

Hochschule für Technik, Wirtschaft und Kultur Leipzig

Fakultät Informatik, Mathematik und Naturwissenschaften

Masterstudiengang Informatik, 15 INM

Masterarbeit

zur Erlangung des akademischen Grades

Master of Science (M.Sc.)

**3D-Objekt-Verfolgung mit Stixel-Darstellung in autonomen
Fahrzeugen**

Eingereicht von: Michael Horn

Matrikelnummer: 66250

Leipzig 26. März 2018

Erstgutachter: Prof. Dr. rer. nat. Sibylle Schwarz

Zweitgutachter: Prof. Dr. rer. nat. Klaus Bastian

Eidesstattliche Erklärung

Ich versichere, dass die Masterarbeit mit dem Titel „3D-Objekt-Verfolgung mit Stixel-Darstellung in autonomen Fahrzeugen“ nicht anderweitig als Prüfungsleistung verwendet wurde und diese Masterarbeit noch nicht veröffentlicht worden ist. Die hier vorgelegte Masterarbeit habe ich selbstständig und ohne fremde Hilfe verfasst. Ich habe keine anderen Quellen und Hilfsmittel als die angegebenen benutzt. Diesen Werken wörtlich oder sinngemäß entnommene Stellen habe ich als solche gekennzeichnet.

Leipzig, 26. März 2018

Unterschrift

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele und Grenzen	2
1.3	Aufbau der Arbeit	3
2	Grundlagen	4
2.1	Aktuelle Entwicklungen im autonomen Fahren hinsichtlich Detekti- on und Verfolgung von Objekten	4
2.2	Intel RealSense R200	5
2.3	Digitale Bildverarbeitung	8
2.3.1	Tiefenbilder	9
2.3.2	Verfahren zur Detektion von Objekten	11
2.4	Positionierung	14
2.5	Überblick	15
3	Konzept	16
3.1	Präzisierung der Ausgangssituation	17
3.2	Konkretisierung der Zielstellung	18
3.3	Wahl der Verfahren	20
3.3.1	Detektion	20
3.3.2	Klassifikation	21
3.3.3	Verfolgung	21
3.4	Feinkonzept	22
3.5	Überblick	24

Inhaltsverzeichnis

4	Bilddatengewinnung	25
4.1	Disparitätsbild	25
4.2	Parameter	27
4.3	Überblick	27
5	Objekt-Detektion	28
5.1	Stixel-Extraktion	28
5.1.1	Anpassung der Parameter	30
5.1.2	Bewertung der Ergebnisse	30
5.1.3	Überblick	32
5.2	Objekt-Extraktion	33
5.2.1	Extraktion von Objektregionen	33
5.2.2	Berechnung zugehöriger Bereiche im Farbbild	35
5.2.3	Berechnung der Objektposition auf der Weltkarte	36
5.2.4	Überblick	39
6	Vergleich Stixel- und Punktwolken-Ansatz	40
6.1	Prinzipielle Funktionsweise des Punktwolken-Ansatzes	40
6.2	Vergleich	41
6.2.1	Detektion und Extraktion	41
6.2.2	Positionierung	41
6.3	Betrachtung der Anforderungen	42
6.4	Überblick	44
7	Objekt-Verfolgung im Weltmodell	45
7.1	Planung	45
7.1.1	Konkretisierung der Anforderungen	45
7.1.2	Ansatz für die Realisierung	46
7.1.3	Prinzipieller Ablauf	47
7.1.4	Vor- und Nachteile	48
7.2	Realisierung	49
7.2.1	Vorfilterung	50
7.2.2	Objekt-Klassifikation	50
7.2.3	Objekt-Verfolgung	51

Inhaltsverzeichnis

7.2.4	Weiterverarbeitung	53
7.3	Grenzen des Verfahrens	54
7.4	Überblick	55
8	Schlussbetrachtung	56
8.1	Ausblick	58
8.2	Persönliches Fazit	58
A	Anhang	I
A.1	Parameter	I
A.1.1	Kamera	I
A.1.2	Stixel	II
A.2	Aufnahmen	II
A.2.1	Ausgangsdaten	II
A.2.2	Stixel-Extraktion	III
A.2.3	Objekt-Detektion	III
A.3	Daten des Pylon-Experiments	III
A.4	Objekt-Verfolgung	IV
A.5	Quellcode	IV
A.6	Konfiguration in ADTF	V
	Glossar	VI
	Literaturverzeichnis	X

Abbildungsverzeichnis

2.2.1 Schematischer Aufbau der Intel RealSense R200	6
2.2.2 Farb- und Infrarotbild der Intel RealSense R200	6
2.2.3 Aufbau des Stereo-Systems der Intel RealSense R200	7
2.3.1 Beispiel für Punktwolke und zugehöriges Disparitätsbild	11
2.3.2 Beispiel für farbkodierte Stixel	12
2.4.1 Schild, Weltkarte und Orientierung im AADC 2017	14
3.2.1 Grobkonzept der Teilaufgaben	19
3.4.1 Vollständiges Konzept der Teilaufgaben	23
4.1.1 Erzeugung des Disparitätsbildes aus linkem und rechtem Infrarotbild	25
4.1.2 Vergleich Disparitätsbilder	26
5.1.1 Infrarotbild und Disparitätsbild einer Szene	29
5.1.2 Stixel mit Originalparametern	29
5.1.3 Stixel mit angepassten Parametern	29
5.1.4 Stixel mit Korrektur	31
5.2.1 Ergebnis der Objekt-Extraktion	34
5.2.2 Relevante Bereiche im Farbbild	36
5.2.3 Verhältnis von Fahrzeugabstand zur Positionsgenauigkeit	38
5.2.4 Fehlerhafte Extraktion von Stixeln im Nahbereich	39
7.1.1 Darstellung der RCC-8 Regeln für Menge X und Y	46
7.1.2 Zu betrachtende Relationen zwischen zwei Mengen A und B	47
7.2.1 Relation zwischen zwei Objekten anhand der Distanz und Radien .	50
7.2.2 Ergebnis einer angewendeten Klassifizierung	51
7.2.3 Prinzipieller Ablauf der Objektverfolgung	53

Tabellenverzeichnis

3.3.1 Vor- und Nachteile von Punktwolken und Disparitätsbildern	20
5.2.1 Übersicht der Parameter für die Objektextraktion aus Stixeln	35
6.2.1 Vergleich der Positionierung vom Stixel- und Punktwolken-Ansatz	42
7.1.1 Topologische Fälle für zwei Mengen	47
7.1.2 Vor- und Nachteile der Objekt-Verfolgung mittels Mengen	48

Abstract

Die vorliegende Arbeit setzt sich mit der Detektion von Objekten mittels Stereo-Sehen sowie der Verfolgung von Objekten auf einer zweidimensionalen Karte auf einem Modellauto auseinander. Ziel der Detektion ist es, eine Klassifikation von Objekten mit Hilfe künstlicher Intelligenz auf Teilbereichen einer Szene zu ermöglichen. Die Klassifikation wird im Rahmen dieser Arbeit nicht betrachtet.

Zunächst werden die Grundlagen sowie die Ausgangssituation dargestellt. Im Anschluss wird ein Konzept erstellt, welches die gegebene Problemstellung löst. Die Detektion erfolgt zunächst mit der abstrakten Datenstruktur „Stixel“. Dazu erfolgt eine Anpassung der Parameter für das Modellauto. Im Anschluss erfolgt die Extraktion von Objekten aus dieser Datenstruktur. Jedes extrahierte Objekt besitzt eine Position und einen Radius auf einer zweidimensionalen Weltkarte sowie einen Bereich auf einem Farbbild. Es folgt anschließend ein Vergleich der Detektion von Objekten mittels Stixel-Ansatz und Punktwolken-Ansatz.

Im Anschluss an die Extraktion und Klassifikation von Objekten, wird ein Verfahren vorgestellt, welches Objekte auf einer zweidimensionalen Karte nachverfolgen kann. Dazu wird die Idee des Mengen-Ansatz „RCC-8“ verwendet.

Schlagwörter: Stereo-Sehen, Autonomes Fahren, Modellfahrzeug, Stixel, RCC-8, Objekt-Detektion, Objekt-Verfolgung

1 Einleitung

Ein aktuelles Thema in der Automobilbranche und Forschung ist das *autonome* Fahren. *Autonom* bedeutet soviel wie eigenverantwortlich oder selbständig. Ein Fahrzeug soll also eigenverantwortlich ohne Zutun eines Fahrzeugführers im Straßenverkehr agieren können. Noch fahren Fahrzeuge im Straßenverkehr nicht vollständig autonom, sondern bestimmte Funktionen unterstützen den Fahrer als Hilfsmittel. In Zukunft soll sich dies ändern und Autos sollen vollständig autonom fahren. Darum forschen viele Automobilhersteller wie beispielsweise Tesla, Volkswagen, BMW und Audi daran. Aber auch andere Unternehmen wie Google oder Apple betreiben Forschungen in diese Richtungen.

Auch das Smart Driving Team der **H**ochschule für **T**echnik, **W**irtschaft und **K**ultur (HTWK) in Leipzig forscht im Bereich des autonomen Fahrens und nimmt beispielsweise am **A**udi **A**utonomous **D**riving **C**up (AADC) teil. Dies ist ein von Audi ausgetragener Wettbewerb zum Thema *Autonomes Fahren*. Ziel des Wettbewerbes ist es, ein vorgegebenes Modellfahrzeug so zu programmieren, dass es eine vorgegebene Strecke vollständig autonom entlang fahren kann. Insbesondere wird auf das Einhalten der Straßenverkehrsordnung geachtet.

Die vorliegende Arbeit entstand im Rahmen der Teilnahme am Audi Autonomous Driving Cup 2017 als Teammitglied des Teams HTWK Smart Driving.

1.1 Motivation

Ein großes und aktuelles Forschungsgebiet im autonomen Fahren ist die Erfassung der Umgebung. Dabei geht es nicht nur um die Erkennung von Verkehrsschildern

1 Einleitung

oder Fahrzeugen, sondern auch von Menschen und Gefahrensituationen.

Unter dem Motto „**K**ünstliche **I**ntelligenz“ (KI) wurde im AADC 2017 insbesondere auf den Umgang mit Personen Wert gelegt. Dazu kann ein Beispiel aus dem Wettbewerb betrachtet werden. Ein autonom fahrendes Fahrzeug nähert sich einem Zebrastreifen und muss sich korrekt verhalten. Nun stellen sich für die Fahrzeugsteuereinheit folgende Fragen:

- Ist ein Objekt am Zebrastreifen?
- Ist es eine Person?
- Wird sich die Person über den Zebrastreifen bewegen oder tut Sie das bereits?
- Kann die Fahrt fortgesetzt werden?

Für einen Menschen ist dieses Problem einfach, denn ein Mensch sieht, analysiert und reagiert binnen von Millisekunden auf die Situation. Aus Prinzip kann ein autonomes Fahrzeug und die damit verbundene Steuereinheit nicht ohne weiteres verstehen was durch eine Kamera erfasst wird. Hier kommt der Begriff Künstliche Intelligenz ins Spiel. Mit KI wird versucht, eine dem Menschen ähnliche Intelligenz nachzubilden. Im Rahmen des AADC ist es die Aufgabe mittels KI Objekte zu klassifizieren, also zu entscheiden ob ein Objekt zur Klasse der Erwachsenen, Kinder, Autos oder Verkehrsschilder gehört und entsprechend zu reagieren. Beispielsweise soll bei der Detektion eines Kindes am Straßenrand das Fahrzeug langsam am Kind vorbei fahren. Begründet ist dies damit, dass Kinder unberechenbar reagieren können, wenn sie am Straßenrand spielen.

1.2 Ziele und Grenzen

Damit Objekte mittels KI klassifizierbar sind, werden Informationen benötigt. Dies können zum Beispiel Bilder von einer Kamera sein. Anhand der Kamerabilder kann festgestellt werden was zu sehen ist. Dies führt zum Begriff der digitalen Bildverarbeitung. Diese Arbeit nutzt die digitale Bildverarbeitung, um Teilbereiche aus Kamerabildern zu extrahieren, welche Kandidaten für zu klassifizierende Objekte

1 Einleitung

sind. Diese Bereiche werden klassifiziert. Die eigentliche Klassifikation der Objekte ist nicht Bestandteil dieser Arbeit. Diese Aufgabe wird durch ein anderes Mitglied des Teams realisiert. Im Beispiel des Zebrastreifens bewegte sich eine Person über die Straße. Daher wird im Anschluss an die Klassifikation eine Möglichkeit betrachtet wie klassifizierte Objekt über die Zeit verfolgt werden können. Nachdem Objekte detektiert, klassifiziert und nachverfolgt wurden, können die gewonnenen Informationen weiterverarbeitet werden. Die Weiterverarbeitung ist nicht Bestandteil dieser Arbeit.

1.3 Aufbau der Arbeit

Im Zuge dieser Arbeit wird zunächst im Kapitel 2 die benötigten Grundlagen für diese Arbeit vorgestellt. Dazu gehört die Betrachtung der Stereo-Kamera sowie die Verfahren der digitalen Bildverarbeitung. Anschließend wird im Kapitel 3 die Ausgangssituation sowie die Zielsetzung dieser Arbeit konkretisiert. Es werden Verfahren zur Lösung des Problems der Objekt-Verfolgung und Detektion gewählt sowie ein Konzept entworfen. Kapitel 4 stellt die Gewinnung von Kameraparametern und die Erzeugung von Disparitätsbildern vor. Auf Basis von erzeugten Bildern wird in Kapitel 5 zunächst die Detektion von Objekten mittels einer abstrakten Datenstruktur realisiert. Aus dieser Datenstruktur erfolgt die Extraktion von Objekten sowie die Berechnung der Position von Objekten auf der Weltkarte. Kapitel 6 vergleicht die Ergebnisse der Extraktion mit einem anderen Verfahren. Außerdem werden die erfüllten Anforderungen betrachtet. Im Anschluss wird im Kapitel 7 ein Verfahren zur Verfolgung von Objekten auf der Weltkarte vorgestellt. Das letzte Kapitel liefert eine Zusammenfassung dieser Arbeit.

2 Grundlagen

Im Rahmen des AADC 2017 wurden komplette Modellfahrzeuge im Maßstab 1 zu 8 vorgegeben. Diese Fahrzeuge bestehen aus Standardcomputerkomponenten, welche mit Hilfe von Mikrocontrollern um Sensorik erweitert wurden. Für die Aufgabe der Objekt-Detektion und -Verfolgung sind insbesondere die 3D-Kamera sowie die Position des Fahrzeuges von Bedeutung. Neben der Hardware werden auch Softwaremodule vorgegeben.

Dieser Abschnitt betrachtet kurz die relevanten Komponenten sowie die daraus resultierenden Grundlagen. Zunächst wird die aktuelle Entwicklung im Bereich der Detektion und Verfolgung von Objekten betrachtet. Dann wird auf die Kamera eingegangen, welche im Anschluss zum Begriff der digitalen Bildverarbeitung führt. Der letzte Abschnitt stellt die Positionierung des Fahrzeuges vor.

2.1 Aktuelle Entwicklungen im autonomen Fahren hinsichtlich Detektion und Verfolgung von Objekten

Eines der großen Probleme im autonomen Fahren ist die Erfassung der Umgebung und damit die Detektion sowie Verfolgung von Objekten. Objekte sind vor allem mögliche Hindernisse wie beispielsweise andere Fahrzeuge, aber auch Verkehrsschilder. Die Detektion ist notwendig, damit das Fahrzeug auf Verkehrssituationen richtig reagiert kann und Unfälle vermieden werden.

2 Grundlagen

Besondere Bedeutung für die Erkennung von Objekten hat der **Light-Detection-And-Ranging-Sensor** (Lidar) gewonnen. Dieser Sensor erzeugt hoch aufgelöste 3D-Daten der Umgebung. Beispielsweise nutzt in [ZT17] die Firma Apple diese 3D-Daten als Eingabe für eine KI zur Objektdetektion. Ein Ansatz ohne KI wird in [Che+16] vorgeschlagen. Eine Alternative zu den Lidar-Sensoren stellen Stereo-Kamera-Systeme dar. Diese können Informationen über die Tiefe der Umgebung liefern. Beispielsweise wird in [BPF09], [PF10] eine Segmentierung auf Basis von Tiefenbildern durchgeführt um die Umgebung zu erfassen.

Sehr neue hochkomplexe Verfahren der KI finden inzwischen selbstständig Objekte in Bildern und klassifizieren diese entsprechend. Dafür sei auf die aktuelle Forschung von Google verwiesen [How+17]. Jedoch muss nach solch einer Klassifikation im autonomen Fahren noch die Objektposition im Raum ermittelt werden.

Im Anschluss an eine mögliche Objekterkennung folgt meist die Verfolgung von Objekten. Oftmals wird diese mit einer Positionsvorhersage kombiniert. In der Regel wird dazu der optische Fluss auf Bilder angewendet. Algorithmen des optischen Flusses extrahieren Merkmale zur Verfolgung aus einem Bild und suchen diese im nachfolgenden Bild. Dadurch können Objekte über die Zeit verfolgt werden. Dieser Ansatz wird oft mit sogenannten Kalman-Filtern kombiniert um eine zuverlässige Vorhersage von Objekten zu realisieren [MRF11].

2.2 Intel RealSense R200

Für die Gewinnung von räumlichen Daten steht die 3D-Kamera Intel RealSense R200 zur Verfügung. Die R200 ist auf dem Fahrzeug parallel zum Boden in Fahr-richtung ausgerichtet. Sie besitzt zwei Infrarot-Kameras, eine Farb-Kamera sowie einen Laseremitter. In *Abbildung 2.2.1* ist der Aufbau der R200 zu sehen [Rea16a, Grafik S. 13].

Mit Hilfe der Kamera können diverse Kamerabilder erzeugt werden. In *Abbildung 2.2.2* sind zwei Beispielaufnahmen zu sehen. Insbesondere ist im Infrarotbild

2 Grundlagen

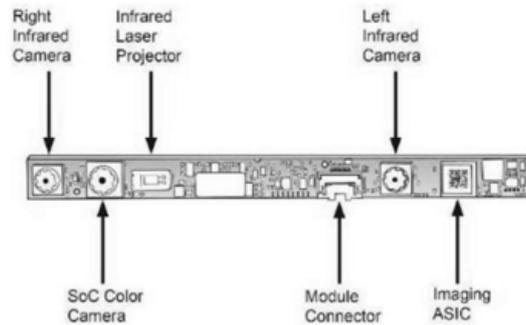


ABBILDUNG 2.2.1: Schematischer Aufbau der Intel RealSense R200

zu sehen, dass der Laseremitter aktiviert ist. Durch den Emitter erhalten Objekte zusätzliche Struktur. Dies verbessert das Berechnen von 3D-Informationen durch die Kamera.

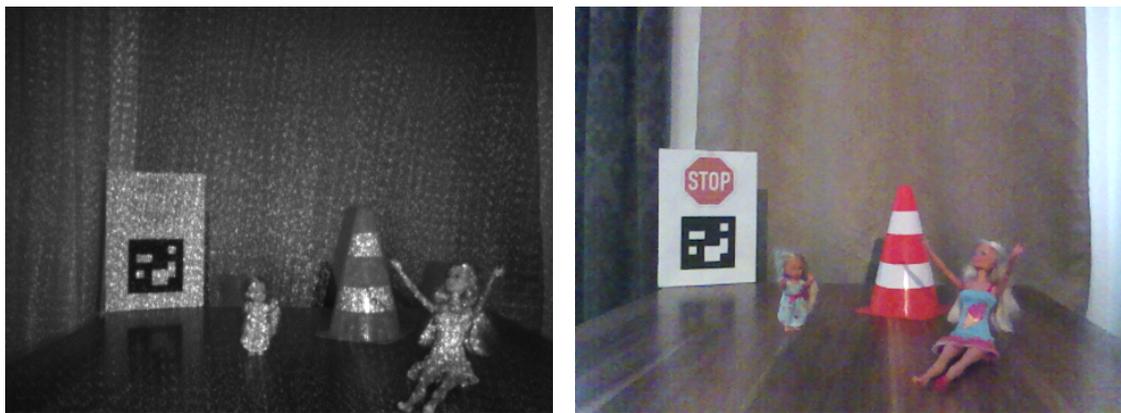


ABBILDUNG 2.2.2: Beispiel Infrarot- (l.) und Farbbild (r.)

Mit Hilfe der zwei identischen Infrarot-Kameras wird ein Stereo-System aufgespannt. Nachfolgend sind wichtige Begriffe im Zusammenhang mit Stereo-Kameras genannt [Rea16c]:

Intrinsische Parameter dienen zur Wiederherstellung des Zusammenhanges zwischen Kamerakoordinatensystem und zugehörigem Kamerabild. Sie hängen weder von der Position noch von der Orientierung der Kamera ab. Diese Parameter definieren und beschreiben die Abbildung zwischen dem Kame-

2 Grundlagen

rabild und der 3D-Kamera sowie die interne Geometrie. Folgende Parameter werden dabei genutzt:

- Brennweite f ist der Abstand der Linse vom Brennpunkt
- Optisches Zentrum der Kamera c

Extrinsische Parameter definieren den Zusammenhang zwischen 3D-Welt- und 3D-Kamerakoordinatensystem und somit die Position und Ausrichtung der Kamera im gegebenen Weltkoordinatensystem. Folgende Parameter werden hierbei genutzt:

- Basislinie b ist der Abstand zwischen linker und rechter Infrarot-Kamera
- Höhe h beschreibt die Entfernung der Kamera zum Boden
- Rotations- und Translations-Parameter zwischen linker und rechter Infrarot-Kamera

Die intrinsischen und extrinsischen Parameter werden benötigt, um mittels Stereosehen Punkte im Raum berechnen zu können. Die Kamera kann diese Punkte $(x, y, z) \in \mathbb{R}^3$ selbständig auf Basis des Stereo-Systems berechnen. In Abbildung 2.2.3 ist schematisch der Aufbau des Stereo-Systems der R200 mit Parametern zu sehen [SF07, Grafik S. 457]. Das 3D-Koordinatensystem der Kamera ist in

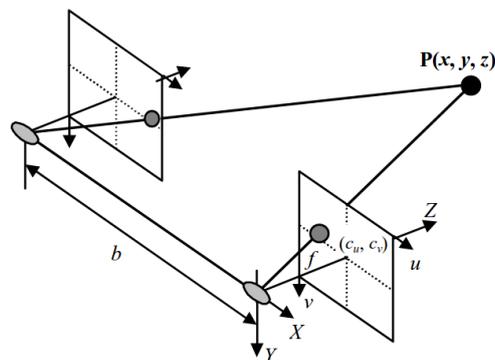


ABBILDUNG 2.2.3: Aufbau des Stereo-Systems der Intel RealSense R200

Metern aufgelöst. Der ideale Bereich für die Tiefenkamera ist mit 0,4 bis 2,8 Meter bei einer Auflösung von 480x360 Pixeln angegeben [Rea16a, S. 9].

2.3 Digitale Bildverarbeitung

Die Verwendung von Kameras führt zur Thematik der digitalen Bildverarbeitung, welche im Folgenden betrachtet wird. Das Ziel der digitalen Bildverarbeitung ist es, abstrakte Informationen aus Bildern zu extrahieren. Im Rahmen des autonomen Fahrens ist das Ziel, mittels Kameras und Bildverarbeitung die Umgebung des Fahrzeuges wahrzunehmen. Beispielsweise müssen Verkehrszeichen erkannt werden, damit das Fahrzeug die Verkehrsregeln befolgen kann. Weiterhin kann mittels des Stereo-Sehens die Entfernung zu einem potenziellen Objekt berechnet werden.

Zunächst muss der Begriff Bild definiert werden: Ein Bild B ist eine Funktion, welche eine Menge von Bildpositionen pos in die Menge der Farben $color$ abbildet.

$$B : pos \rightarrow color \text{ mit } pos, color \subseteq \mathbb{N}^n \quad (2.1)$$

B habe die Breite $cols$ und die Höhe $rows$, dann ist die Menge der möglichen Positionen wie folgt definiert:

$$pos = \{0, \dots, cols - 1\} \times \{0, \dots, rows - 1\} \quad (2.2)$$

Ein Element aus der Menge von Bildpositionen wird in dieser Arbeit durch das Koordinatentupel $(u, v) \in pos$ beschrieben. Je nach Farbraum des Bildes ist die Menge der Farben unterschiedlich definiert. Ein Grauwertbild mit der Farbtiefe von n_{color} -Bit besitzt die Farbmenge beziehungsweise Intensitäten:

$$color_{grau} = \{0, \dots, 2^{n_{color}} - 1\} \quad (2.3)$$

Farbbilder werden in der digitalen Bildverarbeitung durch die Kombination von drei Farbräumen dargestellt. Dies geschieht indem drei graue Farbräume verknüpft werden. Jeweils einem dieser grauen Farbräume entspricht dem Farbraum rot, grün und blau. Durch diese Kombination kann jede beliebige Farbe dargestellt werden.

$$color_{rgb} = \{0, \dots, 2^{n_{color}} - 1\}^3 \quad (2.4)$$

2 Grundlagen

Weiterhin besitzen die Bilder der R200 eine maximale Auflösung mit $rows \times cols$. Diese beträgt bei den Infrarotbildern 640×480 Pixel und für das Farbbild bis zu 1920×1080 Pixel. Im Rahmen dieser Arbeit wird durchgehend mit einer Auflösung von 640×480 für Infrarot- und Farbbilder gearbeitet. Die Infrarotbilder der R200 werden als Grauwertbilder dargestellt.

2.3.1 Tiefenbilder

Wie bereits erwähnt kann die Intel RealSense R200 räumliche Informationen liefern. Generiert werden diese Information aus dem Stereo-System der Infrarot-Kameras und den zugehörigen intrinsischen sowie extrinsischen Parametern. Insbesondere liefert die R200 rektifizierte Bilder. Rektifizierte Bilder sind geometrisch entzerrte und transformierte Bilder [FI11]. Durch eine Rektifizierung werden Bilder eines Stereo-Systems so ausgerichtet, dass eine Bildzeile zwischen linkem und rechtem Bild austauschbar verwendbar ist. Dadurch unterscheiden sich Bildpunkte zwischen linkem und rechtem Infrarotbild nur in den Bildspalten. Dies vereinfacht das Finden von korrespondierenden Bildpunkten. Korrespondierende Bildpunkte beschreiben eine identischen Bildstelle zwischen Bildern aus verschiedenen Blickwinkeln. Ein Punkt im Raum kann mittels Triangulation berechnet werden, wenn diese Bildpunkte und Kameraparameter bekannt sind (vgl. Abbildung 2.2.3, S. 7). Durch die Verwendung von rektifizierten Bildern existiert keine Rotation zwischen Infrarot- und Farb-Kamera bei der R200 [Rea16c].

Die R200 kann die Tiefeninformation in Form von Punktwolken, Tiefen- oder Disparitätsbildern liefern. Eine Punktwolke ist eine Menge von Punkten eines Vektorraumes, die eine unorganisierte räumliche Struktur aufweisen. Ein Tiefenbild codiert für jeden Bildpunkt die Tiefe des Punktes. Ein Disparitätsbild enthält im Gegensatz zum Tiefenbild für jeden Punkt eine Tiefeninformation in Form einer sogenannten Disparität.

Punktwolken

Eine Punktwolke ist ein 3D-Bild, welches mittels Stereo-Verfahren generiert wird. Als Ausgangspunkt für die Erzeugung von Punktwolken dienen Tiefenbilder. Dabei wird im Gegensatz zum Tiefenbild nicht jeder Position des Bildes ein Wert zugewiesen, sondern es werden nur berechneten 3D-Positionen besetzt. Eine zusätzliche Codierung von Farben ist möglich, beispielsweise kann jedem Punkt die Farbe des korrespondierenden Farbbildes zugewiesen werden. In der Abbildung 2.3.1 ist ein Beispiel für eine Punktwolke zu sehen.

Disparitätsbilder

Ein Disparitätsbild ordnet jeder Bildkoordinate eine Disparität d („Unterschiedlichkeit“) mit $d \in \mathbb{R}_{>0}$ zu. Die Disparität ist der Abstand eines korrespondierenden Bildpunktes zwischen linkem und rechtem rektifizierten Infrarotbild und hat folgende Eigenschaften:

- Der Wert ist invers proportional zur Tiefe
- Mittels Disparität und Parameter der Stereo-Kamera kann ein dreidimensionaler Punkt berechnet werden
- Wenn kein korrespondierender Bildpunkt gefunden wurde, so wird ein spezieller Fehlerwert der Disparität zugewiesen

Im Allgemeinen bezieht sich das Disparitätsbild auf das Bildkoordinatensystem des linken Bildes. Dies entspricht auf der R200 dem linkem Infrarotbild. Eine Möglichkeit zur Generierung von Disparitätsbildern aus Stereo-Bildern wird mittels **Semiglobal-Block-Matching-Verfahren** (SGBM) in [Hir08] beschrieben.

Es können folgende Formeln für die Berechnung eines 3D-Punktes (x, y, z) aus einem Disparitätsbild verwendet werden [SF07, S. 456]:

$$x = \frac{b * (u - c_u)}{d}, y = \frac{b * (v - c_v)}{d}, z = \frac{b * f}{d} \quad (2.5)$$

2 Grundlagen

In der Abbildung 2.3.1 ist ein Beispiel zu sehen. Das Bild ist mit Grauwerten wie folgt codiert: Weiß entspricht der größten Disparität, welche in der Distanz kleiner wird. Ferne Objekte sind dunkler dargestellt. Schwarz signalisiert, dass für diese Bildposition zwischen dem linkem und rechtem Bild kein Treffer gefunden wurde.

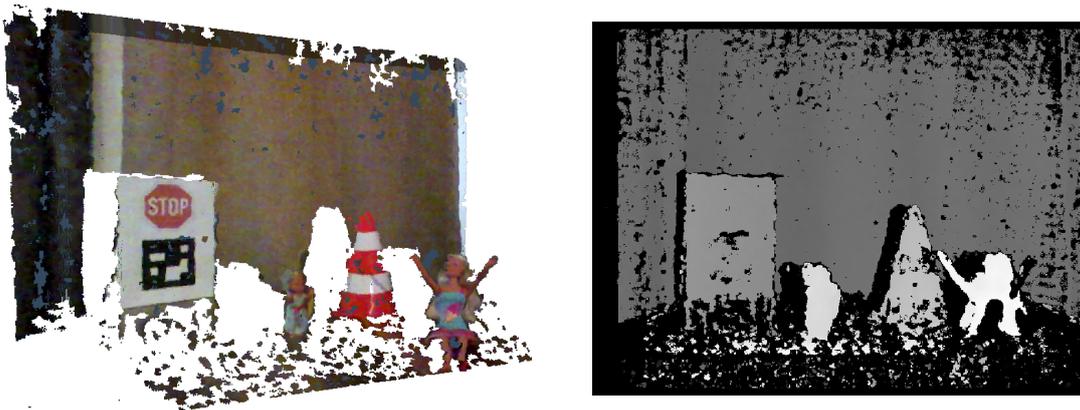


ABBILDUNG 2.3.1: Beispiel für eine Punktwolke (l.) und zugehöriges Disparitätsbild (r.)

2.3.2 Verfahren zur Detektion von Objekten

Eine Möglichkeit zur Objekt-Detektion im autonomen Fahren mit Hilfe von Stereo-Kameras wurde in [BPF09] publiziert. Schrittweise wurde der dort genannte Ansatz weiter verbessert, beispielsweise in [PF10] sowie [Pfe11]. Bei diesem Verfahren werden sogenannte Stixel generiert. Ein Stixel ist eine rechteckige 3D-Datenstruktur, welche einen gewissen Bereich einer Szene beschreibt. Jeder Stixel enthält die relative Position zur Kamera, die Höhe sowie Disparität und damit die Tiefe zu einem potenziellen Objekt. Diese Datenstruktur „steht“ dabei auf dem Boden. Weiterhin hat jeder Stixel eine feste Stixel-Weite. Ziel der Verwendung ist die Abstraktion der 3D-Bilddaten durch eine zusätzliche Repräsentationsschicht (Stixels). In der Abbildung 2.3.2 ist ein Beispiel für farbkodierte Stixel zu sehen [PF10, Grafik S. 1]. Rote Stixel sind nahe und grüne ferne Teilobjekte, welche auf dem Boden stehen. Für die Verwendung des Ansatzes werden die intrinsischen und extrinsischen Parameter sowie Disparitätsbilder benötigt.

2 Grundlagen

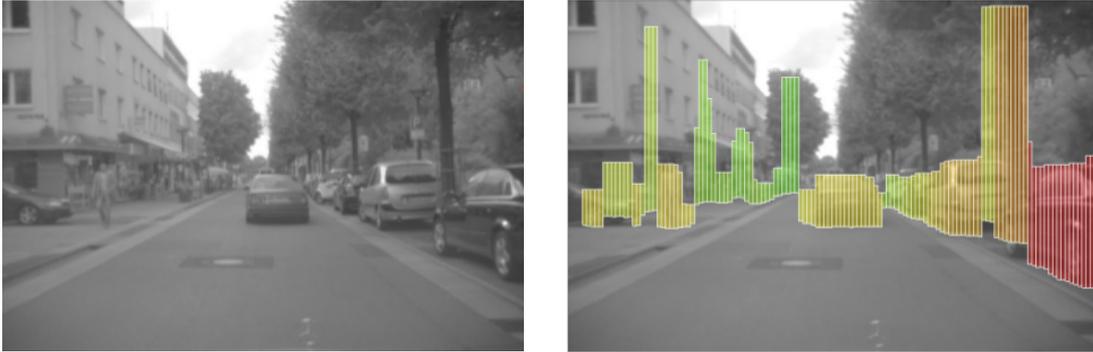


ABBILDUNG 2.3.2: Beispiel für farbkodierte Stixel

Stixel werden in folgende Kategorien eingeteilt (vgl. [PF10] und [Pfe11]):

- Statische Stixel: Für jedes Bild werden Stixel neu berechnet
- Dynamische Stixel: Stixel werden über die Zeit verfolgt
- Standard Stixel: Stixel beschreiben auf dem Boden stehende Objekte
- Erweiterte Stixel: Zusätzliche Segmentierung des Bodens und Hintergrundes

Stixel-Generierung

Grundlage für die Erzeugung von Stixeln ist die Annahme, dass eine Straße eine relativ freie Fläche ist, auf der Objekte nahezu vertikal stehen. Die freie Fläche wird durch darauf stehende Objekte limitiert. Damit Stixel erzeugt werden können, werden folgende Phasen durchlaufen (vgl. [PF10], [PF11] und [BFM07]):

1. **Disparitätsbild:** Dieses muss zunächst erzeugt werden, da es die Grundlage für die Extraktion von Stixeln ist.
2. **Freie-Fahrbahnebenen-Berechnung:** Zunächst muss für jede Bildspalte die Position eines auf der Fahrbahnebene stehenden Objektes berechnet werden. Dazu wird zunächst ein Kostenbild generiert. Dieses enthält für jeden Pixel eine Bewertung ob die Disparität des Disparitätsbildes zu einem

2 Grundlagen

möglichen Objekt oder zur Fahrbahn gehört. Mittels dynamischer Programmierung wird aus diesem Bild in jeder Bildspalte das Optimum für ein auf dem Boden stehendes Objekt gesucht. Als Ergebnis erhalten wir für jede Bildspalte einen Basispunkt.

3. **Objekt-Höhen-Segmentierung:** Ausgehend von den berechneten Basispunkten wird eine optimale Segmentierung der Höhe zwischen Objekt und Hintergrund durchgeführt. Es wird ein Kostenbild berechnet, welches bewertet ob die Disparität des Disparitätsbildes zu einem Objekt oder zum Hintergrund des Bildes gehört. Im Anschluss wird aus dem Kostenbild mittels dynamischer Programmierung die optimale Höhe in jeder Bildspalte berechnet. Als Ergebnis erhalten wir für jede Spalte einen Punkt für die Höhe eines auf dem Boden stehendes Objektes.
4. **Stixel-Extraktion:** Die Resultate aus Phase 2 und 3 werden verwendet um Stixel zu extrahieren. Jeder Stixel hat eine feste Stixel-Weite, welcher mit Hilfe des Basispunktes und der Höhe eines Objektes einen Bereich beschreibt. Die Disparität des Bereiches wird analysiert und ein entsprechender Wert beispielsweise mit einer Häufigkeitsanalyse extrahiert.

Für die Freie-Fahrbahnebenen-Berechnung wird ein sogenanntes Disparitätsprofil der Straße benötigt (vgl. [PF10, Abschnitt B]). Solch ein Profil beschreibt die Beschaffenheit der Oberfläche der Fahrbahn mittels Disparitätswerten. In [PF10] wird die Verwendung aus [Wed+08] empfohlen. Es existieren jedoch noch weitere Möglichkeiten wie beispielsweise in [LAT02] oder [SF07] beschrieben. In [HLU05] wird eine Übersicht über weitere verschiedene Verfahren zur Erzeugung solch eines Profils gegeben. Die einfachste Möglichkeit ein Profil zu erzeugen ist die Annahme, dass die Straße stets planar sowie das Stereo-System parallel zum Boden ausgerichtet ist. Dann kann mit der Gleichung 2.6 und mit den Parametern Kamerahöhe h , Basislinie b sowie dem Optischen Zentrum c für jede Bildzeile v ein Disparitätsprofil der Straße d_R berechnet werden (vgl. [HLU05, Gl. 11]):

$$d_R = \frac{b}{h} * (v - c_v) \quad (2.6)$$

2.4 Positionierung

Ein Softwaremodul zur Positionierung des Fahrzeuges auf einem 2D-Weltkoordinatensystem (Auflösung in Metern) ist von Audi vorgegeben. Die Position wird benötigt damit das Fahrzeug seine Position auf der Weltkarte kennt. Der Begriff Weltkarte ist im Rahmen des AADC gleichwertig zum Begriff Teststrecke. Die Positionierung funktioniert wie folgt: Zunächst existiert eine Liste von bekannten Verkehrsschildern mit Positionen. Bestimmte Schilder in der Liste sind als spezielle Initialisierungsschilder gekennzeichnet und jedem Schild ist eine Position $(x, y) \in \mathbb{R}^2$ auf der Weltkarte zugeordnet. Beim Starten des Fahrzeuges wird zunächst nach einem Initialschild gesucht, damit das Fahrzeug seine Startposition kennt. Die Erkennung von den Schildern erfolgt mittels Bit-Codes. In Abbildung 2.4.1 ist ein Stoppschild mit einem Bit-Code zu sehen. In Kamerabildern wird nach solchen Codes gesucht und mit der Schilderliste abgeglichen.

Im weiteren Verlauf wird mittels der Eigenbewegung des Fahrzeuges, der Vorhersage der aktuellen Position sowie durch Detektierten von Schildern die Eigenposition auf der Weltkarte ermittelt [AGb]. In Abbildung 2.4.1 ist ein Beispiel für eine Weltkarte abgebildet. Der Fahrzeugorientierungswinkel, also die Fahrtrichtung des Fahrzeuges auf einer Weltkarte, wird nach ISO 8055 dargestellt. Norden entspricht dabei 90° , Osten 0° , Süden -90° und Westen 180° . Aus dem Positionierungs-

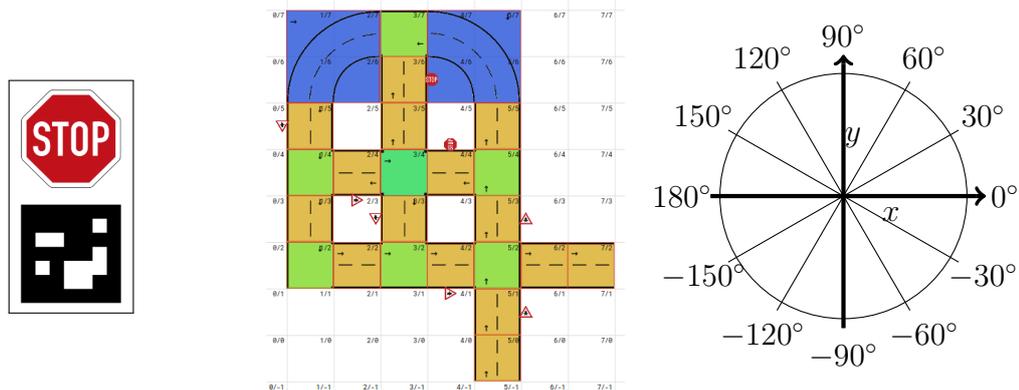


ABBILDUNG 2.4.1: Beispiel Schild, Weltkarte und Orientierung in Grad (ISO 8855)

dul erhalten wir die Position des Fahrzeuges sowie dessen Orientierungswinkel.

2 Grundlagen

Damit erkannte Objekte auf die Weltkarte übertragen werden können, wird die Translation und Rotation benötigt.

Die Translation $(trans_x, trans_y) \in \mathbb{R}^2$ einer Position:

$$\begin{pmatrix} x_{trans} \\ y_{trans} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} trans_x \\ trans_y \end{pmatrix} \quad (2.7)$$

Die Rotation um den Orientierungswinkel α in Rad:

$$\begin{pmatrix} x_{rot} \\ y_{rot} \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} * \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.8)$$

Für den Abstand zwischen zwei Positionen A und B auf der Weltkarte kann die euklidische Distanz verwendet werden:

$$dist = \sqrt{\sum_{i=1}^2 (A_i - B_i)^2} \text{ mit } A, B \in \mathbb{R}^2 \quad (2.9)$$

2.5 Überblick

Aktuelle Forschungen zum Thema der Objektdetektion nutzen vor allem Lidar-Sensoren, aber auch Stereo-Kamera-Systeme werden eingesetzt. Die Verfolgung von Objekten über die Zeit wird oft mit Hilfe von digitalen Bildern realisiert.

Das von Audi zur Verfügung gestellte Modellfahrzeug besitzt die Intel RealSense R200 Tiefenkamera. Mit Hilfe dieser Kamera und der digitalen Bildverarbeitung können Objekte im Raum detektiert und später klassifiziert werden. Es können sowohl Punktwolken als auch Disparitätsbilder in Kombination mit dem Farbbild genutzt werden um dies zu realisieren. Eine Möglichkeit mit Hilfe von Disparitätsbildern Objekte zu detektieren ist die Verwendung von Stixeln. Zusätzlich steht ein Positionierungsmodul zur Verfügung, welches die Position des Fahrzeuges auf der Weltkarte bestimmt. Mit Hilfe der Fahrzeugposition und der 3D-Kamera kann bestimmt werden, wo sich Objekte auf der Weltkarte befinden.

3 Konzept

Das Motto des AADC 2017 stand unter dem Thema KI. Im Wesentlichen ist es die Aufgabe, neben den zu realisierenden Fahrfunktionen, Objekte zu erkennen. Die Anwendung solch einer Erkennung kann als Beispiel dem *Audi Autonomous Driving Cup Regelwerk 2017*, S. 26 entnommen werden:

„Das Fahrzeug muss selbständig in der Lage sein zwischen Person und anderen Gegenständen zu unterscheiden. Dabei ist der Person Vorrang zum Überqueren des Zebrastreifens zu gewähren.“

Die Klassifikation wird dazu verwendet, solch eine Unterscheidung zu realisieren. Damit eine Klassifikation von Objekten ermöglicht wird, werden zu klassifizierende Objekte zunächst detektiert. Im Anschluss werden Objekte über die Zeit verfolgt. Dazu eine Anforderung aus dem *Audi Autonomous Driving Cup Regelwerk 2017*, S. 26.

„Hindernisse können an verschiedenen Positionen im gesamten Parcours auftreten [...]. Die Position des Hindernisses muss an das Backend gesendet werden.“

Als zusätzliche Bedingung ist gefordert, dass die Rückmeldung an die Jury über das Backend für ein detektiertes Objekt nur einmalig geschieht. Die Klassifikation ist nicht Bestandteil dieser Arbeit, sondern die Vor- sowie Nachbearbeitungsschritte.

Zunächst soll die Ausgangssituation und Zielstellung konkretisiert sowie Erfolgskriterien betrachtet werden. Dies führt zu einem ersten Konzept für die Umsetzung.

Im Anschluss erfolgt die Wahl des Verfahrens für Detektion und Verfolgung von Objekten. Auf Basis der Wahl erfolgt eine Verfeinerung des Konzeptes.

3.1 Präzisierung der Ausgangssituation

Es sind im Smart Driving Team keine Ansätze für Detektion, Klassifikation und Verfolgung von Objekten aus den Vorjahren vorhanden. Weiterhin sind für den AADC 2017 mehrere Bestandteile vorgegeben.

Für die Detektion von Objekten steht die Intel RealSense R200 zur Verfügung. Diese Kamera wird über die Intel RealSense Cross Platform API in Version 1.11.0 angesprochen. Über diese kann die Kamera konfiguriert und diverse Parameter sowie die Bilddaten ausgelesen werden [Rea16b].

Weiterhin steht zur digitalen Bildverarbeitung die API **Open Source Computer Vision Library (OpenCV)** in Version 3.2 zur Verfügung. Diese Bibliothek wird als Hilfsmittel für die Bildverarbeitung genutzt und implementiert viele nützliche Algorithmen [Tea17].

Die Programmierung des Fahrzeuges erfolgt mit Hilfe des **Automotive Data and Time-Triggered Framework (ADTF)** von Audi. Dieses Framework abstrahiert die Hardware sowie die Kommunikation des Fahrzeuges und erlaubt eine modulare Softwareentwicklung. In ADTF werden Module in der Programmiersprache C++ programmiert. Ein Modul besitzt sogenannte Ein- und Ausgabe-Pins. Durch das Verbinden von Ein- und Ausgabe-Pins von verschiedenen Modulen wird der Ablauf eines Programms in ADTF gesteuert.

Ein Softwaremodul in ADTF zur Positionierung mit Hilfe von Schildern des Fahrzeuges auf einer 2D-Weltkarte ist vorgegeben. Für die Positionierung ist eine Datei vorgegeben, welche alle Schilder und deren Position auf der Karte enthält.

Außerdem sind Regeln durch [AGa] vorgegeben. Implizit gegeben ist zudem, dass relevante Objekte immer einen gewissen Abstand zueinander haben. Zusätzlich ist im Rahmen des Wettbewerbes die Straße stets planar.

3.2 Konkretisierung der Zielstellung

Mit Hilfe der Tiefenkamera ist eine Detektion von Objekten zu realisieren. Durch die Verwendung dieser Kamera kann die Distanz und relative Position zur Kamera für Bildpunkte berechnet werden. Ziel der Objektdetektion ist es, relevante Bildbereiche zu finden, in denen Objekte sind, welche klassifiziert werden sollen. Dies schränkt den Suchraum für die Klassifikation auf relevante Objekte ein.

Für detektierte Objekte muss der Bildbereich auf dem Farbbild ermittelt werden, damit die Klassifikation anhand des eingegrenzten Bildbereichs auf Farbbildern realisiert werden kann. Die Verwendung von Farbbildern verbessert die Klassifikation mittels KI. Insbesondere muss es möglich sein, kleine Objekte, wie beispielsweise Kinder, zu erkennen.

Zusätzlich müssen die Weltkoordinaten des detektierten Objektes berechnet werden, damit die Koordinaten an die Jury zurückgemeldet werden können. Mit Hilfe von Bildkoordinaten, welche einen Bildbereich beschreiben, erfolgt eine Klassifikation des Objektes. Bei dieser wird festgelegt ob sich das Objekt um einen Erwachsenen, ein Kind, ein Auto oder ein Verkehrsschild handelt. Im Anschluss wird eine Verfolgung von Objekten über die Zeit realisiert, mit dem Ziel, Objekte einmalig an die Jury zurückzumelden. Im Rahmen des AADC ist eine Objektvorhersage nicht von Bedeutung.

Diese Zielstellung führt zum dargestellten groben Ablauf in der Abbildung 3.2.1. In der Abbildung sind die Phasen Bilddatengewinnung, Detektion, Klassifikation, Verfolgung sowie Weiterverarbeitung zu sehen. Die in dieser Arbeit realisierten Phasen sind farblich hervorgehoben. Die Klassifikation und Weiterverarbeitung sind mittels punktierten Rändern dargestellt. Diese sind nicht Bestandteil der Aufgabe.

3 Konzept

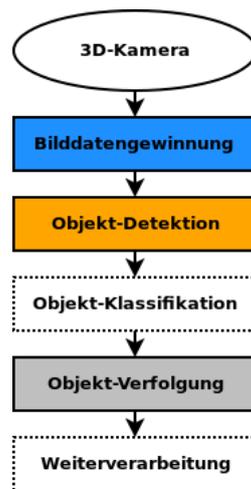


ABBILDUNG 3.2.1: Grobkonzept der Teilaufgaben

Anforderungen

Auf Basis der Ausgangssituation sowie der Zielstellung ergeben sich folgende Zielstellungen zur Lösung der Probleme im Rahmen des AADC:

Detektion wichtiger Objekte Es müssen folgende Objekte erfolgreich detektiert werden können: Schilder, Pylone, Autos, Erwachsene und Kinder

Geringe Fehldetektion Objekte die zu groß sind oder nicht existieren, sollen nicht extrahiert werden. Als Beispiel seien Wände genannt.

Abbildung auf Farbbild Infrarotbildkoordinaten der erkannten Objekte müssen auf Farbbildkoordinaten übertragen werden.

Distanz der Detektion Die Objekt-Detektion sollte bis zu einer Distanz von 2.8m funktionieren. Dies ist der genannte ideale Bereich der R200.

Position von Objekten Die berechnete Position der Objekte auf der Weltkarte muss der realen Position entsprechen.

Wiederverwendbarkeit Die verwendeten Verfahren müssen wiederverwendbar sein, beispielsweise im AADC 2018.

Einmaliges Melden von neuen Objekten Erkannte Objekte sollen nur einmalig an die Jury gemeldet werden

Implementierungszeit Die zu realisierenden Aufgaben müssen innerhalb des Wettbewerbes implementiert werden.

Performance Die vorgestellten Lösungen müssen ausreichend schnell auf dem Fahrzeug funktionieren.

3.3 Wahl der Verfahren

Nachdem die Zielstellung konkretisiert und Erfolgskriterien genannt sowie ein grober Ablauf dargestellt wurde, sind nun mögliche Verfahren für die Detektion und Nachverfolgung zu wählen.

3.3.1 Detektion

Für die Lösung des Problems der Objektdetektion steht die Intel RealSense R200 zur Verfügung. Diese Kamera liefert rektifizierte Stereo-Infrarot-, Farb-, Tiefen- und Disparitätsbilder sowie Punktwolken. Für die Detektion sind vor allem Punktwolken und das Tiefen- beziehungsweise Disparitätsbild von Bedeutung. Aufgrund dessen, dass ein Disparitätsbild eine andere Form eines Tiefenbildes ist, wird das Tiefenbild nicht weiter betrachtet. Dadurch müssen im Wesentlichen nur Punktwolken und Disparitätsbilder betrachtet werden. In der Tabelle 3.3.1 werden die Vor- und Nachteile dargestellt.

Punktwolke		Disparitätsbild	
+	3D-Daten können direkt verwendet werden	-	Zusätzliche Umrechnung notwendig
+	Genaue Positionsbestimmung möglich		
-	Erfordert spezielle Hardware zur direkten Erzeugung		
-	Aufwändigere Erzeugung in Software als Disparitätsbild	+	Einfache Erzeugung mittels OpenCV aus rektifizierten Stereobildern

TABELLE 3.3.1: Vor- und Nachteile von Punktwolken und Disparitätsbildern

3 Konzept

Eines der Kriterien für den Erfolg ist die Wiederverwendbarkeit der vorgestellten Lösung. Für den nächsten AADC ist nicht bekannt, welche Kameras auf den Modellfahrzeugen verbaut sein werden. Dadurch ist ungewiss, ob diese Kameras direkt Punktwolken oder Disparitätsbilder erzeugen können. Die alternative Verwendung eines Lidar-Sensors im nächsten Fahrzeug ist unwahrscheinlich. Diese Geräte sind sehr teuer. Es ist zu erwarten, dass mindestens eine Stereo-Kamera vorhanden sein wird. Mittels OpenCV ist ein Disparitätsbild leicht erzeugbar beispielsweise mit dem SGBM-Verfahren. Die Generierung von Punktwolken benötigt in Software zusätzliche Schritte. Somit fällt die Wahl auf das Disparitätsbild als Ausgangspunkt für die Objektdetektion.

Aufgrund der Wahl der Verwendung des Disparitätsbildes werden in dieser Arbeit Stixel verwendet um das Problem der Objektdetektion zu lösen. Es werden die Standard Stixel verwendet, da diese nur Objekte beschreiben, welche auf dem Boden stehen. Für die Verwendung des Ansatzes benötigen wir zusätzlich die uns bekannten intrinsischen und extrinsischen Parameter der Kamera. Weiterhin muss für die Verwendung der Stixel ein Disparitätsprofil der Straße erzeugt werden (vgl. Abschnitt 2.3.2, S. 12). Aufgrund der begrenzt vorhandenen Zeit sowie der stets planaren Straßen im Rahmen des AADC 2017 wird die einfachste Form zur Erzeugung des Profils verwendet (Siehe Gleichung 2.6, S. 13).

3.3.2 Klassifikation

Die Klassifikation wird in dieser Arbeit als Blackbox betrachtet. Ein anderes Teammitglied des Smart Driving Teams hat die Lösung des Problems entwickelt und diese wird im Rahmen eines Masterprojektes zum AADC 2017 vorgestellt.

3.3.3 Verfolgung

Im Anschluss an die Klassifikation ist die Verfolgung von Objekten zu realisieren. Das Ziel der Objektverfolgung ist das einmalige Rückmelden von neu detektierten Objekten an die Jury.

3 Konzept

Im Abschnitt 2.1 wurde für die Verfolgung von Objekten bereits erwähnt, dass oftmals der optische Fluss auf Bildern verwendet wird. Das Ergebnis des optischen Flusses wird mit Kalman-Filtern kombiniert um Objekte über die Zeit zu verfolgen und Positionen vorherzusagen. Angewendet in Kombination mit Stixeln führt dies zum Begriff der dynamischen Stixel. Im Rahmen des AADC wird keine Vorhersage für die Bewegung des Objektes benötigt. Zusätzlich besitzt das Modellfahrzeug nur begrenzte Ressourcen, allem voran CPU-Leistung, welche für andere Aufgaben ebenfalls benötigt werden. Die Grafikkarte wird vollständig für die Klassifikation von Objekten benötigt. Als zusätzliches Kriterium spielt die vorhandene Implementierungszeit eine wichtige Rolle. Daher wird sich dafür entschieden die statischen Standard Stixel zu verwenden sowie einen einfachen Ansatz zur Objektverfolgung zu realisieren. Bei diesem sollen Objekte auf einer Weltkarte mit Hilfe der Objektklasse, Position und zugehörigem Radius des Objektes über die Zeit verfolgt werden. Es erfolgt keine Vorhersage der Objektpositionen.

3.4 Feinkonzept

Nachdem sich für die Ansätze zur Lösung der Probleme entschieden wurde, kann das Konzept weiter verfeinert werden. Insbesondere abstrahieren Stixel die 3D-Daten aus dem Disparitätsbild auf eine abstrakte Datenstruktur. Mit dieser Datenstruktur ist keine Klassifikation auf Bildern möglich. Durch die Stixel werden Bereiche beschrieben in denen Teile von möglichen Objekten auf dem Boden stehen. Sie extrahieren keine kompletten Objekte. Darum ist eine Verfeinerung des Konzeptes notwendig. Das verfeinerte Konzept ist in Abbildung 3.4.1 zu sehen. Folgende Schritte sind zu sehen:

Bilddatengewinnung Zunächst erfolgt die Bilddatengewinnung aus der Kamera sowie Erfassung und Speicherung der benötigten Kameraparameter. Der Fluss der Parameter wird nicht in der Konzeptgrafik dargestellt, da diese für alle Phasen identisch sind. Als Ergebnis der Bilddatengewinnung erhalten wir das Disparitätsbild der aktuellen Szene.

3 Konzept

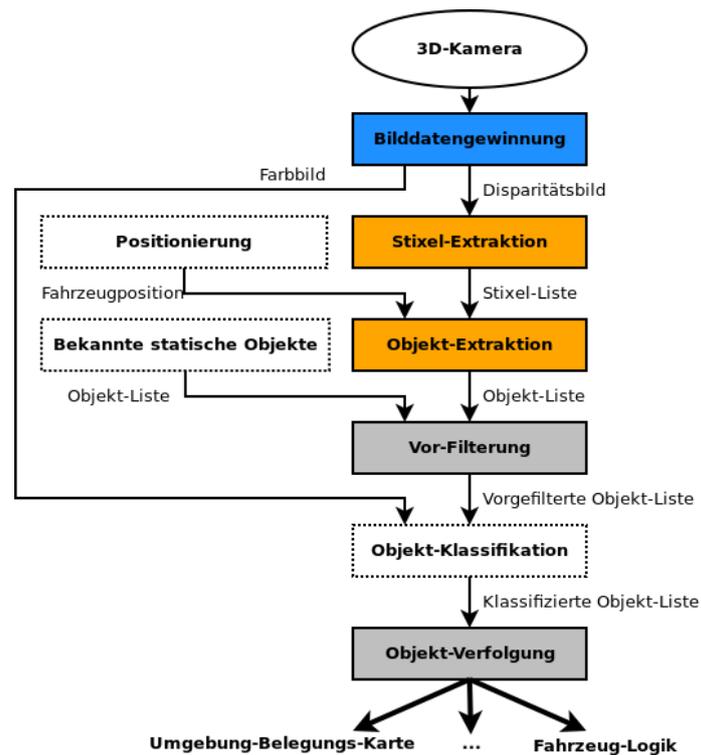


ABBILDUNG 3.4.1: Vollständiges Konzept der Teilaufgaben

Objekt-Detektion Auf Basis des Disparitätsbildes sowie der intrinsischen und extrinsischen Parameter der Intel RealSense R200 werden Stixel extrahiert.

Im Anschluss werden aus den Stixeln Objekte extrahiert. Dies geschieht indem Stixel zu möglichen Objekten zusammengefasst werden. Diese beschreiben als Resultat Bildbereiche auf dem linken Infrarotbild. Solch ein Bildbereich wird durch ein Rechteck mit zwei Tupeln von Bildkoordinaten beschrieben. Die Koordinaten des linken Infrarotbildbereiches sollen auf zugehörige Koordinaten des Farbbildes abgebildet werden. Außerdem wird mit Hilfe des vorgegebenen Positionierungsmodul die Position des extrahierten Objektes auf der Weltkarte berechnet sowie der Radius des Objektes bestimmt.

Objekt-Verfolgung Die gewonnen Informationen aus der Objekt-Detektion werden vorgefiltert. In diesem Schritt werden Objekte, welche an der Position von Schildern sind, aus der Objekt-Liste entfernt. Diese müssen nicht klassifiziert wer-

3 Konzept

den, denn Schilder werden durch ein anderes Modul behandelt (nicht Teil dieser Arbeit).

Im Anschluss an die Vorfilterung erfolgt die Klassifikation auf dem Farbbild mittels KI. Als Ergebnis erhalten wir klassifizierte Objekte.

Anhand der Klasse, Weltkoordinate und dem Radius des Objektes erfolgt die Objekt-Verfolgung. Jedes Objekt besitzt dazu folgende Informationen:

- Bildkoordinaten auf Farbbild
- Position und Radius auf Weltkarte
- Klassen und zugehörige Wahrscheinlichkeiten

Gewonnene Informationen und Kenntnisse werden im Anschluss zur Weiterverarbeitung weitergegeben, beispielsweise an die Fahrzeuglogik, welches entsprechend auf bestimmte Situationen reagiert, oder die Daten werden in eine Umgebung-Belegungs-Karte eingezeichnet. Die Weiterverarbeitung verwendet im Team entwickelte Lösungen. Diese werden in der Arbeit nicht verwendet und somit nicht dokumentiert.

3.5 Überblick

In diesem Abschnitt wurde die Ausgangssituation im Rahmen des AADC 2017 und die Zielstellung konkretisiert. Außerdem wurden die Erfolgskriterien für die Lösung der Aufgaben genannt.

Als Grundlage für eine mögliche Objekt-Detektion werden Stixel verwendet. Dies führte zu einem Konzept, welches folgende Phasen enthält: Bilddatengewinnung, Objekt-Detektion und Objekt-Verfolgung. Zur Lösung der Objektverfolgung wurde sich für einen einfachen Ansatz mittels Positionen und Radien von klassifizierten Objekten auf einer Weltkarte entschieden. Die Klassifikation und Weiterverarbeitung der gewonnenen Informationen ist nicht Bestandteil der zu realisierenden Aufgaben.

4 Bilddatengewinnung

In der Konzeption viel die Entscheidung Objekte mittels Disparitätsbilder zu extrahieren. In diesem Abschnitt wird dieses Bild hinsichtlich der Qualität bewertet. Weiterhin werden in dieser Phase diverse Kameraparameter extrahiert und gespeichert. Auf die Gewinnung der Farbbilder von der Kamera wird nicht eingegangen. Dies bereitet mit Hilfe der RealSense API keine Probleme.

4.1 Disparitätsbild

Ein Disparitätsbild kann mittels verschiedener Verfahren erzeugt werden. Grundlage für solch ein Bild ist das Konzept des Stereo-Sehens. Die Intel RealSense R200 kann solch ein Bild direkt erzeugen. Ausgelesen werden die Bilder synchron über die RealSense API. Ein Beispiel für die Generierung eines Disparitätsbildes aus linkem und rechtem Infrarotbild ist in der Abbildung 4.1.1 dargestellt. Zunächst

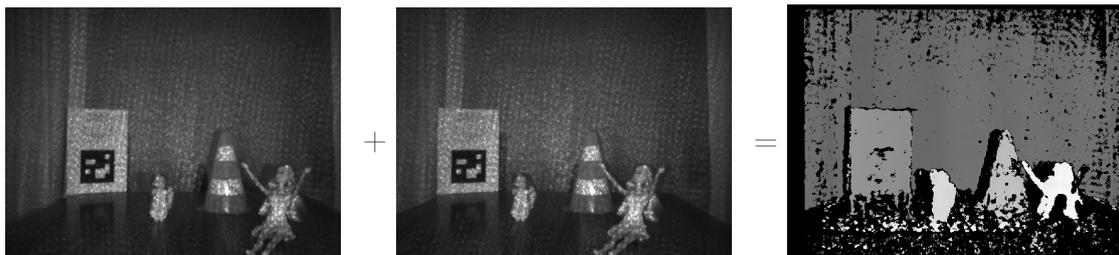


ABBILDUNG 4.1.1: Erzeugung des Disparitätsbildes aus linkem und rechtem Infrarotbild

ist in diesem Beispiel zu sehen, dass das Schild, der Pylon sowie die Kinderpuppe vollständig erfasst werden. Die Erwachsenenpuppe wird nur zu etwa 50% erfasst. Bei genauerer Betrachtung fällt auf, dass dieses Disparitätsbild viele Fehlerwerte

(Schwarz) enthält. Insbesondere enthält der Boden viele Fehlerwerte. Weitere Versuche mit dem Disparitätsbild der R200 bestätigen, dass das Disparitätsbild für die Anwendung von Stixel bedingt geeignet ist. Den zu viele Fehlerwerte würden die Qualität der Stixel negativ beeinflussen. Die Qualität von Tiefenbildern wurde in der Arbeit [Kes+17] bewertet und bestätigt das Ergebnis bezüglich der Qualität des Disparitätsbildes der Kamera.

Erzeugung des Disparitätsbildes

Wie in Abschnitt 2.3.1 (S. 10) bereits erwähnt, kann mittels des SGBM-Verfahren ein Disparitätsbild aus zwei rektifizierten Bildern generiert werden. Tests mit diesem Verfahren haben gezeigt, dass dieses ein besseres Ergebnis als das Verfahren der Kamera liefert. In der Abbildung 4.1.2 ist ein Beispiel zu sehen. Es ist deutlich zu sehen, dass in dem durch das SGBM-Verfahren generierten Bild weniger Fehler enthalten sind.

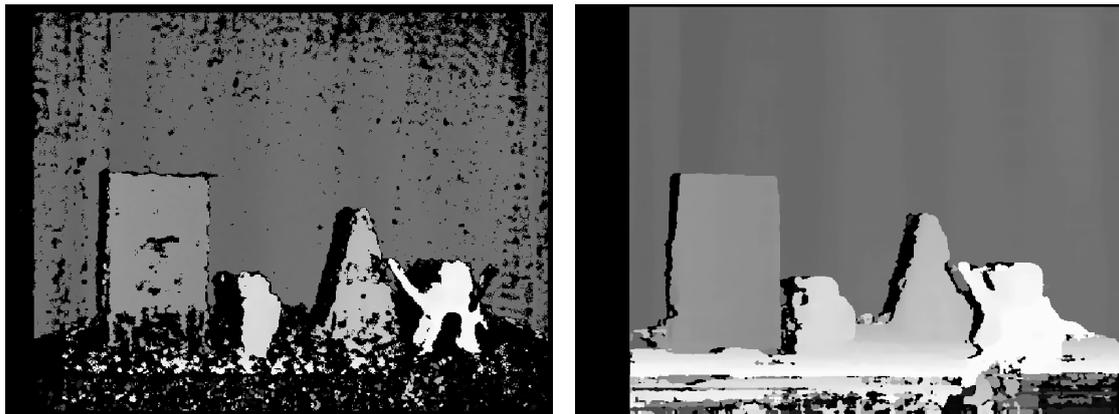


ABBILDUNG 4.1.2: Vergleich der Disparitätsbilder von der Kamera (l.) und vom SGBM-Verfahren (r.)

Darum wird dieses Verfahren zur Erzeugung von Disparitätsbildern gewählt. Weiterhin hat dies den Vorteil, dass dadurch die Wiederverwendbarkeit verbessert wird, denn es werden nur noch zwei rektifizierte Bilder vorausgesetzt, statt einem Disparitätsbild. Jedoch hat dieses Verfahren den Nachteil, dass Bildränder nicht

interpretiert werden können. Dies ist in der rechten Abbildung 4.1.2 zu sehen. Dieser Nachteil wird auf Grund der allgemein besseren Ergebnisse toleriert.

4.2 Parameter

Für die Stixel und Objekt-Detektion werden die intrinsischen Parameter Brennweite f und Bildmittelpunkt c der linken Infrarot-Kamera benötigt. Für die Abbildung von Koordinaten aus dem linken Infrarotbild in das RGB-Bild werden zusätzlich die intrinsischen Parameter der RGB-Kamera benötigt. Ebenfalls sind die extrinsischen Parameter für die Translation von der Infrarot-Kamera zur Farb-Kamera notwendig. Diese Parameter werden mit Hilfe der RealSense API ausgelesen. Weitere benötigte extrinsischen Parameter können gemessen werden. Dies sind die Basislinie b und Höhe h der Kamera über dem Boden. Messungen ergeben für b einen Wert von 7 cm und für h von 24 cm. Alle Parameter werden gespeichert und von Modulen, die sie benötigen, abgerufen. Eine Auflistung der ausgelesenen Parameter ist im Anhang A.1.1 zu finden.

4.3 Überblick

Es hat sich gezeigt, dass erzeugte Disparitätsbilder der Intel RealSense R200 viele Fehlerwerte extrahieren. Deswegen werden Disparitätsbilder mit Hilfe des SGBM-Verfahren aus rektifizierten Infrarotbildern generiert. Dieses Verfahren liefert Disparitätsbilder mit weniger Fehlerwerten.

Alle benötigten Bilder und Kameraparameter werden über die RealSense API ausgelesen. Weitere benötigte Parameter werden gemessen.

5 Objekt-Detektion

Die Aufgabe der Detektion von Objekten gliedert sich in zwei Teilaufgaben. Zunächst müssen mit Hilfe der Kameraparameter und einem Disparitätsbild Stixel extrahiert werden. Dazu muss das Stixel-Verfahren für das Modellfahrzeug parametrisiert werden. Diese Datenstruktur wird im Anschluss dazu verwendet, Objekte zu finden.

Gefundene Objekte werden extrahiert und die zugehörigen Bildbereiche auf dem Farbbild berechnet. Der Bereich ist für die spätere Klassifikation von Bedeutung. Weiterhin wird für jedes Objekt die Position und der Radius auf der Weltkarte berechnet.

5.1 Stixel-Extraktion

Auf Basis der in [BPF09], [PF10] und [PF11] vorgestellten Verfahren werden Stixel extrahiert. Für die Erzeugung des Disparitätsprofils der Straße wird die Gleichung 2.6 (S. 13) verwendet. Eine Beispielszene ist in Abbildung 5.1.1 zu sehen. Zu sehen ist das linke Infrarotbild sowie das erzeugte Disparitätsbild. In der Szene sind ein Modellfahrzeug und ein seitlich orientiertes Verkehrsschild sowie der Hintergrund zu sehen.

Zunächst werden aus diesem Disparitätsbild die Stixel mit den empfohlenen Parametern extrahiert. Das Ergebnis der Anwendung des Verfahrens ist in Abbildung 5.1.2 abgebildet. Durch das Verfahren wird vor allem der Hintergrund extrahiert. Dargestellt durch die farbkodierten Stixel. Je rötlicher die Farbe desto größer ist die extrahierte Disparität für den entsprechenden Stixel. Ferne Stixel

5 Objekt-Detektion

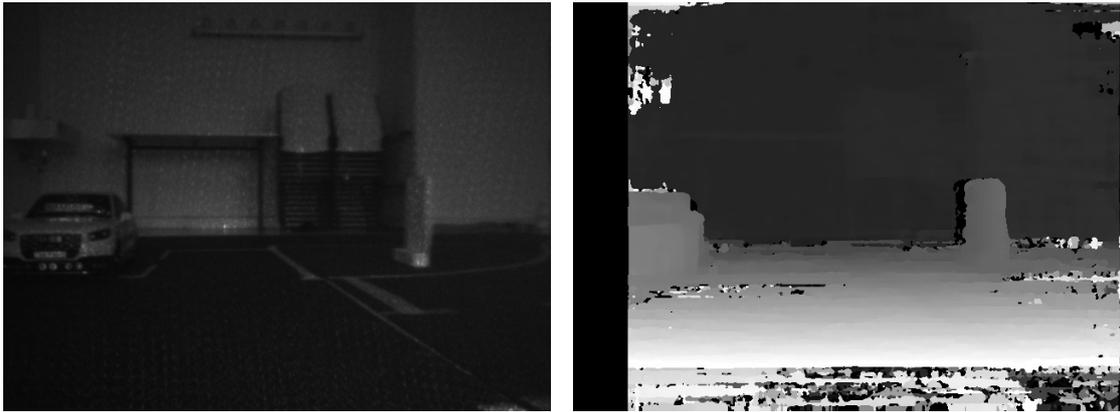


ABBILDUNG 5.1.1: Infrarotbild und Disparitätsbild einer Szene

werden grün dargestellt. Hat ein Stixel die Farbe grau so wurde als Disparität der Fehlerdisparitätswert extrahiert. Dies ist insbesondere am linken Rand zu sehen. Den durch das SGBM-Verfahren können die Ränder eines Bildes nicht interpretiert werden.

Der Grund für die Detektion des Hintergrundes ist, dass die verwendeten Parameter für reale Verkehrsszenen dimensioniert sind. Diese Parameter sind also ungeeignet für das Modellfahrzeug und müssen entsprechend angepasst werden.

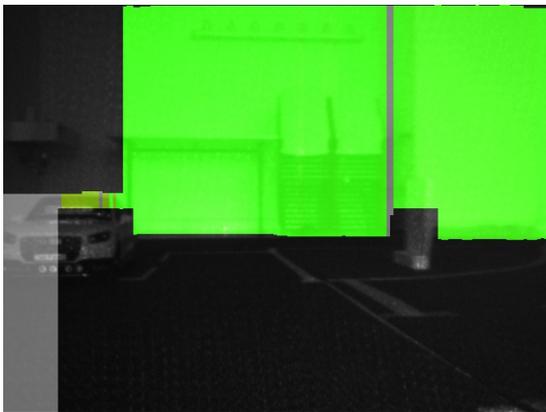


ABBILDUNG 5.1.2: Stixel mit Originalparametern

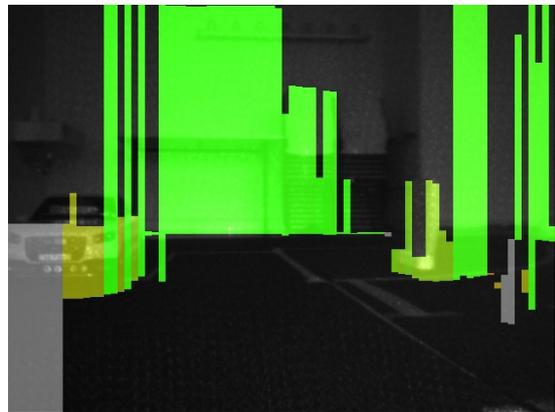


ABBILDUNG 5.1.3: Stixel mit angepassten Parametern

5.1.1 Anpassung der Parameter

Für die Extraktion von Stixeln werden diverse Parameter benötigt, welche Einfluss auf die Berechnung des Basispunktes und die Höhe eines Objektes haben. In [PF11] und [BFM07] werden Werte für die Verwendung der Stixel im realen autonomen Fahren empfohlen. Im Rahmen des AADC wird ein Modellfahrzeug im Maßstab 1 zu 8 verwendet, sodass alle Parameter auf $\frac{1}{8}$ der empfohlenen Werte gesetzt werden. Im Anhang A.1.2 sind die gewählten Parameter vollständig gelistet.

Für die Extraktion von Stixeln wird zusätzlich die sogenannte Stixel-Weite w_S in Pixeln benötigt. Diese gibt an wie viele Pixel bezüglich der Breite auf dem Bild zu einem Stixel zusammengefasst werden. Die Wahl dieses Parameters hat Einfluss auf die Laufzeit des Algorithmus. Je kleiner dieser Wert ist, umso detaillierter können Stixel ein Objekt beschreiben, aber umso länger benötigt der Algorithmus. Empfohlen wird in der Literatur $w_S = 5$ Pixel. Tests haben gezeigt, dass $w_S = 8$ Pixel im Rahmen des AADC ausreichend ist. In der [Abbildung 5.1.3](#) ist das Ergebnis der Extraktion von Stixeln mit angepassten Parameter auf dem Modellfahrzeug dargestellt.

5.1.2 Bewertung der Ergebnisse

Zunächst ist in der [Abbildung 5.1.3](#) zu sehen, dass eine Anwendung von Stixeln auf dem Modellfahrzeug prinzipiell möglich ist, da das Fahrzeug und das Verkehrsschild teilweise von Stixeln umgeben werden. Gleichzeitig fallen in dieser Beispielszene Probleme auf.

Es werden bei dem Fahrzeug zwei sehr große Stixel extrahiert. Dies ist damit begründet, dass die Basispunkte vom Fahrzeug zu niedrig gewählt werden. Diese werden zu niedrig gewählt auf Grund des verhältnismäßig kleinen Maßstabes der Parameter. Insbesondere werden dadurch auftretende Fehler im Disparitätsbild verstärkt. Dieser Fehler tritt in ähnlicher Weise ebenfalls bei dem Schild auf. In der Mitte des detektierten Schildes existiert ein Stixel von geringerer Größe. Dort

5 Objekt-Detektion

wurde der Basispunkt ebenfalls zu niedrig gewählt, sodass das eigentliche Schild dort nicht als Objekt sondern als Hintergrund bewertet wird.

Fehler im Disparitätsbild entstehen zusätzlich durch die Bewegung des Fahrzeuges sowie durch Unebenheiten der Teststrecke. Denn die Strecke besteht aus einzelnen 1x1m großen Teilstücken, welche zusammengelegt werden. Diese schließen nicht bündig ab, sodass die Straße durch kleine Spalte und Überlappungen von Teilstücken unterbrochen ist. Darum befinden sich beispielsweise im Disparitätsbild (vgl. Abbildung 5.1.1, S. 29) unterhalb des Fahrzeuges fehlerhafte Disparitätswerte.

Des Weiteren neigt sich das Fahrzeug leicht bei Beschleunigungs- und Bremsvorgängen, aber auch bei Kurven-Fahrten. Dadurch verändert sich der Blickwinkel der Kamera dynamisch zum Boden. Bei der realisierten Implementierung der Stixel wird davon ausgegangen, dass die Straße stets planar ist und der Blickwinkel der Kamera zur Straße sich nicht verändert. Dadurch werden die berechnete Basispunkte für die Freie-Fahrbahnebenen-Berechnung zusätzlich verschoben (vgl. Abschnitt 2.3.2, S. 12).

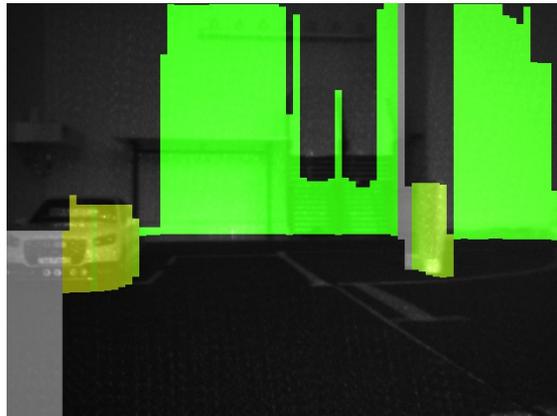


ABBILDUNG 5.1.4: Stixel mit Korrektur

Eine Möglichkeit zur Verbesserung der Ergebnisse ist es die Kamerahöhe linear leicht zu verschieben. Dadurch wird bei der Freie-Fahrbahnebenen-Berechnung mehr Fläche als Boden gewertet und entsprechend höher wird nach einem Basispunkt gesucht. Experimentell wurde eine Verschiebung von 2.5 cm bestimmt,

wodurch die neue angenommene Höhe der Kamera für die Erzeugung des Disparitätsprofils der Straße 21.5 cm beträgt. Das Ergebnis der Anpassungen ist in Abbildung 5.1.4 zu sehen. Dort wird das Fahrzeug sowie das Schild korrekt von Stixeln umgeben und die Wände werden ebenfalls detektiert. Im Allgemeinen verringern diese Anpassungen die Anzahl der fehlerhaft extrahierten Stixel (vgl. Anhang A.2.2.2 mit A.2.2.3).

5.1.3 Überblick

In diesem Abschnitt wurde gezeigt, dass Stixel ebenfalls auf einem 1 zu 8 Modellfahrzeug verwendet werden können. Damit Stixel erfolgreich generiert werden können, müssen entsprechende Parameter auf das Modellfahrzeug angepasst werden.

Außerdem haben wir gesehen, dass fehlerhafte Disparitätswerte dazu führen können, dass Basispunkte von Objekten falsch gewählt werden. Zusätzlich ist die Wahl des planaren Ansatzes zur Erzeugung des Disparitätsprofils der Straße nur bedingt geeignet. Dadurch werden ebenfalls Basispunkte falsch gewählt. Durch Anpassung der Kamerahöhe kann Einfluss auf die gewählten Basispunkte genommen und dadurch das Ergebnis verbessert werden.

5.2 Objekt-Extraktion

Eine Szene wird nun mittels Stixeln abstrahiert. Diese Abstraktion ist nicht direkt für eine Klassifikation auf Bildern mittels KI geeignet. Darum werden zunächst benachbarte Stixel zusammengefasst, damit eine rechteckige Region von Interesse (ROI) auf dem linken Infrarotbild entsteht. Jede Region enthält dabei ein Objekt.

Für die Klassifikation mittels Bildern wird das Farbbild verwendet werden. Daher wird der extrahierte Bereich des linken Infrarotbildes auf den korrespondierenden Bildbereich des Farbbildes abgebildet. Dieser wird dann später zur Klassifikation genutzt und entschieden, zu welcher Klasse von Objekten dieses Objekt gehört.

Zusätzlich soll für jedes extrahierte Objekt die Position auf der Weltkarte mittels der Positionierung des Fahrzeuges berechnet werden.

5.2.1 Extraktion von Objektregionen

Für die Generierung von ROIs müssen Stixel zu möglichen Objekten zusammengefasst werden. Dabei enthält jeder Stixel die Bildkoordinaten u, v auf dem linken rektifizierten Infrarotbild sowie die zugehörige Disparität d . Mittels dieser Informationen sowie der Gleichung 2.5 (S. 10) können Punkte $p \in \mathbb{R}^3$ mit $(x, y, z) \in p$ berechnet werden. Dabei beschreibt p_t den oberen und p_b den unteren Mittelpunkt der Kante eines Stixels im 3D-Koordinatensystem der linken Infrarotkamera. Damit benachbarte Stixel zusammengefasst werden können, müssen folgende Kriterien erfüllt sein:

- Höhe des Stixels liegt im Intervall der zugelassenen Objekthöhe Min_h, Max_h :

$$Min_h < |y_t - y_b| < Max_h \quad (5.1)$$

5 Objekt-Detektion

- Distanz des Stixels zum Fahrzeug liegt im zulässigen Intervall Min_d, Max_d :

$$Min_d < \frac{z_t + z_b}{2} < Max_d \quad (5.2)$$

- Höhendifferenz ΔH zwischen benachbarten Stixeln ist erlaubt:

$$|y_i - y_{i+1}| \leq \Delta H, y = \frac{y_t + y_b}{2}, i \in \mathbb{N} \quad (5.3)$$

- Tiefendifferenz ΔZ zwischen benachbarten Stixeln ist erlaubt:

$$|z_i - z_{i+1}| \leq \Delta Z, z = \frac{z_t + z_b}{2}, i \in \mathbb{N} \quad (5.4)$$

Anhand dieser Kriterien wird die Extraktion von Objekten auf für den AADC wichtige Objekte (Puppen, Autos, Schilder) beschränkt. Erfüllen mindestens n nacheinander folgende Stixel diese Kriterien, dann wird der durch die Stixel an Position i bis Position $i + n$ beschriebene Bereich als ein mögliches Objekt aufgefasst. Aus diesen Stixeln wird im Anschluss die rechteckige Region extrahiert. Eine Region wird durch einen oberen linken pos_{ul} und unteren rechten Eckpunkt pos_{br} auf dem Infrarotbild beschrieben. Die Eckpunkte ergeben sich durch $pos_{ul} = (u_i, \max(v_i, \dots, v_{i+n}))$ und $pos_{br} = (u_{i+n}, \min(v_i, \dots, v_{i+n}))$. In der Abbildung 5.2.1 ist das Ergebnis der Extraktion zu sehen.

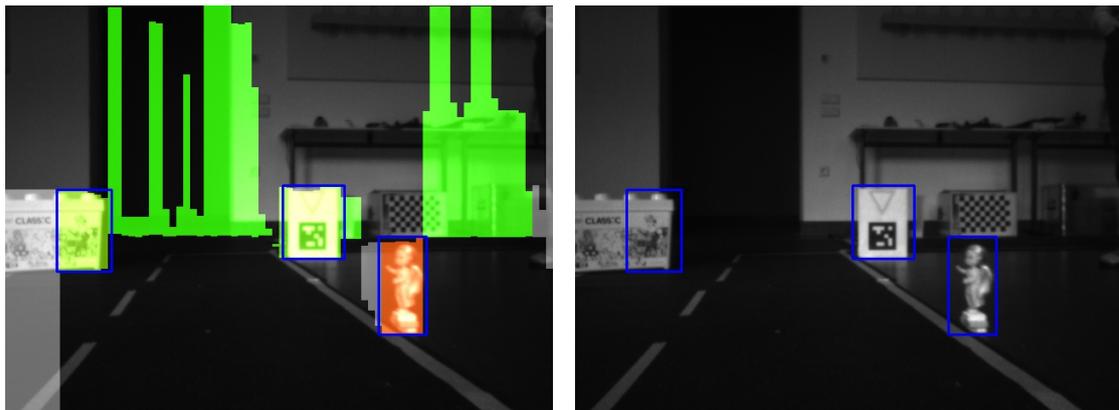


ABBILDUNG 5.2.1: Ergebnis der Objekt-Extraktion

5 Objekt-Detektion

Unter Umständen wird mit diesem Ansatz Objekte, die zu nah beieinander sind, zu einer Region zusammengefasst, da kein formbasierter Ansatz zur Bereichsextraktion gewählt wurde. Im Rahmen des AADC 2017 stehen Objekte stets genügend weit auseinander (≥ 50 cm). Weiterhin ist es möglich das Bereiche extrahiert werden, welche für den AADC unwichtige Objekte enthalten. Diese müssen durch die Klassifikation erkannt und entsprechend heraus gefiltert werden. In der Tabelle 5.2.1 sind alle gewählten und experimentell ermittelten Parameter gelistet.

Parameter	Wert
Minimale Distanz zum Objekt Min_d	0 cm
Maximale Distanz zum Objekt Max_d	300 cm
Minimale Objekt-Höhe Min_h	10 cm
Maximale Objekt-Höhe Max_h	40 cm
Erlaubte Höhendifferenz zwischen benachbarten Stixeln ΔH	10 cm
Zulässige Tiefendifferenz zwischen benachbarten Stixeln ΔZ	10 cm
Minimale Anzahl von Stixeln bei $w_S = 8$ Pixel	3

TABELLE 5.2.1: Übersicht der Parameter für die Objektextraktion aus Stixeln

5.2.2 Berechnung zugehöriger Bereiche im Farbbild

Im Anschluss an die Extraktion von Bildbereichen auf dem linken Infrarotbild muss dieser Bereich auf das rektifizierte Farbbild abgebildet werden. Damit ein Bildbereich auf das Farbbild übertragen werden kann, werden die Eckpunkte pos_{ll} und pos_{br} sowie die Disparitäten d_i und d_{i+n} des i -ten sowie des $i+n$ -ten Stixels benötigt. Weiterhin werden die intrinsischen Kameraparameter der linken Infrarot-Kamera und der RGB-Kamera benötigt. Zusätzlich werden die extrinsischen Parameter für die Translation zwischen linker Infrarot- und Farb-Kamera benötigt.

Es werden folgende Schritte durchgeführt: Mittels der Gleichung 2.5 (S. 10) und der Verwendung der intrinsischen Parameter der linken Infrarot-Kamera werden die Eckpunkte in das 3D-Koordinatensystem der linken Infrarot-Kamera projiziert. Als Ergebnis erhalten wir zwei 3D-Punkte. Im Anschluss werden diese Punkte in das 3D-Koordinatensystem der Farb-Kamera mittels Translation überführt. Dazu wird die Gleichung 2.7 (S. 15) mit einer zusätzlichen Dimension verwendet.

5 Objekt-Detektion

Anschließend erfolgt eine Deprojizierung der Punkte mittels der intrinsischen Parametern der RGB-Kamera. Dazu wird die Gleichung 2.5 entsprechend umgeformt. Als Ergebnis der Umformung erhalten wir:

$$u = \frac{x}{z} * f + c_u \quad (5.5)$$

$$v = \frac{y}{z} * f + c_v \quad (5.6)$$

Wir erhalten die korrespondierenden Eckpunkte auf dem Farbbild. In der Abbildung 5.2.2 ist ein Beispiel zu sehen.

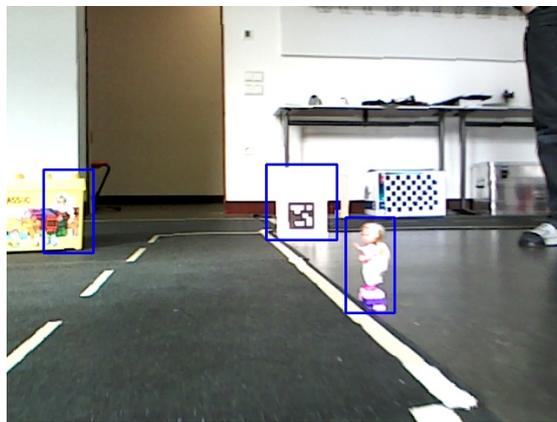


ABBILDUNG 5.2.2: Relevante Bereiche im Farbbild

5.2.3 Berechnung der Objektposition auf der Weltkarte

Im Anschluss an die Ermittlung von ROIs für mögliche Objekte wird nun die Position des Objektes auf der Weltkarte berechnet. Die resultierenden Koordinaten sind für die Rückmeldung an die Jury sowie für die Verfolgung von Objekten auf der Weltkarte von Bedeutung. Für die Berechnung der Objektposition wird die Position des Fahrzeuges auf der Weltkarte sowie der Orientierungswinkel benötigt. Diese Informationen sind durch das Positionierungsmodul gegeben.

Die Position des Objektes wird wie folgt berechnet: Zunächst wird ein Objekt durch die i -ten bis $i+n$ -ten Stixel beschrieben. Der Mittelpunkt jedes Stixels wird

5 Objekt-Detektion

anschließend mit der Gleichung 2.5 (S. 10) in das 3D-Koordinatensystem der linken Infrarot-Kamera projiziert. Mit Hilfe dieser Mittelpunkte wird der geometrische Schwerpunkt berechnet. Als Resultat erhalten wir den Punkt $p \in \mathbb{R}^3$ mit $(x, y, z) \in p$ im lokalen 3D-Koordinatensystem. Auf der 2D-Weltkarte wird die Höhe y des Objektes nicht benötigt, sodass im folgenden nur p' mit (x, z) verwendet wird.

Im nächsten Schritt wird der Punkt p' um den Punkt t_{Pre} zum Zentrum der Positionierung des Fahrzeuges verschoben (Siehe Gleichung 2.7, S. 15). Bei dem Modellfahrzeug entspricht dies einer Verschiebung von 4 cm in x -Richtung und 36 cm in z -Richtung.

Anschließend wird der neue Punkt durch die Gleichung 2.8 (S. 15) um den Orientierungswinkel α des Fahrzeuges minus $\frac{\pi}{2}$ rotiert. Die Subtraktion von $\frac{\pi}{2}$ ist notwendig, damit nach der Rotation das Fahrzeug korrekt ausgerichtet ist. Der neu erhaltene Punkt wird anschließend um die aktuelle Fahrzeugposition t_{Post} auf der Weltkarte verschoben. Als Ergebnis erhalten wir die Position des Objektes auf der Weltkarte (p_{Welt}). In Gleichung 5.7 sind die Translations und Rotationschritte zusammengefasst dargestellt.

$$p_{Welt} = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} * [(p' + t_{Pre})] + [t_{Post}] \quad \text{mit } \beta = \alpha - \frac{\pi}{2} \quad (5.7)$$

Zusätzlich wird noch der Radius des Objektes benötigt. Dieser ist gegeben durch den Abstand der 3D-Koordinaten der Ränder des Bildbereiches zum 3D-Mittelpunkt.

Validierung

Nach dem die Positionierung von Objekten auf der Weltkarte realisiert ist, wird nun ein Beispiel betrachtet. Dazu wird ein Experiment realisiert, in dem ein Fahrzeug auf einer Weltkarte auf einen Pylon zu fährt und die entsprechenden Daten aufgezeichnet. Die Position des Pylons auf der Weltkarte sowie die Position des Fahrzeuges sind bekannt. Die Abweichung in Metern des berechneten Punktes zum realen Punkt des Pylons wird mittels Gleichung 2.9 (S. 15) berechnet. Ebenfalls wird die Distanz vom Fahrzeug zur realen Position des Fahrzeuges berechnet.

5 Objekt-Detektion

In Abbildung 5.2.3 ist das Ergebnis des Versuches für 3 verschiedene Stixel-Weiten mit den aufgezeichneten Daten abgebildet.

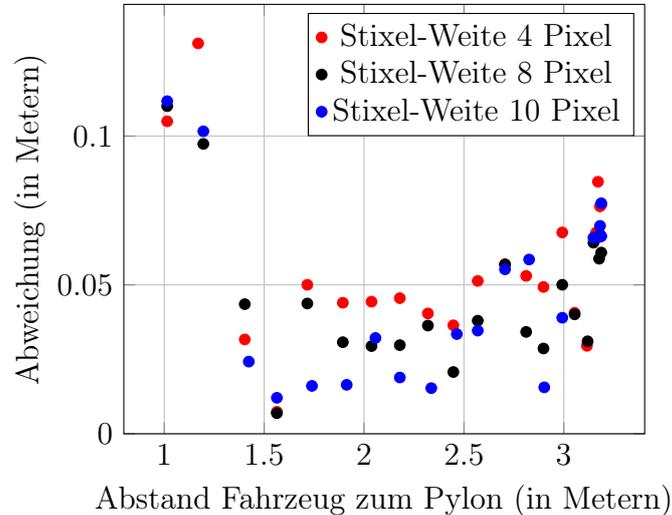


ABBILDUNG 5.2.3: Verhältnis von Fahrzeugabstand zur Positionsgenauigkeit für verschiedene Stixel-Weiten

Es ist zu sehen, dass die berechnete Position des Objektes von der erwarteten Position abweicht. Im Wesentlichen basiert der hier vorgestellte Ansatz auf der Annahme das alle Stixel in einem detektierten Bildbereich zu einem Objekt gehören. Daher unterscheiden sich die Ergebnisse in der Positionierung in Abhängigkeit der verwendeten Stixel-Weite. Weiterhin kann durch Fehldetektionen (vgl. Unterabschnitt 5.1.2, S. 30) die Größe des Bildbereiches schwanken und somit Einfluss auf die Positionierung nehmen. Insbesondere wenn Objekte nah ($\sim 1\text{m}$) am Fahrzeug sind, werden vermehrt fehlerhafte Stixel extrahiert, denn dann treten verstärkt Fehler im Disparitätsbild auf, da zwischen linkem und rechtem Infrarotbild keine Übereinstimmung gefunden wird. Verstärktes Auftreten von Disparitätsfehlern führt dazu, dass Stixel nicht korrekt die Kontur des Objektes beschreiben. Dadurch schwankt die Positionierung bei geringer Distanz stärker. In Abbildung 5.2.4 ist solch eine Situation zu sehen. Weitere nicht dokumentierte Versuche bestätigen das Ergebnis.

Die reine Verwendung von Stixeln zur Positionsbestimmung von Objekten im Rahmen des Modellfahrzeuges ist bedingt geeignet, da vor allem im Nahbereich die Ab-

5 Objekt-Detektion

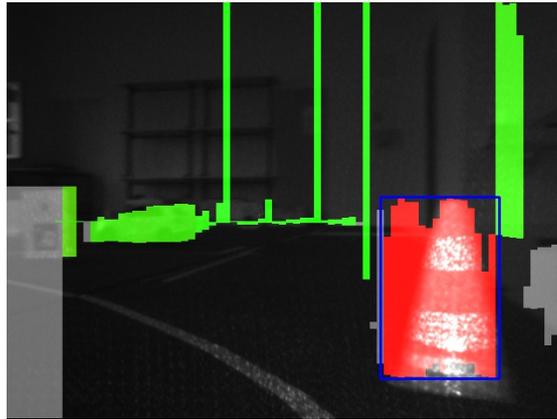


ABBILDUNG 5.2.4: Fehlerhafte Extraktion von Stixeln im Nahbereich ($w_s = 8$ Pixel)

weichung der Position von Objekten zu groß ist. Zur Verbesserung des Ergebnisses muss eine räumliche Analyse von Objekten auf dem Disparitätsbild realisiert werden, damit die Positionen genauer extrahiert werden kann. Eine Verbesserung wird im Rahmen dieser Arbeit nicht weiter betrachtet. Parallel zu der Objekt-Detektion mit Hilfe von Stixeln wurde ein anderer Ansatz entwickelt (Siehe Kapitel 6).

5.2.4 Überblick

In diesem Abschnitt wurde gezeigt, dass für eine Extraktion von Objekten Stixel verwendet werden können. Dazu wurden Kriterien vorgestellt, welche es erlauben, ein für den AADC wichtiges Objekt zu extrahieren. Als Ergebnis solch einer Extraktion entsteht ein Bildbereich auf dem linken Infrarotbild. Dieser Bildbereich kann durch die Verwendung von rektifizierten Bildern mit Hilfe einer Translation auf das zugehörige Farbbild übertragen werden.

Anhand der Fahrzeugposition sowie der Informationen aus den Stixeln für einen Bildbereich können die Position von Objekten auf der Weltkarte bestimmt werden. Es wurde gezeigt, dass die prinzipielle Positionsbestimmung von Objekten möglich ist. Jedoch wird die reine Verwendung von Stixeln zur Positionsbestimmung aufgrund von Ungenauigkeiten im Nahbereich nur bedingt empfohlen.

6 Vergleich Stixel- und Punktwolken-Ansatz

Im Rahmen dieser Arbeit wurde die Detektion und Extraktion von Objekten mittels Stixel-Ansatz betrachtet. Parallel dazu hat ein anderes Mitglied des Smart Driving Teams ein Verfahren zur Detektion und Extraktion mittels Punktwolken realisiert. Zunächst wird die Arbeitsweise des Punktwolken-Ansatzes kurz vorgestellt ohne die verwendete Verfahren genauer zu erklären. Im Anschluss werden beide Ansätze hinsichtlich der Detektion, Extraktion und Positionsbestimmung verglichen. Weiterhin werden die im Konzept geforderten Anforderungen betrachtet.

6.1 Prinzipielle Funktionsweise des Punktwolken-Ansatzes

Im Folgenden wird kurz die Funktionsweise des Punktwolken-Ansatzes dargestellt ohne diese weiter zu kommentieren [Kle18]. Für diesen Ansatz wird die Punktwolke aus der Intel RealSense R200 verwendet sowie die Programmbibliothek **P**oint **C**loud **L**ibrary (PCL) [RC11].

Zunächst wird die Anzahl der Punkte durch ein sogenannten Voxelgrid-Filter verringert. Dies geschieht in dem Punktwolken zu vielen kleinen 3D-Boxen zusammengefasst werden. Der geometrische Schwerpunkt der Box repräsentiert die zusammengefassten Punkte [OP18b]. Ausreißer werden mittels eines euklidischen Filters

entfernt [OP18a]. Im Anschluss wird der Fahrbereich aus der Punktwolke mittels achsenbeschränktem **Random Sample Consensus (RANSAC)** bestimmt [FB81]. Punkte unterhalb des Bereiches werden entfernt. Dann erfolgt die Objektlokalisierung. Mittels euklidischer Distanz werden nah beieinander liegende Punkte der Punktwolke gruppiert. Um die daraus entstehende Gruppen werden 3D-Boxen gelegt. Aus diesen 3D-Boxen wird der Bildbereich für das Farbbild sowie die Position auf der 2D-Weltkarte berechnet.

6.2 Vergleich

6.2.1 Detektion und Extraktion

Die Detektion von Stixeln erfolgt mittels der räumlichen Daten aus dem Disparitätsbild. Mit diesen abstrahierten Daten wird im Anschluss weiter gearbeitet. Im Gegensatz dazu arbeitet der Punktwolken-Ansatz durchgehend mit allen vorhandenen räumlichen Informationen.

Beim Stixel-Ansatz werden unter Umständen Straßenelemente als Objekte detektiert und entsprechend extrahiert (vgl. Unterabschnitt 5.1.2, S. 30). Der Punktwolken-Ansatz extrahiert keine Bodenelemente. Beide Verfahren extrahieren die für den AADC wichtigen Objekte wie Schild, Erwachsenen- und Kindpuppe sowie Autos, sodass eine Klassifikation erfolgreich darauf angewandt werden kann.

Zusätzlich extrahieren beide Verfahren noch weitere unrelevante Objekte. Die unwichtigen Objekte müssen bei der Klassifikation entsprechend klassifiziert werden, damit diese anschließend entfernt werden können. Für die Detektion und Extraktion von Objekten für die Klassifizierung sind beide Ansätze gleich gut geeignet.

6.2.2 Positionierung

Die Positionsbestimmung von Objekten auf der Weltkarte erfolgt bei dem Stixel-Ansatz mittels den zusammengehörenden Stixeln eines Objektes. Für diese wird

6 Vergleich Stixel- und Punktwolken-Ansatz

der geometrische Schwerpunkt aus den Stixeln berechnet. Das Punktwolken-Verfahren berechnet den geometrischen Schwerpunkt aus gruppierten räumlichen Daten. Für die Berechnung der Positionen in den Punktwolken wird einmalig der identische Versuch wie in Abschnitt 5.2.3 (S. 37) durchgeführt. In Tabelle 6.2.1 wird das Ergebnis dargestellt. Es ist zu sehen, dass die Positionierung durch die Punkt-

	Stixel	Punktwolke
Minimale Abweichung	0,6 cm	0,2 cm
Maximale Abweichung	11,0 cm	8,3 cm
Mittlere Abweichung	4,7 cm	2,7 cm

TABELLE 6.2.1: Vergleich der Positionierung vom Stixel- und Punktwolken-Ansatz

wolken deutlich genauer ist. Insbesondere ist die Abweichung von der realen Position kleiner. Aufgrund der ungenauen Positionsbestimmung im Nahbereich bei den Stixeln ist die Positionsbestimmung durch Punktwolken erheblich besser (vgl. Abschnitt 5.2.3, S. 37). Weitere nicht dokumentierte Beobachtungen im Rahmen des AADC bestätigen die ermittelte Genauigkeit der Positionsbestimmung in dem Versuch.

6.3 Betrachtung der Anforderungen

Im Folgenden sollen die im Konzept geforderten Anforderungen für beide Verfahren betrachtet und bewertet werden.

Detektion wichtiger Objekte Sowohl die Stixel als auch die Punktwolken detektieren relevante Objekte.

Geringe Fehldetektion Bei den Stixeln werden unter Umständen Bodenteile als Objekt extrahiert. Beide Verfahren detektierten einen Großteil korrekt, daher wird diese Anforderung als erfüllt angesehen.

Abbildung auf Farbbild Bildbereiche werden durch beide Verfahren korrekt auf das Farbbild abgebildet.

6 Vergleich Stixel- und Punktwolken-Ansatz

Distanz der Detektion Beide Verfahren detektieren Objekte bis zu der geforderten Distanz von 2,8 m.

Position von Objekten Prinzipiell funktioniert die Positionierung der Objekte mittels Stixeln, jedoch gibt es größere Schwankungen der Position im Nahbereich (Siehe Abschnitt 5.2.3, S. 37). Im Gegensatz dazu erzeugt die Positionsbestimmung des Punktwolken-Ansatzes ein sehr gutes Ergebnis mit der durchschnittlichen Abweichung von 2.7 cm. Daher ist die Positionsbestimmung von Objekten mittels Stixel nur bedingt erfüllt. Der Punktwolken-Ansatz erfüllt diese Anforderung.

Wiederverwendbarkeit Die Stixel benötigen für die Wiederverwendbarkeit insbesondere ein Stereo-Kamera-System, zugehörige Kameraparamter sowie rektifizierte Bilder. Der Punktwolken-Ansatz benötigt diese Informationen ebenfalls sowie ein Verfahren zur Erzeugung von Punktwolken aus dem Stereo-Kamera-System, falls diese nicht direkt Punktwolken liefern kann. Beide Verfahren sind prinzipiell in Abhängigkeit der Kamera wiederverwendbar.

Implementierungszeit Die Verfahren wurden im Rahmen des Wettbewerbes realisiert.

Performance¹ Beide Verfahren funktionieren ausreichend schnell auf dem Fahrzeug, sodass auch nur kurz sichtbare Objekte detektiert werden können. Der Stixel-Ansatz arbeitet mit etwa 8 Bilder pro Sekunde (BpS). 8 BpS sind im Rahmen des AADC noch ausreichend performant denn das Fahrzeug sowie Objekte bewegen sich langsam. Der Punktwolken-Ansatz ist deutlich leistungsfähiger als der Stixel-Ansatz. Dieser erreicht fast 60 BpS in Abhängigkeit davon ob große Flächen wie Wände vorhanden sind. Ist eine Wand vorhanden, dann benötigt der Algorithmus mehr Zeit, da eine Wand viele nah beieinander liegende Punkte in der Punktwolke generiert. Bei solch einem Leistungseinbruch arbeitet der Algorithmus mit etwa 10 BpS.

¹CPU: Intel® Core i3-6100T, Mainboard: GIGABYTE GA-Z170N-WIFI, RAM: 8 GB DDR4 PC-2133

6 Vergleich Stixel- und Punktwolken-Ansatz

Aufgrund der besseren Positionsbestimmung sowie der Leistung des Punktwolken-Ansatzes wird dieses für die Detektion von Objekten empfohlen. Für die Verwendung zwecks der Verfolgung ist der Punktwolken-Ansatz klar besser geeignet.

6.4 Überblick

In diesem Abschnitt wurde der Punktwolken-Ansatz, welcher durch ein anderes Smart Driving Teammitglied realisiert wurde, vorgestellt. Es hat sich gezeigt das beide Verfahren prinzipiell zur Objekt-Detektion geeignet sind. Jedoch ist der Punktwolken-Ansatz im Vergleich zu dem in dieser Arbeit realisierten Verfahren bei der Positionierung deutlich genauer.

Weiterhin wurden die im Konzept geforderten Anforderungen betrachtet. Im Wesentlichen erfüllen der Stixel-Ansatz sowie der Punktwolken-Ansatz die Anforderungen. Unterschiede ergeben sich im Wesentlichen in der Positionierung und in der Performance. In diesen ist der Punktwolken-Ansatz deutlich besser als der Stixel-Ansatz.

7 Objekt-Verfolgung im Weltmodell

Im Anschluss an die Objektdetektion und Berechnung der Weltkoordinaten eines Objektes wird nun die Verfolgung eines Objektes über die Zeit realisiert. In diesem Abschnitt wird ein einfaches Verfahren realisiert, welches den Anforderungen des AADC 2017 genügt. Verwendet wird auf Grund der Genauigkeit der Positionierung das Punktwolken-Verfahren (vgl. Unterabschnitt 6.2.2, S. 41).

7.1 Planung

Durch die Detektion und Extraktion von Objekten besitzt jedes Objekt eine Position auf einer Weltkarte sowie einen Radius. Weiterhin ist bekannt, dass sich Objekte im Rahmen des Wettbewerbes sehr langsam bewegen und die Entfernungen zwischen Objekten ausreichend groß sind. Zusätzlich ist eine Liste von Schildern mit Positionen gegeben. Zunächst sollen die Anforderungen für die Verfolgung von Objekten erweitert werden.

7.1.1 Konkretisierung der Anforderungen

Ziel der Verfolgung ist es Objekte einmalig an die Jury zurück zu melden. Diese Meldung an die Jury enthält nur die Position des Objektes auf der Weltkarte. Die Rückmeldung wird nicht näher erläutert, denn diese ist nach einer erfolgreichen Verfolgung von Objekten trivial.

7 Objekt-Verfolgung im Weltmodell

Aufgrund dessen, dass eine Verfolgung von Objekten auch für das Fahrzeug zur Weiterverarbeitung interessant ist, werden die Anforderungen erweitert. Objekte sollen zusätzlich in statische und dynamische unterschieden werden. Als statisches Objekt werden jene angesehen, die sich nicht bewegen können. Weiterhin sollen statische Objekte zur Weiterverarbeitung nur einmalig gesendet werden. Dynamische Objekte werden mit jeder Detektion gesendet. Objekte welche über einen längeren Zeitraum nicht gesehen werden, müssen aus der Verfolgung über die Zeit entfernt werden. Bei einer anschließenden erneuten Detektion soll dieses Objekt dann als neues behandelt werden. Zusätzlich müssen Schwankungen in der Positionierung (vgl. Unterabschnitt 6.2.2, S. 41) berücksichtigt werden.

7.1.2 Ansatz für die Realisierung

Detektierte Objekte besitzen eine Position $x, y \in \mathbb{R}^2$ auf einer zweidimensionalen Karte, welche in Metern aufgelöst ist sowie einen Radius r in Metern. Die Position entspricht dem Mittelpunkt des Objektes. Mit Hilfe dieser Daten können Objekte als Menge betrachtet werden, welche einen kreisförmigen Bereich auf einer zweidimensionalen Ebene einnehmen. Diese Menge enthält belegte Koordinaten auf dieser Ebene. Dies führt zum Begriff der **Region Connection Calculus-8 (RCC-8)**, welcher in [Ren98] betrachtet wird. RCC-8 beschreibt die Topologie zwischen Flächen mittels Acht grundlegenden Regeln. In der Abbildung 7.1.1 sind alle Regeln dargestellt [Ren98, Grafik S. 4]. Diese Regeln können kombiniert werden um wei-

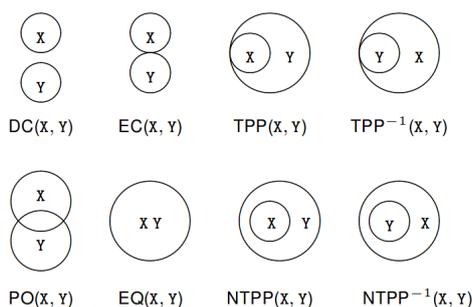


ABBILDUNG 7.1.1: Darstellung der RCC-8 Regeln für Menge X und Y

7 Objekt-Verfolgung im Weltmodell

tere Fälle abzudecken. Folgende RCC-8 Regeln können zusammengefasst werden, da eine Unterscheidung von tangentialer Berührung nicht notwendig ist:

- Objekte berühren oder überlappen: $ECPO(A, B) = EC(A, B) \cup PO(A, B)$
- Objekt ist Teilmenge: $PP(A, B) = TPP(A, B) \cup NTTP(A, B)$ beziehungsweise $PP(A, B)^{-1} = TPP(A, B)^{-1} \cup NTTP(A, B)^{-1}$

In der Tabelle 7.1.1 sind alle benötigten Relationen für die nicht leeren Mengen A und B durch Regeln der Mengenlehre und RCC-8 dargestellt. Mit Hilfe die-

	Menge	RCC-8
(a)	$A \cap B = \emptyset$	$DC(A, B)$
(b)	$(A \cap B \neq \emptyset) \wedge (A \setminus B \neq \emptyset) \wedge (B \setminus A \neq \emptyset)$	$ECPO(A, B)$
(c)	$A = B$	$EQ(A, B)$
(d)	$(A \neq B) \wedge ((A \subset B) \vee (B \subset A))$	$PP(A, B) \cup PP(A, B)^{-1}$

TABELLE 7.1.1: Topologische Fälle für zwei nicht leeren Mengen A und B

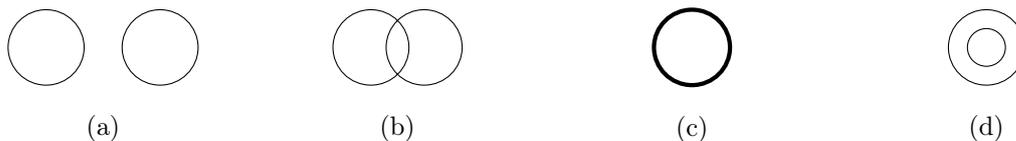


ABBILDUNG 7.1.2: Zu betrachtende Relationen zwischen zwei Mengen A und B

ser Relationen, welche in der Abbildung 7.1.2 vereinfacht dargestellt sind, kann ein einfaches Verfahren realisiert werden. Dazu werden detektierte Objekte gespeichert. Im nächsten Zeitschritt wird dann mittels der Relationen nach Übereinstimmungen zwischen detektierten und neu detektierten Objekten gesucht. Falls keine Übereinstimmung gefunden wurde muss es sich um ein neues Objekt handeln.

7.1.3 Prinzipieller Ablauf

Die Objektverfolgung wird in drei Teilschritte zerlegt. Zunächst sollen Objekte vor-gefiltert, dann klassifiziert werden. Die eigentliche Verfolgung wird erst im letzten Schritt realisiert. Im Folgenden werden die Teilschritte dargestellt:

7 Objekt-Verfolgung im Weltmodell

Vorfilterung Ein detektiertes Objekt soll zunächst mit bekannten statischen Objekten abgeglichen werden. In diesem Fall die bekannte Liste von Schildern. Falls die Objektposition mit einem Schild übereinstimmt, dann handelt es sich bei dem Objekt um ein Schild. Dieses muss nicht klassifiziert werden und kann aus der Liste der unbekanntenen Objekte entfernt werden.

Klassifizierung Vorgefilterte Objekte werden klassifiziert. Jedes Objekt enthält eine Liste von Klassen und dazugehörige Wahrscheinlichkeiten. Falls das Objekt eine Wahrscheinlichkeit von 99% in der Klasse Auto besitzt, so ist dieses Objekt sehr wahrscheinlich ein Fahrzeug. Objekte werden um diese Informationen bei der Klassifikation erweitert. Die Klassifizierung ist nicht Bestandteil der Arbeit.

Nachverfolgung Nachdem Objekte erfolgreich klassifiziert wurden, können Objekte anhand ihrer Position, Typs und Radien nachverfolgt werden. Im Anschluss kann eine Weiterverarbeitung dieser Informationen erfolgen.

7.1.4 Vor- und Nachteile

Durch den Ansatz der Verfolgung auf Weltkoordinaten wird sich erhofft, dass dieser schneller realisiert ist als andere Verfahren, welche beispielsweise auf Bildern arbeiten. Dies ist notwendig angesichts der beschränkten Zeit zur Lösung der Problemstellung innerhalb des AADC 2017. Die Wesentlichen Vor- und Nachteile sind in Tabelle 7.1.2 dargestellt.

Vorteile	Nachteile
Einfaches Verfahren	Erfordert korrekte Objektinformationen: Klasse, Radius, Position
Keine aufwändige Merkmals-Verfolgung auf Bildern	Keine Vorhersage
Schnell implementiert	

TABELLE 7.1.2: Vor- und Nachteile der Objekt-Verfolgung mittels Mengen

7.2 Realisierung

Wie in Unterabschnitt 7.1.2 (S. 46) bereits dargestellt betrachten wir Objekte als eine Menge von Punkten. Diese Menge belegt Positionen auf einer metrischen 2D-Karte. Dazu besitzt ein Objekt eine Position und einen Radius. Mit Hilfe der euklidischen Distanz $dist \in \mathbb{R}$ zwischen den Mittelpunkten von zwei nicht leeren Mengen A, B und den zugehörigen Radien r_A, r_B mit $r \in \mathbb{R}_{>0}$ können folgende Bedingungen formuliert werden, welche Aufschluss über die Relation der Objekte geben:

(a) $DC(A, B)$:

$$dist > r_A + r_B \quad (7.1)$$

(b) $ECPO(A, B)$:

$$(dist \leq r_A + r_B) \wedge (dist > 0) \wedge ((r_A < dist + r_B) \vee (r_B < dist + r_A)) \quad (7.2)$$

(c) $EQ(A, B)$:

$$(dist = 0) \wedge (r_A = r_B) \quad (7.3)$$

(d) $PP(A, B) \cup PP(A, B)^{-1}$:

$$((r_B \geq dist + r_A) \vee (r_A \geq dist + r_B)) \wedge (r_A \neq r_B) \quad (7.4)$$

In der Abbildung 7.2.1 sind die Zusammenhänge vereinfacht dargestellt. Die gestrichelte Linie entspricht den Radien der Kreise und die punktierte Linie der Distanz von Mittelpunkt zu Mittelpunkt. Als zusätzliche Einschränkung gilt, dass nur Objekte ins Verhältnis gesetzt werden dürfen, welche einen identischen Klassifizierten-Typ besitzen. Mit Hilfe dieses Ansatzes können nun die drei Teilschritte betrachtet werden.

7 Objekt-Verfolgung im Weltmodell

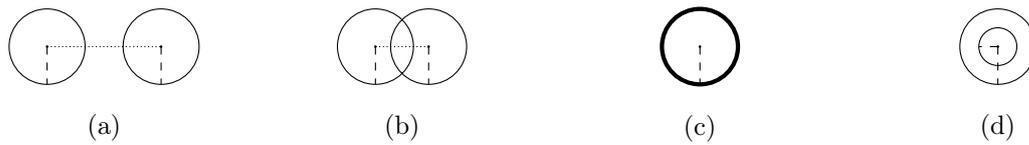


ABBILDUNG 7.2.1: Relation zwischen zwei Objekten anhand der Distanz und Radien

7.2.1 Vorfilterung

Durch eine Vorfilterung der Objekte ist es möglich, die Last für die nachfolgende Klassifikation zu verringern. Im Wesentlichen werden dazu statische bekannte Objekte gefiltert. Mit Hilfe des in Abschnitt 7.2 vorgestellten Ansatzes werden die Positionen von Objekten mit den Positionen aus der Schilderliste vom AADC verglichen. Ist eine der Gleichung 7.2, 7.3 oder 7.4 erfüllt, dann wird dieses Objekt aus der Liste der zu klassifizierenden Objekte entfernt, denn Schilder müssen nicht separat klassifiziert werden. Insbesondere werden Schilder durch Algorithmen der Schilddetektion anderweitig bereits behandelt (nicht Bestandteil dieser Arbeit). Im Anschluss werden die vorgefilterten Objekte an den Klassifikator weiter gereicht.

7.2.2 Objekt-Klassifikation

Die eigentliche Klassifikation ist nicht Bestandteil dieser Arbeit. Mit Hilfe der Koordinaten des ROIs eines Objektes auf dem Farbbild wird dieses klassifiziert. Als Ergebnis der Klassifikation wurden die Objekte um die Gewichte Ihrer zu klassifizierenden Klassen erweitert. Das Gewicht beschreibt die Wahrscheinlichkeit mit der ein Objekt zu einer Klasse zugeordnet wird. Im Rahmen des AADC 2017 sind dies folgende Klassen: Pylon, Kind, Erwachsen, Auto, Straße, Schild sowie „Nichts“. Die Klasse „Nichts“ erhalten Objekte, welche nicht einer der anderen Klassen zugeordnet werden kann. Die akkumulierten Wahrscheinlichkeiten der Klassen eines Objektes entsprechen 100%. In Abbildung 7.2.2 ist das Ergebnis einer angewendeten Klassifizierung zu sehen.

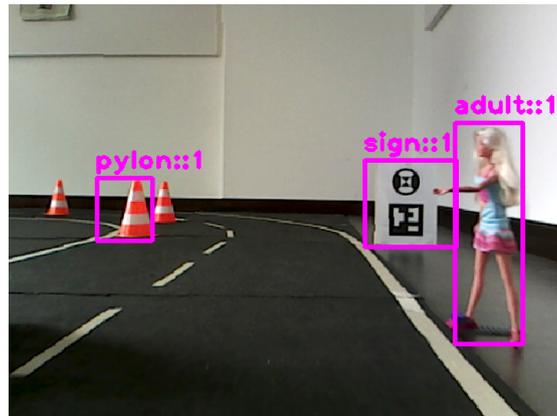


ABBILDUNG 7.2.2: Ergebnis einer angewendeten Klassifizierung

7.2.3 Objekt-Verfolgung

Nach der Vorfilterung sowie Klassifizierung wird nun die Verfolgung realisiert. Dies läuft wie folgt ab: Es wird mit einer Liste von klassifizierten Objekten gearbeitet. Objekte, deren höchste auftretenden Wahrscheinlichkeit unter einem Schwellwert liegen (beispielsweise 80%), werden entfernt, da keine genaue Aussage über die Klasse des Objektes gemacht werden kann. Objekte, welche über dem Schwellwert liegen, werden weiterhin untersucht ob sie als Straße, Schild oder „Nichts“ klassifiziert wurden. Objekte mit diesen Typen werden entfernt, da diese nicht für die Verfolgung von Bedeutung sind.

Die resultierende Liste wird als nachzuverfolgende Objekte gespeichert und auf die nächste Liste von klassifizierten Objekten gewartet. Nun kommt der in den Gleichungen 7.1 bis 7.4 dargestellte Ansatz zum Tragen. Mit Hilfe der Relation von Kreisen, welche den Objekten mit Ihren Radien entsprechen, werden Aussagen über die Zugehörigkeit von Objekten realisiert. Dazu werden Objekte in der Liste der nachzuverfolgende Objekte mit der neuen empfangen Liste verglichen. Die Liste der nachzuverfolgenden Objekte wird mit jedem Zeitschritt aktualisiert und erweitert.

Objekte mit gleichem Typ werden zu einem Objekt zusammengefasst, wenn sie die Gleichung 7.2 oder 7.4 erfüllen. Sollte das Objekt ein statisches sein und eine dieser Bedingungen erfüllt sein, dann wird die ursprüngliche Position beibehalten

7 Objekt-Verfolgung im Weltmodell

und der Radius aktualisiert. Um sicher zu gehen, dass Objekte aus verschiedenen Blickwinkeln zusammengefasst werden können, wird stets der größte Radius verwendet. Bei einem dynamischen Objekt wird bei Gleichung 7.2 die aktuelle Position übernommen und der Radius wie bei dem statischen Objekt übernommen. Andernfalls wird es wie ein statisches Objekt behandelt. Die Gleichung 7.3 ist ein Spezialfall von 7.4. Wenn dieser Fall eintritt können Objekte ohne Anpassungen zusammengefasst werden. Statische Objekte werden nur einmalig zur Weiterverarbeitung gemeldet und dynamische mit jeder Detektion.

Wird ein Objekt nicht in der Liste der nachzuverfolgenden Objekte gefunden, dann wird in einem nahen Umkreis nach einem Treffer gesucht. Dies ist notwendig falls sich Objekte zu schnell bewegen und dadurch keine Überschneidung stattfand sowie um Fehler in der Positionierung auszugleichen. Ein so gefundenes Objekt wird entsprechend der Überschneidung behandelt. Wird kein zugehöriges Objekt gefunden, dann wird das Objekt als neues Objekt in die Liste der nachzuverfolgenden Objekte hinzugefügt und an die Jury gemeldet. Im Rahmen des AADC hat sich der experimentell bestimmte Wert von 15 Zentimetern als geeignet ergeben. In der Abbildung 7.2.3 ist der Ablauf der Objekt-Verfolgung vereinfacht dargestellt.

Gleichzeitig besitzt jedes gesehene Objekt einen Zeitstempel, welcher mit jeder Übereinstimmung aktualisiert wird. Objekte, welche eine Zeit lang nicht gesehen werden, werden aus der Liste entfernt, damit die Liste der nachverfolgten Objekte möglichst klein ist. Im Rahmen des AADC muss die Zeitspanne größer als 60 Sekunden sein, damit ein Objekt entfernt wird. Dies ist damit begründet, dass die Laufzeit des nicht optimierten Algorithmus im schlechtesten Fall quadratisch ist, da jedes Element mit jedem verglichen wird. Werden die Objekte später erneut gesehen, so werden diese als neu behandelt. Dieses Vorgehen ist im Rahmen des AADC 2017 tolerabel. Im Anhang A.4 ist schematisch der Ablauf einer Objekt-Verfolgung dargestellt.

7 Objekt-Verfolgung im Weltmodell

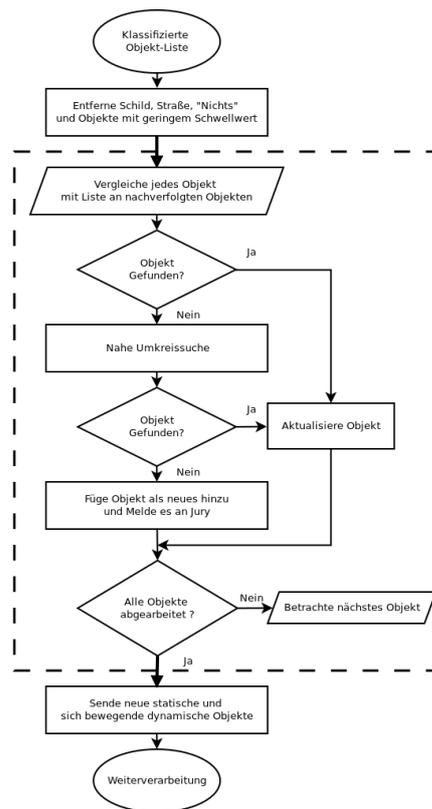


ABBILDUNG 7.2.3: Prinzipieller Ablauf der Objektverfolgung

7.2.4 Weiterverarbeitung

Statische und Dynamische Objekte werden im Anschluss an die Nachverfolgung als Information weitergereicht. Jedes Objekt, welches zur Weiterverarbeitung übertragen wird, besitzt folgende Informationen:

- Radius des Objektes auf der Weltkarte
- 2D-Position auf der Weltkarte
- Dynamisches/Statisches Objekt
- Klasse des Objektes (Klasse mit höchster Wahrscheinlichkeit)
- Wurde Objekt neu erfasst

Damit bei der Weiterverarbeitung die Unterscheidung möglich ist, ob ein neues dynamisches Objekt hinzugekommen ist oder ob es ein bereits erfasstes Objekt ist, besitzt jedes Objekt eine entsprechende Information.

Die zur Weiterverarbeitung übertragenen Objekte können beispielsweise in eine Umgebungskarte eingezeichnet werden. Weiterhin kann die Verhaltenslogik anhand der Objekte entscheiden, wie das Fahrzeug reagieren muss. Als Beispiel sei hier das Heranfahen an einen Zebrastreifen genannt.

7.3 Grenzen des Verfahrens

Der hier dargestellte Ansatz zur Verfolgung von Objekten unterliegt mehreren Einschränkungen. Zum Einen müssen Objekte gleicher Klasse genügend weit auseinander stehen. Zum Anderen muss die Abweichung von der realen Position sowie dem realen Radius des Objektes möglichst gering sein. Stehen mehrere Objekte gleicher Klasse in unmittelbarer Nähe zueinander, so werden diese Objekte zusammengefasst. Weiterhin ist es notwendig, dass Objekte der korrekten Klasse zugeordnet werden, damit ein Auto nicht fälschlicherweise als Schild klassifiziert wird. In der hier verwendeten Klassifizierung von Objekten kommt es vor, dass Objekte falsch zugeordnet werden und ein hoher Wert für die Wahrscheinlichkeit dieser Klasse existiert. Dadurch kann es passieren, dass Objekte verschiedener Klassen an einer identischen Position sind. Zur Sicherheit werden diese Objekte als neues Objekt hinzugefügt damit auch bei einem Fehler der Klassifizierung eine Objektverfolgung möglich ist. Ebenfalls wird das Objekt entsprechend mehrfach an die Jury gemeldet. Dieser Fehler wird im Rahmen des AADC 2017 in Kauf genommen, da dieser selten auftritt, denn meist wird bei Fehlklassifizierung der gesetzte Schwellwert (80%) nicht erreicht (vgl. Unterabschnitt 7.2.3, S. 51). Im Rahmen des AADC hat sich dieser einfache Ansatz als ausreichend bewährt, da Objekte gleicher Klasse immer eine gewisse Distanz zueinander haben sowie Fehler in der Position von Objekten durch die Umkreissuche abgeschwächt werden. Für die Übertragung auf das reale autonome Fahren ist eine restriktivere Herangehensweise notwendig damit Gefahren minimiert werden.

7.4 Überblick

In diesem Abschnitt wurde das Verfahren der Verfolgung von Objekten über die Zeit vorgestellt. Zunächst wurden die Anforderungen für die Verfolgung erweitert und die Idee aufgezeigt. Objekte werden anhand von Position sowie Objektradius als Menge von Punkten auf einer zweidimensionalen Weltkarte betrachtet. Mittels Regeln der Mengenlehre werden diese Mengen betrachtet und geprüft in welchem Verhältnis identisch klassifizierte Objekte über die Zeit stehen. Berühren sich beispielsweise identisch klassifizierte Objekte, dann wird angenommen, dass es sich um das gleiche Objekt handelt. Weiterhin wird dadurch die Bewegung eines dynamischen Objektes überprüft.

In zukünftigen Wettbewerben sollte es möglich sein, diesen Ansatz erneut zu verwenden. Für den realen Einsatz ist dieses Vorgehen nicht geeignet und muss entsprechend verbessert werden.

8 Schlussbetrachtung

Das Ziel dieser Arbeit war es, eine Möglichkeit zu schaffen, Objekte im Rahmen des AADC 2017 zu detektieren und über die Zeit zu verfolgen. Ausgangspunkt für diese Problemstellung war das vorgegebene 1 zu 8 Modellfahrzeug von Audi. Insbesondere ist für die Lösung dieser Problemstellung die 3D-Kamera Intel RealSense R200 von Bedeutung. Aus dieser können dreidimensionale Informationen der Umgebung gewonnen werden.

Für die Detektion von Objekten mit Hilfe der Intel RealSense R200 sind im Wesentlichen zwei Ansätze möglich. Zum Einen die Extraktion von dreidimensionalen Informationen per Disparitätsbild und zum Anderen per Punktwolken.

Zunächst wurde ein Konzept zur Lösung der Problemstellung vorgeschlagen und im Anschluss durch die Wahl eines Verfahrens verfeinert. Es wurde sich für die Verwendung des Disparitätsbildes als Grundlage für die Detektion von Objekten entschieden. Dies führte zum Begriff der Stixel. Diese abstrahieren dreidimensionale Daten eines Disparitätsbildes auf eine einfache Datenstruktur.

Es wurde im Verlauf der Arbeit gezeigt, dass der Stixel-Ansatz durch Anpassung von Parametern auf einem 1 zu 8 Modellfahrzeug verwendet werden kann. Aufgrund des kleinen Maßstabes kann es vorkommen, dass kleinere Fehler im Disparitätsbild dazu führen, dass Stixel fehlerhaft extrahiert werden. Zusätzlich erwies sich die Wahl des planaren Ansatzes zur Erzeugung des Disparitätsprofils der Straße als bedingt geeignet.

Auf Grundlage der generierten Stixel wurde die Detektion von Objekten realisiert. Dazu werden die Stixel entsprechend ihrer enthaltenen Informationen analysiert und gegebenenfalls zu Objekten zusammengefasst. Solche Objekte beschreiben einen

8 Schlussbetrachtung

bestimmten Bereich auf einem Bild. Dieser Bereich wird auf das zugehörige Farbbild übertragen. Dadurch kann eine spätere Klassifikation von Objekten auf den Bereichen eines Farbbildes realisiert werden. Die Klassifikation wurde im Rahmen dieser Arbeit nicht betrachtet. Ebenfalls kann aus den zusammengefassten Stixeln die Position und Radius des Objektes auf einer Weltkarte ermittelt werden.

Ein Vergleich des Punktwolken-Ansatzes mit dem Stixel-Ansatz hat Unterschiede in der Qualität der Ergebnisse gezeigt. Insbesondere kommt es bei dem Vorgehen der Positionsbestimmung mittels Stixel zu stärkeren Positionsabweichungen von der realen Position. Begründet ist die Abweichung durch die fehlende räumliche Analyse von Punkten auf dem Disparitätsbild sowie auftretende Fehlextraktion der Stixel. Zusätzlich arbeitet der Punktwolken-Ansatz im Allgemeinen effizienter als der Stixel-Ansatz. Dennoch kann prinzipiell ebenfalls der Stixel-Ansatz auf einem Modellfahrzeug verwendet werden. Für den AADC 2017 wurde sich jedoch gegen den vorgestellten Stixel-Ansatz und für die Objekt-Verfolgung Anhand des Punktwolken-Ansatzes entschieden.

Mit Hilfe von klassifizierten Objekten und deren Position sowie Radius wurde ein Verfahren zur Verfolgung von Objekten auf einer zweidimensionalen Weltkarte vorgestellt. Durch die Analyse von kreisförmigen Mengen und deren Positionen über die Zeit können Aussagen über die Zugehörigkeit von detektierten Objekten gemacht werden. Für den realen Einsatz im autonomen Fahren muss dieser Ansatz jedoch weiter verbessert werden. Die anschließende Weiterverarbeitung der gewonnenen Informationen wurde im Rahmen dieser Arbeit nicht betrachtet.

Alle vorgestellten Lösungen sind für die Anwendung im Rahmen des AADC ausgerichtet. Durch weitere Verbesserungen sollte es jedoch möglich sein, diese auf das reale autonome Fahren zu übertragen. Im Anhang A.6 ist die komplette Modulübersicht der gelösten Problemstellung für den AADC 2017 dargestellt.

8.1 Ausblick

Die Entwicklung und Implementierung der Module ist abgeschlossen. Das Ziel der Entwicklung war es stets, dass die Module ohne größere Anpassungen im nächsten AADC wiederverwendet werden können. Daher ist die Wiederverwendung möglich.

Obwohl die Entwicklung abgeschlossen ist können die vorgestellten Ansätze weiter verfeinert werden. Dies obliegt nachfolgenden Mitgliedern des Smart Driving Teams der HTWK. Zum Einen sollte die Fehlextraktionen von Stixeln weiter vermindert werden und zum Anderen die Positionsbestimmung von Objekten verbessert werden. Fehlextraktionen können beispielsweise vermindert werden in dem ein geeigneteres Verfahren zur Erzeugung des Disparitätsprofils gewählt wird. Außerdem könnten die verwendeten statischen Stixel zu dynamischen Stixel erweitert werden.

Weiterhin ist es denkbar, die Verfolgung von Objekten durch bestimmte Bedingungen zu verbessern, beispielsweise durch die Berücksichtigung der Objekthöhe. Durch die Erweiterung von Bedingungen und Verbesserungen im Algorithmus sollte es möglich sein, dieses Verfahren auf reales autonomes Fahren zu übertragen. Ebenfalls denkbar ist es, weitere Informationen von anderen Sensoren in die Algorithmen einfließen zu lassen, die sogenannte Sensor-Fusion, damit die Qualität der Ergebnisse weiter verbessert werden.

Alles in allem lässt sich festhalten, dass mit dieser Arbeit eine Möglichkeit geschaffen wurde, Objekte zu detektieren und über die Zeit zu verfolgen. Dies sollte den Arbeitsaufwand für den nächsten AADC in diesen Bereichen verringern. Dadurch kann mehr Zeit für andere wichtige Teilaufgaben aufgewendet werden.

8.2 Persönliches Fazit

Im Rahmen dieser Arbeit habe ich mich intensiv mit Detektion und Verfolgung von Objekten befasst. Ich verstehe nun warum der Punktwolken-Ansatz große

8 Schlussbetrachtung

Bedeutung in der Forschung des autonomen Fahrens hat. Jedoch ist insbesondere die Detektion von Objekten mittels Kameras äußerst komplex und eben darum ein großes Forschungsgebiet.

Weiterhin habe ich durch die Teilnahme am AADC 2017 viel gelernt und meine Fähigkeiten in einem Team zu agieren weiter verbessert. Leider hat es im AADC 2017 nicht für die vorderen Plätze gereicht. Es ist aber bereits ein Erfolg den ersten Platz für den wissenschaftlichen Vortrag zu belegen sowie eine bessere Platzierung als bei den vorherige Teilnahmen erreicht zu haben. Leider ist es mir nicht möglich im nächsten AADC mitzuwirken, da ausschließlich die Teilname von Studenten und Doktoranden erlaubt ist. Ich wünsche den nächsten Teams der HTWK viel Erfolg.

A Anhang

A.1 Parameter

A.1.1 Kamera

Intrinsische und extrinsische Parameter der verwendeten Intel RealSense R200 bei einer konfigurierten Auflösung von 640×480 Pixeln:

Parameter	Wert
Brennweite der linken Infrarot-Kamera	$f_u = 617.059$ $f_v = 617.059$
Optisches Zentrum der linken Infrarot-Kamera	$c_u = 319.500$ $c_v = 238.048$
Brennweite der Farb-Kamera	$f_u = 624.795$ $f_v = 624.795$
Optisches Zentrum der Farb-Kamera	$c_u = 324.873$ $c_v = 210.238$
Translation (in cm) zwischen linker Infrarot- und Farb-Kamera	$x = -5.8937000$ $y = -0.0114893$ $z = -0.0536086$

A.1.2 Stixel

Übersicht über die verwendeten Parameter der Stixel aus [PF11] und [BFM07]. Im Rahmen des AADC wurden alle Werte auf $\frac{1}{8}$ der empfohlenen Werte angepasst.

Parametername	Empfohlener Wert	Angepasster Wert
C_s^{fs}	2	0.25
T_s^{fs}	5 m	0.625 m
h_o	1 m	0.125 m
C_s^{hs}	8	1
N_Z	5 m	0.625 m
ΔZ_u	2 m	0.25 m

A.2 Aufnahmen

Im folgenden werden die auf der beiliegenden CD enthaltenen Videos kurz vorgestellt. Es existieren Aufnahmen von 2 verschiedenen Strecken, welche in dieser Arbeit verwendet wurden. Jede Strecke besitzt mehrere Videos, welche wie folgt gegliedert sind. Hinter jedem Video ist der Dateiname ohne Endung angegeben.

A.2.1 Ausgangsdaten

In diesem Ordner sind die durch die Intel RealSense R200 erzeugten Videos enthalten. Diese bilden die Grundlage für die folgenden Schritte. Folgende Aufzeichnungen befinden sich in diesem Ordner:

1. linke Infrarot-Kamera (**leftIR**)
2. rechte Infrarot-Kamera (**rightIR**)
3. Farb-Kamera (**RGB**)

4. Generierte Disparitätsdaten per SGBM aus linkem und rechtem Infrarotbild (**Disparity**)

A.2.2 Stixel-Extraktion

In diesem Ordner sind die durch die Anwendung der Stixel generierten Videos enthalten:

1. Stixel mit Originalparametern (**StixelMitOriginalParametern**)
2. Stixel mit angepassten Parametern ohne Korrektur (**StixelAngepasstOhneKorrektur**)
3. Stixel mit angepassten Parametern mit Korrektur (**StixelAngepasstMitKorrektur**)

A.2.3 Objekt-Detektion

In diesem Ordner sind die Ergebnisse der Objekt-Detektion als Videos enthalten:

1. Stixel und ROIs auf Infrarotvideo (**StixelUndROIaufIR**)
2. ROIs auf Infrarotvideo (**ROIaufIR**)
3. ROIs auf Farbvideos (**ROIaufRGB**)

A.3 Daten des Pylon-Experiments

Auf der CD befinden sich die ermittelten Daten aus diesem Experiment als Comma-Separated-Values-Datei für:

1. Stixel mit Stixel-Weite 4 (**Stixel4**)
2. Stixel mit Stixel-Weite 8 (**Stixel8**)

3. Stixel mit Stixel-Weite 10 (**Stixel10**)

4. Punktwolke (**Punktwolke**)

A.4 Objekt-Verfolgung

Dieser Ordner auf der CD enthält ein Video namens **SchematischeDarstellung-DerObjektVerfolgung**, welches schematisch das Prinzip der Verfolgung von Objekten zeigt. Die Ausgangsdaten wurden programmatisch erzeugt und der Algorithmus darauf angewandt. In diesem Video ist zu sehen, dass zunächst die drei Objekte Auto (blau), Pylon (rot) und Person (grün) detektiert werden. Kennzeich gemacht durch Darstellung von farbig gefüllten Kreisen. Im nächsten Zeitschritt werden bei erneuter Detektion die Objekte aktualisiert. Da Objekte bereits detektiert wurden, werden diese nun ohne gefüllte Kreise dargestellt. Über die Zeit bewegen sich die Person und das Fahrzeug weiter. Auszugsweise sind ebenfalls einzelne Schritte als Bilder dargestellt.

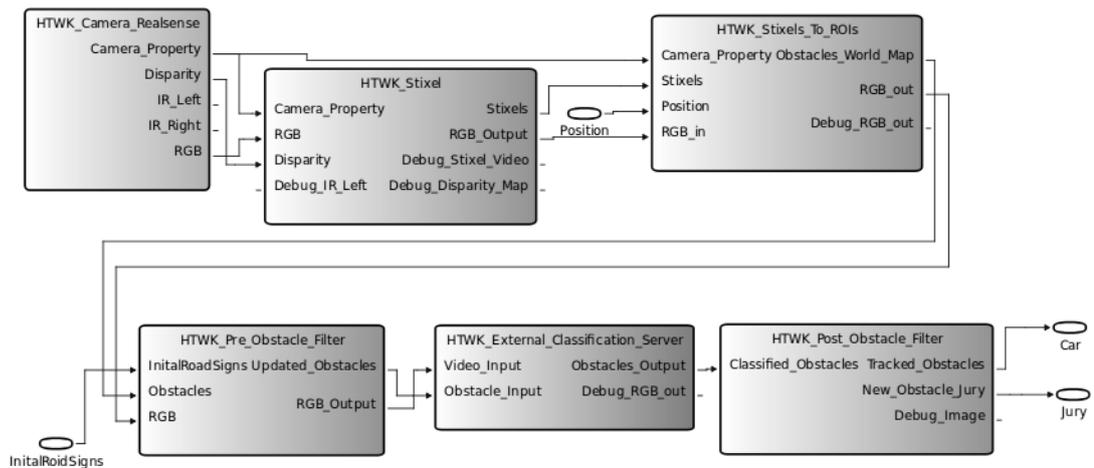
A.5 Quellcode

Der im Rahmen des AADC geschriebene Quellcode zur Lösung der in dieser Arbeit beschriebenen Aufgaben mittels ADTF befindet sich auf der CD.

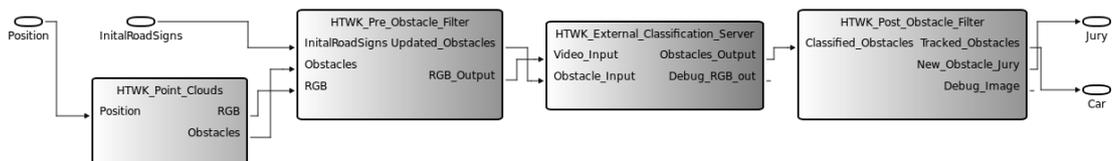
A.6 Konfiguration in ADTF

Die im Verlauf dieser Arbeit realisierte Aufgaben werden in ADTF wie folgt konfiguriert:

Stixel



Punktwolke



Glossar

AADC Audi **A**utonomous **D**riving **C**up ist ein Wettbewerb zum Thema des autonomen Fahrens

ADTF Automotive **D**ata and **T**ime-Triggered **F**ramework ist eine grafische Umgebung in der programmierte Module grafisch verknüpft werden

API Application **P**rogrammable **I**nterface bezeichnet eine Programmierschnittstelle zu einer Programmbibliothek, siehe Programmbibliothek

b **B**asislinie ist der Abstand zwischen zwei Kameras in einem Stereo-Kamera-System, siehe Extrinsische Parameter

Basispunkt Kennzeichnet den Bildpunkt an der ein Objekt auf der Fahrbahn steht

BpS Beschreibt wie viele **B**ilder **p**ro **S**ekunde komplett abgearbeitet werden können

c zentraler Punkt der Kamera (optisches Zentrum), siehe Intrinsische Parameter

d siehe Disparität

DC **D**isconnected, die Schnittmenge zwischen 2 Mengen ist die Leermenge, siehe RCC-8

Disparität ist der Abstand eines korrespondierendes Bildpunktes zwischen linkem und rechtem rektifizierten Bild, siehe Disparitätsbild

Glossar

Disparitätsbild enthält für jeden Punkt des Bildes die Tiefeninformation in Form einer Disparität

Disparitätsprofil beschreibt die Beschaffenheit der Oberfläche der Fahrbahn mittels Disparitätswerten

EC **E**xternally **C**onnected, die Schnittmenge zwischen 2 Mengen enthält 1 Element, siehe RCC-8

ECPO ist die Vereinigung von **EC** und **PO**, siehe EC sowie PO

EQ **E**qual, 2 Mengen sind identisch, siehe RCC-8

Extrinsische Parameter beschreiben den Zusammenhang des 3D-Stereo-Systems der Kamera und dem 3D-Weltkoordinatensystem

f fokale Länge ist der Abstand der Linse vom Brennpunkt (auch Brennweite), siehe Intrinsische Parameter

geometrische Schwerpunkt Der Mittelpunkt eines Objektes, man erhält diesen Punkt durch Mittelung aller Punkte innerhalb eines Objektes

HTWK Hochschule für **T**echnik, **W**irtschaft und **K**ultur

Intrinsische Parameter beschreiben den Zusammenhanges zwischen Kamerakordinatensystem und Kamerabild

Kalman-Filter ist ein Filter zur Zustandsschätzung linearer und nichtlinearer Systeme

KI **K**ünstliche **I**ntelligenz beschreibt den Versuch einem Computer eine dem Menschen ähnliche Intelligenz beizubringen

Klassifikation Mittels Klassifikation werden Objekte in Kategorien eingeteilt (Kategorie Auto, Kind, Erwachsener, ...)

Glossar

Korrespondierende Bildpunkte beschreiben eine identischen Bildstelle zwischen Bildern aus verschiedenen Blickwinkeln

NTPP **N**on **T**angential **P**roper **P**art, eine Menge liegt vollständig in der größeren Menge, siehe RCC-8

OpenCV **O**pen **S**ource **C**omputer **V**ision **L**ibrary ist eine Programmbibliothek, welche viele Algorithmen für die digitale Bildverarbeitung implementiert

PCL **P**oint **C**loud **L**ibrary ist eine Programmbibliothek, welche viele Algorithmen für das Arbeiten mit Punktwolken implementiert

PO **P**artially **O**verlapping, 2 Mengen überlappen teilweise, siehe RCC-8

PP **P**roper **P**art ist die Vereinigung von TPP und NTPP, siehe TPP sowie NTPP

Programmbibliothek ist in einer Programmiersprache geschrieben und enthält üblicherweise eine Sammlung von Funktionen

Punktwolke ist eine Menge von Punkten eines Vektorraumes die eine unorganisierte räumliche Struktur aufweisen

R200 Abkürzung für die 3D-Kamera Intel RealSense R200

RANSAC **R**andom **S**ample **C**onsensus ist ein Verfahren um mittels zufälligen Stichproben ein Modell zu schätzen

RCC-8 **R**egion **C**onnection **C**alculus-**8** beschreibt die Topologie zwischen Mengen mittels 8 grundlegender Regeln

Rektifizierung Durch eine Rektifizierung werden geometrisch entzerrte Bilder eines Stereo-Systems so ausgerichtet das eine Bildzeile zwischen linkem und rechtem Bild austauschbar verwendbar ist

ROI **R**egion **O**f **I**nterest ist ein rechteckiger Bereich auf einem Bild, welcher von Relevanz ist

Glossar

SGBM **S**emiglobal-**B**lock-**M**atching ein Verfahren zur Erzeugung eines Disparitätsbildes aus rektifizierten Stereo-Bildern mittels Energieminimierung auf diesen Bildern

Stixel ist eine rechteckige 3D-Datenstruktur, welche einen gewissen Bereich auf einem Bild beschreibt und die eigentlichen Bilddaten abstrahiert

TPP **T**angential **P**roper **P**art, eine Menge liegt in der anderen und berührt an einem Punkt die Kante der größeren Menge, siehe RCC-8

u beschreibt in dieser Arbeit die Spalte auf einem digitalem Bild

v beschreibt in dieser Arbeit die Zeile auf einem digitalem Bild

Literaturverzeichnis

- [AGa] Audi AG. *Audi Autonomous Driving Cup Regelwerk 2017*. URL: https://www.audi-autonomous-driving-cup.com/wp-content/uploads/2017/03/2017-03-20_Regelwerk_2017_V1.0_DACHL.pdf (besucht am 13. 12. 2017).
- [AGb] Audi AG. *Vorgegebener Quellcode des AADC 2017*.
- [BFM07] Hernan Badino, Uwe Franke und Rudolf Mester. „Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming“. In: *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil* (Okt. 2007).
- [BPF09] Hernan Badino, David Pfeiffer und Uwe Franke. „The Stixel World - A Compact Medium Level Representation of the 3D-World“. In: *DAGM Symposium, Jena, Germany* (2009).
- [Che+16] Xiaozhi Chen u. a. „Multi-View 3D Object Detection Network for Autonomous Driving“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Nov. 2016).
- [FB81] Martin A. Fischler und Robert C. Bolles. „Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography“. In: *Commun. ACM* 24 (1981), S. 381–395.
- [FI11] Andrea Fusiello und Luca Irsara. „Quasi-Euclidean Epipolar Rectification of Uncalibrated Images“. In: *Mach. Vis. Appl.* 22 (Juli 2011), S. 663–670.

Literaturverzeichnis

- [HLU05] Zhencheng Hu, Francisco Lamosa und Keiichi Uchimura. „A complete U-V-disparity study for stereovision based 3D driving environment analysis“. In: *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM* (Juli 2005), S. 204 –211.
- [Hir08] Heiko Hirschmuller. „Stereo processing by semiglobal matching and mutual information“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008), S. 328 –341.
- [How+17] Andrew G. Howard u. a. „MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications“. In: *CoRR* abs/1704.04861 (2017). arXiv: [1704.04861](https://arxiv.org/abs/1704.04861). URL: <http://arxiv.org/abs/1704.04861>.
- [Kes+17] Leonid Keselman u. a. „Intel RealSense Stereoscopic Depth Cameras“. In: *CoRR* abs/1705.05548 (2017). arXiv: [1705.05548](https://arxiv.org/abs/1705.05548). URL: <http://arxiv.org/abs/1705.05548>.
- [Kle18] HTWK Smart Driving Team: Philipp Kleinhenz. *Persönliche Kommunikation*. Jan. 2018.
- [LAT02] R. Labayrade, D. Aubert und J. P. Tarel. „Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation“. In: *Intelligent Vehicle Symposium, 2002. IEEE*. Bd. 2. Juni 2002, 646–651 vol.2. DOI: [10.1109/IVS.2002.1188024](https://doi.org/10.1109/IVS.2002.1188024).
- [MRF11] T. Müller, C. Rabe und U. Franke. „Dense6D - Position und Bewegung robust für jeden Bildpunkt“. In: *Proceedings of the 7th Workshop Fahrerassistenzsysteme, Walting, Deutschland* (2011).
- [OP18a] Inc. Open Perception. *Documentation Euclidean Cluster Extraction*. 2018. URL: http://pointclouds.org/documentation/tutorials/cluster_extraction.php (besucht am 29.01.2018).
- [OP18b] Inc. Open Perception. *Downsampling a PointCloud using a Voxel-Grid filter*. 2018. URL: http://pointclouds.org/documentation/tutorials/voxel_grid.php (besucht am 29.01.2018).

Literaturverzeichnis

- [PF10] David Pfeiffer und Uwe Franke. „Efficient Representation of Traffic Scenes by Means of Dynamic Stixels“. In: *IEEE Intelligent Vehicles Symposium, San Diego, CA, USA* (2010).
- [PF11] David Pfeiffer und Uwe Franke. „Modeling Dynamic 3D Environments by Means of The Stixel World“. In: 3 (Sep. 2011), S. 24–36.
- [Pfe11] David Pfeiffer. „The Stixel World - A Compact Medium Level Representation for Efficiently Modeling Dynamic Three dimensional Environments“. Dissertation. Humboldt-Universität, Berlin, 2011.
- [RC11] Radu Bogdan Rusu und Steve Cousins. „3D is here: Point Cloud Library (PCL)“. In: Shanghai, China, Mai 2011.
- [Rea16a] Intel® RealSense™. *Intel RealSense Camera R200 Product Datasheet*. 2016. URL: http://www.mouser.com/pdfdocs/intel_realsense_camera_r200.pdf (besucht am 13. 12. 2017).
- [Rea16b] Intel® RealSense™. *Intel® RealSense™ Cross Platform API*. 2016. URL: <https://github.com/IntelRealSense/librealsense/tree/legacy> (besucht am 28. 11. 2017).
- [Rea16c] Intel® RealSense™. *Projection and Deprojection in librealsense*. 2016. URL: <https://github.com/IntelRealSense/librealsense/blob/legacy/doc/projection.md> (besucht am 27. 11. 2017).
- [Ren98] Jochen Renz. „A Canonical Model of the Region Connection Calculus“. In: *Journal of Applied Non-Classical Logics* 12 (März 1998).
- [SF07] N. Suganuma und N. Fujiwara. „An Obstacle Extraction Method Using Virtual Disparity Image“. In: *IEEE Intelligent Vehicles Symposium, Istanbul, Turkey* (2007), S. 456–461.
- [Tea17] OpenCV Team. *Open Source Computer Vision Library*. 2017. URL: <https://opencv.org/> (besucht am 28. 11. 2017).
- [Wed+08] Andreas Wedel u. a. „B-spline modeling of road surfaces for freespace estimation“. In: *Conference: Intelligent Vehicles Symposium, IEEE* (Juli 2008), S. 828–833.

Literaturverzeichnis

- [ZT17] Yin Zhou und Oncel Tuzel. „VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection“. In: *Research Apple Inc* (Nov. 2017). URL: <https://arxiv.org/abs/1711.06396>.

Index

- AADC, 1, 4, 16
- Basislinie, 7, 27
- Disparität, 10, 25, 28
- Disparitätsprofil der Straße, 13
- Euklidische Distanz, 15, 37
- Extrinsische Parameter, 7, 27
- Freie-Fahrbahnebenen-Berechnung, 12
- Intel RealSense R200, 5
- Intrinsische Parameter, 6, 27
- Kalman-Filter, 5, 22
- Klassifikation, 2, 5, 18, 21, 48, 50
- Lidar, 5, 21
- Objekt-Detektion, 20, 23, 28, 41
- Objekt-Extraktion, 23, 33, 41
- Objekt-Verfolgung, 21, 24, 45
- Optischer Fluss, 5, 22
- Orientierungswinkel, 14, 37
- PCL, 40
- Positionierung, 14, 36, 41
- Punktwolke, 10, 40
- RANSAC, 41
- RCC-8, 46
- Rektifizierte Bilder, 9, 26
- ROI, 33, 50
- Rotation, 15, 37
- Semiglobal-Block-Matching, 10, 26, 29
- Stixel, 11, 21, 28
- Stixel-Kategorien, 12
- Tiefenbild, 9, 20
- Translation, 15, 37