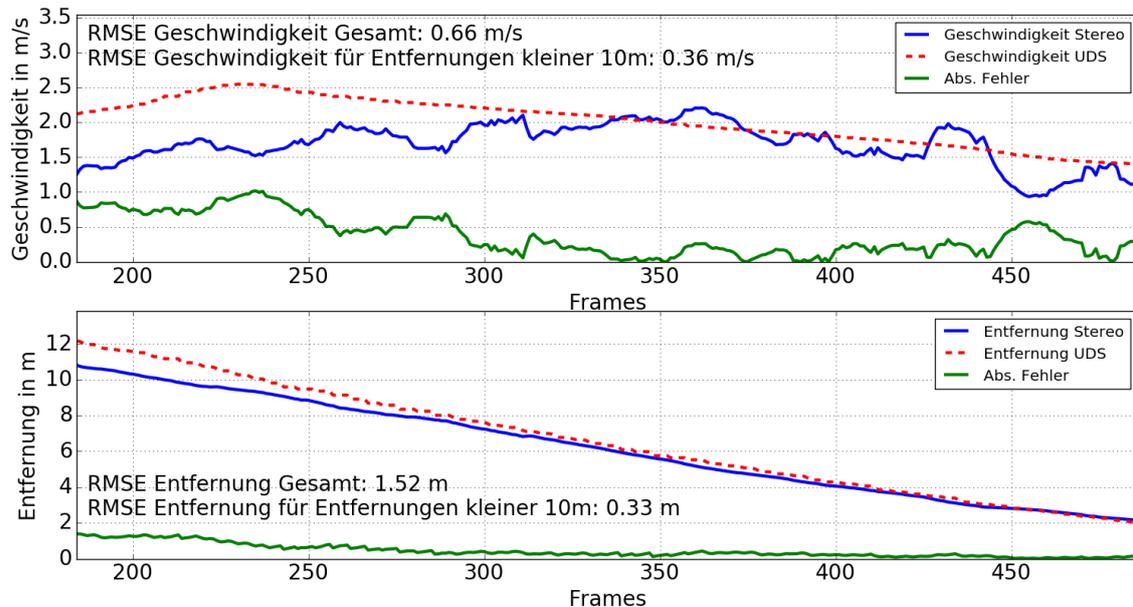
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

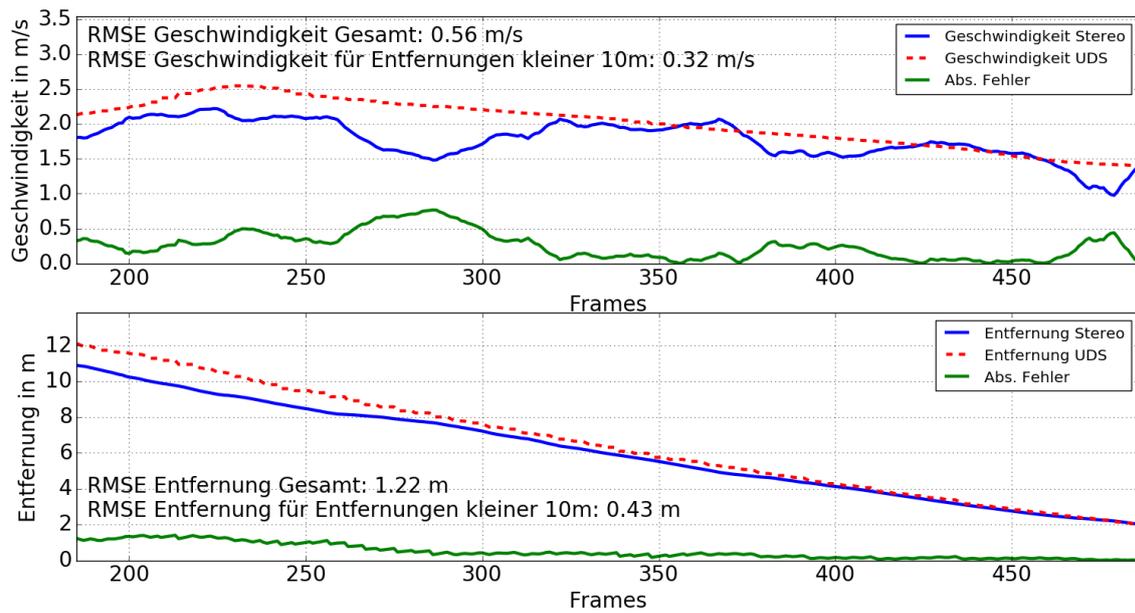
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



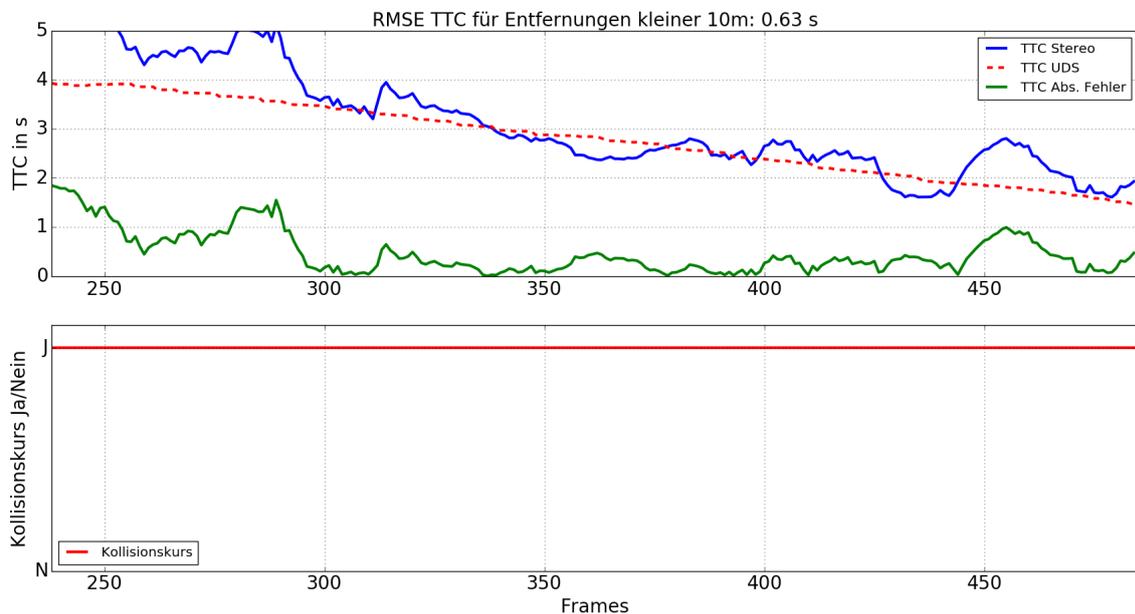
**Abbildung A.129**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.130**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

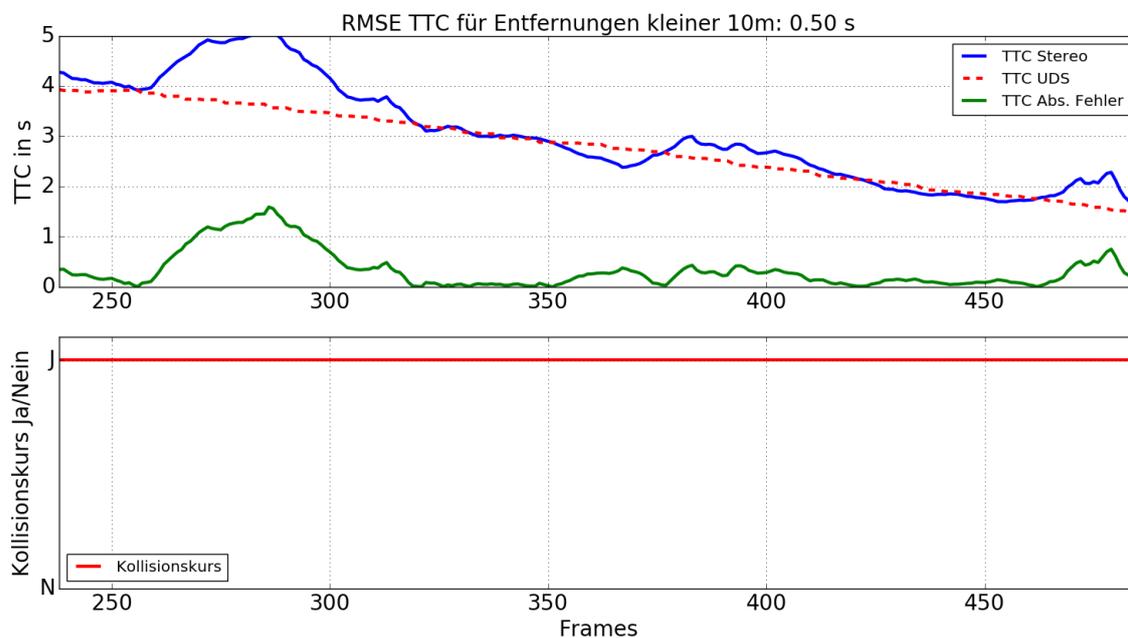


**Abbildung A.131**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

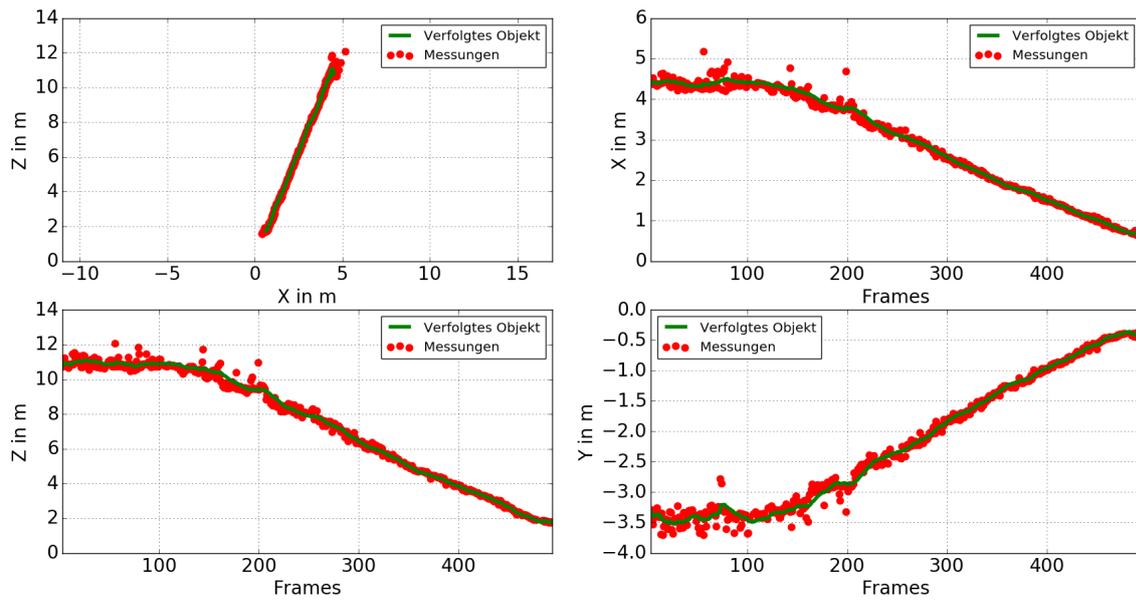


**Abbildung A.132**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den zweiten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.133**

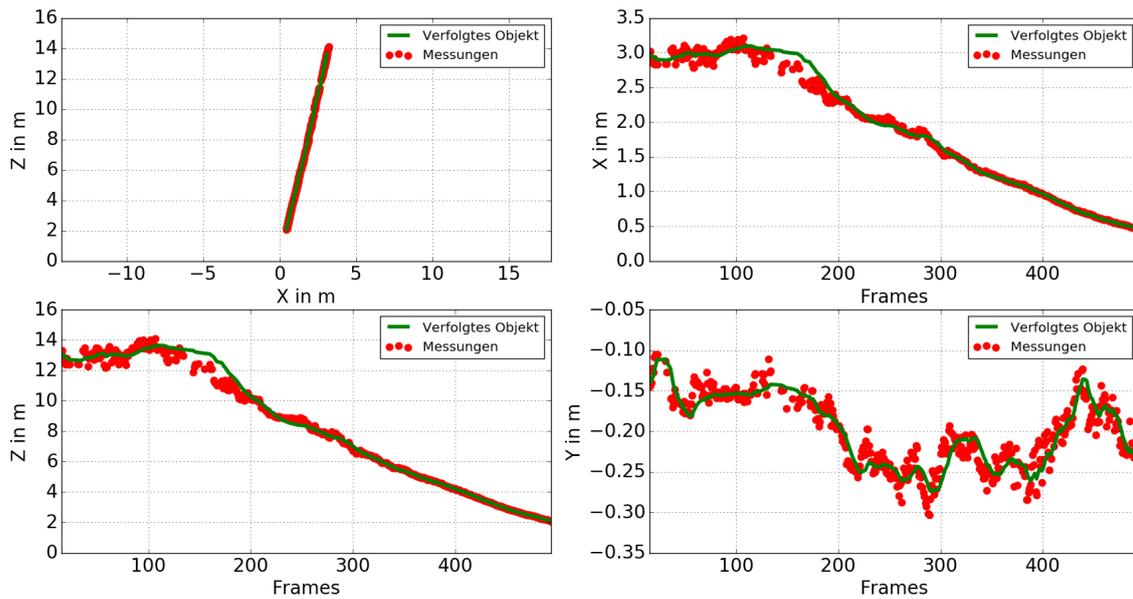
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.134**

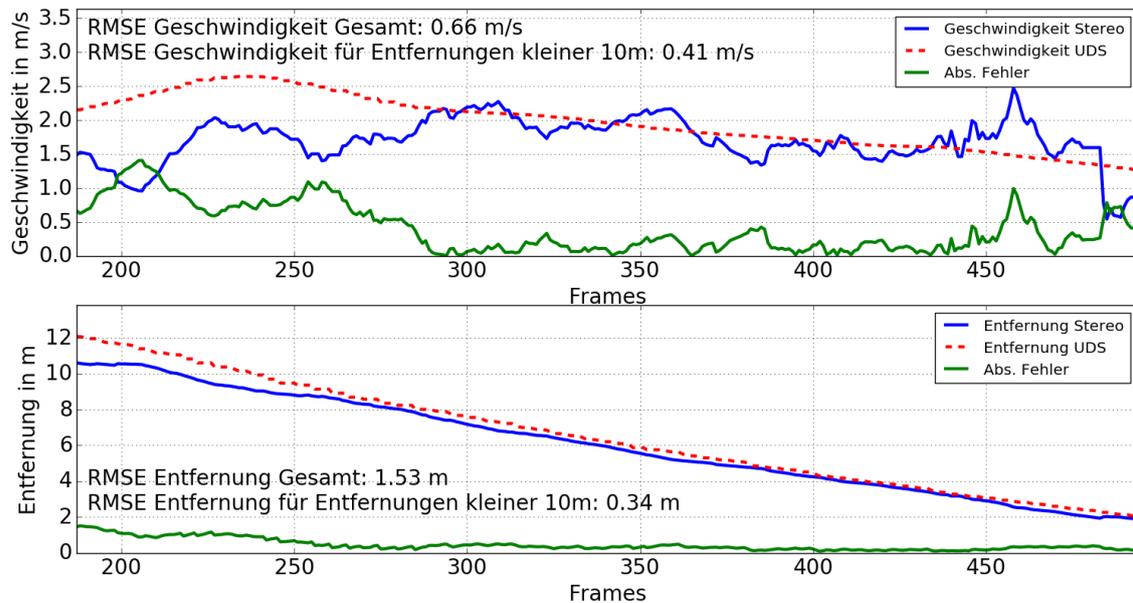
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

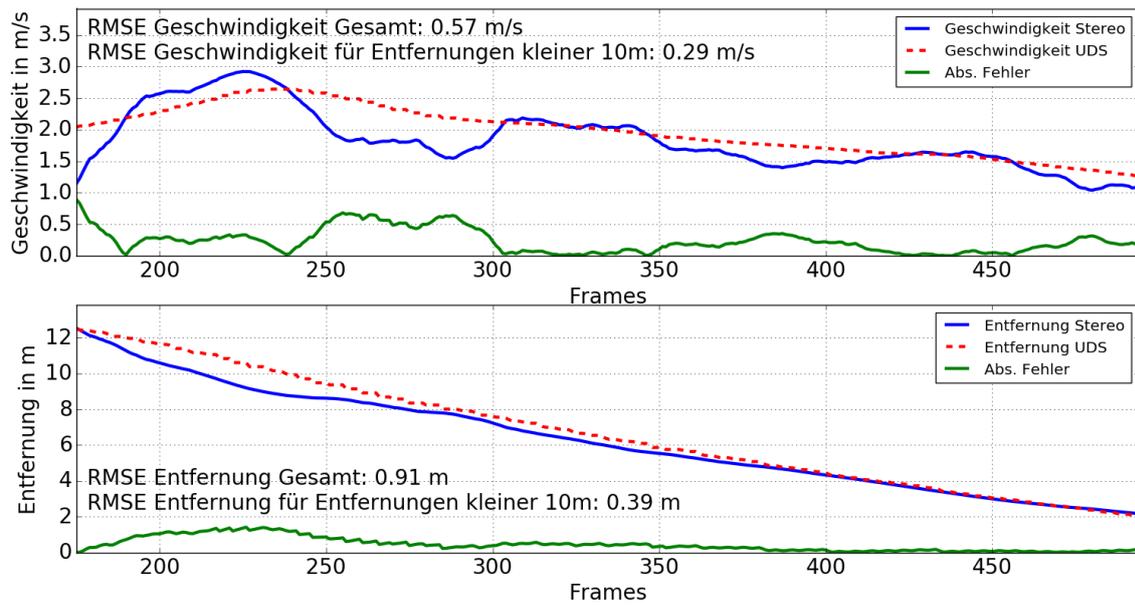
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



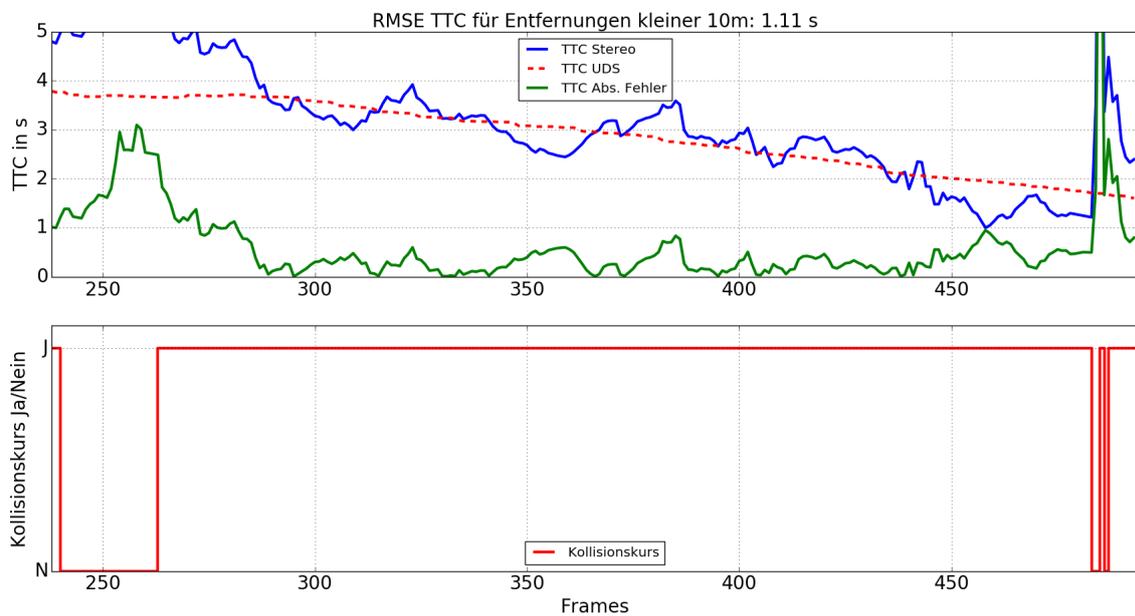
**Abbildung A.135**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.136**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

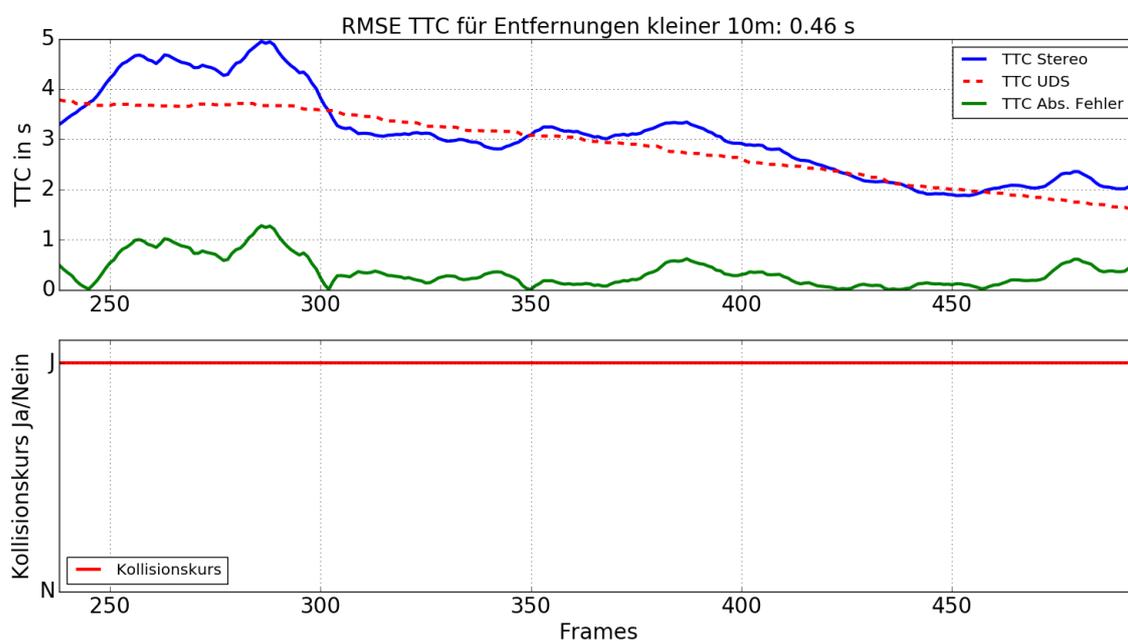


**Abbildung A.137**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

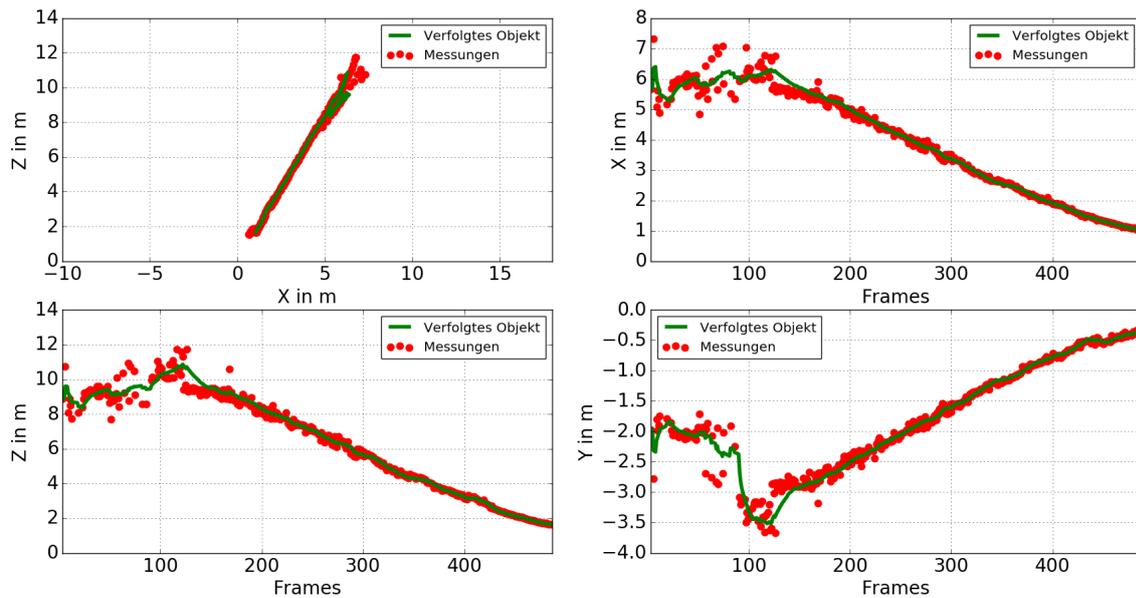


**Abbildung A.138**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den dritten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.139**

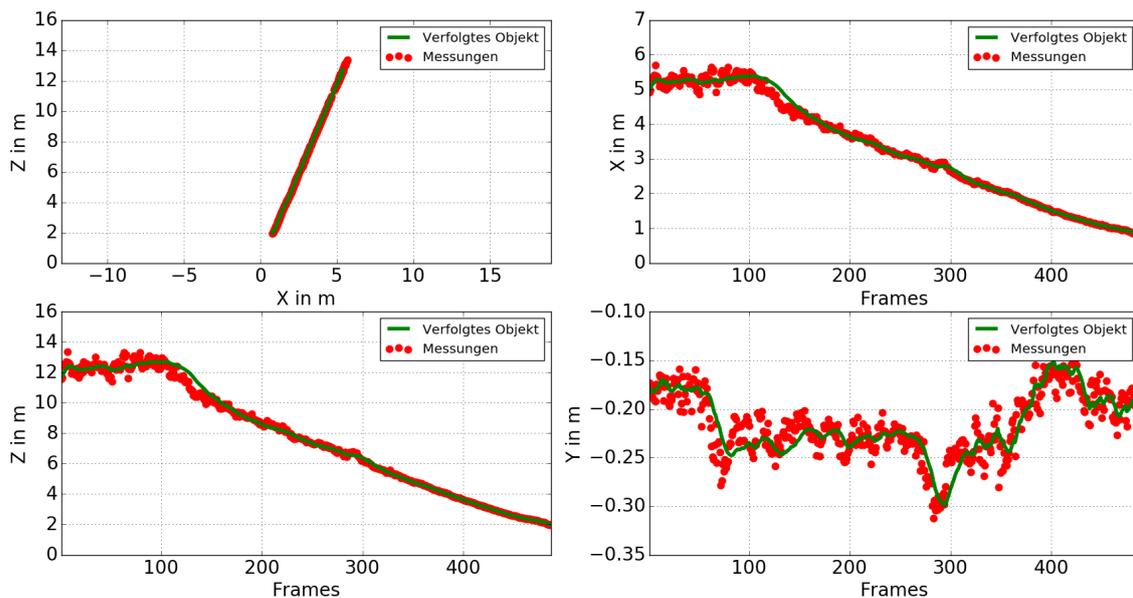
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.140**

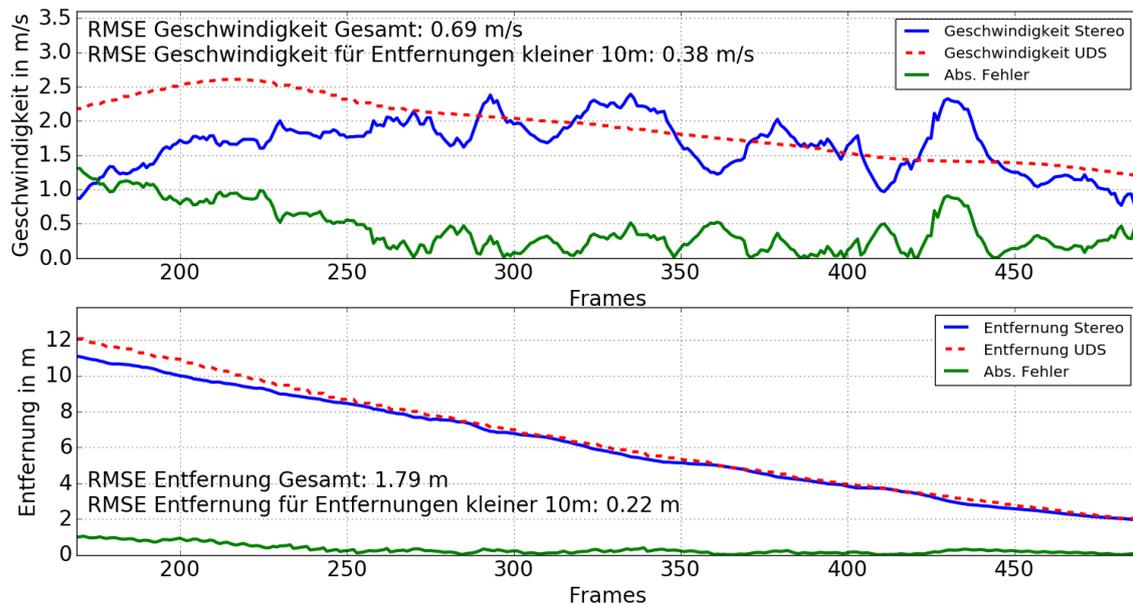
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

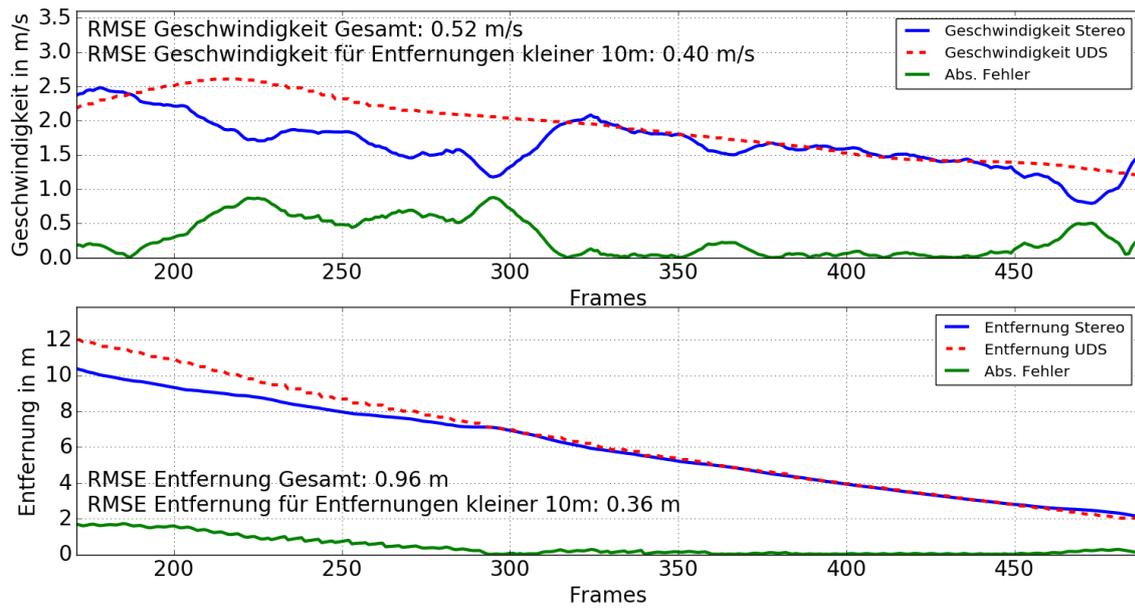
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



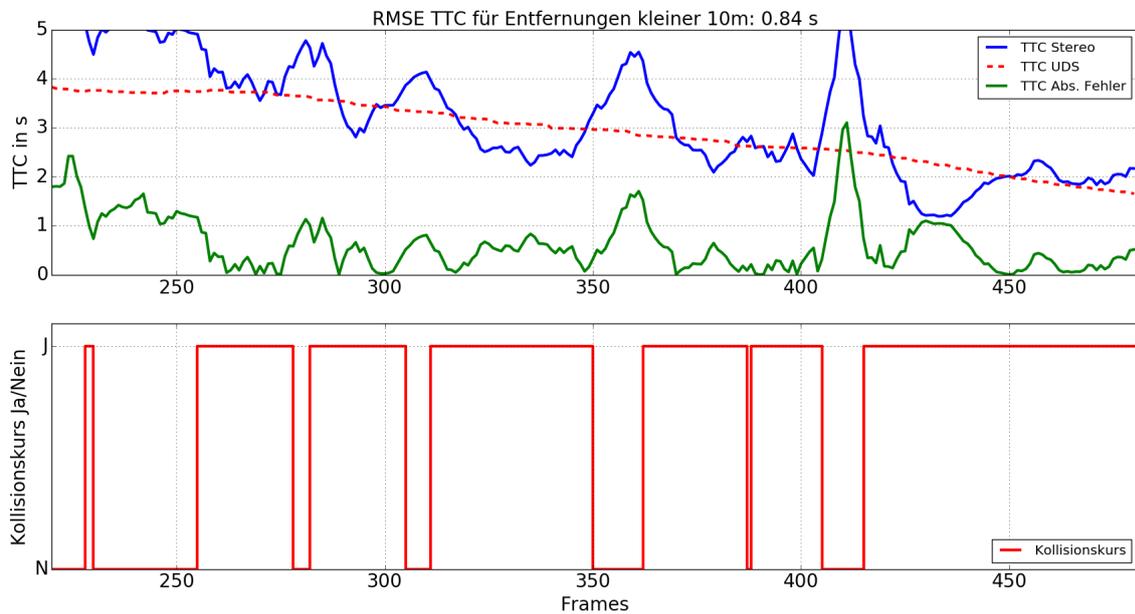
**Abbildung A.141**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.142**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

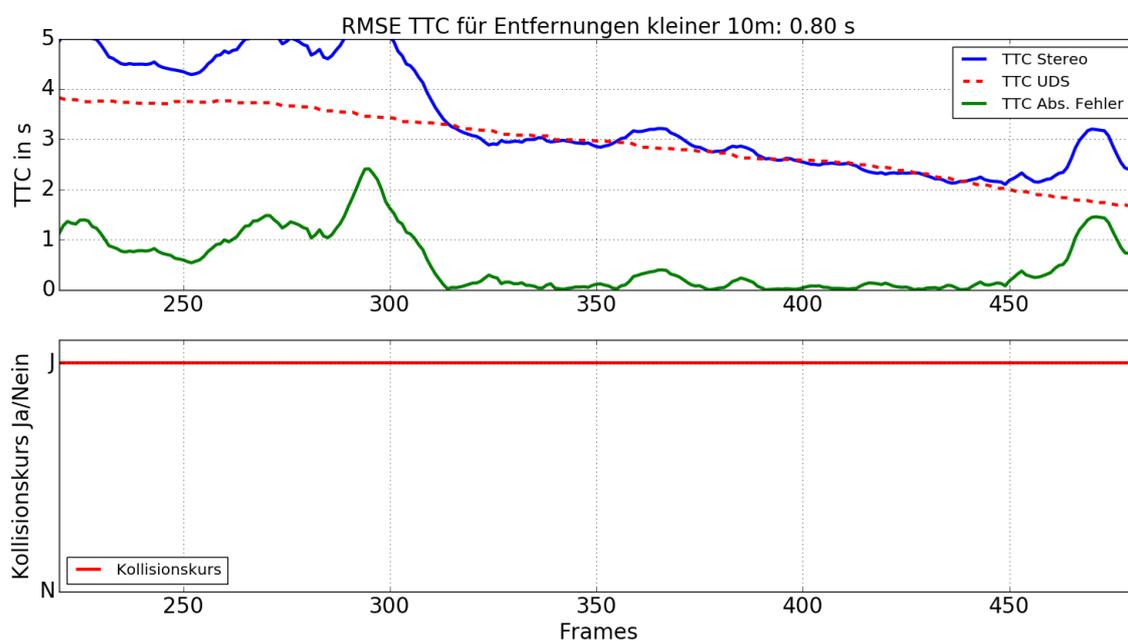


**Abbildung A.143**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

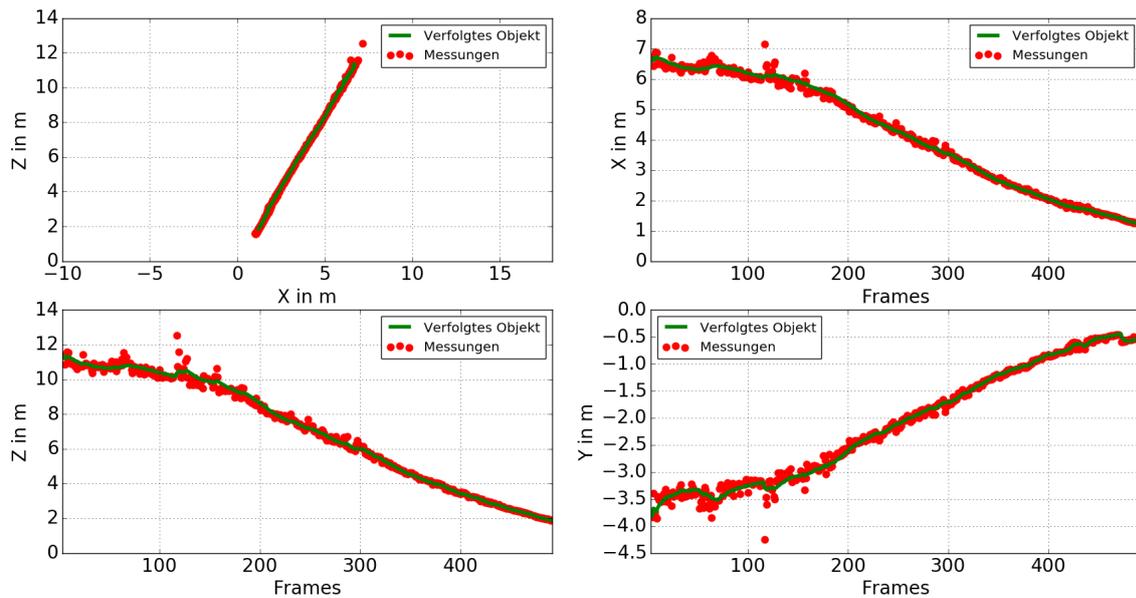


**Abbildung A.144**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den vierten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.145**

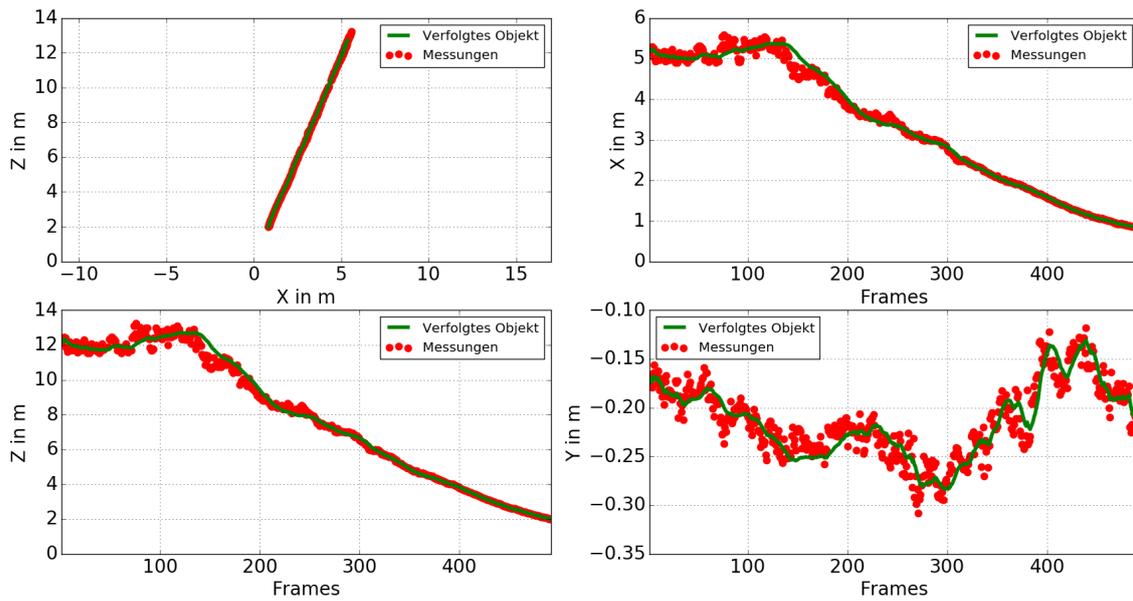
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.146**

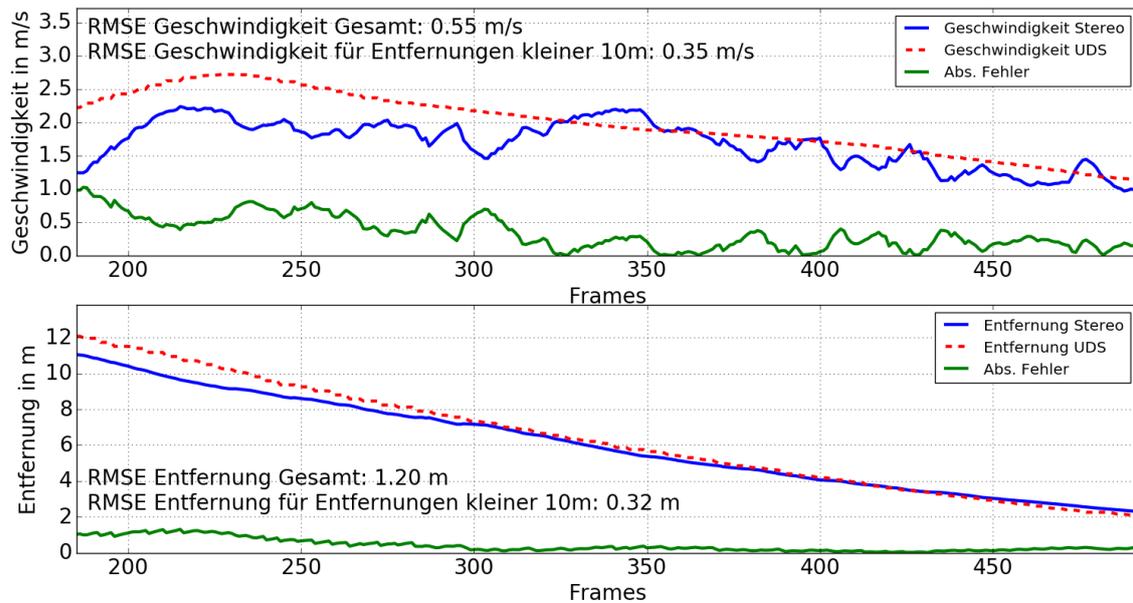
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

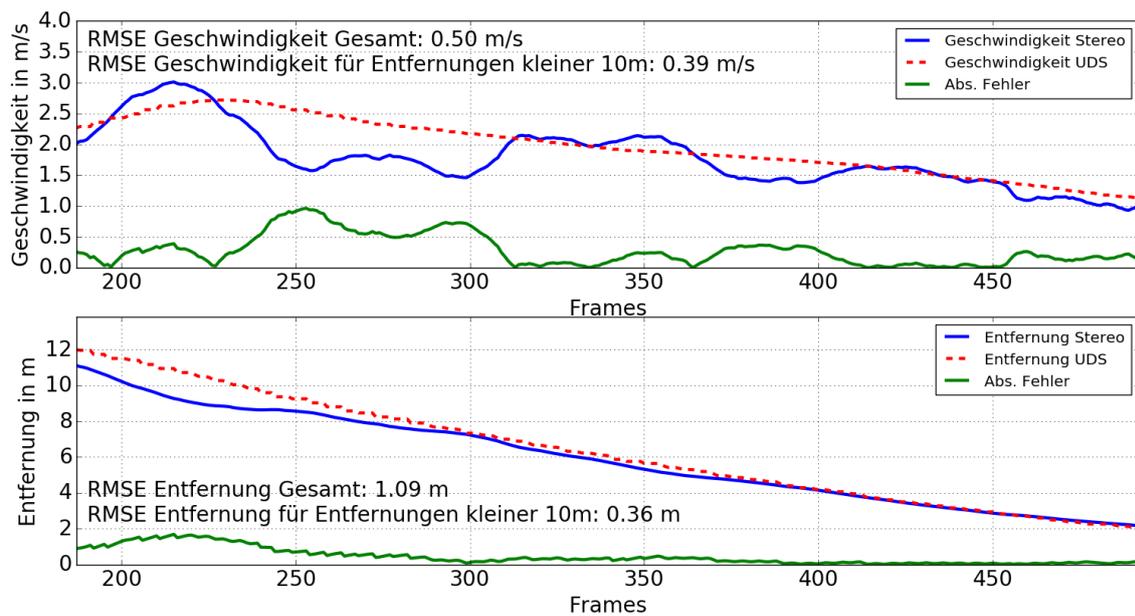
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



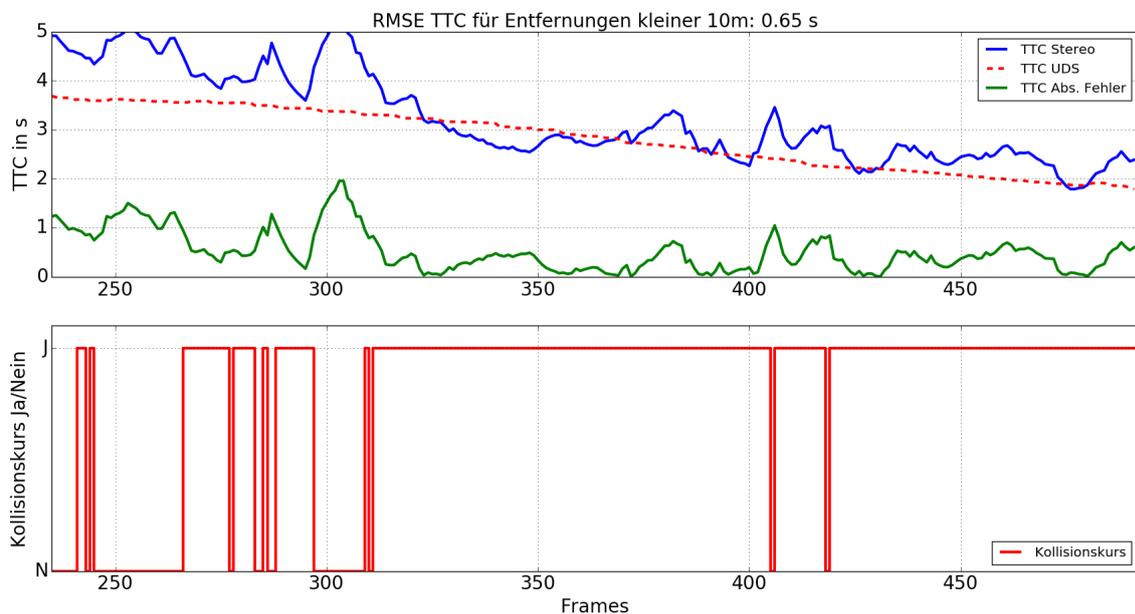
**Abbildung A.147**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.148**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

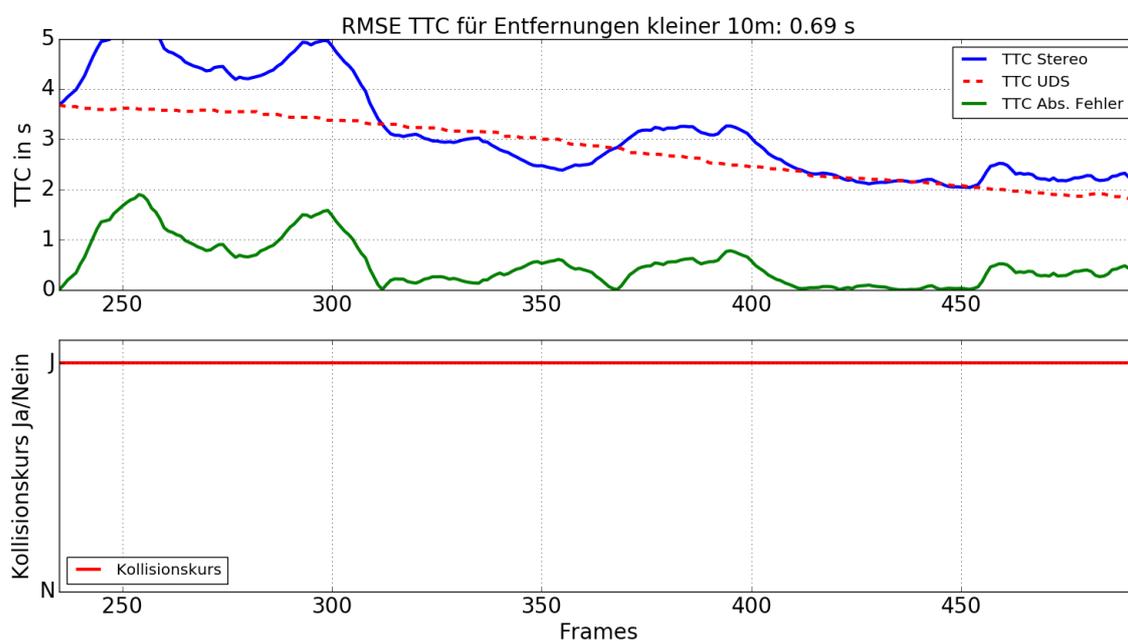


**Abbildung A.149**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

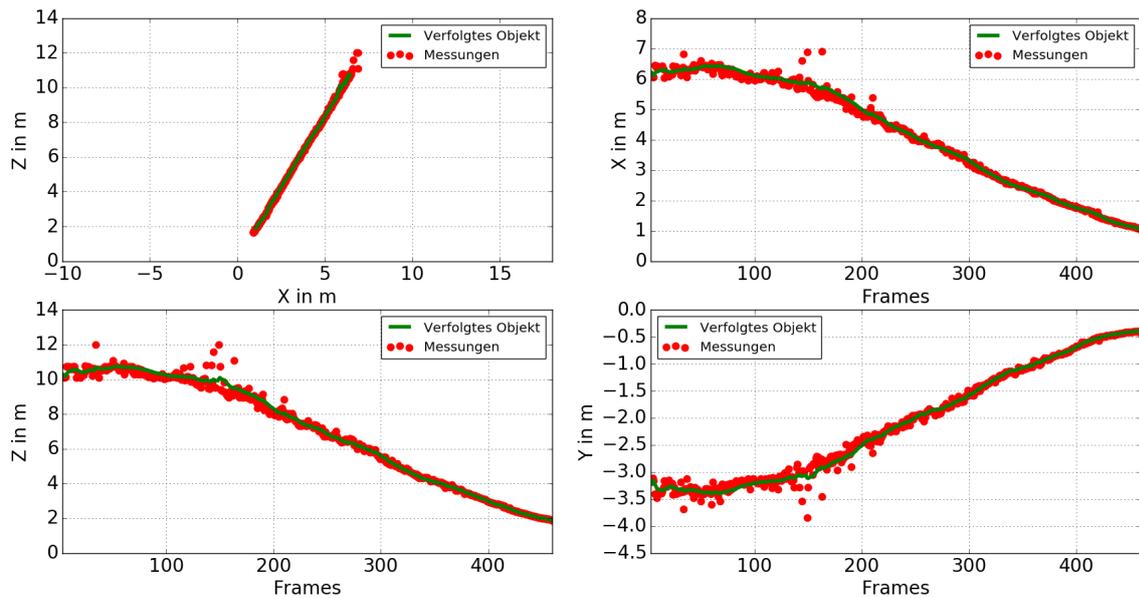


**Abbildung A.150**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den fünften Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.151**

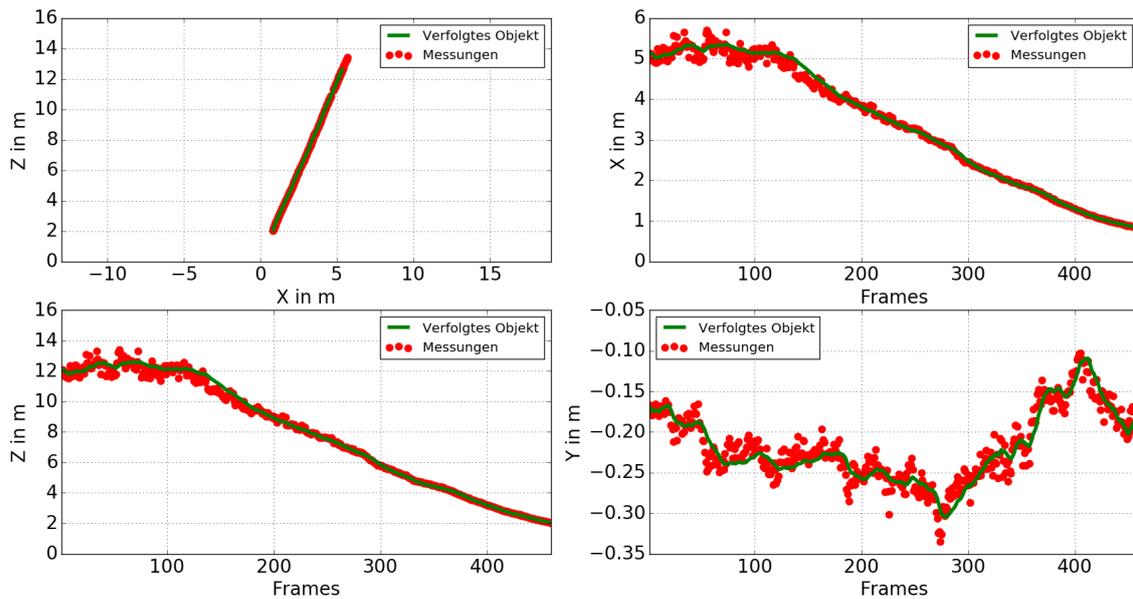
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.152**

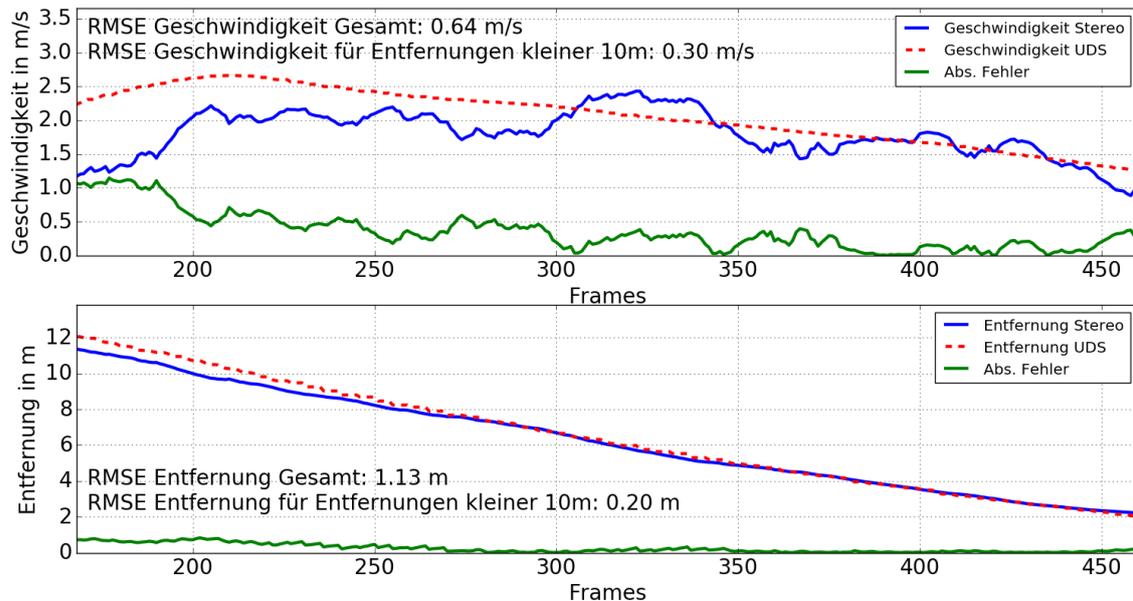
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 2. Es liegt kein *Offset* vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

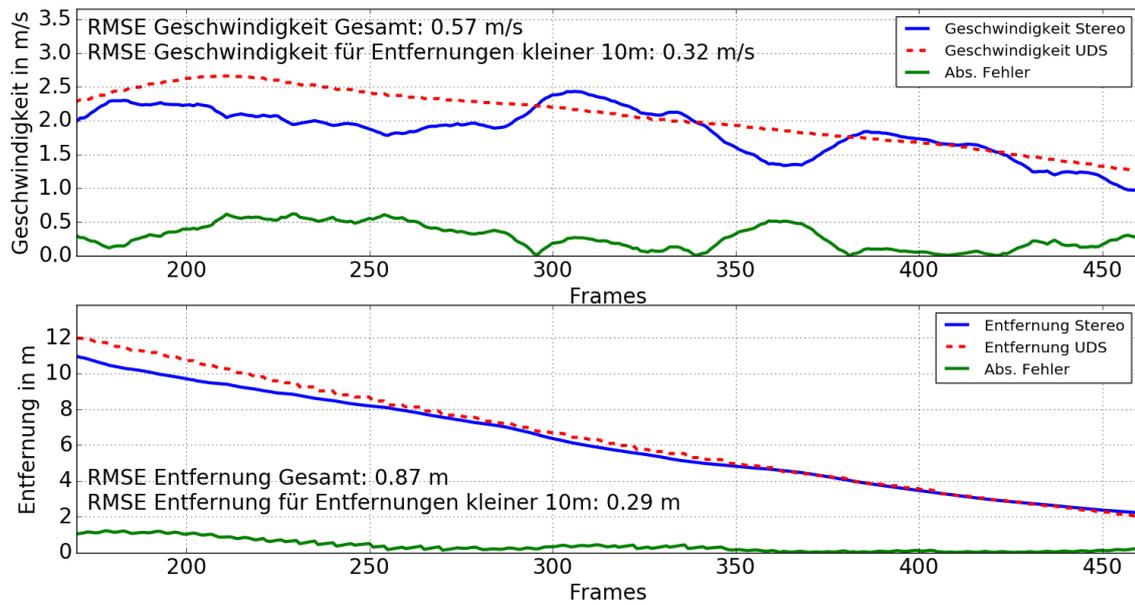
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



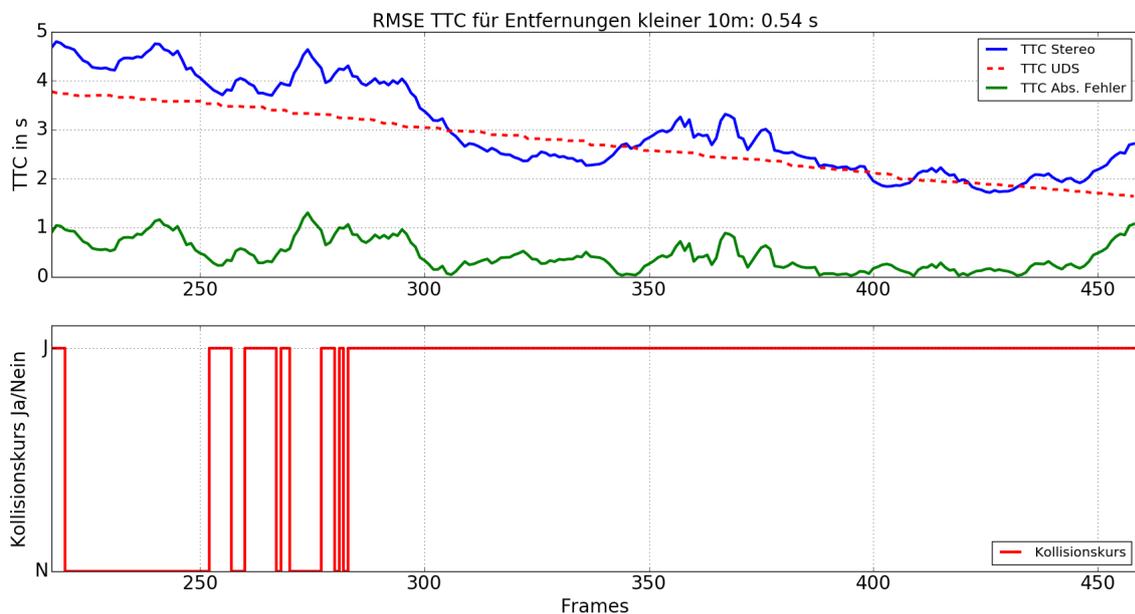
**Abbildung A.153**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.154**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

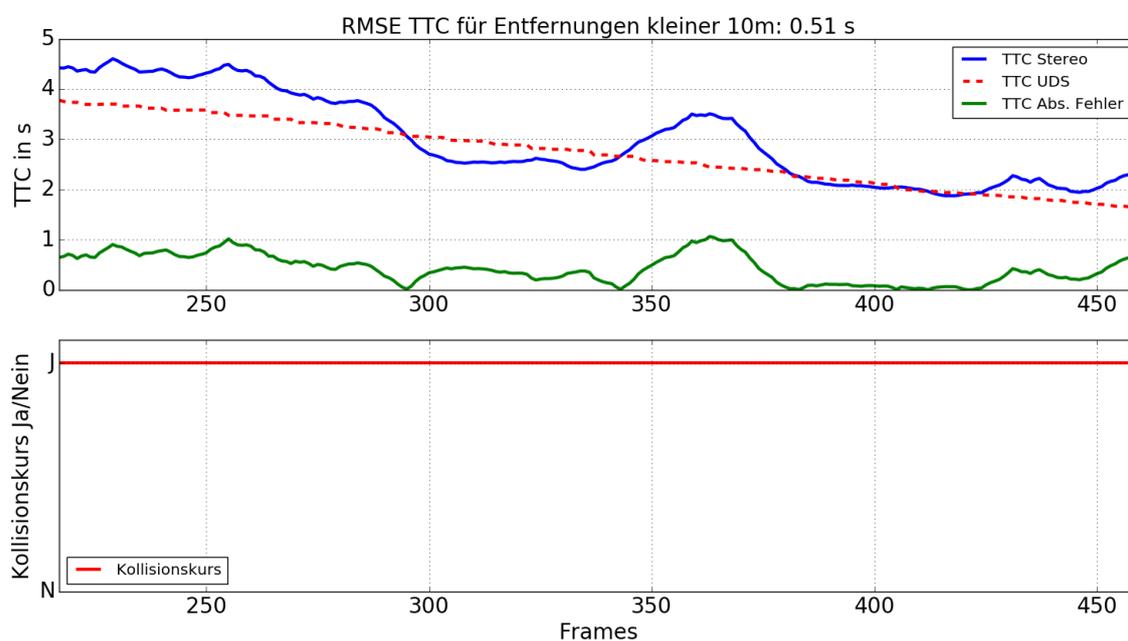


**Abbildung A.155**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

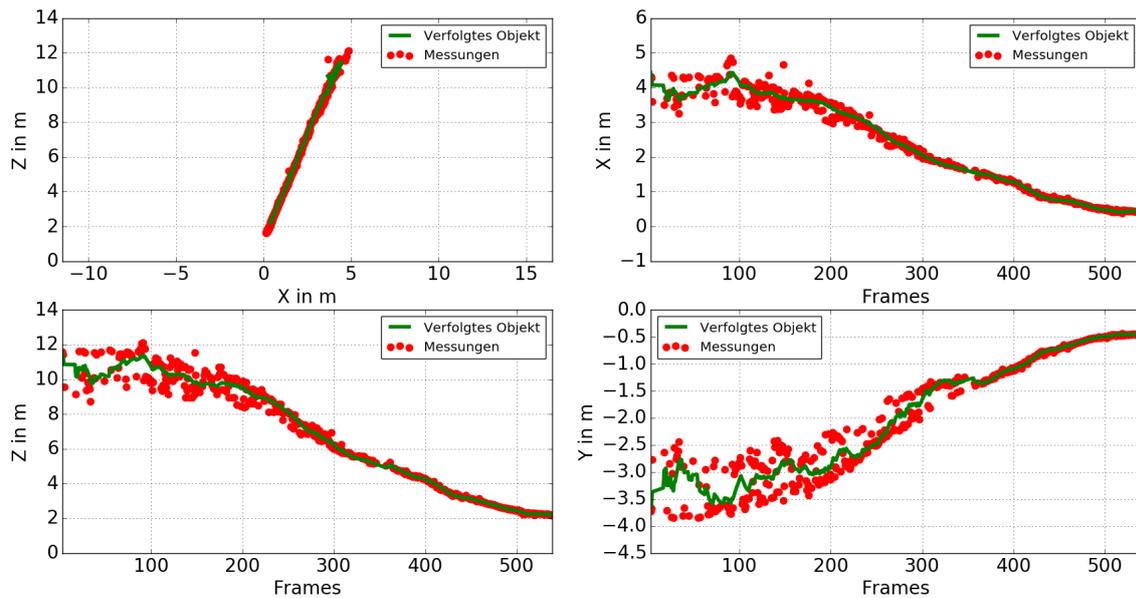


**Abbildung A.156**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den sechsten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.157**

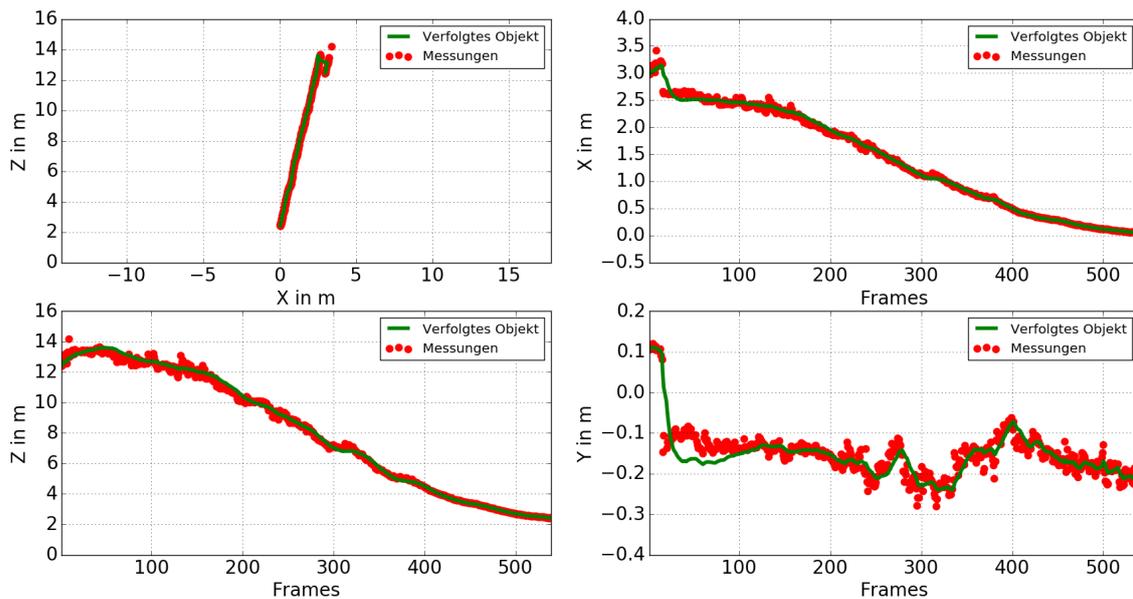
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.158**

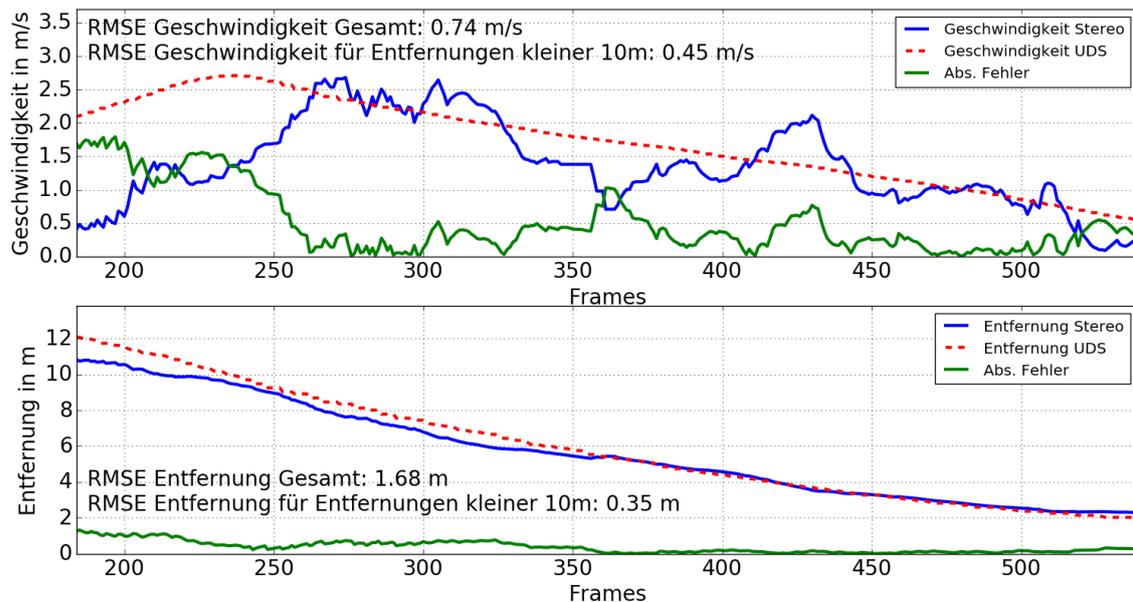
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

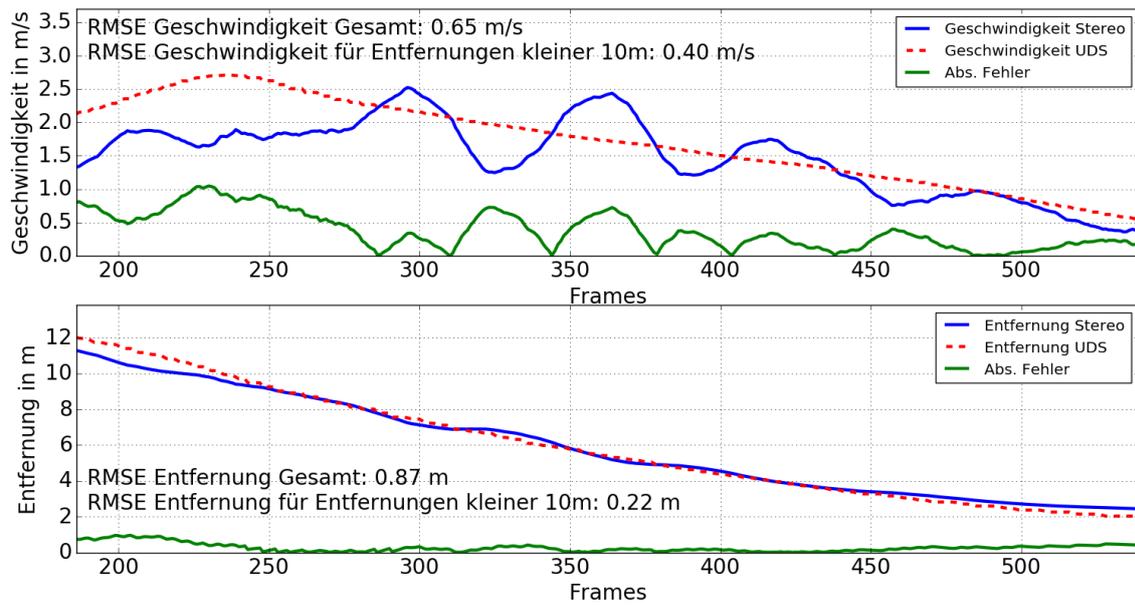
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



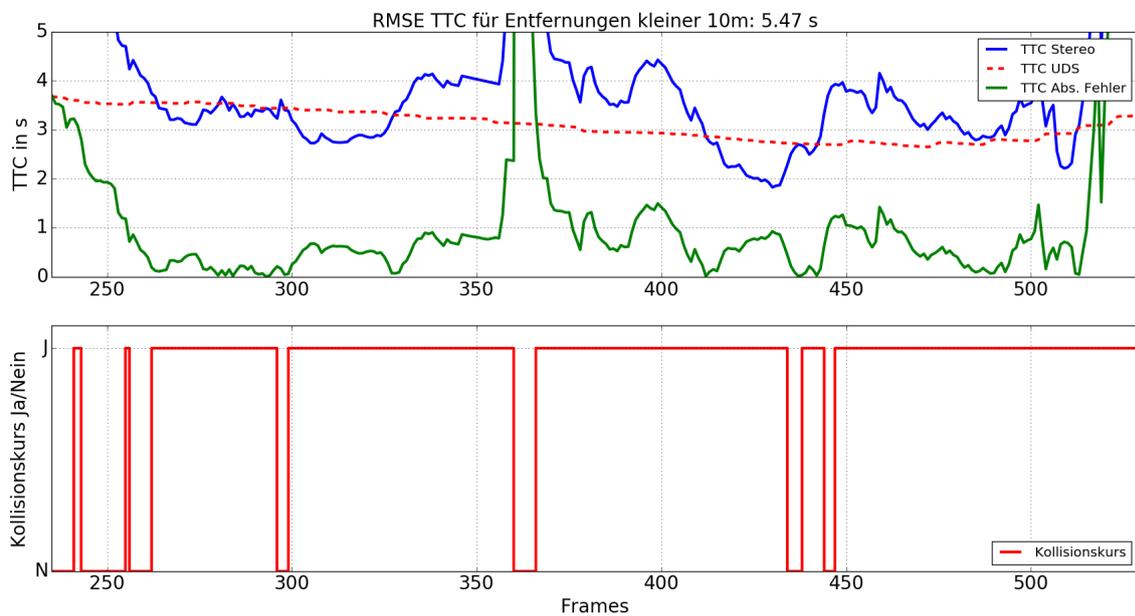
**Abbildung A.159**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.160**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

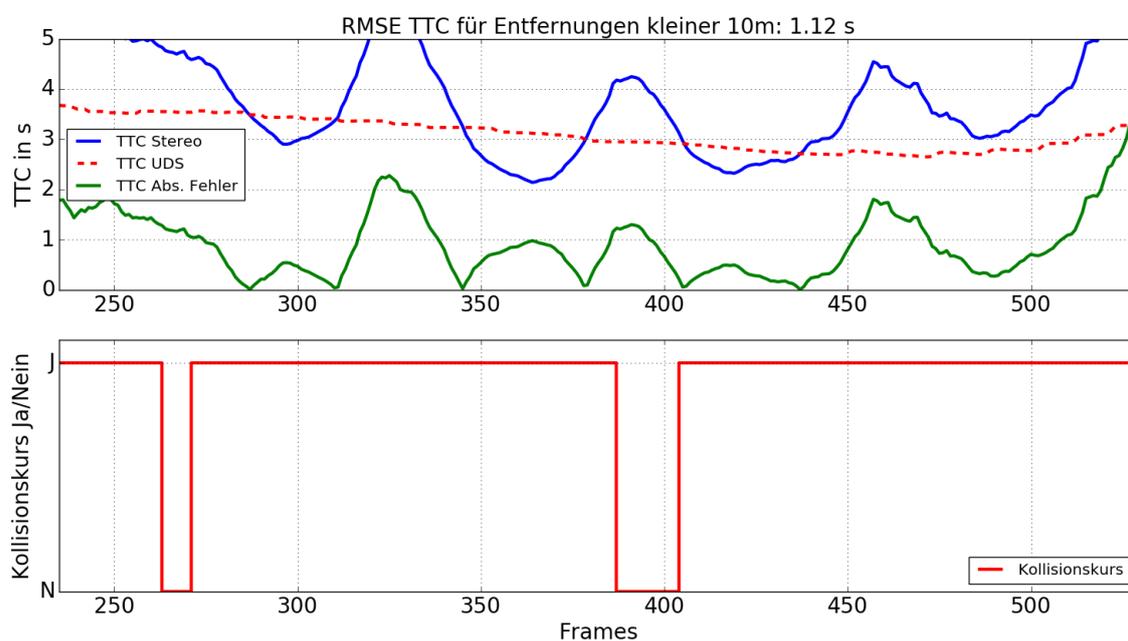


**Abbildung A.161**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

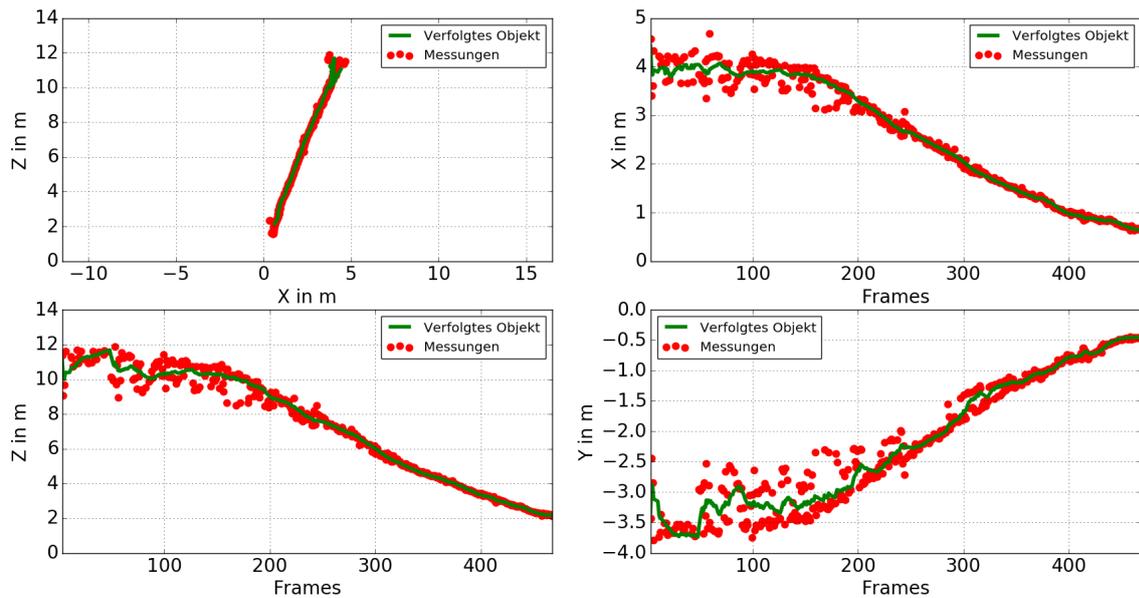


**Abbildung A.162**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den siebten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.163**

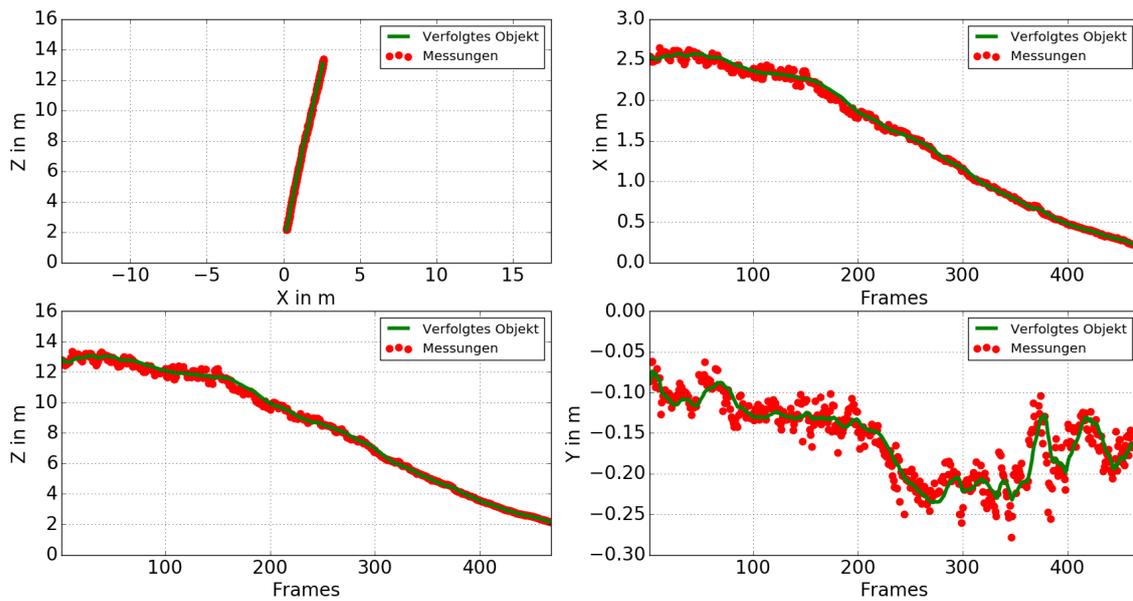
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.164**

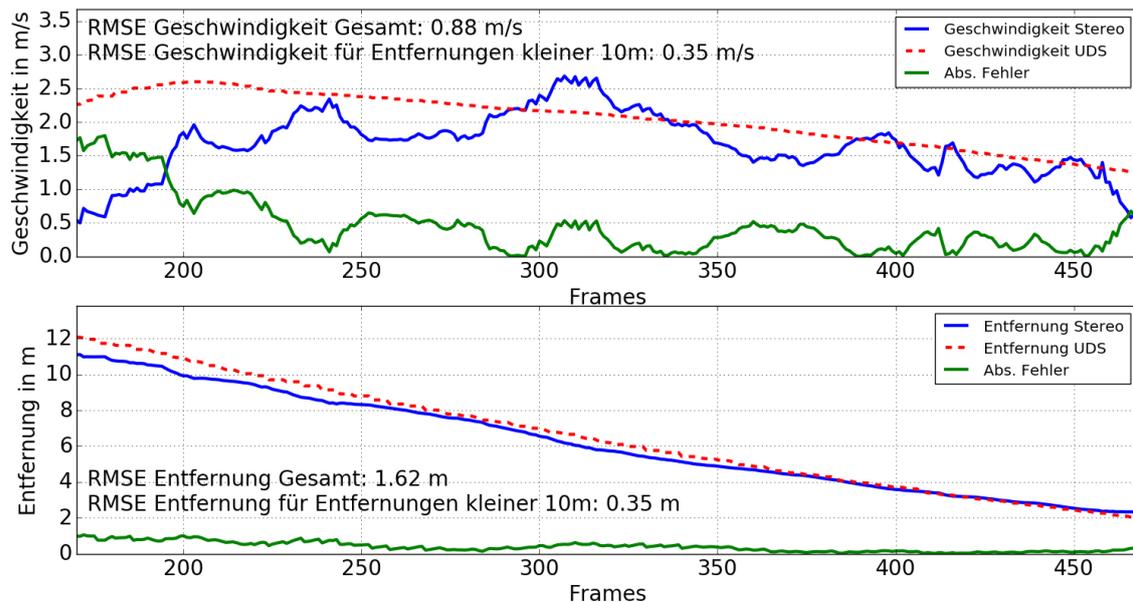
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

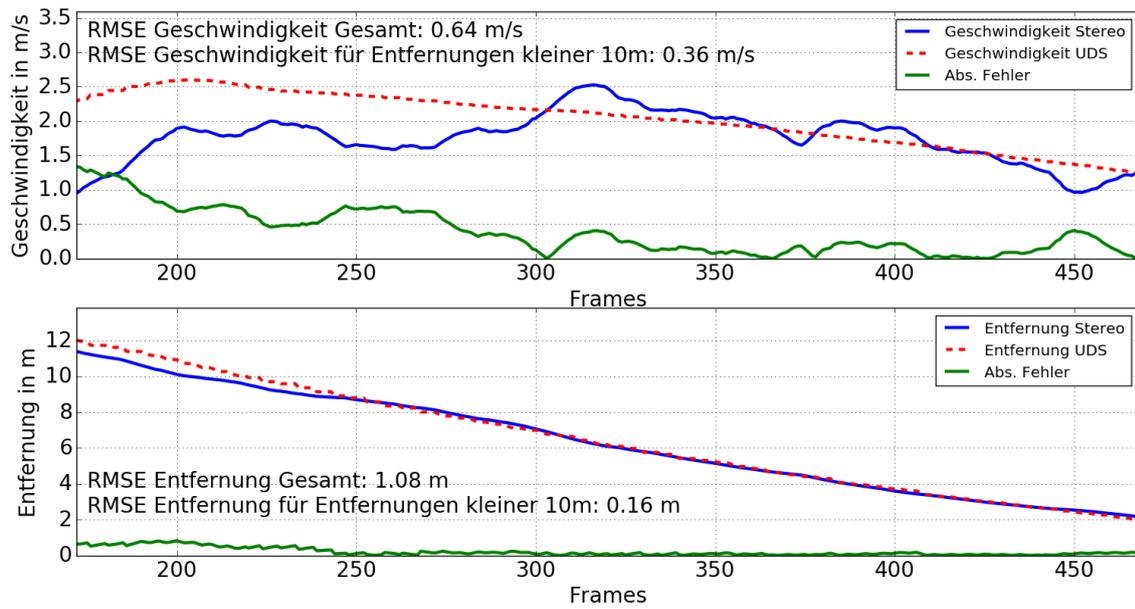
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



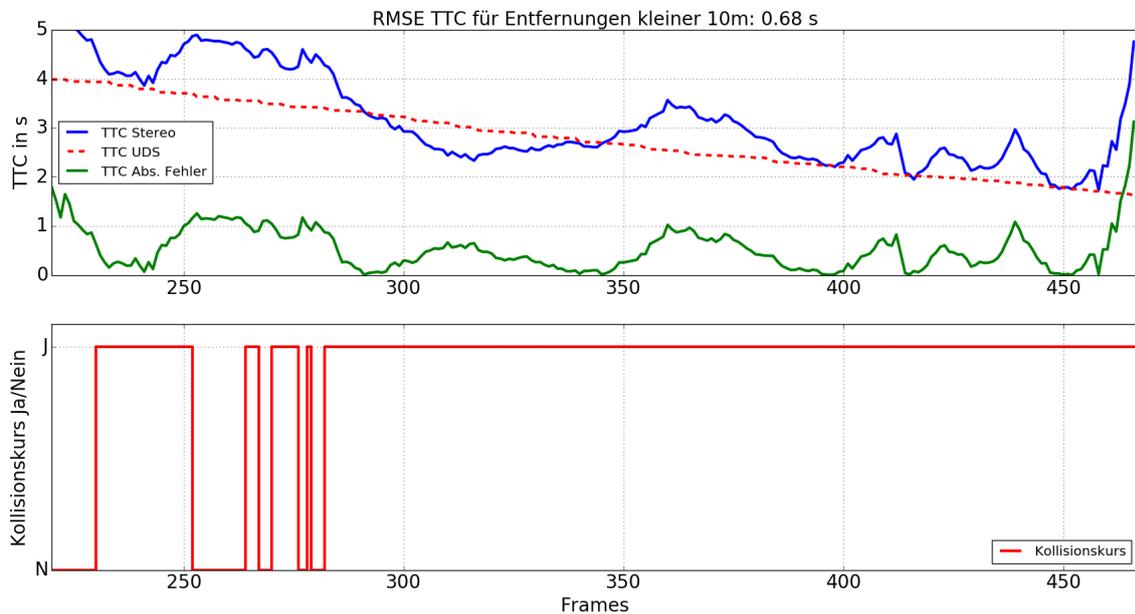
**Abbildung A.165**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.166**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

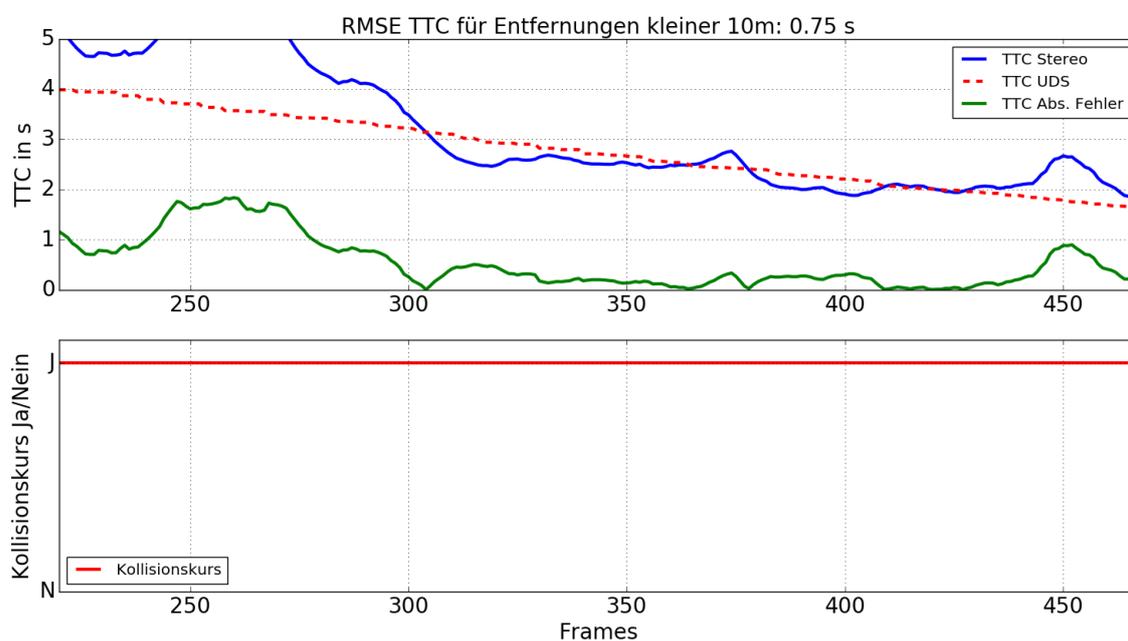


**Abbildung A.167**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

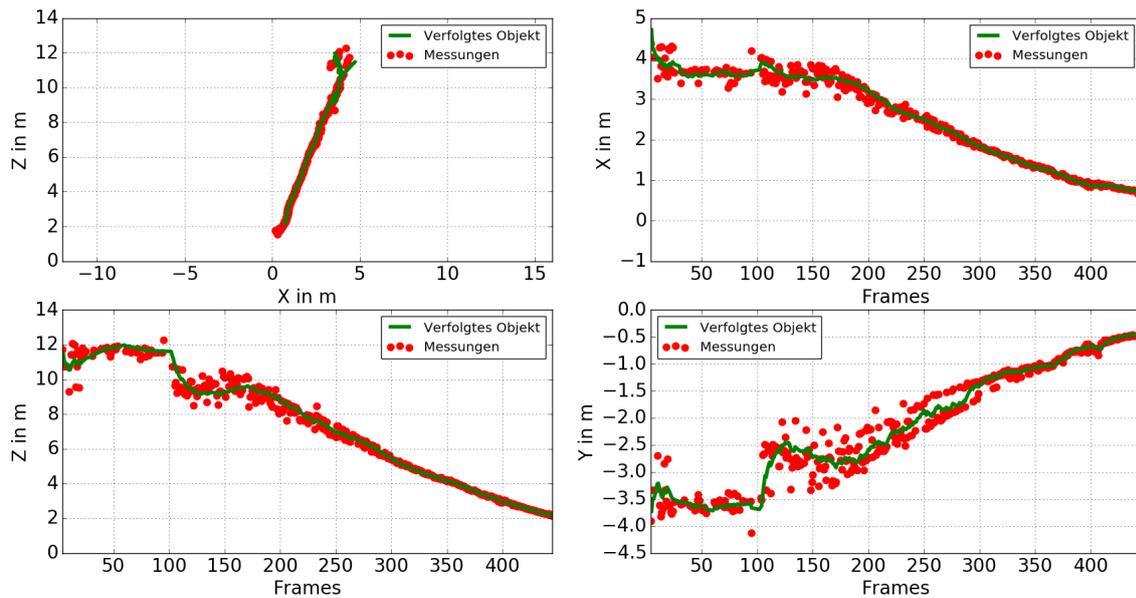


**Abbildung A.168**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den achten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.169**

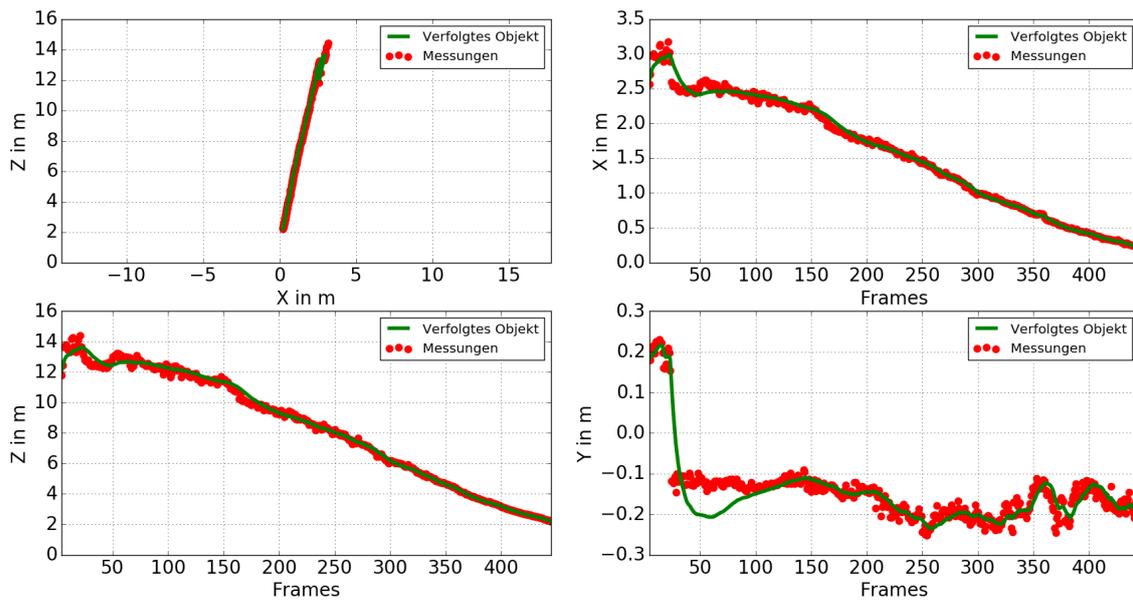
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.170**

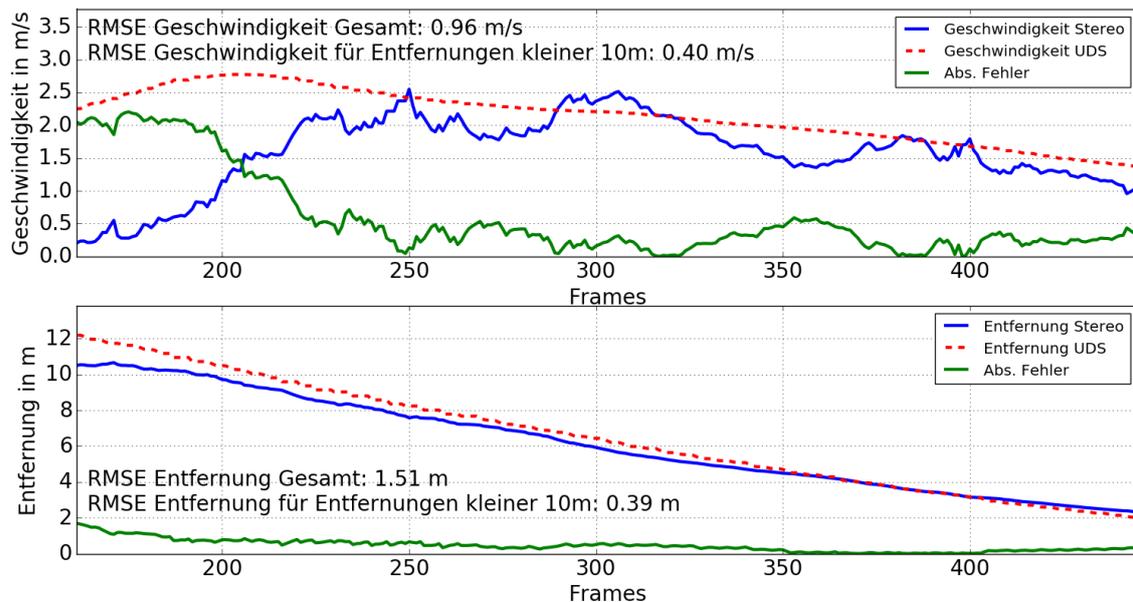
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $10^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

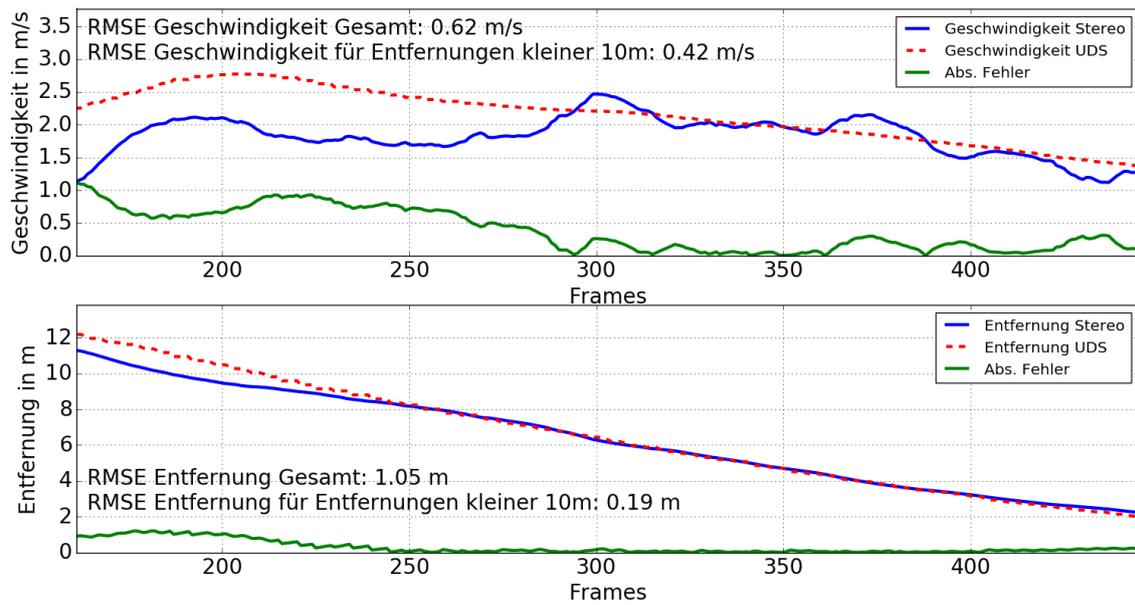
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



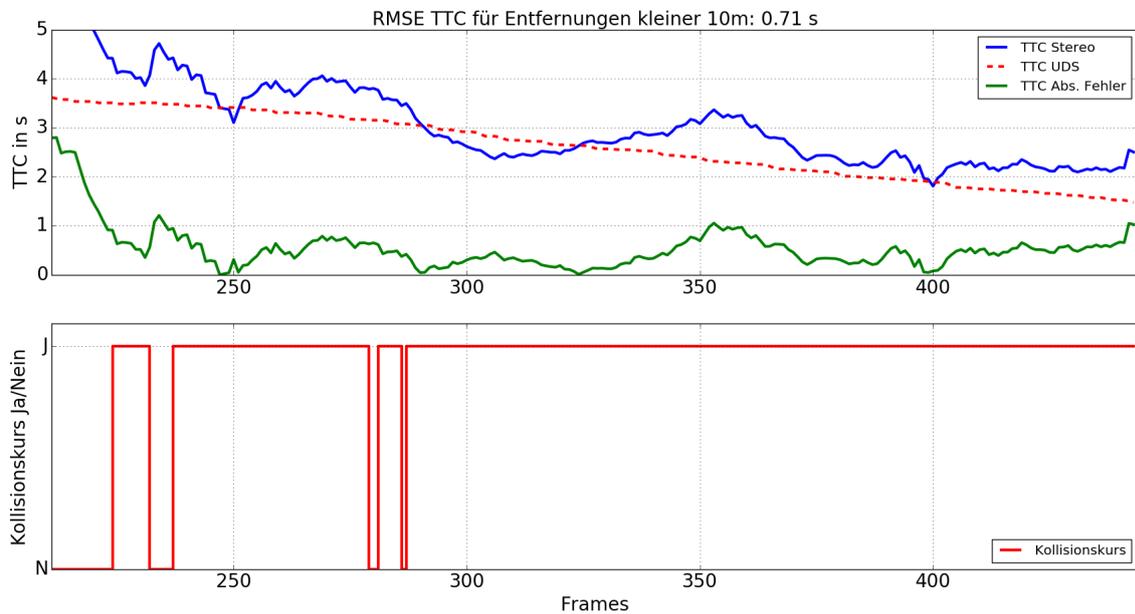
**Abbildung A.171**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.172**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

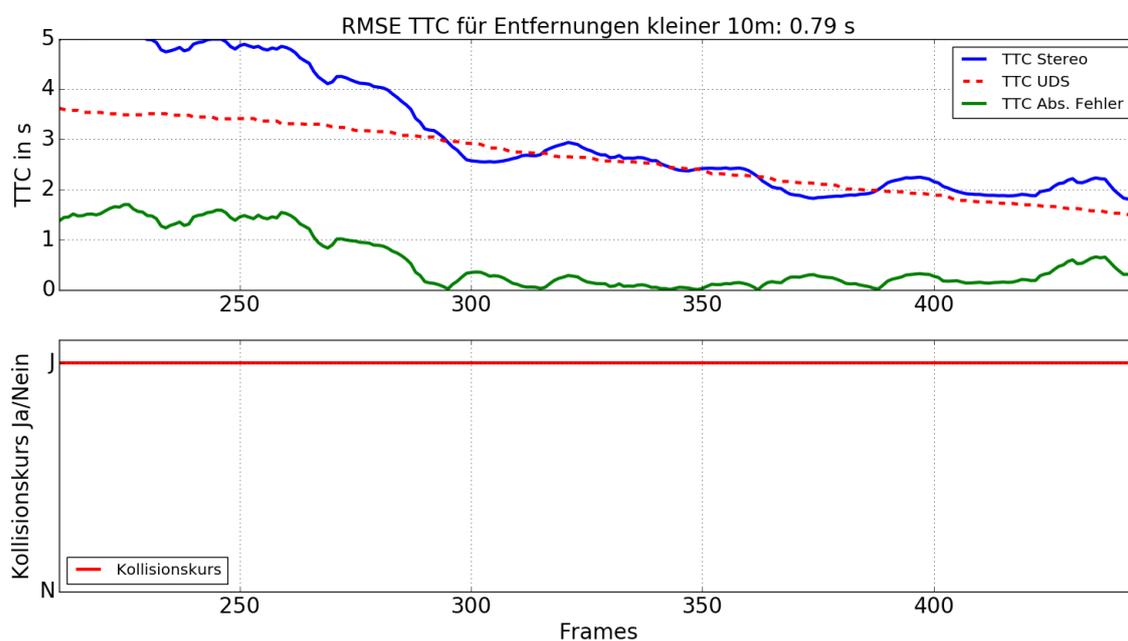


**Abbildung A.173**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

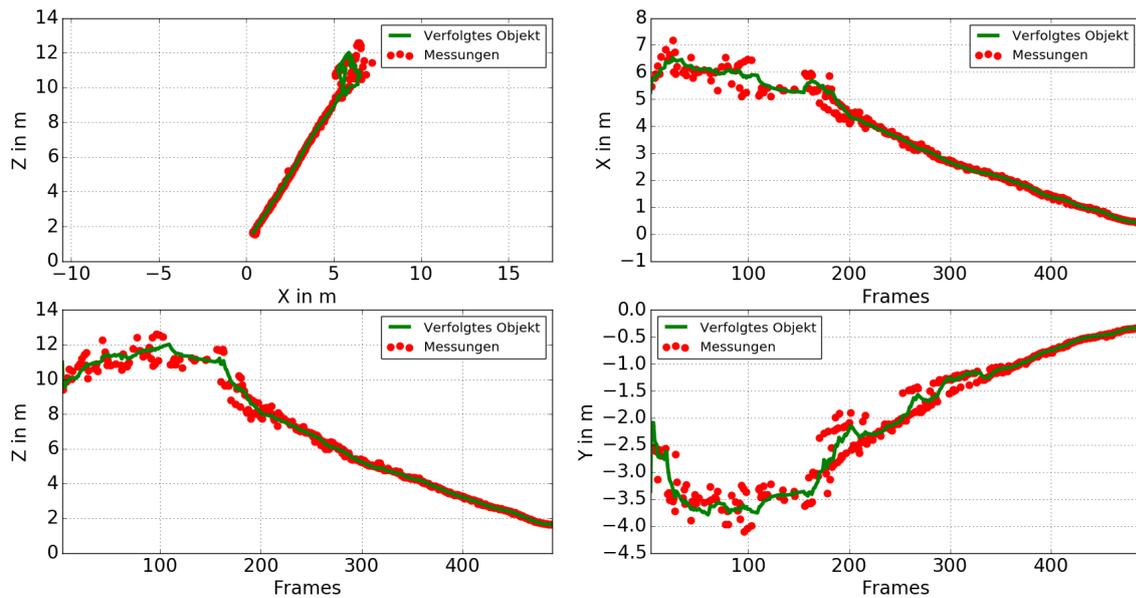


**Abbildung A.174**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den neunten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.175**

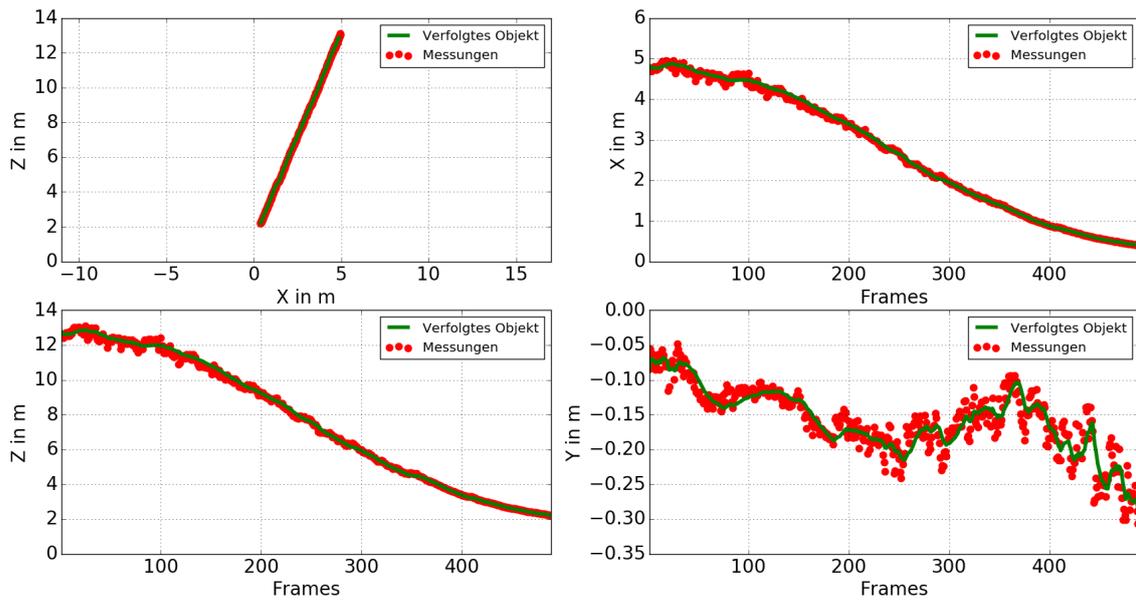
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4 m vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.176**

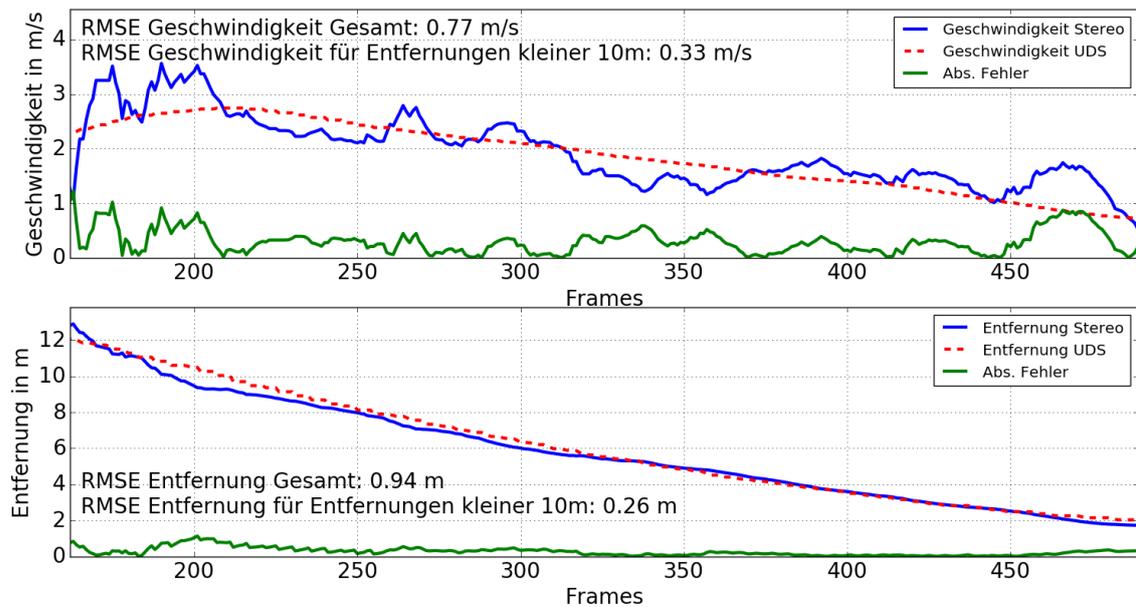
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4 m vor, der Winkel beträgt 20°. Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

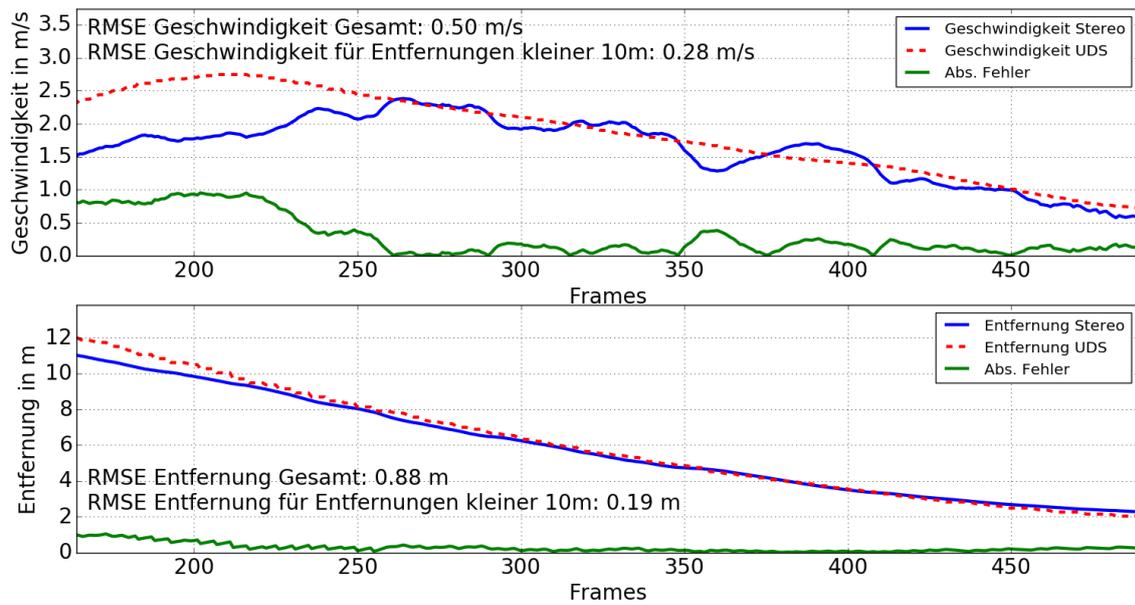
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



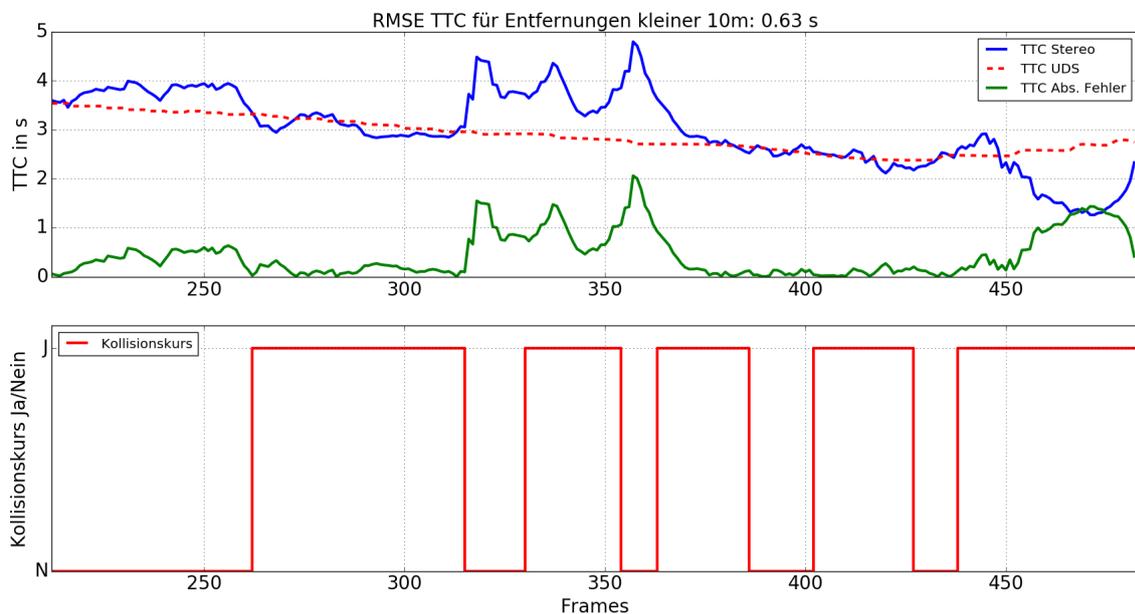
**Abbildung A.177**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.178**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

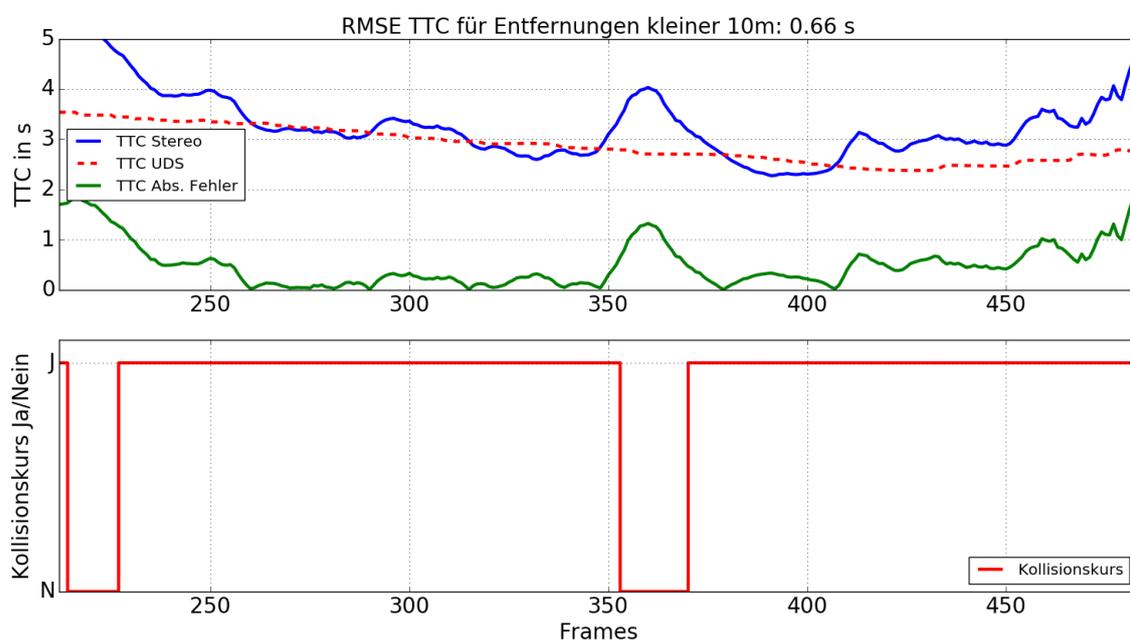


**Abbildung A.179**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

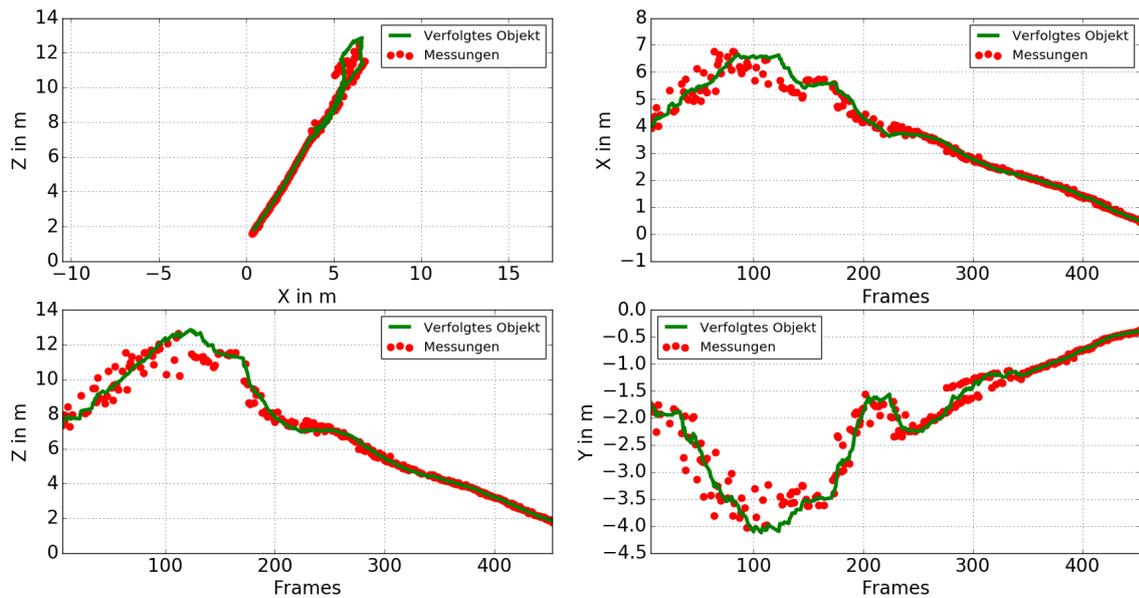


**Abbildung A.180**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den zehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.181**

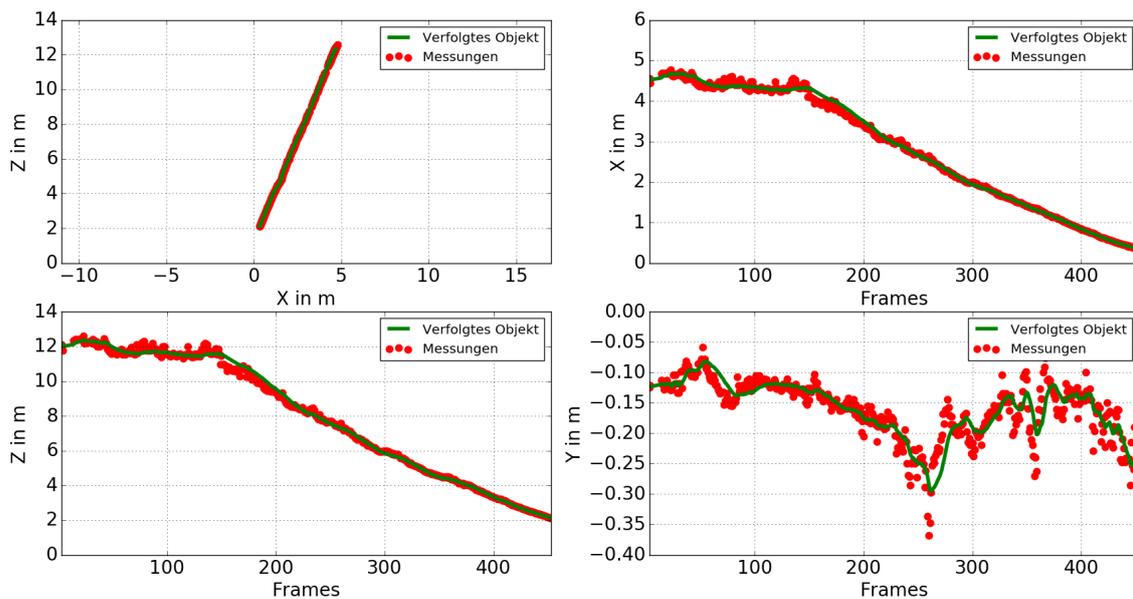
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.182**

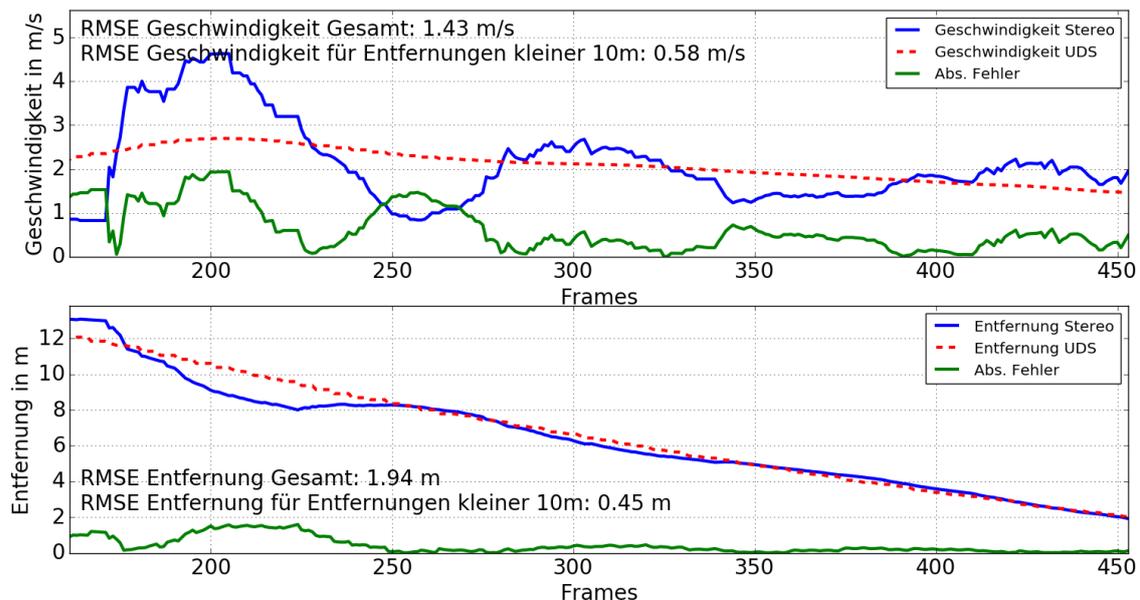
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

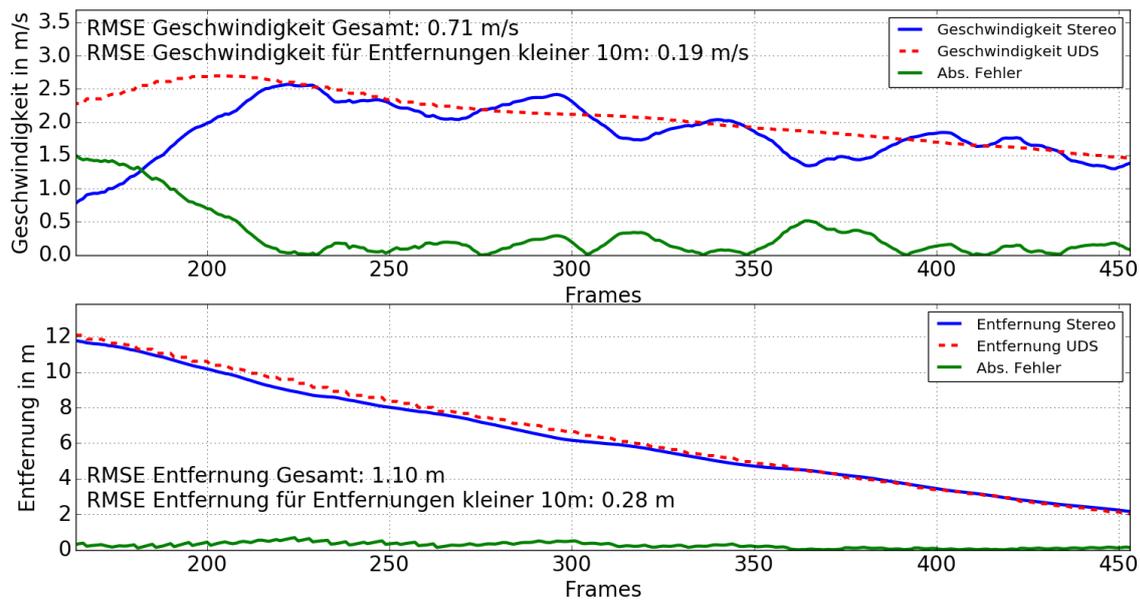
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



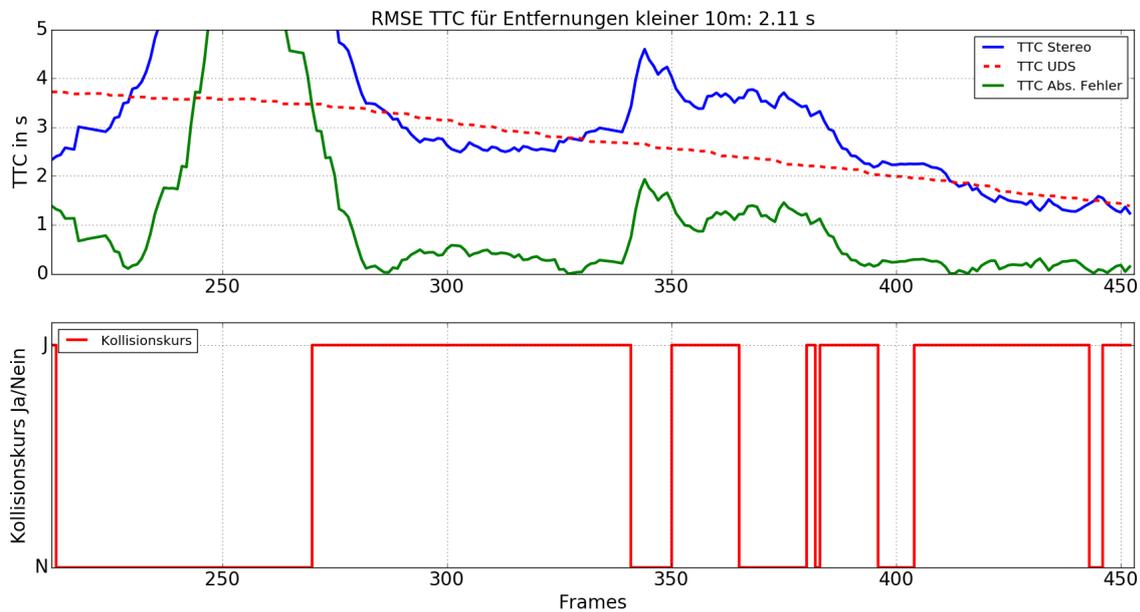
**Abbildung A.183**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.184**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

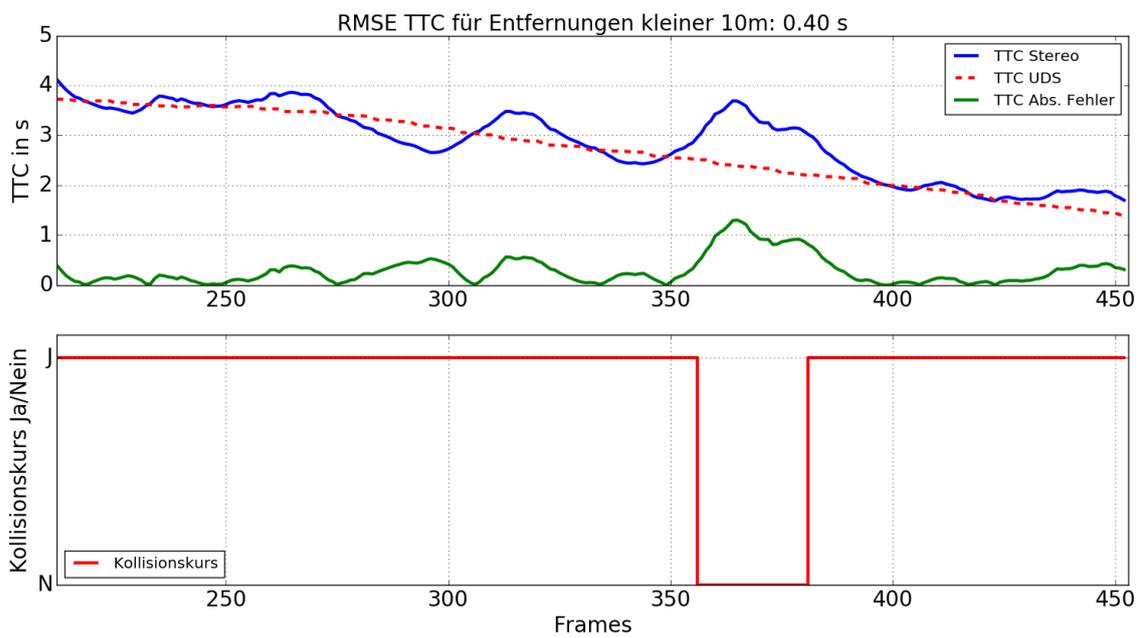


**Abbildung A.185**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

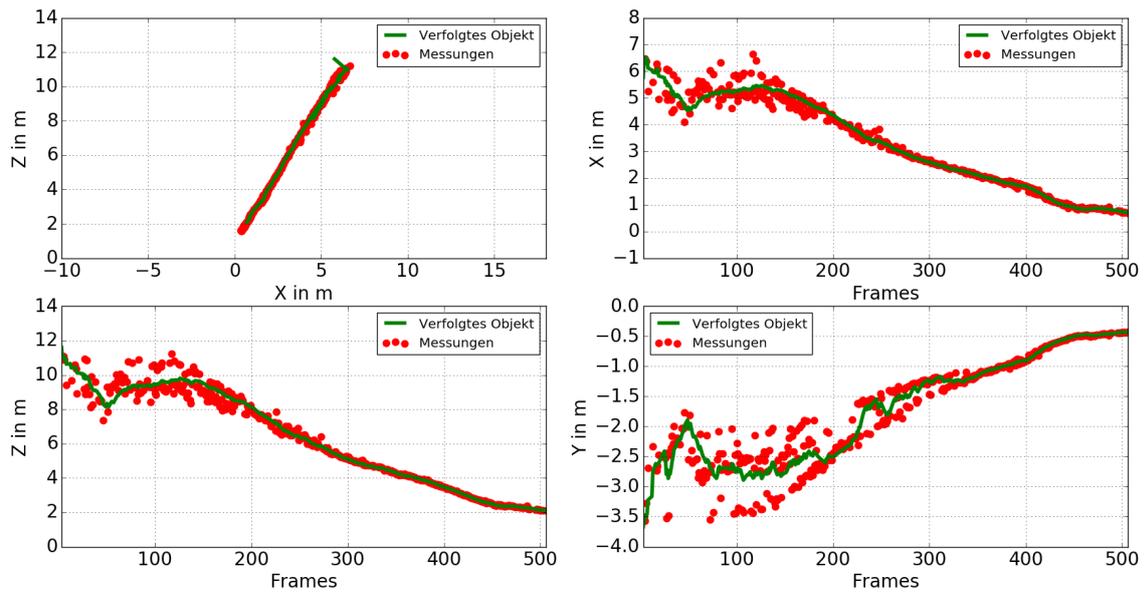


**Abbildung A.186**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den elften Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.187**

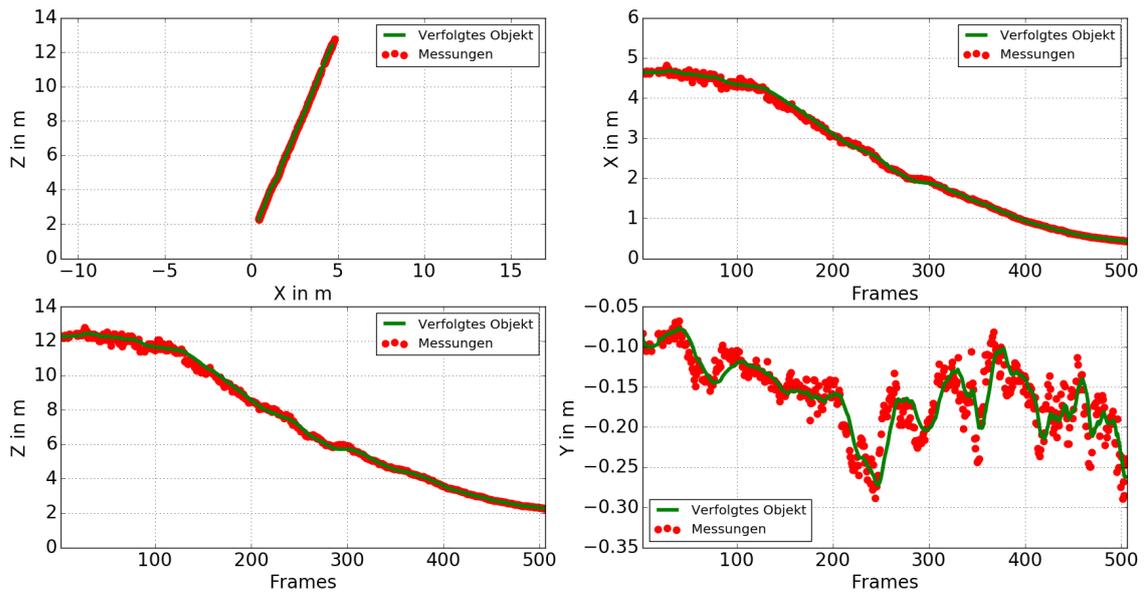
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $20^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.188**

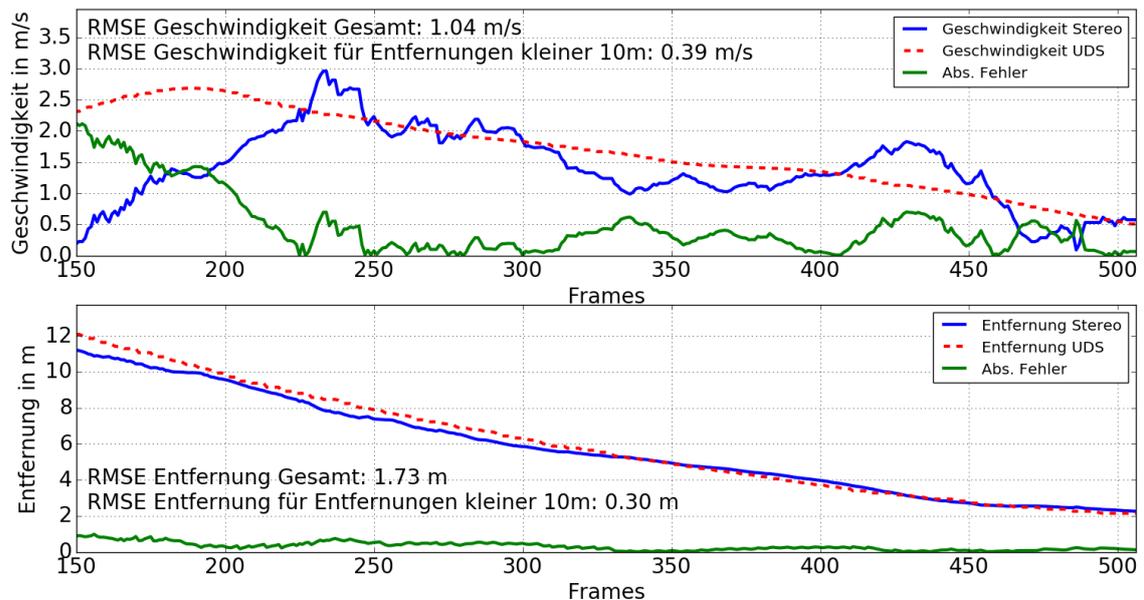
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt 20°. Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

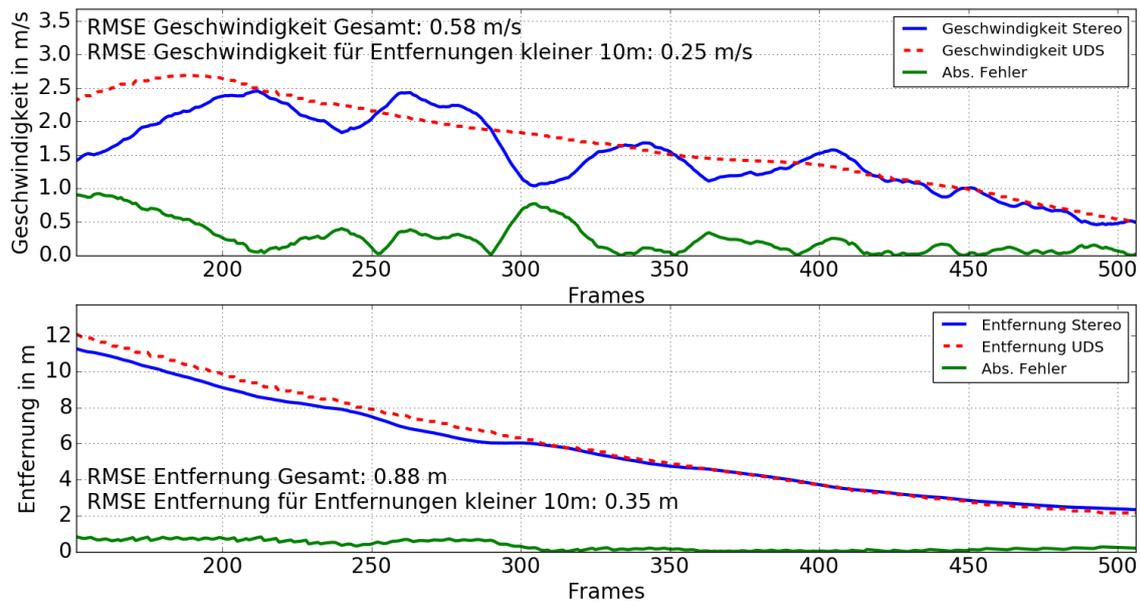
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



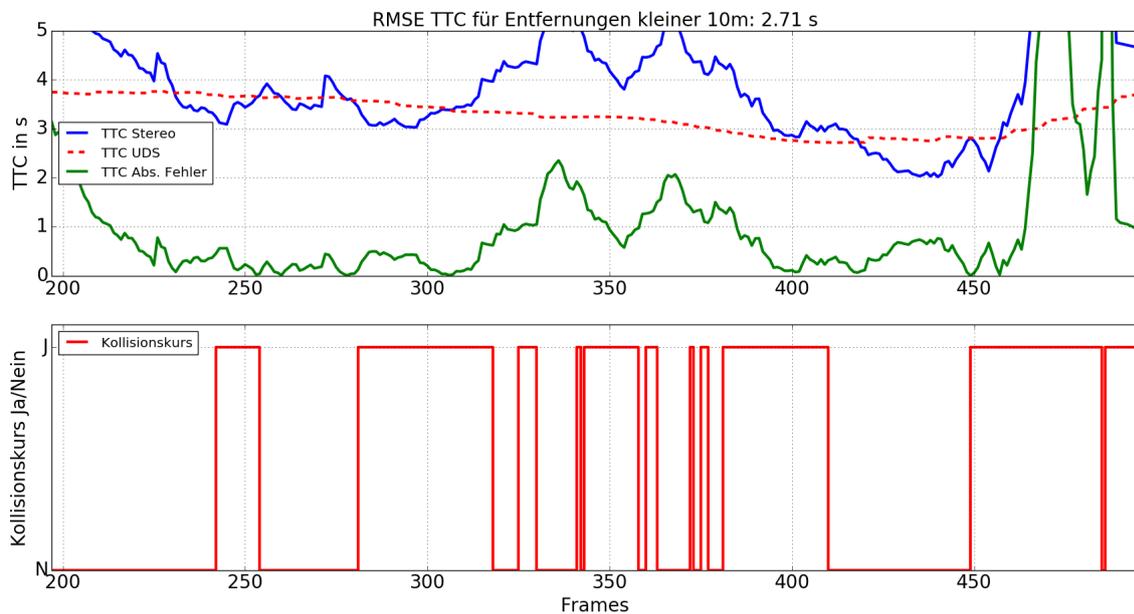
**Abbildung A.189**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.190**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

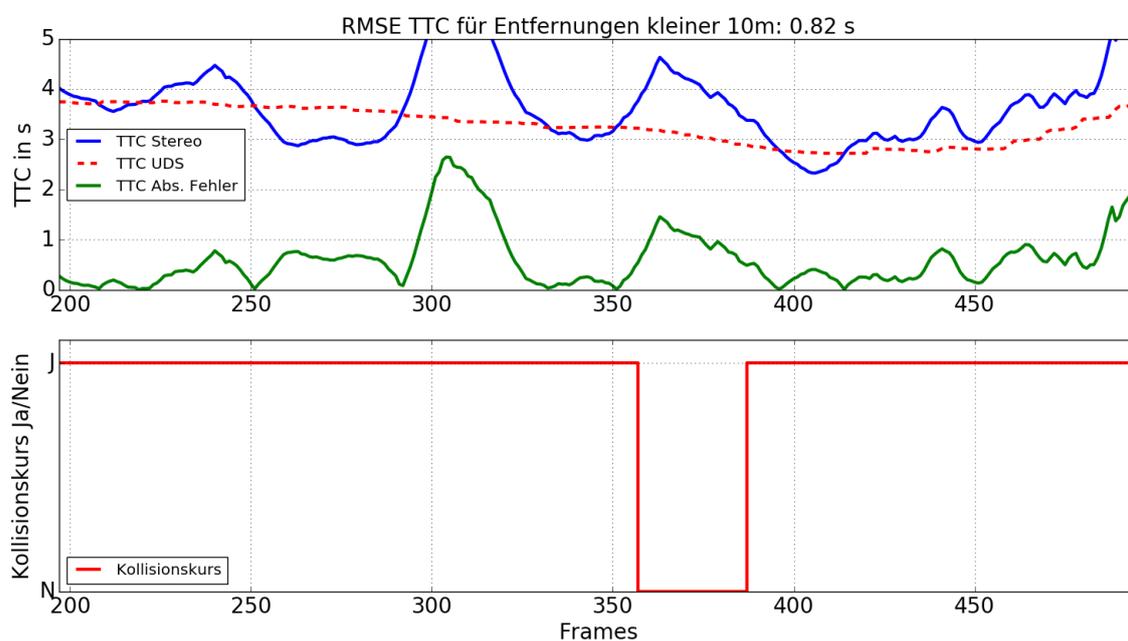


**Abbildung A.191**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

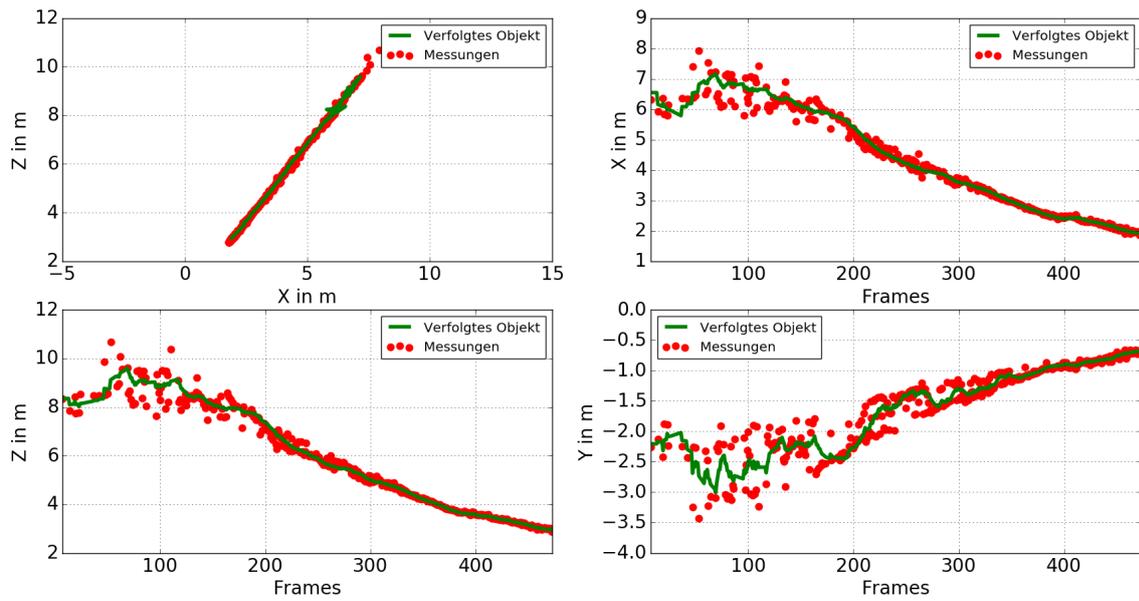


**Abbildung A.192**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den zwölften Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.193**

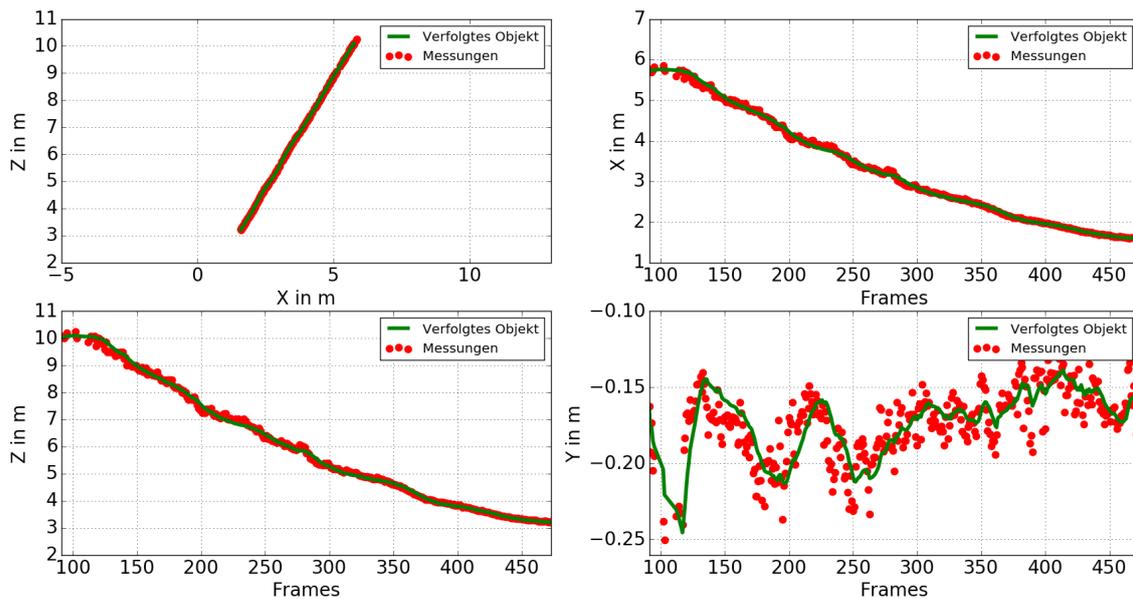
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $30^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.194**

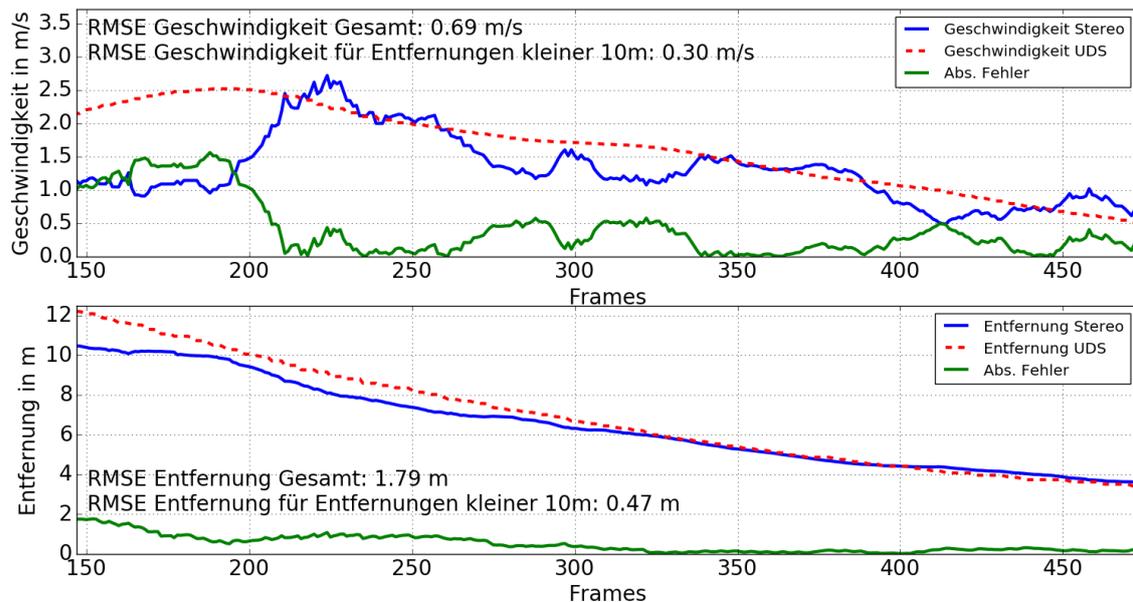
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4 m vor, der Winkel beträgt  $30^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

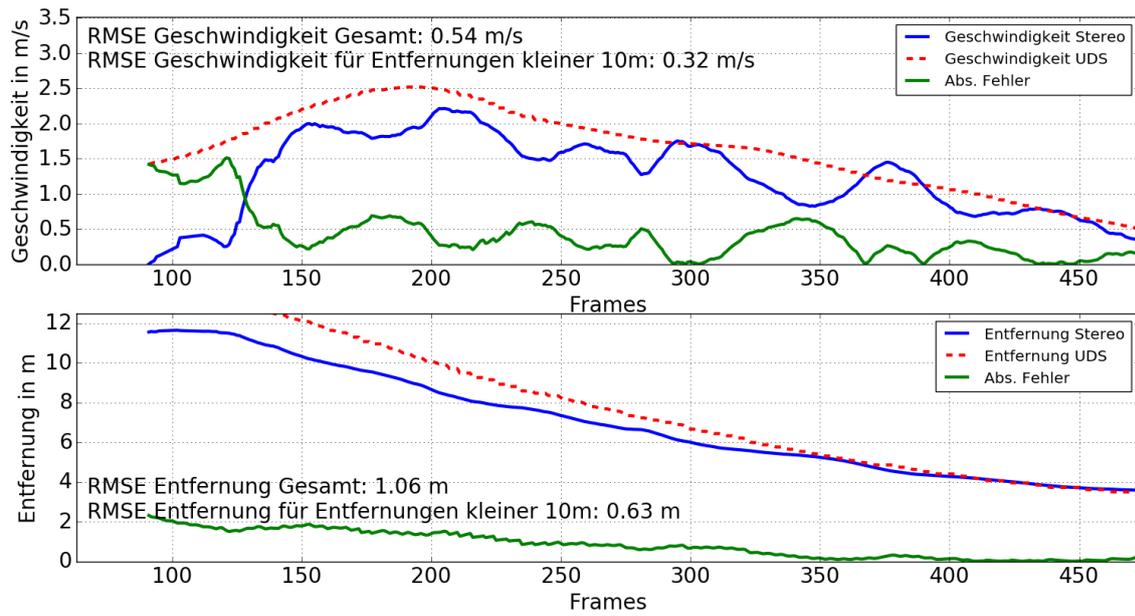
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



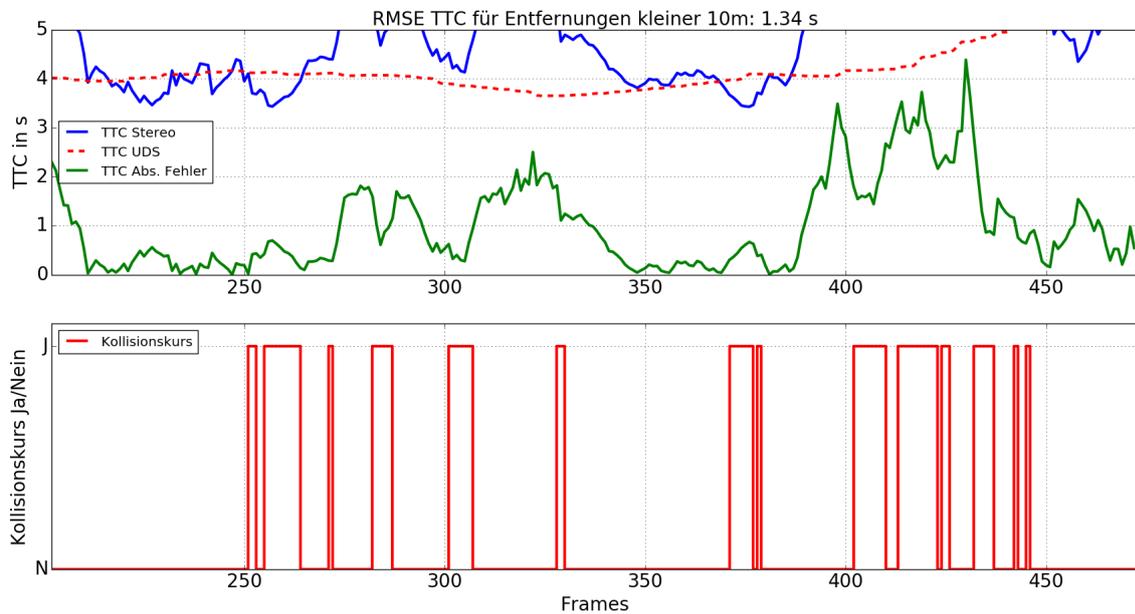
**Abbildung A.195**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



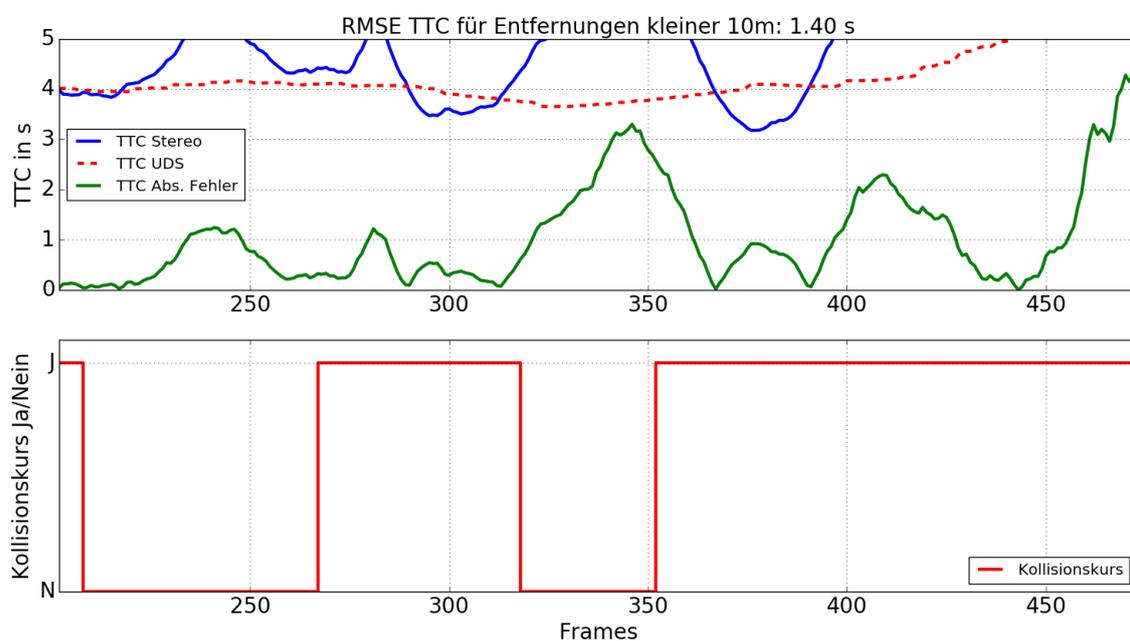
**Abbildung A.196**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.197**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).  
 Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.  
 Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

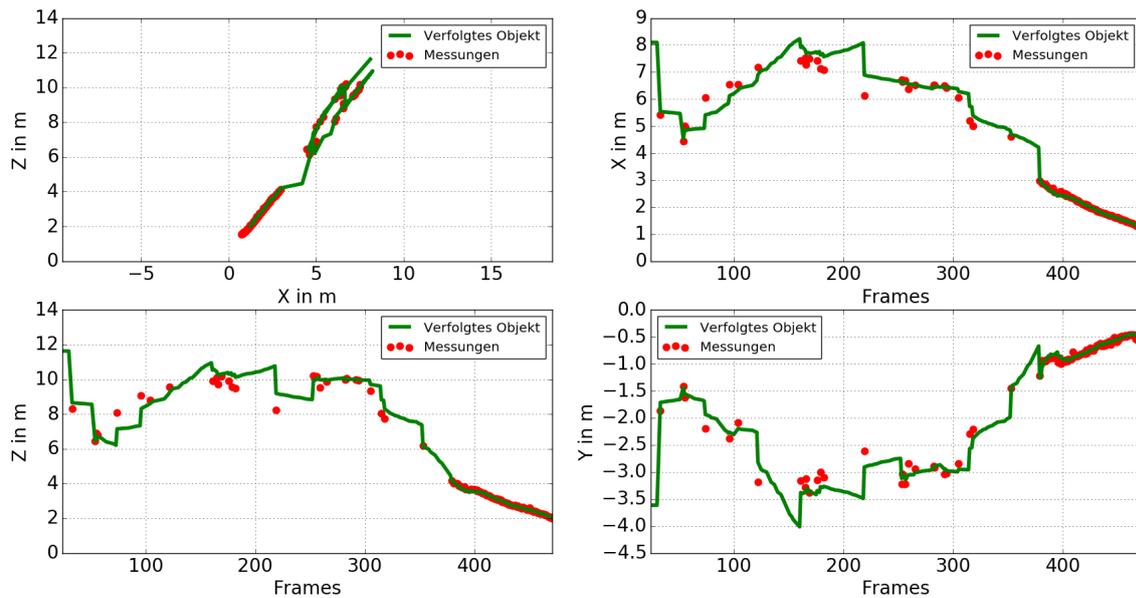


**Abbildung A.198**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den dreizehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.199**

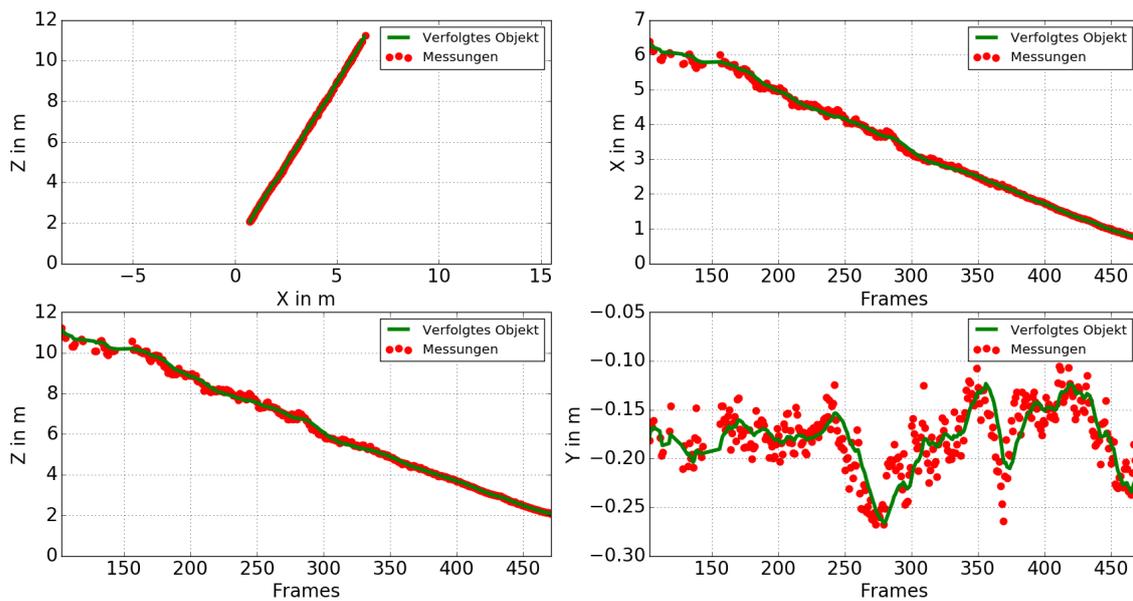
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $30^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.200**

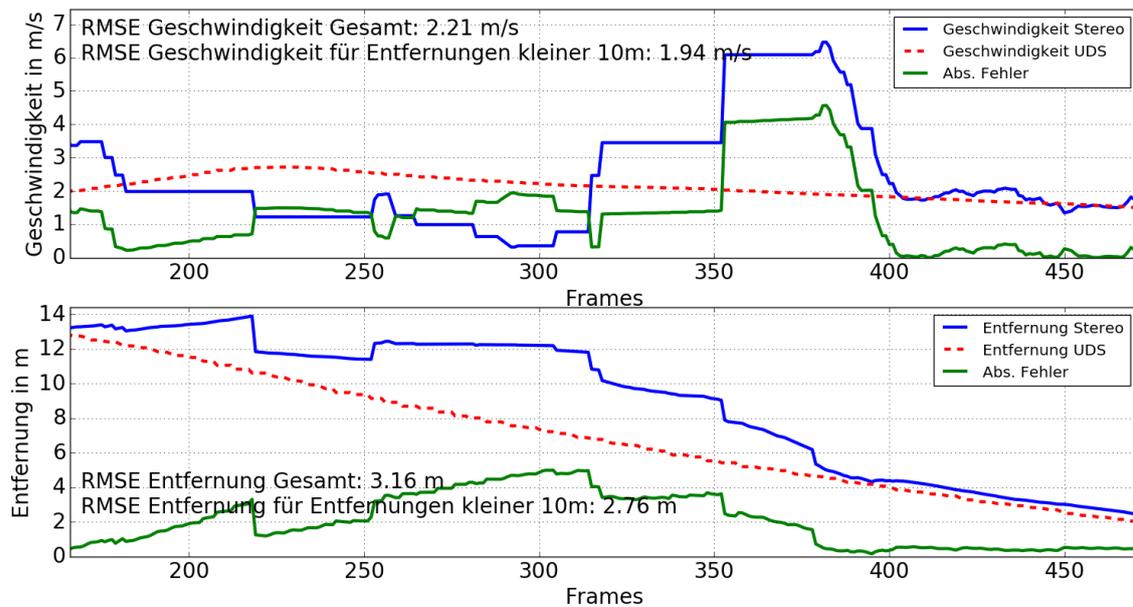
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4 m vor, der Winkel beträgt  $30^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

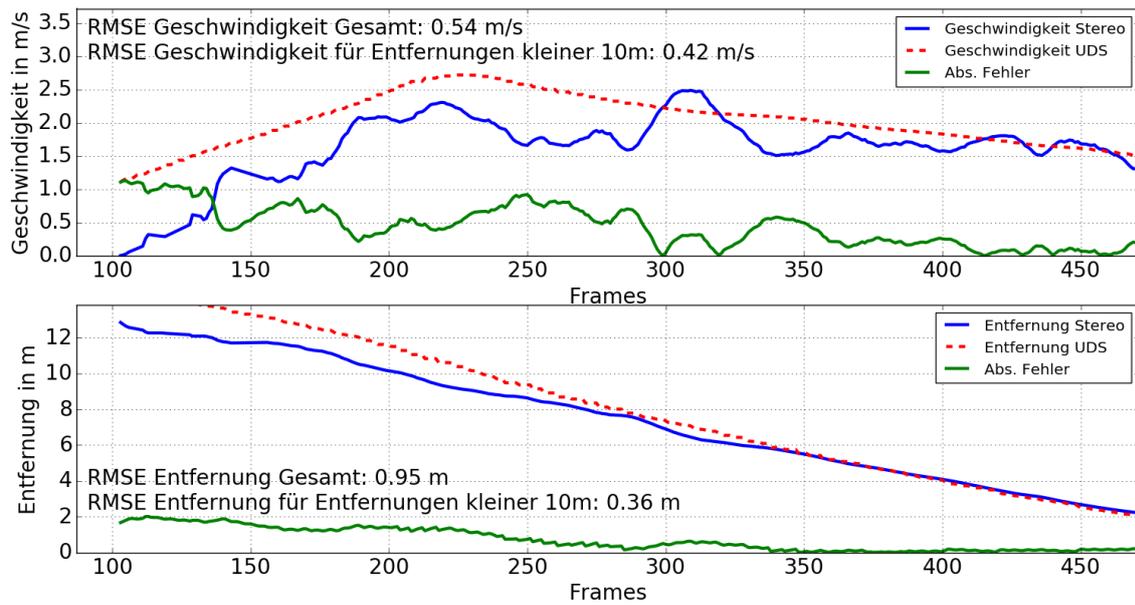
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



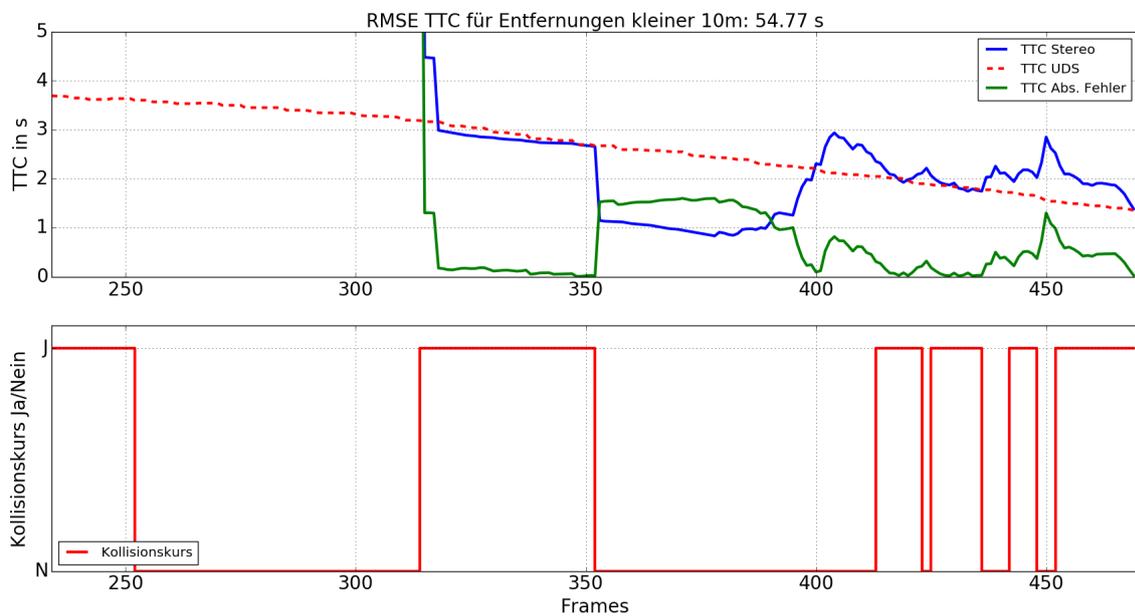
**Abbildung A.201**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.202**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

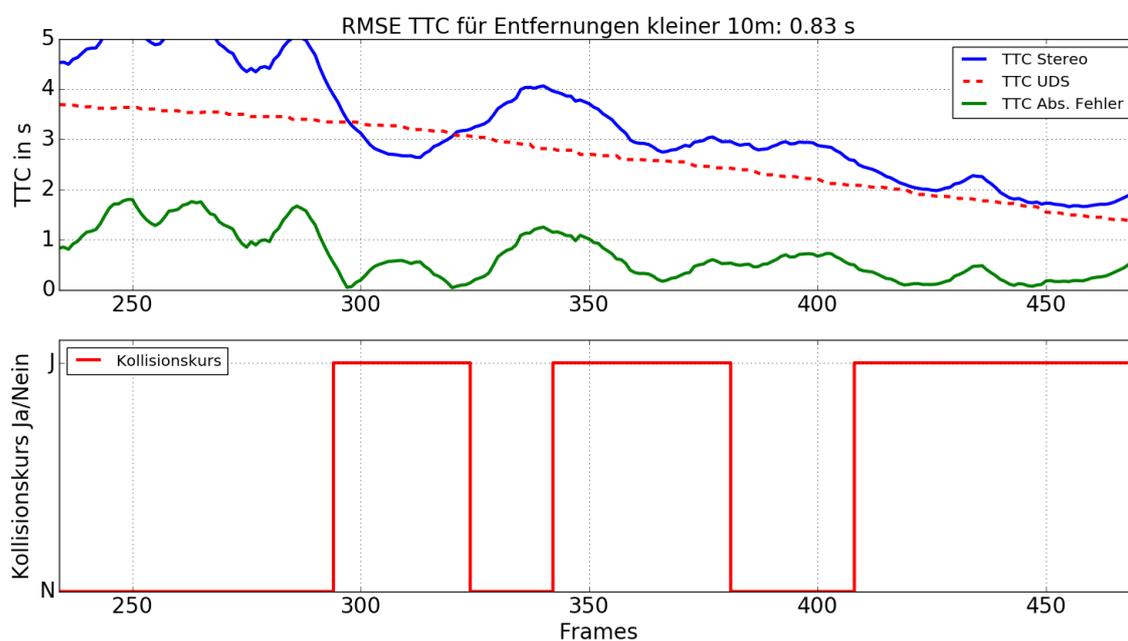


**Abbildung A.203**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10m Entfernung.

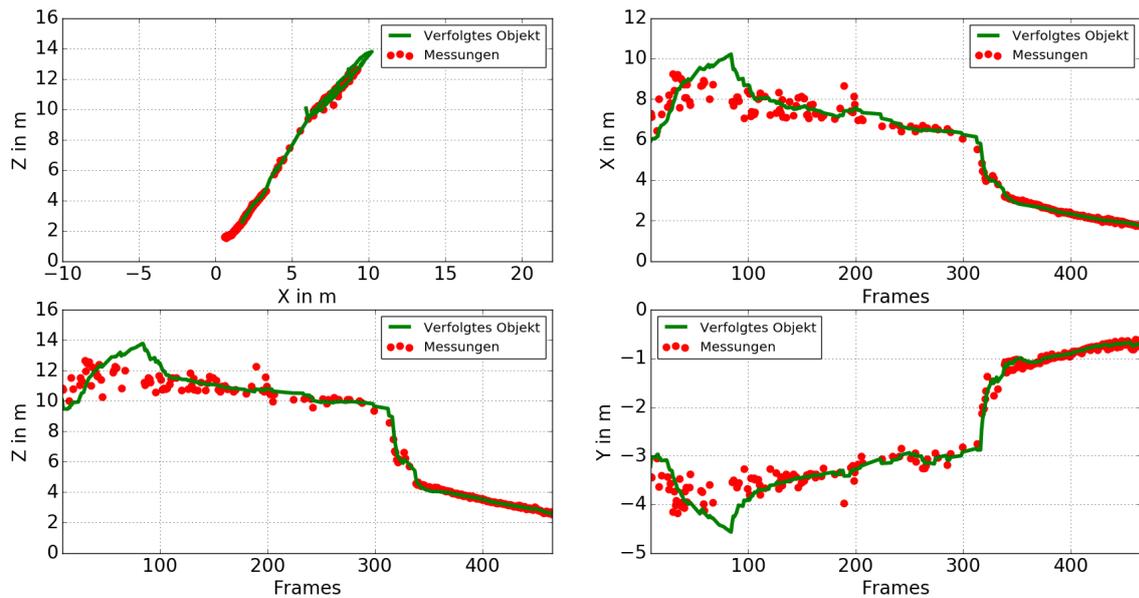


**Abbildung A.204**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den vierzehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

**Abbildung A.205**

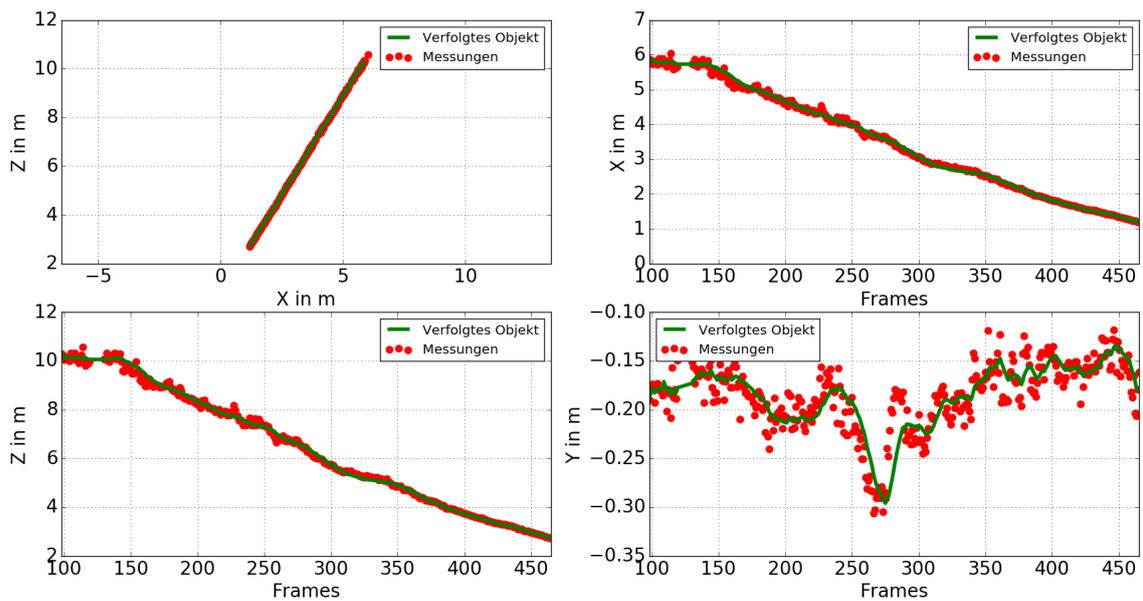
Trajektorie des Objekts berechnet mit dem regelbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4m vor, der Winkel beträgt  $30^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.206**

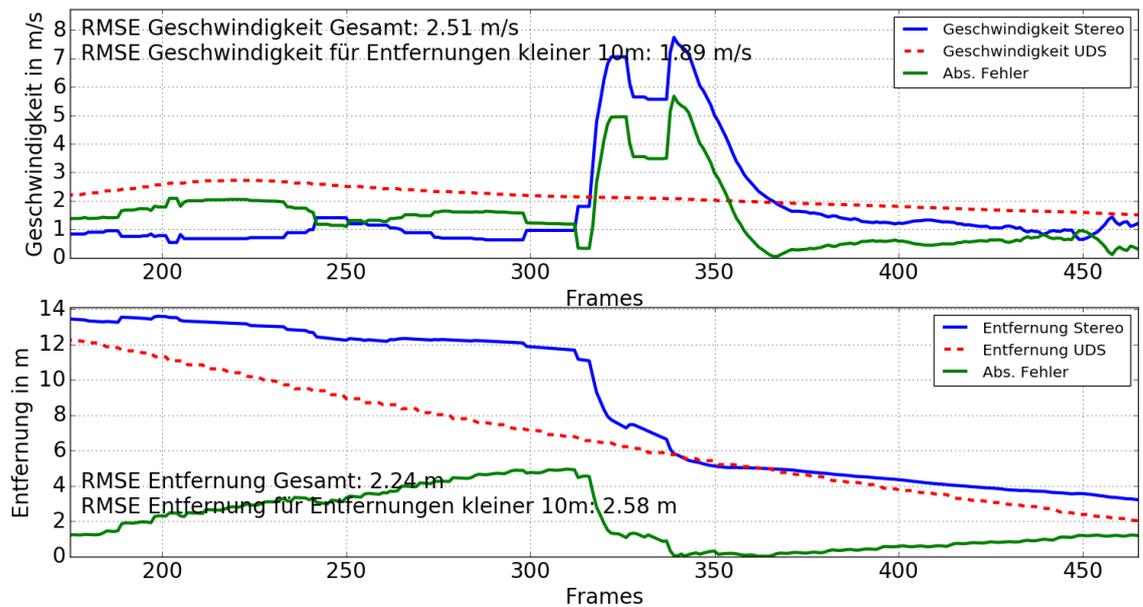
Trajektorie des Objekts berechnet mit dem modellbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 2. Es liegt ein *Offset* von 0,4 m vor, der Winkel beträgt  $30^\circ$ . Das Signal des *Trackers* wird als grüne Linie, zum Update verwendete Messwerte als roter Punkt dargestellt.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

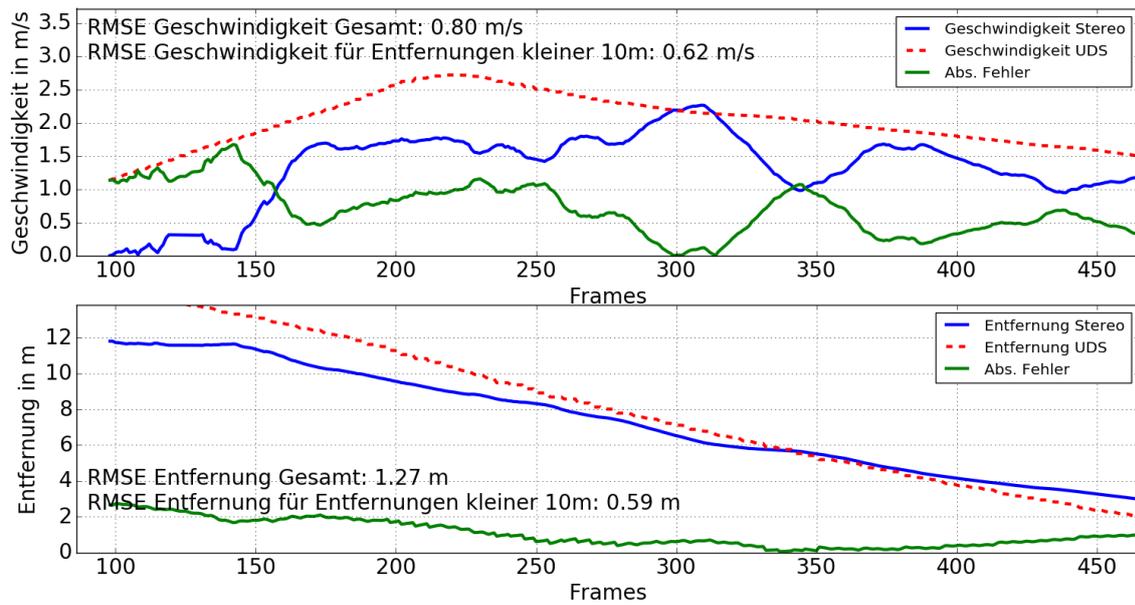
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



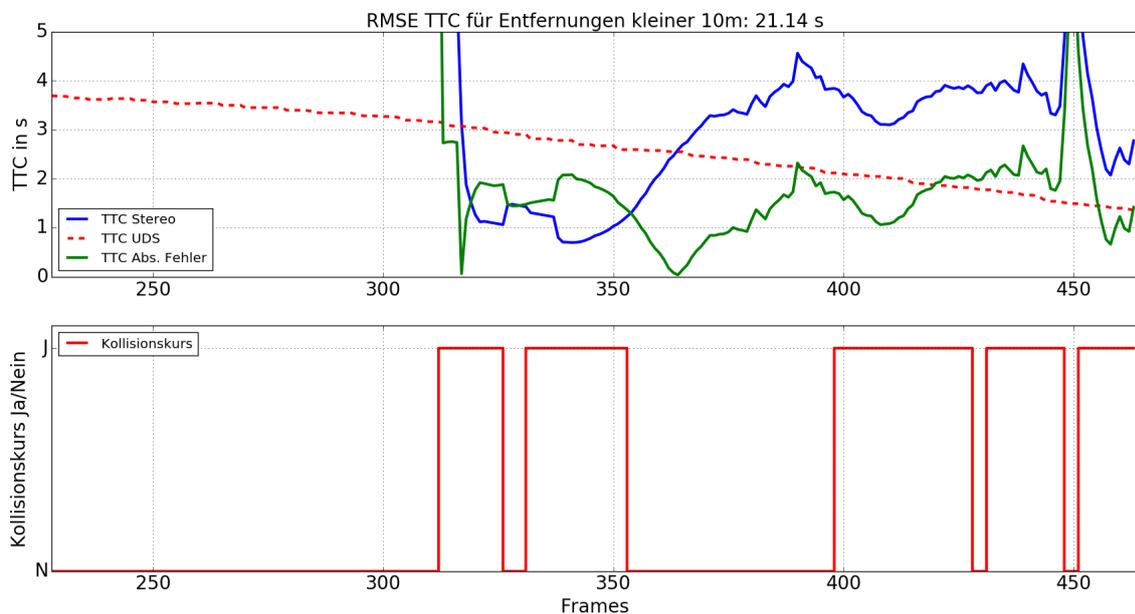
**Abbildung A.207**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem regelbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.208**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts berechnet mit dem modellbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 2. Die UDS Daten werden als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

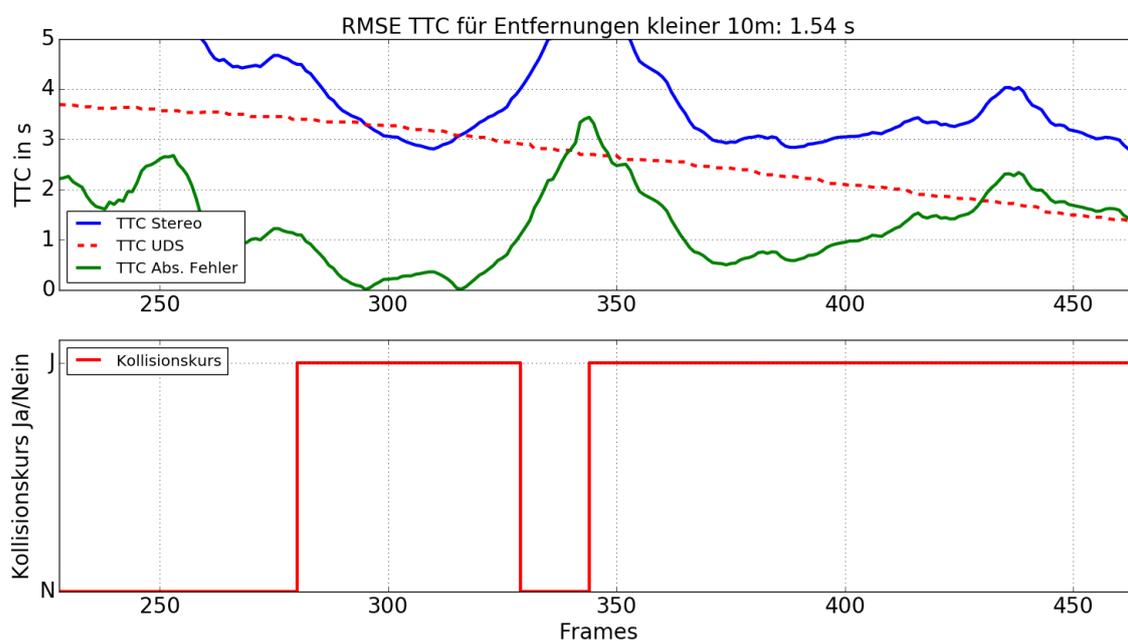


**Abbildung A.209**

TTC und Kollisionsdetektion des Objekts berechnet mit dem regelbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.



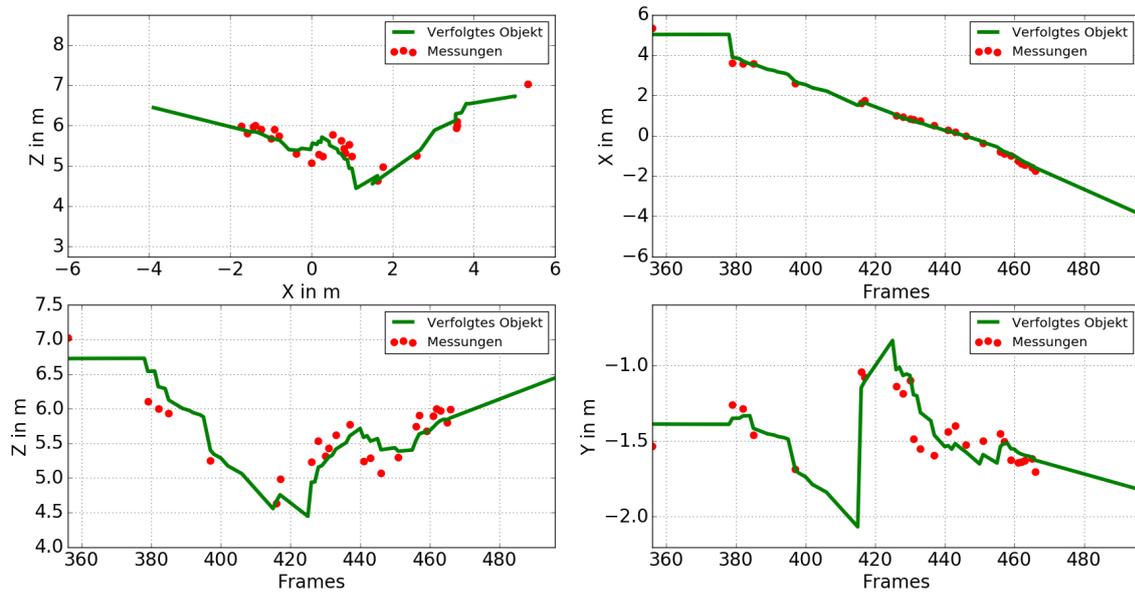
**Abbildung A.210**

TTC und Kollisionsdetektion des Objekts berechnet mit dem modellbasierten Ansatz für den fünfzehnten Versuch der TTC Validationsversuche im Szenario 2. Für die TTC werden die UDS Daten als Grundwahrheit angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: TTC des Objekts für den Nahbereich ab 10 m Entfernung.

Unten: Kollisionsdetektion des Objekts für den Nahbereich ab 10 m Entfernung.

## Ergebnisse der Einzelversuche der PET Validationsversuche

**Abbildung A.211**

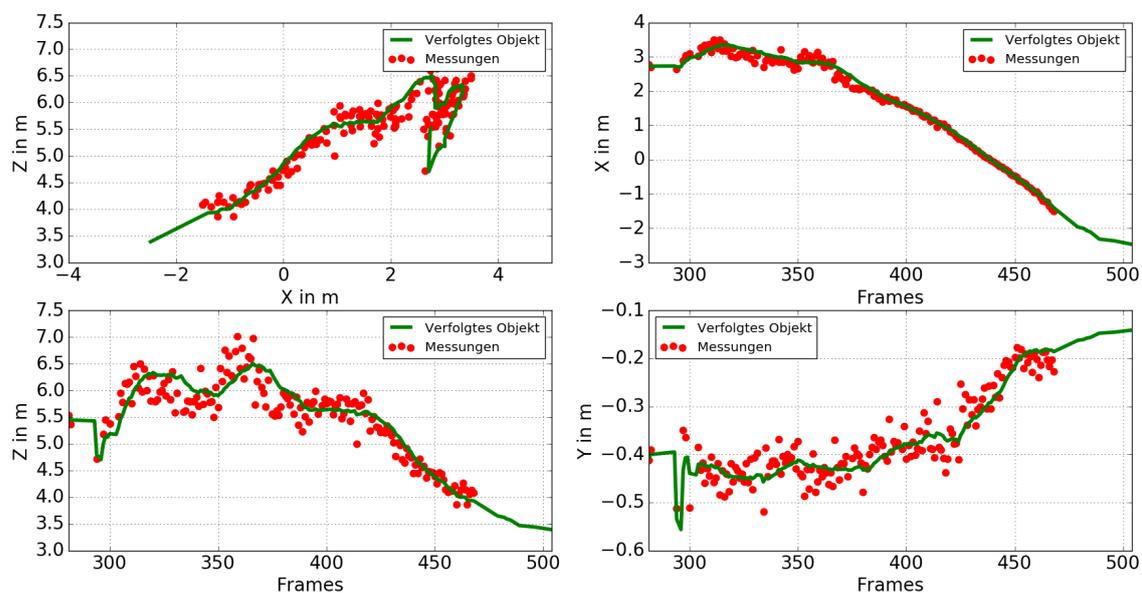
Trajektorie des Objekts für den PET Versuch 1, berechnet mit dem regelbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.212**

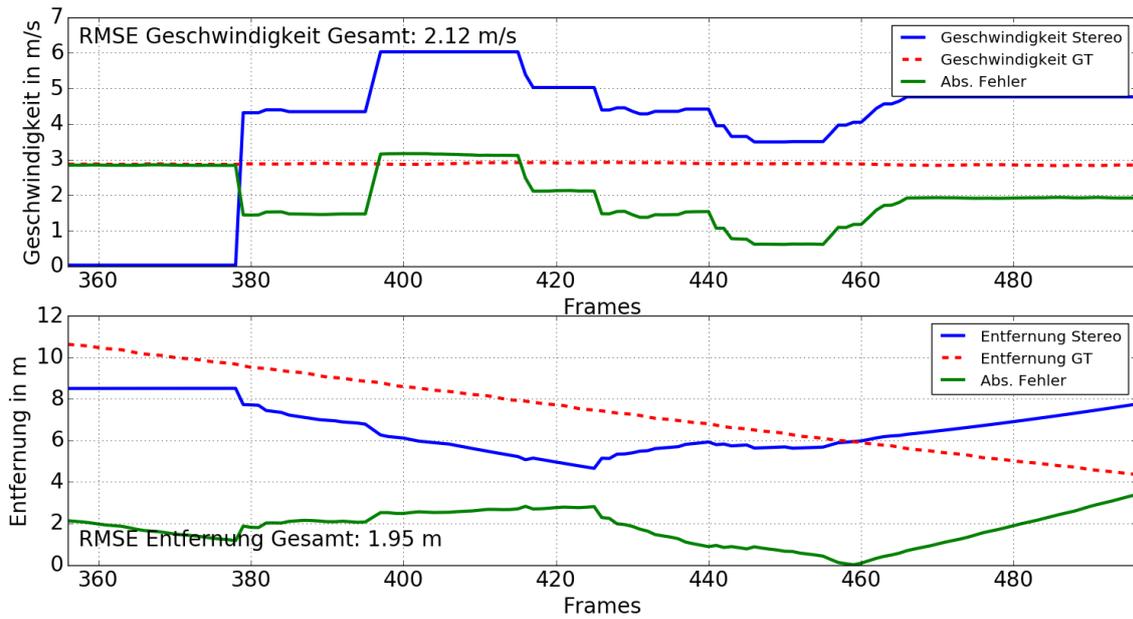
Trajektorie des Objekts für den PET Versuch 1, berechnet mit dem modellbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

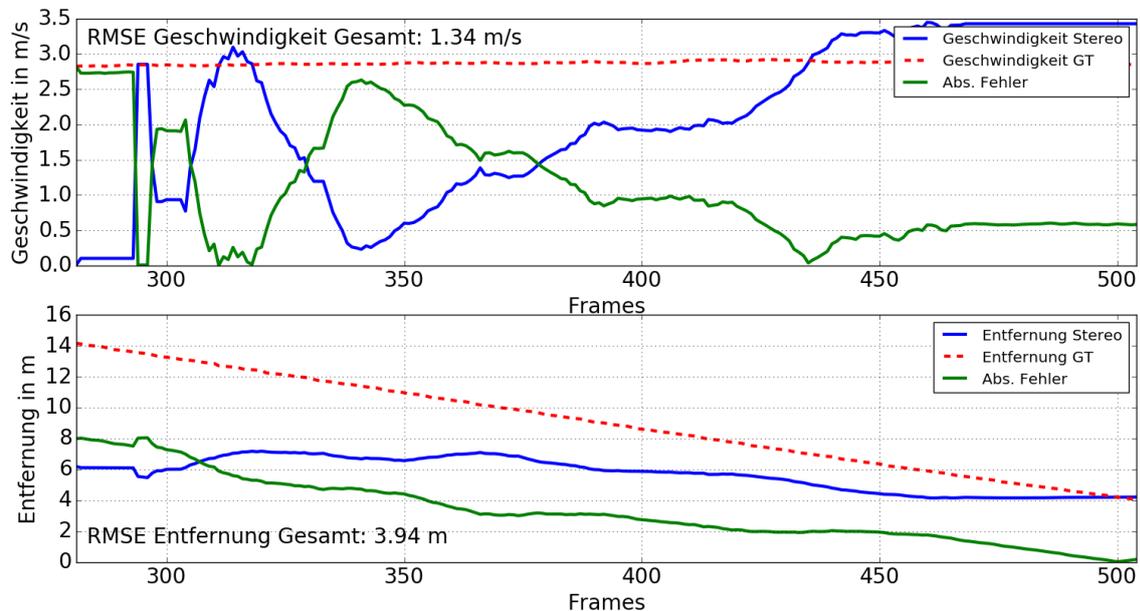
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



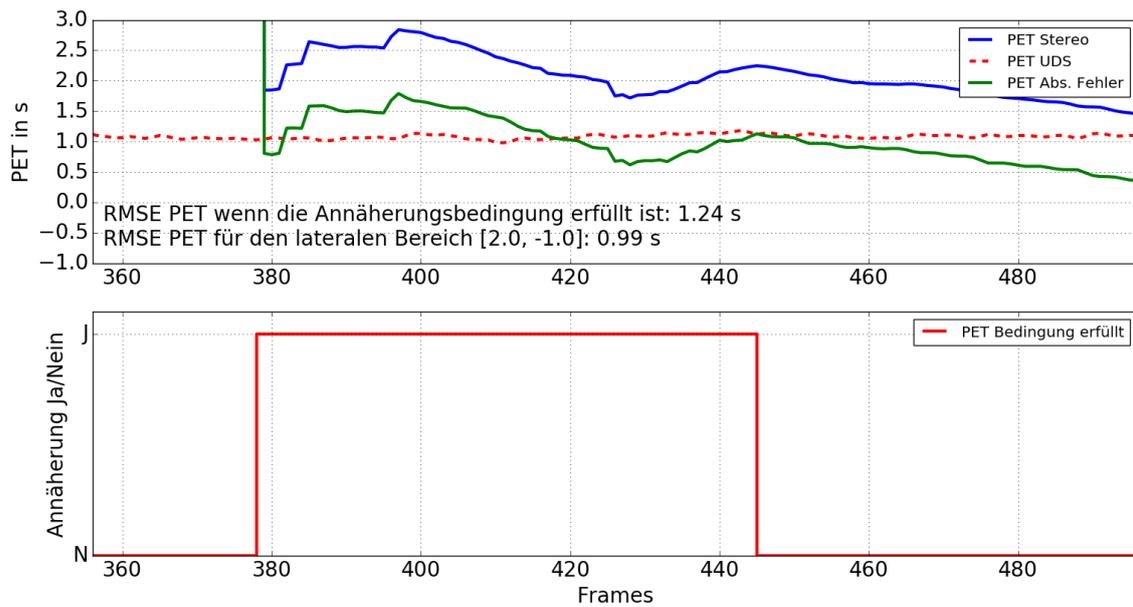
**Abbildung A.213**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 1 berechnet mit dem regelbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.214**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 1 berechnet mit dem modellbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).

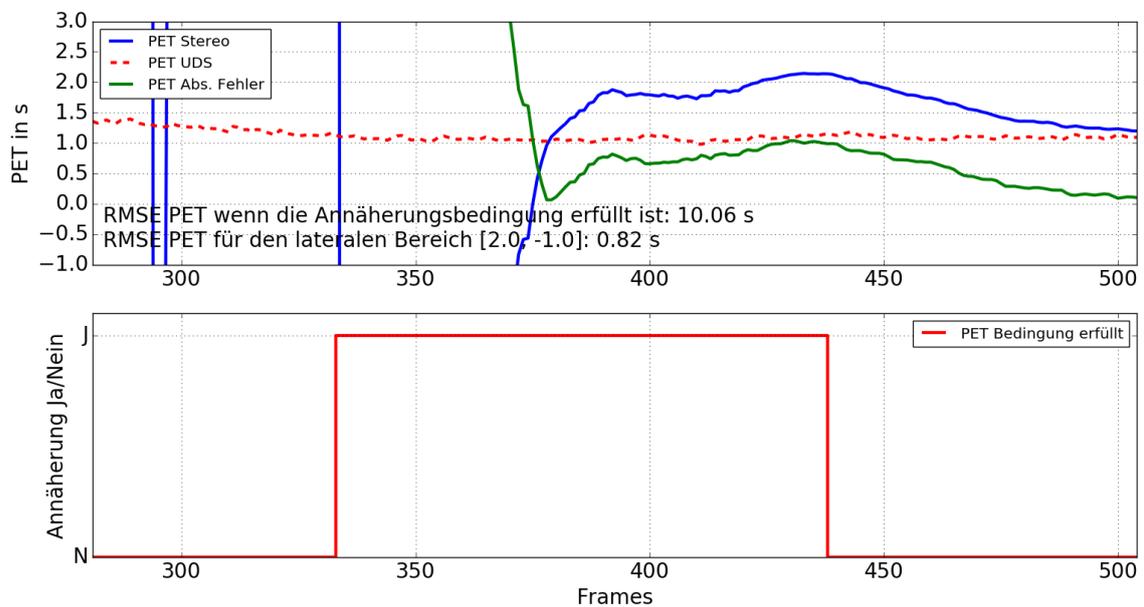


**Abbildung A.215**

PET und Grundbedingung des Objekts für den PET Versuch 1 berechnet mit dem regelbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.

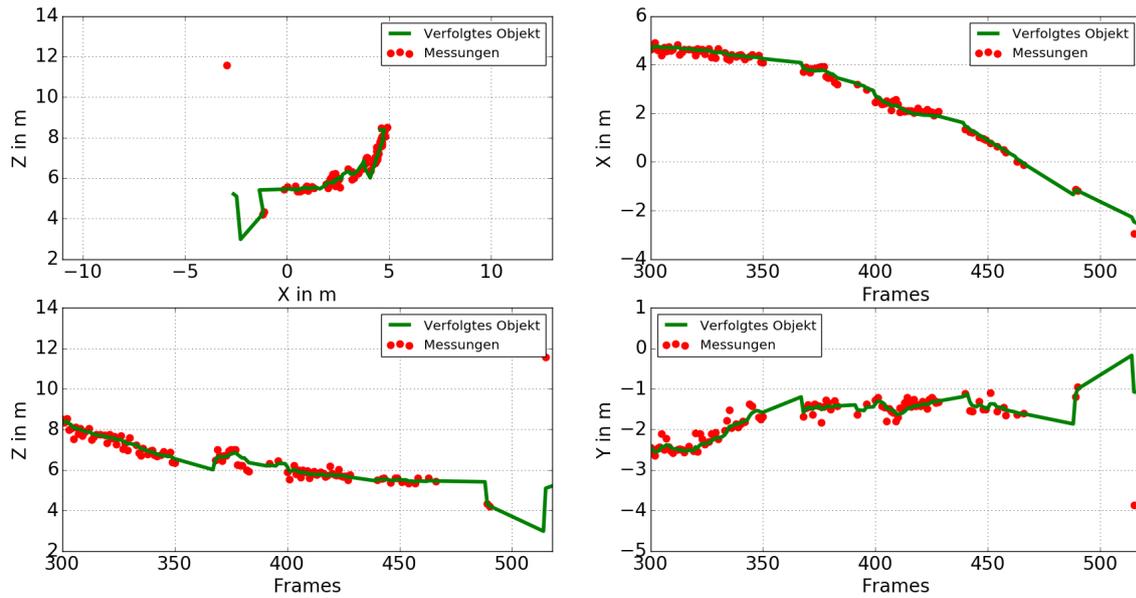


**Abbildung A.216**

PET und Grundbedingung des Objekts für den PET Versuch 1 berechnet mit dem modellbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.



**Abbildung A.217**

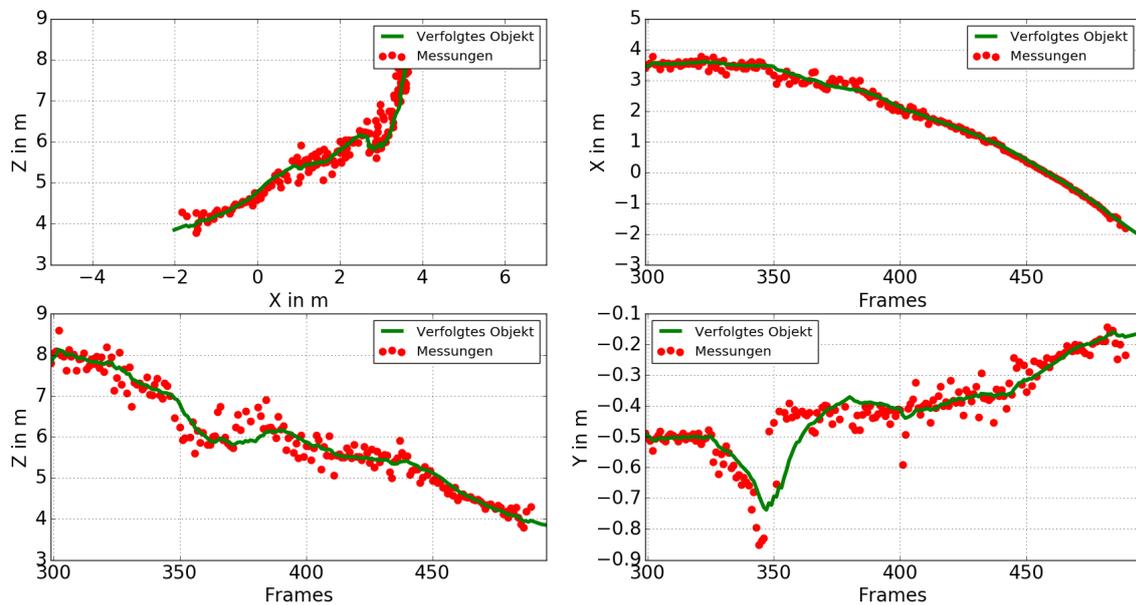
Trajektorie des Objekts für den PET Versuch 2, berechnet mit dem regelbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.218**

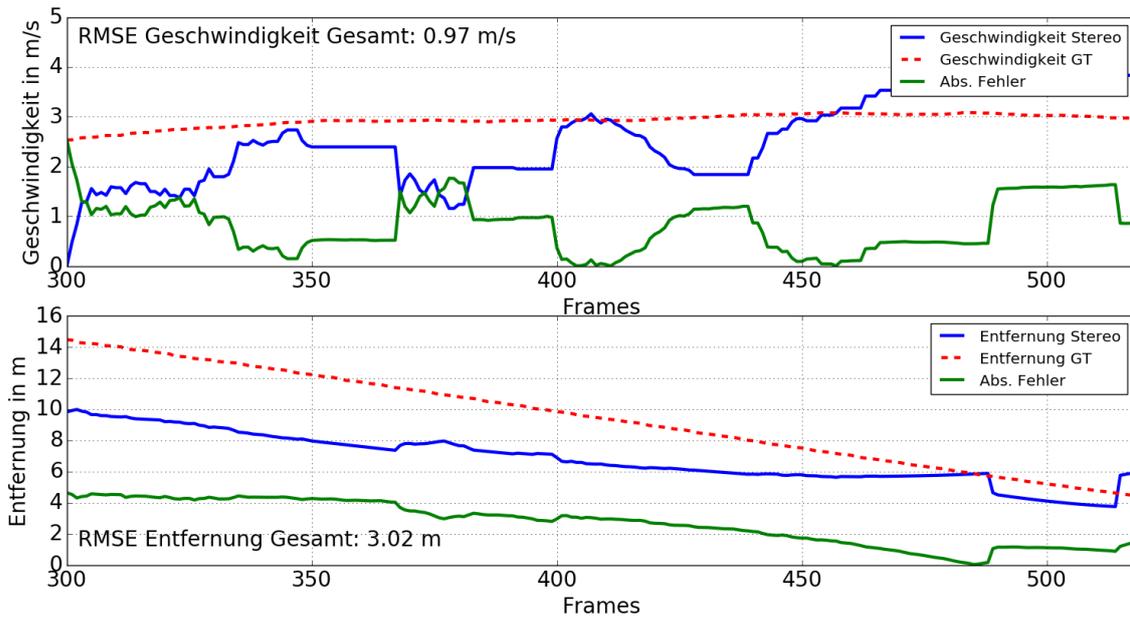
Trajektorie des Objekts für den PET Versuch 2, berechnet mit dem modellbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

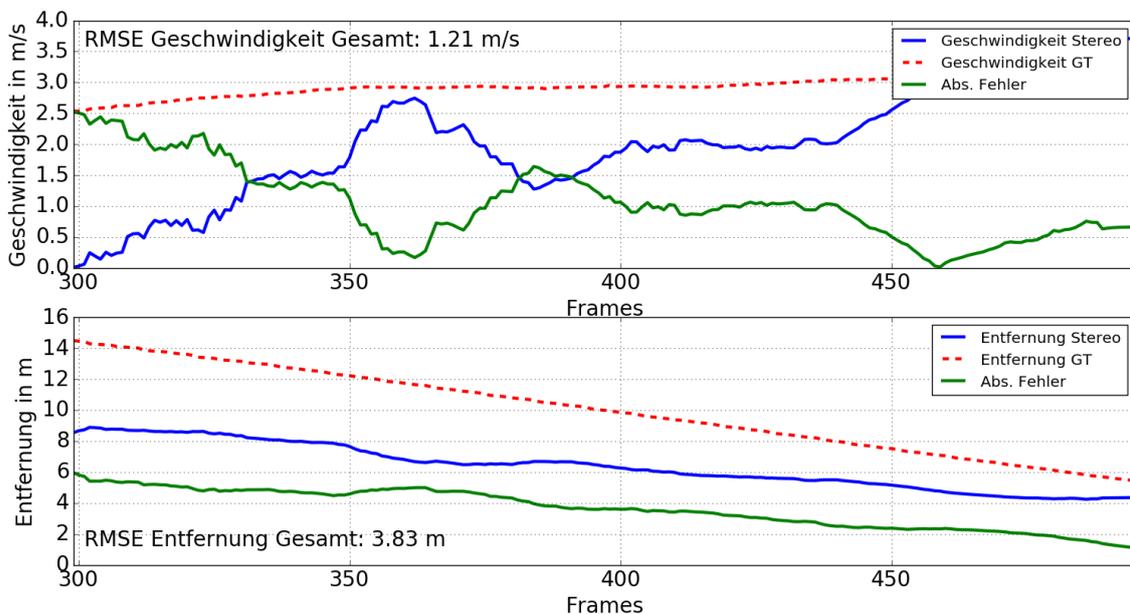
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



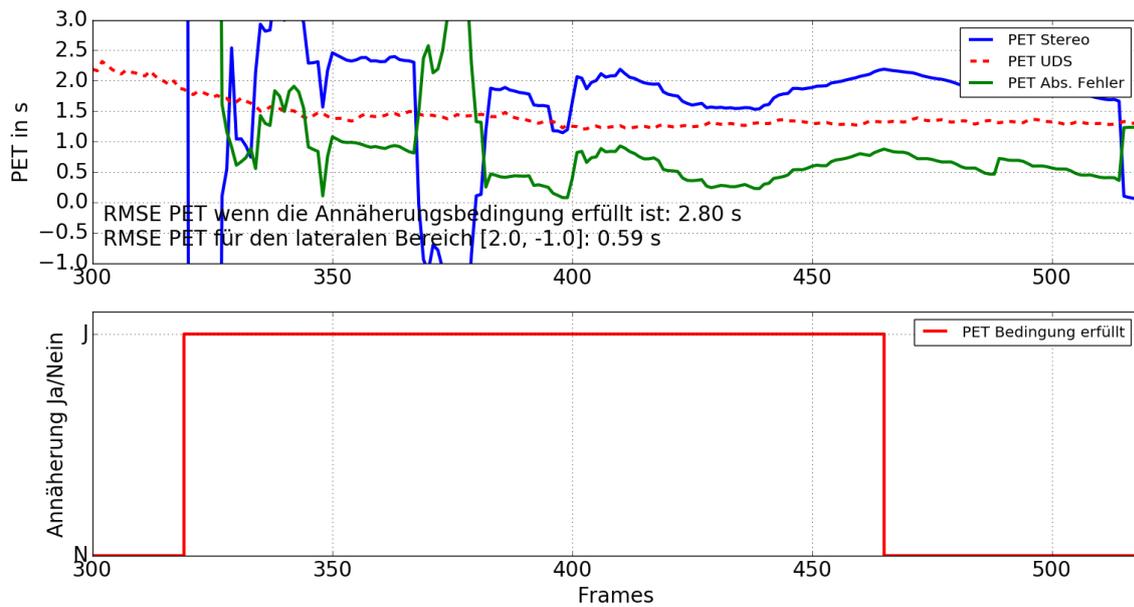
**Abbildung A.219**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 2 berechnet mit dem regelbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.220**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 2 berechnet mit dem modellbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).

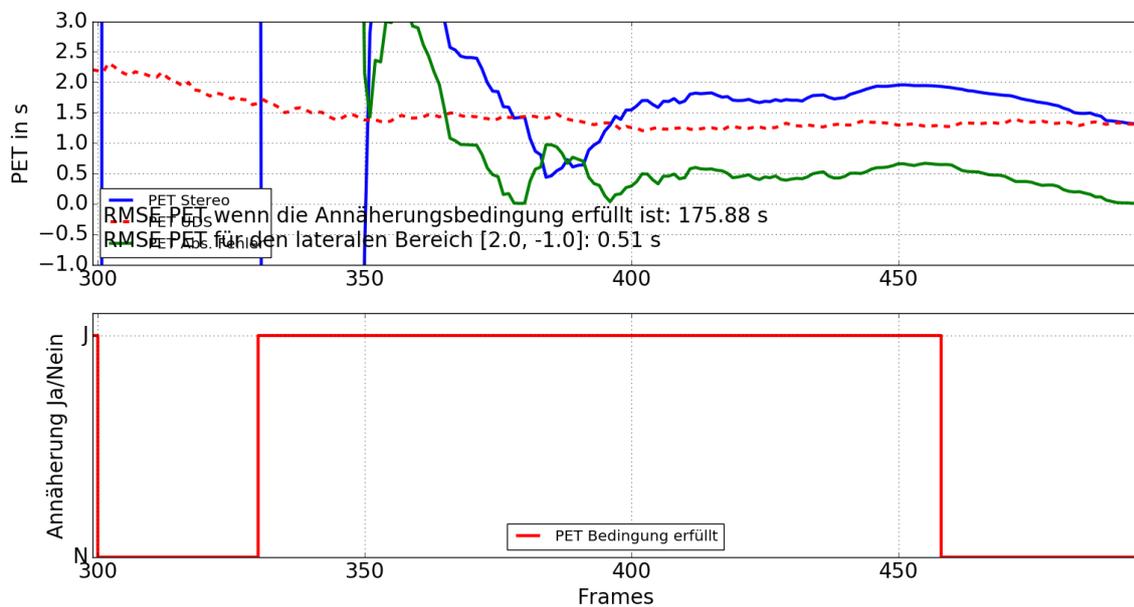


**Abbildung A.221**

PET und Grundbedingung des Objekts für den PET Versuch 2 berechnet mit dem regelbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.

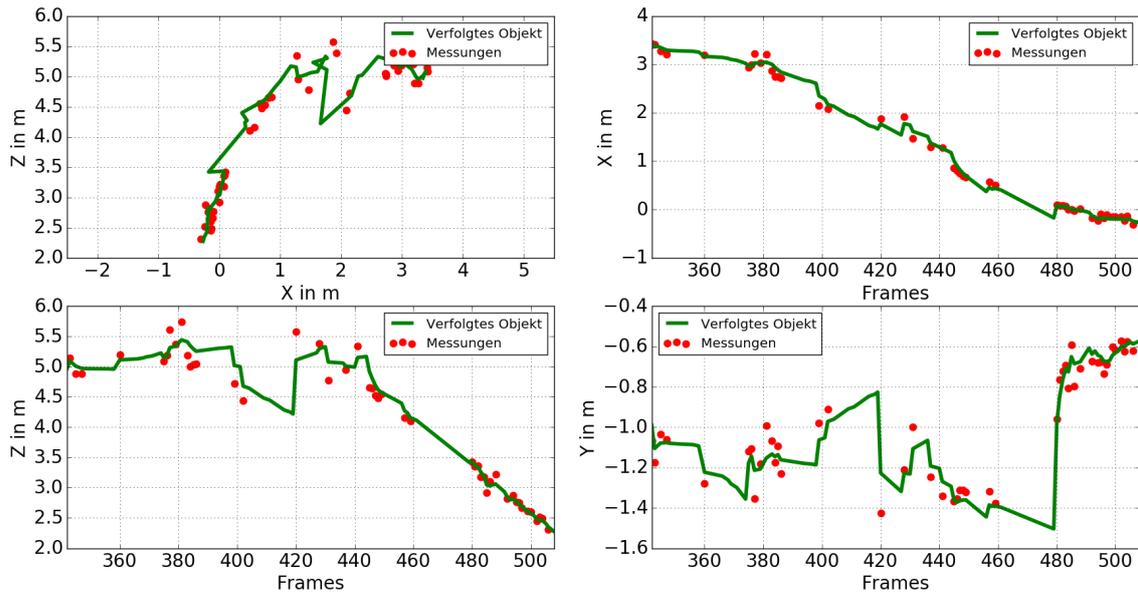


**Abbildung A.222**

PET und Grundbedingung des Objekts für den PET Versuch 2 berechnet mit dem modellbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.



**Abbildung A.223**

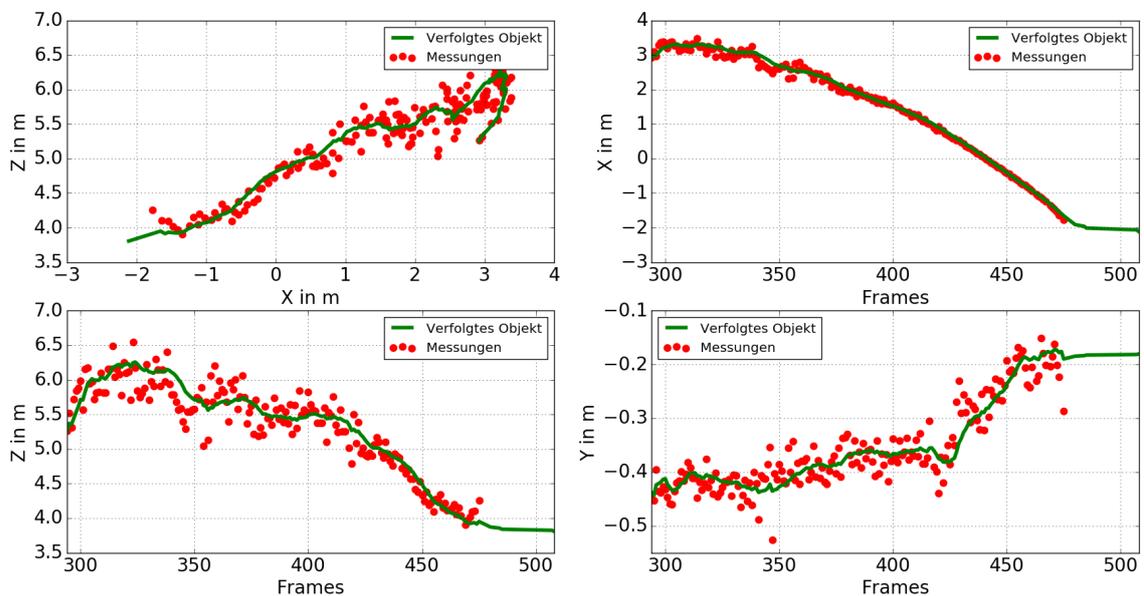
Trajektorie des Objekts für den PET Versuch 3, berechnet mit dem regelbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.224**

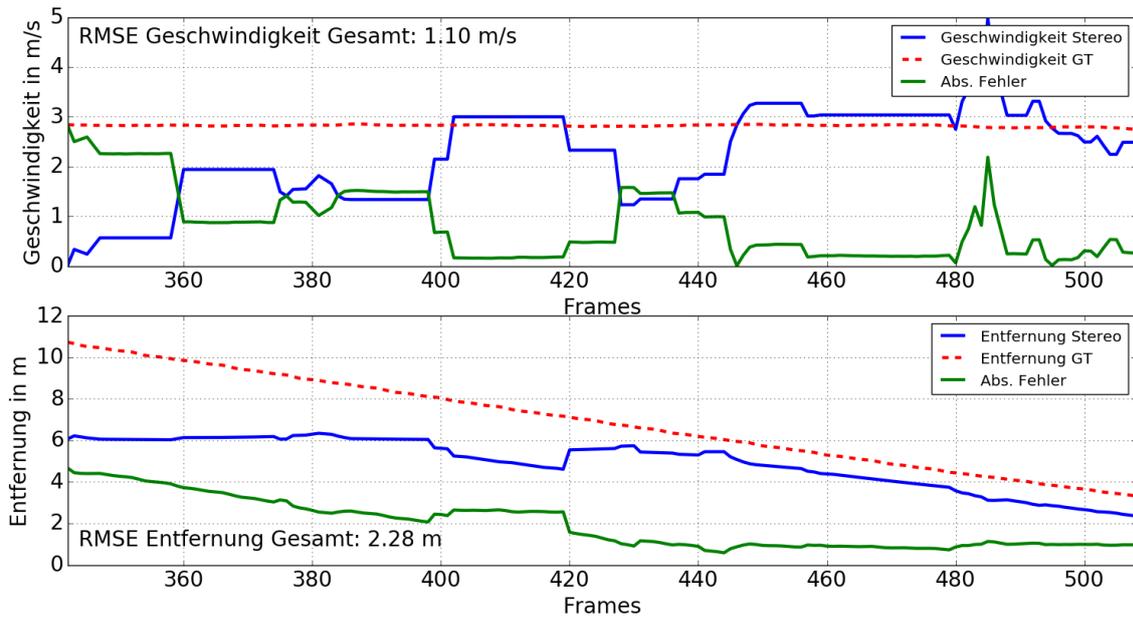
Trajektorie des Objekts für den PET Versuch 3, berechnet mit dem modellbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

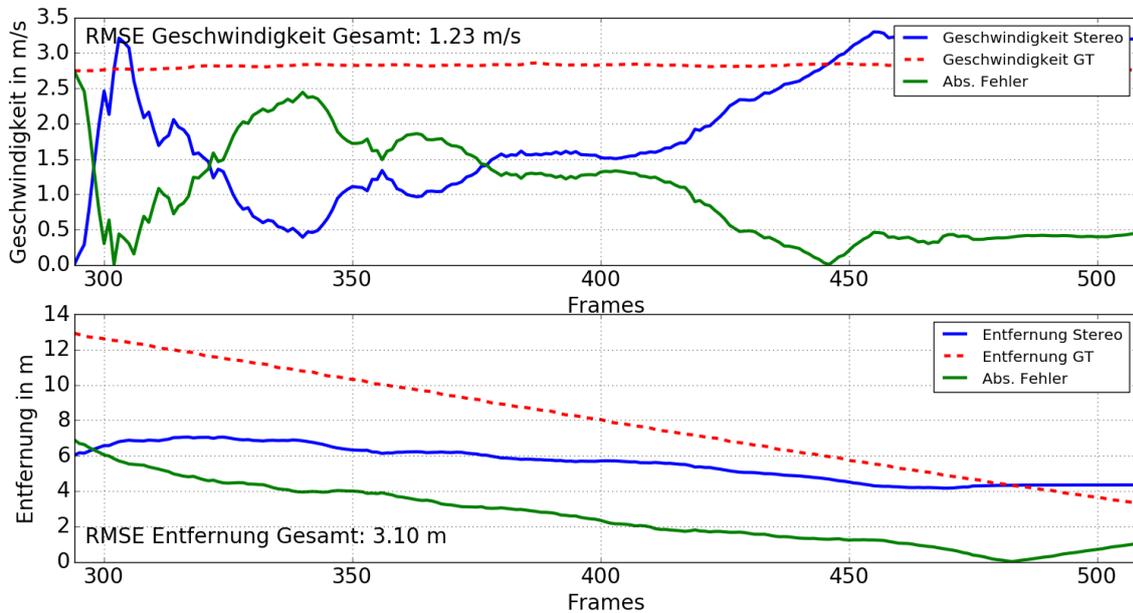
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



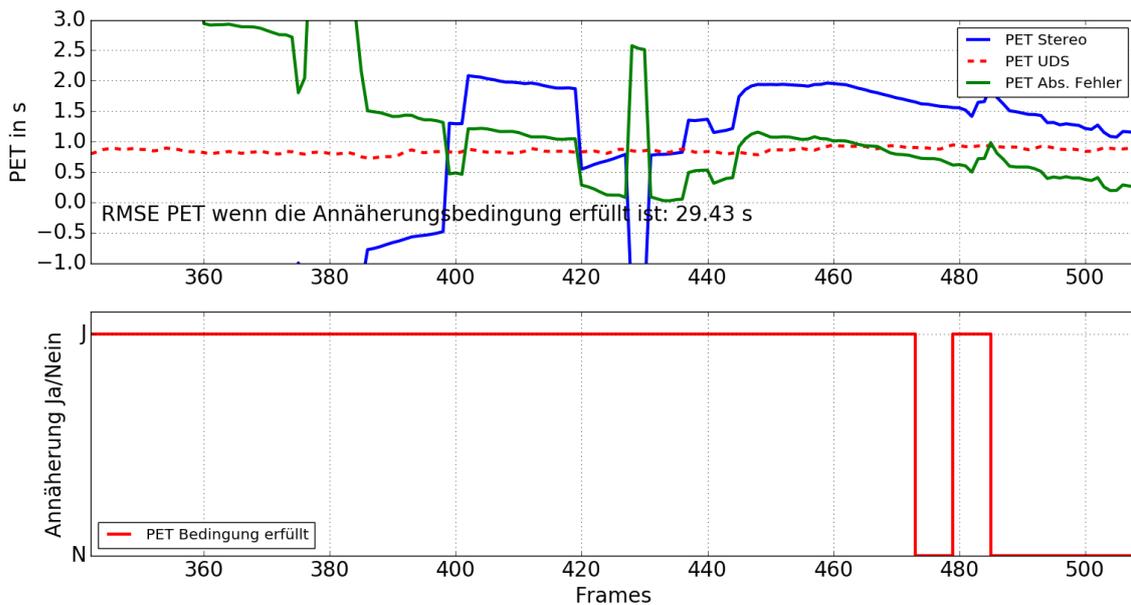
**Abbildung A.225**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 3 berechnet mit dem regelbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.226**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 3 berechnet mit dem modellbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).

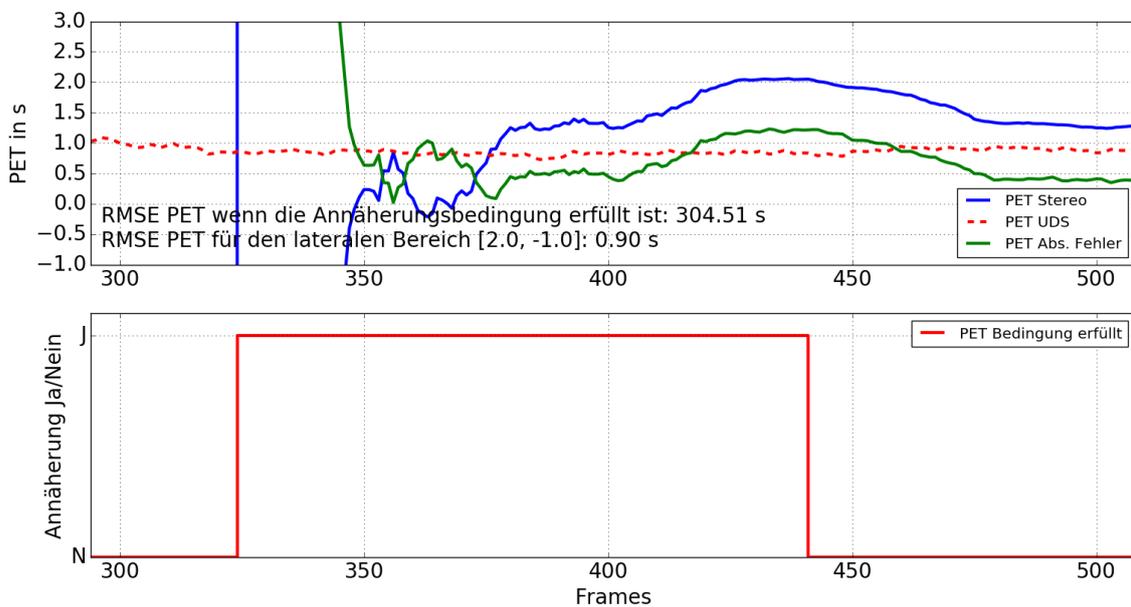


**Abbildung A.227**

PET und Grundbedingung des Objekts für den PET Versuch 3 berechnet mit dem regelbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.

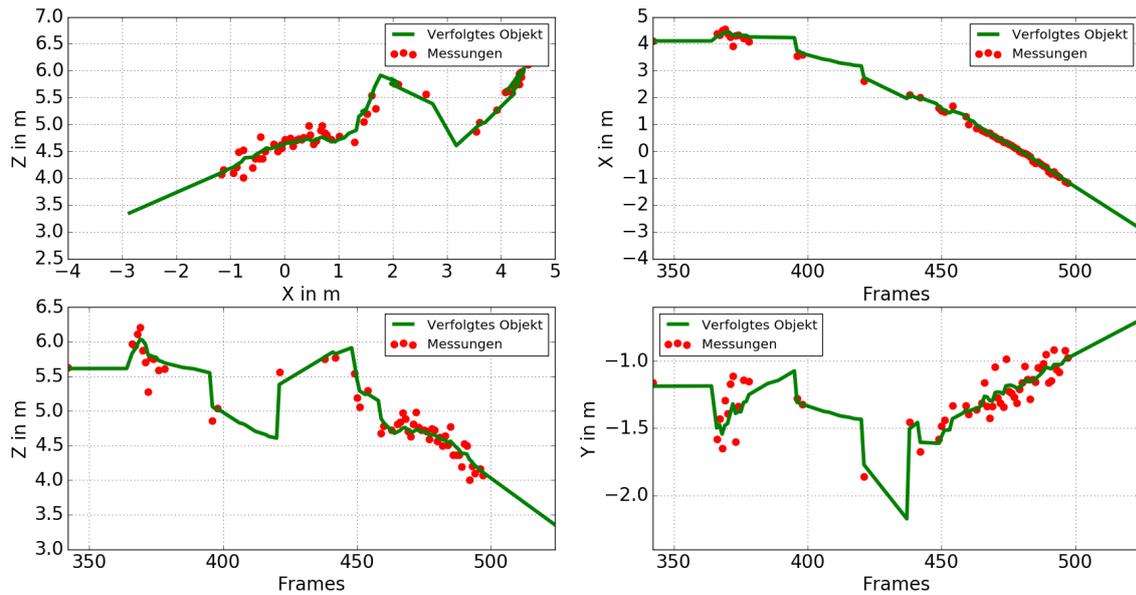


**Abbildung A.228**

PET und Grundbedingung des Objekts für den PET Versuch 3 berechnet mit dem modellbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.



**Abbildung A.229**

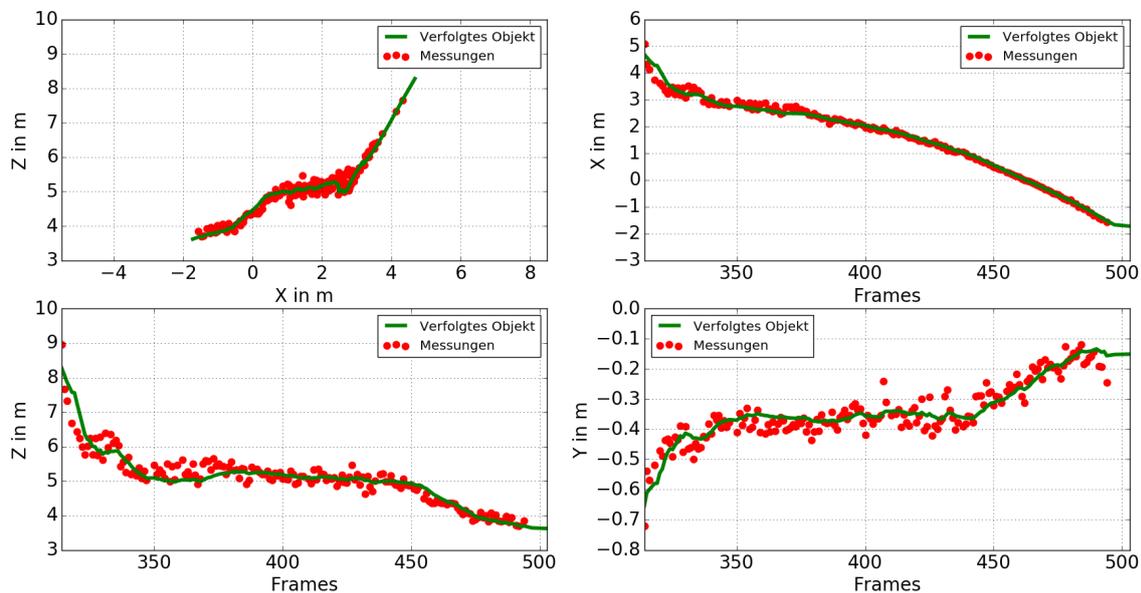
Trajektorie des Objekts für den PET Versuch 4, berechnet mit dem regelbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.230**

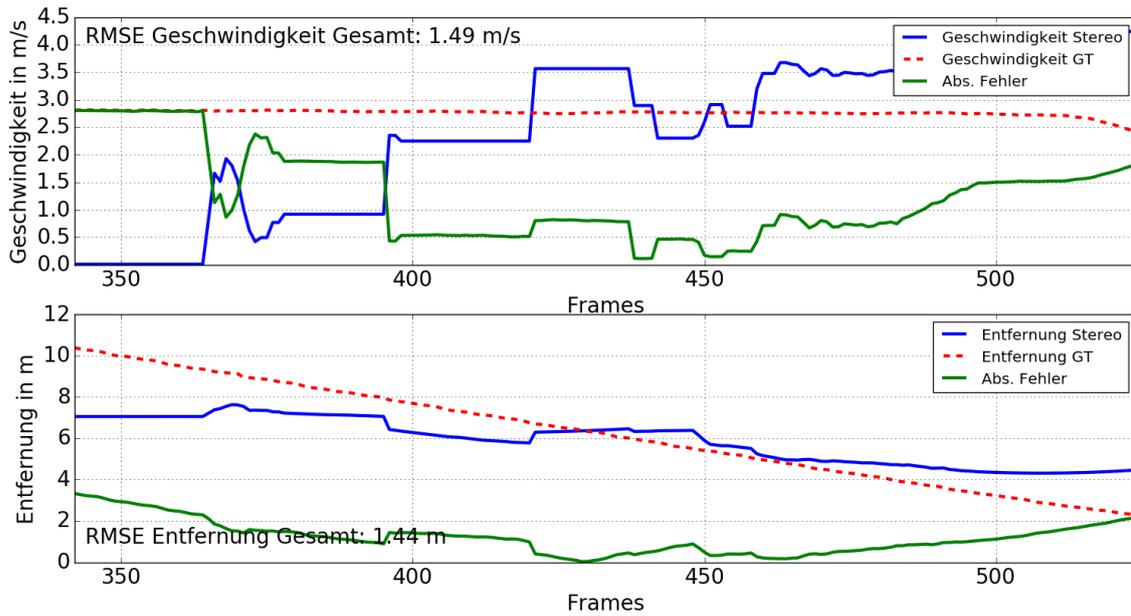
Trajektorie des Objekts für den PET Versuch 4, berechnet mit dem modellbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

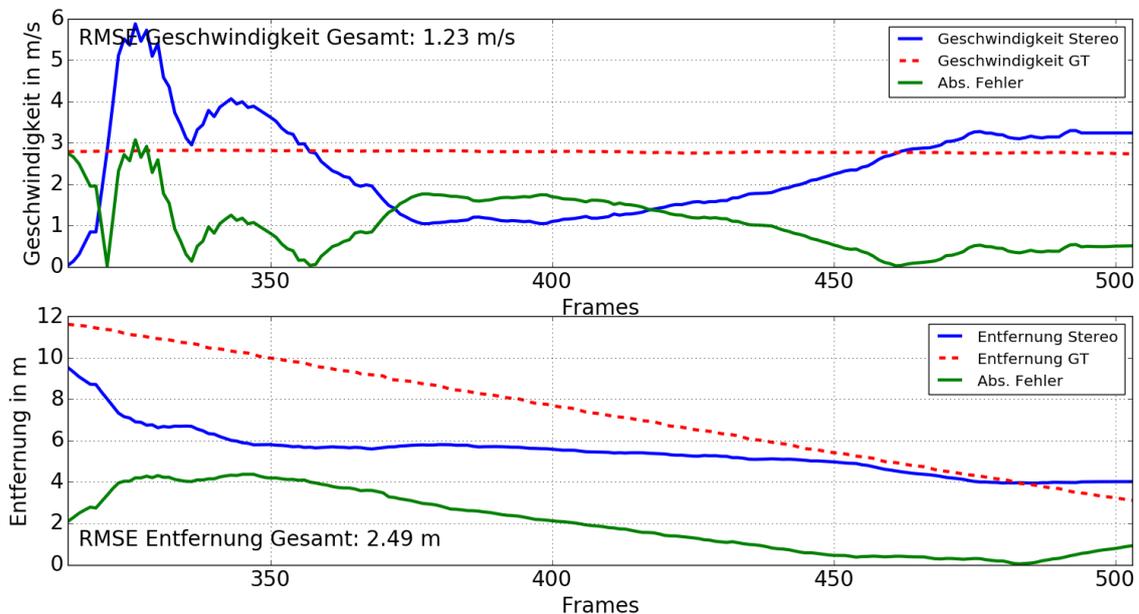
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



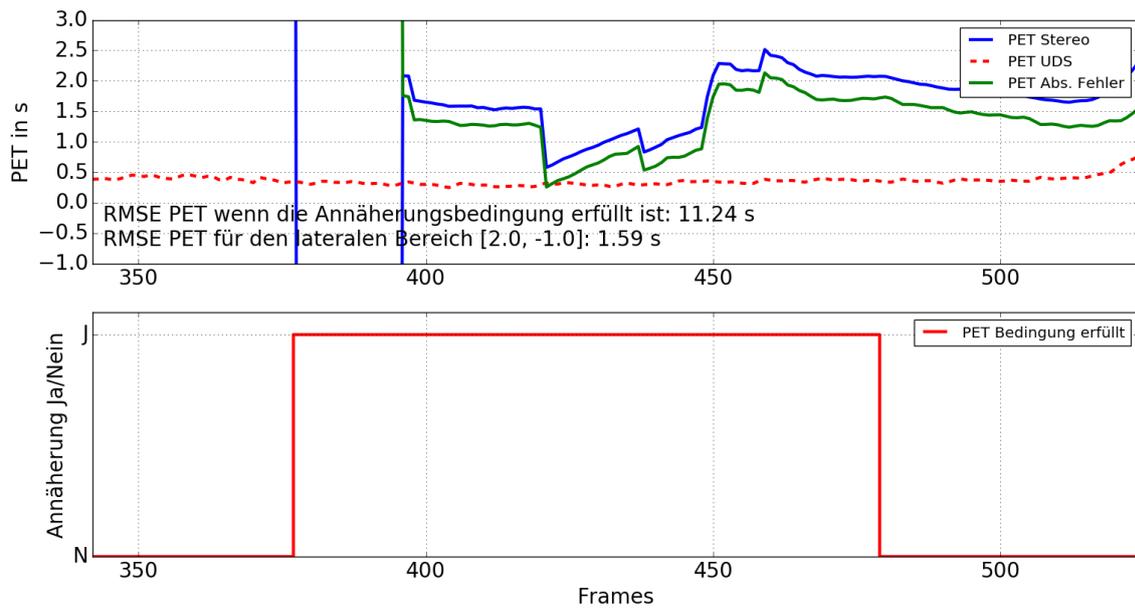
**Abbildung A.231**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 4 berechnet mit dem regelbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.232**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 4 berechnet mit dem modellbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).

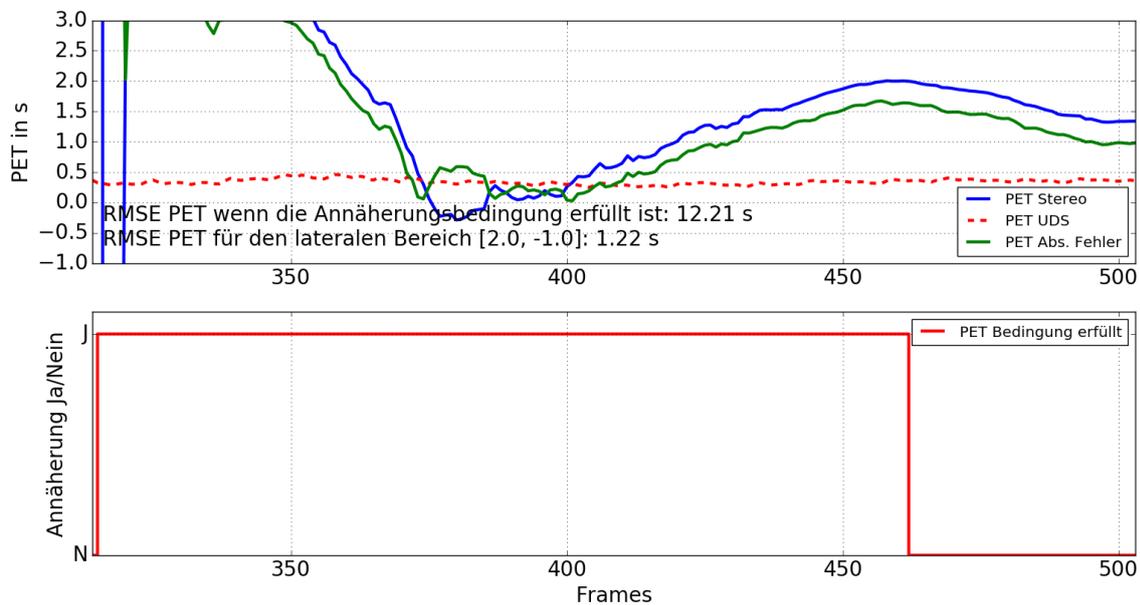


**Abbildung A.233**

PET und Grundbedingung des Objekts für den PET Versuch 4 berechnet mit dem regelbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.

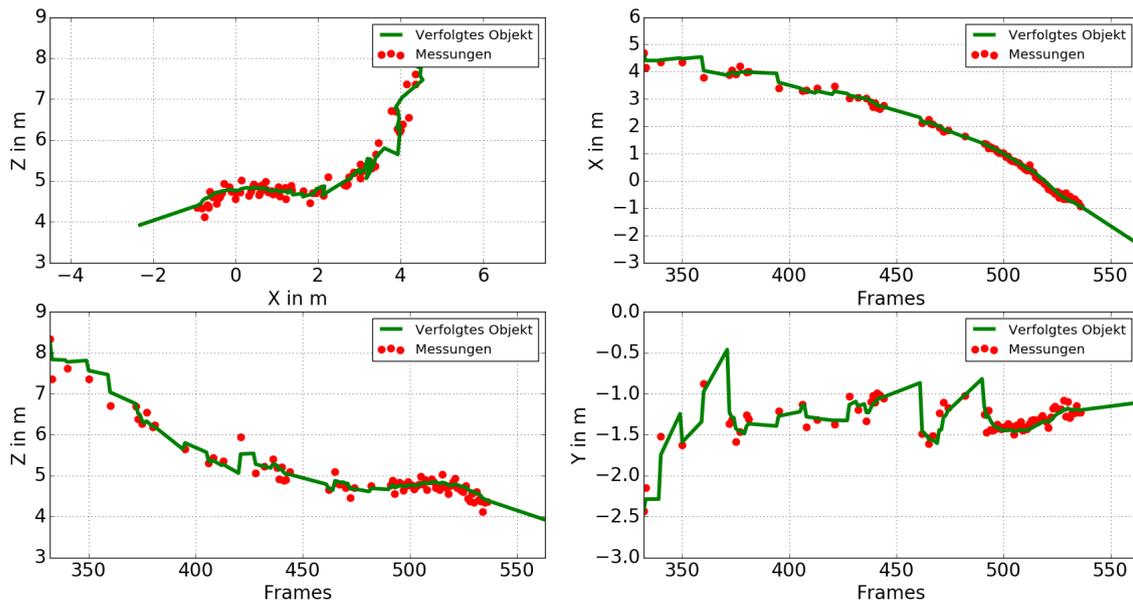


**Abbildung A.234**

PET und Grundbedingung des Objekts für den PET Versuch 4 berechnet mit dem modellbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.



**Abbildung A.235**

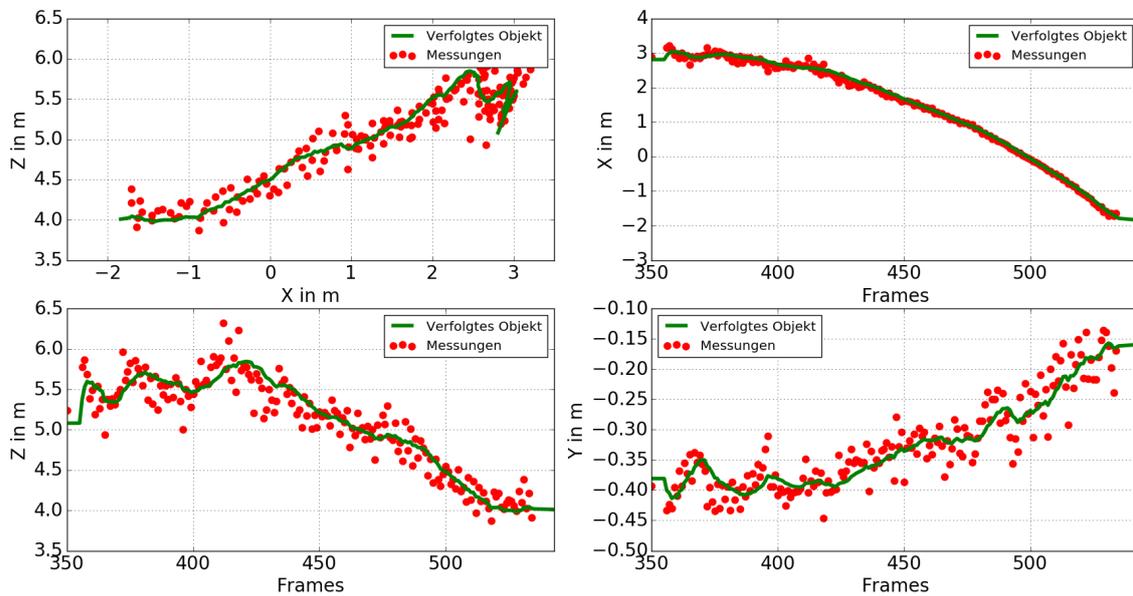
Trajektorie des Objekts für den PET Versuch 5, berechnet mit dem regelbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



**Abbildung A.236**

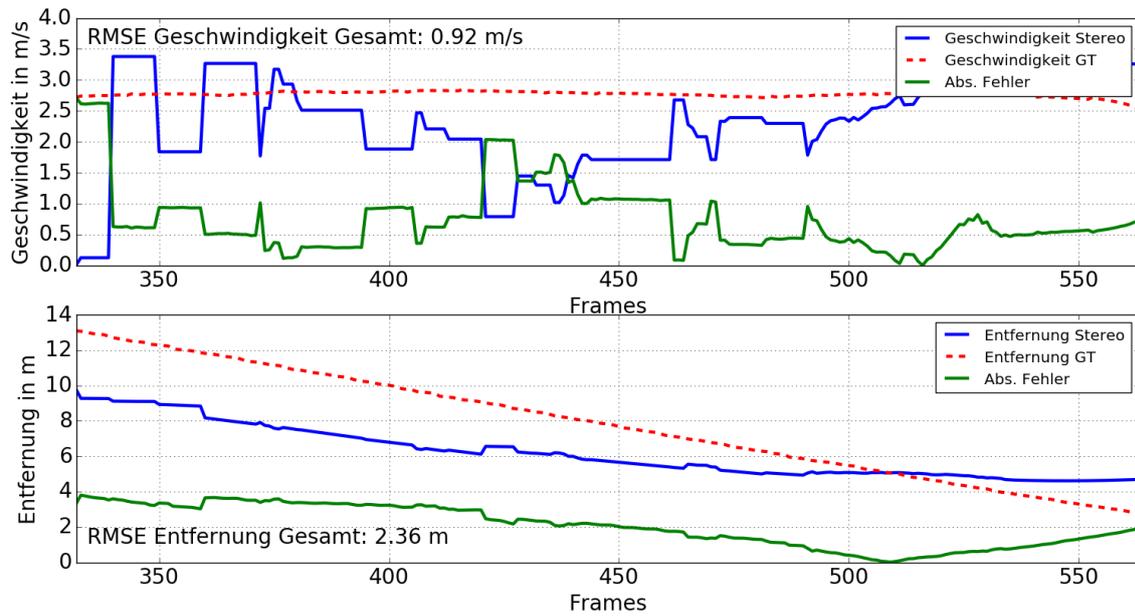
Trajektorie des Objekts für den PET Versuch 5, berechnet mit dem modellbasierten Ansatz. Das Signal des *Trackers* wird als grüne Linie dargestellt. Zum Update verwendete Messwerte sind als roter Punkt eingezeichnet.

Links Oben: Trajektorie in der horizontalen X-Z-Ebene.

Rechts Oben: Position in X-Richtung über die Kameraframes dargestellt.

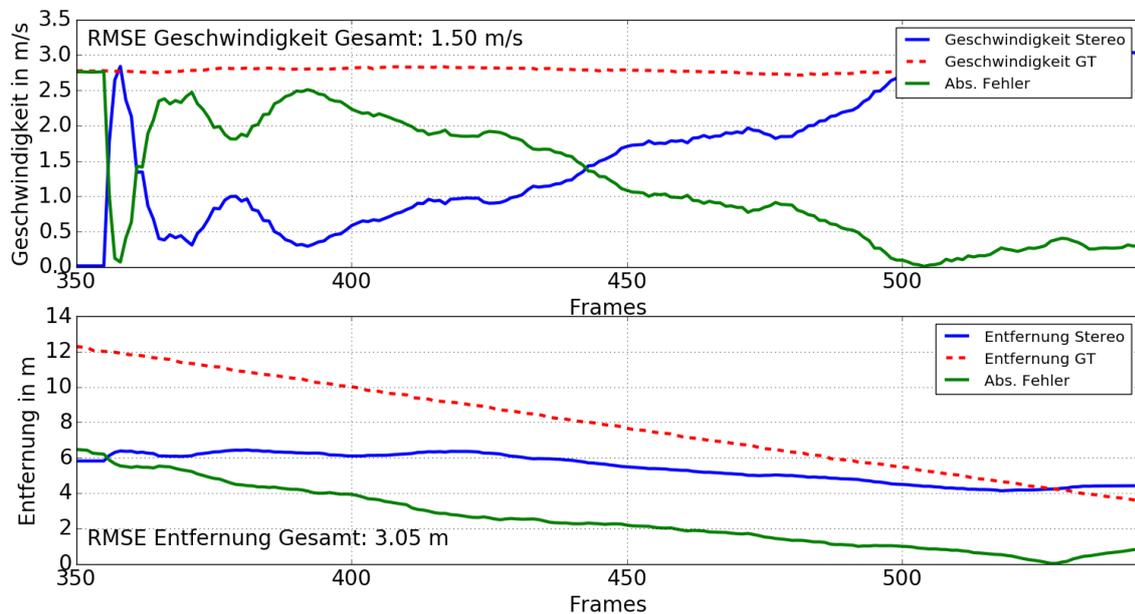
Links Unten: Position in Z-Richtung über die Kameraframes dargestellt.

Rechts Unten: Position in Y-Richtung über die Kameraframes dargestellt.



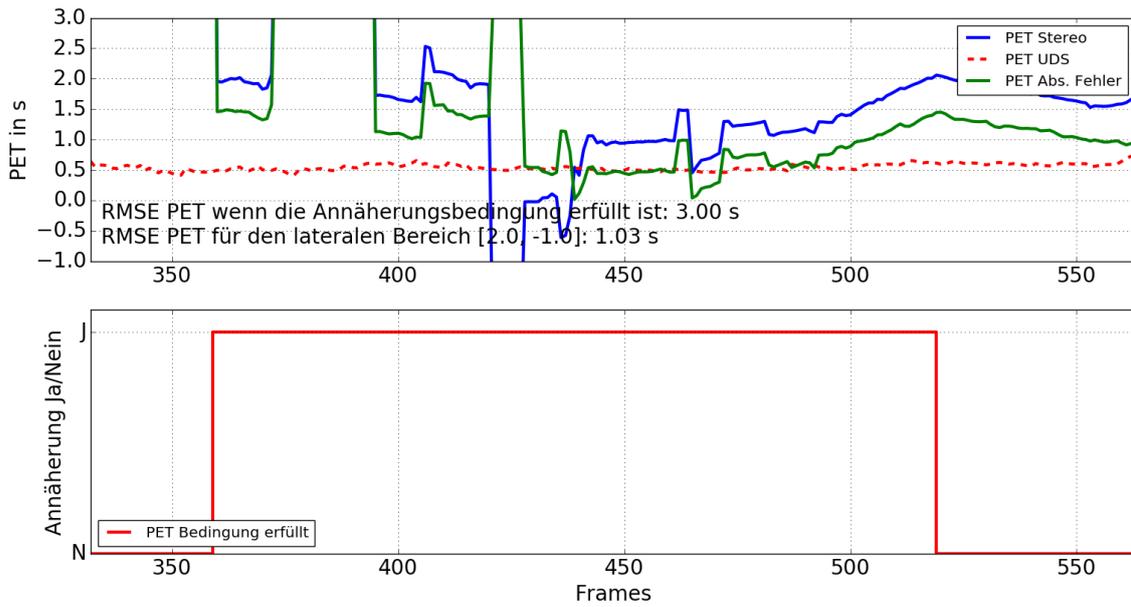
**Abbildung A.237**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 5 berechnet mit dem regelbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).



**Abbildung A.238**

Geschwindigkeit (oben) und Entfernung (unten) des Objekts für den PET Versuch 5 berechnet mit dem modellbasierten Ansatz. Die aus UDS und Correvit berechnete Grundwahrheit wird angegeben (Rot) und der absolute Fehler berechnet (Grün).

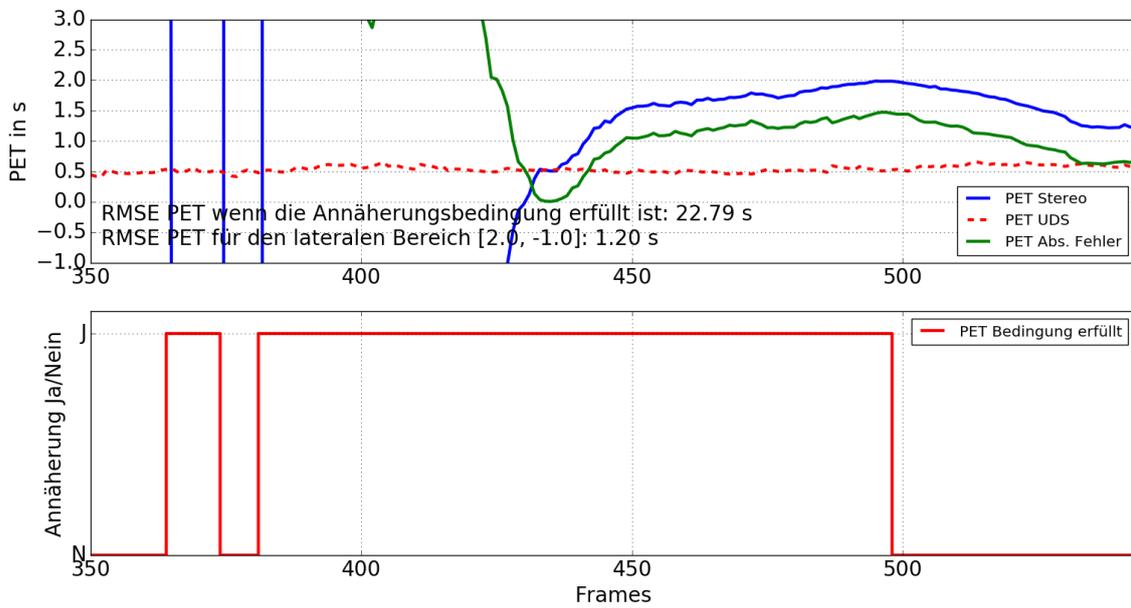


**Abbildung A.239**

PET und Grundbedingung des Objekts für den PET Versuch 5 berechnet mit dem regelbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.



**Abbildung A.240**

PET und Grundbedingung des Objekts für den PET Versuch 5 berechnet mit dem modellbasierten Ansatz. Für die PET wird aus UDS und Correvit die Grundwahrheit berechnet und angegeben (Rot) und der absolute Fehler berechnet (Grün).

Oben: PET des Objekts.

Unten: Grundbedingung zur Berechnung der PET des Objekts.