

Universität der Bundeswehr München
Fakultät für Luft- und Raumfahrttechnik
Institut für Technik Autonomer Systeme

Objektbildung in dreidimensionalen Messdaten für automobiler Anwendungen

Philipp Steinemann, M.Sc.

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. habil. Alexander Lion
1. Berichterstatter:	Prof. Dr.-Ing. Hans-Joachim Wünsche
2. Berichterstatter:	Prof. Dr. rer. nat. Bernd Radig

Die Dissertation wurde am 12.08.2018 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 16.07.2019 angenommen. Die mündliche Prüfung fand am 26.08.2019 statt.

Danksagung

Die vorliegende Dissertation entstand während meiner Zeit als Doktorand bei der Daimler AG im Forschungszentrum Ulm. Im Rahmen einer Kooperation mit der Universität der Bundeswehr München wurde ich durch Herrn Prof. Dr.-Ing. Hans-Joachim Wünsche betreut. Ihm gilt mein besonderer Dank, da er mir richtungsweisend zur Seite stand und dennoch große Freiheiten gewährte.

Herrn Prof. Dr. rer. nat. Bernd Radig von der Technischen Universität München danke ich für die Übernahme des Koreferats und das Interesse an meiner Arbeit.

Meinem Teamleiter Dr.-Ing. Jürgen Dickmann möchte ich danken, da er es mir ermöglicht hat, die Arbeit bei der Daimler AG anzufertigen. Des Weiteren schaffte er mir die notwendigen Freiräume, um mich auf die wissenschaftliche Arbeit zu konzentrieren. Dem ganzen Team Aktive Sensoren danke ich für die offene und gute Zusammenarbeit. In ganz besonderem Maß gilt mein Dank Dr. Jens Klappstein, der mich mit zahlreichen Ideen und Diskussionen unterstützte.

Den Kollegen des Instituts Technik Autonomer Systeme danke ich für die Zusammenarbeit und den gemeinsamen Austausch. Ein spezieller Dank gilt Thorsten Lüttel.

An meine beiden damaligen Studenten Editha Kasisari (geb. Bednorz) und Witalij Wambold geht ein großer Dank für die wertvollen Beiträge zu meiner Arbeit.

Für das Korrekturlesen meiner Arbeit danke ich Nicolas Schächtele, Dr.-Ing. Antje Muntzinger (geb. Westenberger), Dr.-Ing. Michael Barjenbruch, Kathleen Krause und Melissa Schaich.

Ein großer Dank geht an meine Familie und meine Eltern, die mich jeder Zeit unterstützen haben.

Vaihingen an der Enz, 15.10.2019

Zusammenfassung

Lasersensoren bieten, im Vergleich zu aktuell im Automobil eingesetzten Sensoren, wie Radare, Kameras und Ultraschallsensoren, eine sehr genaue Entfernungsmessung bei gleichzeitig sehr guter Winkelauflösung. Sie eignen sich daher für den Einsatz in modernen Fahrerassistenzsystemen oder auch für die Anwendung als Referenz-Sensorik.

Die vorliegende Arbeit beschreibt alle notwendigen Schritte, ausgehend von den Rohdaten des Lasersensors bis zur Objektbildung und der sich daran anschließenden Anwendungen. Die Objektbildung stellt ein zentrales Thema der Arbeit dar. Es werden zwei im Rahmen dieser Arbeit neue entwickelte Ansätze zur Repräsentation von Objekten in dreidimensionalen Lasermessungen vorgestellt.

Zu Beginn wird auf die Segmentierung sowohl in zwei, als auch in drei Dimensionen eingegangen. Hierbei werden punkt-, bild- und gitterbasierte Ansätze behandelt.

Der erste Ansatz zur Repräsentation von Objekten in dreidimensionalen Lasermessungen modelliert die Außenkontur von Fahrzeugen als dreidimensionales parametrisches Flächenmodell. Dieses wird in die segmentierte dreidimensionale Punktwolke einer Lasermessung eingepasst. Dabei sorgen verschiedene Randbedingungen dafür, dass das Flächenmodell fahrzeugtypische Eigenschaften einhält.

Im zweiten Ansatz wird ein Partikel-Filter mit einem objekt-lokalen dreidimensionalen Gitter kombiniert. Jedes Partikel entspricht dabei einem Gitter. Ein Gitter stellt zusätzlich zur Hypothese des Zustandes auch eine Hypothese der Objektform dar. Neue Messungen werden über die Zeit in die einzelnen Gitter integriert, wobei die Integration durch diverse Kostenfunktionen bewertet wird. Dies ermöglicht die Auswahl der besten Zustands- und Objektformhypothesen. Über die Zeit akkumulieren sich in den Gittern die Messungen und ergeben eine dreidimensionale Beschreibung des Objekts.

Für beide Ansätze werden jeweils Anwendungen aufgezeigt, die sich aus der Objektrepräsentation selbst oder aus dem zugehörigen Verfahren ergeben.

Summary

In comparison to current automotive sensors laser sensors provide a precise distance measurement with an accurate angle resolution at the same time. Therefore laser sensors are a good choice for modern driver assistance or reference systems.

This thesis shows the necessary pre processing steps and introduces two new approaches for representing objects in three dimensional laser measurements.

At first the segmentation in two and also in three dimensions is presented. This includes point, image and grid based approaches.

The first approach for characterizing objects in three dimensional laser measurements represents the outline contour of vehicles by a parametric surface model. This surface model is fit to segmented three dimensional point clouds of vehicles. Different constraints are used for the surface model to fulfill vehicle specific properties.

The second approach combines a particle filter and an object centered three dimensional grid. Each grid represents one particle of the particle filter. Therefore a grid stands for one hypothesis of the state and additionally one hypothesis of an object shape. New measurements are added to the grids over time. The integration is scored by different cost functions. The scoring allows the selection of the best state and object shape hypothesis. As more and more measurements are added, a three dimensional object representation is evolving in each grid.

For each approach different example applications are presented.

Inhaltsverzeichnis

Symbolverzeichnis	XI
1 Einleitung	1
1.1 Hochgenaue Formrepräsentation von Objekten	4
1.2 Aufbau der Arbeit	5
2 Grundlagen	7
2.1 Koordinatensystem	7
2.2 Sensorik im automobilen Bereich für die Umfelderkennung	8
2.2.1 Lasersensor	9
2.2.1.1 Distanzmessung	10
2.2.1.2 Modellvergleich	11
2.2.1.3 Aufweitung des Laserstrahls	14
2.2.2 Radar	15
2.2.3 Kamera	18
2.2.3.1 Monokamera	18
2.2.3.2 Stereokamera	18
2.3 Bewegungskompensation bei scannenden Sensoren	19
2.4 Grundbegriffe der Wahrscheinlichkeitstheorie	21
2.5 Bayes-Filter	22
2.6 Erzeugung von Referenzdaten mit Inertialmessensorik und Globalem Positionssystem mit Differentialsignal (DGPS)	24
3 Segmentierung	27
3.1 Punktbasierte Ansätze	27
3.1.1 Bestimmung von Bodenpunkten	28
3.1.2 Punktbasierte Segmentierungsalgorithmen	29
3.1.2.1 K-Means	29
3.1.2.2 Mean-Shift	30
3.1.2.3 DBSCAN	32
3.1.3 Bewertung	33
3.2 Bildbasierter Ansatz	33
3.3 Gitterbasierte Ansätze	35

3.3.1	2D Ansatz	35
3.3.1.1	Bestimmung von Bodenpunkten	35
3.3.1.2	Erzeugung von Segmenten	37
3.3.2	3D Ansatz	39
3.3.2.1	Aufbereitung der Rohdaten	39
3.3.2.2	Erzeugung von zusammenhängenden Teilsegmenten und Bestimmung von Bodenpunkten	42
3.3.2.3	Erzeugung von Segmenten	48
3.4	Klassifikation	49
3.5	Zusammenfassung	49
4	Außenkonturbestimmung	51
4.1	Stand der Technik	52
4.2	Gitterfunktion	53
4.3	Vorverarbeitungsschritt	54
4.4	Flächenfunktion	60
4.5	Flächeneigenschaften	63
4.5.1	Glattheit	65
4.5.2	Konvexität	66
4.5.3	Negative Gradienten	67
4.5.4	Winkelbegrenzung	67
4.5.5	Orthogonalitätsbedingung	70
4.6	Das Optimierungsproblem	70
4.7	Lösung des Optimierungsproblems	72
4.7.1	Algorithmus von Goldfarb und Idnani	72
4.7.2	Effiziente Lösung des Optimierungsproblems	74
4.7.2.1	Allgemeine Möglichkeiten zur effizienten Lösung des Optimierungsproblems	74
4.7.2.2	Parallelisierte Implementierung auf einer Grafikkarte	76
4.8	Flächeninterpolation	78
4.9	Konfidenzbestimmung	78
4.9.1	Flächenkonfidenz	78
4.9.2	Nachbarschaft	80
4.9.3	Konfidenzmaß	81
4.9.4	Zuschneiden der Fläche	83
4.10	Nachverarbeitungsschritt	84
4.11	Robustheit der Flächenbestimmung	84
4.11.1	Fehlende Messungen	85
4.11.2	Rauschen	85
4.11.3	Ausreißer	86
4.11.4	Zusammenfassung	86
4.12	Bestimmung der zweidimensionalen Außenkontur	87
4.13	Übertragung des Verfahrens zur Außenkonturbestimmung auf andere Sensoren	92
4.13.1	Lasersensor mit vier Ebenen	92
4.13.2	Stereokamera	93

4.14	Vergleich der zweidimensionalen Außenkontur mit der realen Fahrzeugkontur	94
4.15	Anwendung der Außenkontur	97
4.15.1	Bestimmung des kürzesten Abstandes zum nächsten Fahrzeug . .	97
4.15.2	Vergleich zwischen der Lage der Außenkontur und einer Radarmes- sung	97
4.15.3	Geschwindigkeitsbestimmung durch Registrierung von Außenkon- turen	101
4.15.3.1	(Weighted) Iterative Closest Point Algorithmus ((W)ICP)	101
4.15.3.2	Linearer Kalman-Filter	102
4.15.3.3	Auswertung	104
4.16	Zusammenfassung	110
5	Objektverfolgung mit einem objekt-lokalen Gittermodell	113
5.1	Stand der Technik	113
5.2	Partikel-Filter	114
5.2.1	Der Partikel-Filter	114
5.3	Objekt-lokale Gittermodelle	117
5.4	Kostenfunktionen	118
5.4.1	Integrationsverhältnis	118
5.4.2	Besetzungsverhältnis	119
5.4.3	Besetzungsgradverhältnis	119
5.4.4	Besetzungsdistanz	120
5.4.5	Schwerpunktdistanz	121
5.4.6	Vergleich und Auswahl der Kostenfunktionen	121
5.5	Bewegungsmodell	123
5.6	Fahrzeugdynamikanalyse	124
5.7	Überblick über den Algorithmus	126
5.8	Akkumulation von Messdaten	128
5.9	Auswertung	129
5.10	Anwendung von objekt-lokalen Gittermodellen	135
5.11	Zusammenfassung	135
6	Zusammenfassung und Ausblick	137
	Literaturverzeichnis	140
	Literaturquellen	140
	Eigene Veröffentlichungen	140

Symbolverzeichnis

Das Symbolverzeichnis gliedert die in der vorliegenden Arbeit vorkommenden Symbole entsprechend den Kapiteln. Einfache Symbole, wie Zählvariablen, werden definiert, sobald sie auftreten und können in folgenden Kapiteln ebenfalls Verwendung finden.

'2 Grundlagen'

Symbol	Beschreibung
Ψ	Gieren
Θ	Nicken
Φ	Rollen
r_O	Abstand eines Objekts bezüglich einer Stereokamera
D	Disparität
r	Radius
φ	Azimutwinkel
θ	Elevationswinkel
x, y, z	Kartesische Koordinaten
d_o	Abstand zwischen Velodyne HDL-64E und Schachbrett
d_a	Durchmesser des Laserstrahls
α_l	Strahlaufweitung (Winkel)
t_{tof}	Lichtlaufzeit
d	Distanz
c_l	Lichtgeschwindigkeit im Medium Luft
c_r	Ausbreitungsgeschwindigkeit einer elektromagnetischen Welle in einem Medium
f_s	Sendesignal
f_e	Empfangssignal
v_{rel}	relative Geschwindigkeit
b	Basisbreite (Stereokamera)
f	Brennweite
v_x	Tangentialgeschwindigkeit
ω	Winkelgeschwindigkeit
α_{Offset}	Winkelversatz

t	Zeitwert
f_{Sensor}	Drehfrequenz eines scannenden Sensors
T	Zykluszeit
γ	Delta des Gierwinkels vom Sensor mit Offset

'3 Segmentierung'

Symbol	Beschreibung
k	Anzahl Punkte für den DBSCAN-Algorithmus
d_{mz}	Breite eines Bandes entlang der z-Achse
d_{z_i}	Distanz zwischen maximalem und minimalem z-Wert von Messungen in einer Zelle mit dem Index i
n_{Zellen}	Gesamtzahl der Zellen in einem zweidimensionalen Gitter
m_{Zelle_i}	Index auf eine Punktmenge von Messungen in einer Zelle i
ν	Öffnungswinkel eines dreidimensionalen polaren Gitters
ξ_B	Schwellwert für die Zuordnung von zwei benachbarten Ebenen zu einem Segment
d_s	Distanz bis zu der Werte in ein dreidimensionales polares Gitter eingetragen werden
i	Zählvariable
P_i	Punkt i aus einer Punktmenge
φ_i	Azimutwinkel eines Punktes aus einer Punktmenge
θ_i	Elevationswinkel eines Punktes aus einer Punktmenge
ρ_i	Distanzwert eines Punktes aus einer Punktmenge ausgehend vom Ursprung des Koordinatensystems
A, B, C	Messpunkte
j	Zählvariable
k	Zählvariable
E_i, E_j	Ebenen aus einer Ebenenmenge
S_k	Segment aus einer Segmentmenge
\vec{n}_i, \vec{n}_j	Normalenvektoren aus einer Menge
P_{ps}	Pseudopunkt

'4 Außenkonturbestimmung'

Symbol	Beschreibung
V	Gitterfunktion
h_x, h_z	Äquidistante Schrittweiten in einer Gitterfunktion entlang der x, z-Achse
n_x, n_z	Anzahl Knoten einer Gitterfunktion entlang der x, z-Achse
N_x, N_z	Menge an Knotenpunkte einer Gitterfunktion entlang der x, z-Achse

\vec{c}_p	Vektor der eine Punktmenge in den Koordinatenursprung verschiebt
$R_z(P, \alpha)$	Funktion die eine Punktmenge P mit dem Winkel α um die Z-Achse rotiert
$MIN_x(P)$	Funktion bestimmt den minimalen x-Wert in der Punktmenge P
$MIN_y(P)$	Funktion bestimmt den minimalen y-Wert in der Punktmenge P
$MIN_z(P)$	Funktion bestimmt den minimalen z-Wert in der Punktmenge P
$B(V, P)$	Funktion projiziert alle Punkte einer Punktmenge P entlang der y-Achse in das Gitter V und zählt dabei die Anzahl der Punkte, die in jede Zelle fallen
$T_1(P)$	Funktion verschiebt alle Punkte einer Punktmenge entlang der x- und z-Achse so, dass alle x- und z-Werte größer oder gleich Null sind
\vec{s}_o	Sensorursprung
\vec{s}'_o	Sensorursprung verschoben und um einen Winkel um die z-Achse gedreht
\vec{c}_E	Schnittpunkt zwischen Gerade und Ebene
d_1, d_2	Distanzwerte
α_R	Drehwinkel, um den eine Punktmenge im Vorverarbeitungsschritt zur Außenkonturbestimmung gedreht wird
P_R	Gedrehte und verschobene Punktmenge nach dem Vorverarbeitungsschritt zur Außenkonturbestimmung
P_{int}	Interpolierter Punkt
N_g	Anzahl der Knoten einer Gitterfunktion
N_p	Anzahl Punkte in Punktmenge P
N_s	Menge von Eins bis N_p . Wird genutzt um eine Zählvariable zu inkrementieren
\mathbf{A}_1	Matrix für die bilineare Interpolation. Enthält jeweils zeilenweise die Koeffizienten der Punkte aus der Punktmenge P_R
\vec{v}	Vektor für die gesuchten y-Werte der Knotenpunkte bei der bilinearen Interpolation und später Ergebnisvektor des gesamten Optimierungsproblems
\vec{y}_1	Vektor für die y-Werte der Punkte aus der Punktmenge P_R
\mathbf{G}	Matrix enthält die Glattheitsbedingungen
\mathbf{C}_1	Matrix enthält die Konvexitätsbedingungen
\vec{c}_1	Vektor enthält 0 für die Konvexitätsbedingung
\mathbf{C}_2	Matrix enthält die Bedingungen für die negativen Gradienten

\vec{c}_2	Vektor enthält 0 für die Bedingung der negativen Gradienten
\mathbf{C}_3	Matrix enthält die Bedingungen für die Winkelbegrenzung
β	Winkel für die Winkelbegrenzung
ζ	Parameter für die Winkelbegrenzung
\vec{c}_3	Vektor enthält β für die Winkelbegrenzung
\mathbf{B}	Matrix enthält die Orthogonalitätsbedingungen
\vec{b}	Vektor enthält 0 für die Orthogonalitätsbedingung
\mathbf{A}	Matrix fasst Matrix \mathbf{A}_1 und \mathbf{G} zusammen
\vec{y}	Vektor beinhaltet Vektor \vec{y}_1 . Der Rest ist aufgefüllt mit 0
\vec{c}	Vektor beinhaltet die Vektoren \vec{c}_1 , \vec{c}_2 und \vec{c}_3
N_t	Anzahl an Glattheitsbedigungen
N_c	Anzahl an Elemente in Vektor \vec{c}
N_m	Summe aus der Anzahl an Glattheitsbedigungen und der Anzahl an Punkten in der Punktmenge P
\mathbf{C}	Matrix enthält die Matrizen \mathbf{C}_1 , \mathbf{C}_2 und \mathbf{C}_3
F_k	Punktmenge enthält die Knotenpunkte als Lösung des Optimierungsproblems
\vec{e}_k	Vektor für die Flächenkonfidenz
\vec{e}_{kn}	Vektor für die Flächenkonfidenz normiert auf das Intervall $[0, 0.5]$
\vec{e}_{In}	Vektor für die Flächenkonfidenz mit interpolierten Daten und normiert auf das Intervall $[0, 0.5]$
d_n	Abstandswert für die Bestimmung der benachbarten Messungen zu einem Knotenpunkt
\vec{n}_m	Vektor enthält für jeden Knotenpunkt die Anzahl der benachbarten Messungen im Abstand d_n
N_N	Parameter über den sich definieren lässt, ab welcher Anzahl an Messungen in der Nachbarschaft der Knoten volle Vertrauenswürdigkeit erhalten soll.
\vec{s}_n	Vektor für die Vertrauenswürdigkeit der Knotenpunkte definiert über die Nachbarschaft
\vec{s}_{nI}	Vektor für die Vertrauenswürdigkeit der interpolierten Knotenpunkte definiert über die Nachbarschaft
\vec{k}	Konfidenzmaß für jeden Knotenpunkt
\vec{k}_I	Konfidenzmaß für jeden interpolierten Knotenpunkt
K_I	Interpolierte Knotenpunkte
K_{IK}	Interpolierte Knotenpunktmenge mit Konfidenzinformation
$P_{xz}(P_R)$	Funktion projiziert Punktmenge P_R in die xz -Ebene
$conv(\cdot)$	Funktion gibt die konvexe Hülle einer Punktmenge zurück

C_H	Punktmenge der Punkte, die die konvexe Hülle definieren
$inConv(\cdot, \cdot)$	Funktion bestimmt die Punkte, die innerhalb einer konvexen Hülle liegen
K_{IKC}	Interpolierte Knotenpunkte, die innerhalb einer definierten konvexen Hülle liegen
K_{IKCN}	Interpolierte Knotenpunkte, die innerhalb einer definierten konvexen Hülle liegen, und auf die der Nachverarbeitungsschritt angewandt wurde
$N_x(\cdot, \cdot)$	Normalverteilung entlang der x-Achse mit Mittelwert und Varianz
$N_y(\cdot, \cdot)$	Normalverteilung entlang der y-Achse mit Mittelwert und Varianz
$N_z(\cdot, \cdot)$	Normalverteilung entlang der z-Achse mit Mittelwert und Varianz
$pol(\cdot)$	Funktion berechnet aus kartesischen Koordinaten Polarkoordinaten
C_{HP}	Punktmenge der Punkte, die innerhalb einer definierten konvexen Hülle liegen und in Polarkoordinaten vorliegen
$MIN_\varphi(\cdot)$	Funktion berechnet den minimalen Azimuth-Winkel in einer Punktmenge
$MAX_\varphi(\cdot)$	Funktion berechnet den maximalen Azimuth-Winkel in einer Punktmenge
\vec{m}_{B_1}	Vektor auf den Schwerpunkt, berechnet über das arithmetische Mittel, einer Punktmenge B_1 in kartesischen Koordinaten
\vec{m}_{B_2}	Vektor auf den Schwerpunkt, berechnet über das arithmetische Mittel, einer Punktmenge B_2 in kartesischen Koordinaten
C_{2D}	Zweidimensionale Außenkontur
K_{IKCN_t}	Interpolierte Knotenpunkte, die innerhalb einer definierten konvexen Hülle liegen, und auf die der Nachverarbeitungsschritt angewandt wurde zum Zeitpunkt t
\vec{m}_{icp}	Vektor der die korrespondierenden Punkte in zwei Punktmengen enthält

'5 Objektverfolgung mit einem objekt-lokalen Gittermodell'

Symbol	Beschreibung
$\hat{\chi}_k$	Aktueller Schätzwert des Zustandes
χ_k^*	Prädizierter Schätzwert des Zustandes
κ	Kantenlänge
G	Gittermodell
n_x	Anzahl Zellen eines Gittermodells in x-Richtung

n_y	Anzahl Zellen eines Gittermodells in y-Richtung
n_z	Anzahl Zellen eines Gittermodells in z-Richtung
n_I	Anzahl der Messungen, die innerhalb des objekt-lokalen Gittermodells liegen
N	Anzahl Messungen
c_{IV}	Kostenfunktion: Integrationsverhältnis
λ	Maximale Anzahl an schon besetzten Zellen
nO	Summe über alle Messungen, die in der aktuellen Hypothese in schon besetzte Zellen fallen
c_{BV}	Kostenfunktion: Besetzungsverhältnis
sI	Summe über alle Messungen, die sich bereits in den Zellen befinden
NI	Maximale Anzahl der bereits in einer Zelle integrierten Messungen über alle Hypothesen hinweg
c_{BGV}	Kostenfunktion: Besetzungsgradverhältnis
$dMin$	Summe über die kürzesten Abstände zwischen Messungen und den nächsten besetzten Zellen
$dMax$	Maximale Distanz zwischen Messungen und zur nächsten besetzten Zelle über alle Hypothesen hinweg
c_{BD}	Kostenfunktion: Besetzungsdistanz
$inGrid(\cdot)$	Funktion zeigt an, ob sich ein Punkt innerhalb eines Gittermodells befindet
c_{SD}	Kostenfunktion: Schwerpunktdistanz
cog	Schwerpunkt (center of gravity)
c_G	Gesamtkostenfunktion
l	Länge des Einspurmodells
v_R	Tangentialgeschwindigkeit des Hinterrades
δ	Lenkwinkel
M	Momentanpol
$C(x_C, y_C)$	Schwerpunkt des Fahrzeugs in globalen Koordinaten
v_C	Fahrzeuggeschwindigkeit im Fahrzeugschwerpunkt
Ψ	Orientierungswinkel des objekt-lokalen Gittermodells
s_{ar}	Beschleunigung wird als Rauschen modelliert
s_δ	Lenkgeschwindigkeit wird als Rauschen modelliert
\dot{x}	Geschwindigkeit in x-Richtung
\dot{y}	Geschwindigkeit in y-Richtung
\dot{v}_R	Beschleunigung in tangentialer Richtung
$\dot{\delta}$	Lenkgeschwindigkeit
$\dot{\Psi}$	Winkelgeschwindigkeit um die z-Achse
a_L	Laterale Beschleunigung

Einleitung

Moderne Fahrzeuge unterstützen den Fahrer bereits in den unterschiedlichsten Situationen im Straßenverkehr. Sei es beim Wechseln des Radiosenders durch Spracheingabe, oder der intelligenten Routenplanung für die Fahrzeugnavigation. Ein Bereich, der sich in den letzten Jahren außergewöhnlich stark entwickelt hat, ist der der Fahrerassistenzsysteme. Von der parallelen und schnellen Entwicklung der Rechnertechnik konnten die Fahrerassistenzsysteme sehr profitieren, da sie eine Echtzeitverarbeitung von Sensordaten zur Entscheidungsfindung von einzuleitenden Warn-, Brems- oder Ausweichmanövern erfordern. Mit dem Blick in die Zukunft gelten die heute verfügbaren Systeme letztlich nur als kleiner Baustein auf dem Weg zum unfallfreien und autonomen Fahren.

Im Rahmen einer Studie [37], die von der Gesellschaft für Konsumforschung (GfK) im Jahr 2012 im Auftrag der Firma AutoScout24 durchgeführt wurde, wurden 8800 Teilnehmer aus sieben europäischen Ländern zum Fahrzeug und der Mobilität in 25 Jahren befragt. Gegenüber dem autonomen Fahren waren ca. zwei Drittel der Befragten bereits heute aufgeschlossen. Der größte Teil unter den Befürwortern möchte jedoch weiterhin die Wahl haben, ob sie den Autopiloten einschalten oder selbst fahren.

Dass autonomes Fahren schon möglich ist, hat Ernst Dickmanns [20] bereits in den 1980er Jahren mit unterschiedlichen Versuchsfahrten gezeigt. Wettbewerbe, wie die DARPA Grand Challenge im Jahre 2005 [6], haben dazu beigetragen die Entwicklung des autonomen Fahrens voranzutreiben und spiegelten den aktuellen Stand der Forschung wider.

Weltweit beschäftigen sich zahlreiche Universitäten mit dem Aufbau von Fahrzeugen zur Erforschung des autonomen Fahrens und der damit entstehenden Herausforderungen. In Deutschland sind dies unter anderem die Universität der Bundeswehr München mit ihrem Fahrzeug MuCAR-3 [45], welches auch erfolgreich an dem Roboterwettbewerbs Elrob [32] teilgenommen hat, und die Technische Universität Braunschweig mit ihrem Projekt Stadtpilot [86] und dem Fahrzeug Leonie. Auch die Freie Universität Berlin forscht auf dem Gebiet des autonomen Fahrens mit ihrem Fahrzeug MadeInGermany im Projekt AutoNOMOS [33].

Dass es sich beim autonomen Fahren nicht um reine Forschung für ein zeitlich weit in der Zukunft liegendes Vorhaben handelt, zeigt das Engagement diverser Automobilhersteller. Im Jahre 2012 wurde für die folgende S-Klasse Generation von Mercedes-Benz die Möglichkeit zu einer teilautonomen Fahrt angekündigt [14]. Forbes.com berichtete im Jahr 2012 auch von Arbeiten im Bereich des autonomen Fahrens bei BMW, Audi und Volvo [30]. Bereits im Jahr 2013 präsentierte Mercedes-Benz ein Fahrzeug mit seriennahe Technologie, das sich autonom im Stadt- und Überlandverkehr bewegen kann [16]. Die Fahrt zwischen Mannheim und Pforzheim ist auch bekannt als „Bertha Benz Fahrt“.

Firmen wie Google Inc. die zunächst keine direkte Verbindung zur Fahrzeugtechnik vermuten lassen, beginnen sich ebenfalls für das autonome Fahren zu interessieren. Sebastian Thrun berichtete bereits im Jahr 2010 [87] von fortgeschrittenen Entwicklungen auf diesem Gebiet.

Zum aktuellen Zeitpunkt hindern rechtliche Schranken und die ungeklärten Haftungsregelungen die Zulassung von autonomen Fahrzeugen im realen Straßenverkehr. In diesem Zusammenhang wird meist auf die Vorschriften des Wiener Übereinkommens über den Straßenverkehr aus dem Jahr 1968 verwiesen [7]. Speziell geht es um den Artikel 8 Absatz 1 der besagt, dass ein sich bewegendes Fahrzeug einen Führer haben muss. Dies jedoch widerspricht der Idee des autonomen Fahrens.

Im Jahr 2011 meldete The New York Times [63], dass sich Google Inc. für einen entsprechenden Gesetzentwurf für autonomes Fahren im US Bundesstaat Nevada bemühe, was auch kurze Zeit später zu einem Gesetz führte. Die erste Lizenz für ein autonomes Fahrzeug wurde dementsprechend auch an Google Inc. ausgegeben. Dies erfolgte einem Bericht der Nachrichtenagentur Reuters zufolge im Jahr 2012 [52]. Auch die derzeit noch sehr hohen Kosten für die notwendigen Sensoren und Systeme verlangsamten die Einführung in den Markt. Eine dritte Hürde ergibt sich durch die hohe Komplexität, die bei gemischtem Verkehr aus autonomen Fahrzeugen, Fahrzeugen, die von Menschen gefahren werden und allen weiteren Verkehrsteilnehmern wie Fußgänger, Radfahrer etc. entsteht. Christof Vieweg [90] beschreibt im Jahre 2011, dass Automobilhersteller die Bevormundung des Fahrers versuchen zu vermeiden, was einen vierten Grund für die zögerliche Einführung von autonomen Fahrzeugen darstellt.

Für zukünftige Fahrzeuge wird sich der Sensorbedarf nach einer Studie von Kevin Mak, Strategy Analytics [61], gegenüber dem Jahr 2012 bis zum Jahre 2019 mehr als verdoppeln. Dabei sieht Mak für alle dargestellten Sensoren ein stetiges Wachstum in den verbauten Stückzahlen. Im Jahr 2009 verzeichnete der Laser bereits einen sprunghaften Anstieg in seinen Stückzahlen, was sich auch auf die kostengünstigere Herstellung zurückführen lässt. Die von Mak ermittelten Zahlen für die Sensoren sind in Abbildung 1.1 dargestellt. Die Angaben bis 2012 entsprechen den wirklichen Stückzahlen.

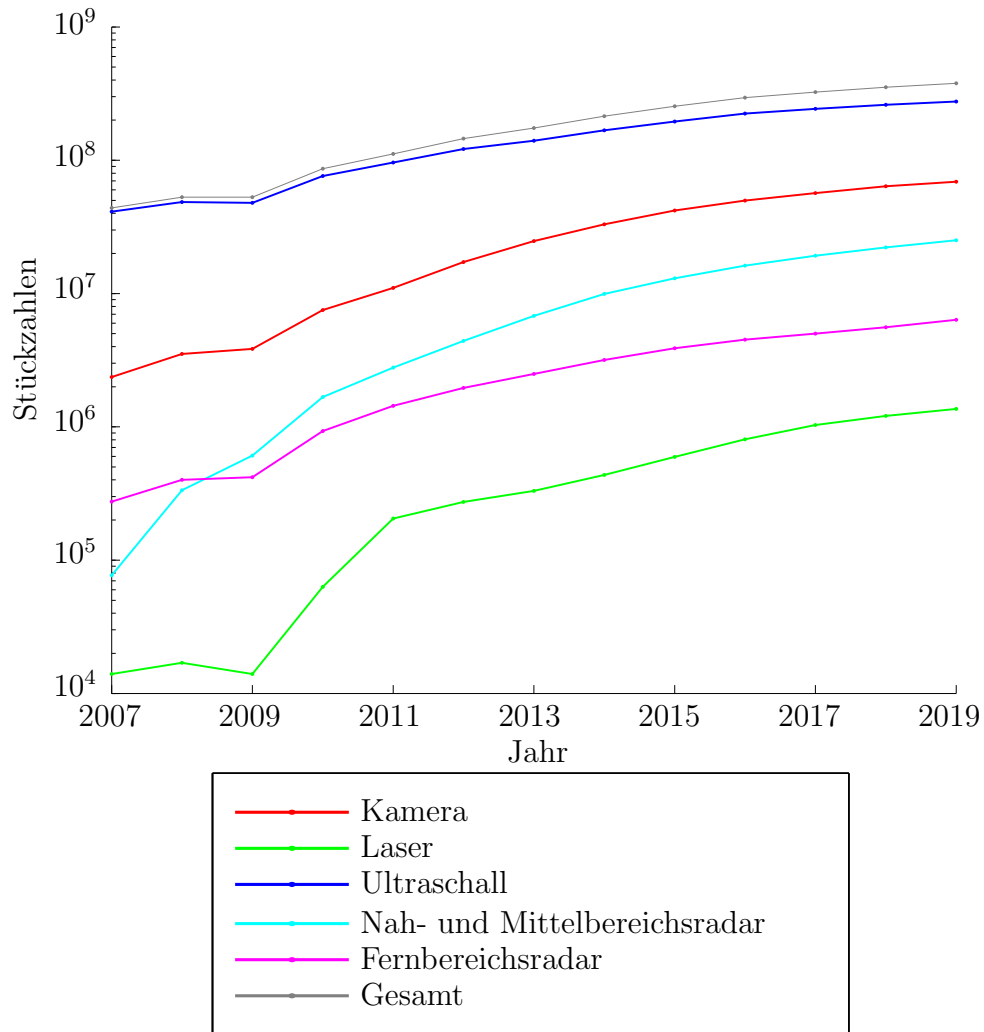


Abbildung 1.1: Bedarf an Sensoren in den kommenden Jahren (Strategy Analytics [61])

Höhere Auflösungen der Sensordaten, mehr Messungen in einem Zeitschritt pro Objekt und die Kombination von verschiedenen Sensoren ermöglichen die Entwicklung immer komplexerer Systeme. Ein Punkt, der dabei vermehrt an Bedeutung gewinnt, ist die Erzeugung von Referenzdaten, oder vereinfacht ausgedrückt die Bestimmung des wahren Werts. Speziell in der Absicherung der Systeme, aber auch während der Entwicklung, haben Referenzsysteme immer größere Bedeutung.

Schwerpunkt der vorliegenden Arbeit ist die Entwicklung von Möglichkeiten ein Objekt, in diesem Fall ein Fahrzeug, derart zu repräsentieren, dass die Objektrepräsentation als Basis für ein laserbasiertes Referenzsystem eingesetzt werden kann. Ferner lässt sich jeder Ansatz, der als Grundlage für ein Referenzsystem dient, auch bei entsprechend zur Verfügung stehender Rechenleistung, in die Funktionsentwicklung übernehmen. In Anbetracht des Mooreschen Gesetzes, nach dem sich die Komplexität von integrierten Schaltkreisen regelmäßig verdoppelt, stellt es lediglich eine Frage der Zeit dar, bis sich die Algorithmen,

die heute als zu rechenintensiv gelten, in Steuergeräte integrieren lassen.

1.1 Hochgenaue Formrepräsentation von Objekten

Durch die bereits angesprochene stetige Entwicklung neuer und die Weiterentwicklung bestehender Sensoren für den Einsatz im automobilen Umfeld erhöht sich die Genauigkeit und die Dichte der Messungen allgemein. Dies erfordert eine Anpassung der bisher verwendeten Objektbeschreibungsformen, die in der Sensorsignalverarbeitung Anwendung finden.

Eine weit verbreitete und durchaus praktikable Objektbeschreibungsform stellt die Approximation des Objekts als Box dar. Beispiele hierfür finden sich bei Michael Himmelsbach et al. [44] oder bei Sylvia Pietzsch et al. [71]. Wenn allerdings, für die anschließende Funktion, präzise Aussagen über das Objekt benötigt werden, ist die Approximation als Box nicht länger ausreichend. Hierbei ist es unerheblich, ob es sich um die wirkliche Fahrerassistenzfunktion oder das Referenzsystem handelt.

Die im Folgenden beschriebene Simulation, welche in Abbildung 1.2 grafisch dargestellt ist, führt einen Vergleich zwischen der Approximation eines Fahrzeugs durch eine Box und der wirklichen Form des Fahrzeugs durch. Die Simulation zeigt zwei Fahrzeuge. Das rechte Fahrzeug ist stationär, wohingegen das linke Fahrzeug mit maximalem Lenkwinkel nach rechts ausschert. Das Bewegungsmodell des Fahrzeugs folgt dem Einspurmodell ('5.5 Bewegungsmodell'). Die Geschwindigkeit des Fahrzeugs beträgt $v_x = 5 \frac{m}{s}$. Der Ausweichvorgang wird mehrfach wiederholt, wobei das stationäre Fahrzeugs jeweils um $0.02m$ in negative x-Richtung verschoben wird. Bei jedem Ausweichvorgang wird auf eine Kollision zwischen beiden Objektbeschreibungen geprüft. Die Simulation wird so lange wiederholt, bis eine Kollision zwischen beiden Objektbeschreibungsformen aufgetreten ist.

Das Ergebnis zeigt, dass es mit der Objektbeschreibung als Box $0.14s$ früher zu einem Zusammenstoß kommt, als mit der wirklichen Kontur. Je stärker die Rundungen an den Fahrzeugen ausgeprägt sind, desto größer wird der Effekt. Das Beispiel zeigt, dass die Objektform nicht vernachlässigt werden darf, da sonst bei engen Vorbeifahrten unter Umständen Fehlfunktionen zu erwarten sind. Eine Fehlfunktion kann schwerwiegende Auswirkungen haben, wenn beispielsweise die Information über die Kollision mit der Aktivierung von irreversiblen Sicherheitsfunktionen gekoppelt ist.

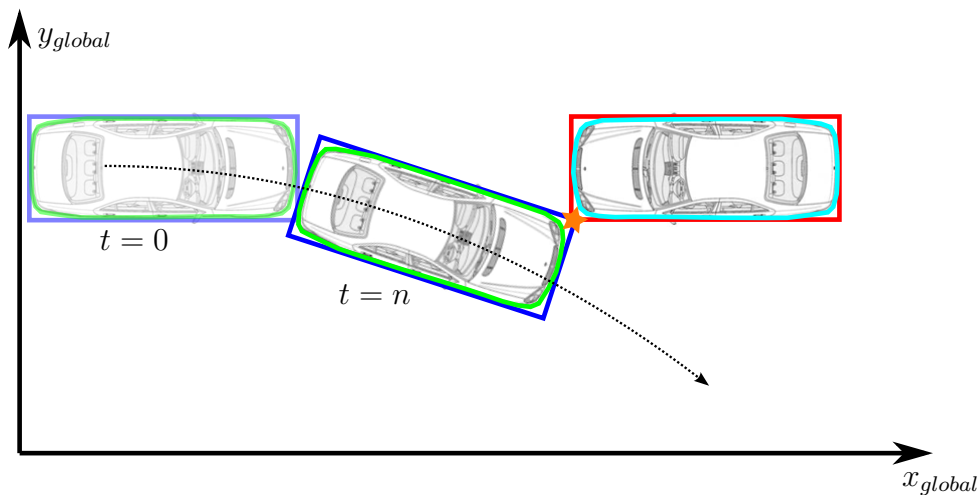


Abbildung 1.2: Vergleich der Kollisionen zwischen der realen Außenkontur und der Annäherung der Außenkontur durch ein Boxmodell.

1.2 Aufbau der Arbeit

Die Arbeit gliedert sich in sechs einzelne Kapitel. Zu Beginn werden in Kapitel '2 Grundlagen' die Grundlagen erläutert. Dies schließt die Erläuterung des Koordinatensystems und die Beschreibung der im Laufe der Arbeit vorkommenden Sensoren ein. Einen wesentlichen Schritt bei der Verarbeitung von Daten eines Lasersensors stellt die Segmentierung der Daten dar. Hierzu erfolgt in Kapitel '3 Segmentierung' ein Vergleich von verschiedenen aus der Literatur bekannten Algorithmen mit im Rahmen der Arbeit entstandenen Ansätzen. Den Kern der Arbeit stellen die beiden Kapitel '4 Außenkonturbestimmung' und '5 Objektverfolgung mit einem objekt-lokalen Gittermodell' dar. Sie beschreiben jeweils einen neuen Ansatz zur Objektrepräsentation von Fahrzeugen in Daten eines Lasersensors. Ferner wird dabei auf Erweiterungen der Ansätze eingegangen und es werden Anwendungen beschrieben. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick in Kapitel '6 Zusammenfassung und Ausblick'. Abbildung 1.3 zeigt den Aufbau und die Reihenfolge der Kapitel in einer grafischen Darstellung.

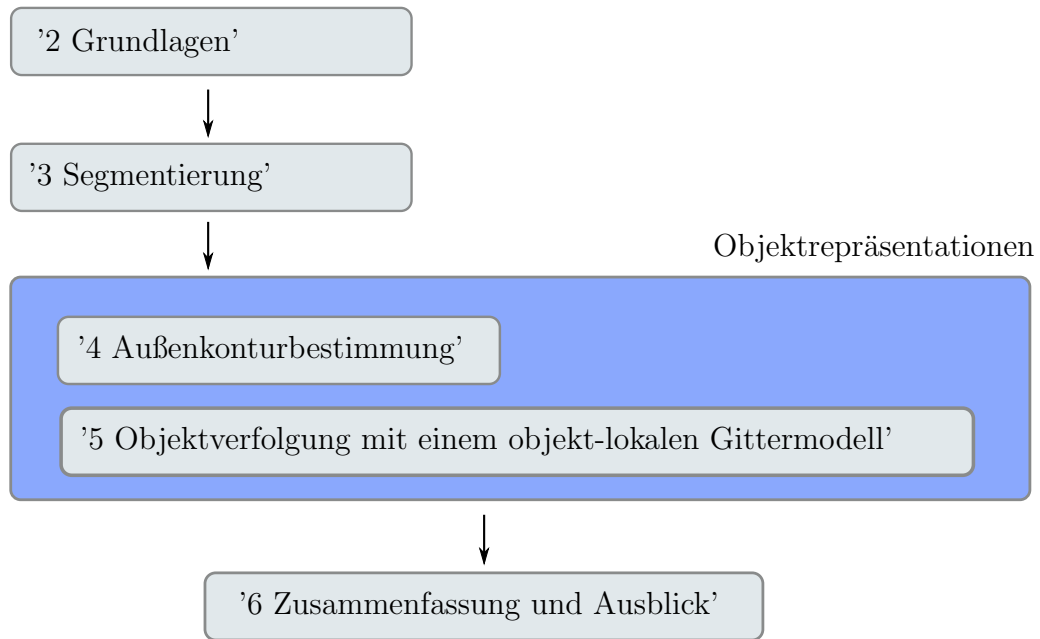


Abbildung 1.3: Kapitelübersicht

Die im folgenden vorgestellten Grundlagen dienen zum allgemeinen Verständnis der sich anschließenden Kapitel. Zunächst wird auf das verwendete Koordinatensystem eingegangen. Als nächstes folgt eine Beschreibung der im automobilen Bereich eingesetzten Sensorik. Der darauf folgende Abschnitt zeigt die Notwendigkeit für eine Bewegungskompensation bei scannenden Sensoren und beschreibt die Durchführung. In den beiden anschließenden Abschnitten werden die Grundbegriffe der Wahrscheinlichkeitstheorie und der Bayes-Filter eingeführt. Der letzte Abschnitt zeigt ein Verfahren zur Erzeugung von Referenzdaten, welches in späteren Kapiteln Anwendung finden wird.

2.1 Koordinatensystem

Bei Bernd Heißing et al. [42] wird das in DIN 70000 definierte Koordinatensystem allgemein beschrieben. Dabei liegt der Ursprung im Schwerpunkt des Fahrzeugs, die x-Achse zeigt in Fahrtrichtung, die y-Achse verläuft nach links und die z-Achse nach oben. Die Definition der Reihenfolge bei Drehungen ist: Drehung um z (Gieren), Drehung um y (Nicken), Drehung um x (Rollen). Die verwendeten Formelzeichen sind: Ψ , Θ , Φ .

Da in den folgenden Abschnitten überwiegend nur ein Sensor verwendet wird, wird der Ursprung des Koordinatensystems in den Nullpunkt des Sensors gelegt. Hierfür wird der Begriff des Sensorkoordinatensystems verwendet. Abbildung 2.1 zeigt beispielhaft ein Sensorkoordinatensystem eines auf dem Fahrzeugdach angebrachten Sensors. Werden mehrere Sensoren genutzt, werden die Sensorkoordinatensysteme in ein gemeinsames Koordinatensystem, das zuvor definiert werden muss, umgerechnet.

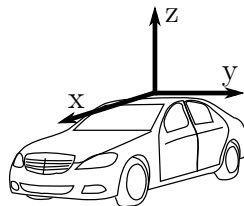


Abbildung 2.1: Sensorkoordinatensystem für einen Sensor, der auf dem Fahrzeugdach angebracht wurde

Bei dem definierten Sensorkoordinatensystem handelt es sich um ein kartesisches Koordinatensystem. Einige Sensoren stellen allerdings die Messungen in einem Kugelkoordinatensystem zur Verfügung. Das bedeutet, dass eine Messung über den Abstand r und den zwei Raumwinkeln in Azimut φ und in Elevation θ beschrieben wird (Abbildung 2.2). In Kugelkoordinaten liegen beispielsweise auch die Messungen der im Folgenden vorgestellten Lasersensoren vor. Aus diesem Grund muss eine Umrechnung zwischen den Koordinatensystemen durchgeführt werden. Dies erfolgt über den Zusammenhang

$$\begin{aligned} x &= r \cos(\theta) \cos(\varphi) \\ y &= r \cos(\theta) \sin(\varphi) \\ z &= r \sin(\theta) \end{aligned} \tag{2.1}$$

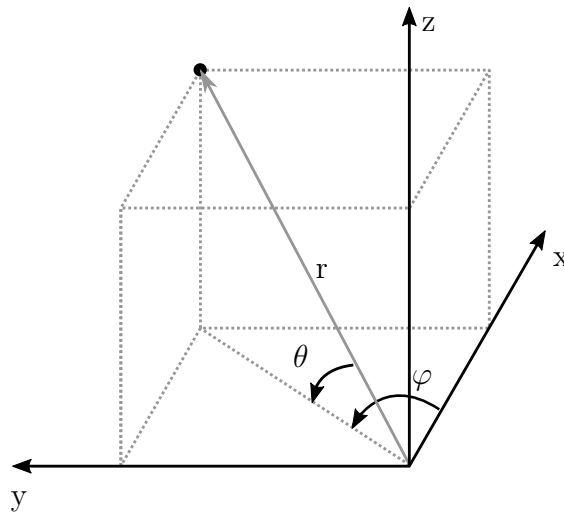


Abbildung 2.2: Darstellung von Kugelkoordinaten und eines kartesischen Koordinatensystems

Wenn nicht anders angegeben, sind die Einheiten im kartesischen Koordinatensystem entlang der Achsen in der SI-Einheit Meter.

2.2 Sensorik im automobilen Bereich für die Umfeldfassung

Die folgenden Abschnitte beschreiben Sensoren für moderne Fahrerassistenzsysteme. Bei den vorgestellten Kameras und Radaren handelt es sich bereits um serienmäßig in Fahrzeugen verbaute Sensoren. Im Serieneinsatz reichen die Funktionen von der Verkehrszeichenerkennung über die Totwinkelerkennung bis hin zur nahezu autonomen Fahrgeländeerkennung.

Lasersensoren finden sich zurzeit einzig in einer sehr einfachen Ausführung in Fahrzeugen. Ein Beispiel hierfür ist ein von der Continental AG entwickelter Sensor, der dem Erkennen von Hindernissen bei niedrigen Geschwindigkeiten im Stadtverkehr dient [11].

Dabei gilt die Vermeidung von Auffahrunfällen als einer der Anwendungsfälle. Obwohl Mitsubishi bereits im Jahre 1995 ein Abstandsregelsystem auf Basis eines Lasersensors auf den Markt brachte (Bogdan Fijalkowski [28]), ist die Verbaurrate von Lasersensoren in Fahrzeugen aktuell gering.

Im Folgenden werden zunächst verschiedene Lasersensoren beschrieben. Sie werden im Fokus der weiteren Kapitel liegen. Darauf folgt die Beschreibung von Radarsensoren woran sich im letzten Abschnitt eine Beschreibung von Kameras anschließt. Sowohl die Radarsensoren, als auch die Kameras werden im späteren Verlauf für Anwendungen herangezogen werden.

2.2.1 Lasersensor

Der Begriff Laser ist ein Akronym für Light Amplification by Stimulated Emission of Radiation. Übersetzt bedeutet dies: Verstärkung des Lichts durch stimulierte Emission. Bei Lasern handelt es sich um eine noch relativ junge Technologie. Im Jahre 1960 gelang es Theodore Maiman den ersten Laserblitz mit Hilfe eines Rubins, der als aktives Medium diente, zu zünden [60]. Das dabei entstehende Licht hat bei einem Laser stets die gleiche Wellenlänge. Dies unterscheidet beispielsweise den Laser von einer Glühlampe, die Licht mit unterschiedlichen Wellenlängen aussendet. Jürgen Eichler et al. [23] beschreiben, dass bis heute viele verschiedene Laserquellen in Wellenlängenbereichen von unter $0.01\mu m$ bis über $1000\mu m$ bekannt sind. Damit wird der Spektralbereich von weicher Röntgenstrahlung bis hin zu Millimeterwellen abgedeckt.

Für die Anwendung von Lasersensoren im automobilen Bereich beschreibt Jörg Heerlein [41] den Einsatz von Pulslaserdioden. Ein Beispiel ist die von der OSRAM Opto Semiconductors GmbH hergestellte Pulslaserdiode SPL PL 90 [69]. Die Dioden emittieren Licht im Wellenlängenbereich von $905nm$, welches für das menschliche Auge nicht mehr sichtbar ist. Diese Tatsache, dass das Licht für das menschliche Auge nicht sichtbar ist, erfordert erhöhte Vorsicht, denn hierdurch werden die natürlichen Schutzmechanismen des Auges ausgeschaltet. Das Auge kann nicht mit dem sonst üblichen Lidschlussreflex reagieren, was daraufhin zu einer Schädigung des Auges führen kann. Speziell im Bereich des sichtbaren und angrenzenden nicht sichtbaren Bereichs erhöht sich die Gefahr für das Auge durch eine Fokussierung des Lichtes durch die Augenlinse. Die Durchlässigkeit (Transmission) des menschlichen Auges auf die Netzhaut in Abhängigkeit der Wellenlänge ist in Abbildung 2.3 dargestellt.

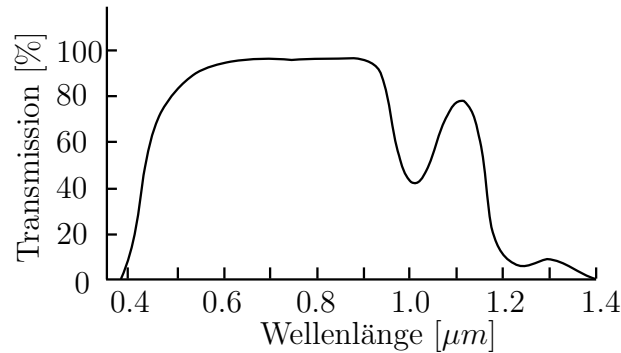


Abbildung 2.3: Transmission des menschlichen Auges auf die Netzhaut in Abhängigkeit der Wellenlänge (entsprechend den Daten aus Jürgen Eichler et al. [23])

Um die Augensicherheit zu gewährleisten und die Laserklasse 1 oder 1M nach EN 60825-1 einzuhalten, wird im Allgemeinen mit sehr kurzen Impulsen (ca. $10ns$) von geringer Energie (ca. $1mW$) gearbeitet. Bei Lasersensoren mit horizontaler Ablenkung des Lichtimpulses ('2.2.1.2 Modellvergleich'), welche Lichtebenen zur Vermessung nutzen, wird die Verweildauer des Lasers zusätzlich durch die Messfrequenz von ca. $5-50Hz$ begrenzt.

Nach dem Austritt des Laserstrahls aus dem Sensor ist das Licht den Einflüssen aus der Luft ausgesetzt, die eine Dämpfung hervorrufen. Kleine Partikel in der Luft, wie z.B. Staub oder Wasser, tragen zu dieser Dämpfung bei. An ihnen wird Licht einerseits diffus reflektiert oder andererseits absorbiert. Diese Effekte begrenzen die Reichweite eines Laserstrahls. Bei der Distanzmessung, die im folgenden Abschnitt vorgestellt wird, trägt die Reflektivität des Objekts, das das vom Sensor ausgesandte Licht reflektiert, zusätzlich zur maximalen Reichweite der Distanzmessung bei.

2.2.1.1 Distanzmessung

Lasersensoren für die Umfelderkennung werden häufig auch mit dem Begriff LIDAR zusammengefasst. LIDAR ist dabei ein Akronym für Light Detection And Ranging.

Die Distanzmessung erfolgt üblicherweise über die Laufzeitmethode. Dabei wird ein Lichtimpuls ausgesendet, der sich in der Luft bis zu einem Objekt ausbreitet und zum Sensor zurück reflektiert wird. Man spricht dabei auch von einem Echo. Am Sensor befindet sich eine Photodiode, die das reflektierte Licht wieder empfängt. Wird die Laufzeit t_{tof} des Lichts vom Aussenden bis zum Empfang gemessen, lässt sich die Distanz d mit Hilfe der Lichtgeschwindigkeit im Medium Luft c_l wie folgt berechnen:

$$d = \frac{c_l t_{tof}}{2} \quad (2.2)$$

Mit der Laufzeitmethode ist die Erweiterung des Sensors auf eine Mehrzielfähigkeit möglich. Dabei wird nicht direkt nach dem ersten Eintreffen eines Echos die Messung abgebrochen, sondern es wird auf weitere Echos gewartet. Möglich ist dies, da sich das Licht nicht zu einem unendlich dünnen Strahl bündeln lässt. Der Strahl weitet sich mit zunehmender Entfernung vom Sensor. Abbildung 2.4 veranschaulicht das Verfahren der

Laufzeitmethode mit Mehrzielfähigkeit. Es sind drei Hindernisse mit unterschiedlichen Höhen farbig gekennzeichnet. Der ausgesandte, aufgeweitete Laserstrahl wird an den Hindernissen zurück zum Sensor reflektiert. Dies ergibt verschiedene Laufzeiten des Lichts, die vom Sensor ausgewertet werden müssen.

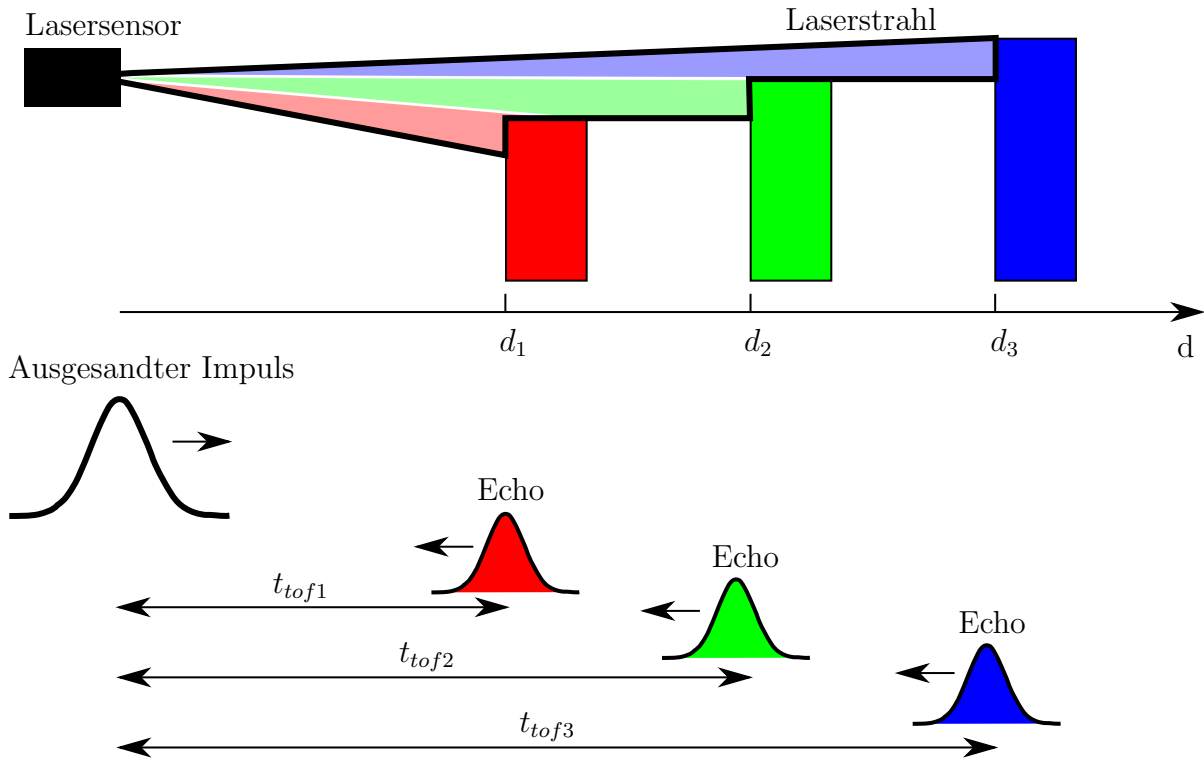


Abbildung 2.4: Laufzeitmethode mit Mehrzielfähigkeit zur Messung der Distanz von Objekten. Die Farben zeigen die Reflexionen an den verschiedenen Hindernissen an.

2.2.1.2 Modellvergleich

In der Automatisierungstechnik, der Architektur und der Vermessung werden Lasersensoren bereits in vielen Bereichen eingesetzt. Langsam folgt der Einsatz von Lasersensoren im Automobilbereich. Nachfolgend sollen zwei für diese Arbeit relevante Lasersensoren vorgestellt werden. Zunächst folgt eine Vorstellung des SICK LD-MRS [84]. Daran schließt sich die Beschreibung des Velodyne HDL-64E [89] an.

SICK LD-MRS Die Sensoren der SICK LD-MRS Serie [84] sind baugleich mit denen, die die Firma Ibeo Automotive Systems GmbH unter dem Produktnamen IBEO LUX [48] verkauft. Die SICK AG konzentriert sich dabei beispielsweise auf die Automatisierungstechnik in der Landwirtschaft und dem Bergbau. Die IBEO LUX Sensoren werden im Zusammenhang mit Anwendungen im automobilen Bereich angeboten. Aufgrund der Baugröße des Sensors lässt er sich auch nahezu verdeckt im unteren Teil des Stoßfängers eines Fahrzeugs verbauen. Da im Kapitel '4 Außenkonturbestimmung' ein SICK LD-MRS verwendet wird, wird im Folgenden dieser Sensor vorgestellt und auch die Bezeichnung

beibehalten. Abbildung 2.5 zeigt ein Bild des Sensors.



Abbildung 2.5: SICK LD-MRS Lasersensor (Quelle: SICK AG)

Die prinzipielle Funktionsweise des Lasersensors ist in Abbildung 2.6 dargestellt. Die Abstandsmessung erfolgt nach der Laufzeitmethode, wie sie im Abschnitt zuvor vorgestellt wurde. Zunächst erzeugt die Sendeeinheit einen Lichtimpuls, der entlang der rot markierten Strecke über den Spiegel abgelenkt wird und über das Fenster den Sensor verlässt. Der Lichtimpuls erreicht das Objekt, an dem er reflektiert wird. Der reflektierte Lichtimpuls gelangt wieder über das Fenster in den Sensor, wird am Spiegel umgelenkt und wird von der Empfangseinheit detektiert. Der über einen Motor drehbar gelagerte Spiegel ermöglicht eine horizontale Ablenkung des Lichtimpulses, was dazu führt, dass nicht nur ein einzelner Punkt, sondern eine Ebene vermessen werden kann.

Sensoren, die mehrere Messebenen aufweisen, besitzen demnach entsprechend viele Sendeeinheiten und Empfangseinheiten an verschiedenen Stellen, oder eine entsprechend angepasste Optik zur Bündelung des Lichts.

Eine Zusammenfassung der technischen Daten des SICK LD-MRS findet sich in Tabelle 2.1.

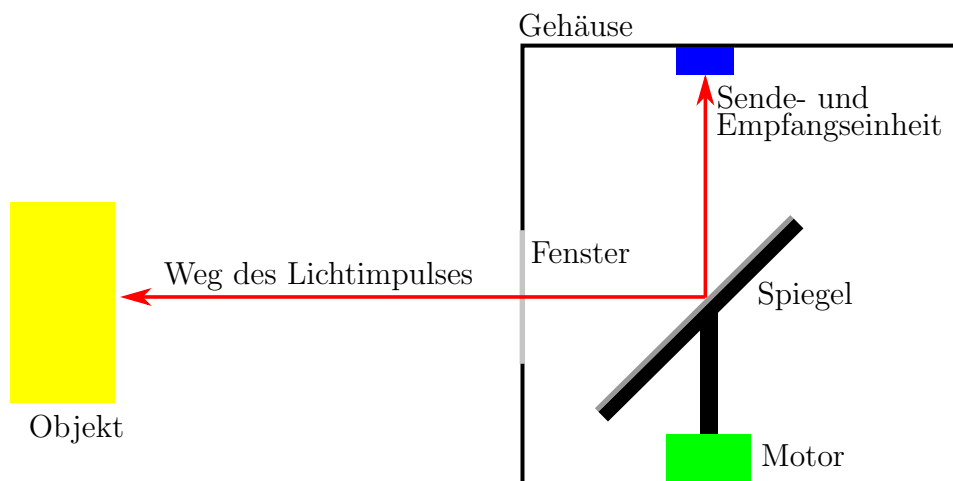
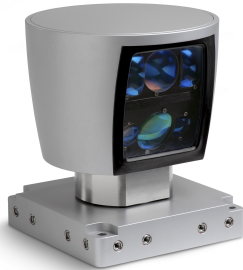


Abbildung 2.6: Prinzip der Funktionsweise des SICK LD-MRS

Velodyne HDL-64E Der Velodyne HDL-64E [89] ist ein Lasersensor, der einen horizontalen Sichtbereich von 360° aufweist. Um den Sichtbereich nutzen zu können, ist

es erforderlich den Sensor an einer exponierten Lage anzubringen. Bei der Verwendung in Kombination mit Fahrzeugen wird der Sensor meist auf dem Dach mit Hilfe eines Dachträgers montiert. Abbildung 2.7 (a) zeigt ein vergrößertes Bild des Lasersensors. Abbildung 2.7 (b) zeigt den Lasersensor auf einem Versuchsträger.



(a) Bild des Velodyne HDL-64E
(Quelle: Velodyne Lidar, Inc.)



(b) Velodyne HDL-64E auf einem Versuchsträger

Abbildung 2.7: Velodyne HDL-64E Lasersensor

Der Lasersensor hat 64 Sende- und Empfangseinheiten, die in zwei Blöcke mit je 32 Einheiten aufgeteilt sind. Beide Blöcke sind auf einem rotierenden Sensorkopf angebracht, was den horizontalen Sichtbereich von 360° ermöglicht. Abbildung 2.8 zeigt den internen Aufbau des Lasersensors. Die Abstandsmessung erfolgt, wie im zuvor beschriebenen Abschnitt, über die Laufzeitmethode.

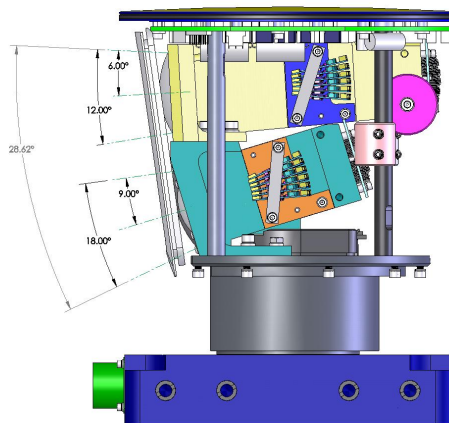


Abbildung 2.8: Interner Aufbau des Velodyne HDL-64E (Quelle: Velodyne Lidar, Inc.)

Weitere Eigenschaften des Velodyne HDL-64E finden sich in Tabelle 2.1.

Lasersensor	SICK LD-MRS	Velodyne HDL-64E
Ebenen	4	64
Reichweite	250 <i>m</i>	120 <i>m</i>
Distanzgenauigkeit	0.1 <i>m</i>	0.02 <i>m</i>
Sichtbereich horizontal	85° – 110°	360°
Winkelauflösung horizontal	0.125°	0.09° - 0.27°
Sichtbereich vertikal	3.2°	26.8°
Winkelauflösung vertikal	0.8°	0.4° (0.3° für die oberen 32 Dioden, 0.5° für die unteren 32 Dioden)
Mehrzielfähigkeit	Ja (3)	Nein (ab dem Modell HDL-64E S3 sind zwei Ziele möglich)
Messfrequenz	12.5 <i>Hz</i> – 50 <i>Hz</i>	5 <i>Hz</i> – 15 <i>Hz</i>

Tabelle 2.1: Vergleich der beiden verwendeten Lasersensoren

2.2.1.3 Aufweitung des Laserstrahls

Obwohl das Licht in einem Laser gebündelt ist, weitet es sich mit einem gewissen Öffnungswinkel auf. Diese Eigenschaft wird auch mit dem Begriff der Strahldivergenz beschrieben. Der Effekt kann genutzt werden, um einen größeren Bereich über eine einzige Sende- und Empfangseinheit abzudecken, oder um die Mehrzielfähigkeit zu ermöglichen. Die Ausnutzung der Aufweitung des Laserstrahls findet beispielsweise zur Vergrößerung des Abdeckungsbereichs bei dem von der Continental AG entwickelten Sensor [11] Anwendung. Der größere Bereich der Abdeckung resultiert jedoch in einer geringeren Auflösung der Position. Somit wird die Positionsbestimmung kleinerer Objekte unmöglich. Es ist in diesen Fällen lediglich eine Bereichsangabe erreichbar. Für die Untersuchung der Aufweitung des Laserstrahls des Velodyne HDL-64E wurde hierfür eine im nahen Infrarotbereich empfindliche Kamera eingesetzt, um das Laserlicht sichtbar zu machen.

Der schematische Messaufbau ist in Abbildung 2.9 dargestellt. Der Abstand d_o zwischen Velodyne HDL-64E und dem Schachbrett wurde auf 7.37 *m* gemessen.

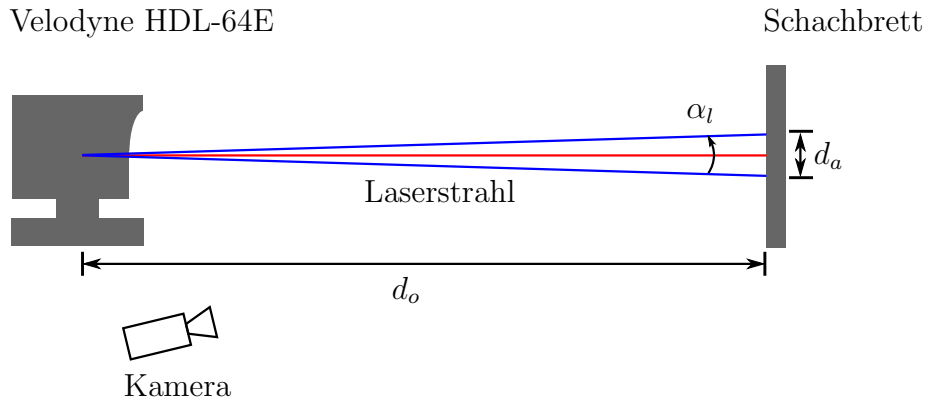


Abbildung 2.9: Schematische Darstellung des Messaufbaus zur Bestimmung der Strahlaufweitung des Velodyne HDL-64E. In rot ist die optische Achse, in blau ist die Aufweitung des Laserstrahls dargestellt.

Abbildung 2.10 zeigt das Kamerabild und einen vergrößerten Ausschnitt mit den gemessenen Werten für die Aufweitung des Laserstrahls. Es ergibt sich für $d_a = 0.015 \text{ m}$.

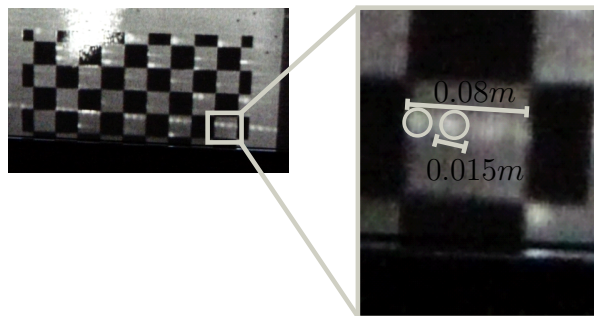


Abbildung 2.10: Aufweitung des Laserstrahls des Velodyne HDL-64E

Die Strahlaufweitung berechnet sich wie folgt:

$$\alpha_l = 2 \arctan \left(\frac{d_a}{2d_o} \right) \quad (2.3)$$

Nach Einsetzen der gemessenen Werte ergibt sich $\alpha_l = 2 \text{ mrad}$ was nach Craig Glennie und Derek Lichti [38] auch den Herstellerangaben entspricht.

2.2.2 Radar

Der Begriff Radar steht als Abkürzung für die englische Bezeichnung Radio Detection and Ranging. Ein Radar sendet elektromagnetische Wellen aus und detektiert sie nach der Reflexion an Objekten wieder. Die im Automobilbereich verwendeten Frequenzen der Radarsensoren bewegen sich, nach Hermann Winner et al. [93], in den Frequenzbändern von $24.0 - 24.25 \text{ GHz}$, $76 - 77 \text{ GHz}$ und $77 - 81 \text{ GHz}$. Für den Nahbereich findet auch das Frequenzband von $21.65 - 26.65 \text{ GHz}$ Anwendung.

Nach Vipul Jain et al. [91] lassen sich die relevanten Radararchitekturen in zwei Kategorien unterteilen: Puls- und Dauerstrichradare.

Die Pulsradare verwenden im Prinzip das bereits bei den Lasersensoren beschriebene Verfahren der Laufzeitmethode ('2.2.1.1 Distanzmessung'). Es wird ein impulsförmiges Signal vom Sensor ausgesendet, dieses breitet sich aus, wird an einem Objekt reflektiert und im Sensor wieder detektiert. Die Distanz zwischen Objekt und Sensor lässt sich über den Zusammenhang in Gleichung 2.4 zwischen der Laufzeit t_{tof} der elektromagnetischen Welle und der Ausbreitungsgeschwindigkeit c_r in einem Medium berechnen:

$$d = \frac{c_r t_{tof}}{2} \quad (2.4)$$

Die Geschwindigkeitsbestimmung erfolgt über die Messung des Dopplereffekts. Dieser beschreibt eine Frequenzverschiebung des ausgesandten Signals bei sich bewegenden Objekten, an denen die elektromagnetische Welle reflektiert wird. Aufgrund der geringeren Frequenzen beim Radar, verglichen mit dem Laser, lässt sich hier der Dopplereffekt mit geringerem und kostengünstigerem Aufwand nutzen.

Eine direkte Laufzeitmessung, wie zuvor beschrieben, ist technisch aufwändig zu realisieren. Aus diesem Grund wird im Folgenden eine indirekte Messmethode beschrieben, welche die Distanz- und die Geschwindigkeitsmessung zugleich ermöglicht.

Dauerstrichradare senden kontinuierlich ein Signal. Dabei wird, um sowohl die Geschwindigkeit als auch die Distanz eines Objekts gleichzeitig zu bestimmen, ein moduliertes Signal verwendet. Dieses Verfahren ist als Dauerstrich-Frequenzmodulation (Frequency Modulated Continuous Wave, FMCW) bekannt (Hermann Winner et al. [93]). Als Beispiel kann die Beschreibung von Konrad Reif [73] verwendet werden, bei der das Sendesignal kontinuierlich in seiner Frequenz entsprechend eines Sägezahns verändert und ausgesandt wird. Dabei entsteht ein Verlauf wie in Abbildung 2.11 dargestellt. Das Sendesignal ist mit f_s und das empfangene Signal, ohne relative Geschwindigkeit zwischen Sensor und Objekt, ist mit f_e gekennzeichnet. Das Empfangssignal ist aufgrund der Laufzeit, die es zum Objekt und wieder zurück benötigt hat, zeitlich zum gesendeten Signal verzögert. Dadurch ergibt sich eine Differenz Δf zwischen den beiden Signalen auf der Frequenzachse. Aus der Frequenzdifferenz Δf und dem Wissen über die Steigung $\frac{df_s}{dt}$ der Rampen sowie der Ausbreitungsgeschwindigkeit der elektromagnetischen Welle c_r lässt sich der Abstand bestimmen:

$$d = \left(\frac{df_s}{dt} \right)^{-1} |\Delta f| \frac{c_r}{2} \quad (2.5)$$

Liegt eine relative Geschwindigkeit zwischen Sensor und Objekt vor, ergibt sich der in Abbildung 2.11 mit f_{ev} gekennzeichnete Verlauf. f_{ev} weist eine gegenüber dem Empfangssignal f_e , bei dem keine relativen Geschwindigkeiten zwischen Sensor und Objekt vorliegen, erhöhte Frequenz f_s auf. Der Frequenzunterschied ist mit Δf_D gekennzeichnet.

Dieser ist auf den Dopplereffekt zurückzuführen. Die Differenz zwischen den Signalverläufen unterscheidet sich in der ansteigenden und der abfallenden Rampe. Die Frequenzunterschiede werden durch Δf_1 für die ansteigende und durch Δf_2 für die abfallende Rampe beschrieben. Es gelten die folgenden Zusammenhänge für den Abstand d und die relative Geschwindigkeit v_{rel} :

$$d \propto \Delta f_1 + \Delta f_2 \quad (2.6)$$

$$v_{rel} \propto \Delta f_2 - \Delta f_1 \quad (2.7)$$

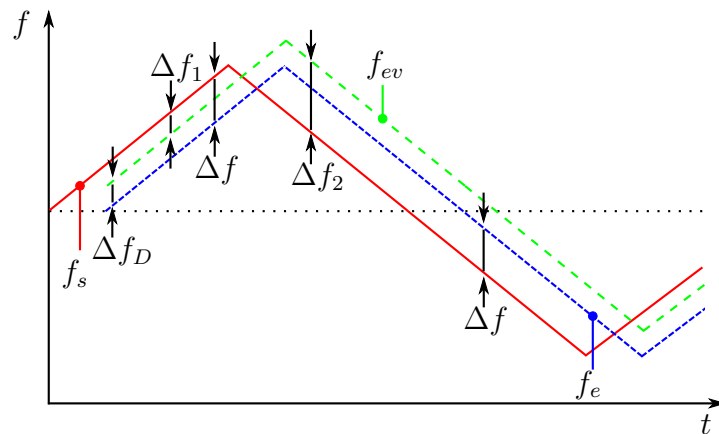
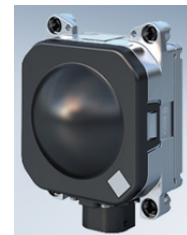


Abbildung 2.11: FMCW Verfahren

Abbildung 2.12 zeigt zwei Beispiele für Radarsensoren, die in aktuellen Fahrzeugmodellen für Fahrerassistenzfunktionen eingesetzt werden. In Abbildung 2.12 (a) ist der ARS300 der Continental AG [12], in Abbildung 2.12 (b) der LRR3 der Robert Bosch GmbH [74] dargestellt. Detaillierte Beschreibungen zu beiden Sensoren finden sich bei Hermann Winner et al. [93].



(a) ARS300 Radarsensor (Quelle: Continental AG) (b) LRR3 Radarsensor (Quelle: Robert Bosch GmbH)

Abbildung 2.12: Radarsensoren für den automobilen Bereich

2.2.3 Kamera

Kameras kommen schon längere Zeit als Sensoren in Fahrzeugen zur Anwendung. Seit 2013 sind zusätzlich zu den bekannten Monokameras auch Stereokameras in Serienfahrzeugen als Unterstützung für Fahrerassistenzsysteme erhältlich.

2.2.3.1 Monokamera

Bei einer Monokamera handelt es sich um eine einzelne Kamera. Durch die von ihr aufgezogenen Bilder lassen sich beispielsweise Funktionen wie die Verkehrszeichenerkennung realisieren. Weitere Funktionen sind die Warnung beim Verlassen der Spur oder das automatische Abblenden des Fernlichts.

Für die Unterstützung beim Einparken werden neuerdings die Bilder von mehreren Monokameras kombiniert, um eine Vogelperspektive auf das Fahrzeug anzudeuten. Die Monokameras befinden sich dabei meist in den Außenspiegeln, dem Kühlergrill und am Kofferraum.

2.2.3.2 Stereokamera

Über die Zeit betrachtet können Monokameras die Umgebung dreidimensional erfassen. Durch die direkte Kombination von zwei Kameras zu einem Stereosystem entsteht ein Sensor, der die dreidimensionale Umgebungserfassung mit nur einer Messung ermöglicht.

Ab dem Modelljahr 2013 der Mercedes E-Klasse findet sich eine entsprechende Stereokamera als Sensor in einem Serienfahrzeug [15].

Die zwei Kameras müssen räumlich voneinander versetzt sein. Voraussetzung ist jedoch, dass sich der Sichtbereich beider Kameras größtmöglich überschneidet. Hieraus folgt, dass die gleiche Szene von beiden Kameras aus einer leicht unterschiedlichen Perspektive erfasst wird. Die räumliche Verschiebung und Drehung der beiden Kameras zueinander wird über den Begriff der extrinsischen Parameter zusammengefasst und muss bekannt sein. Dies gilt ebenso für die intrinsischen Parameter, wie z.B. die Brennweite oder der Hauptpunkt, die ebenfalls bekannt sein müssen, um eine ideale Stereorepräsentation zu erhalten.

Es wird vereinfacht angenommen, dass eine Szene von beiden Kameras betrachtet wird und dass beide Kameras lediglich um einen Wert entlang der y -Achse zueinander verschoben sind. Die Verschiebung entlang der y -Achse wird als Basisbreite b bezeichnet. Ferner wird angenommen, dass die Kameramodelle einer Lochkamera entsprechen. Ist zusätzlich bekannt, welcher Bildpunkt im einen Kamerabild dem gleichen im anderen Kamerabild entspricht, lässt sich nach Bernd Jähne [51] der Abstand r_O eines Objekts bezüglich der beiden Kameras bestimmen. Es folgt der Zusammenhang

$$r_O = b \frac{f}{D} \quad (2.8)$$

Dabei entspricht f der Brennweite der verwendeten Objektive und D der Disparität. Die Disparität ergibt sich aus der Subtraktion der Abstände, die bei der Abbildung in jeder Kamera gegenüber der optischen Achse entstehen. Erweitern lässt sich das Verfahren, um die Raumkoordinaten zu bestimmen.

Abbildung 2.13 zeigt beispielhaft eine für Fahrerassistenzsysteme entwickelte Stereokamera der Continental AG.

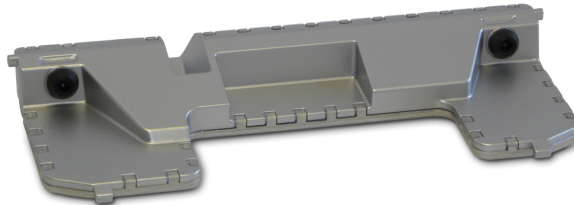


Abbildung 2.13: Stereokamera der Continental AG

2.3 Bewegungskompensation bei scannenden Sensoren

Die Effekte, die bei der Arbeit mit Kameras durch den Rolling Shutter (Christian Demant [17]) auftreten, finden sich auch bei scannenden Lasersensoren. Wie bei Kameras, die ihr Bild zeilen- oder spaltenweise aufbauen, werden die Daten bei einem scannenden Lasersensor spaltenweise aufgebaut. Die Daten von einem Messzyklus werden zusammengefügt und mit einem Zeitstempel versehen. Wenn sich der Lasersensor während der Messung bewegt, werden die Abstände zwar richtig gemessen, passen jedoch nicht zur ursprünglichen Position auf die die Daten bezogen werden. Ist die Bewegung des Sensors während eines Messzyklus bekannt, können die Daten korrigiert werden

Während eines Messzyklus kann vereinfachend angenommen werden, dass sich der Sensor mit konstanter Geschwindigkeit auf einer Kreisbahn bewegt. Dadurch hat der Sensor während eines Messzyklus eine Bewegung in x und y , sowie eine Gierbewegung vollzogen. Nicken und Rollen des Sensors soll vernachlässigt werden. Es wird lediglich die Drehung um die z -Achse beachtet. Abbildung 2.14 zeigt die Bewegung des Sensors auf der Kreisbahn und die globale Position und Orientierung, sowie deren Änderungen (Δx , Δy und $\Delta \Psi$).

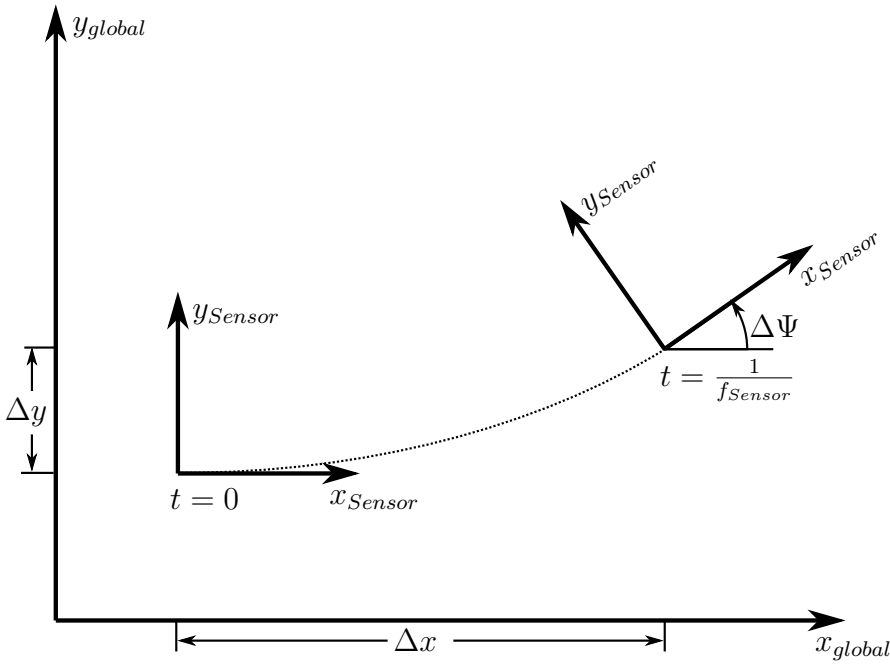


Abbildung 2.14: Bewegung des Sensors während einem Messzyklus

Abbildung 2.15 veranschaulicht den Effekt durch ein fiktives Beispiel. Der Sensor bewegt sich mit einer Tangentialgeschwindigkeit von $v_x = 24 \frac{m}{s}$ und einer Winkelgeschwindigkeit von $\omega = 4.0 \frac{rad}{s}$ in einem sechseckigen Raum. Die Winkelgeschwindigkeit wurde für dieses Beispiel als sehr hoch angenommen, um den Effekt der Drehung sichtbar zu machen. Abhängig davon, wo der Sensor mit der Messung beginnt, weisen die Messpunkte die entsprechende Verschiebung und Drehung auf. Im gezeigten Beispiel beginnt der Sensor unter dem Winkel $\alpha_{Offset} = -\frac{\pi}{2}$. Die erste Messung ist mit $t = 0$ gekennzeichnet und die Drehrichtung des Sensors ist im Uhrzeigersinn. Die letzte Messung ist durch $t = \frac{1}{f_{Sensor}}$ gekennzeichnet, wobei f_{Sensor} die Drehfrequenz der Messeinheit des Sensors ist. Es lässt sich über das Fortschreiten der Zeit eine entsprechende Veränderung der gemessenen im Vergleich mit der wahren Position beobachten. Durch die Bewegung entlang der x -Achse entsteht dabei ein Versatz der Messungen in x -Richtung. Mit der Überlagerung der Bewegung des Sensors auf einer Kreisbahn entsteht ein zusätzlicher Versatz in y -Richtung. Die Kompensation der eigenen Bewegung lässt sich über die Zusammenhänge in Gleichung 2.9 und Gleichung 2.10 erreichen, wobei φ der Azimuth-Winkel ist, unter dem der Sensor einen Punkt gemessen hat.

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} \cos(-t\omega) & -\sin(-t\omega) \\ \sin(-t\omega) & \cos(-t\omega) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} r \cos(\gamma) \\ r + r \sin(\gamma) \end{pmatrix} \quad (2.9)$$

$$t = \begin{cases} \frac{\varphi - \alpha_{Offset}T}{2\pi} & \text{falls } \varphi \geq \alpha_{Offset} \\ \frac{\varphi + 2\pi - \alpha_{Offset}T}{2\pi} & \text{sonst} \end{cases} \quad (2.10)$$

wobei gilt: $\gamma = t\omega + \frac{3}{2}\pi$, $T = \frac{1}{f_{Sensor}}$, $r = \frac{v_x}{\omega}$

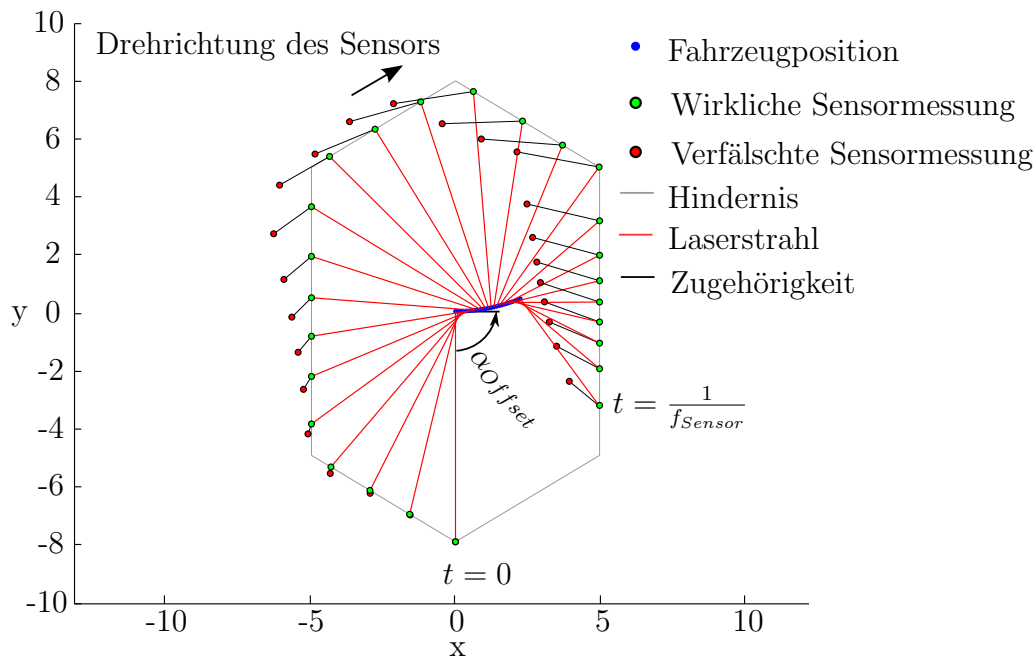


Abbildung 2.15: Bewegung des Sensors während eines Messzyklus

Die getroffenen Annahmen gelten nur für stehende Objekte, die vom Sensor gemessen werden. Bei sich bewegend Objekten, wie beispielsweise Fahrzeugen, ist jedoch der überstrichene Azimutwinkel über dem Objekt entsprechend klein, so dass der Effekt vernachlässigt werden kann. Die Kompensation der Sensorbewegung wird auf alle folgenden gezeigten Messungen angewendet.

2.4 Grundbegriffe der Wahrscheinlichkeitstheorie

Im Verlauf der Arbeit werden drei verschiedene Filterverfahren vorgestellt, die jeweils auf der Wahrscheinlichkeitstheorie aufbauen. Dazu gehört der Bayes-Filter ('2.5 Bayes-Filter'), der Kalman-Filter ('4.15.3.2 Linearer Kalman-Filter') und der Partikel-Filter ('5.2.1 Der Partikel-Filter').

Wenn X eine Zufallsvariable ist und x ein Wert, der von X angenommen werden kann, so bezeichnet

$$p(X = x) \tag{2.11}$$

die Wahrscheinlichkeit, dass X den Wert x annimmt. x wird allgemein auch als Ereignis bezeichnet. Die Schreibweise lässt sich zu $p(x)$ abkürzen, falls die Zufallsvariable bekannt ist. Im kontinuierlichen Fall wird $p(x)$ als Wahrscheinlichkeitsdichte bezeichnet.

Die Wahrscheinlichkeit für das Ereignis x liegt im Intervall $[0, 1]$ sodass gilt

$$0 \leq p(x) \leq 1 \quad (2.12)$$

Die Summe im diskreten Fall, wie auch das Integral im kontinuierlichen Fall, über alle Ereignisse ergibt jeweils den Wert Eins. Somit gilt

$$\sum_x p(x) = 1 \quad (2.13)$$

$$\int p(x)dx = 1 \quad (2.14)$$

Werden zwei Zufallszahlen X und Y betrachtet, so wird $p(x, y)$ als Verbundwahrscheinlichkeit bezeichnet. Dies ist die Wahrscheinlichkeit, dass beide Ereignisse eintreten. Falls beide Ereignisse jedoch unabhängig sind, gilt

$$p(x, y) = p(y, x) = p(x)p(y) \quad (2.15)$$

Für die bedingte Wahrscheinlichkeit, dass x eintritt, falls y bereits eingetreten ist und $p(y) \neq 0$, gilt

$$p(x|y) = \frac{p(x, y)}{p(y)} \quad (2.16)$$

Falls die beiden Ereignisse statistisch unabhängig sind, gilt

$$p(x|y) = p(x) \quad (2.17)$$

Der Zusammenhang

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (2.18)$$

mit $p(y) \neq 0$ wird als Satz von Bayes bezeichnet und findet im folgenden Kapitel ('2.5 Bayes-Filter') für die Zustandsschätzung Anwendung. Wird der Satz von Bayes auf drei Variablen erweitert, folgt

$$p(x|y, z) = \frac{p(y|x, z)p(x|z)}{p(y|z)} \quad (2.19)$$

2.5 Bayes-Filter

Ausgangspunkt für den Bayes-Filter ist die Schätzung einer Zustandsgröße \vec{x} bei der es sich in aller Regel um eine mehrdimensionale Größe handelt. Da die Messungen \vec{y} , die zur Schätzung des Zustands \vec{x} beitragen, Unsicherheiten aufweisen, werden diese durch eine Wahrscheinlichkeitsdichte $p(\vec{y})$ ausgedrückt. Dies führt zu einer Unsicherheit der Zustandsgröße, die daher über die Wahrscheinlichkeitsdichte $p(\vec{x})$ beschrieben wird.

Die Änderung des Systemzustandes zwischen zwei aufeinanderfolgenden diskreten Zeitschritten wird über die Zustandsübergangsfunktion f beschrieben. Die diskreten Zeitschritte werden dabei über den Index k ausgedrückt. Es wird angenommen, dass die Störgröße \vec{s} eine bekannte Wahrscheinlichkeitsverteilung aufweist. Falls Steuergrößen, die auf das System einwirken, bekannt sind, werden diese über den Steuervektor \vec{u} in das System eingebracht. Es ergibt sich die Zustandsübergangsfunktion f zu

$$\vec{x}_k = f(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{s}_{k-1}) \quad (2.20)$$

Gleichung 2.20 wird auch als Bewegungsmodell bezeichnet.

Die Messgleichung wird über

$$\vec{y}_k = g(\vec{x}_k, \vec{w}_k) \quad (2.21)$$

beschrieben. Dabei weist \vec{w} eine bekannte Wahrscheinlichkeitsverteilung für das Messrauschen auf, das statistisch unabhängig von dem von \vec{s} ist.

Der zu schätzende Zustand \vec{x} wird als bedingte Wahrscheinlichkeitsverteilung geschrieben

$$p(\vec{x}_k | \vec{y}_{1:k}, \vec{u}_{1:k-1}) \quad (2.22)$$

Es fließen alle bisherigen Messungen bis zum Zeitpunkt k und alle bisherigen Steuereingaben bis zum Zeitpunkt $k-1$ in die Wahrscheinlichkeitsverteilung ein. Man spricht hierbei von der a posteriori Wahrscheinlichkeitsdichtefunktion. Falls die Integration der letzten Messung noch fehlt, spricht man von der a priori Wahrscheinlichkeitsdichtefunktion, die sich zu

$$p(\vec{x}_k | \vec{y}_{1:k-1}, \vec{u}_{1:k-1}) \quad (2.23)$$

ergibt.

Durch Anwendung der Markov-Annahme, dass ein Zustand nur vom vorherigen Zustand abhängt, und der Regel von Bayes aus Gleichung 2.19, ergibt sich der Bayes-Filter mit seinen zwei Schritten.

Prädiktion:

$$p(\vec{x}_k | \vec{y}_{1:k-1}, \vec{u}_{1:k-1}) = \int p(\vec{x}_k | \vec{x}_{k-1}, \vec{u}_{k-1}) p(\vec{x}_{k-1} | \vec{y}_{1:k-1}, \vec{u}_{1:k-2}) d\vec{x}_{k-1} \quad (2.24)$$

Innovation:

$$p(\vec{x}_k | \vec{y}_{1:k}, \vec{u}_{1:k-1}) = \eta p(\vec{y}_k | \vec{x}_k) p(\vec{x}_k | \vec{y}_{1:k-1}, \vec{u}_{1:k-1}) \quad (2.25)$$

mit

$$\eta^{-1} = \int p(\vec{y}_k | \vec{x}_k) p(\vec{x}_k | \vec{y}_{1:k-1}, \vec{u}_{1:k-1}) d\vec{x}_k \quad (2.26)$$

als Normalisierungsfaktor.

Herleitungen finden sich beispielsweise bei Sebastian Thrun et. al [88] oder bei Zhe Chen [9].

2.6 Erzeugung von Referenzdaten mit Inertialmesssensorik und Globalem Positionssystem mit Differentialsignal (DGPS)

Für die Erzeugung von Referenzdaten im allgemeinen gibt es eine Vielzahl an Methoden, die sich je nach Anwendungsfall unterscheiden.

Im Bereich der Bildverarbeitung hat sich das Markieren von relevanten Bildbereichen durch den Menschen als eine weit verbreitete Methode durchgesetzt. Dieses Verfahren findet etwa bei der Entwicklung von Fahrerassistenzsystemen zur Fußgängererkennung Anwendung. Hierbei werden speziell Fußgänger in Bildsequenzen markiert. Diese Daten werden daraufhin beispielsweise für das Training eines Klassifikators verwendet oder als Vergleichsdaten genutzt um den entwickelten Algorithmus zu testen. Eine Aufstellung von verschiedenen Datensätzen findet sich bei Markus Enzweiler et al. [26]. Ein weiterer Datensatz, in dem allgemeine Kategorien den Bildbereichen zugeordnet wurden, wird von Bryan Russell et al. [76] beschrieben.

Frederik Sarholz et al. [78] nutzen die vom Fahrzeug abgefahrte Trajektorie als Referenzdaten für die Straßenverlaufsschätzung. Dabei werden die Odometriedaten, welche sich aus den im Fahrzeug bereits verbauten Sensoren (Rad-, Drehraten- und Beschleunigungssensoren) ergeben, genutzt. Diese können über kurze Zeitbereiche als zuverlässige Beschreibung des Straßenverlaufs angesehen werden. Bei der Auswertung wird der durch Radardaten geschätzte Straßenverlauf mit dem aus den Odometriedaten verglichen. Dieses Verfahren lässt sich nur in einer Offline-Verarbeitung oder mit einer Verzögerung der Messdaten realisieren.

Falls entsprechende physikalische Modelle für Sensoren und die zu vermessenden Objekte vorhanden sind, lassen sich für diese ebenfalls die zugehörigen Sensordaten aus definierten Szenarien künstlich erstellen. Virginia Garate et al. [35] beschreiben beispielsweise eine Simulationssoftware für das Testen von Fahrerassistenzsystemen mit dem Fokus auf Fußgänger. Dabei werden Radar- und Kameramessungen simuliert. Michael Gschwandtner [40] beschreibt die Entwicklung einer Software namens Blesor, die in Blender [5], einem 3D-Modellierungswerkzeug, integriert wurde. Dabei wurden verschiedene Modelle von Lasersensoren mit ihren physikalischen Eigenschaften, wie dem Rauschen der Messdaten, implementiert.

Die Grundlage für ein Inertialmesssystem bildet die Verbindung von Drehraten- und Beschleunigungssensoren. Durch Integration der Sensordaten ist somit eine relative Positionsbestimmung, bezogen auf den Startpunkt, möglich.

Eine Drift der Sensordaten ist, bedingt durch die Hardware, bei einem Inertialmesssystem jedoch unvermeidbar. Dies führt über die Zeit betrachtet zu einer steigenden Ungenauigkeit in der Positionsbestimmung. Durch die nötige Integration der Sensordaten wirkt sich der Effekt der Drift sogar verstärkt bei der Positionsbestimmung aus.

Um den Fehler in der Positionsbestimmung, der durch die Drift entsteht, zu kompensieren und um eine globale Positionsbestimmung zu erhalten, wird ein globales satellitengestütztes Navigationssystem verwendet. Dabei ist das NAVSTAR GPS (Navigational Satellite Timing and Ranging Global Positioning System), welches oft nur mit GPS bezeichnet wird, das am häufigsten verwendete System. Entwickelt wurde es vom amerikanischen Verteidigungsministerium. Neben GPS gibt es aktuell das europäische System Galileo, das russische System Glonass und das chinesische System Beidou.

Um das globale Positionssignal aus den Satellitendaten weiter zu verbessern, werden Korrekturdaten von Bodenstationen genutzt. Hierbei handelt es sich um Differenzsignale wodurch sich der Name DGPS (Differential Global Positioning System) für diese Art von Systemen ergibt. Die Differenzsignale errechnen die Bodenstationen, deren globale Position hinreichend genau bekannt ist, über den Vergleich ihrer eigenen und der über die Satellitendaten berechneten Position. Die aktuelle Abweichung wird daraufhin an die sich im Umfeld aufhaltenden DGPS-Empfänger verteilt. Die Verteilung erfolgt beispielsweise beim Satellitenpositionierungsdienst SAPOS [19] über eine GSM-Verbindung (Global System for Mobile Communication). Mit der Verwendung von Differenzsignalen lassen sich die globalen Positionen, welche grob eine Genauigkeit im einstelligen m -Bereich haben, in den einstelligen cm -Bereich verbessern. Eine ausführliche Beschreibung findet sich bei Hans Dodel et al. [21].

Entsprechende Produkte werden unter anderem von den Firmen GeneSys Elektronik GmbH [36] und iMAR Navigation Solutions [49] angeboten. Abbildung 2.16 zeigt zwei Produktbeispiele der genannten Hersteller.



(a) GeneSys Elektronik GmbH: ADMA-G



(b) iMAR Navigation Solutions: iTraceRT-F200

Abbildung 2.16: Zwei Produktbeispiele für Inertialmesssysteme mit Möglichkeiten zur Erweiterung um ein globales Positionssystem mit Differentialsignal

Mit einer Kombination aus Inertialmesssensorik und DGPS sind verschiedene Varianten zur Erzeugung von Referenzdaten möglich.

Der einfachste Aufbau besteht darin, dass ein Messfahrzeug mit der Inertialmesssensorik und dem DGPS ausgerüstet wird. Marcus Konrad et al. [55] nutzen diesen Aufbau für die Evaluierung einer Straßenverlaufsprädiktion. Die Prädiktion erfolgt durch eine Kombination von Belegungskarten aus Sensordaten und digitalen Karten. Matthias Schmid

2. Grundlagen

[80] verwendet ebenfalls einen derartigen Aufbau für die Auswertung von Algorithmen zur simultanen Selbstlokalisierung und Kartierung.

Für die im späteren Verlauf folgenden Auswertungen ('4 Außenkonturbestimmung', '5 Objektverfolgung mit einem objekt-lokalen Gittermodell') wurden zwei Fahrzeuge mit Inertialmessensorik und DGPS ausgerüstet. Damit ist es möglich die Positionen und Orientierungen beider Fahrzeuge zeitgleich in einem globalen Koordinatensystem präzise zu bestimmen. Zusätzlich wurde in eines der Fahrzeuge ein Fahrroboter integriert. Somit lassen sich zuvor definierte Szenarien auch wiederholt abfahren. Ein Beispiel für einen Aufbau eines Fahrroboters zeigt Abbildung 2.17.



(a) Fahrzeug ausgerüstet mit einem Fahrroboter und Inertialmessensorik mit DGPS



(b) Fahrroboter

Abbildung 2.17: Beispiel für einen Fahrroboter

Bei der Arbeit mit Messungen von einem Lasersensor stellt die Segmentierung einen ersten Schritt in der Datenverarbeitungskette dar.

Die Segmentierung ist ein Verfahren, das aus einer großen Datenmenge zusammenhängende kleinere Strukturen bestimmt. Dabei sind die Art der Datenmenge und das Anwendungsfeld zunächst unerheblich. Die Segmentierung findet sich beispielsweise in der Bildverarbeitung oder im Data-Mining. Im Data-Mining wird auch von der Clusteranalyse gesprochen.

Dreidimensionale Punktwolken lassen sich auf verschiedenste Arten bearbeiten. Zunächst können sie als einzelne Punkte aufgefasst werden. Dies ermöglicht die Anwendung von punktbasierten Ansätzen, wie sie in Abschnitt '3.1 Punktbasierte Ansätze' vorgestellt werden. Je nach Sensor weisen die dreidimensionalen Punkte eine natürliche Sortierung auf, die es auch möglich macht, die Punkte als Bild aufzufassen. Dies soll in Abschnitt '3.2 Bildbasierter Ansatz' beschrieben werden. Mit der Reduktion der Datenmenge um eine Dimension folgt oft eine erhebliche Verringerung der Rechenzeit bei dennoch guter Qualität des Ergebnisses der Segmentierung. Dabei erfolgt meist zusätzlich eine Diskretisierung der Daten auf ein regelmäßiges zweidimensionales Gitter. In '3.3.1 2D Ansatz' werden zwei dieser Verfahren vorgestellt. '3.3.2 3D Ansatz' stellt ein Verfahren vor, das als Ausgangspunkt ein dreidimensionales Gitter nutzt.

Alle vorgestellten Verfahren, bis auf den bildbasierten Ansatz, wurden für die folgenden Vergleiche implementiert und gehen von einer Punktmenge aus, die von einem Velodyne HDL-64E aufgezeichnet wurde. Dabei ist zu beachten, dass sich auch Messungen des Bodens in der Punktmenge befinden. Diese verhindern in fast allen Fällen die direkte Anwendung der Verfahren zur Segmentierung. Daher werden zusätzlich zu den Verfahren zur Segmentierung passende Algorithmen zur Erkennung und Entfernung von Bodenpunkten vorgestellt.

3.1 Punktbasierte Ansätze

Bei den punktbasierten Ansätzen wird jeder Punkt in der Punktwolke einzeln behandelt. Dabei werden jeweils die Punkte in der Nachbarschaft zur Segmentierung einbezogen.

Dies erfordert zunächst das Bestimmen und Löschen von Bodenpunkten. Bei Bodenpunkten handelt es sich um Punkte, die einen vom Fahrzeug befahrbaren Bereich definieren. Dies sind zum Beispiel Punkte, die auf der Straße oder in einer Hofeinfahrt gemessen wurden.

3.1.1 Bestimmung von Bodenpunkten

Ein sehr einfaches und schnelles Verfahren zur Bodenpunkterkennung stellt die Annahme des Bodens als horizontale Ebene dar. Für einige Anwendungen, wie die Erkennung von Bodenpunkten auf einer Autobahn, einem Parkplatz oder auch auf kleineren Straßen, trifft die Annahme einer Ebene zumindest im Nahbereich bis ca. 25 – 40m zu.

Die Bodenebene wird mit einem robusten Schätzverfahren, wie beispielsweise dem RANSAC Algorithmus (Random Sample Consensus) von Martin Fischler et. al [29], bestimmt. Dabei wird die räumliche Orientierung der Ebene im Fahrzeugkoordinatensystem geschätzt. Alle Punkte, die der Ebene über einen Distanzparameter d im Suchbereich S zugeordnet wurden, werden als Bodenpunkte markiert. Abbildung 3.1 zeigt schematisch die Trennung zwischen Bodenpunkten und nicht zugeordneten Punkten. Das Beispiel zeigt das Fahrzeug bei konstanter Geschwindigkeit und bei Geradeausfahrt. Befindet sich das Fahrzeug in einem Beschleunigungsvorgang, hebt sich das Chassis vorne an und gegenüber dem Fahrzeugkoordinatensystem vollzieht der Sensor auf dem Dach eine Nickbewegung. Somit liegen die Messungen des Laser Sensors nicht mehr in der xy -Ebene des Fahrzeugkoordinatensystems, sondern weisen gegenüber diesem einen Nickwinkel auf. Mit der Schätzung der Bodenebene lässt sich dieser bestimmen und kompensieren. Gleiches gilt für Rollbewegungen des Sensors in Kurven.

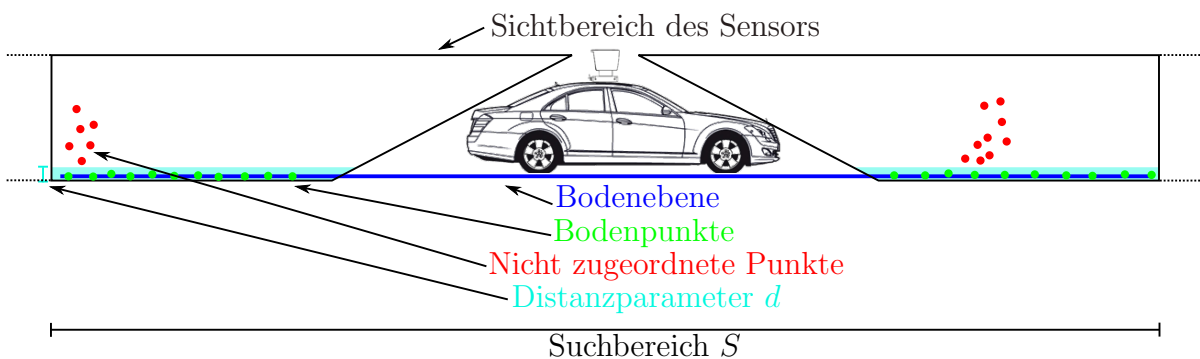
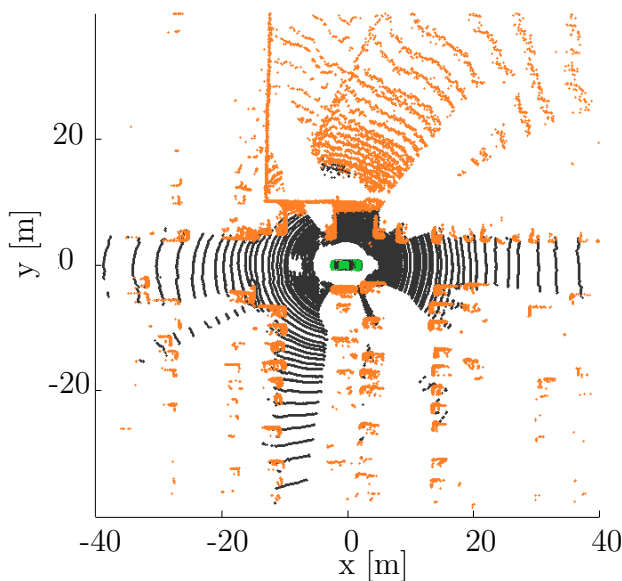


Abbildung 3.1: Schematische Darstellung zur Erkennung von Bodenpunkten über die Annahme einer Ebene. Zur besseren Visualisierung ist ein Schnitt durch die Ebene dargestellt.

Es empfiehlt sich trotz der Verwendung eines robusten Schätzverfahrens die Parameter der Ebene über die Zeit zu glätten. Hierzu werden die Parameter in einem Ringpuffer gespeichert. Für die Bestimmung der aktuellen Ebene werden alle Ebenen, deren Parameter im Ringpuffer gespeichert sind, berechnet. Es werden jeweils drei Punkte auf den Ebenen bestimmt und in alle Punkte wird eine Ebene mit kleinstem quadratischem

Abstand eingepasst.

Abbildung 3.2 (a) zeigt beispielhaft das Ergebnis der Bodenpunkterkennung auf einem Parkplatz. Die Bodenpunkte sind schwarz, die restlichen Punkte orange markiert. Das Messfahrzeug ist durch ein grünes Fahrzeugmodell an der Position $(0, 0)$ dargestellt. Zur besseren Darstellung wurden die Messungen, die weiter als $40m$ in x - und y -Richtung entfernt liegen, nicht visualisiert. In Abbildung 3.2 (b) ist zur besseren Orientierung eine Luftaufnahme der gleichen Szene dargestellt. Aufgrund der Zeitdifferenzen zwischen den beiden Aufnahmen, stimmen diese nur bedingt überein. Die Struktur des Parkplatzes lässt sich aber dennoch erkennen. Das Messfahrzeug mit dem Lasersensor befand sich auf dem Parkplatz in etwa an der mit einem roten Stern markierten Stelle in Abbildung 3.2 (b).



(a) Ergebnis der Bestimmung von Bodenpunkten (Bodenpunkte in schwarz, andere Punkte in orange)



(b) Vergleichende Luftaufnahme der Szene. Der Standort des Fahrzeugs zum Zeitpunkt der Aufnahme ist mit einem roten Stern gekennzeichnet (Quelle der Luftaufnahme: Google)

Abbildung 3.2: Bestimmung von Bodenpunkten über die Annahme der Fahrbahn als Ebene

3.1.2 Punktbasierte Segmentierungsalgorithmen

Die folgenden Abschnitte zeigen drei verschiedene Algorithmen, die zur Segmentierung von Punktwolken eingesetzt werden können.

3.1.2.1 K-Means

Für die Bestimmung der Segmente muss beim K-Means-Algorithmus [58] die Anzahl k der zu suchenden Segmente bekannt sein. Jedes Segment wird über seinen Segmentmittelpunkt S_m definiert. Zu Beginn werden die Segmentmittelpunkte im Initialisierungsschritt

beliebig im Suchraum verteilt. Im nächsten Schritt, dem Zuordnungsschritt, werden über eine Distanzfunktion die Punkte demjenigen Segmentmittelpunkt zugeordnet, dem sie am nächsten sind. Daraufhin wird im Aktualisierungsschritt der Schwerpunkt über die zugeordneten Punkte neu berechnet und die Segmentmittelpunkte S_m werden in diesen verschoben. Es wiederholt sich der Zuordnungsschritt, bei dem wieder alle Punkte den neuen Segmentmittelpunkten über die Distanzfunktion zugeordnet werden. Die beiden Schritte Zuordnung und Aktualisierung werden so lange durchgeführt, bis eine vorgegebene Anzahl an Iterationen erreicht wurde, oder sich die Segmentmittelpunkte nicht mehr bewegen. Eine detaillierte Beschreibung des Algorithmus findet sich bei David MacKay [58].

Bei der Segmentierung von Daten eines Lasersensors ist in den meisten Fällen die Anzahl der zu suchenden Segmente im Voraus nicht bekannt, was die Anwendung des K-Means-Algorithmus einschränkt. Auch bei bekannter Anzahl der Segmente entstehen durch die zufällig im Suchraum verteilten Segmentmittelpunkte Segmente, die nicht dem gewünschten Ergebnis entsprechen. Abbildung 3.3 zeigt eine Reihe von sechs parkenden Fahrzeugen, die durch den K-Means-Algorithmus segmentiert wurden. Jedes Segment ist durch eine unterschiedliche Farbe dargestellt. Die Szene entspricht einem Ausschnitt aus dem in Abbildung 3.2 gezeigten Parkplatz. Das Ergebnis zeigt eine nicht korrekte Segmentierung, da jeweils zwei Fahrzeuge zu einem Segment zusammengefasst wurden (schwarzes und pinkes Segment) und ein Fahrzeug durch drei Segmente repräsentiert wurde (rot, blau, grün). Für die Segmentierung wurde die K-Means Implementierung der Statistik Toolbox (Version 9.0) von Mathworks Matlab R2014a verwendet.

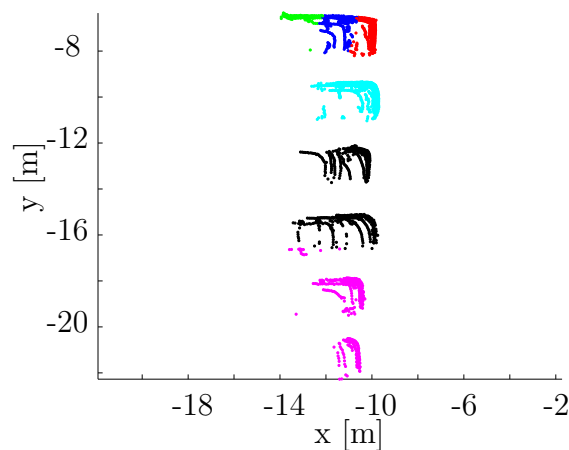


Abbildung 3.3: Segmentierung von sechs parkenden Fahrzeugen mit dem K-Means-Algorithmus

3.1.2.2 Mean-Shift

Der Mean-Shift-Algorithmus gehört zur Kategorie der Dichte basierten Algorithmen. Eine Beschreibung findet sich bei Dorin Comaniciu und Peter Meer [10]. Der Algorithmus basiert auf der Annahme, dass die Punkte von einer Wahrscheinlichkeitsdichtefunktion stammen. Er bestimmt ausgehend von einem Startpunkt x den Gradienten in Richtung

des Bereiches mit einer höheren Punktdichte. Zur Berechnung des Gradienten wird ein Kernel-Dichte-Schätzer verwendet, der die Wahrscheinlichkeitsverteilung der Punkte bestimmt und worüber die Gewichtung der benachbarten Punkte erfolgt. Der Startpunkt x wird entlang des Gradienten verschoben und bildet den Ausgangspunkt für die nächste Bestimmung des Gradienten. Dies wird so lange wiederholt, bis sich der Gradient nicht weiter ändert. Ist dieser Zustand erreicht, ist ein Segmentzentrum gefunden worden.

Bei der Segmentierung werden alle Punkte, die sich in einem entsprechenden Fenster um den sich im Suchraum bewegenden Punkt x befunden haben, einem Segment zugeordnet. Ein neues Segment entsteht, wenn das Segmentzentrum eine genügend große Entfernung von einem anderen Segmentzentrum aufweist. Andernfalls werden die beiden Segmente zu einem Segment vereint. Der Algorithmus wird so lange wiederholt, bis alle Punkte einem Segment zugeordnet sind.

Der Mean-Shift-Algorithmus geht im Gegensatz zum K-Means-Algorithmus von keiner festen Anzahl zu suchender Segmente aus. Als Parameter erwartet der Algorithmus eine Bandbreite h für den Kernel-Dichte-Schätzer. Eine zu groß gewählte Bandbreite führt zu einer inkorrekten Segmentierung. Es entstehen große Segmente, die zum Teil durch Vereinigung von kleineren Segmenten entstehen. Eine zu klein gewählte Bandbreite resultiert in zu vielen kleinen Segmenten. Die Wahl des Parameters muss daher auf die entsprechende Situation und Anwendung angepasst werden.

Abbildung 3.4 zeigt die selbe Szene, die auch für die Auswertung des K-Means-Algorithmus verwendet wurde. Die Farben der Punkte visualisieren die Zuordnung zu einem Segment. Das Ergebnis zeigt eine korrekte Segmentierung aller sechs Fahrzeuge.

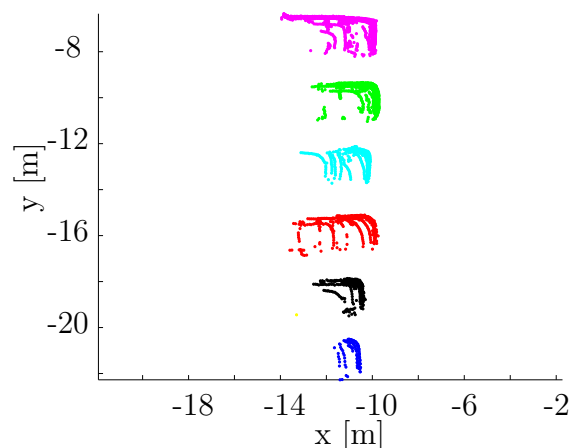


Abbildung 3.4: Segmentierung von sechs parkenden Fahrzeugen mit dem Mean-Shift-Algorithmus

3.1.2.3 DBSCAN

Der DBSCAN-Algorithmus von Martin Ester et al. [27] gehört ebenfalls wie der Mean-Shift-Algorithmus zur Kategorie der Dichte basierten Algorithmen. Auch er benötigt im Vergleich zum K-Means-Algorithmus keine Annahme über die Anzahl der zu suchenden Segmente. Als Startpunkt genügt ein beliebiger Punkt aus der Punktmenge. Über ein Abstandsmaß, meist anhand des euklidischen Abstands, werden Punkte in der Nachbarschaft in einem Radius r um den Startpunkt herum gesucht. Werden Punkte in der Nachbarschaft gefunden, wird die Anzahl der Punkte bestimmt. Entspricht die Anzahl der Punkte nicht dem vorgegebenen Minimalwert k , wird der Punkt als Ausreißer markiert. Finden sich jedoch mindestens k Punkte in der Nachbarschaft wird der Punkt als Segmentpunkt aufgefasst und alle benachbarten Punkte werden diesem Segment zugeordnet. Im folgenden Schritt werden die Nachbarn des Ausgangspunktes auf ihre Nachbarn überprüft. Finden sich weitere Punkte, werden sie dem ursprünglichen Segment zugeordnet. Werden keine weiteren Punkte gefunden, gilt das Segment als abgeschlossen und der Algorithmus beginnt mit einem noch nicht zugeordneten Punkt neu. Der Algorithmus wird so lange wiederholt, bis alle Punkte zugeordnet sind, oder als Ausreißer markiert wurden.

DBSCAN erwartet, wie in der Beschreibung des Algorithmus erwähnt, zwei Parameter. Der erste Parameter beschreibt den Suchradius r , der für die Nachbarschaftsbestimmung genutzt wird. Der zweite Parameter k bestimmt, ab wann ein gültiges Segment entsteht. Die Wahl der beiden Parameter ist von der Situation und der Anwendung abhängig. Ein großes r lässt keine Trennung eng zusammen liegender Punktmenge zu. Wird r zu klein gewählt, zerfallen Segmente in Untersegmente. Der Parameter k steuert die Entstehung von neuen Segmenten. Er erweist sich als nicht besonders kritisch in Situationen mit ausreichender Datendichte. Sollen jedoch auch kleine Punktmenge in verrauschter Umgebung segmentiert werden, gilt dies nicht mehr.

Abbildung 3.5 zeigt die selbe Szene, die auch für die Auswertung des K-Means-Algorithmus und des Mean-Shift-Algorithmus verwendet wurde. Die Farben der Punkte visualisieren die Zuordnung zu einem Segment. Das Ergebnis zeigt, dass alle sechs Fahrzeuge korrekt segmentiert wurden.

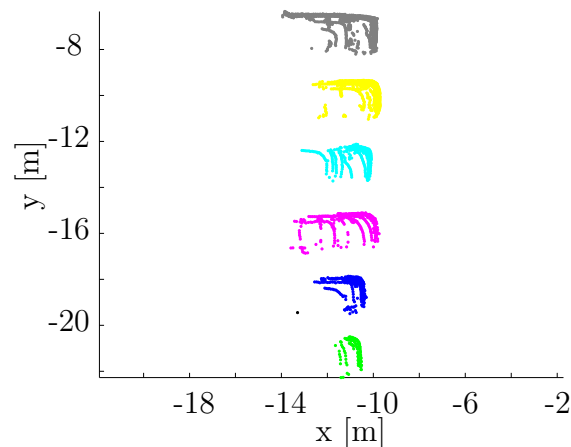


Abbildung 3.5: Segmentierung von sechs parkenden Fahrzeugen mit dem DBSCAN-Algorithmus

3.1.3 Bewertung

Mit Ausnahme des K-Means-Algorithmus zeigen die beiden anderen punktbasierten Ansätze, in den hier untersuchten Anwendungsfällen, sehr gute Ergebnisse bei der Segmentierung. Gleichzeitig erfordern sowohl der Mean-Shift-, als auch der DBSCAN-Algorithmus nur wenige Parameter (Mean-Shift: 1, DBSCAN: 2). Diese hängen jedoch stark von den Anforderungen an das Ergebnis der Segmentierung ab und können zwischen verschiedenen Szenen variieren.

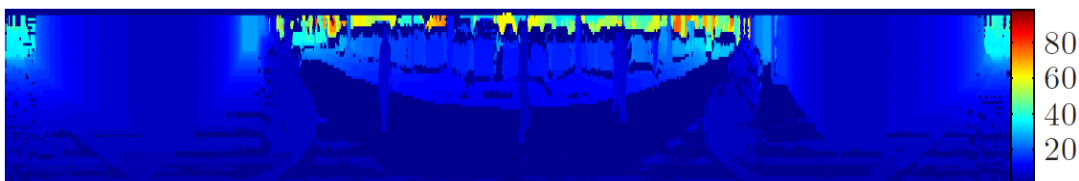
Mit steigender Zahl an Punkten steigt auch die notwendige Zeit für die Berechnung. Der größte Teil der Zeit wird für die Berechnung der benachbarten Punkte verwendet. Dies lässt sich beispielsweise durch effiziente Speicherung der Punkte in Suchbäumen minimieren. Untersuchungen mit dem DBSCAN-Algorithmus in Verbindung mit einem KD-Baum, der von Jerome Friedman et al. [34] vorgestellt wurde, haben jedoch keine echtzeitfähige Implementierung für die hier gezeigten Daten ergeben (Velodyne HDL-64E mit einer Messfrequenz von $10Hz$).

3.2 Bildbasierter Ansatz

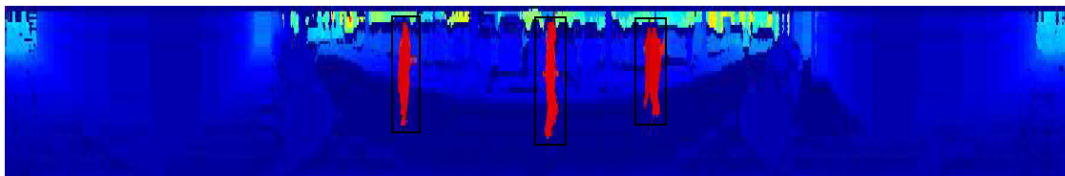
Bildbasierte Interpretationen von Laser Sensor Messungen finden sich nicht sehr häufig in der Literatur. Ein Grund ist sicherlich die zur Zeit noch geringe Auflösung von 4, 8 oder 64 Messungen in vertikaler Richtung (Kapitel '2.2.1 Lasersensor'). Im Vergleich dazu stehen die Stereokamerasysteme (Kapitel '2.2.3.2 Stereokamera') mit mehreren hundert Messungen in vertikaler Richtung. Aus diesem Grund soll der bildbasierte Ansatz für Laser Sensoren hier auch nur kurz erwähnt werden. Interessant dabei ist, dass der bildbasierte Ansatz bei gleicher Datenbasis eine unterschiedliche Betrachtungsweise auf die Problemstellung ermöglicht.

Wie bereits erwähnt, liegen die Messungen des Velodyne HDL-64E in einer natürlichen Sortierung vor. Die 64 Ebenen, welche vertikal angeordnet sind, erzeugen nahezu gleichzeitig jeweils eine Messung. Nach der Messung dreht sich die Messeinheit um die z-Achse und es erfolgt die nächste Messung. Die auf diese Weise entstandene Punktmenge kann auch als Bild aufgefasst werden. Dabei repräsentieren die Pixel die Distanzwerte und die Achsen des Bildes entsprechen den Azimut- und Elevationswinkeln.

Für die Segmentierung des Bildes kann, wie von Bharath Kalyan et. al [54] vorgeschlagen, der Mean-Shift-Algorithmus mit einer anschließenden Nachbarschaftssuche, wie sie vom Prinzip in '3.3.1.2 Erzeugung von Segmenten' vorgestellt wird, verwendet werden. Die Suche nach benachbarten Pixeln im Bild wird bei Bharath Kalyan et. al [54] als Blob-Extraction bezeichnet. Die interpretierten 3D-Messungen als Bild sind in Abbildung 3.6 (a) und das Ergebnis der Segmentierung in Abbildung 3.6 (b) dargestellt. Die Farbskala mit ihren zugeordneten Zahlenwerten visualisiert die Distanzwerte in Metern. Beide Abbildungen sind aus der Arbeit von Bharath Kalyan et. al [54] entnommen und zeigen drei Fußgänger. Auch wenn es in der Arbeit von Bharath Kalyan et. al nicht näher erläutert wird, sollten die Bodenpunkte in einem Vorverarbeitungsschritt entfernt werden. Die entnommenen Bilder aus der Arbeit zeigen einen gleichbleibenden Farbwert für die Bodenpunkte um die Personen, was diesen Schritt nahe legt. Würde dieser Schritt nicht erfolgen, wäre ein Farbverlauf, der sich über die Distanzänderung der Messungen ergibt, entlang des Bodens zu erwarten, welcher alle Objekte miteinander verbindet. Somit wäre eine Trennung der Objekte schwieriger, da die Erkennung des Farbverlaufs und der Übergang in einen konstanten Farbwert im Segmentierungsalgorithmus mit beachtet werden müsste. Die Bestimmung der Bodenpunkte kann beispielsweise im dreidimensionalen Raum mit der Annahme des Bodens als horizontale Ebene ('3.1.1 Bestimmung von Bodenpunkten') erfolgen.



(a) Distanzbild. Jedes Pixel entspricht einer Distanzmessung entsprechend der Farbskala



(b) Segmentiertes Distanzbild. Die roten Bereiche in den Boxen zeigen die Ausgabe des Segmentierungsalgorithmus

Abbildung 3.6: Segmentierung von 3D-Messungen nach der Interpretation der Daten als Bild

3.3 Gitterbasierte Ansätze

Bei gitterbasierten Ansätzen zur Segmentierung wird eine gröbere Diskretisierung der Daten durchgeführt. Durch die Diskretisierung entsteht einerseits eine spezielle Ordnung der Daten, die bei der Bearbeitung ausgenutzt werden kann. Andererseits können Datenpunkte, die nahe zusammen liegen, durch einen Datenpunkt ersetzt werden. Dies führt zu einer Reduktion der Datenmenge was in einer kürzeren Verarbeitungszeit resultiert. Im Folgenden wird zunächst ein 2D und anschließend ein 3D Ansatz vorgestellt.

3.3.1 2D Ansatz

Dreidimensionale Messungen von Lasersensoren lassen sich durch Projektion entlang der z-Achse in ein in der xy-Ebene aufgespanntes zweidimensionales Gitter eintragen. Ein zweidimensionales Gitter wird durch seine aneinanderhängenden Zellen beschrieben. Häufig finden hierbei kartesische Gitter Anwendung. Polargitter wie sie beispielsweise bei Sylvia Pietzsch et al. [71] genutzt werden, werden dagegen weniger verwendet.

Im Folgenden soll ein kartesisches Gitter verwendet werden, das durch seine Zellgröße ν definiert wird. Die Zellgröße entspricht dabei der Kantenlänge eines Quadrats. Ferner wird die Länge und Breite des Gitters festgelegt. Der Lasersensor befindet sich mittig im Gitter.

Werden die Messungen des Lasersensors in die xy-Ebene projiziert, entsteht eine Zuordnung zwischen Messpunkt und Zelle.

3.3.1.1 Bestimmung von Bodenpunkten

Die Bestimmung und Entfernung der Bodenpunkte kann auf gleiche Weise wie bei den punktbasierten Ansätzen ('3.1 Punktbasierte Ansätze') geschehen. Dieser Schritt muss vor der Zuordnung zwischen Messpunkt und Zelle erfolgen.

Eine weitere Möglichkeit bietet sich mit dem zweidimensionalen Gitter. Sobald alle Messpunkte den Zellen zugeordnet wurden, kann pro Zelle die Distanz d_z zwischen höchstem und niedrigstem Messpunkt entlang der z-Achse bestimmt werden. Über einen Parameter d_{mz} lässt sich bestimmen, ob die Punkte in der Zelle innerhalb eines vorgegebenen Bands liegen. Ist dies der Fall, handelt es sich um eine flache Struktur, bei der es sich mit hoher Wahrscheinlichkeit um Bodenpunkte handelt, die daraufhin entsprechend markiert oder gelöscht werden können.

Bezeichnet n_{Zellen} die Gesamtzahl der Zellen im zweidimensionalen Gitter und m_{Zelle_i} mit $i \in \{0, \dots, n_{Zellen} - 1\}$ ein Index auf eine Punktmenge von Messungen in Zelle i , so ergibt sich die Bestimmung der Zellen, deren Messungen als Bodenpunkte markiert werden sollen als

$$m_{Zelle_i} = \begin{cases} \text{Boden,} & \text{falls } d_{z_i} < d_{mz} \\ \text{keinBoden,} & \text{sonst} \end{cases} \quad (3.1)$$

Werte für d_{mz} liegen je nach Zellgröße und Anwendung im Bereich zwischen 0.05 und 0.3m. Je größer die Zellen, desto wahrscheinlicher ist ein größeres d_z , da Objekt- und Bodenmessungen zusammen in eine Zelle fallen. Die Zellgröße selbst hat im späteren Verlauf wiederum Auswirkungen auf das Segmentierungsergebnis. Empirisch ermittelte Zellgrößen für gute Ergebnisse mit dem Velodyne HDL-64E liegen bei der hier vorliegenden Arbeit im Bereich zwischen 0.2 und 0.4m.

Vor allem bei Messungen in größerer Distanz lässt die Punktdichte nach. Dies führt dazu, dass Zellen zum Teil nur eine Messung beinhalten. Da bei der Segmentierung Punktmengen mit einem Punkt nicht beachtet werden, empfiehlt es sich, auch aus Gründen der Laufzeit, diese Messungen als Bodenpunkte zu markieren.

In Abbildung 3.8 sind die Bodenpunkte in schwarz dargestellt. Die übrigen Punkte sind orange markiert. Es handelt sich dabei um eine beispielhafte Szene eines Parkplatzes, die bereits in '3.1 Punktbasierte Ansätze' beschrieben wurde.

Abbildung 3.7 zeigt eine Limitierung des vorgestellten Algorithmus zur Bestimmung von Bodenpunkten. Da die Methode nur Messungen lokal pro Zelle betrachtet, kann eine Struktur, die nicht flach ist, jedoch innerhalb einer Zelle in dem vorgegebenen Band von d_{mz} bleibt, als flach interpretiert werden. Ebenfalls werden Strukturen, die zunächst einen Anstieg aufweisen und danach in eine flache Struktur übergehen, als Boden angenommen. Dies kann dazu führen, dass Fahrzeugdächer als Bodenpunkte aufgefasst werden.

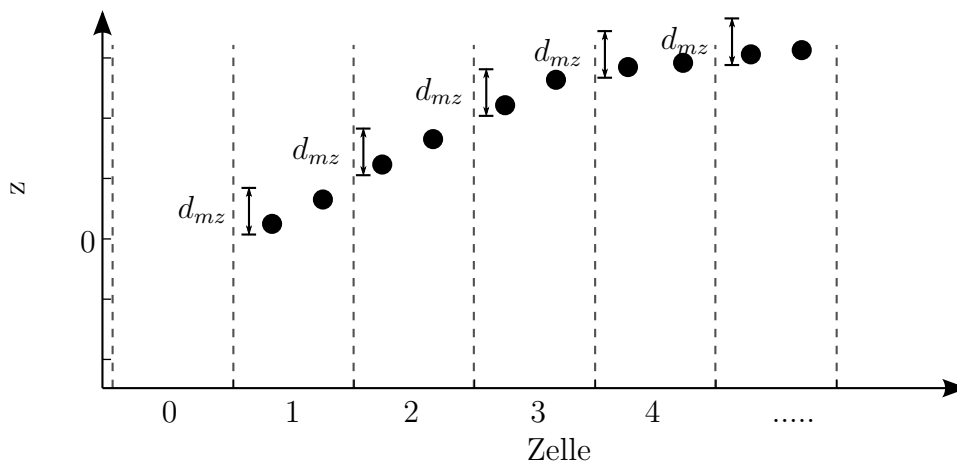


Abbildung 3.7: Limitierung der Bodenpunktbestimmung: Nicht flache Struktur wird als Boden erkannt

Erweitern lässt sich der Parameter d_{mz} indem eine Abhängigkeit zwischen der aktuell zu bearbeitenden Zelle und der Distanz zum Sensor eingeführt wird. Dabei kann sich der Parameter d_{mz} entsprechend einer beliebigen Funktion verändern. So lässt sich beispielsweise d_{mz} mit steigender Distanz vom Sensor vergrößern und es werden mehr Messungen

als Bodenpunkte interpretiert. Hierdurch können bereits mit Rauschen behaftete Objekte herausgefiltert werden.

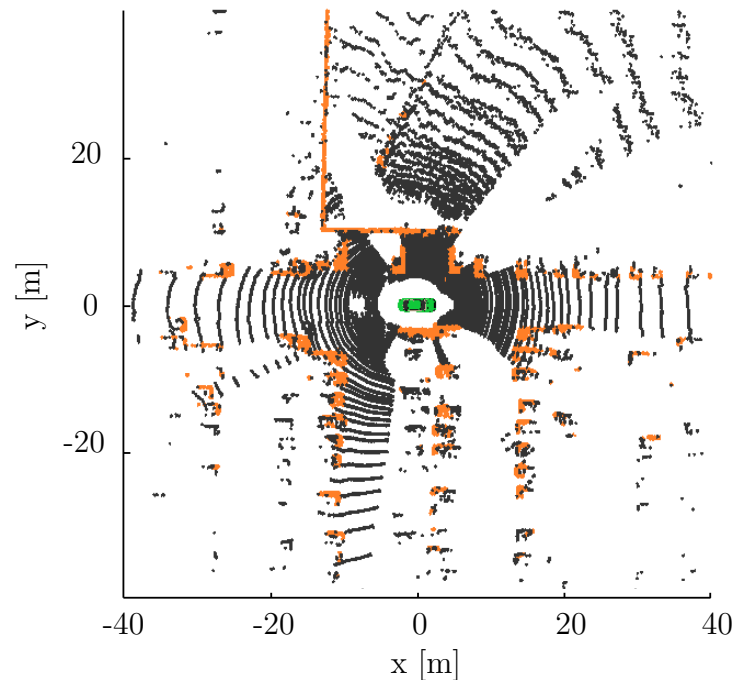


Abbildung 3.8: Bestimmung von Bodenpunkten über die z -Wert-Differenz in den Gitterzellen (Bodenpunkte in schwarz, andere Punkte in orange)

3.3.1.2 Erzeugung von Segmenten

Nach der Entfernung der Bodenpunkte befinden sich nur noch die Zuordnungen von Punkten im zweidimensionalen Gitter, die eine erhabene Struktur entlang der z -Achse aufweisen. Wird jede Zelle als Pixel aufgefasst, kann das zweidimensionale Gitter als Bild interpretiert werden. Jede Zelle, die eine Punktzuordnung erhalten hat, wird im Bild ein schwarzes Pixel (Wert 1). Alle anderen Zellen, die keine Zuordnung erhalten haben, werden als weißes Pixel (Wert 0) gespeichert. Auf das entstandene Schwarz-/Weiß-Bild lässt sich ein Algorithmus anwenden, der zusammenhängende Pixel sucht und sie einem Segment zuweist. Die Interpretation des Gitters als Bild und die Suche nach zusammenhängenden Strukturen findet sich auch bei Michael Himmelsbach et al. [44].

Für die Suche nach zusammenhängenden Strukturen können verschiedene Algorithmen eingesetzt werden. Die Wahl des Algorithmus hängt beispielsweise von der Größe des Bildes ab. Passt das Bild komplett in den Speicher, fällt die Wahl eher auf einen rekursiven Algorithmus. Ist der Speicher zu gering, muss das Bild stückweise bearbeitet werden. Da es sich hier eher um kleine Bilder handelt, kann ein rekursiver Ansatz gewählt werden. Die folgende Beschreibung lehnt sich an die von Linda Shapiro [83] an.

Die Pixel des Bildes werden zunächst erweitert, sodass sie beliebige ganze Zahlen aus der Menge \mathbb{Z} annehmen können. Im ersten Schritt werden alle Pixel, die den Wert 1 tragen auf -1 gesetzt. Da für die Markierung der Segmente positive ganze Zahlen verwendet werden, ermöglicht eine Umkehrung des Vorzeichens die Unterscheidung von bereits zugeordneten und von nicht zugeordneten Pixeln. Begonnen wird in der linken oberen Ecke des Bildes. Das Bild wird zeilenweise durchlaufen, bis das erste Pixel mit dem Wert -1 erreicht wird. Diesem wird daraufhin der Wert 1 zugewiesen. Dies entspricht dem ersten Segment. Ausgehend von diesem Pixel werden alle Nachbarn in einer Vierer- oder Achter-Nachbarschaft gesucht, die den Wert -1 tragen. Alle gefundenen Pixel werden ebenfalls auf den Wert 1 gesetzt. Die Suche nach neuen Nachbarn wird daraufhin auf die neu hinzugefügten Pixel erweitert. Die Suche erfolgt demnach rekursiv. Der Vorgang wird so lange wiederholt, bis keine neuen Nachbarn gefunden werden. Anschließend wird das Bild weiter zeilenweise durchlaufen bis das nächste Pixel mit dem Wert -1 erreicht wird. Diesem wird der Wert 2 zugewiesen und es beginnt die rekursive Suche nach Nachbarn erneut. Die Suche nach verbundenen Strukturen ist abgeschlossen, wenn allen Pixeln mit dem Wert -1 ein neuer Wert zugewiesen wurde.

Abbildung 3.9 zeigt die selbe Szene mit den sechs Fahrzeugen, die auch für die Auswertung des K-Means und des Mean-Shift-Algorithmus verwendet wurde. Die Farben der Punkte visualisieren die Zuordnung zu einem Segment. Die Bodenpunkte wurden in diesem Fall jedoch nicht über die Annahme einer Ebene, sondern über den zuvor beschriebenen Ansatz basierend auf dem Gitter, entfernt. Das Ergebnis zeigt eine Segmentierung der sechs Fahrzeuge in sieben Segmente.

Durch die Wahl der Zellgröße wird festgelegt, ob zusammenhängende Strukturen entstehen, oder nicht. Wird eine zu kleine Zellgröße gewählt, finden sich keine besetzten Nachbarzellen, die mit der aktuellen besetzten Zelle verbunden werden können. Somit entstehen viele einzelne Segmente. Wird eine zu große Zellgröße gewählt, werden hingegen zu viele besetzte Nachbarzellen gefunden und Objekte verschmelzen zu großen Segmenten.

In Abbildung 3.9 ist die Aufteilung des roten Segments in ein weiteres kleines graues Segment zu erkennen. In diesem Fall konnte keine Verbindung der beiden Segmente gefunden werden, was zur Aufspaltung geführt hat.

Die Kombination mit der Bodenpunkterkennung kann aufgrund der Diskretisierung zu Bereichen führen, die sowohl Objektpunkte als auch Bodenpunkte enthalten, was wiederum zu einer nicht exakten Segmentierung führt. Dieser Effekt der Übersegmentierung lässt sich beispielsweise am dunkelgrünen Segment in Richtung der positiven y-Achse in Abbildung 3.9 erkennen.

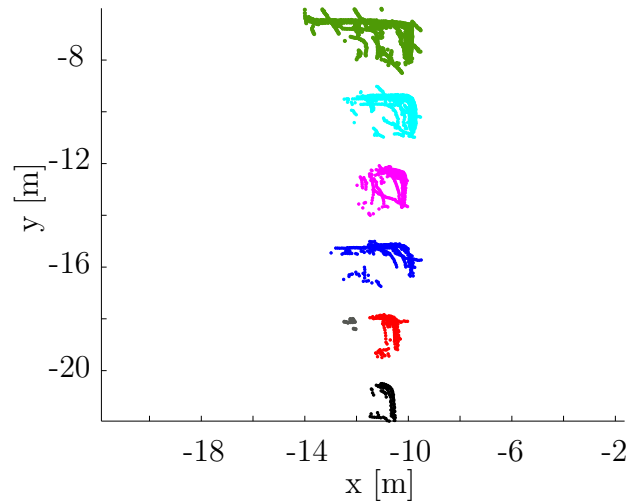


Abbildung 3.9: Segmentierung von sechs parkenden Fahrzeugen mit dem zweidimensionalen gitterbasierten Ansatz

Vorteile des zweidimensionalen gitterbasierten Ansatzes sind die einfache Implementierbarkeit und die Geschwindigkeit bei der Ausführung.

3.3.2 3D Ansatz

Für den zweidimensionalen Fall eines Gitters wurden bereits die beiden Möglichkeiten zur Darstellung als kartesisches Gitter oder als Polargitter angesprochen. Im dreidimensionalen Fall können diese beiden Fälle ebenfalls betrachtet werden. Prinzipiell nähert das Polargitter die physikalischen Eigenschaften der ausgesandten Laserstrahlen besser an. Je weiter ein Objekt vom Sensor entfernt ist, desto weniger Messungen sind auf ihm zu erwarten. Daher wird im Polargitter auch die entsprechende Zelle in ihrer azimuthalen Ausdehnung breiter.

Grund für die Repräsentation der Messungen in einem dreidimensionalen Polargitter ist wiederum die Reduktion der Datenmenge. Einen weiteren Vorteil, der in einem späteren Schritt genutzt wird, stellen die Sortierung und die Möglichkeiten des Zugriffs auf die Daten dar.

3.3.2.1 Aufbereitung der Rohdaten

Bei der Erstellung eines dreidimensionalen polaren Gitters entstehen ausgehend vom Koordinatenursprung Formen, die näherungsweise als Pyramiden interpretiert werden können. Diese werden in einzelne Abschnitte unterteilt. Jeder dieser Abschnitte entspricht einer Zelle im dreidimensionalen polaren Gitter. Die Wahl der Zellgröße ist an die Eigenschaften des Sensors angepasst. Es handelt sich hierbei um die Winkelauflösung in Azimut und Elevation. Für die folgenden Anwendungen wird ein Öffnungswinkel von $\nu = 1^\circ$ in beide Richtungen angenommen. Die Länge der Zellen wird in Abhängigkeit der beiden Winkel in Azimut und Elevation gewählt. Da die Dichte der Daten mit größerer

Entfernung vom Sensor abnimmt, soll eine Längen Anpassung dafür sorgen, dass die Länge der Zellen mit steigender Entfernung zunimmt. Randbedingung ist, dass die Länge jeweils der Breite und der Höhe der Zelle entspricht. Da die Messungen des Sensors nur bis zu einer Distanz d_S in das Gitter eingetragen werden sollen, wird die Bestimmung der Abschnitte in Richtung des Koordinatenursprungs ab dieser Distanz abgebrochen. Abbildung 3.10 (a) zeigt ein dreidimensionales polares Gitter. Abbildung 3.10 (b) stellt beispielhaft einen Ausschnitt und die Zellen des polaren Gitters dar. Die Farben zeigen bis zum Erreichen der Distanz d_S Kanten gleicher Längen an.

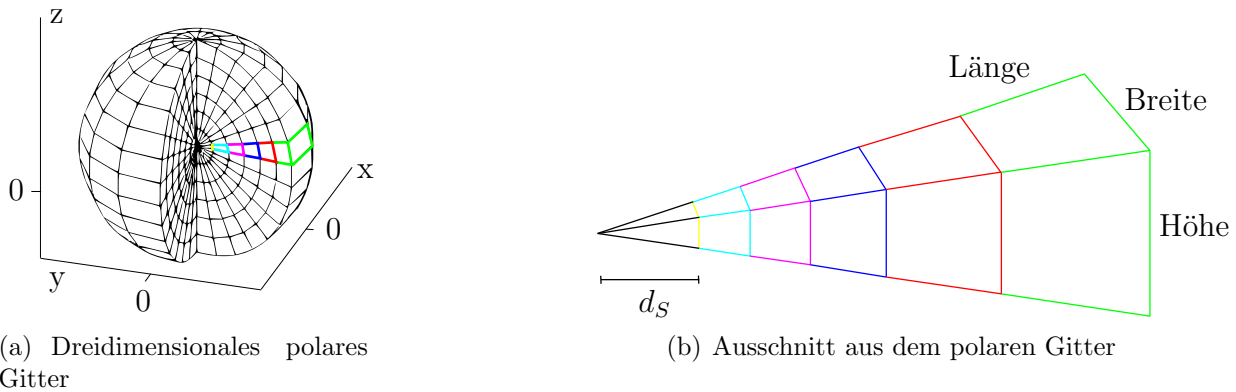


Abbildung 3.10: Dreidimensionales polares Gitter mit einem vergrößerten Ausschnitt

Die Belegung der Zellen wird bestimmt, indem die einzutragenden Datenpunkte in das Gitter gelegt werden und die Datenpunkte jeweils der nächsten Zelle zugeordnet werden. Fallen mehrere Datenpunkte in eine Zelle, werden diese auf einen einzigen Punkt reduziert. Jede belegte Zelle i wird durch einen Punkt P_i repräsentiert. Jedem Punkt P_i kann ein Azimutwert φ_{P_i} und ein Elevationswert θ_{P_i} sowie ein Distanzwert ρ_{P_i} zugeordnet werden. Die Winkel entsprechen dem Mittel aus den beiden Winkeln der jeweiligen Zellkanten. Der Distanzwert ergibt sich aus dem Median der Datenpunkte, die der jeweiligen Zelle zugeordnet wurden.

Im Folgenden wird nicht, wie in den vorherigen Abschnitten, auf die Daten der Parkplatzszene zurückgegriffen. Die Parkplatzszene weist eine weitgehend ebene Fahrbahn auf, wohingegen in der im Weiteren verwendeten Szene auch stärkere Änderungen des Fahrbahnverlaufs bezüglich der Höhe enthalten sind. Abbildung 3.11 zeigt beispielhaft die Rohdaten einer Szene innerhalb einer Ortschaft. Die in ein polares dreidimensionales Gitter eingetragenen Datenpunkte sind in Abbildung 3.12 dargestellt.

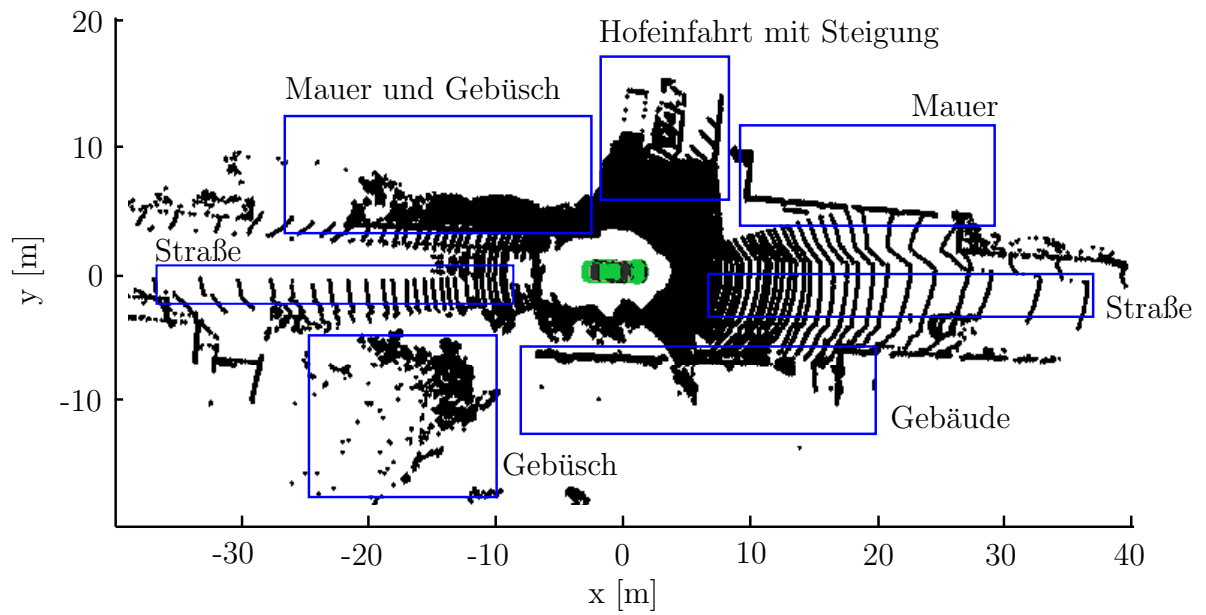


Abbildung 3.11: Rohdaten einer Szene innerhalb einer Ortschaft

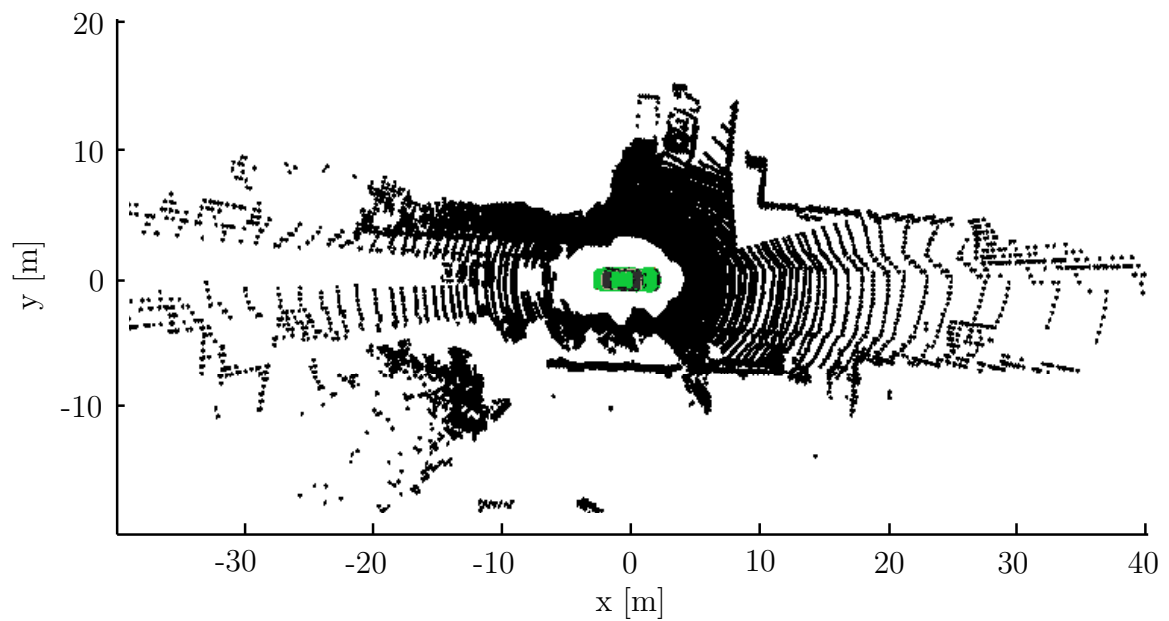


Abbildung 3.12: Diskretisierte Datenpunkte in einem polaren dreidimensionalen Gitter

3.3.2.2 Erzeugung von zusammenhängenden Teilssegmenten und Bestimmung von Bodenpunkten

Vor der Segmentierung der Daten in einzelne Objekte müssen auch hier, wie bereits bei den zuvor genannten Verfahren, die Bodenpunkte detektiert werden. Anna Petrovskaya et al. [70] stellen ein Verfahren vor, das die Eigenschaften des verwendeten Velodyne HDL-64E Lasersensor ausnutzt. Der Lasersensor hat 64 Laserdioden übereinander angeordnet. Das bedeutet, dass die Messungen an einer Winkelposition des Sensors nahezu gleichzeitig an allen 64 Dioden erfolgen. Anna Petrovskaya et al. teilen daher den Bereich um den Sensor in vertikale Bereiche auf, die jeweils die Punktmessungen von einer Winkelposition enthalten. In einem weiteren Schritt werden ausgehend von demjenigen Messpunkt A , der am nächsten beim Sensor liegt, die nächsten Nachbarn B und C dieses Messpunkts innerhalb des Bereichs bestimmt. Diese werden durch Geraden verbunden. Werden die Strecken AB und BC normiert und das Skalarprodukt gebildet, kann dies als Merkmal zur Erkennung von Bodenpunkten genutzt werden. Liegt das Skalarprodukt nahe dem Wert Eins, handelt es sich mit hoher Wahrscheinlichkeit um Bodenpunkte.

Michael Himmelsbach et al. [46] beschreiben ein ähnliches Verfahren wie bei Anna Petrovskaya et al. [70]. Es wird die xy -Ebene durch einen Kreis mit unendlichem Radius dargestellt. Dieser wird in eine feste Anzahl von Segmenten unterteilt. Die Messungen werden daraufhin den jeweiligen Segmenten zugeordnet, welche selbst wiederum in Zellen aufgeteilt sind und denen die Messungen ebenfalls zugeordnet werden. Fallen mehrere Messungen in eine Zelle, wird ein Prototypenpunkt bestimmt, der den niedrigsten z -Wert aus allen Messungen der Zelle aufweist, und diese Zelle repräsentiert. Durch das Einpassen von lokalen Geradenstücken in die Prototypenpunkte der Segmente und unter Einbeziehung von Randbedingungen, wie z.B. die Steigung oder die Höhe der Geraden, wird der Verlauf des Bodens bestimmt. Ein Distanzparameter zwischen letztem eingepasstem Geradenstück und dem nächsten zu verwendenden Punkt sorgt für einen weichen Übergang und somit für einen stufenlosen Verlauf des Bodens.

Frank Moosmann et al. [66] beschreiben ein Verfahren, das auf Basis von lokalen Konvexitätsbedingungen sowohl Bodenpunkte, als auch Objekte segmentiert. Aufgrund der natürlichen Sortierung der Messpunkte, wie sie sich beispielsweise auch durch das Messprinzip des Velodyne HDL-64E ergibt, ist es ohne Aufwand möglich einen Nachbarschaftsgraphen aufzustellen. Zu einem Knoten des Graphen, der ein Messpunkt ist, lässt sich über die vier benachbarten Knoten der Normalenvektor bestimmen. Mit den Normalenvektoren wird eine lokale Konvexität zwischen benachbarten Knoten überprüft. Trifft diese zu und liegen die Knoten in einer definierten Reichweite zueinander, werden sie über ein Region-Growing-Verfahren zu einem Segment zusammengefügt.

Der folgende Ansatz stellt eine Kombination aus den aus der Literatur vorgestellten Algorithmen dar. Dies ermöglicht einerseits die robuste Erkennung von Objekten und andererseits lassen sich kleine Bereiche mit Bodenpunkten finden, die sich hinter Objekten befinden, jedoch mit benachbarten Bodenbereichen verbunden sind.

Mit der Speicherung der Messdaten in einem dreidimensionalen polaren Gitter ist ein schneller Zugriff auf die Zelleninhalte möglich. Zusätzlich lassen sich dadurch Nachbarschaftsbeziehungen leicht herstellen. Zwischen vier benachbarten Punkten im Gitter wird jeweils eine Ebene gespannt aus der sich der zugehörige Normalenvektor der Ebenen ergibt. In Abbildung 3.13 sind die drei Schritte visualisiert. Abbildung 3.13 (a) zeigt die Rohdaten einer Messung von Bodenpunkten und einer Wand. In Abbildung 3.13 (b) sind die Messungen in das dreidimensionale polare Gitter eingetragen und die Punkte in den Zellen dienen als Eckpunkte für die Bestimmung von Ebenen, die durch unterschiedliche Farben dargestellt sind. Die Normalenvektoren der Ebenen sind in Abbildung 3.13 (c) durch rote Geradenstücke repräsentiert.

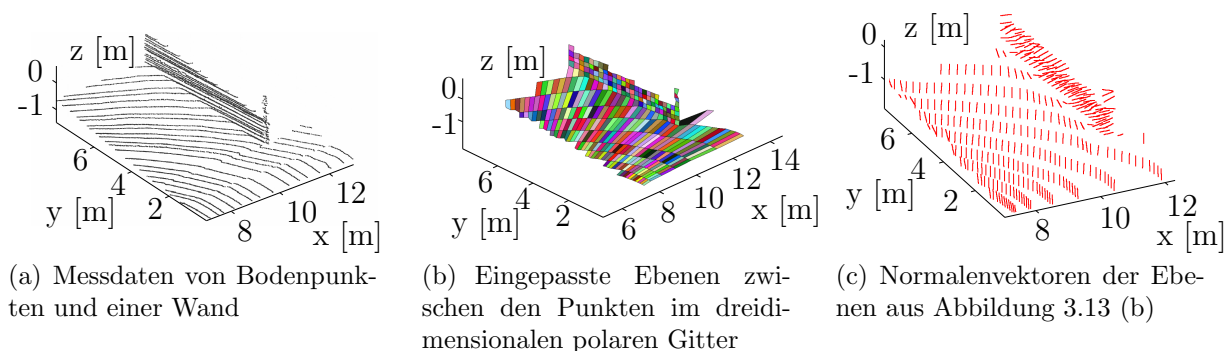


Abbildung 3.13: Bestimmung von Normalenvektoren für diskretisierte Messdaten in einem polaren dreidimensionalen Gitter

Im weiteren Verlauf werden die Ebenen zu Teilsegmenten zusammengefasst. Zwei benachbarte Ebenen E_i und E_j werden demselben Segment S_k zugeordnet, falls das Skalarprodukt ihrer beiden normierten Normalenvektoren $\frac{\vec{n}_i}{|\vec{n}_i|}$ und $\frac{\vec{n}_j}{|\vec{n}_j|}$ einen definierten Schwellwert ξ_B nicht unterschreitet. Falls

$$E_i \subset S_k \quad (3.2)$$

gilt

$$E_j \subset S_k \Leftrightarrow \frac{\vec{n}_i}{|\vec{n}_i|} \cdot \frac{\vec{n}_j}{|\vec{n}_j|} \geq \xi_B \quad (3.3)$$

Begonnen wird mit einer beliebigen Ebene, deren Nachbarn gesucht werden. Können keine weiteren Nachbarn gefunden werden, ist das Teilsegment abgeschlossen und es wird mit einer Ebene, die noch keinem Teilsegment zugeordnet wurde fortgefahren. Dies wird so lange wiederholt, bis alle Ebenen eine Zuordnung erhalten haben. Es empfiehlt sich den Parameter ξ_B nahe dem Wert Eins zu wählen, sodass flache Strukturen zusammengefügt, bei auftretenden Kanten aber abgebrochen wird.

Abbildung 3.14 zeigt die berechneten Teilsegmente in den Messungen. Die Messdaten zeigen eine Szene in einem Wohngebiet. Vor und hinter dem Fahrzeug befindet sich die

Straße. Links vom Fahrzeug beginnt eine Hofeinfahrt, die in eine Garage mündet. Erhabene Objekte sind entweder Hauswände, oder Mauerstücke, die den Garten von der Straße trennen. Alle gefundenen Teilsegmente sind durch Farben hervorgehoben. Die Farben wurden zufällig gewählt und dienen lediglich der Unterscheidung der Teilsegmente.

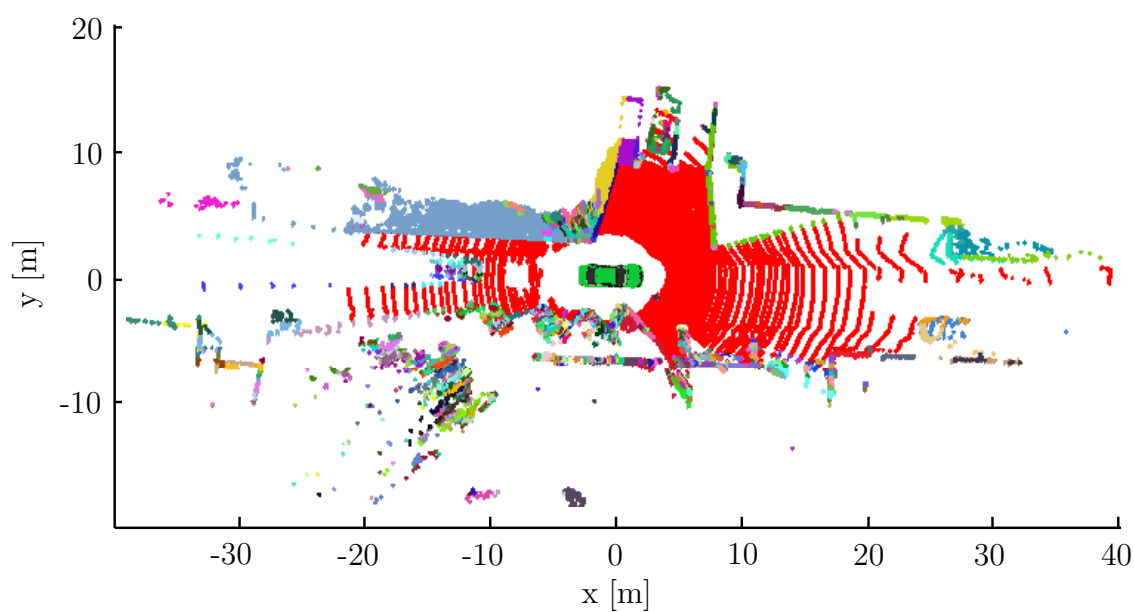


Abbildung 3.14: Teilsegmente, die über benachbarte Ebenen und ihre Normalenvektoren bestimmt wurden

In Abbildung 3.14 zeigt das rote Teilsegment die Bodenpunkte, die es im Folgenden zu bestimmen gilt. Es wird die Annahme getroffen, dass die am nächsten liegenden Messungen vor dem Fahrzeug Bodenpunkte darstellen und daher dem Segment der Bodenpunkte zugeordnet werden können. Befindet sich jedoch direkt vor dem Sensor bereits das erste Hindernis, würde dieses, aufgrund der Annahme, ebenfalls als Boden erkannt werden.

Die Struktur des dreidimensionalen polaren Gitters erlaubt die Punkte der Zellen mit gleichem azimuthalen Winkel zu bestimmen. Die Punkte werden im Folgenden mit P_1 bis P_l bezeichnet und weisen eine aufsteigende Sortierung der Elevationswinkel auf. Ein Bereich aus dem die Punkte jeweils bestimmt werden, ist beispielhaft in Abbildung 3.15 mit rot hervorgehoben.

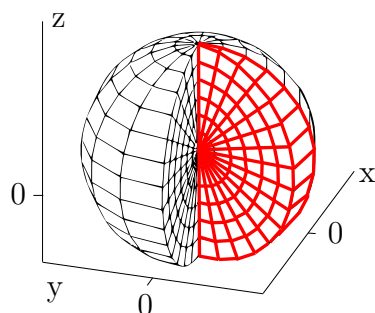


Abbildung 3.15: Bereich mit Punkten mit gleichem Azimutwinkel

Der Punkt P_1 ist der Punkt mit dem kleinsten Elevationswinkel und wurde im vorherigen Schritt bereits einem Teilsegment zugewiesen. Er stellt möglicherweise ein Bodenpunkt dar. Um dies zu überprüfen, wird ein Pseudopunkt P_{ps} eingeführt. Die Position des Pseudopunktes berechnet sich aus der Position des Punktes P_1 , wobei die Höhenkomponente z_1 in kartesischen Koordinaten als gleich angenommen wird und gelten soll:

$$z_1 < 0 \quad (3.4)$$

Die Forderung, dass z_1 kleiner als Null sein muss, kann gestellt werden, da sich der Sensor auf dem Dach eines Fahrzeugs befindet und die Berechnung im Sensorkoordinatensystem erfolgt. Sie verhindert, dass in Gleichung 3.6 θ_{ps} Null wird und in Gleichung 3.7 eine Division durch Null erfolgt.

Wird $z_1 < 0$ nicht erfüllt, wird ein Hindernis angenommen.

Es gilt für die Bestimmung des Pseudopunktes P_{ps} aus P_1

$$P_1(\rho_1, \varphi_1, \theta_1) \Rightarrow P_{ps}(\rho_{ps}, \varphi_{ps}, \theta_{ps}) \quad (3.5)$$

mit

$$\theta_{ps} = \theta_1 - \nu \quad (3.6)$$

$$\rho_{ps} = \frac{z_1}{\sin(\theta_{ps})} \quad (3.7)$$

$$\varphi_{ps} = \varphi_1 \quad (3.8)$$

Die Variable ν (Öffnungswinkel des dreidimensionalen polaren Gitters) verändert den Elevationswinkel derart, dass der Pseudopunkt einen Winkelschritt tiefer liegt als der Punkt P_1 .

Der nachfolgende Schritt berechnet den Anstieg zwischen den drei aufeinanderfolgenden Punkten P_{ps} , P_1 und P_2 . Ist das Skalarprodukt der normierten Normalenvektoren auf den Verbindungsstrecken zwischen den drei Punkten größer oder gleich dem Schwellwert ξ_B , wird angenommen, dass es sich bei den Punkten um Bodenpunkte handelt. Das Segment, dem der Punkt P_1 zugeordnet wurde, kann somit als Bodensegment markiert

werden. Abbildung 3.16 zeigt schematisch die beiden möglichen Fälle. Der beschriebene Schritt wird für alle Winkelsegmente in azimuthaler Richtung durchgeführt.

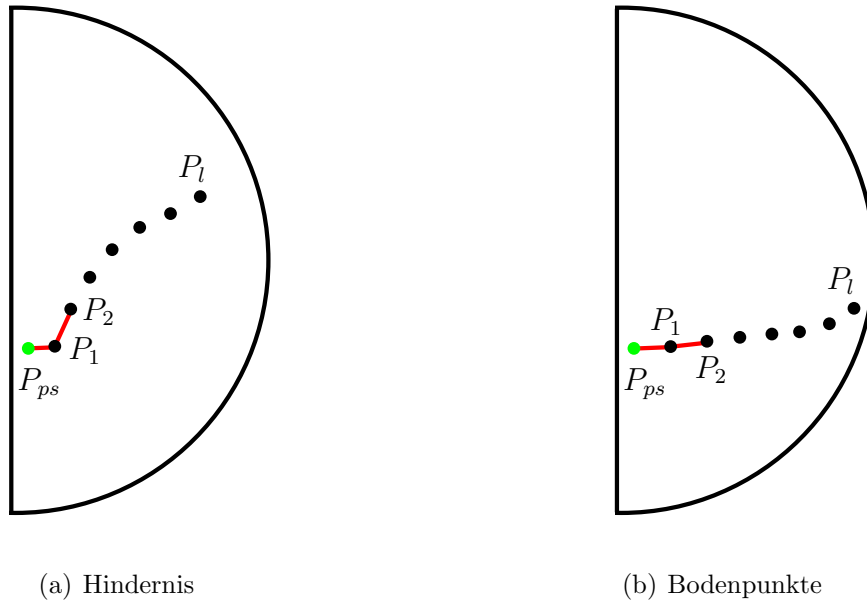


Abbildung 3.16: Pseudopunkt zur Unterscheidung zwischen Hindernissen und Bodenpunkten

Abbildung 3.17 zeigt das erkannte Bodensegment. Das Bodensegment ist durch die Farbe schwarz visualisiert, die übrigen Objekte sind in orange dargestellt. Die Messungen zeigen, dass die Fahrbahn unterschiedliche Höhen aufweist und daher eine Annahme der Fahrbahn als Ebene nicht möglich ist. Die Höhen sind an vier Stellen in der Grafik mit den Nummern 1 bis 4 gekennzeichnet und betragen in Sensorkoordinaten:

Bereich	Höhe in Sensorkoordinaten
1	-1.6 m
2	-0.7 m
3	-1.3 m
4	-1.7 m

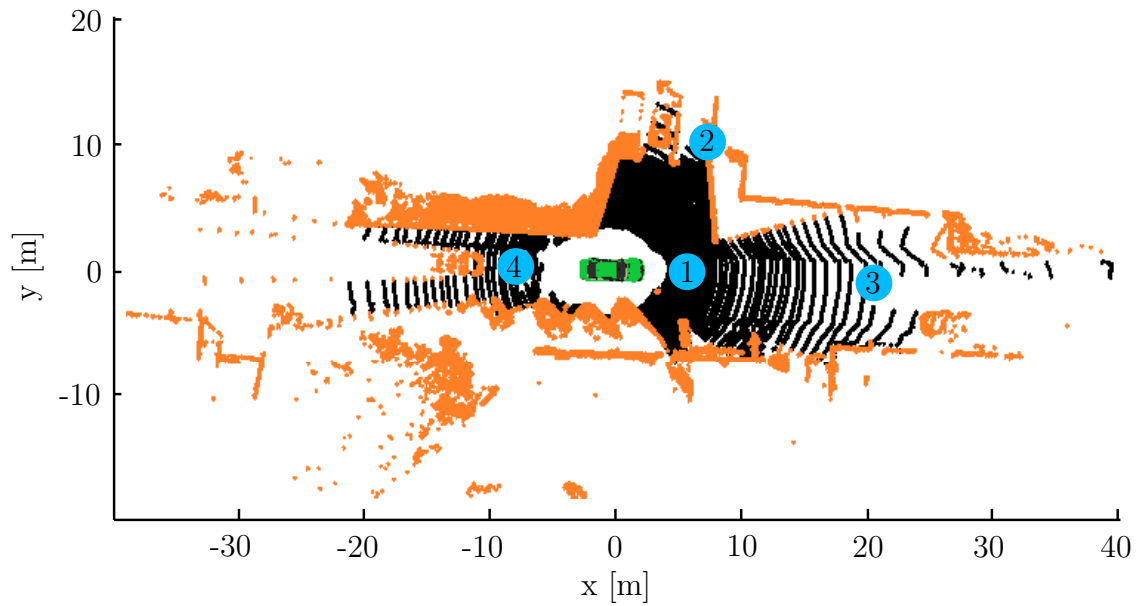


Abbildung 3.17: Erkennung des Bodensegments

Abbildung 3.18 zeigt einen vergrößerten Ausschnitt aus Abbildung 3.17. Der Ausschnitt zeigt einen Teil der Hofeinfahrt mit den sich anschließenden Garagen. In der linken Garage steht ein Fahrzeug, die rechte Garage ist leer. Selbst die Bodenpunkte unter dem Fahrzeug werden als solche erkannt.

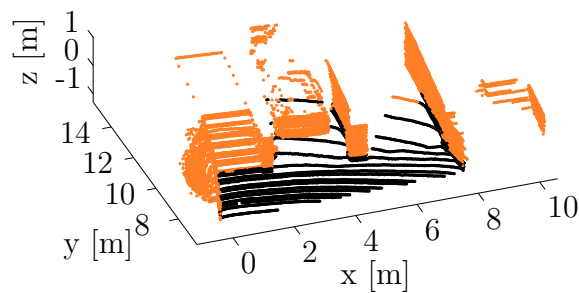


Abbildung 3.18: Vergrößerter Ausschnitt aus Abbildung 3.17

Nach der Erkennung der Bodenpunkte gilt es die übrigen Punkte zu Segmenten zusammenzufassen. Das Vorgehen wird im folgenden Abschnitt erläutert.

3.3.2.3 Erzeugung von Segmenten

In Abbildung 3.14 sind alle gefundenen Teilsegmente durch unterschiedliche Farben dargestellt. Es lässt sich erkennen, dass zusammengehörige Punktmengen aus einzelnen Teilsegmenten bestehen. Diese gilt es im Folgenden zu Segmenten zusammenzufassen.

Da die Daten, anders als bei Michael Himmelsbach et al. [46], bereits in einem dreidimensionalen polaren Gitter angeordnet sind, können die Segmente durch die rekursive Suche nach benachbarten besetzten Zellen erzeugt werden. Es handelt sich dabei um den auf drei Dimensionen erweiterten Algorithmus aus Abschnitt '3.3.1.2 Erzeugung von Segmenten'. Für die Ausgabe der Segmente werden die originalen Messdaten genutzt, die in die zusammengefassten Zellen eingetragen wurden.

Eine optionale Filterung kann anschließend dafür sorgen, dass zu kleine Segmente gelöscht werden. Dies erfolgt über folgende Parameter, gegen die die Segmente geprüft werden:

- Länge der Punktmenge
- Breite der Punktmenge
- Höhe der Punktmenge
- Anzahl der Punkte in der Punktmenge

Abbildung 3.19 zeigt die segmentierte Punktmenge aus Abbildung 3.11. Das parkende Fahrzeug in der Garage sowie die am Straßenrand rechts neben dem Messfahrzeug abgestellten Fahrzeuge wurden als eigenständige Segmente erkannt. Die Farben der Segmente wurden zufällig gewählt und dienen lediglich der Unterscheidung einzelner Segmente.

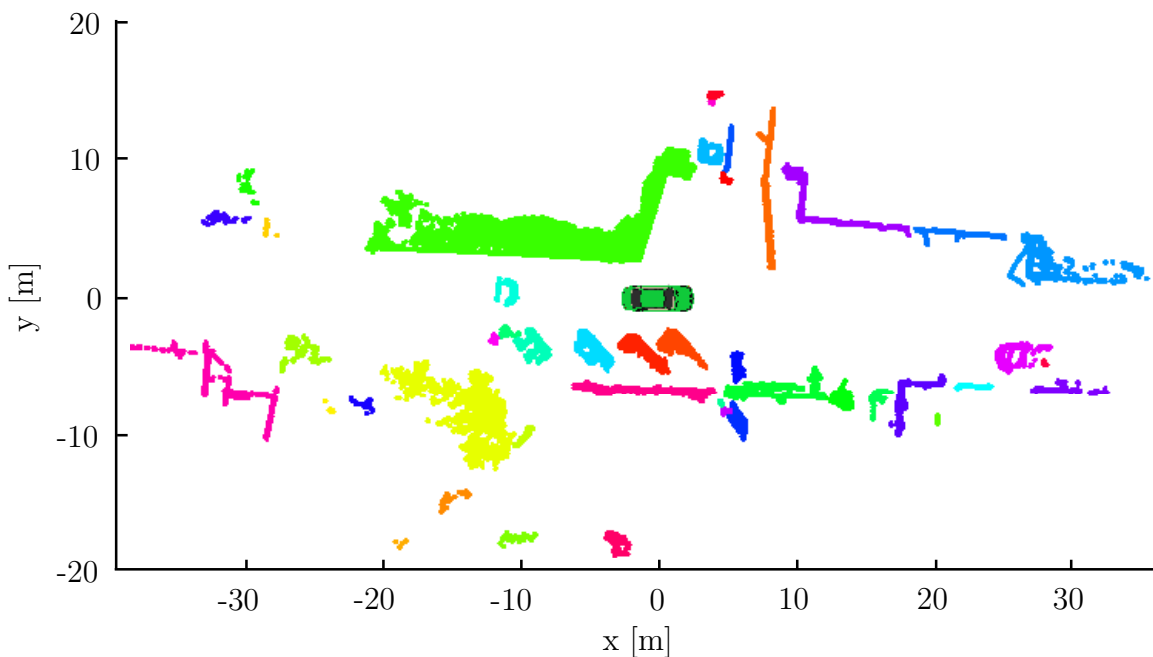


Abbildung 3.19: Segmentierte Punktmenge aus Abbildung 3.11

3.4 Klassifikation

Die Segmentierung stellt ein Element in der gesamten Verarbeitungskette dar. Nach der Segmentierung ist es erforderlich die Segmente auszuwählen, die in einem weiteren Verarbeitungsschritt verwendet werden sollen. Dieses Vorgehen wird als Klassifikation bezeichnet. Dabei wird den Segmenten eine Klassenzugehörigkeit, wie beispielsweise PKW, LKW, Fußgänger, etc., zugewiesen, um die anschließende Verarbeitung anzupassen. Für die Klassifikation finden sich in der Literatur verschiedenste Ansätze.

Michael Himmelsbach et al. [43] verwenden eine Support Vector Machine für die Klassifikation. Dabei beschreiben sie verschiedene Merkmale, basierend auf der Punktwolke der Messungen, die sich für ihre Methode eignen. Eine Beschreibung von Support Vektor Machines findet sich bei Christopher Burges [8].

Einen einfachen Klassifikator stellt der naive Bayes-Klassifikator dar. Er findet bei der Auswahl der gefundenen Segmente in dieser Arbeit Anwendung. Der Bayes-Klassifikator weist einem Segment aufgrund seiner Merkmale eine Wahrscheinlichkeit zur Klassenzugehörigkeit zu. Merkmale können beispielsweise die Länge, Breite, das Volumen oder die Anzahl der Punkte sein. Der naive Bayes-Klassifikator geht dabei davon aus, dass die Merkmale jeweils voneinander unabhängig sind. Dies stellt eine Vereinfachung dar, führt jedoch zu einem einfachen und oft ausreichenden Algorithmus. Eine kurze Einführung findet sich unter anderem bei Ben Coppin [13].

3.5 Zusammenfassung

In diesem Kapitel wurden verschiedene Ansätze zur Segmentierung von Daten von 3D-Lasersensoren vorgestellt. Dabei wurde zunächst auf punktbasierte Algorithmen eingegangen. Hierauf folgte eine Beschreibung zur Interpretation der Messdaten als Bild, worauf sich aus der Bildverarbeitung bekannte Algorithmen zur Segmentierung anwenden lassen. Der dritte Teil widmete sich den gitterbasierten Ansätzen, die sowohl in zwei, als auch drei Dimensionen betrachtet wurden. Für die punkt- und gitterbasierten Ansätze wurden jeweils Möglichkeiten zur Erkennung von Bodenpunkten beschrieben, welche für die folgenden Schritte der Segmentierung notwendig sind.

Der K-Means-Algorithmus bei den punktbasierten Ansätzen ist ein einfach zu implementierender und zu nutzender Ansatz zur Segmentierung. Er erweist sich allerdings aufgrund der vorzuziehenden Anzahl an zu suchenden Segmenten in diesem Fall als nicht einsetzbar.

Bei denjenigen punktbasierten Ansätzen (Mean-Shift, DBSCAN), die auf die Bestimmung von benachbarten Punkten angewiesen sind, ergibt sich schnell eine Grenze bei der Verarbeitung in Echtzeit. Auch die effiziente Speicherung in einer Baumstruktur kann dies ab einer gewissen Menge an Punkten nicht vermeiden.

Der vorgestellte bildbasierte Ansatz zeigt einerseits eine weitere Möglichkeit zur Interpretation der Messdaten und andererseits die Anwendung von Bildverarbeitungsalgorithmen

3. Segmentierung

auf diesen Daten.

Die gitterbasierten Ansätze in zwei Dimensionen ermöglichen die echtzeitfähige Segmentierung einer gesamten Messung. Hier wird jedoch die vorhandene z-Komponente in den Messungen nicht beachtet.

Der vorgestellte dreidimensionale gitterbasierte Ansatz bietet den Vorteil, dass alle Schritte, auch die Bestimmung der Bodenpunkte, direkt im Gitter vollzogen werden können. Bei der Segmentierung wird hier, im Vergleich zum zweidimensionalen Ansatz, die z-Komponente der Messungen beachtet. Die aktuelle Implementierung ermöglicht jedoch keine Verarbeitung in Echtzeit.

Tabelle 3.1 fasst die Ergebnisse tabellarisch zusammen. Die Bewertung gliedert sich in sehr gut (++), gut (+), neutral (0), schlecht (-) und sehr schlecht (- -).

Alle vorgestellten Ansätze, bis auf den K-Means-Algorithmus, zeigen Ergebnisse, die sich für den hier vorgestellten Anwendungsbereich eignen. Im Verlauf der Arbeit wird aufgrund der Echtzeitfähigkeit die Segmentierung mit dem zweidimensionalen gitterbasierten Ansatz verwendet.

Ansatz	Bodenpunkte	Segmentierung	Bewertung	Bemerkung
Punktbasiert	Ebene	K-Means	- -	Segmentanzahl muss bekannt sein
Punktbasiert	Ebene	Mean-Shift	+	Gute Ergebnisse, Rechenzeit hoch
Punktbasiert	Ebene	DBSCAN	+	Gute Ergebnisse, Rechenzeit sehr hoch
Bildbasiert	Ebene	Mean-Shift	0	Vergleich nur über Veröffentlichung (Bharath Kalyan et. al [54])
Gitterbasiert 2D	Ebene / Höhe in Zellen	Nachbarschaftsbestimmung 2D	++	Gute Ergebnisse, geringe Rechenzeit
Gitterbasiert 3D	Anstieg der Zellen ausgehend vom Ursprung	Nachbarschaftsbestimmung 3D	+	Gute Ergebnisse, sehr hohe Rechenzeit

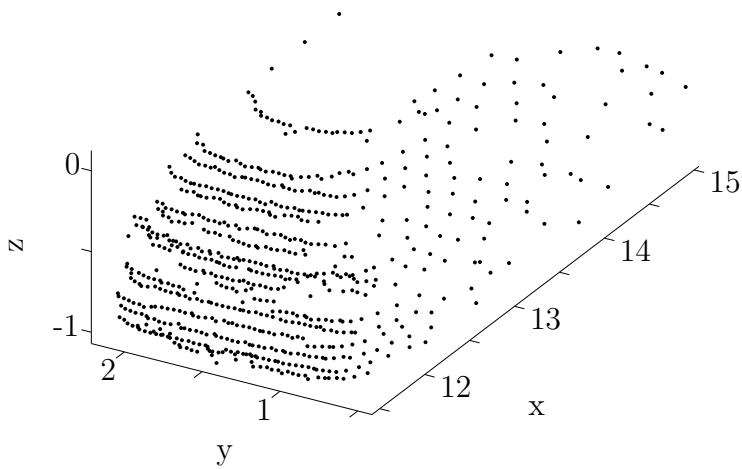
Tabelle 3.1: Vergleich der vorgestellten Ansätze zur Segmentierung

Außenkonturbestimmung

Sensoren, wie beispielsweise ein Lasersensor, liefern 3D Punktmessungen ihrer Umgebung. Die 3D Punktmessungen werden daraufhin durch die Segmentierung (Kapitel '3 Segmentierung') in einzelne Punktmengen geteilt. Nach dem Schritt der Segmentierung liegen die 3D Punktmessungen zwar in einer geordneten Struktur vor, sodass bekannt ist, welche Punkte zusammengehören und eine Punktmenge bilden, jedoch lassen sich in dieser Form noch keine direkten Informationen, wie z.B. die Länge und die Breite, aus der Punktmenge ablesen. Die Forderung an den sich anschließenden Schritt ist eine Überführung der Punktmengen in eine Objektrepräsentation. Eine Objektrepräsentation kann im einfachsten Fall der Schwerpunkt einer Punktmenge, oder eine umgebende Box der Punktmenge sein. Die im Folgenden vorgestellte Objektrepräsentation beschreibt das Objekt durch seine Außenkontur. Der Fokus dabei liegt auf der Beschreibung von Fahrzeugaußenkonturen. Die Anpassung hierauf erfolgt durch Eigenschaften, die in '4.5 Flächeneigenschaften' beschrieben werden. Das Verfahren ist jedoch allgemein formuliert und lässt sich durch Veränderung der Eigenschaften auf andere Außenkonturen übertragen.

Die Außenkonturbestimmung von Fahrzeugen erfolgt, wie eingangs erwähnt, auf segmentierten 3D Punktmessungen im euklidischen Raum. Dabei können die Punkte in einer ungeordneten Menge $P = \{P_1, \dots, P_{N_p}\}$ vorliegen. N_p ist dabei die Anzahl der Punkte und $P_j = (x_j, y_j, z_j)^T$ repräsentiert einen Punkt in der Punktmenge. Es gilt $j \in N_s$ und $N_s = \{1, \dots, N_p\}$. Der Schwerpunkt von P ist gegeben als $\vec{c}_P \in \mathbb{R}^3$. Berechnet wird er über den Mittelwert über die Dimensionen x , y und z .

Im Verlauf des Kapitels werden Zwischenschritte des Verfahrens zur Außenkonturbestimmung mit Hilfe einer Beispielmessung erläutert. Es handelt sich um eine Messung mit dem Velodyne HDL-64E, der in Kapitel '2.2.1 Lasersensor' beschrieben wurde. Abbildung 4.1 zeigt die 3D Punktmessungen und einen Ausschnitt des zugehörigen Kamerabildes der Dokumentationskamera.



(a) 3D Punktmessungen



(b) Ausschnitt aus der Dokumentationskamera

Abbildung 4.1: Beispielmessung für die Erläuterung der Zwischenschritte. Die Distanzangaben sind in der Einheit Meter angegeben.

4.1 Stand der Technik

Obwohl bei der Aufzeichnung von Messdaten Sensoren zum Einsatz kommen, die Messungen in x -, y - und z -Koordinaten aufnehmen, finden dennoch Projektionsansätze Anwendung. Dabei werden die Messungen entlang der z -Achse nicht als solche genutzt, sondern in die xy -Ebene projiziert. Dabei geht eine Dimension der Messung verloren, wobei die Dichte der Messungen erhalten bleibt. Dieser Ansatz findet sich beispielsweise in den Arbeiten von Takashi Ogawa et al. [68] und bei S. Sato et al. [79] jeweils bei der Verfolgung von Fußgängern. Auch bei der Verfolgung von Fahrzeugen wird der Projektionsansatz von Christian Stimming et al. [85] eingesetzt. Speziell bei Sensoren wie dem Velodyne HDL-64E, der aufgrund seiner Anzahl an vertikal angeordneten Laserdioden eine detaillierte Messung entlang der z -Achse liefert, ist es je nach Anwendung nicht mehr sinnvoll die Messungen entlang der z -Achse zu vernachlässigen. Bei Fahrzeugen, die schräge Strukturen aufweisen, wie dies beispielsweise am Heck von Limousinen auftritt, führt die Projektion der Messungen in die xy -Ebene zu Verschmierungen und keinen klaren Kanten.

Zur Objektrepräsentation von Punktwolken aus der Segmentierung werden meist Modelle wie Boxen eingesetzt. Dieser Ansatz findet sich bei den Arbeiten von Michael Himmelsbach et al. [44] und Sylvia Pietzsch et al. [71]. Bei Anwendungen, die präzise Objektformen benötigen, wie dies bereits in Kapitel '1.1 Hochgenaue Formrepräsentation von Objekten' beschrieben wurde, reichen einfache Formen zur Beschreibung nicht mehr aus.

Je nach Dichte der Punktmengen lassen sich definierte Formen wie Ebenen, Kugeln, Zylinder und Kegel finden. Ruwen Schnabel et al. [82] stellen einen Algorithmus vor, der in einer ungeordneten Punktmenge Formen über einen RANSAC-Algorithmus (Random Sample Consensus) sucht. Der RANSAC-Algorithmus wurde von Martin Fischler et al. [29] beschrieben und sorgt für ein robustes Auffinden von definierten Modellen. Die Punktmengen sollten allerdings entsprechend dicht sein. In der Arbeit wird auf Punktmengen mit mehreren tausend Punkten zurückgegriffen. Sung Joon Ahn et al. [1] stellen ebenfalls ein Verfahren vor, das es erlaubt verschiedene Formen in Punktmengen einzupassen. Dabei wird der quadratische Fehler minimiert, der sich aus dem orthogonalen Abstand zwischen Modell und Punktmenge ergibt. Das Einpassen von Ebenen in Punktmengen findet häufig bei aus der Luft aufgenommenen Messdaten Anwendung. Dabei werden Dächer und Wände von Häusern rekonstruiert, um eine dreidimensionale Beschreibung zu erhalten. Beispiele hierzu finden sich bei Qihong Zeng et al. [95]. Aparajithan Sampath et al. [77] nutzen Normalenvektoren die sich für jeden Messpunkt aus seinen benachbarten Messpunkten ergeben. Benachbarte Messpunkte, die ähnliche Normalenvektoren aufweisen, ergeben zusammen eine Ebene, die Dächer von Häusern beschreiben.

Carmen Witte et al. [94] nutzen Lasermessungen um automatisch 3D-Modelle von Fahrzeugen zu erstellen. Die 3D-Modelle werden direkt durch eine Flächenapproximation der Messpunkte mit Hilfe der Delaunay-Triangulation erstellt.

Daniel Meissner und Klaus Dietmayer [64] verwenden Active Shape Models um verschiedene Fahrzeugtypen in Messdaten von Lasersensoren an Kreuzungen zu erkennen. Die Bestimmung der Position und Orientierung der Fahrzeuge erfolgt in einem weiteren Schritt über einen Registrierungsalgorithmus, der die bestimmte Modellform auf die Messdaten registriert.

4.2 Gitterfunktion

Für die Berechnung der Außenkontur wird eine diskrete Gitterfunktion verwendet. Sie findet sowohl im Vorverarbeitungsschritt ('4.3 Vorverarbeitungsschritt') als auch bei der Definition der Flächenfunktion ('4.4 Flächenfunktion') Anwendung. Sie wird in der xz -Ebene wie folgt definiert

$$\begin{aligned}
 V : \Gamma &\rightarrow \mathbb{R} \text{ mit} & (4.1) \\
 \Gamma &:= \{(ih_x, kh_z) | i \in N_x, k \in N_z\}, \\
 N_x &:= \{0, 1, \dots, n_x - 1\}, \\
 N_z &:= \{0, 1, \dots, n_z - 1\}
 \end{aligned}$$

Dabei bezeichnen h_x und h_z die äquidistanten Schrittweiten der Gitterfunktion in x - und z -Richtung. n_x und n_z geben die Anzahl der Knotenpunkte entlang der jeweiligen Achse an. Ein Knotenpunkt ist definiert als der Punkt im Gitter, an dem sich die horizontalen und vertikalen Verbindungslinien schneiden. Vier Knotenpunkte ergeben entsprechend der folgenden Definition eine Zelle

$$\begin{aligned} \text{Zelle} := \{V_1, V_2, V_3, V_4\} = & \quad (4.2) \\ & \{V(x_i, z_k), V(x_{i+1}, z_k), \\ & V(x_{i+1}, z_{k+1}), V(x_i, z_{k+1})\} \end{aligned}$$

Dabei gilt $x \in \{0, h_x, \dots, (n_x - 1)h_x\}$ und $z \in \{0, h_z, \dots, (n_z - 1)h_z\}$.

Abbildung 4.2 zeigt ein Beispiel für eine Gitterfunktion.

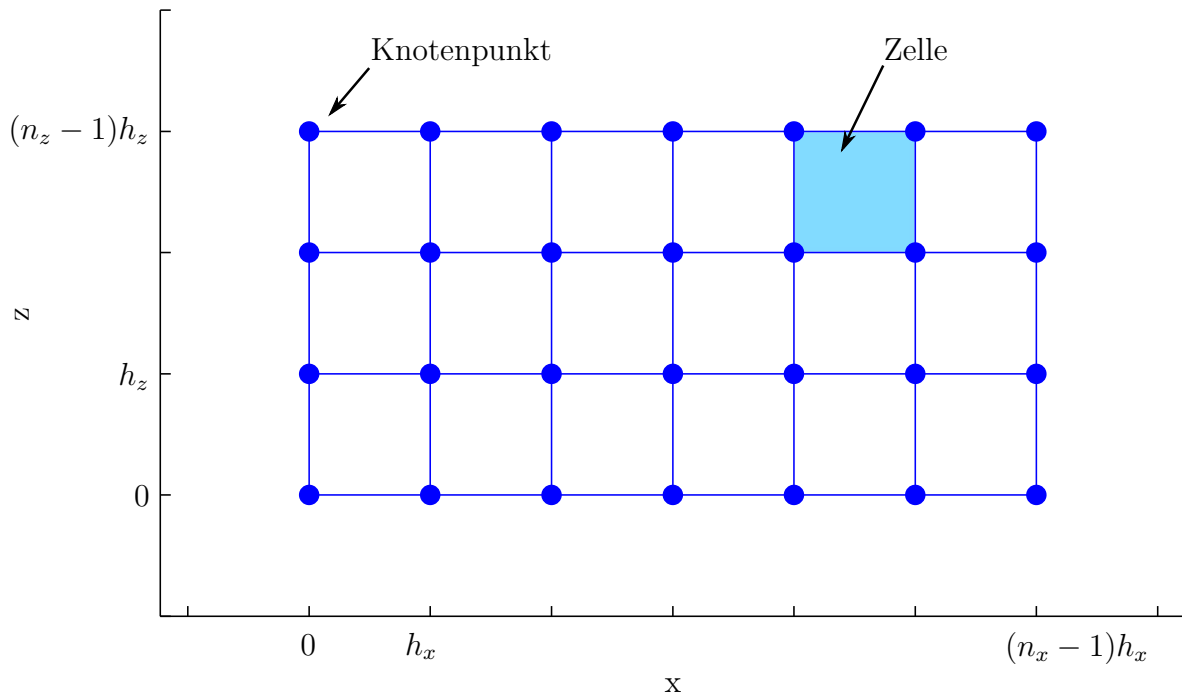


Abbildung 4.2: Grafische Darstellung der Gitterfunktion

4.3 Vorverarbeitungsschritt

Um die Außenkontur eines Fahrzeugs detailliert zu berechnen, ist es erforderlich die Eingangsdaten (Punktmenge P) in einem Vorverarbeitungsschritt zu bearbeiten. Dabei wird die Punktmenge P verschoben und rotiert, sodass die Anzahl der Punkte, die in eine Zelle fallen, gesehen über alle Zellen, minimal ist. Somit wird sichergestellt, dass möglichst viele Zellen, mit jedoch minimaler Anzahl an 3D Punktmessungen pro Zelle, gestützt werden. Diese Bedingung wird im Folgenden als Kriterium für ein Minimierungsproblem formuliert.

Abbildung 4.3 zeigt exemplarisch die Notwendigkeit zur Drehung der Punktmenge. Dargestellt ist die Draufsicht in die x-y-Ebene. Zur vereinfachten Erklärung wird, nur für dieses eine Beispiel, davon ausgegangen, dass es sich um eine Projektion der 3D Punktmessungen in die x-y-Ebene handelt. Nachteil der Vereinfachung ist, dass dabei die Krümmung des Fahrzeugs, die sich entlang der z-Achse verändert, nicht beachtet wird und die

schematisch gezeigte Außenkontur in die Punktmenge hineingezogen wird. Das Prinzip lässt sich jedoch anhand der Vereinfachung besser erklären, da die dritte Dimension fehlt. 3D Messungen vom Fahrzeugdach wurden in der Grafik entfernt.

In Abbildung 4.3 ist in blau das Gitter mit den Knotenpunkten zu sehen. Die grünen Linien mit den runden Endpunkten zeigen schematisch die spätere Verschiebung der Knotenpunkte entlang der y -Achse und deuten den Verlauf der zu berechnenden Außenkontur an. In Abbildung 4.3 a) ist die Punktmenge gedreht dargestellt. Es fallen die meisten 3D Punktmessungen, welche vom Heck des Fahrzeugs stammen, in zwei Zellen. Zu sehen ist dies auf der rechten Seite, hervorgehoben durch ein gelbes Rechteck. Es lässt sich erkennen, dass die Streuung der Punkte innerhalb der Zellen sehr groß im Vergleich zu den Zellen auf der linken Seite der Grafik ist. Da die Flächenberechnung auf einer Ausgleichsrechnung basiert, wird der Knotenpunkt in die Mitte des Fahrzeughecks gezogen und entspricht nicht mehr dem wirklichen Verlauf. Abbildung 4.3 b) zeigt eine bessere Drehung, da sich hier die 3D Punktmessungen des Fahrzeughecks auf fünf Zellen aufteilen (gelbes Rechteck) und es sich somit eine geringe Streuung in den Zellen ergibt.

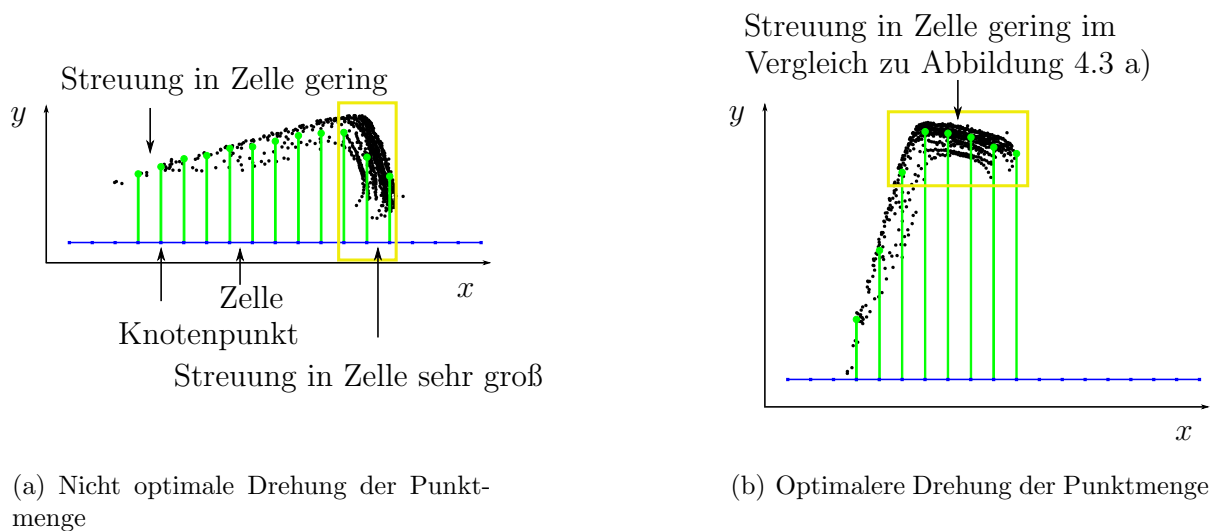


Abbildung 4.3: Detailliertere Außenkonturbeschreibung durch Drehung der Punktmenge im Vorverarbeitungsschritt

Die Wahl der Zellgröße für die Gitterfunktion V ist abhängig von der Dichte der Messungen in Punktmenge P . Diese hängt wiederum vom verwendeten Sensor ab. Für die Anwendung mit einem Velodyne HDL-64E wird eine Kantenlänge der Zellen von $0.2m$ verwendet. Um etwas höhere Geschwindigkeiten des Verfahrens zu erzielen, kann die Kantenlänge der Zellen auch vergrößert werden.

Zunächst erfolgt eine Verschiebung der Punktmenge P mit Hilfe des Vektors \vec{c}_P , sodass der neue Schwerpunkt im Koordinatenursprung liegt:

$$P'_j = P_j - \vec{c}_P \text{ mit } j \in N_s \quad (4.3)$$

Für die Formulierung des Minimierungsproblems werden weitere Funktionen definiert.

$$R_z(P, \alpha) \tag{4.4}$$

rotiert eine Punktmenge P oder einen einzelnen Punkt mit dem Winkel α um die z -Achse.

$$MIN_x(P) \tag{4.5}$$

bestimmt den minimalen x -Wert der Punktmenge P .

$$MIN_z(P) \tag{4.6}$$

bestimmt den minimalen z -Wert der Punktmenge P .

$$MAX_y(P) \tag{4.7}$$

bestimmt den maximalen y -Wert der Punktmenge P .

Die Funktion

$$B(V, P) \tag{4.8}$$

projiziert alle Punkte einer Punktmenge P entlang der y -Achse in das Gitter V und zählt dabei die Anzahl der Punkte, die in jede Zelle fallen.

Abbildung 4.4 zeigt beispielhaft die Projektion einer Punktmenge in ein Gitter.

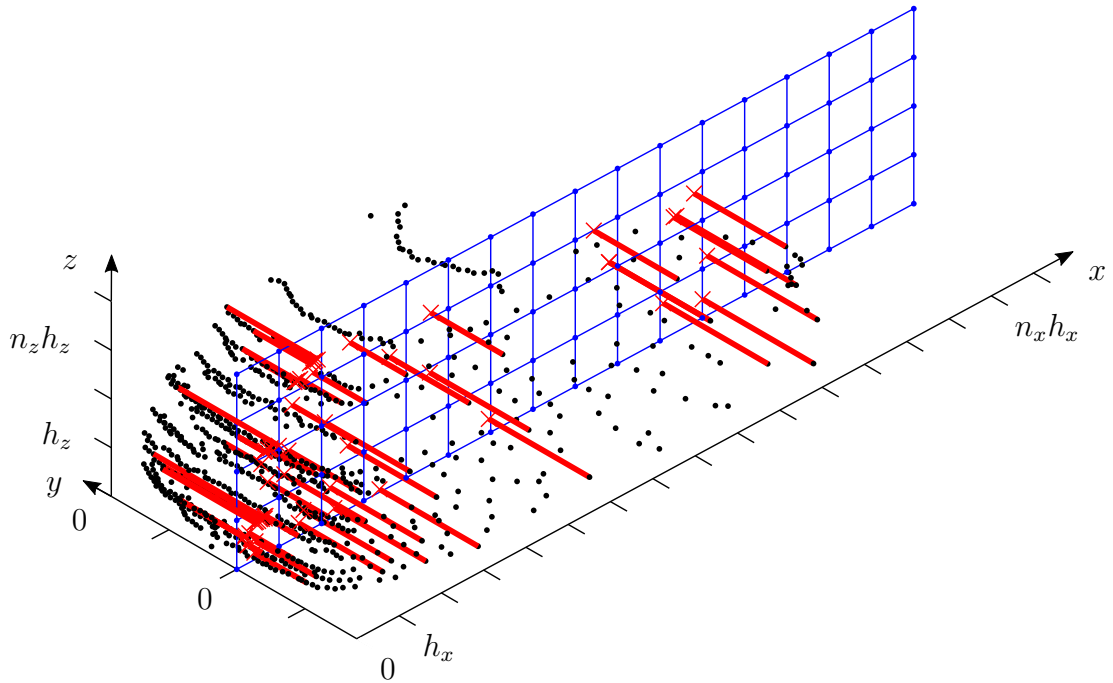


Abbildung 4.4: Beispiel für die Projektion einer Punktmenge in ein Gitter. Die 3D Messungen sind in schwarz dargestellt. In rot wird die Projektion für einige Punkte der Punktmenge gezeigt. Das Auftreffen eines projizierten Punktes auf das Gitter ist durch ein Kreuz gekennzeichnet.

Die Verschiebung einer Punktmenge P entlang der x -Achse und der z -Achse mit der Bedingung, dass alle x - und z -Werte größer oder gleich Null sein sollen wird in der Funktion

$$T_1(P) \quad (4.9)$$

zusammengefasst. Dabei wird von jedem Punkt der Punktmenge P der Vektor

$$(x_{min}, 0, z_{min})^T$$

mit $x_{min} = MIN_x(P)$ und $z_{min} = MIN_z(P)$ subtrahiert.

Das Minimierungsproblem wird mit den zuvor definierten Funktionen wie folgt formuliert

$$\min_{\alpha} B\left(V, T_1(R_z(P', \alpha))\right) \text{ mit } \alpha = -\frac{\pi}{2} \dots \frac{\pi}{2} \quad (4.10)$$

Unabhängig davon in welchem Quadranten sich die 3D Messungen befinden und unabhängig davon unter welchem Aspektwinkel das Fahrzeug gesehen wurde (Drehungen des Fahrzeugs im Quadranten), wird das Minimierungsproblem im Bereich $-\frac{\pi}{2} \dots \frac{\pi}{2}$ definiert. In diesem Bereich weist die Funktion ein Minimum (α) auf. Bedingt durch die Definition der späteren Flächenfunktion und ihren Flächeneigenschaften, im speziellen die Konvexität, kann es sein, dass die gefundene Drehung um den Winkel π angepasst werden muss. Erfolgt dieser Schritt nicht, kann die Konvexität aus Sicht der Flächenfunktion nicht erfüllt werden, da es sich um eine konkave Kontur handelt. Die Überprüfung wird nach der Lösung des Minimierungsproblems durchgeführt. Abbildung 4.5 zeigt beispielhaft für die in Abbildung 4.4 dargestellte Punktmenge den Funktionsgraphen der zu minimierenden Funktion aus Gleichung 4.10. Die Lösung des Minimierungsproblems ist mit einem grünen Kreis markiert.

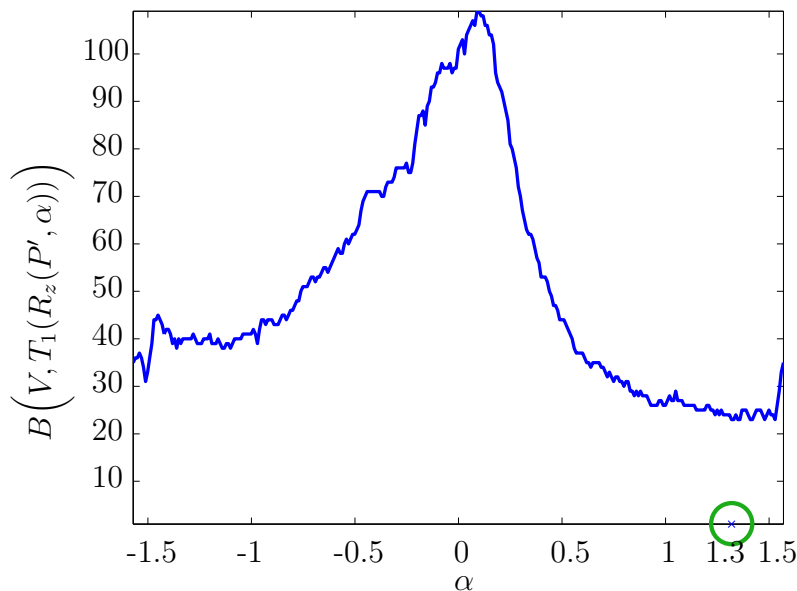


Abbildung 4.5: Beispiel eines Funktionsgraphen der Minimierungsfunktion aus Gleichung 4.10. In grün ist das zu suchende Minimum markiert.

Für die Prüfung auf Gültigkeit des Winkels α wird ein beliebiger Punkt P_j mit $j \in N_s$ aus der Punktwolke P' ausgewählt. Der Sensorursprung, der sich zuvor in $\vec{s}_o = (0, 0, 0)^T$ befunden hat, wird entsprechend der Verschiebung des Schwerpunktes der Punktwolke P ebenfalls verschoben und danach mit dem Winkel α um die z-Achse gedreht. Der Sensorursprung befindet sich daraufhin in

$$\vec{s}'_o = R_z(\vec{s}_o - \vec{c}_p, \alpha) \quad (4.11)$$

Mit beiden Punkten \vec{s}'_o und P_j lässt sich eine Gerade definieren, deren Schnittpunkt \vec{c}_E mit einer Ebene E bestimmt wird. Die Ebene E ist definiert als

$$E : \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \left[\vec{x} - \begin{pmatrix} 0 \\ y_E \\ 0 \end{pmatrix} \right] = 0 \quad (4.12)$$

wobei gilt $y_E > MAX_y(P')$

d_1 beschreibt die Distanz zwischen verschobenem und gedrehtem Sensorursprung \vec{s}'_o und dem Schnittpunkt \vec{c}_E

$$d_1 = \left\| \vec{s}'_o - \vec{c}_E \right\|_2 \quad (4.13)$$

d_2 beschreibt die Distanz zwischen verschobenem und gedrehtem Sensorursprung \vec{s}'_o und dem Punkt P_j aus der Punktmenge P'

$$d_2 = \left\| \vec{s}'_o - P_j \right\|_2 \quad (4.14)$$

Die Gültigkeitsprüfung, und falls notwendig die Anpassung des Winkels, erfolgt über nachfolgende Funktion

$$\alpha_R = \begin{cases} \alpha & \text{wenn } d_2 > d_1 \\ \alpha + \pi & \text{sonst} \end{cases} \quad (4.15)$$

Abbildung 4.6 verdeutlicht die beschriebenen Schritte grafisch.

Die endgültige Drehung und Verschiebung der Punktmenge P' erfolgt über

$$P_R = T_1 \left(R_z(P', \alpha_R) \right) \quad (4.16)$$

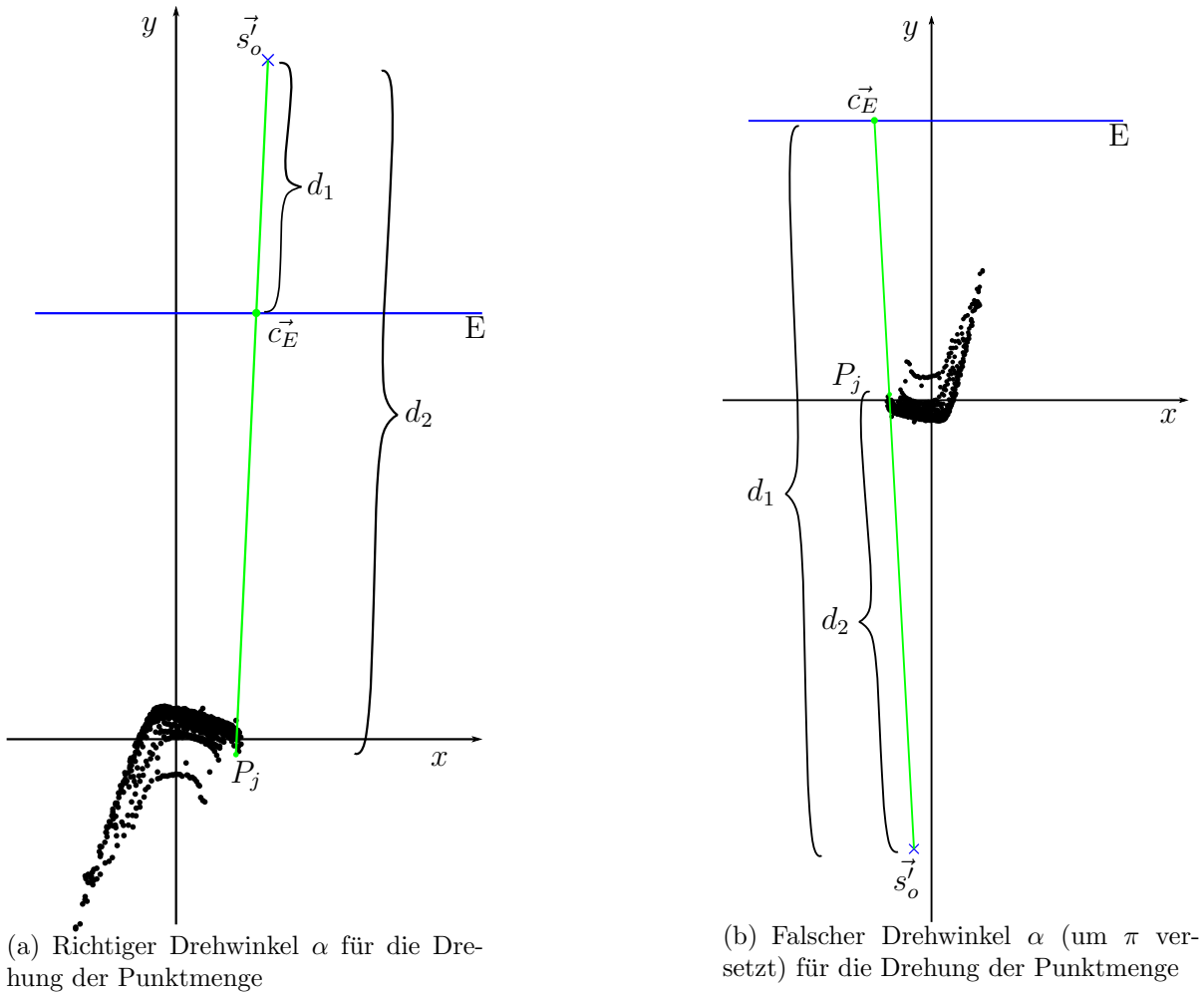


Abbildung 4.6: Prüfung auf Gültigkeit des Drehwinkels α

Aus Abbildung 4.6 geht hervor, dass in dem gezeigten Beispiel zum gefundenen Winkel α der Winkel π addiert werden muss, um die späteren Bedingungen für die Flächenfunktion zu erfüllen (Konvexität).

4.4 Flächenfunktion

Die in Gleichung 4.1 eingeführte Gitterfunktion V wurde im Vorverarbeitungsschritt für die Bestimmung der notwendigen Drehung der Punktmenge genutzt. Im weiteren Verlauf wird auf der Gitterfunktion V aufbauend die Flächenfunktion definiert, die später die Außenkontur eines Fahrzeugs repräsentieren wird. Die Flächenfunktion basiert auf einem Ansatz, der ursprünglich von John D'Errico [18] vorgestellt wurde.

Unter Umständen kann es sinnvoll sein, gegenüber dem Vorverarbeitungsschritt die Zellgröße der Gitterfunktion V für die Flächenfunktion zu verfeinern. Im Folgenden wird mit einer Zellgröße von $0.2m$ gerechnet.

Die Flächenfunktion soll sich an die Punktmenge P_R anpassen. Dabei dienen die y -Werte der Knotenpunkte als Freiheitsgrade. Dies bedeutet, dass sich die y -Werte der Knoten durch Variation an die y -Werte der Punktmenge P_R derart anpassen, dass der quadratische Abstand minimiert wird. Wie im Vorverarbeitungsschritt wird auch hier bestimmt, in welche Zelle jeder Punkt aus der Punktmenge P_R fällt. Jeder Punkt ist dadurch eindeutig einer Zelle zugewiesen und trägt zum y -Wert der vier Knotenpunkte, die die Zelle definieren, bei. Der Grad, zu wie viel Prozent der Punkt den y -Wert der jeweiligen Knotenpunkte beeinflusst, wird über die bilineare Interpolation bestimmt.

Prinzipiell sind bei der bilinearen Interpolation die Knotenpunkte bekannt und es soll ein dazwischen liegender Punkt bestimmt werden. Als Beispiel sind in Abbildung 4.7 vier Knotenpunkte

- 1.) $(x_i, V(x_i, z_k), z_k)^T$,
- 2.) $(x_{i+1}, V(x_{i+1}, z_k), z_k)^T$,
- 3.) $(x_{i+1}, V(x_{i+1}, z_{k+1}), z_{k+1})^T$,
- 4.) $(x_i, V(x_i, z_{k+1}), z_{k+1})^T$

und ein zu interpolierender Punkt $P_{int} = (x_M, V(x_M, z_M), z_M)^T$ dargestellt.

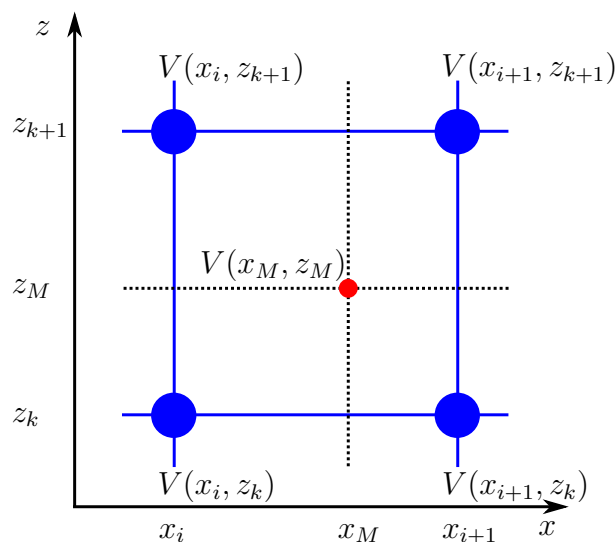


Abbildung 4.7: Grafische Darstellung der bilinearen Interpolation

Für die Berechnung des y -Wertes des Punktes P_{int} wird zunächst eine lineare Interpolation in Richtung der x -Achse durchgeführt. Es ergibt sich

$$a_x = \frac{x_{i+1} - x_M}{x_{i+1} - x_i} \quad (4.17)$$

$$b_x = \frac{x_M - x_i}{x_{i+1} - x_i} \quad (4.18)$$

$$V(x_M, z_k) \approx a_x V(x_i, z_k) + b_x V(x_{i+1}, z_k) \quad (4.19)$$

und

$$V(x_M, z_{k+1}) \approx a_x V(x_i, z_{k+1}) + b_x V(x_{i+1}, z_{k+1}) \quad (4.20)$$

Danach folgt die Interpolation entlang der z -Achse

$$a_z = \frac{z_{k+1} - z_M}{z_{k+1} - z_k} \quad (4.21)$$

$$b_z = \frac{z_M - z_k}{z_{k+1} - z_k} \quad (4.22)$$

$$V(x_M, z_M) \approx a_z V(x_M, z_k) + b_z V(x_M, z_{k+1}) \quad (4.23)$$

Da in diesem Fall der Punkt P_{int} bekannt ist, jedoch die y -Werte der Knotenpunkte unbekannt sind, ergibt sich aus Gleichung 4.19, Gleichung 4.20 und Gleichung 4.23 ohne Berücksichtigung des Approximationsfehlers

$$V(x_M, z_M) = \underbrace{a_z a_x}_{c_1} V(x_i, z_k) + \underbrace{a_z b_x}_{c_2} V(x_{i+1}, z_k) + \underbrace{b_z a_x}_{c_3} V(x_i, z_{k+1}) + \underbrace{b_z b_x}_{c_4} V(x_{i+1}, z_{k+1}) \quad (4.24)$$

Die Gleichung 4.24 lässt sich für jeden Punkt aus der Punktmenge P_R aufstellen. Je nach Anzahl der Punkte (N_p) in der Punktmenge und der Anzahl der Knoten ($N_g = n_x n_z$) entsteht entweder ein unter- oder ein überbestimmtes Gleichungssystem

$$\mathbf{A}_1 \vec{v} = \vec{y}_1 \quad (4.25)$$

mit $\mathbf{A}_1 \in \mathbb{R}^{N_p \times N_g}$, $\vec{v} \in \mathbb{R}^{N_g}$ und $\vec{y}_1 \in \mathbb{R}^{N_g}$. Dabei steht \vec{v} für den Vektor der gesuchten y -Werte der Knotenpunkte und \vec{y}_1 für die y -Werte der Punkte in der Punktmenge P_R . Matrix \mathbf{A}_1 hält jeweils zeilenweise die Koeffizienten c_1 , c_2 , c_3 und c_4 für den zugehörigen Punkt aus P_R . Die Nummerierung des Vektors \vec{v} und die damit verbundenen Zuordnung zu den Knotenpunkten erfolgt nach folgendem Schema

$$\begin{aligned} \vec{v}_0 &= V(x_0, z_0) \\ \vec{v}_1 &= V(x_0, z_1) \\ &\dots \\ \vec{v}_{n_z-1} &= V(x_0, z_{n_z-1}) \\ \vec{v}_{n_z} &= V(x_1, z_0) \\ &\dots \\ \vec{v}_{N_g-1} &= V(x_{n_x-1}, z_{n_z-1}) \end{aligned}$$

Abbildung 4.8 veranschaulicht das Schema der Nummerierung.

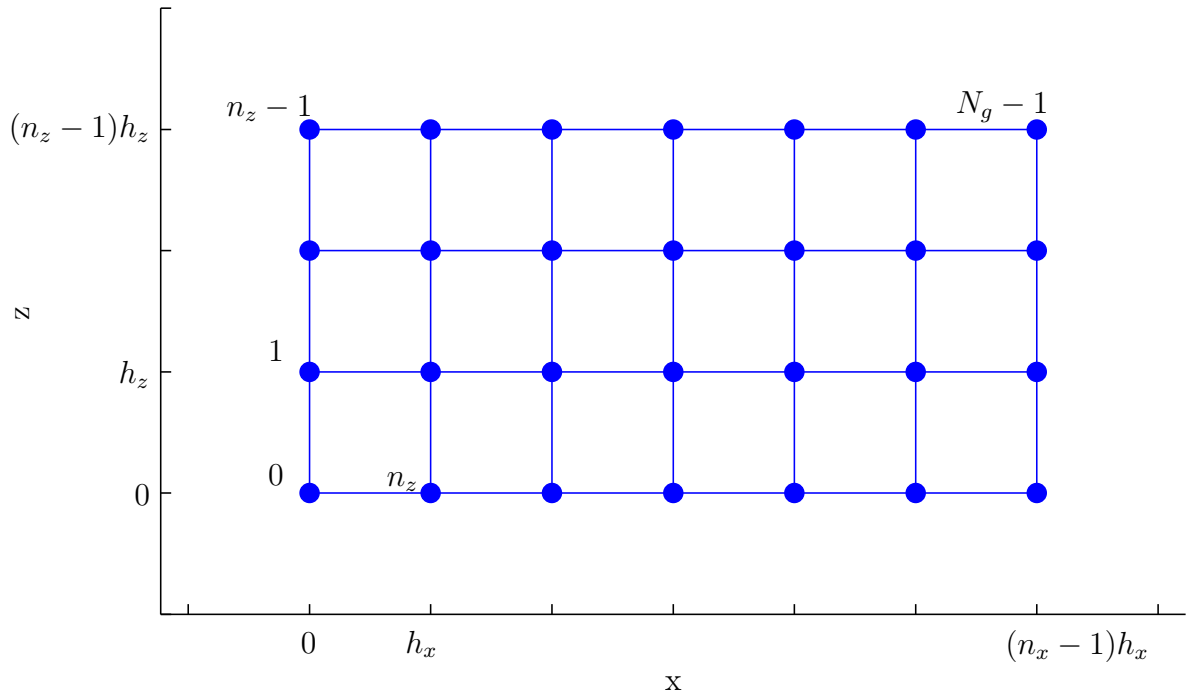


Abbildung 4.8: Nummerierung der Knotenpunkte der Flächenfunktion

Das Gleichungssystem aus Gleichung 4.25 lässt sich, falls es überbestimmt ist, lösen, indem es als Minimierungsproblem aufgefasst wird

$$\min_{\vec{v}} \|\mathbf{A}_1 \vec{v} - \vec{y}_1\|_2^2 \quad (4.26)$$

Falls das Gleichungssystem unterbestimmt ist, liegen mehr Knotenpunkte vor, als es Gleichungen gibt. Dies ist bei realen Messungen in der Regel der Fall. Im folgenden Kapitel ergeben sich weitere Gleichungen, die aus geforderten Flächeneigenschaften abgeleitet werden können. Diese werden mit \mathbf{A}_1 verknüpft und sorgen einerseits dafür, dass die Fläche eine gewünschte Form annimmt und andererseits dafür, dass das Gleichungssystem lösbar wird.

4.5 Flächeneigenschaften

Fahrzeuge weisen spezielle geometrische Eigenschaften auf, über die sie sich beschreiben lassen. Dazu zählen die Glattheit der Oberfläche ('4.5.1 Glattheit'), die Konvexität der Oberfläche entlang der x-Achse ('4.5.2 Konvexität'), die Verjüngung des Fahrzeugs entlang der z-Achse ('4.5.3 Negative Gradienten'), die Winkelbegrenzung zwischen den Fahrzeugseitenflächen ('4.5.4 Winkelbegrenzung') und ein Winkel von 90° zwischen den Fahrzeugseitenflächen und der Straßenoberfläche ('4.5.5 Orthogonalitätsbedingung'). Alle Eigenschaften sind jeweils in einer allgemeingültigen Form formuliert, sodass sie nicht speziell auf nur eine Fahrzeugklasse zutreffen. Über die Eigenschaften lassen sich somit

sowohl PKW als auch LKW beschreiben. Abbildung 4.9 zeigt alle formulierten Eigenschaften anhand eines Beispiels. Die Eigenschaften werden im Folgenden näher beschrieben und mathematisch definiert.

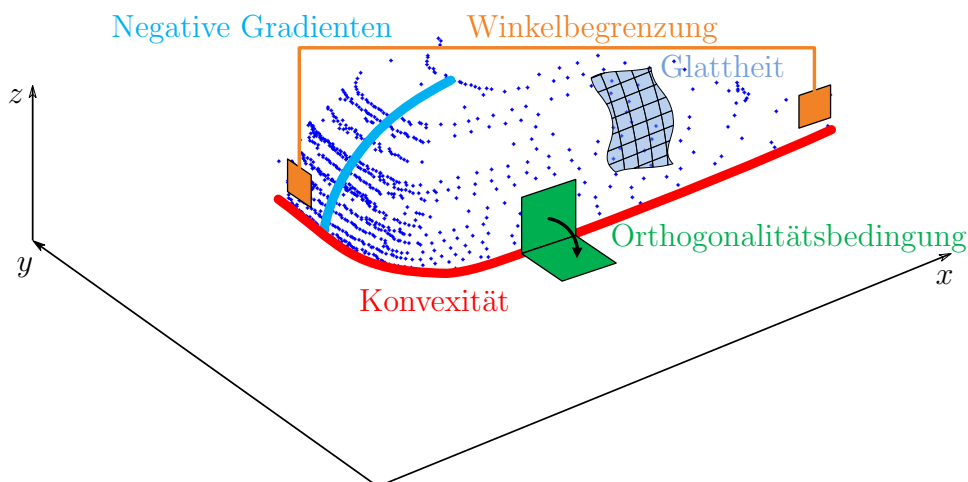


Abbildung 4.9: Grafische Darstellung der Flächeneigenschaften

Für die mathematische Beschreibung der Bedingungen werden diese im Folgenden jeweils entlang der einzelnen Achsen definiert (x -Achse und z -Achse), sofern die Bedingungen entlang der jeweiligen Achse gültig sind. Die Knoten werden für eine vereinfachte Beschreibung über zwei Indizes adressiert. Der erste Index ist die Position auf der x -Achse, der zweite auf der z -Achse. Dabei kann zu jedem Knoten durch Inkrementieren oder Dekrementieren des Index der rechte oder linke, bzw. der obere oder untere, Nachbarknoten gefunden werden. Abbildung 4.10 zeigt die vereinfachte Adressierung der Knotenpunkte.

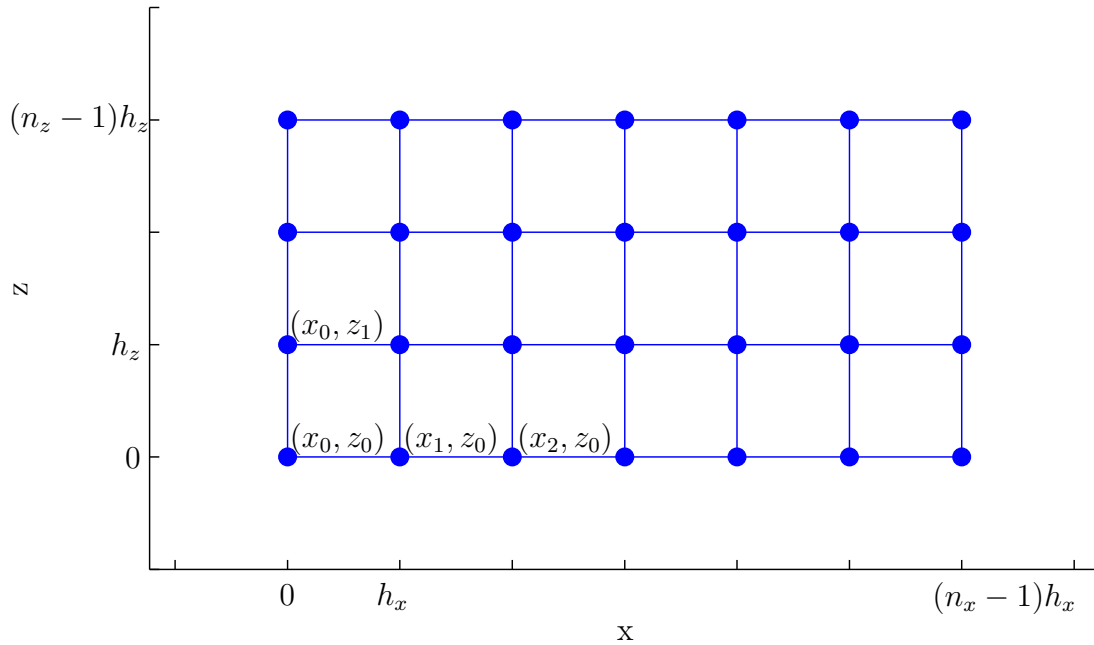


Abbildung 4.10: Vereinfachte Adressierung der Knotenpunkte

4.5.1 Glattheit

Die Oberfläche von Fahrzeugen kann prinzipiell als glatt angenommen werden. Dies bedeutet, dass keine Elemente aus der Oberfläche der Fahrzeuge herausragen. Eine Ausnahme stellt in der Regel der Spiegel dar. Dieser lässt sich meist in den Messungen des Lasersensors nicht eindeutig charakterisieren, da nur wenige Messungen von ihm in den Daten vorhanden sind. Mit steigender Entfernung des Fahrzeugs zum Sensor kann es durchaus vorkommen, dass keine Messungen mehr vom Spiegel vorhanden sind. Aus diesem Grund wird bei der weiteren Betrachtung die Oberfläche durchgängig als glatt angenommen.

Auf die Flächenfunktion übertragen bedeutet dies, dass die Änderung der Steigung aufeinanderfolgender Knotenpunkte, sowohl in x- als auch in z-Richtung, minimal sein soll. Es gilt, dass der Differenzenquotient zweiter Ordnung minimal ist.

Entlang der x-Achse gilt

$$\min V(x_{i-1}, z_k) - 2V(x_i, z_k) + V(x_{i+1}, z_k) \quad (4.27)$$

wobei

$$i \in \{1, \dots, n_x - 2\}, k \in \{0, \dots, n_z - 1\}$$

Entlang der z-Achse gilt

$$\min V(x_i, z_{k-1}) - 2V(x_i, z_k) + V(x_i, z_{k+1}) \quad (4.28)$$

wobei

$$i \in \{0, \dots, n_x - 1\}, k \in \{1, \dots, n_z - 2\}$$

Die Bedingungen lassen sich in einem weiteren Minimierungsproblem formulieren. Es ergibt sich

$$\min_{\vec{v}} \mathbf{G}\vec{v} \quad (4.29)$$

mit $\mathbf{G} \in \mathbb{R}^{N_t \times N_g}$, wobei $N_t = (n_x - 2)n_z + (n_z - 2)n_x$.

4.5.2 Konvexität

Fahrzeuge weisen bei Betrachtung aus der Vogelperspektive eine konvexe Außenkontur auf. Auch hier soll, aus gleichem Grund wie in Kapitel '4.5.1 Glattheit', der Spiegel vernachlässigt werden.

Die Konvexität für die Flächenfunktion soll nur entlang der x-Achse gelten. Entsprechend der mathematischen Definition von Konvexität und der Anordnung der Messungen aus der Punktmenge P_R ergibt sich für die Beschreibung eine Forderung nach einer konkaven Form. Das bedeutet, dass bei drei aufeinander folgenden Knotenpunkten der mittlere Knotenpunkt keinen größeren y-Wert aufweisen darf, als der maximale y-Wert zwischen linkem und rechtem Knotenpunkt.

Damit ergibt sich

$$V\left(\frac{x_{i-1} + x_{i+1}}{2}, z_k\right) \geq \frac{V(x_{i-1}, z_k) + V(x_{i+1}, z_k)}{2} \quad (4.30)$$

$$V(x_i, z_k) \geq \frac{V(x_{i-1}, z_k) + V(x_{i+1}, z_k)}{2} \quad (4.31)$$

$$V(x_{i-1}, z_k) - 2V(x_i, z_k) + V(x_{i+1}, z_k) \leq 0 \quad (4.32)$$

wobei

$$i \in \{0, \dots, n_x - 2\}, k \in \{0, \dots, n_z - 1\}$$

Die Konvexitätsbedingungen werden in einem Ungleichungssystem zusammengefasst

$$\mathbf{C}_1 \vec{v} \leq \vec{c}_1 \quad (4.33)$$

mit $\mathbf{C}_1 \in \mathbb{R}^{(n_x-2)n_z \times N_g}$, $\vec{c}_1 = 0 \in \mathbb{R}^{(n_x-2)n_z}$

4.5.3 Negative Gradienten

Bei der Betrachtung von Fahrzeugen entlang der z-Achse lässt sich bei PKW eine Verjüngung erkennen. Diese tritt in der Regel bei LKW nicht auf. Hier bleibt der y-Wert entlang der z-Achse konstant. Beide Eigenschaften lassen sich in einem Gleichungssystem formulieren. Dabei wird zusätzlich zum negativen Gradienten auch der Gradient von Null zugelassen.

$$\frac{V(x_i, z_{k+1}) - V(x_i, z_k)}{h_z} \leq 0 \quad (4.34)$$

mit

$$i \in \{0, \dots, n_x - 1\}, k \in \{0, \dots, n_z - 2\}$$

Die Bedingungen werden in einem weiteren Ungleichungssystem zusammengefasst

$$\mathbf{C}_2 \vec{v} \leq \vec{c}_2 \quad (4.35)$$

mit $\mathbf{C}_2 \in \mathbb{R}^{(n_x-1)n_z \times N_g}$, $\vec{c}_2 = \mathbf{0} \in \mathbb{R}^{(n_x-1)n_z}$

4.5.4 Winkelbegrenzung

Der Winkel zwischen zwei Seitenflächen eines Fahrzeugs beträgt in der Regel 90° . Das bedeutet, dass zwischen dem linken und dem rechten Rand der Flächenfunktion ein entsprechender Winkel β gefordert wird. Hierbei wird jedoch ein Winkel von 90° und größer erlaubt. Falls nur eine Seite eines Fahrzeugs aufgrund des Sichtbereichs des Sensors zu sehen ist, ist der Winkel zwischen linkem und rechtem Rand maximal 180° . Da die Konvexität nicht als streng konvex formuliert wurde, ist dieser Winkel möglich.

Mathematisch beschrieben werden kann die Winkelbegrenzung indem jeweils der Gradient zwischen den beiden letzten Knotenpunkten und der Gradient zwischen den ersten beiden Knotenpunkten entlang der x-Achse in Winkel umgerechnet werden und ein entsprechender Schwellwert für die Differenz festgelegt wird. Wenn aus den Gradienten am linken und rechten Rand der Flächenfunktion Winkel berechnet werden, ergeben sich nicht lineare Bedingungen. Um dies zu umgehen, wird nur die Differenz der Gradienten genutzt, die kleiner als ein Parameter ζ sein muss. Dies hat zur Folge, dass die Winkelbegrenzung nur noch näherungsweise gilt.

Grafisch sind die Gradienten in Abbildung 4.11 in grün dargestellt. Winkel β , den es zu begrenzen gilt, ist ebenfalls eingezeichnet.

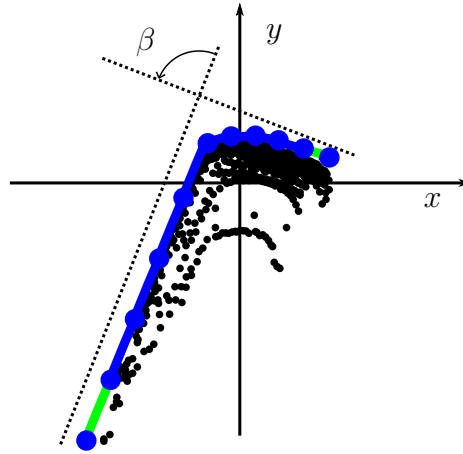


Abbildung 4.11: Grafische Darstellung der Winkelbegrenzung entlang der x-Achse.

Es gilt:

$$\frac{1}{h_x}(V(x_1, z_k) - V(x_0, z_k)) - \frac{1}{h_x}(V(x_{n_x-1}, z_k) + V(x_{n_x-2}, z_k)) < \zeta \quad (4.36)$$

wobei $k \in \{0, \dots, n_z - 1\}$ und h_x die Schrittweite der Gitterfunktion entlang der x-Achse ist.

Die Bedingungen werden in einem weiteren Ungleichungssystem zusammengefasst

$$\mathbf{C}_3 \vec{v} \leq \vec{c}_3 \quad (4.37)$$

mit $\mathbf{C}_3 \in \mathbb{R}^{n_z \times N_g}$, $\vec{c}_3 = \zeta \in \mathbb{R}^{n_z}$.

Das folgende Beispiel verdeutlicht den Zusammenhang zwischen dem Winkel β und dem Parameter ζ . Wird für ζ frei der Wert 0.8 gewählt, kann eine Ebene E_1 in den Raum gelegt werden, der von den beiden möglichen Gradienten am linken und rechten Rand der Flächenfunktion aufgespannt wird. Zwischen dieser Ebene und der, die durch die Differenz der beiden Gradienten im selben Raum aufgespannt wird (E_2), kann die Schnittgerade berechnet werden. Diese ist in Abbildung 4.12 in schwarz dargestellt.

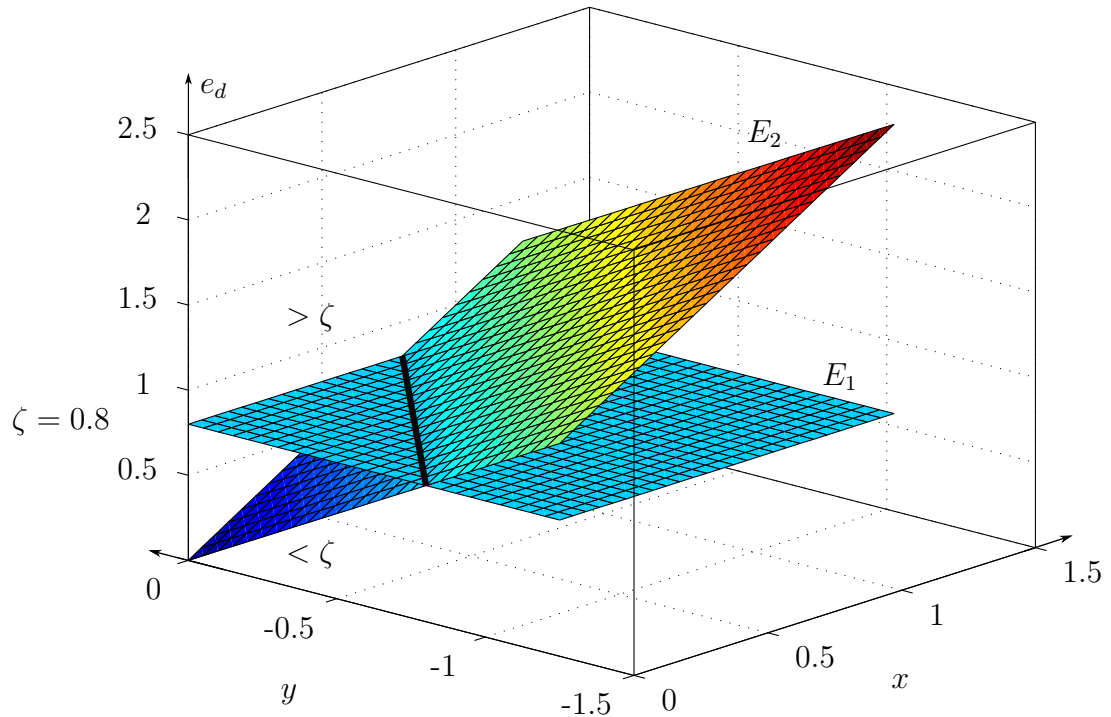


Abbildung 4.12: Grafische Darstellung der Schnittgeraden, wobei gilt, $x = \frac{1}{h_x}(V(x_1, z_k) - V(x_0, z_k))$ und $y = \frac{1}{h_x}(V(x_{n_x-1}, z_k) + V(x_{n_x-2}, z_k))$

Wird statt der Differenz der Gradienten der Winkel berechnet und die Schnittgerade aus dem vorherigen Schritt durch eine Ebene E_3 ersetzt, zeigt sich in Abbildung 4.13 welche Winkel β möglich sind. Je nach Zusammenhang zwischen dem Gradienten auf der linken und dem auf der rechten Seite sind unterschiedliche Winkel erlaubt. In diesem Beispiel ergeben sich Winkel zwischen 180° bis ca. 50° .

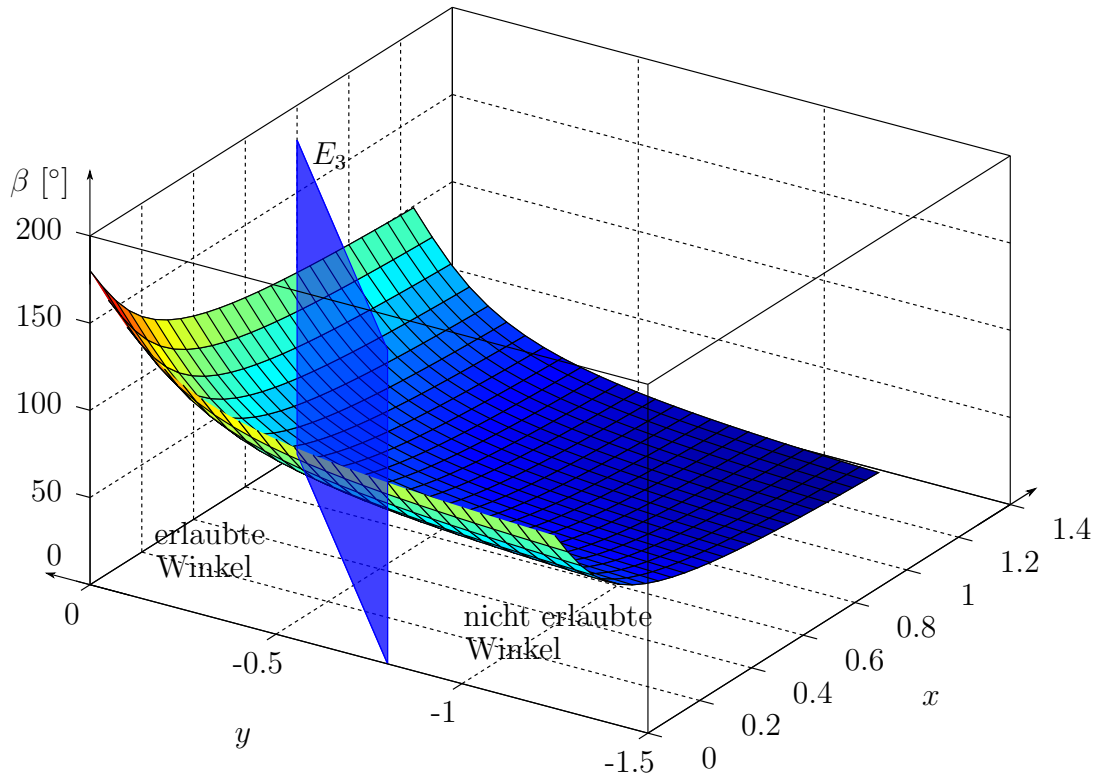


Abbildung 4.13: Grafische Darstellung der erlaubten Winkel β , wobei gilt $x = \frac{1}{h_x}(V(x_1, z_k) - V(x_0, z_k))$ und $y = \frac{1}{h_x}(V(x_{n_x-1}, z_k) + V(x_{n_x-2}, z_k))$

4.5.5 Orthogonalitätsbedingung

Da sich Fahrzeuge auf der Straße befinden, ergibt sich ein Winkel von 90° zwischen Straße und der Berührungsstelle der Seitenfläche des Fahrzeugs. Die Bedingung lässt sich formulieren indem gefordert wird, dass die Steigung entlang der z -Achse zwischen den ersten beiden Knotenpunkten Null sein muss. Es ergibt sich daraus

$$V(x_i, z_0) - V(x_i, z_1) = 0 \quad (4.38)$$

wobei $i \in \{0, \dots, n_x - 1\}$

Die Bedingungen werden in einem Gleichungssystem zusammengefasst

$$\mathbf{B}\vec{v} = \vec{b} \quad (4.39)$$

mit $\mathbf{B} \in \mathbb{R}^{n_x \times N_g}$, $\vec{b} = 0 \in \mathbb{R}^{n_x}$

4.6 Das Optimierungsproblem

Im Folgenden wird das gesamte Optimierungsproblem zusammengestellt. Dabei werden alle zuvor definierten Bedingungen, die an die Außenkontur eines Fahrzeugs gestellt werden (Kapitel '4.5.1 Glattheit' bis Kapitel '4.5.5 Orthogonalitätsbedingung'), mit dem

Grundgleichungssystem aus Gleichung 4.26 verknüpft.

Mit Gleichung 4.29 lässt sich das Gleichungssystem aus Gleichung 4.26 erweitern. Es folgt

$$\min_{\vec{v}} \quad \|\mathbf{A}\vec{v} - \vec{y}\|_2^2 \quad (4.40)$$

mit $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{G} \end{pmatrix} \in \mathbb{R}^{N_m \times N_g}$, $\vec{y} = \begin{pmatrix} \vec{y}_1 \\ 0 \end{pmatrix} \in \mathbb{R}^{N_m}$ und $N_m = N_t + N_p$.

Die Ungleichungssysteme der Konvexität (Gleichung 4.33), der negativen Gradienten (Gleichung 4.35) und der Winkelbegrenzung (Gleichung 4.37) lassen sich in einem gemeinsamen Ungleichungssystem zusammenfassen. Es ergibt sich mit

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{pmatrix}, \vec{c} = \begin{pmatrix} \vec{c}_1 \\ \vec{c}_2 \\ \vec{c}_3 \end{pmatrix}$$

$$\mathbf{C}\vec{v} \leq \vec{c} \quad (4.41)$$

$\mathbf{C} \in \mathbb{R}^{N_c \times N_g}$ und $\vec{c} \in \mathbb{R}^{N_c}$

wobei $N_c = (n_x - 2)n_z + (n_x - 1)n_z + n_z$

Das Optimierungsproblem ergibt sich zu

$$\min \quad \|\mathbf{A}\vec{v} - \vec{y}\|_2^2 \quad \text{auf} \quad (4.42)$$

$$\left\{ \vec{v} \in \mathbb{R}^{N_g} \quad : \quad \mathbf{B}\vec{v} - \vec{b} = 0, \mathbf{C}\vec{v} - \vec{c} \leq 0 \right\}$$

wobei

$$\mathbf{A} \in \mathbb{R}^{N_m \times N_g},$$

$$\vec{y} \in \mathbb{R}^{N_m},$$

$$\mathbf{C} \in \mathbb{R}^{N_c \times N_g},$$

$$\vec{c} \in \mathbb{R}^{N_c},$$

$$\mathbf{B} \in \mathbb{R}^{n_x \times N_g},$$

$$\vec{b} \in \mathbb{R}^{n_x}$$

Das Optimierungsproblem aus Gleichung 4.42 lässt sich auf die allgemeine Form eines quadratischen Optimierungsproblems umwandeln. Dazu wird die Funktion $F = \|\mathbf{A}\vec{v} - \vec{y}\|_2^2$ aus Gleichung 4.42 ausmultipliziert

$$F = (\mathbf{A}\vec{v} - \vec{y})^T (\mathbf{A}\vec{v} - \vec{y}) \quad (4.43)$$

$$= \vec{v}^T \mathbf{A}^T \mathbf{A} \vec{v} - 2\vec{y}^T \mathbf{A} \vec{v} + \underbrace{\vec{y}^T \vec{y}}_{\text{konstant}}$$

Bei der Minimierung verändert sich das Minimum durch die Addition einer Konstanten oder die Multiplikation mit einem Faktor nicht. Daher folgt

$$F = \frac{1}{2} \vec{v}^T \mathbf{A}^T \mathbf{A} \vec{v} - \vec{y}^T \mathbf{A} \vec{v} \quad (4.44)$$

Die allgemeine Form des strikt quadratischen Optimierungsproblems lautet mit $\mathbf{Q} = \mathbf{A}^T \mathbf{A}$ und $q^T = -\vec{y}^T \mathbf{A}$

$$\begin{aligned} \min \quad & f(\vec{v}) = \frac{1}{2} \vec{v}^T \mathbf{Q} \vec{v} + q^T \vec{v} \\ \text{auf } \{ \vec{v} \in \mathbb{R}^{N_g} \quad & : \quad \mathbf{B} \vec{v} - \vec{b} = 0, \mathbf{C} \vec{v} - \vec{c} \leq 0 \} \end{aligned} \quad (4.45)$$

Die Lösung des Optimierungsproblems entspricht den Knotenpunkten der Flächenfunktion. Die Knoten werden in einer Punktmenge $F_k \in \mathbb{R}^{N_g \times 3}$ zusammengefasst.

4.7 Lösung des Optimierungsproblems

Im folgenden Abschnitt wird ein für das beschriebene Optimierungsproblem aus Gleichung 4.45 geeignetes Lösungsverfahren vorgestellt. Daran schließt sich ein Abschnitt an, der Eigenschaften des Optimierungsproblems und des Lösungsverfahrens beschreibt, die sich für eine effiziente Lösung nutzen lassen.

4.7.1 Algorithmus von Goldfarb und Idnani

Bei dem in Gleichung 4.45 gezeigten Optimierungsproblem handelt es sich um ein strikt konvexes Optimierungsproblem mit Gleichungs- und Ungleichungsbedingungen. Lösen lässt es sich durch unterschiedliche in der Literatur bekannte Algorithmen. Ein Beispiel stellt die Aktive Mengen Methode dar, welche unter anderem von Walter Alt [2] beschrieben wird.

Als nachteilig erweist sich bei der üblichen Aktiven Mengen Methode, dass eine Startlösung bekannt sein muss. Diese liegt im hier beschriebenen Fall nicht vor, was dazu führt, dass sich diese Methode direkt nicht einsetzen lässt. Andere Methoden sind diejenigen, die beispielsweise Transformationen nutzen, um das ursprüngliche quadratische Optimierungsproblem mit seinen Restriktionen auf eine andere Form zu bringen. Dabei gilt es jedoch zu beachten, dass sich bei diesen Methoden die Formen der Matrizen ändern können. Diese können in ihrer Form beispielsweise größer werden, was sich auf die Laufzeit bei der Lösung der Gleichungssysteme negativ auswirkt. Des Weiteren kann sich der Besetzungsgrad der, wie später dargestellt wird, sehr dünn besetzten Matrizen, ändern. Dadurch lassen sich keine auf dünn besetzte Matrizen optimierte Lösungsverfahren für Gleichungssysteme einsetzen, was sich ebenfalls negativ auf die Laufzeit auswirkt. Ferner erhöht sich der nötige Speicherbedarf, je mehr Elemente der Matrizen besetzt sind.

Ein für die vorliegende Anwendung geeignetes Verfahren stellt das von Donald Goldfarb und Ashok Idnani [39] vorgeschlagene dar. Es handelt sich hierbei um ein auf der Aktiven

Mengen Methode aufbauendes Verfahren. Es wird zunächst von einem Optimierungsproblem ohne Randbedingungen ausgegangen. Die sich daraus errechnete Lösung wird als Startpunkt für die folgende iterative Bestimmung der optimalen Lösung genutzt. Die Eigenschaften der Matrizen bleiben erhalten, sodass sich deren dünn besetzte Struktur nutzen lässt.

Für eine detaillierte Beschreibung des Verfahrens sei auf die entsprechende Veröffentlichung ([39]) verwiesen. Die nachfolgende Beschreibung soll das Grundprinzip des Verfahrens veranschaulichen.

Als Startlösung wird die Lösung des unrestringierten Optimierungsproblems verwendet. Hieraus ergibt sich ein Lösungspaar (\vec{v}_0, \emptyset) . Der Index bei \vec{v} zeigt den Iterationsschritt an. Das Lösungspaar beinhaltet die Lösung \vec{v} und die Menge der dabei beachteten Restriktionen. Die Menge beinhaltet dabei die Indizes der Restriktionen, die bei der Lösung beachtet wurden. Zu Beginn werden keine Restriktionen beachtet, woraus sich die leere Menge ergibt. Im nächsten Schritt wird eine beliebige verletzte Restriktion gewählt und deren Index in die Menge aufgenommen. Wird eine Lösung für das Gleichungssystem mit den Restriktionen gefunden, ergibt sich ein neues Lösungspaar (\vec{v}_1, \mathcal{M}) . Kann keine Lösung gefunden werden, bricht das Verfahren ab. Da bei dem unrestringierten Optimierungsproblem begonnen wurde, folgt $f(\vec{v}_0) < f(\vec{v}_1)$. Das bedeutet, dass die Lösung mit einer weiteren Restriktion einen größeren Funktionswert aufweist. Die Iterationen werden so lange wiederholt, bis alle Restriktionen erfüllt sind, oder das Verfahren abbricht, falls keine Lösung möglich ist.

Die Lösung des Optimierungsproblems ist in Abbildung 4.14 dargestellt. Die Knotenpunkte stellen die eigentliche Lösung dar. Zur Visualisierung sind sie miteinander verbunden. Die Messungen entsprechen denen, die auch in den vorherigen Abschnitten zur Illustration verwendet wurden.

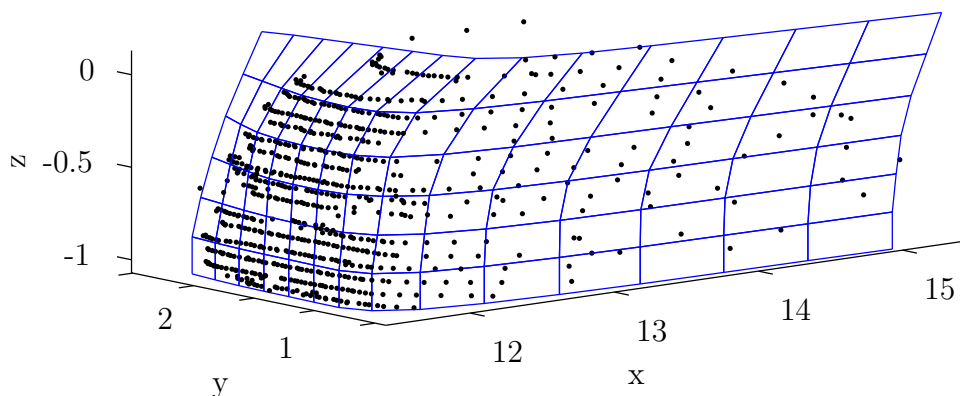


Abbildung 4.14: Lösung des Optimierungsproblems. Die Knotenpunkte sind zur Visualisierung miteinander verbunden.

4.7.2 Effiziente Lösung des Optimierungsproblems

Das in '4.7.1 Algorithmus von Goldfarb und Idnani' vorgestellte Verfahren ist ein iterativer Ansatz zur Lösung des restringierten quadratischen Optimierungsproblems aus Gleichung 4.45. Da es in jedem Schritt erforderlich ist für die Bestimmung des nächsten Lösungspaars ein Gleichungssystem zu lösen, muss der Ansatz zur Lösung des Gleichungssystems an die Gegebenheiten angepasst werden, um eine möglichst kurze Laufzeit des gesamten Algorithmus zu erzielen.

4.7.2.1 Allgemeine Möglichkeiten zur effizienten Lösung des Optimierungsproblems

Im vorherigen Abschnitt wurde bereits angesprochen, dass die Matrizen des Optimierungsproblems sehr dünn besetzt sind. Werden die Messungen, welche in Abbildung 4.1 (a) dargestellt sind, als Beispiel verwendet, ergeben sich die besetzten Elemente der Matrizen wie in Abbildung 4.15 (a) - (c) dargestellt. Alle besetzten Elemente der Matrizen sind blau hervorgehoben, die unbesetzten Elemente sind weiß. Die Grafiken ergeben sich bei einer Zellgröße von $0.1m$. Matrix **A** hat eine Besetzung von ca. 1.5 %, Matrix **B** von ca. 0.2 % und Matrix **C** von ca. 0.4 %. Die Grafiken legen nahe, dass bei der Implementierung auf effiziente Datenstrukturen zu achten ist. Dies ermöglicht einerseits geringeren Speicherverbrauch und andererseits eine höhere Geschwindigkeit, sofern die Algorithmen zur Lösung der Gleichungssysteme ebenfalls auf dünn besetzte Matrizen angepasst sind.

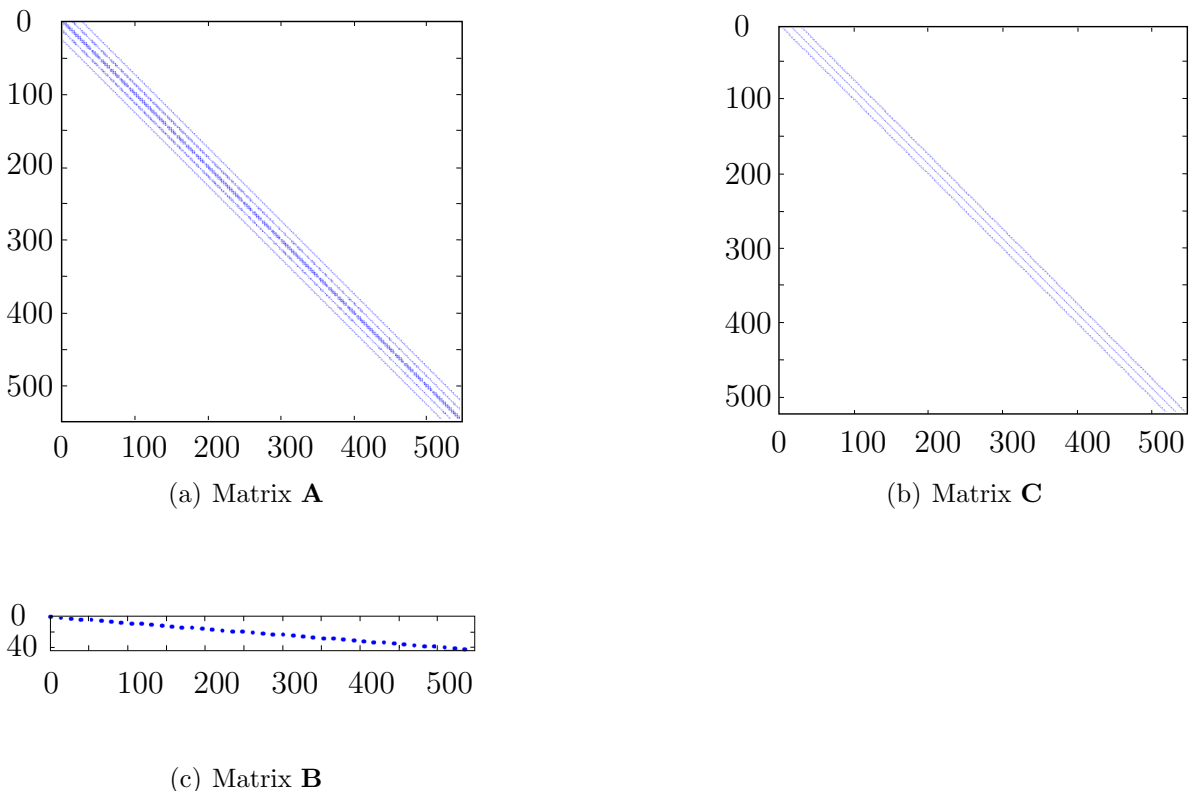


Abbildung 4.15: Besetzte Elemente der Matrizen des Optimierungsproblems

Die dünnbesetzte Struktur der Matrizen lässt sich bei der Implementierung derart nutzen, dass nur die besetzten Elemente der Matrix gespeichert werden. Diese werden in einem Array abgelegt. Zur Zuordnung der Werte an die Position in der Matrix werden drei weitere Arrays benötigt. Das verwendete Format zur Speicherung der dünnbesetzten Matrix wird CSR (compressed sparse row, komprimierte dünnbesetzte Zeile) genannt.

Der nachfolgende Vergleich zeigt den Unterschied zwischen der Verwendung einer vollbesetzten Matrix gegenüber einer dünnbesetzten Matrix mit dem CSR Format. Als Beispiel wird die in Abbildung 4.15 (a) gezeigte Matrix verwendet. Für die vollbesetzte Matrix ergibt sich ein Speicherbedarf von 705.600Bytes , für die selbe Information werden bei der Verwendung des CSR Formats 14.008Bytes verwendet. Dies entspricht ungefähr einer Reduktion des Speicherbedarfs um den Faktor 50.

Für die weitere Implementierung sei auf die Veröffentlichung von Donald Goldfarb und Ashok Idnani [39] verwiesen. Darin wird auch auf die Verwendung von Update-Techniken bei der QR Zerlegung eingegangen, die bei der Implementierung Anwendung finden. Dies hat den Vorteil, dass beim Hinzufügen, oder Entfernen von einer Spalte, die QR Zerlegung nicht in jedem Schritt komplett neu berechnet werden muss.

Abbildung 4.16 zeigt einen Vergleich der Laufzeiten zwischen der speziell auf das Problem angepassten Implementierung und einer allgemeinen Implementierung mit vollbesetzten Matrizen. Für den allgemeinen Ansatz wurde die Implementierung von Luca Di Gasparo [57] verwendet. Mit steigender Anzahl an Gitterpunkten (steigende Problemgröße), was einer Vergrößerung der Elementanzahl in einer Spalte der quadratischen Matrix Q bedeutet, wird der Vorteil der angepassten Implementierung deutlich.

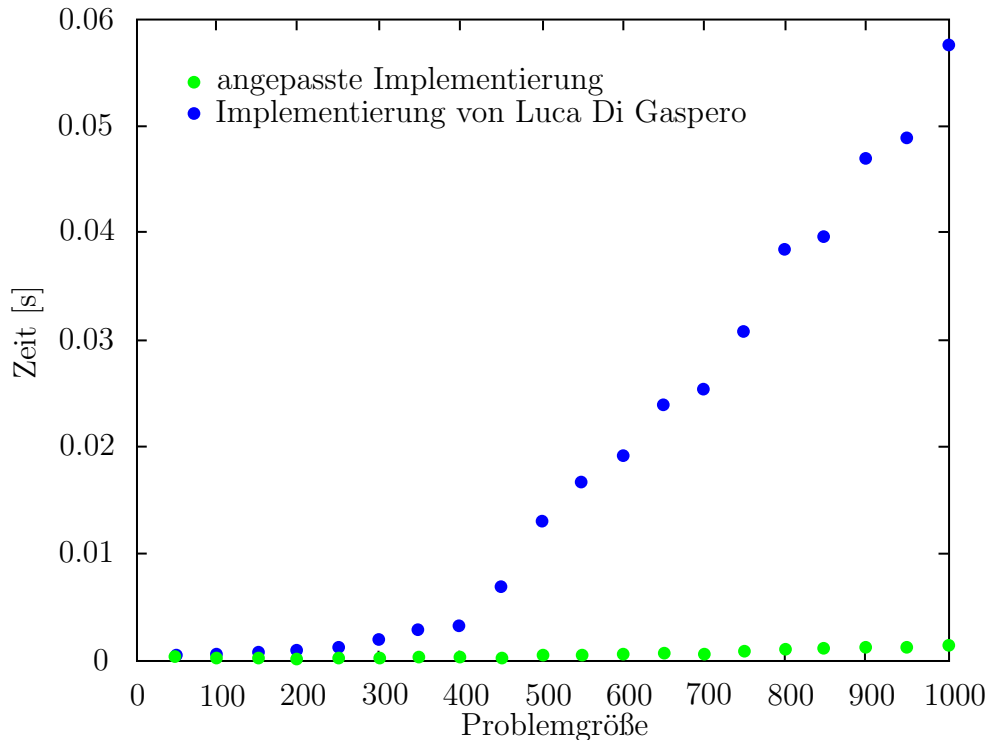


Abbildung 4.16: Vergleich zwischen der angepassten Implementierung des Goldfarb und Idnani Ansatzes (grün) mit einer allgemeinen Implementierung von Luca Di Gaspero [57] (blau).

4.7.2.2 Parallelisierte Implementierung auf einer Grafikkarte

Moderne Grafikkarten bieten die Möglichkeit die vorhandenen Rechenkerne für die parallele Verarbeitung von Aufgaben zu nutzen. Das Unternehmen NVIDIA Corporation bietet für seine Grafikkarten eine spezielle Softwareschnittstelle mit dem Namen CUDA ([67]), um auf der Grafikkarte parallelisiert zu rechnen.

Abbildung 4.17 zeigt einen Vergleich der Laufzeiten der angepassten Implementierung des Goldfarb und Idanai Algorithmus, welche auf der CPU (Central Processing Unit) berechnet wird, gegenüber einer Portierung des Algorithmus auf eine GPU (Graphics Processing Unit). Für den Vergleich wurde ein Intel Core 2 Quad mit $2.66GHz$ und $4GB$ Hauptspeicher, sowie eine NVIDIA Tesla C2050 Grafikkarte mit 448 Kernen und $3GB$ Speicher verwendet. Es zeigt sich, dass bei steigenden Problemgrößen die Implementierung auf der Grafikkarte bezüglich der Laufzeit klare Vorteile aufweist. Vergleicht man die Laufzeiten bei den typischen Problemgrößen für diesen Anwendungsfall (siehe Abbildung 4.18), erweist sich die Berechnung auf der CPU als die schnellere Variante. Erklären lässt sich dies durch die langsamere Speicheranbindung der Grafikkarte. Bei kleinen Problemgrößen wird dabei die meiste Zeit für den Transfer der Daten in den Speicher verwendet. Die parallele Implementierung kann dabei den Zeitverlust bei größeren Problemen durch eine schnelle Verarbeitung kompensieren.

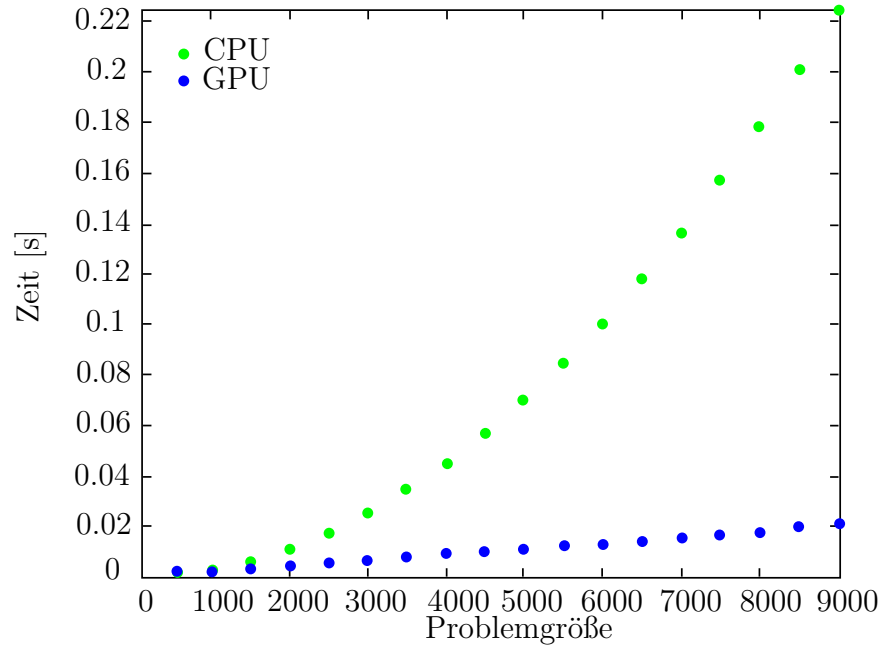


Abbildung 4.17: Vergleich zwischen der Implementierung auf der CPU (grün) mit der Implementierung auf der GPU (blau)

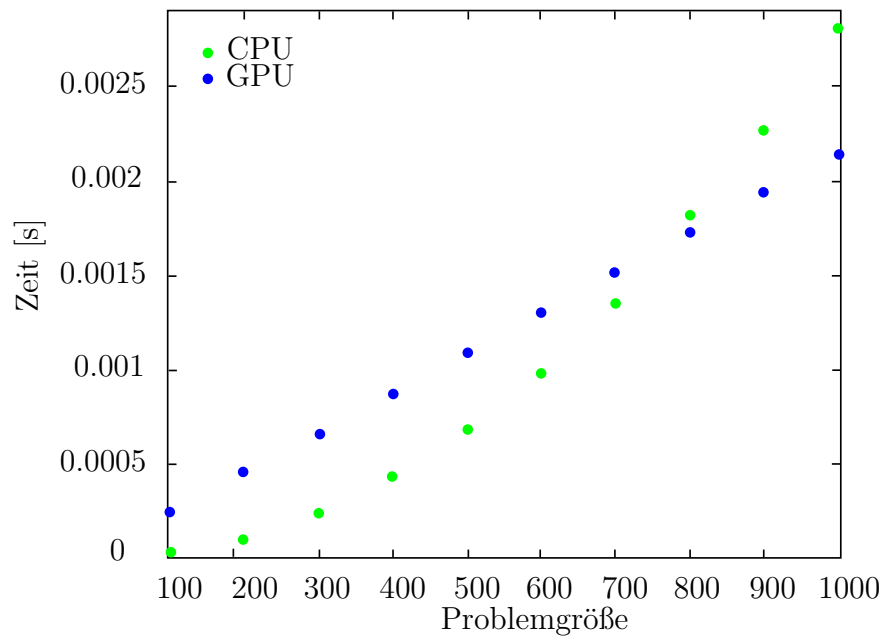


Abbildung 4.18: Vergrößerter Ausschnitt aus Abbildung 4.17. Vergleich zwischen der Implementierung auf der CPU (grün) mit der Implementierung auf der GPU (blau)

4.8 Flächeninterpolation

Aus Sicht der Berechnungsdauer und der Dichte der Punktmenge P_R ist es nicht sinnvoll die Abstände der Knotenpunkte beliebig zu verringern um eine feinere Auflösung der Außenkontur zu erhalten. Es empfiehlt sich daher zwischen den Knotenpunkten mit einer bilinearen Interpolation zu interpolieren. Die neu entstandenen Knotenpunkte, einschließlich der schon vorhandenen, werden in der Punktmenge K_I zusammengefasst. Abbildung 4.19 zeigt den Übergang von nicht interpolierten zu interpolierten Knotenpunkten.

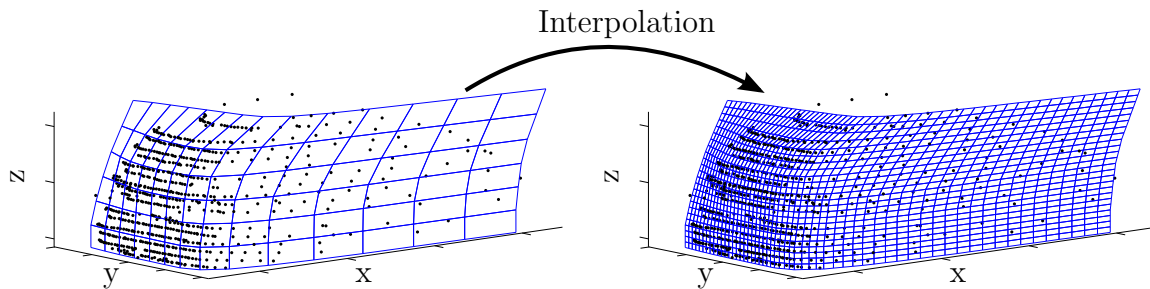


Abbildung 4.19: Übergang zwischen Knotenpunkten und interpolierten Knotenpunkten

4.9 Konfidenzbestimmung

In '4.7 Lösung des Optimierungsproblems' wurde die Fläche berechnet, die die Außenkontur eines Fahrzeugs repräsentiert. Dabei liegt noch keine Information über die Vertrauenswürdigkeit der einzelnen Knotenpunkte, die die Außenkontur bestimmen, vor. Im Folgenden wird ein Konfidenzmaß vorgestellt, das Auskunft über die Vertrauenswürdigkeit der Knotenpunkte liefert. Das Konfidenzmaß setzt sich aus zwei Komponenten zusammen. Zum einen beachtet es, wie gut die Außenkontur den vorgegebenen Flächeneigenschaften folgt ('4.9.1 Flächenkonfidenz'), zum anderen fließt die Anzahl der vorhandenen Messwerte, die den betreffenden Knotenpunkt gestützt haben, mit ein. Beide Komponenten werden zu gleichen Teilen gewichtet und werden jeweils im Intervall $[0, 0.5]$ definiert. Die Summe aus beiden Komponenten ergibt das Konfidenzmaß, welches im Intervall $[0, 1]$ definiert ist.

4.9.1 Flächenkonfidenz

Die Flächenkonfidenz beschreibt über einen Distanzwert, wie gut der entsprechende Knoten zu dem geforderten Wert aus dem Gleichungssystem passt (Gleichung 4.42). Er ist definiert als $\vec{e}_k \in \mathbb{R}^{N_g}$ und ergibt sich durch Umformen von Gleichung 4.42 zu

$$\vec{e}_k = \mathbf{A}^T \vec{y} - \mathbf{A}^T \mathbf{A} \vec{v} \quad (4.46)$$

Vektor \vec{e}_k lässt sich auf das Intervall $[0, 0.5]$ skalieren, was zum Vektor $e_{kn}^{\vec{}}$ führt. Je kleiner der Wert, desto weniger vertrauensvoll ist der Knotenpunkt. Die Elemente des Vektors ergeben sich wie folgt

$$e_{kni} = -\frac{e_{ki} - \text{MIN}(\vec{e}_k)}{2(\text{MAX}(\vec{e}_k) - \text{MIN}(\vec{e}_k))} + \frac{1}{2} \quad (4.47)$$

für $i \in \{1, \dots, N_g\}$

Der Wert für die Flächenkonfidenz eines jeden Knotens ergibt sich beispielhaft für die Punktmenge aus Abbildung 4.14 wie in Abbildung 4.20 dargestellt. Die Nummerierung der Knoten entspricht der Definition in Abbildung 4.8. Es zeigt sich, dass der Mittelwert der Konfidenz bei ca. 0.2 liegt, was bedeutet, dass die Konfidenz ausgeglichen ist. In der zweiten Hälfte in Abbildung 4.20 treten stärkere Schwankungen in der Konfidenz auf. Dies ist das Heck des Fahrzeugs in diesem Beispiel. Hier sind viele Messungen vorhanden, die in die Flächenbestimmung einfließen. Zur Straße und zum Dach des Fahrzeugs werden die Messungen weniger und die Bestimmung der Fläche muss allein mit Hilfe der Flächeneigenschaften bestimmt werden. Die Fläche wird jedoch noch durch die hohe Anzahl an Messpunkten zwischen Straße und Heck und der dort erfüllten Flächeneigenschaften bestimmt. Dies hat eine Senkung der Konfidenz zur Folge, da die Annahmen nicht voll erfüllt werden können. Abbildung 4.23 zeigt zwar bereits die kombinierte Konfidenz, der eben beschriebene Bereich ist dennoch deutlich zu erkennen.

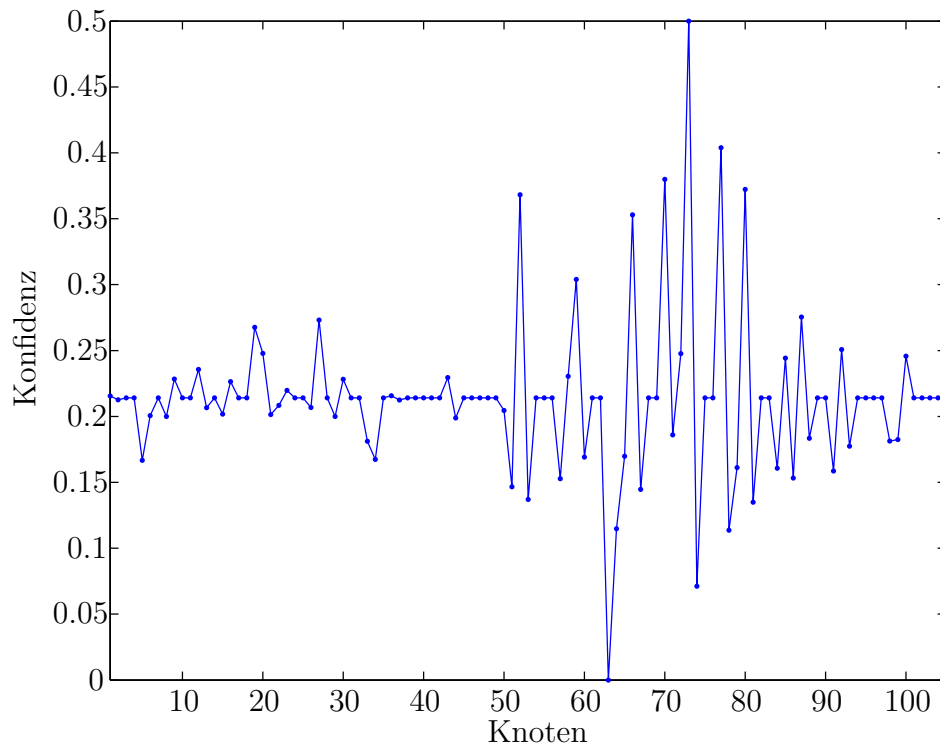


Abbildung 4.20: Beispielhafte Darstellung der Werte für Flächenkonfidenz eines jeden Knotens

Vektor \vec{e}_k lässt sich ebenfalls entsprechend der Flächeninterpolation ('4.8 Flächeninterpolation') auf die Zwischenknotenpunkte interpolieren. Auf das Intervall $[0, 0.5]$ interpoliert ergibt sich der Vektor e_{I_n} , dessen Elemente sich wie folgt ergeben

$$e_{Ini} = -\frac{e_{Ii} - MIN(\vec{e}_I)}{2(MAX(\vec{e}_I) - MIN(\vec{e}_I))} + \frac{1}{2} \quad (4.48)$$

4.9.2 Nachbarschaft

Die Nachbarschaft eines Knotenpunktes wird definiert über die Anzahl der Messungen (Punkte aus der Punktmenge P_R) in einem Abstand d_n um den Knotenpunkt herum. Als Abstandsmaß wird der euklidische Abstand genutzt. Befinden sich Messungen im Abstand d_n , wird die Fläche in dieser Region durch eben diese gestützt. Es ergibt sich ein Maß für eine höhere Vertrauenswürdigkeit der Fläche. Befinden sich jedoch wenige bis keine Messungen im Abstand d_n , wurde die Fläche in dieser Region allein über die Flächeneigenschaften aus Kapitel '4.5 Flächeneigenschaften' berechnet. Dies ist weniger vertrauensvoll und es ergibt sich ein geringeres Maß für die Vertrauenswürdigkeit.

Vektor $\vec{n}_m \in \mathbb{R}^{N_g}$ enthält für jeden Knotenpunkt die Anzahl der benachbarten Messungen im Abstand d_n . Über einen Wert N_N lässt sich definieren, ab welcher Anzahl an Messungen in der Nachbarschaft der Knoten volle Vertrauenswürdigkeit erhalten soll. Die Funktion zur Berechnung ist in Gleichung 4.49 definiert. Der Ergebnisvektor ist beschrieben durch \vec{s}_n und ist auf dem Intervall $[0, 0.5]$ definiert. Je kleiner der Wert, desto weniger vertrauensvoll ist der Knotenpunkt.

$$s_{n_i} = \begin{cases} \frac{1}{2N_N} n_{m_i} & \text{wenn } n_{m_i} \leq N_N \\ \frac{1}{2} & \text{sonst} \end{cases} \quad (4.49)$$

für $i \in \{0, \dots, N_g\}$.

Der Wert für die Konfidenz eines jeden Knotens entsprechend der Messungen in der Nachbarschaft ergibt sich beispielhaft für die Punktmenge aus Abbildung 4.14 wie in Abbildung 4.21 dargestellt.

Der linke Bereich in Abbildung 4.21 repräsentiert die Fahrzeugseite. Hier sind weniger Messungen auf dem Fahrzeug vorhanden. Dies zeigt sich in der Konfidenzbestimmung mit geringen Werten von bis zu Null. Der rechte Bereich stellt das Heck des Fahrzeugs dar. Hier treten viele Messungen in der Nachbarschaft auf, was sich mit der höchsten Konfidenz bemerkbar macht. Die in Abbildung 4.21 deutlich erkennbare Oszillation zwischen maximaler und minimaler Konfidenz liegt an der Nummerierung der Knotenpunkte, wie sie bereits in Abbildung 4.8 beschrieben wurde. Da diese entlang der z-Achse erfolgt, durchläuft sie jedes Mal den Bereich zwischen Straße und Dach des Fahrzeugs. Auf Höhe der Straße und am Dach liegen weniger Punkte vor als direkt auf dem Heck. Darum entsteht die Oszillation.

Vektor \vec{s}_n lässt sich ebenfalls entsprechend der Flächeninterpolation ('4.8 Flächeninterpolation') auf die Zwischenknotenpunkte interpolieren. Es ergibt sich der Vektor $s_{nI}^{\vec{}}$.

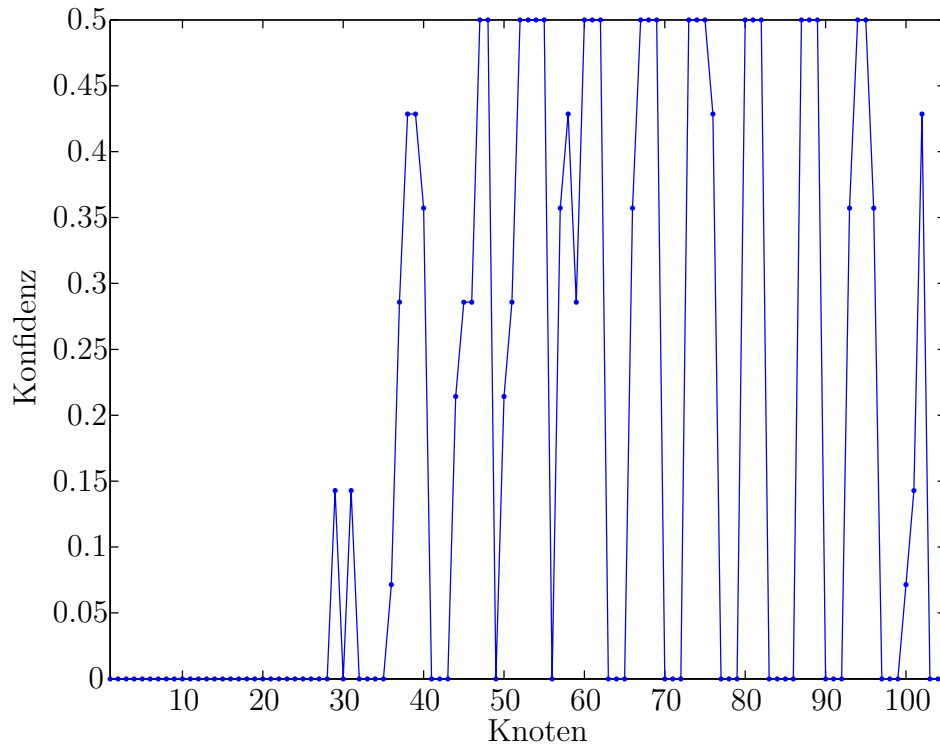


Abbildung 4.21: Beispielhafte Darstellung der Werte für die Konfidenz eines jeden Knotens entsprechend der Messungen in der Nachbarschaft

4.9.3 Konfidenzmaß

Das Konfidenzmaß ergibt sich durch die Kombination der beiden Komponenten aus Kapitel '4.9.1 Flächenkonfidenz' und Kapitel '4.9.2 Nachbarschaft'. Beide werden durch eine Addition mit einander kombiniert. Es ergibt sich der Konfidenzvektor \vec{k} im Intervall $[0, 1]$ zu

$$\vec{k} = e_{kn}^{\vec{}} + \vec{s}_n \quad (4.50)$$

Der Wert für die Konfidenz eines jeden Knotens ergibt sich beispielhaft für die Punktmenge aus Abbildung 4.4 wie in Abbildung 4.22 dargestellt.

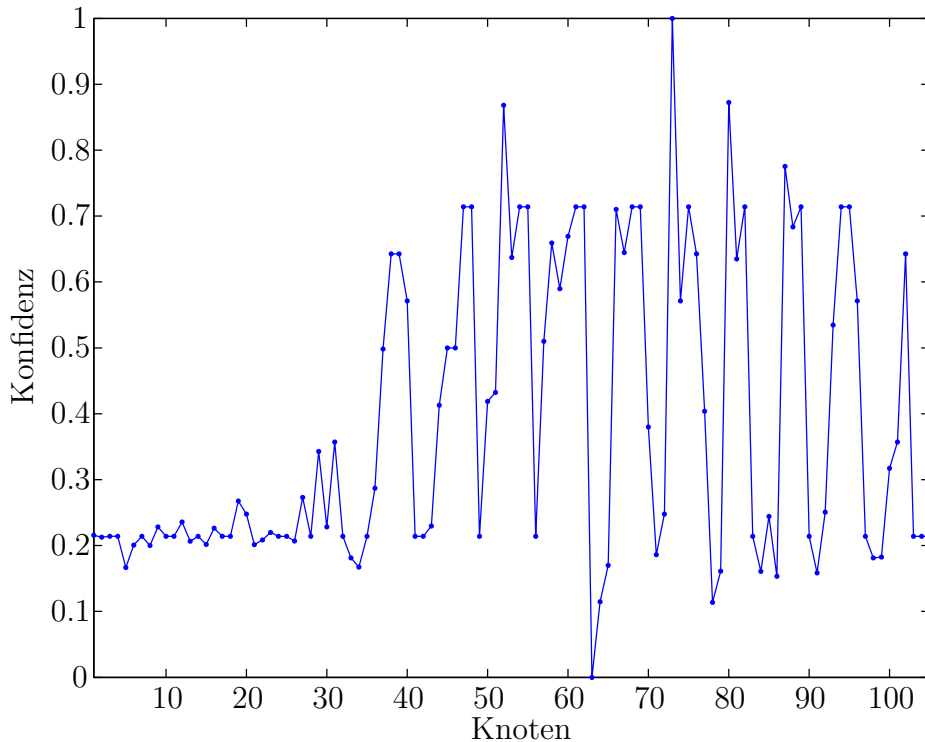


Abbildung 4.22: Beispielhafte Darstellung der Werte für Konfidenz eines jeden Knotens

Für die interpolierten Knotenpunkte gilt entsprechend der interpolierte Konfidenzvektor

$$\vec{k}_I = e_{In}^{\vec{}} + s_{nI}^{\vec{}} \quad (4.51)$$

Jedem Knotenpunkt, sei es ein interpolierter oder ein nicht interpolierter, wird somit ein Konfidenzwert zugewiesen. Je größer der Konfidenzwert, desto höher ist die Vertrauenswürdigkeit des Knotenpunktes.

Im weiteren Verlauf des Kapitels soll zur übersichtlicheren Darstellung nur noch auf die interpolierten Knotenpunkte K_I eingegangen werden.

Die Punktmenge der Knotenpunkte K_I wird um eine weitere Dimension für das Konfidenzmaß erweitert. Es ergibt sich eine neue Menge K_{IK} .

Abbildung 4.23 zeigt die Konfidenzwerte für die Knotenpunkte. Die Konfidenzwerte sind entsprechend ihrem Wert farblich kodiert. Der Farbwert blau zeigt eine geringe Vertrauenswürdigkeit des Knotens, wohingegen der Farbwert rot eine hohe Vertrauenswürdigkeit des Knotens anzeigt. Die Farbgebung im Zusammenhang mit der Vertrauenswürdigkeit wird bei folgenden Schaubildern beibehalten. In Abbildung 4.23 zeigt sich an den Rändern eine geringe Vertrauenswürdigkeit. Hier liegen wenige, bis keine Messungen und die Knotenpunkte werden nur über die Flächeneigenschaften ('4.5 Flächeneigenschaften') bestimmt. Der stark von hoher Vertrauenswürdigkeit dominierte Bereich (rot) ist das Heck

des Fahrzeugs. Entsprechend dem Aspektwinkel unter dem der Sensor die Messung vorgenommen hat, liegen hier die meisten Messpunkte. Dieser Bereich erzielt einen hohen Konfidenzwert bezüglich der Nachbarschaft aus '4.9.2 Nachbarschaft'. Zusätzlich passen auch die Flächeneigenschaften zu den Messungen. Unter dem eben beschriebenen Bereich liegt ein Gebiet mit geringer Vertrauenswürdigkeit. Hier passen die Flächeneigenschaften nicht ausreichend gut zur bestimmten Fläche, da ein hohes Rauschen in den Messungen vorliegt. Dieses Rauschen entsteht durch sich abwechselnde Messungen des Bodens und des Fahrzeugs. Entlang der positiven x-Achse nimmt die Vertrauenswürdigkeit der Fläche ab. Durch die geringer werdende Anzahl an Messungen wird die Fläche mehr und mehr nur noch über die Flächeneigenschaften bestimmt.

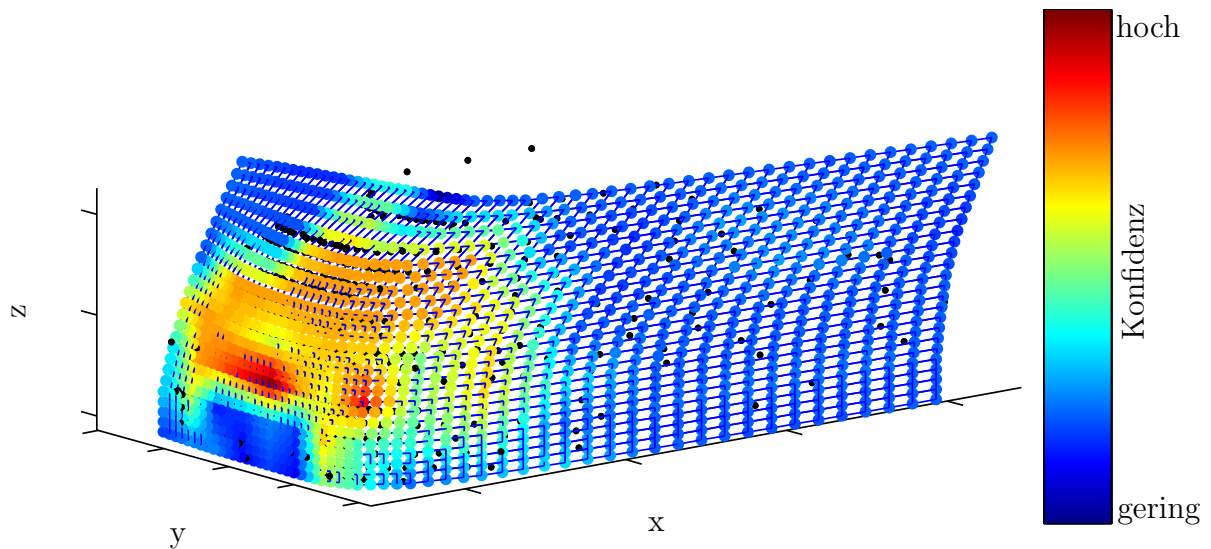


Abbildung 4.23: Farbliche Kodierung der Knotenpunkte entsprechend ihren Konfidenzwerten

4.9.4 Zuschneiden der Fläche

Im Allgemeinen füllen die Messungen die Zellen an den Rändern der Flächenfunktion nicht aus. Hier wird die Fläche daher allein über die in Kapitel '4.5 Flächeneigenschaften' definierten Eigenschaften bestimmt. Dies führt jedoch in den meisten Fällen zu falschen Extrapolationen. Daher wird die Fläche am Rand entsprechend zugeschnitten.

Zunächst wird die Punktmenge P_R über die Funktion $P_{xz}(\cdot)$ in die xz -Ebene projiziert. Die im zweidimensionalen definierte Funktion der konvexen Hülle $conv(\cdot)$ wird auf die projizierten Punkte angewendet. Es ergibt sich demnach die konvexe Hülle zu

$$C_H = conv(P_{xz}(P_R)) \quad (4.52)$$

Mit der Funktion $inConv(C_H, K_{IK})$ werden diejenigen Knotenpunkte bestimmt, die sich innerhalb der konvexen Hülle C_H befinden.

$$K_{IKC} = inConv(C_H, K_{IK}) \quad (4.53)$$

In Abbildung 4.24 ist das Ergebnis der Anwendung der konvexen Hülle für das Zuschneiden der Fläche dargestellt. Aufgrund der perspektivischen Sicht auf die zugeschnittene Außenkontur und die Messpunkte entsteht der Eindruck, dass einige Messpunkte außerhalb der Außenkontur liegen würden, tatsächlich befinden sie sich jedoch auf dem Fahrzeugdach und dadurch tiefer in der Bildebene.

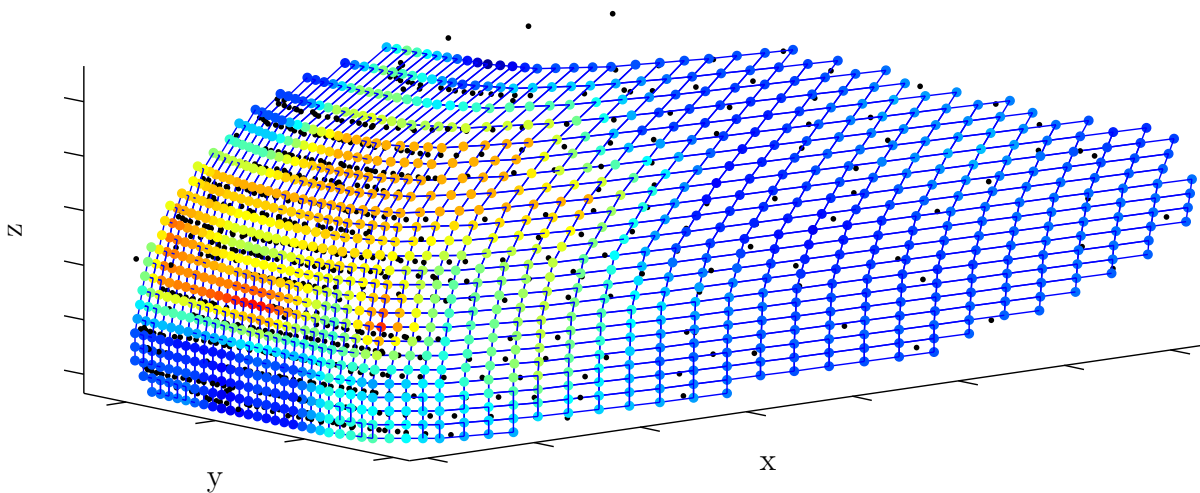


Abbildung 4.24: Zugeschnittene Fläche über die konvexe Hülle

In allen weiteren Grafiken zur Außenkontur wird diese über die vorgestellte Methode zugeschnitten.

4.10 Nachverarbeitungsschritt

Die in Kapitel '4.3 Vorverarbeitungsschritt' durchgeführte Translation und Rotation der Punktmenge P , die zur Punktmenge P_R geführt hat, hat entsprechende Auswirkungen auf die berechnete Außenkontur. Die Außenkontur gilt für die Punktmenge P_R . Um sie allerdings für die Punktmenge P zu erhalten, muss die Translation ($T_1(\cdot)$) und Rotation ($R_z(\cdot)$) in umgekehrter Reihenfolge auf die Knotenpunkte aus K_{IKC} angewendet werden. Die Umkehrung der Translation $T_1(\cdot)$ wird als $T_1^{-1}(\cdot)$ definiert. Für K_{IKC} gilt

$$K_{IKCN} = R_z(T_1^{-1}(K_{IKC}), -\alpha_R) + \vec{c}_p \quad (4.54)$$

4.11 Robustheit der Flächenbestimmung

Um die Robustheit der Flächenbestimmung gegenüber Veränderungen in den Messdaten zu zeigen, werden im Folgenden drei Fälle untersucht. Die Ausgangsmessung ist in Abbildung 4.25 dargestellt. Für die Untersuchung wird jeweils die Ausgangsmessung künstlich verändert, um verschiedene Szenarien darzustellen.

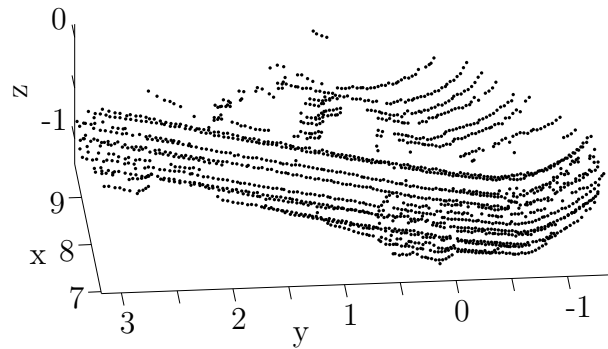
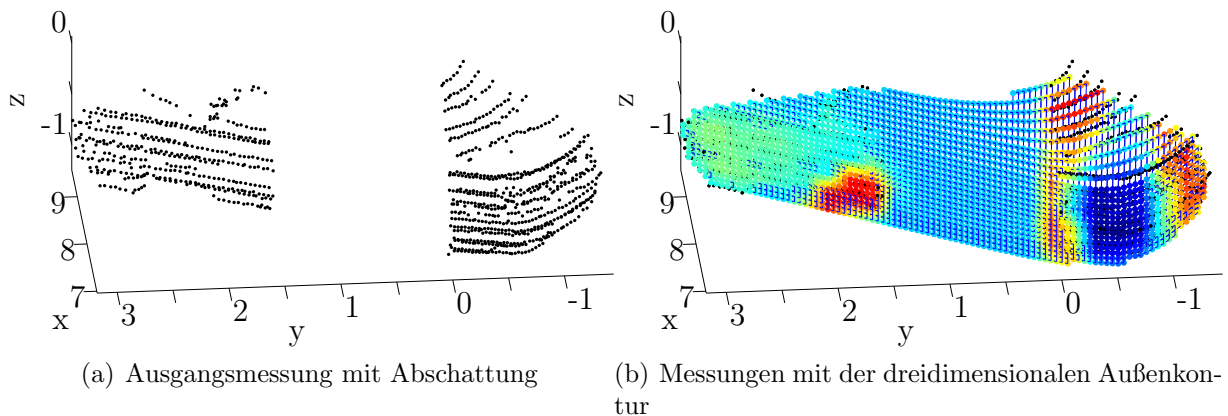


Abbildung 4.25: Ausgangsmessung für die Untersuchung der Robustheit.

4.11.1 Fehlende Messungen

Fehlende Messungen auf einem Objekt können beispielsweise die Folge von Abschattungen durch andere Objekte sein. Abbildung 4.26 zeigt, dass fehlende Messungen die Flächenbestimmung wenig beeinträchtigen. Zu erkennen ist, dass die Konfidenz an den Stellen mit fehlenden Messungen herabgesetzt wird.



(a) Ausgangsmessung mit Abschattung

(b) Messungen mit der dreidimensionalen Außenkontur

Abbildung 4.26: Auswirkungen von fehlenden Messungen auf die Flächenbestimmung

4.11.2 Rauschen

Rauschen in den Messungen kann beispielsweise durch Wettereinflüsse hervorgerufen werden. Abbildung 4.27 zeigt, dass rauschbehaftete Messungen die Flächenbestimmung wenig beeinträchtigen. Für das Rauschen wurde jeweils eine Normalverteilung entsprechend $\mathcal{N}_x(0, 0.1^2 m^2)$, $\mathcal{N}_y(0, 0.1^2 m^2)$, $\mathcal{N}_z(0, 0.1^2 m^2)$ angenommen.

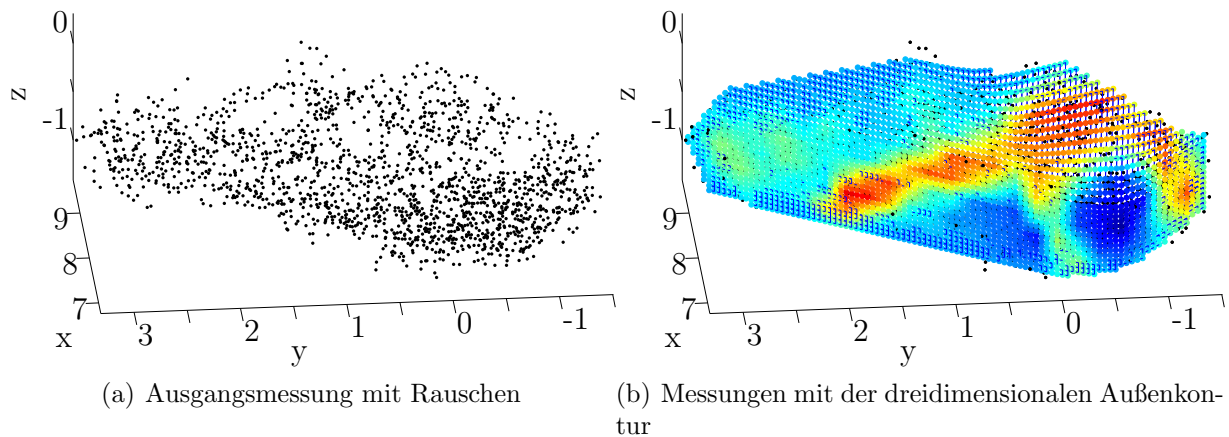


Abbildung 4.27: Auswirkungen von rauschbehafteten Messungen auf die Flächenbestimmung

4.11.3 Ausreißer

Ausreißerdaten, die beispielsweise durch fehlerhafte Segmentierung entstehen können, wirken sich, wie in Abbildung 4.28 dargestellt ist, wenig auf die Flächenbestimmung aus. Es ist allerdings zu beachten, dass ab einem bestimmten Verhältnis zwischen wirklichen Messungen und Ausreißerdaten die Flächenbestimmung nicht mehr möglich ist. Der gezeigte Fall stellt ein extremes Beispiel dar, welches in dieser Art prinzipiell nicht vorkommen kann, da die Punktwolken durch die Segmentierung getrennt werden müssten.

4.11.4 Zusammenfassung

Die Flächenbestimmung stellt ein robustes Verfahren dar, um die dreidimensionale Außenkontur von Fahrzeugen zu bestimmen. Das Verfahren kann fehlende Messungen mit Hilfe der definierten Flächeneigenschaften kompensieren. Durch die dem Algorithmus zu Grunde liegende Ausgleichsrechnung können durch Rauschen gestörte Messungen und größere Ausreißer geglättet werden. Somit lassen sich trotz nicht optimaler Eingangsdaten Außenkonturen berechnen, die bei Betrachtung durch einen Menschen, zu erwarten sind.

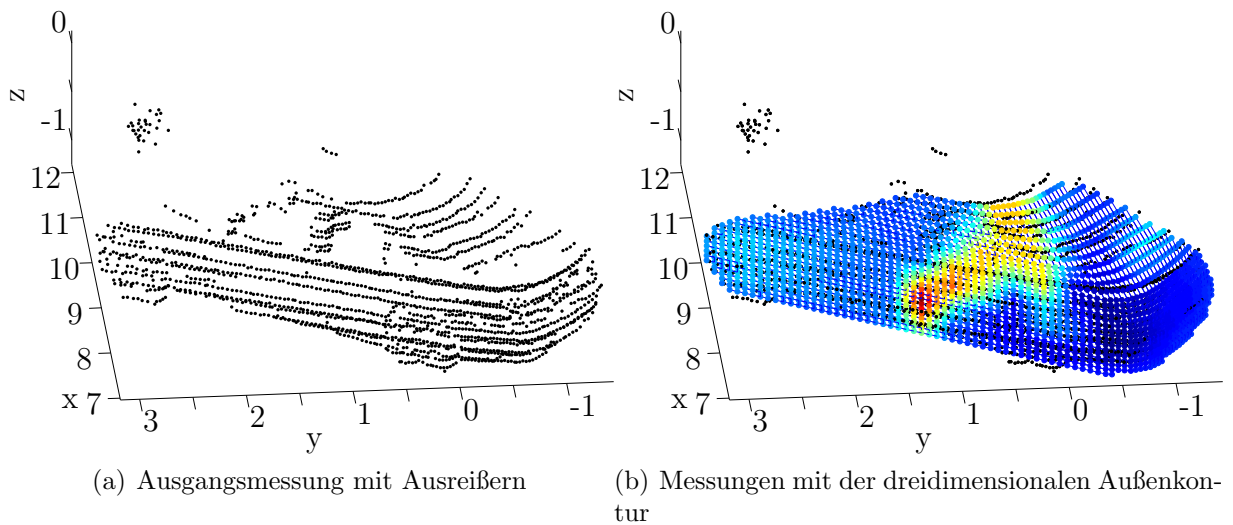


Abbildung 4.28: Auswirkungen von Ausreißerdaten in den Messungen auf die Flächenbestimmung

4.12 Bestimmung der zweidimensionalen Außenkontur

In manchen Anwendungen wird lediglich eine zweidimensionale statt der hier berechneten dreidimensionalen Außenkontur benötigt. Ein Beispiel stellt die Objektverfolgung dar, für die lediglich die äußerste Kontur von Interesse ist. Mit der Verwendung einer zweidimensionalen Außenkontur lässt sich in den nachfolgenden Algorithmen auch Rechenzeit einsparen. Die berechnete dreidimensionale lässt sich auf eine zweidimensionale Außenkontur reduzieren. Der Vorteil dieser Reduktion liegt darin, dass für die Berechnung der dreidimensionalen Außenkontur alle vorhandenen Informationen, die die Messung zur Verfügung gestellt hat, eingeflossen sind. Bei der Reduktion bleiben diese Informationen in der zweidimensionalen Außenkontur enthalten. Die Reduktion der dreidimensionalen auf die zweidimensionale Außenkontur erfolgt durch eine Projektion der Knotenpunkte in die xy -Ebene mit anschließender Bestimmung der konvexen Hülle.

Für die Berechnung der zweidimensionalen Außenkontur werden die Knotenpunkte der dreidimensionalen Außenkontur in die xy -Ebene über die Funktion $P_{xy}(\cdot)$ projiziert. Auf die projizierten Knotenpunkte wird die im zweidimensionalen Raum definierte konvexe Hülle $\text{conv}(\cdot)$ angewendet. Diese ergibt sich zu

$$C_H = \text{conv}(P_{xy}(K_{IKCN})) \quad (4.55)$$

Die Menge der Punkte $C_H \subset K_{IKCN}$ definiert demnach die Randpunkte der konvexen Hülle der Punktmenge K_{IKCN} . Diese beschreibt einen geschlossenen Graphen in der xy -Ebene. Die Punkte in C_H liegen in einer derartigen Sortierung vor, dass die Verbindung der Punkte nacheinander den Graphen ergeben. Für die Bestimmung der zweidimensionalen Außenkontur wird nur diejenige Seite des Graphen herangezogen, die auch durch

das Messprinzip des eingesetzten Sensors sichtbar ist. Die in kartesischen Koordinaten vorliegenden Punkte C_H der konvexen Hülle werden in Polarkoordinaten über die Funktion $pol(\cdot)$ umgerechnet.

$$C_{HP} = pol(C_H) \quad (4.56)$$

Über die Funktion $MIN_\varphi(\cdot)$ lässt sich der minimale Azimuth-Winkel in einer Punktmenge bestimmen. Entsprechendes gilt für die Funktion $MAX_\varphi(\cdot)$, die den maximalen Azimuth-Winkel in einer Punktmenge bestimmt. Es gilt

$$\varphi_{min} = MIN_\varphi(C_{HP}) \quad (4.57)$$

$$\varphi_{max} = MAX_\varphi(C_{HP}) \quad (4.58)$$

Nach Rücktransformation der beiden Punkte mit dem kleinsten Winkel φ_{min} und dem größten Winkel φ_{max} in kartesische Koordinaten teilt die Gerade zwischen beiden Punkten den Graphen in zwei Bereiche. Der eine Bereich liegt dem Sensor zu-, der andere dem Sensor abgewandt. Die beiden Bereiche lassen sich durch die Punktmenge $B_1 \subset C_H$ und $B_2 \subset C_H$ beschreiben, welche jeweils einen neuen Graphen definieren. Abbildung 4.29 zeigt ein Beispiel. Dabei ist die Gerade, die den Graphen teilt, in grün, die Fläche innerhalb des Graphen B_1 in gelb und die Fläche innerhalb des Graphen B_2 in hellblau dargestellt.

Die Bereiche B_1 und B_2 entstehen durch die vorhandenen Dachpunkte und die Messungen, die an der Seite des Fahrzeugs wenige cm hinter der Außenkontur, aber weiter entfernt vom Sensor liegen.

Für die Unterscheidung der beiden Bereiche werden zunächst pro Bereich B_1 und B_2 die Schwerpunkte über das arithmetische Mittel \vec{m}_{B_1} und \vec{m}_{B_2} in kartesischen Koordinaten bestimmt. Als relevanter Bereich, der die Außenkontur definiert, wird derjenige ausgewählt, dessen Abstand zwischen arithmetischem Mittel und Sensor am geringsten ist. Es gilt für die zweidimensionale Außenkontur C_{2D}

$$C_{2D} = \begin{cases} B_1 & \text{wenn } \|\vec{m}_{B_1}\|_2 < \|\vec{m}_{B_2}\|_2 \\ B_2 & \text{sonst} \end{cases} \quad (4.59)$$

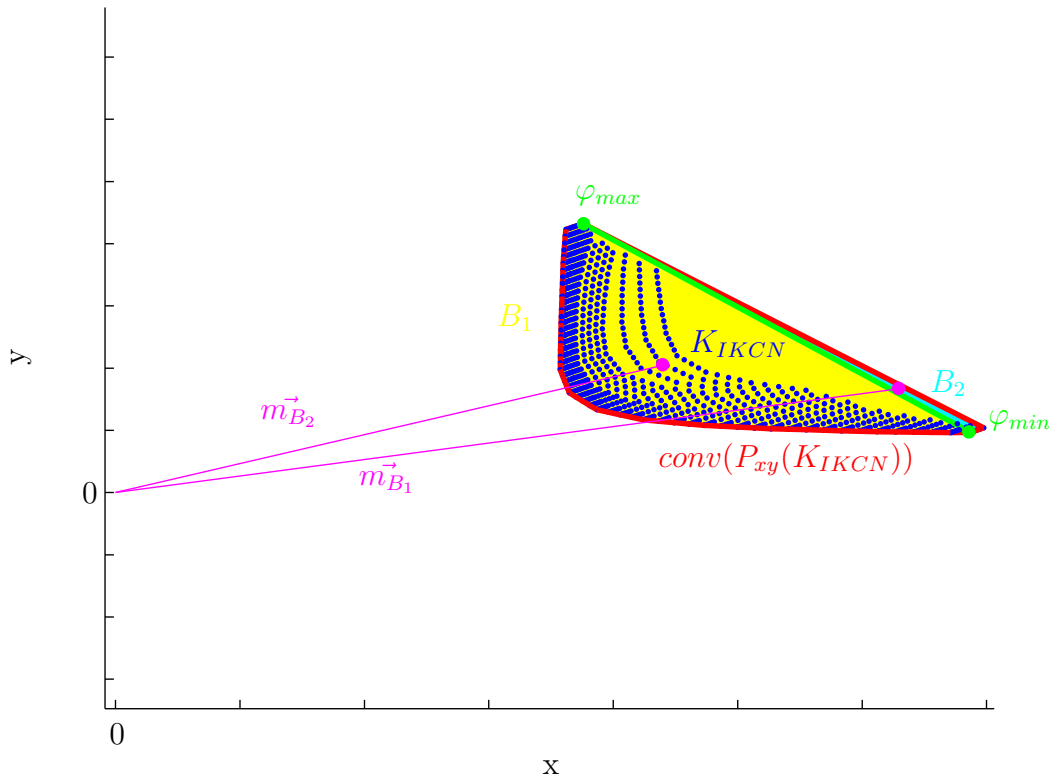


Abbildung 4.29: Vorgehensweise zur Bestimmung der zweidimensionalen Außenkontur

In Abbildung 4.30 ist das Ergebnis der Reduktion der dreidimensionalen auf eine zweidimensionale Außenkontur dargestellt. Die z -Position der zweidimensionalen Außenkontur ergibt sich über den minimalen z -Wert in der Punktmenge der Messungen P .

Die Abbildung 4.31 zeigt Messungen eines PKW, dessen Heck in den Messungen sichtbar ist. Für die Messungen ist eine zweidimensionale Außenkontur bestimmt und in rot eingezeichnet worden. Die Abbildung zeigt zwei unterschiedliche Perspektiven auf die Messdaten mit der Außenkontur.

Abbildung 4.32 zeigt Messungen eines Transporters, dessen Heck und Seitenfläche in den Messungen sichtbar sind. Zusätzlich sind Messungen aus dem Inneren des Transporters sichtbar, da der Lasersensor durch die Heckscheibe gemessen hat.

In Abbildung 4.33 ist derselbe Transporter zu einem späteren Zeitpunkt zu sehen. Dabei ist nur noch das Heck in den Messungen sichtbar.

Die vorgestellten Ergebnisse zur Bestimmung der zweidimensionalen Außenkontur zeigen, dass das Verfahren mit unterschiedlichen Fahrzeugformen und auch starken Krümmungsänderungen, wie beim Transporter gezeigt, zurecht kommt. Die Lage der Kontur entspricht der, die auch bei der Markierung von Hand durch einen Menschen erwartet worden wäre. Eine detaillierte Auswertung der Abweichungen zwischen der wirklichen und der berechneten Außenkontur findet sich in '4.14 Vergleich der zweidimensionalen

Außenkontur mit der realen Fahrzeugkontur'.

Zu beachten ist, dass für die Bestimmung der Außenkontur des LKW und des PKW keine Parameter geändert wurden.

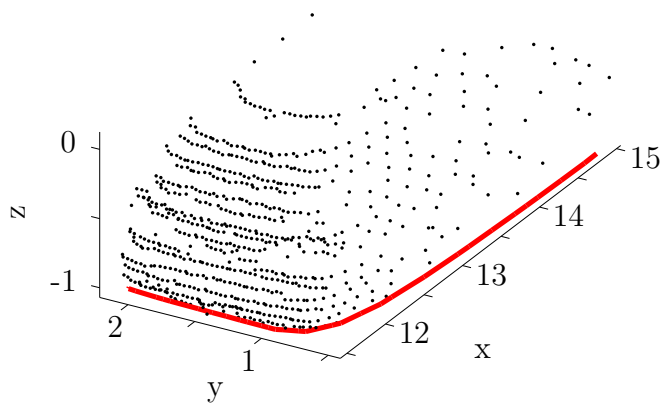
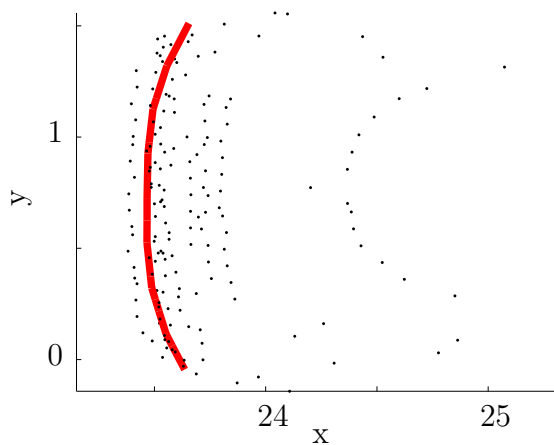
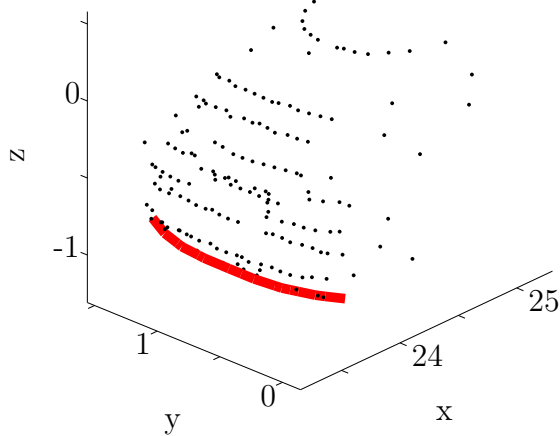


Abbildung 4.30: Zweidimensionale Außenkontur mit 3D Punktmessungen



(a) Sicht aus Vogelperspektive



(b) Seitenansicht

Abbildung 4.31: Zweidimensionale Außenkontur mit 3D Punktmessungen eines PKW, dessen Heck in den Messungen sichtbar ist

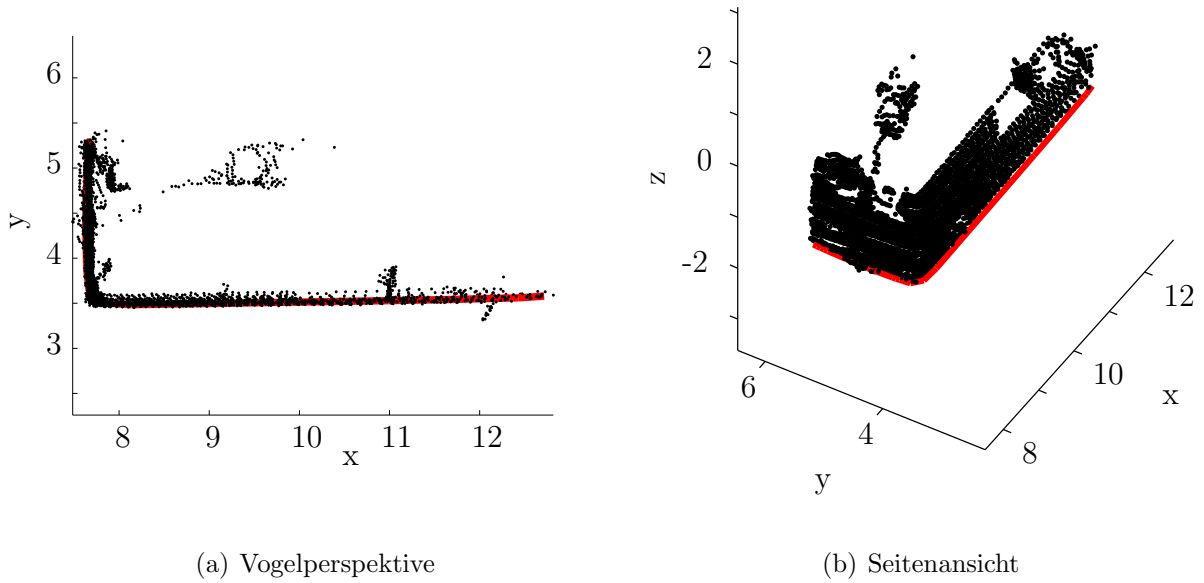


Abbildung 4.32: Zweidimensionale Außenkontur mit 3D Punktmessungen eines Transporters, dessen Heck und Seitenfläche in den Messungen sichtbar sind

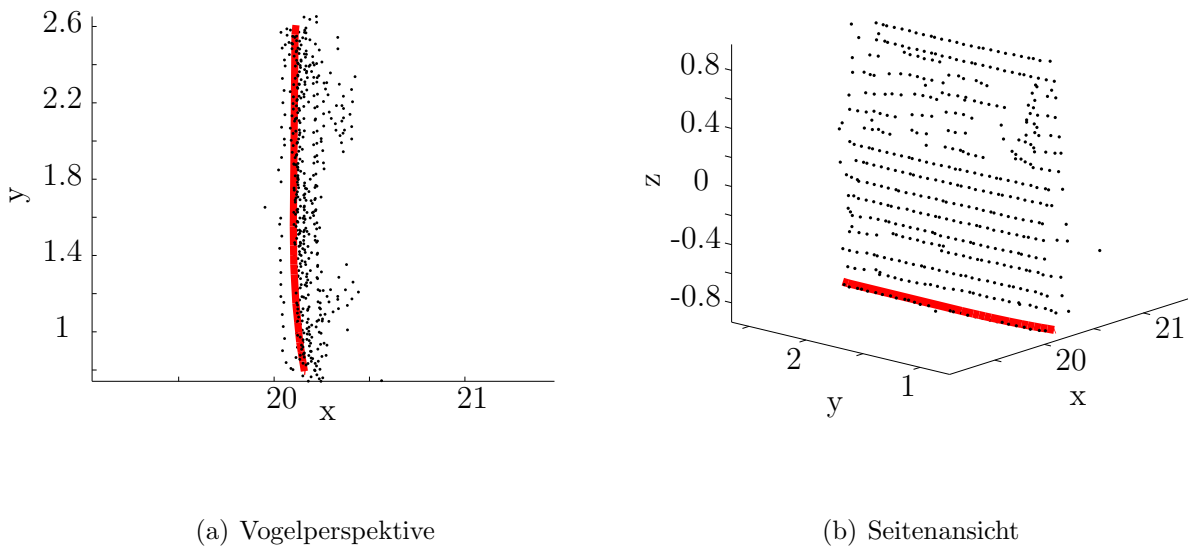


Abbildung 4.33: Zweidimensionale Außenkontur mit 3D Punktmessungen eines Transporters, dessen Heck in den Messungen sichtbar ist

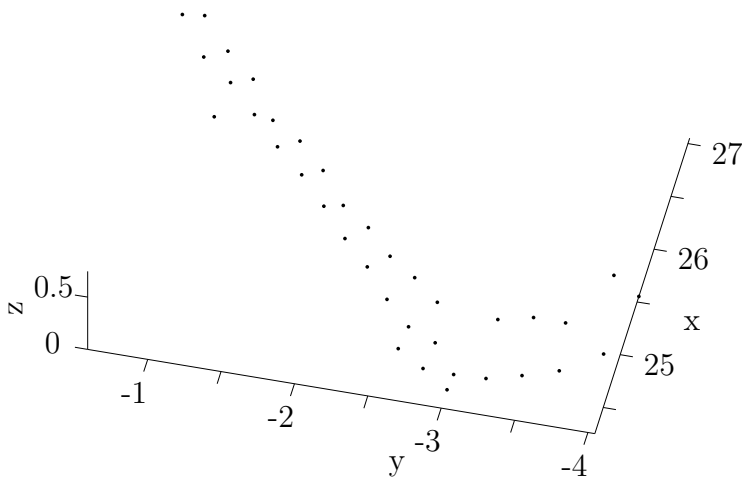
4.13 Übertragung des Verfahrens zur Außenkonturbestimmung auf andere Sensoren

Das Verfahren zur Außenkonturbestimmung ist in den vorangegangenen Kapiteln allgemein formuliert worden. Das bedeutet, dass keine sensorspezifischen Eigenschaften in das Verfahren eingeflossen sind und dass es sich daher auf beliebige Daten von Sensoren, die dreidimensionale Messungen von einem Objekt liefern, anwenden lassen müsste. Bisher wurden allerdings nur Messungen des Velodyne HDL-64E betrachtet. Im Folgenden wird das Verfahren auf einen Lasersensor mit vier Ebenen und auf Daten von einer Stereokamera angewendet.

4.13.1 Lasersensor mit vier Ebenen

Für die Anwendung der Außenkontur auf einen Lasersensor mit vier Ebenen wird der im Kapitel '2.2.1 Lasersensor' vorgestellte SICK LD-MRS verwendet.

Die in Kapitel '4.5 Flächeneigenschaften' beschriebenen Eigenschaften treffen allgemein auf Fahrzeuge zu. Dabei wird davon ausgegangen, dass das Fahrzeug sowohl in seiner Ausdehnung in x - und y -Richtung, als auch in z -Richtung sichtbar ist. Bei einem Lasersensor mit vier Ebenen lässt sich die Ausdehnung in z -Richtung nicht ausreichend messen. Daher ist die Eigenschaft aus Kapitel '4.5.3 Negative Gradienten' nicht notwendig. Dennoch kann sie bei der Berechnung beibehalten werden. Es muss keine Anpassung des Verfahrens erfolgen. Die 3D Punktmessungen und das zugehörige Kamerabild der Dokumentationskamera sind in Abbildung 4.34 angegeben.



(a) 3D Punktmessungen



(b) Bild der Dokumentationskamera

Abbildung 4.34: Messung des SICK LD-MRS mit zugehörigem Dokumentationskamerabild

In Abbildung 4.35 sind die dreidimensionale (a) und die zweidimensionale (b) Außenkontur für die in Abbildung 4.34 (a) gezeigten 3D Punktmessungen dargestellt.

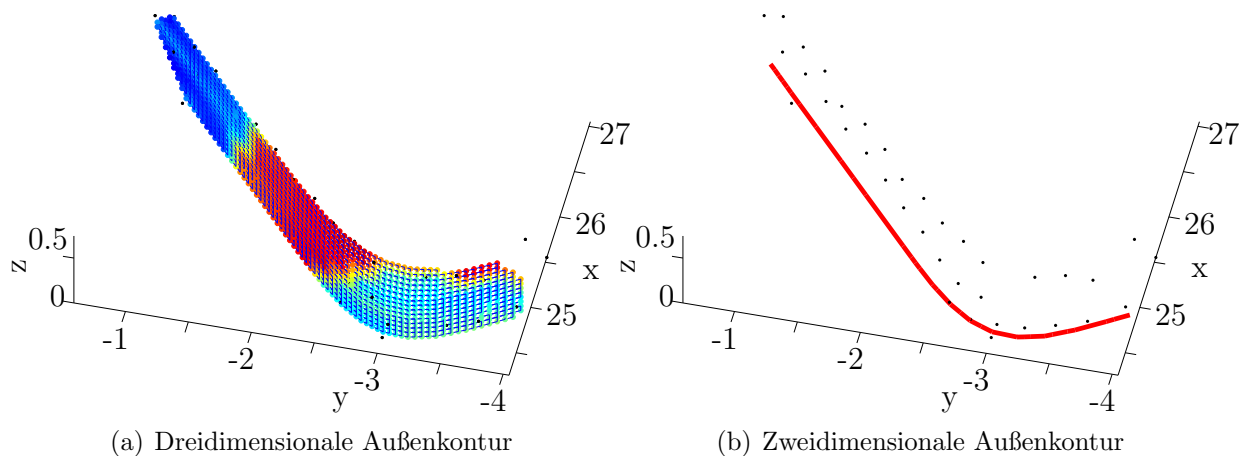


Abbildung 4.35: Außenkonturbestimmung mit 3D Punktmessungen eines SICK LD-MRS

Es zeigt sich, dass sich das Verfahren zur Außenkonturbestimmung ohne weitere Anpassungen auf 3D Punktmessungen mit wenigen Ebenen und somit auf seriennahe Sensoren anwenden lässt. Trotz der vier Ebenen des SICK LD-MRS liegen in dieser Messung nur drei Ebenen auf dem Fahrzeug. Die Konfidenz der dreidimensionalen Kontur ist im mittleren Teil des Fahrzeugs am höchsten. Zu den Rändern hin und an der Stelle der größten Änderung der Richtung der Kontur lässt die Konfidenz nach. An diesen Stellen liegen nur wenige Messungen vor, die die Flächenbestimmung stützen und sich somit auf die Eigenschaften aus '4.5 Flächeneigenschaften' verlassen werden muss. Durch die wenigen Punkte, die ein leichtes Rauschen aufweisen, kann hier die Kontur nicht durch alle Punkte gelegt werden.

4.13.2 Stereokamera

Das Verfahren zur Außenkonturbestimmung wird im Folgenden auf 3D Punktmessungen einer Stereokamera ('2.2.3.2 Stereokamera') angewendet. Abbildung 4.36 zeigt die 3D Punktmessungen und einen Ausschnitt aus dem Kamerabild.

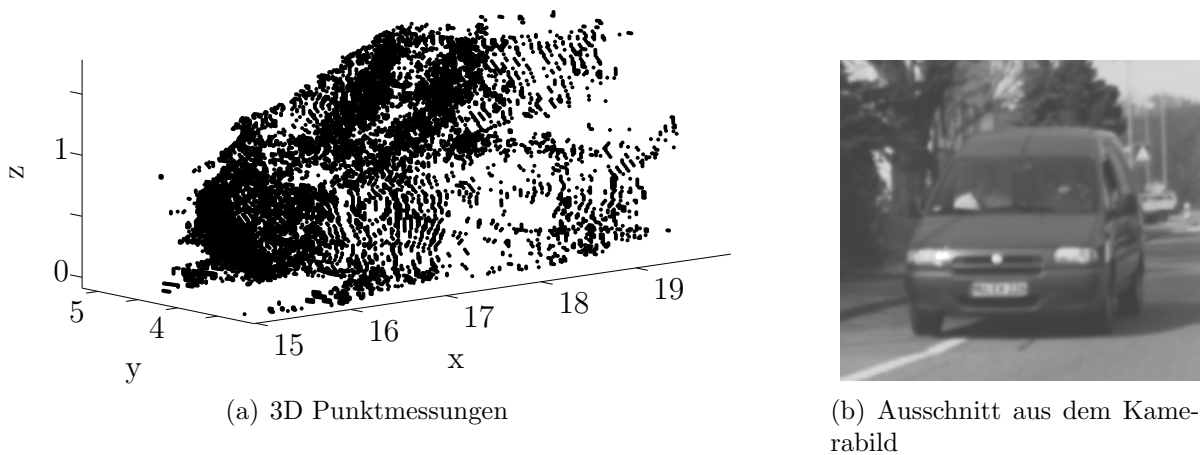


Abbildung 4.36: Messung der Stereokamera und Ausschnitt des Kamerabildes

In Abbildung 4.37 sind die dreidimensionale (a) und die zweidimensionale (b) Außenkontur für die in Abbildung 4.36 (a) gezeigten 3D Punktmessungen dargestellt.

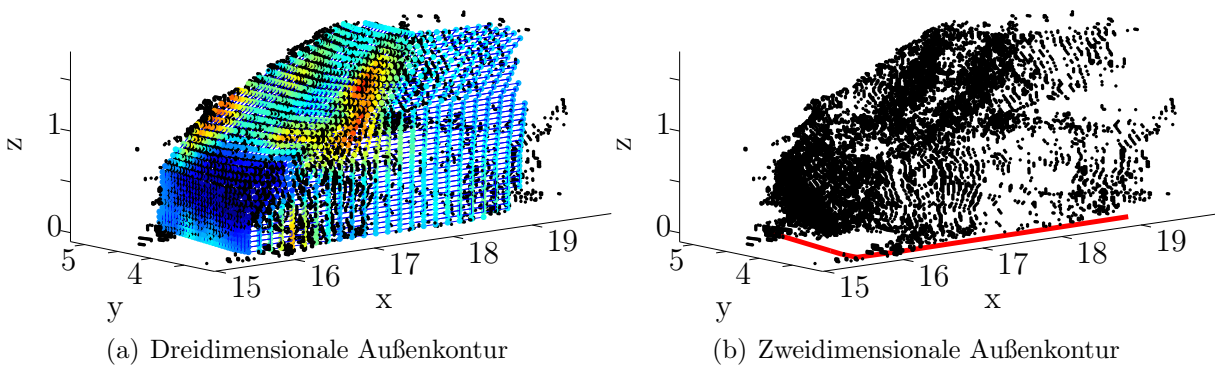


Abbildung 4.37: Außenkonturbestimmung mit 3D Punktmessungen einer Stereokamera

Das Verfahren zur Außenkonturbestimmung lässt sich, wie in Abbildung 4.37 gezeigt, auch auf 3D Punktmessungen einer Stereokamera anwenden. Von Vorteil ist die dichte Punktmenge, die aus den 3D Punktmessungen der Stereokamera entsteht. Es zeigt sich jedoch, im Vergleich zu den Lasersensoren, ein höheres Rauschen, was die Außenkonturbestimmung erschwert. Speziell die Fahrzeugfront in den 3D Punktmessungen weist hohe Rauschwerte auf. Die Konfidenz an dieser Stelle deutet daher auf eine höhere Unsicherheit hin.

4.14 Vergleich der zweidimensionalen Außenkontur mit der realen Fahrzeugkontur

Um einen Vergleich zwischen der automatisch bestimmten Außenkontur und der wirklichen Außenkontur eines Fahrzeugs herzustellen, werden zwei Fahrzeuge auf eine Freifläche

gestellt. Das vom Messfahrzeug, ausgerüstet mit einem Velodyne HDL-64E, gemessene zweite Fahrzeug wird dabei als Zielfahrzeug bezeichnet. Die Positionen der beiden Fahrzeuge werden über mehrere Referenzpunkte zueinander vermessen, sodass der Abstand und die Orientierung des Zielfahrzeugs bekannt ist. Über eine technische Zeichnung des Zielfahrzeugs (Mercedes-Benz S-Klasse, W221), wird die reale Außenkontur bestimmt und an die entsprechende Position im Sensorkoordinatensystem des Velodyne HDL-64 eingetragen. Die segmentierten Laserdaten sind zusammen mit der realen Außenkontur und der über das zuvor vorgestellte Verfahren zur Außenkonturbestimmung in Abbildung 4.38 dargestellt. Abbildung 4.39 zeigt den Übergang zwischen Seitenfläche und Heck des Fahrzeugs in einer vergrößerten Darstellung.

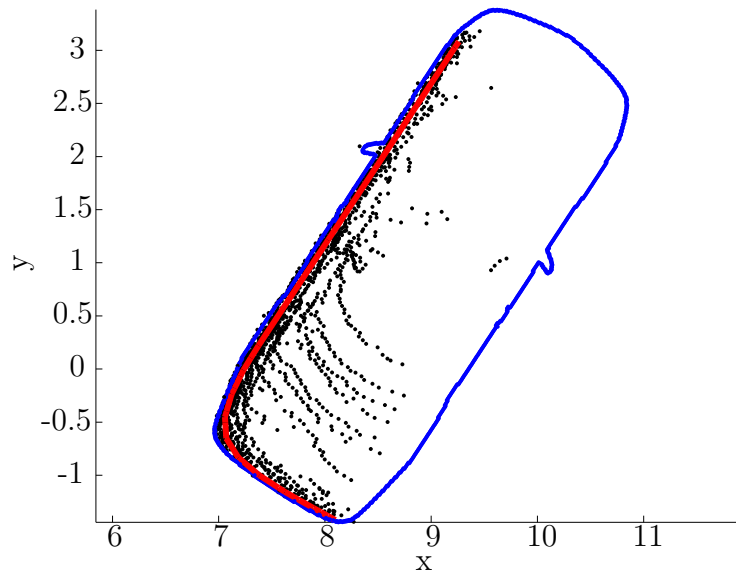


Abbildung 4.38: Vergleich der realen Außenkontur (blau) mit der berechneten zweidimensionalen Außenkontur (rot). 3D Punktmessungen sind in schwarz dargestellt.

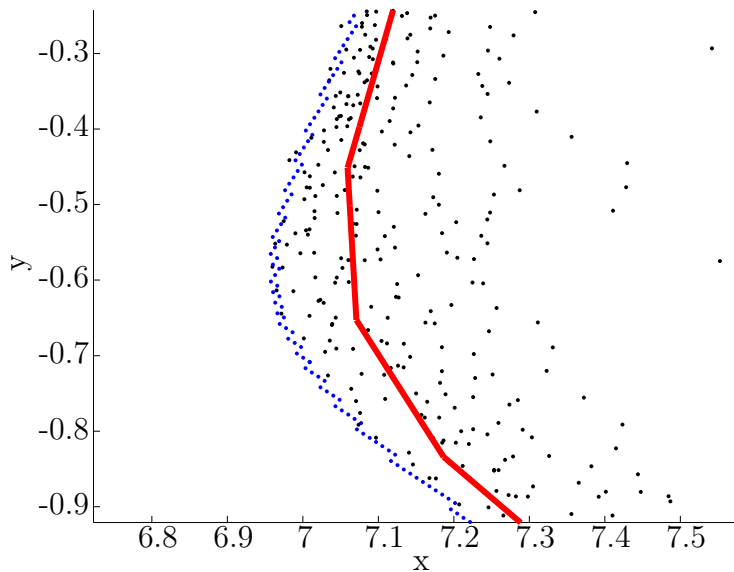


Abbildung 4.39: Übergang zwischen Seitenfläche und Heck. Vergleich der realen Außenkontur (blau) mit der berechneten zweidimensionalen Außenkontur (rot). 3D Punktmessungen sind in schwarz dargestellt.

Der Vergleich zwischen der realen und der berechneten zweidimensionalen Außenkontur zeigt eine gute Übereinstimmung. Die zweidimensionale Außenkontur liegt an der Fahrzeugseite ca. 6cm innerhalb der realen Außenkontur. Im hinteren Bereich des Fahrzeugs, am Übergang zwischen Seitenfläche und Heck, vergrößert sich der Fehler auf ca. 10cm . Bei der Erstellung der in Abbildung 4.38 blau gezeigten Außenkontur wurden die äußeren Elemente des Fahrzeugs über eine Projektion in die xy -Ebene gebracht. Hierbei wurde nicht beachtet, ob es sich um Designelemente handelt, die nur wenige cm Ausdehnung entlang der z -Achse haben und somit vom Lasersensor nicht mit ausreichender Punktdichte erfasst werden können. Daher entsteht keine Ebene auf denen die Messungen zum liegen kommen, sondern Messungen, die in ihrer Distanz zum Sensor entlang der z -Achse variieren. Durch die im Algorithmus zur Außenkonturbestimmung enthaltene Bedingung die quadratischen Abstände zwischen Messungen und Außenkontur zu minimieren, wird die Außenkontur aus diesem Grund etwas nach innen versetzt. Im hinteren Bereich des Fahrzeugs wird die Bedingung der Glattheit durch den Übergang von Fahrzeugseite zu Fahrzeugheck verletzt. Der Algorithmus versucht die Glattheit dennoch zu erfüllen und die Außenkontur wird in das Fahrzeug gezogen.

Im vorderen Bereich des Fahrzeugs lässt sich in der realen Außenkontur der Spiegel erkennen. In den Messdaten ist lediglich eine Messung vom Spiegel enthalten. Dies rechtfertigt die Vernachlässigung des Spiegels bei der Modellierung der Außenkontur, da nicht mit Sicherheit angenommen werden kann, dass es sich hierbei um eine reale Messung des Spiegels handelt.

4.15 Anwendung der Außenkontur

In den folgenden Abschnitten werden drei verschiedene Anwendungsbeispiele für die Verwendung der Außenkontur vorgestellt. Zunächst wird auf die Abstandsbestimmung mit Hilfe der Außenkontur eingegangen. Der zweite Abschnitt zeigt einen Vergleich der Außenkontur mit Radarmessungen. Im dritten Abschnitt wird eine Geschwindigkeitsbestimmung auf Grundlage der Außenkontur vorgestellt.

4.15.1 Bestimmung des kürzesten Abstandes zum nächsten Fahrzeug

Für viele Fahrer-Assistenz-Funktionen ist die Auswertung des kürzesten Abstands zwischen dem eigenen und einem anderen Fahrzeug von Interesse. Hierbei kann es sich beispielsweise um Funktionen handeln, die eine Notbremsfunktion aktivieren. Mit der berechneten Außenkontur liegt eine modellhafte Beschreibung des anderen Fahrzeugs vor, die es ermöglicht diesen Abstand zu bestimmen. Abbildung 4.40 zeigt den kürzesten Abstand d zwischen dem Sensor und der berechneten Außenkontur eines Fahrzeugs.

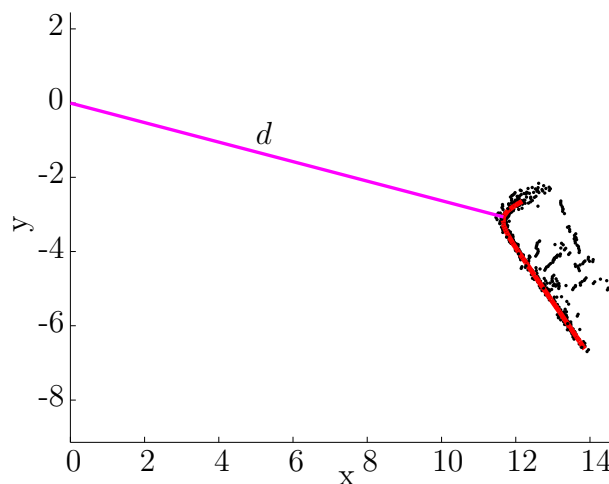


Abbildung 4.40: Minimaler Abstand d zwischen Außenkontur (rot) und Sensor

4.15.2 Vergleich zwischen der Lage der Außenkontur und einer Radarmessung

Durch die physikalischen Eigenschaften des Lichtes, die sich ein Lasersensor zunutze macht, ist es möglich ein Fahrzeug in seiner Ausdehnung zu vermessen. Dabei ist es nahezu unerheblich, ob beispielsweise der Stoßfänger aus Kunststoff gefertigt ist, oder metallische Elemente aufweist. Im Gegensatz dazu geben die metallischen Verstrebungen im Fahrzeug hinter dem aus Kunststoff gefertigten Stoßfänger für einen Radarsensor ein starkes Rückstreuungszentrum. Das bedeutet, dass die meiste Energie von dieser Stelle zurückgegeben wird. Die Rückstreuungszentren sind im Allgemeinen eher hinter der äußeren Hülle des Fahrzeugs zu vermuten. Aus Sicht der Radarsignalverarbeitung ist es daher notwendig zu wissen, wo an einem Fahrzeug die Rückstreuungszentren zu erwarten sind. Mit

diesem Wissen lassen sich daraufhin entsprechende Modelle erstellen, die beispielsweise eine Objektverfolgung robuster und genauer werden lassen. Sylvia Pietzsch et al. [72] verwenden ein Modell für die Zuordnung von typischen Rückstreucentren für die Objektverfolgung bei einer Fusion von Daten eines Lasersensors und Daten eines Radarsensors. Ferner lassen sich durch derartige Untersuchungen Radare von verschiedenen Herstellern auf ihre Funktionsweise untersuchen und vergleichen. Um detaillierte Aussagen über die Rückstreucentren, sowie deren lokale Veränderungen über die Zeit, bei verschiedenen Fahrmanövern zu erhalten, ist es notwendig die Außenkontur des Fahrzeugs zu jedem Zeitschritt zu kennen. Der Algorithmus zur Außenkonturbestimmung von Fahrzeugen in Daten eines Lasersensors kann in diesem Fall als Referenzmessung eingesetzt werden.

Bevor ein Vergleich der Außenkontur mit Radarmessungen durchgeführt werden kann, muss die Kalibrierung zwischen beiden Sensoren bestimmt werden. Zunächst werden die Einbaupositionen der Sensoren vermessen und deren Koordinatensysteme zur Deckung gebracht. Im nächsten Schritt wird ein Kalibrierobjekt benötigt. In diesem Fall wird ein Winkelreflektor für das Radar und eine im Zentrum angebrachten Fläche, die vom Lasersensor angemessen werden kann, verwendet. Der Aufbau ist in Abbildung 4.41 dargestellt.

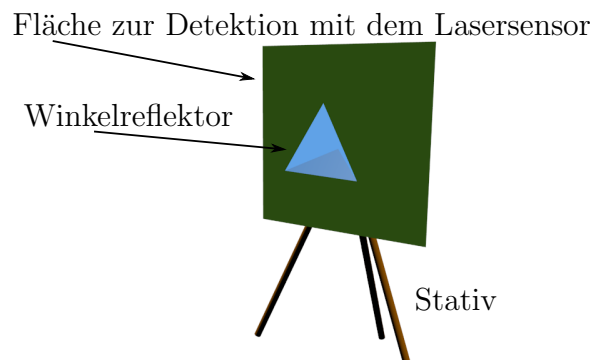


Abbildung 4.41: Kalibrierobjekt

Bei konstanter Zufahrt auf das Kalibrierobjekt werden die Zeitstempel und die Distanzen zum Kalibrierobjekt für jeden Sensor aufgezeichnet. Der Unterschied zwischen den Positionen ermöglicht die Bestimmung der Latenz. Gleichzeitig erfolgt auf diese Weise eine räumliche Kalibrierung, die durch einen Zeitwert ausgedrückt mit in den Latenzwert einfließt. Damit ergibt sich eine Kalibrierung des einen auf den anderen Sensor. Eine Trennung zwischen Latenz und räumlicher Kalibrierung kann mit dieser Methode nicht erfolgen, da sie beide in dem bestimmten Zeitwert zusammengefasst vorliegen. Abbildung 4.42 zeigt eine beispielhafte Auswertung mit echten Daten des Velodyne HDL-64E und einem Radarsensor.

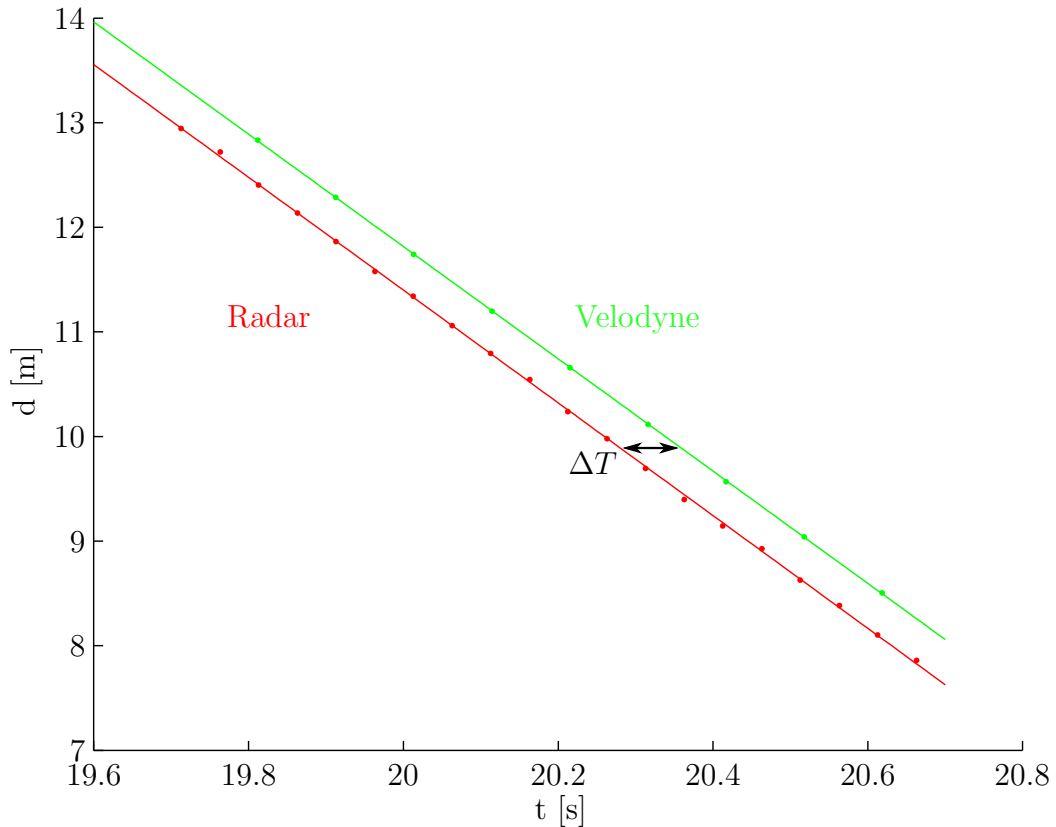
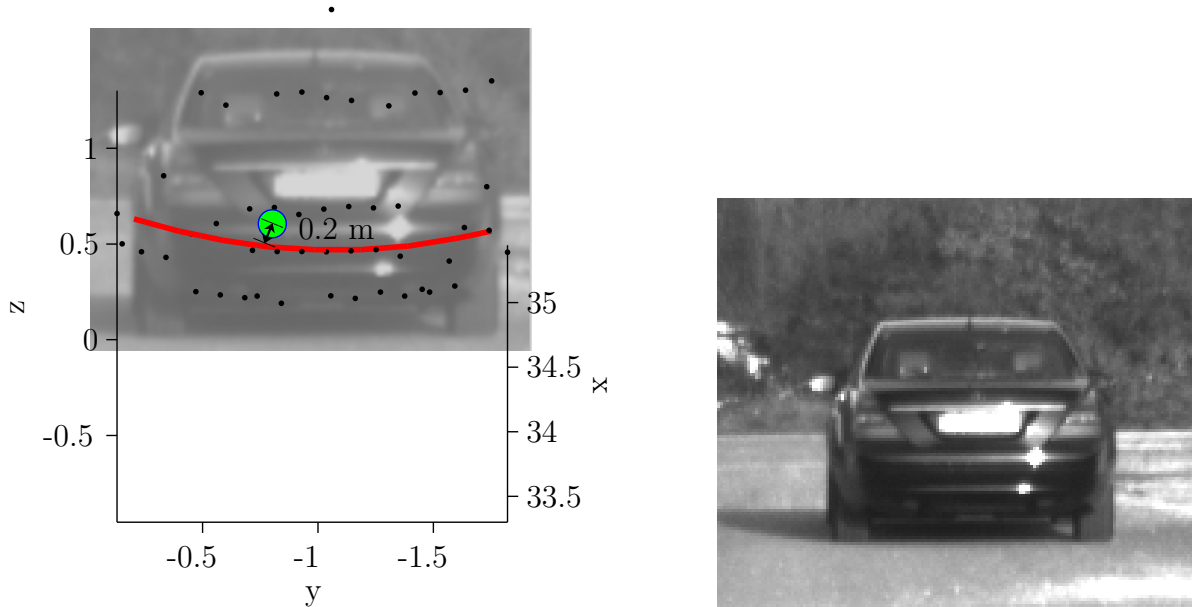


Abbildung 4.42: Bestimmung der räumlichen und zeitlichen Kalibrierung zwischen einem Radar und einem Lasersensor

Abbildung 4.43 (a) zeigt beispielhaft an einem Auszug aus einer Sequenz den Vergleich der Außenkontur, bestimmt aus Daten des Lasersensors, mit einer Radarmessung. Der Ausschnitt zeigt ein im Abstand von ca. $34m$ fahrenden PKW. Das zugehörige Kamerabild zur Dokumentation ist in Abbildung 4.43 (b) dargestellt. Bei der Radarmessung handelt es sich um eine ungefilterte Messung. Das bedeutet, dass die Messung nicht über die Zeit durch den Sensor geglättet wurde. Der Sensor liefert, wie vermutet, eine Reflexion hinter dem Stoßfänger. Der Abstand entlang der x-Achse zwischen Radarreflexion und Außenkontur beträgt ca. $0.2m$.



(a) Abstand der Radarmessung (grün) zur Außenkontur (rot). Die Messung ist perspektivisch dargestellt. Die Daten des Lasersensors sind nicht mit dem Kamerabild kalibriert. Das Kamerabild ist lediglich zum besseren Verständnis dargestellt.

(b) Abbildung des Fahrzeugs

Abbildung 4.43: Vergleich der Radarmessung mit der Außenkontur und Abbildung des Fahrzeugs

Mit der vorgestellten Methode lassen sich auf der einen Seite die Positionen der Messungen des Radarsensors in x -Richtung mit der der Außenkontur vergleichen und andererseits die Position der Radarmessungen auf dem Fahrzeug entlang der y -Achse bestimmen. Im Verlauf der Szene zeigt sich, dass die Radarreflexion nicht an der gezeigten Position bleibt. Sie wandert in diesem Fall von der Mitte auf die linke Kante des Fahrzeugs. Beim Abbiegen des Fahrzeugs nach rechts wandert die Radarreflexion auf die rechte hintere Fahrzeugkante. Somit zeigt sich, dass sich die Reflexionszentren für den Radarsensor je nach Winkel zwischen Objekt und Radarsensor verändern. Die untersuchte Szene beschreibt in diesem Fall beispielhaft nur das Verhalten für den als Zielobjekt verwendeten Fahrzeugtyp. Für andere Fahrzeuge sind unterschiedliche Ergebnisse zu erwarten, da sich die Streuzentren aus der Konstruktion des Fahrzeugs ergeben. Der Effekt des Wanderns wird jedoch ebenfalls auftreten.

Weitere Auswertungen, wie die Anzahl der Radarmessungen auf einem Fahrzeug, lassen sich auf diese Weise ebenfalls realisieren.

4.15.3 Geschwindigkeitsbestimmung durch Registrierung von Außenkonturen

Es wird angenommen, dass zum Zeitschritt $t - 1$ eine dreidimensionale Außenkontur eines Fahrzeugs bestimmt wurde. Im darauffolgenden Zeitschritt t lässt sich vom selben

Fahrzeug wiederum die dreidimensionale Außenkontur bestimmen. Es entstehen zwei Knotenmengen $K_{IKCN_{t-1}}$ und K_{IKCN_t} . Diese hängen über die Eigenschaft zusammen, dass sie dasselbe Fahrzeug beschreiben. Wenn sich das Fahrzeug jedoch zwischen den beiden Messungen bewegt hat, liegen die Knotenmengen nicht deckungsgleich übereinander. Die Deckung der Knotenmengen lässt sich über Registrierungsalgorithmen wiederherstellen. Über die aus der Registrierung berechneten Drehungen und Translationen und der bekannten Zykluszeit des Sensors lässt sich die Geschwindigkeit des Fahrzeugs bestimmen. Zur Registrierung lässt sich der “Iterative Closest Point“ Algorithmus einsetzen (4.15.3.1 (Weighted) Iterative Closest Point Algorithmus ((W)ICP)). Da zu jedem Knoten in den beiden Knotenmengen zusätzlich ein Konfidenzwert bekannt ist, kann dieser mit in den Registrierungsalgorithmus integriert werden. Für die Registrierung der Knotenmengen lässt sich als Erweiterung zum vorherigen Ansatz der “Weighted Iterative Closest Point“ Algorithmus (WICP, [75]) verwenden. Bei beiden Methoden werden die Geschwindigkeitswerte über einen sich anschließenden Kalman-Filter (4.15.3.2 Linearer Kalman-Filter) geglättet.

Bei Änderungen der Perspektive unter der der Sensor die Außenkontur des Fahrzeugs erfasst, können sich die Knotenmengen zwischen zwei Zeitschritten stark voneinander unterscheiden. Dies tritt auf, wenn von einem Zeitschritt auf den anderen eine zweite Seite der Fahrzeugkontur sichtbar wird (vergleiche Abbildung 4.47). In diesen Situationen schlägt die Registrierung fehl, da die ICP Algorithmen auf minimale Abstände optimieren und keine weiteren markanten Objekte wie Kanten am Fahrzeug sichtbar sind. Aus diesem Grund wird eine Überprüfung auf Ähnlichkeit mit Hilfe der folgenden Kriterien durchgeführt:

- Fläche, die von der Knotenmenge aufgespannt wird
- Anzahl der Knoten in der Knotenmenge
- Seitenlängen die sich aus den Knotenmengen bei Annahme einer Rechteckform ergeben

Kann keine Ähnlichkeit zwischen zwei aufeinander folgende Knotenmengen festgestellt werden, wird der Bewegungsvektor aus den Mittelwerten der Knotenkoordinaten bestimmt.

4.15.3.1 (Weighted) Iterative Closest Point Algorithmus ((W)ICP)

Der ICP-Algorithmus von Paul Besl und Neil McKay [4] geht von zwei Punktmenge (P_1 , P_2) aus, die durch Drehung und Verschiebung einer von beiden Punktmenge aneinander angepasst werden. Um die optimale Anpassung der beiden Punktmenge zueinander zu finden, wird ein Fehlermaß minimiert. Für den hier gezeigten Anwendungsfall entsprechen die Punktmenge P_1 und P_2 den Knotenmengen $K_{IKCN_{t-1}}$ und K_{IKCN_t} , wobei lediglich auf die Positionen (x, y, z) der Knoten zugegriffen wird.

Der Algorithmus gliedert sich in zwei Schritte. Im ersten Schritt werden zu den Punkten in der ersten Punktmenge $K_{IKCN_{t-1}}$ jeweils die nächsten Punkte in der zweiten

Punktmenge K_{IKCN_t} gesucht. Dabei wird in diesem Fall vom euklidischen Abstand ausgegangen. Die Punktpaare werden in einem entsprechenden Vektor $m_{icp}^{\vec{}} \in \mathbb{R}^{N_{K_{t-1}}}$ gespeichert. $N_{K_{t-1}}$ ist dabei die Anzahl der Knotenpunkte in der ersten Punktmenge. Der zu Punkt j ($j \in \{0 \dots N_{K_{t-1}}\}$) aus der ersten Punktmenge gehörende Punkt in der zweiten Punktmenge wird durch $m_{icp_j}^{\vec{}}$ beschrieben. Im zweiten Schritt wird eine entsprechende Transformation (Drehung und Verschiebung) bestimmt, die den Fehler, der sich hier über die Summe der quadratischen Abstände der Punkte mit ihren nächsten Nachbarn zusammensetzt, minimiert. Im Anschluss wird die Transformation auf die Punktmenge angewandt und der Algorithmus beginnt wieder beim ersten Schritt. Als Abbruchbedingung kann entweder die Anzahl der Durchläufe, oder auch ein maximal erlaubter Fehler dienen.

Eine Erweiterung des ICP-Algorithmus stellt der gewichtete ICP-Algorithmus (WICP) dar. Dabei existiert für jeden Abstand zwischen den Punkten der Punktmenge ein Gewichtungsfaktor. Dieser bestimmt, wie viel Einfluss der jeweilige Abstand auf die Lösung haben soll. Ist beispielsweise bekannt, dass ein Punkt in einer Punktmenge mit hoher Wahrscheinlichkeit fehlerhaft ist, so sollte dieser möglichst wenig Einfluss auf die Lösung nehmen und daher einen geringen Gewichtungsfaktor erhalten. Für den hier vorgestellten Anwendungsfall berechnet sich der Gewichtungsfaktor aus den Konfidenzen der Knoten. Zur vereinfachten Schreibweise wird $K_{IKCN_{t-1}}$ durch Q mit $Q_j = (x_j, y_j, z_j, k_j)$ ersetzt. Q beinhaltet die Positionen der Knoten (x, y, z) und den zugehörigen Konfidenzwert (k). Entsprechendes gilt für Q' , welches K_{IKCN_t} substituiert. Der Gewichtungsfaktor berechnet sich entsprechend Gleichung 4.60.

$$\nu_j = \frac{Q_{k(j)} + Q'_{k(m_{icp_j}^{\vec{}})}}{2} \quad (4.60)$$

4.15.3.2 Linearer Kalman-Filter

Der Kalman-Filter ist ein rekursiver Filter, der den Zustand eines linearen Systems durch die Minimierung des mittleren quadratischen Fehlers schätzt. Er basiert auf dem im Grundlagenkapitel vorgestellten Bayes-Filter (2.5).

Ausgegangen wird von einem linearen zeitdiskreten System der Form:

$$\vec{x}_k = \mathbf{F}\vec{x}_{k-1} + \mathbf{B}\vec{u}_{k-1} + \vec{s}_{k-1} \quad (4.61)$$

$$\vec{y}_k = \mathbf{G}\vec{x}_k + \vec{w}_k \quad (4.62)$$

Dabei stellt \vec{x}_k den Zustand des Systems zum Zeitpunkt k dar. Da der wahre Zustand unbekannt ist, wird dieser durch den Kalman-Filter später geschätzt werden. Matrix \mathbf{F} wird die Zustandsübergangsmatrix und \mathbf{B} die Eingangsmatrix genannt. Die Zustandsübergangsmatrix sorgt für den Übergang des Zustandes von einem Zeitpunkt zum nächsten entsprechend dem zu Grunde gelegten Modells. Falls bekannte Steuergrößen \vec{u} auf das System einwirken, werden diese über die Eingangsmatrix \mathbf{B} zum Zustand addiert. Unsicherheiten in der Modellannahme werden über das Prozessrauschen \vec{s} eingebracht.

Gleichung 4.61 entspricht der vorgestellten Zustandsübergangsfunktion in Gleichung 2.20.

In Gleichung 4.62 wird Matrix \mathbf{G} Messmatrix genannt. Sie stellt die Verbindung zwischen Messung und Systemzustand her. Die Unsicherheiten bei der Messung werden über das Messrauschen \vec{w} modelliert. Gleichung 4.62 entspricht der Messgleichung aus Gleichung 2.21.

Für den Kalman-Filter wird angenommen, dass die beiden Zufallsvariablen, die das Prozess- bzw. das Messrauschen modellieren, unabhängig voneinander, mittelwertfrei und normalverteilt sind. Weiter wird angenommen, dass es sich um weißes Rauschen handelt. Mit \mathbf{Q} als Kovarianzmatrix des Prozessrauschens und \mathbf{R} als Kovarianz des Messrauschens gilt:

$$\vec{s} \sim \mathcal{N}(0, \mathbf{Q}) \quad (4.63)$$

$$\vec{w} \sim \mathcal{N}(0, \mathbf{R}) \quad (4.64)$$

Die Gleichungen des Kalman-Filters können wie beim Bayes-Filter (2.5) in zwei Schritte aufgeteilt werden:

1. Prädiktion
2. Innovation

Wird der geschätzte Zustand mit \hat{x} und die Prozessfehlerkovarianz mit \mathbf{P} angegeben, lassen sich die Gleichungen des Kalman-Filters wie in Abbildung 4.44 darstellen. Matrix \mathbf{K} wird als Kalman Verstärkung bezeichnet und Matrix \mathbf{I} ist die Einheitsmatrix.

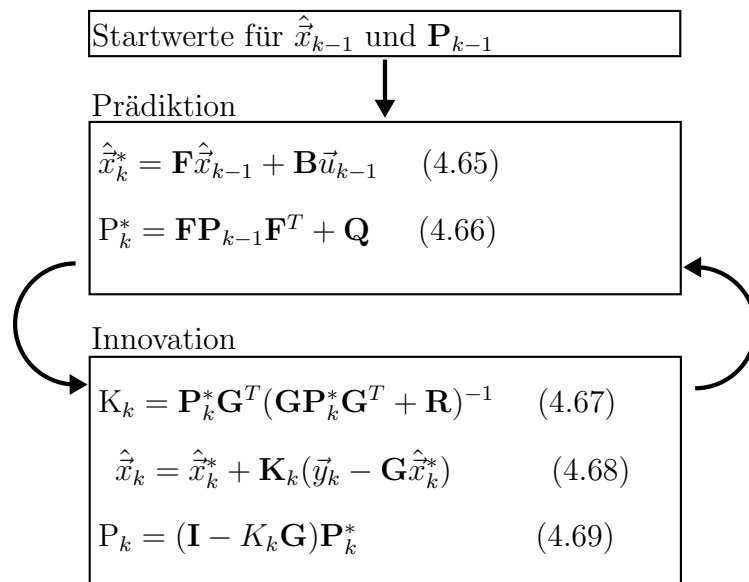


Abbildung 4.44: Gleichungen des Kalman-Filters

Eine detaillierte Beschreibung des Kalman-Filters findet sich in der ursprünglichen Veröffentlichung von Rudolph Kalman [53] oder bei Greg Welch und Gary Bishop [92].

Bei der hier verwendeten Implementierung des Kalman-Filters soll die Geschwindigkeit als konstanter Wert geschätzt werden. Der Zustandsvektor, sowie auch der Messvektor werden jeweils zu einem Skalar.

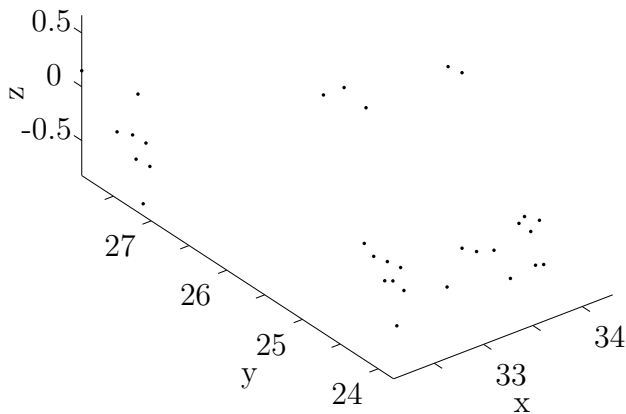
4.15.3.3 Auswertung

Für die Auswertung der Geschwindigkeitsbestimmung wird ein Messaufbau, wie in Kapitel '2.6 Erzeugung von Referenzdaten mit Inertialmessensorik und Globalem Positionssystem mit Differentialsignal (DGPS)' beschrieben, verwendet. Das beinhaltet, dass das Messfahrzeug, an dem der Sensor (Velodyne HDL-64E) angebracht ist, und das Zielfahrzeug jeweils mit einem Inertialmesssystem und einem DGPS ausgerüstet sind. Über die Kombination des Inertialmesssystems mit einem DGPS am Zielfahrzeug wird eine präzise Geschwindigkeitsmessung des Zielfahrzeugs ermöglicht.

Um das Zielfahrzeug eine zuvor definierte Strecke abfahren zu lassen, kommt der beschriebene Fahrroboter zum Einsatz.

In der verwendeten Messung treten drei verschiedene Störquellen auf, die Einfluss auf die Qualität der Messungen haben. Hierzu zählen die Einflüsse des Wetters, die Lackfarbe des Zielfahrzeugs und je nach Auftreffwinkel des Lasers auf das Zielfahrzeug die Reflexion des Laserlichts an der Karosserie.

Das Wetter war zur Zeit der Aufnahme regnerisch, was sich in einem erhöhten Rauschen der 3D Punktmessungen bemerkbar macht. Das Rauschen entsteht dabei durch Reflexionen an Regentropfen und in der Nähe des Zielfahrzeugs durch Reflexionen an Wassertropfen, die durch die Reifen des fahrenden Fahrzeugs aufgewirbelt werden. Zusätzlich zu den Wetterbedingungen wurden die Messungen durch die schwarze Lackierung des Zielfahrzeugs beeinträchtigt. Die schwarze Farbe führt zu einer erhöhten Absorption des vom Laser ausgesandten Lichts. Die erhöhte Absorption kann bis zu einer vollständigen Absorption führen. Der dritte Effekt ist auf die Ausführung des Fahrzeugdesigns zurückzuführen. Hierbei entstehen Regionen auf dem Fahrzeug, die das Laserlicht nicht mehr zurück reflektieren. Eine beispielhafte Messung zeigt Abbildung 4.45. Im Bereich der vorderen und der hinteren Türen verschwinden die Messungen vollständig.



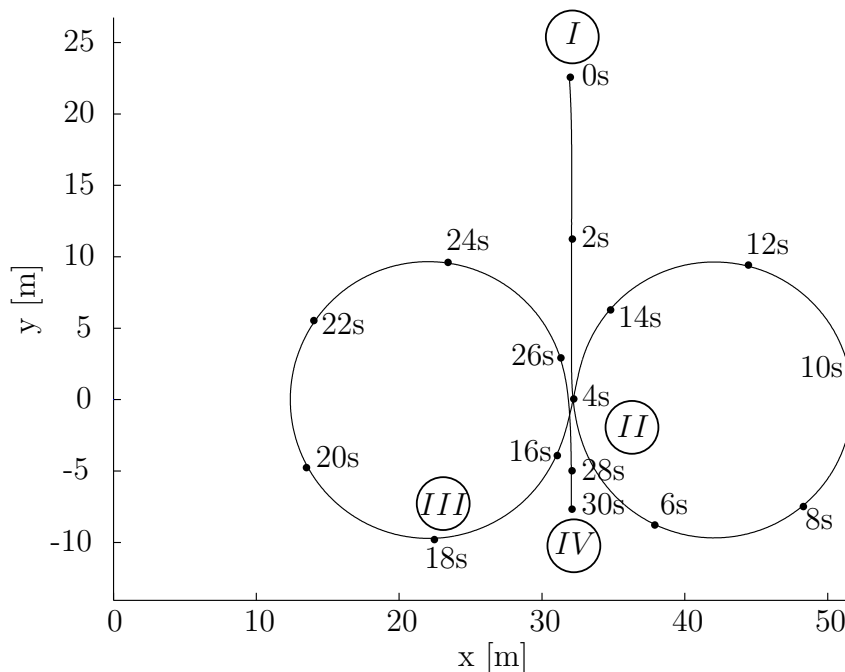
(a) 3D Punktmessungen



(b) Abbildung des Fahrzeugs

Abbildung 4.45: 3D Punktmessungen und Abbildung des Fahrzeugs

Die Route, die der Fahrroboter abfahren soll, ist als versetzte Kreisfahrt mit Zu- und Abfahrt definiert. Der Sensor auf dem Messfahrzeug steht fest an Position $(0, 0)$. Die Route ist in Abbildung 4.46 dargestellt. Die Messung beginnt in Punkt I bei schon vorhandener Maximalgeschwindigkeit. In Punkt II erfolgt eine Abbiegung nach links wobei nach dem ersten Kreis die Fahrt in den zweiten Kreis in Richtung Punkt III erfolgt. Nach der zweiten Kreisfahrt erfolgt ein Abbremsen bis zum Stillstand in Punkt IV. Für die spätere Erläuterung der Ergebnisse sind in Abbildung 4.46 neben einigen Positionen des Fahrroboters die zugehörigen Zeitstempel der Messungen dargestellt.

**Abbildung 4.46:** Route, die der Fahrroboter abfährt

Die versetzte Kreisfahrt stellt ein anspruchsvolles Szenario dar. Die Distanz zwischen Fahrroboter und Sensor ändert sich über einen Bereich von ca. 12 bis 52m entlang der x-Achse und ca. 22 bis $-10m$ entlang der y-Achse. Mit steigendem Abstand zwischen Sensor und Fahrroboter sinkt die Messpunktdichte bedingt durch die vom Sensor vorgegebene Winkelauflösung. Zusätzlich erfolgt ein ständiger Wechsel der Seiten des Zielfahrzeugs, die vom Sensor gemessen werden. Abbildung 4.47 zeigt schematisch welche Seiten des Zielfahrzeugs für den Sensor im versetzten Kreisfahrtszenario sichtbar sind. Die acht möglichen Kombinationen sind durch die Ziffern 1 bis 8 gekennzeichnet.

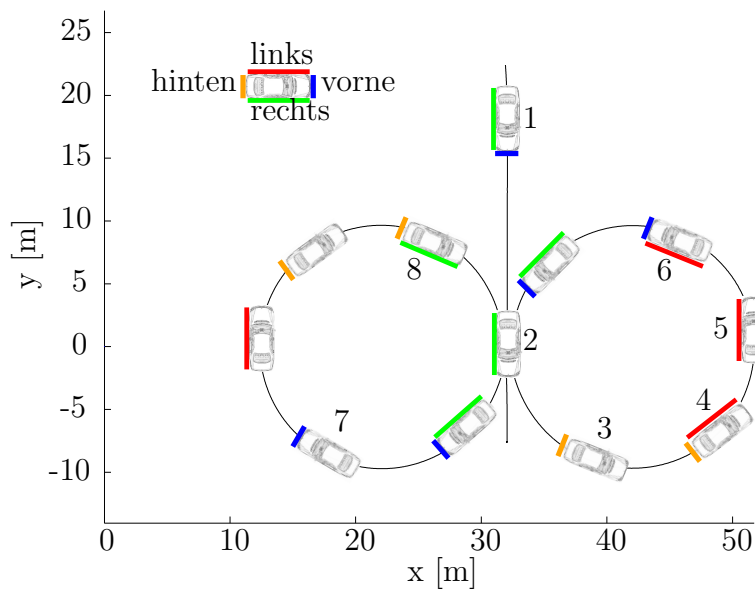


Abbildung 4.47: Darstellung der sichtbaren Seiten des Fahrzeugs in dem versetzten Kreisfahrtszenario

In Abbildung 4.48 ist das Ergebnis der Geschwindigkeitsbestimmung dargestellt. Die Abbildung zeigt den Vergleich der realen Geschwindigkeit aus der Kombination von Inertialmesssystem und DGPS mit der Geschwindigkeitsbestimmung über die Registrierung der Außenkonturen über den WICP- und den ICP-Ansatz. Zusätzlich ist die mit einem Kalman-Filter geglättete Geschwindigkeit über den Schwerpunkt dargestellt. In Abbildung 4.49 sind die jeweiligen Abweichungen der Algorithmen von der realen Geschwindigkeit aufgezeigt. Abbildung 4.50 stellt dar, ob die Geschwindigkeit aus der Registrierung oder aus dem Mittelwert der beiden Knotenmengen bestimmt wurde.

Für die Anfangsgeschwindigkeit wird jeweils ein Wert von $5 \frac{m}{s}$ angenommen. Die Prozessfehlerkovarianz \mathbf{P} des Kalman-Filters wird zu Beginn auf 1 gesetzt, was bedeutet, dass der initiale Zustand als nicht korrekt angesehen werden soll.

$$\mathbf{P} = [1] \quad (4.70)$$

Es wird angenommen, dass der Betrag der maximalen Beschleunigung und Verzögerung des Fahrzeugs bei $a_{max} = 9.81 \frac{m}{s^2}$ liegt. Die Standardabweichung σ_{aP} ergibt sich

bei Interpretation der Maximalbeschleunigung als 3σ Intervall zu $\sigma_{aP} = 3.27 \frac{m}{s^2}$. Mit der Messfrequenz von $10Hz$ des Velodyne Laser Sensors ergibt sich die Standardabweichung $\sigma_{vP} = 0.327 \frac{m}{s}$ und somit die Prozessfehlerkovarianzmatrix \mathbf{Q} zu

$$\mathbf{Q} = \left[\left(0.327 \frac{m}{s} \right)^2 \right] \quad (4.71)$$

Bei der Verwendung des Objektschwerpunkts als Messung ergibt sich, für eine Fahrzeuglänge von $5m$ und einer Breite von $2m$, bei einem Perspektivenwechsel von Seitenansicht auf Rückansicht des Fahrzeugs ein Sprung des Objektschwerpunkts von

$$d_M = \sqrt{\left(\frac{1}{2} 5m \right)^2 + \left(\frac{1}{2} 2m \right)^2} = 2.69m \quad (4.72)$$

Wird d_M als 3σ Intervall angesehen, folgt für die Standardabweichung $\sigma_{dM} = 0.898m$. Mit der Messfrequenz von $10Hz$ des Velodyne Laser Sensors ergibt sich die Standardabweichung $\sigma_{vM} = 8.98 \frac{m}{s}$ und somit die Messfehlerkovarianzmatrix \mathbf{R} zu

$$\mathbf{R} = \left[\left(8.98 \frac{m}{s} \right)^2 \right] \quad (4.73)$$

Die Messfehlerkovarianzmatrix deckt in diesem Fall auch die Modellierungsfehler in der Außenkontur ab.

Für die Ähnlichkeitsbestimmung zwischen zwei zeitlich aufeinanderfolgende Messungen werden die unten genannten Parameter verwendet:

- Verhältnis der Flächeninhalte zueinander: < 1.4
- Differenz der Knotenpunkte: < 100
- Seitenlängen die sich aus den Knotenmengen bei Annahme einer Rechteckform ergeben: jede Seitenlänge muss größer als $0.25m$ sein

Es zeigt sich zu Beginn der Messreihe, dass die über die Registrierungsalgorithmen bestimmten Geschwindigkeiten größere Abweichungen zur realen Geschwindigkeit aufweisen als die Geschwindigkeit, die über den Schwerpunkt bestimmt wurde. Dadurch, dass nur eine Seite des Fahrzeugs sichtbar ist und die Länge der Seite sich von Zeitschritt zu Zeitschritt verändert, ergibt sich ein ähnlicher Effekt wie das Blendenproblem aus der Bildverarbeitung. Dadurch lässt sich die Position der beiden Knotenmengen gegeneinander verschieben, ohne die Kosten im ICP Algorithmus zu verändern. Die Abweichungen um den Zeitstempel von ca. $10s$ ergeben sich aus den wenigen und verrauschten Messpunkten die keine präzise Außenkonturbestimmung mehr ermöglichen. Ab ca. dem Zeitstempel von $26s$ wird die Annahme des Filtermodells von einer konstanten Geschwindigkeit verletzt, was zu einem erhöhten Fehler führt.

In Tabelle 4.1 sind die Fehlerwerte (RMSE engl. Root Mean Square Error) der Algorithmen dargestellt. Als Vergleichswert ist der Fehler angegeben, der sich ergibt wenn der Schwerpunkt der Knotenmenge als Referenzpunkt zur Geschwindigkeitsbestimmung genutzt wird. Zusätzlich sind Werte angegeben, die sich aus den Untersuchungen von Matthias Schmid [80] mit Algorithmen zur Bewegungsdetektion in Belegungskarten ergeben haben. Der Vergleich ist möglich, da auf die gleichen Messdaten zurückgegriffen wurde. Zusätzlich ist ein Vergleichswert mit einem Algorithmus mit lokalen Gittermodellen in Kombination mit einem Partikel-Filter dargestellt. Dieses Verfahren wird in Kapitel 5 beschrieben.

Der Vergleich zeigt dass nahezu kein Unterschied zwischen dem ICP und dem WICP Algorithmus besteht. Daher kann, um die Laufzeit zu optimieren, auf die Berechnung der Gewichte verzichtet werden. Gegenüber dem reinen Schwerpunkt Algorithmus ergibt sich durch die Kombination mit der Registrierung ein Vorteil. Die Algorithmen mit Belegungskarten- bzw. Gittermodellansätzen zeigen im Vergleich geringere Fehlerwerte, was sich über die Akkumulation der Messungen zu einer stabileren Objektrepräsentation erklären lässt. Innerhalb der Algorithmen, die Messdaten akkumulieren, weist der Gittermodellansatz gegenüber dem Belegungskartenansatz einen geringeren Fehler auf. In Kapitel 5 findet sich eine detaillierte Beschreibung des auf lokalen Gittermodellen basierenden Partikel-Filter Ansatzes.

Algorithmus	Fehler (RMSE)
Schwerpunkt Kalman-Filter	$0.82 \frac{m}{s}$
ICP	$0.70 \frac{m}{s}$
WICP	$0.69 \frac{m}{s}$
Lokales Gittermodell Partikel-Filter (Kapitel 5)	$0.23 \frac{m}{s}$
Belegungskarte Kalman-Filter [80]	$0.49 \frac{m}{s}$
Belegungskarte Partikel-Filter Gauss [80]	$0.57 \frac{m}{s}$
Belegungskarte Partikel-Filter Epanechnikov [80]	$0.56 \frac{m}{s}$

Tabelle 4.1: Vergleich der Fehlerwerte (RMSE) für unterschiedliche Algorithmen zur Geschwindigkeitsbestimmung

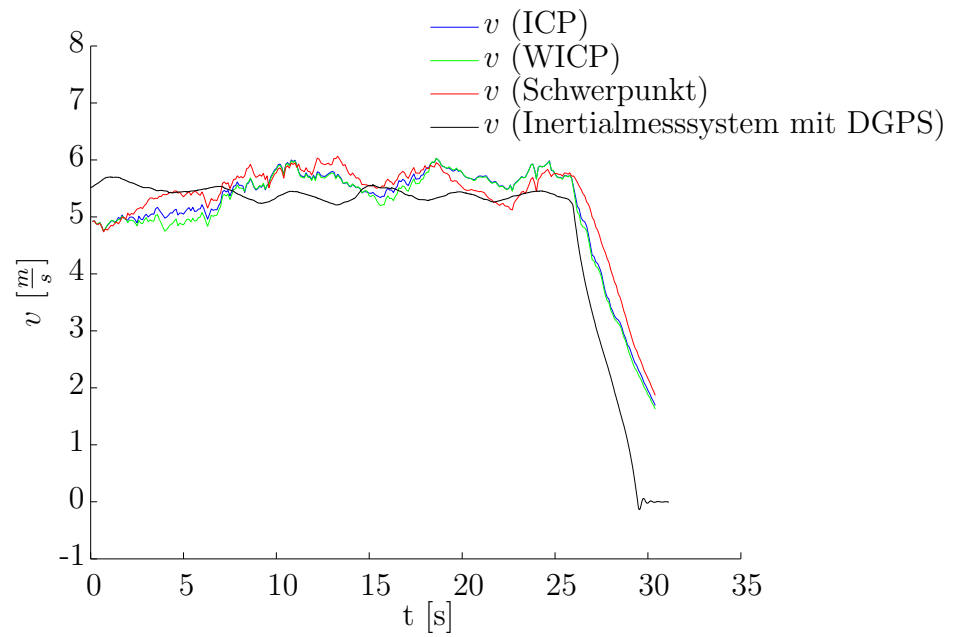


Abbildung 4.48: Vergleich der Geschwindigkeitsbestimmung mit der realen Geschwindigkeit

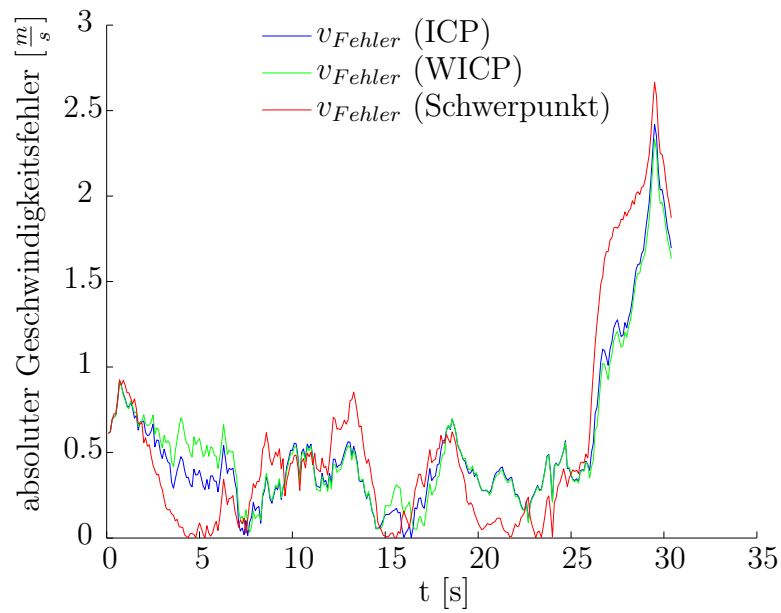


Abbildung 4.49: Absoluter Fehler der Geschwindigkeitsbestimmungen

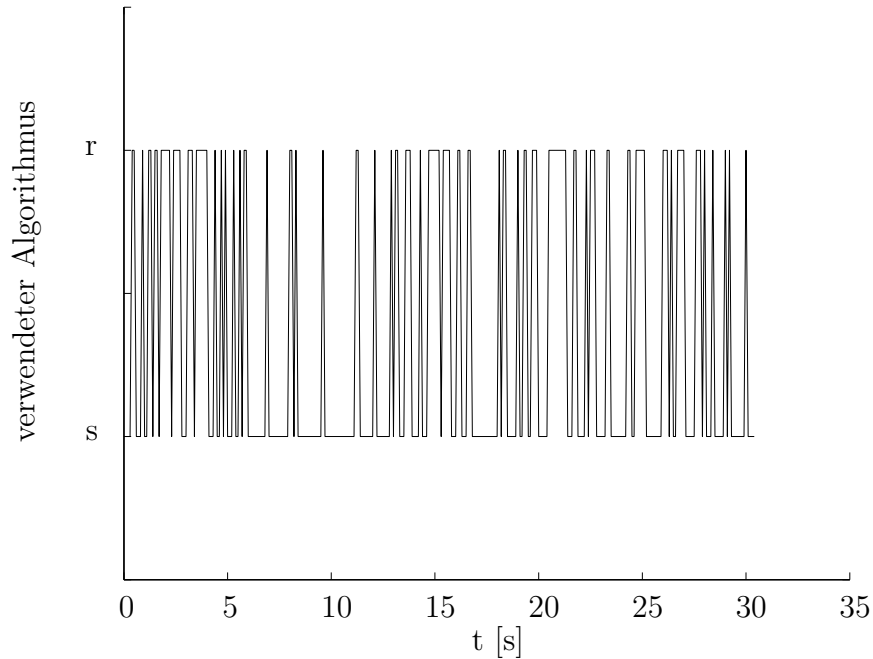


Abbildung 4.50: Verwendeter Algorithmus zur Geschwindigkeitsbestimmung.
r - Registrierungsalgorithmus, s - Schwerpunktalgorithmus

4.16 Zusammenfassung

In diesem Kapitel wurde ein neues und gegen Ausreißerdaten sehr robustes Verfahren zur Bestimmung einer Außenkontur von Fahrzeugen in dreidimensionalen Lasermessungen vorgestellt. Bei der beschriebenen Außenkonturbestimmung wird eine Fläche in die Lasermessungen eingepasst. Die Fläche wird dabei über eine Gitterfunktion beschrieben. Die Einpassung der Gitterfunktion in die Lasermessungen erfolgt über die Variation von Knotenpunkten, die die Gitterfunktion definieren. Die Robustheit der Außenkonturbestimmung wird einerseits dadurch erreicht, dass sie auf einer Ausgleichsrechnung aufbaut. Hierzu wird die Methode der kleinsten quadratischen Abstände verwendet. Auf der anderen Seite wird die Robustheit durch die Definition von Flächeneigenschaften erreicht. Dabei handelt es sich um Modellwissen über die geometrischen Eigenschaften von Fahrzeugen. Dieses Modellwissen fließt bei dem sich ergebenden Optimierungsproblem in Form von Randbedingungen ein. Die Robustheit wurde mit Hilfe von künstlich veränderten realen Messdaten gezeigt.

Für eine echtzeitfähige Implementierung des gesamten Algorithmus zur Außenkonturbestimmung lässt sich die Eigenschaft ausnutzen, dass die Matrizen, welche das Optimierungsproblem mit seinen Randbedingungen definieren, sehr dünn besetzt sind.

Über die Kombination der Abweichungen, die sich aus der Lösung des Optimierungsproblems ergeben, und der Anzahl von benachbarten Messungen zu einem Knotenpunkt der Gitterfunktion, kann ein Konfidenzmaß für die Außenkontur bestimmt werden.

Im Regelfall genügt für viele Anwendungen die Bestimmung einer zweidimensionalen Außenkontur. Diese lässt sich als Referenz nutzen, wenn der zu vergleichende Sensor keine Höheninformationen liefert. Ein Beispiel stellt der Vergleich mit einem Radarsensor dar. Es wurde gezeigt, wie sich die dreidimensionale Außenkontur mit ihrer Flächenbeschreibung auf eine zweidimensionale Außenkontur reduzieren lässt. Der Vorteil dieser Vorgehensweise ist, dass indirekt alle Messungen in der zweidimensionalen Außenkontur enthalten sind.

Das Verfahren zur Außenkonturbestimmung, welches hauptsächlich für die Anwendung mit Messungen des Velodyne HDL-64E entwickelt wurde, kann prinzipiell ohne Anpassung von Parametern auch auf andere Sensoren übertragen werden. Je nach verwendetem Sensor empfiehlt sich jedoch unter Umständen die Anpassung der Zellgröße der Gitterfunktion. Es wurde gezeigt, dass sich das Verfahren auf dreidimensionale Messungen einer Stereokamera genauso anwenden lässt, wie auf Messungen von einem Lasersensor mit vier Ebenen.

Der letzte Teil des Kapitels hat sich mit verschiedenen Anwendungen der Außenkonturbestimmung beschäftigt. Dabei wurde gezeigt, wie sich der kürzeste Abstand zum nächsten Fahrzeug bestimmen lässt, wie die Außenkonturbestimmung als Referenz für Radarsensoren genutzt werden kann und wie über die Außenkontur die Geschwindigkeit eines Fahrzeugs bestimmt werden kann. Die gezeigten Beispiele dienen der Veranschaulichung der Anwendungsmöglichkeiten und sind nicht als vollständige Aufzählung zu sehen.

Objektverfolgung mit einem objekt-lokalen Gittermodell

Bei der Repräsentation der statischen Umgebung mit Sensordaten werden häufig gitterbasierte Ansätze verwendet. Dabei wird die Umgebung durch viele einzelne Quadrate, oder im dreidimensionalen Fall durch Würfel, diskretisiert. Jedes Quadrat, bzw. jeder Würfel, speichert einen Wert über seine Belegtheit. Modellieren lässt sich die Belegtheit beispielsweise über einen Wahrscheinlichkeitswert oder durch das Zählen der Messungen, die in ein entsprechendes Quadrat, bzw. einen entsprechenden Würfel, gefallen sind. Für die Repräsentation der statischen Umgebung wird eine Region festgelegt, in der ein zusammenhängendes Gittermodell aufgebaut wird, in das alle Messdaten eingetragen werden.

Bei der Objektverfolgung mit einem objekt-lokalen Gittermodell wird der Ansatz zur Repräsentation der statischen Umgebung über ein Gittermodell auf ein sich bewegendes, sich aber in seiner Form nicht veränderndes, Objekt übertragen. Das bedeutet, dass jedes Objekt sein eigenes Gittermodell besitzt. Durch die Kombination von einem objekt-lokalen Gittermodell und einem Partikel-Filter lässt sich eine präzise Objektverfolgung erstellen. Gleichzeitig entsteht ein dreidimensionales Modell des Objektes in einem objekt-lokalen Gittermodell. Der Vorteil bei der Verwendung eines objekt-lokalen Gittermodells ist, dass keine vorherige Entscheidung über die Form des zu verfolgenden Objekts getroffen werden muss. Des Weiteren ergibt sich nicht die Problematik der Bestimmung von statischen und dynamischen Objekten, da jedes Objekt seinen Bewegungszustand ändern kann.

5.1 Stand der Technik

Gitterbasierte Ansätze zur Umgebungsrepräsentation von statischen Objekten gehen auf den von Alberto Elfes [24] vorgestellten Ansatz zurück. Durch die Entwicklung von Sensoren, die dreidimensionale Messungen im Raum ermöglichen, und durch immer höhere Rechenleistungen, wird es möglich den zweidimensionalen auf einen dreidimensionalen Ansatz zu erweitern. Matthias Schmid et al. [81] zeigen einen Ansatz, der es ermöglicht trotz des höheren Speicherbedarfs bei der Verwendung von dreidimensionalen Gittern, eine dennoch effiziente Speichernutzung zu erreichen. Dabei wird auf die Speicherung

der Daten im Octree-Format zurückgegriffen. Octrees speichern Daten über eine Baumstruktur, deren Knotenpunkte sich jeweils in acht oder keine Nachfolger aufteilen. Dies hat den Vorteil, dass der Speicher des dreidimensionalen Raums nicht bereits zu Beginn komplett reserviert werden muss, sondern er wird erst angefordert und in den Baum eingetragen, wenn Messungen an einer Position im Raum aufgetreten sind. Dies erfordert eine entsprechende Abbildung der Baumstruktur in den dreidimensionalen Raum. Armin Hornung et al. [47] zeigen einen vergleichbaren Ansatz zur effizienten Speicherung von dreidimensionalen Messdaten.

Weitere Ansätze verbinden die Kartierung der Sensordaten mit einer eigenen Lokalisierung. Dies wird allgemein als SLAM (Simultaneous Localization And Mapping) bezeichnet. Austin Eliazar [25] stellt einen Lösungsansatz für die Problemstellung des SLAMs vor. Der mit DP-SLAM (Distributed Particle - SLAM) bezeichnete Algorithmus nutzt einen Partikel-Filter (Kapitel '5.2 Partikel-Filter') um die multivariate Verteilung über verschiedene Karten und Lokalisierungen zu bestimmen.

Frank Moosmann et al. [65] akkumulieren 3D-Messungen eines Objekts in einem objektbezogenen lokalen Koordinatensystem. Die Akkumulation neuer Messungen erfolgt mit Hilfe des Iterative-Closest-Point-Algorithmus, welcher bereits in Kapitel '4.15.3.1 (Weighted) Iterative Closest Point Algorithmus ((W)ICP)' vorgestellt wurde. Über die Integration der Messdaten wird eine verfeinerte Bewegung bestimmt, die in den nachgelagerten Kalmanfilter [53] integriert wird.

Jan Effertz [22] beschreibt einen Ansatz mit objekt-lokalen Gittermodellen, in denen zweidimensionale Konturen akkumuliert werden. Um die Datenmengen gering zu halten, werden die Messungen nach der Verarbeitung über Polygonzüge repräsentiert. Es wird lediglich die berechnete zweidimensionale Kontur in dem objekt-lokalen Gittermodell abgelegt.

5.2 Partikel-Filter

Der Ansatz zur Objektverfolgung mit einem objekt-lokalen Gittermodell basiert auf der Anwendung eines Partikel-Filters, der den Zustand eines Objekts schätzt. Bei einem Partikel-Filter handelt es sich um eine numerische Implementierung des Bayes-Filters aus Kapitel 2.5.

5.2.1 Der Partikel-Filter

Wie bereits eingangs erwähnt, handelt es sich beim Partikel-Filter um eine numerische Implementierung des Bayes-Filters. Dabei werden die Verteilungen, die eine beliebige Form annehmen können, durch eine feste Anzahl M an Werten approximiert. Diese Werte werden im Folgenden als Partikel bezeichnet. Die Anzahl der Partikel muss je nach Anwendung entsprechend angepasst werden. Allgemein gilt, dass die Berechnungszeit und die Wahrscheinlichkeit einen wahren Systemzustand zu schätzen von der Anzahl der

Partikel abhängt. Je mehr Partikel genutzt werden, desto höher ist die Wahrscheinlichkeit einen wahren Systemzustand zu schätzen. Doch gleichzeitig steigt auch die nötige Rechenzeit. Die Anzahl der Partikel M liegt meist im Bereich von 100 bis 1000 Partikeln.

Ziel des Partikel-Filters ist es die a posteriori Wahrscheinlichkeitsdichte (Gleichung 2.22) aus dem Innovationsschritt des Bayes-Filters (Gleichung 2.25) nachzubilden. Dabei repräsentiert jedes Partikel eine mögliche Hypothese der Zustandsgröße \vec{x} . Die Partikelmenge zum Zeitpunkt k wird entsprechend Gleichung 5.1 definiert.

$$\hat{\chi}_k = \{\hat{\chi}_k^{[m]} : m \in [1, M]\} \quad (5.1)$$

Für die Partikel gilt

$$\hat{\chi}_k^{[m]} \sim \underbrace{p(\vec{x}_k | \vec{y}_{1:k}, \vec{u}_{1:k-1})}_{\text{Gleichung 2.22}} \quad (5.2)$$

Die Nachbildung der Wahrscheinlichkeitsverteilung wird mit der Monte-Carlo-Integration erreicht.

Die Monte-Carlo-Integration ermöglicht es Integrale numerisch zu berechnen. Speziell lässt sich dies anwenden auf die Berechnung des Erwartungswertes einer Funktion $f(x)$ mit einer Zufallsvariablen x und der Wahrscheinlichkeitsdichte $p(x)$. Dabei gilt

$$E\{f(x)\} = \int f(x)p(x)dx \approx \frac{1}{M} \sum_{m=1}^M f(x^m) \quad (5.3)$$

Es werden M Stützstellen (x^m), die entsprechend der Wahrscheinlichkeitsdichte $p(x)$ verteilt sind, gezogen und der Mittelwert über die Funktionswerte an den Stützstellen berechnet. Falls $M \rightarrow \infty$ gilt, konvergiert der Mittelwert gegen den wahren Erwartungswert.

Beim Partikel-Filter ist es erforderlich aus einer Partikelmenge, die die Wahrscheinlichkeitsdichte $p(\vec{x}_{k-1} | \vec{y}_{1:k-1}, \vec{u}_{1:k-2})$ beschreibt, eine neue Partikelmenge zu erzeugen, die die Wahrscheinlichkeitsdichte $p(\vec{x}_k | \vec{y}_{1:k}, \vec{u}_{1:k-1})$ approximiert. Das weitere Vorgehen soll am Beispiel des Condensation-Algorithmus von Michael Isard et al. [50] beschrieben werden. Hier erhält jedes Partikel einen Gewichtungsfaktor w , welcher im Bayes-Filter der approximierten a posteriori Wahrscheinlichkeit entspricht. Für die Gewichte gilt

$$\sum_{m=1}^M w_k^{[m]} = 1 \quad (5.4)$$

Der Condensation-Algorithmus gliedert sich in drei Schritte, wobei davon ausgegangen wird, dass eine Initialschätzung zum Zeitpunkt $k = 0$ vorliegt:

1. Schritt

Ziehe mit Zurücklegen M Partikel, entsprechend ihrer Wahrscheinlichkeit, aus der Partikelmenge $\hat{\chi}_{k-1}$, welche den letzten Schätzwert darstellt. Dies entspricht zugleich dem Ziehen nach den Gewichten. Durch das Ziehen mit Zurücklegen können Partikel mit höherem Gewicht auch häufiger gezogen werden. Es entsteht eine neue Partikelmenge $\hat{\chi}'_k$.

2. Schritt

Prädiziere die Partikelmenge $\hat{\chi}'_k$ entsprechend dem Bewegungsmodell aus Gleichung 2.20 und füge ein Rauschen hinzu. Es entsteht eine neue Partikelmenge χ_k^*

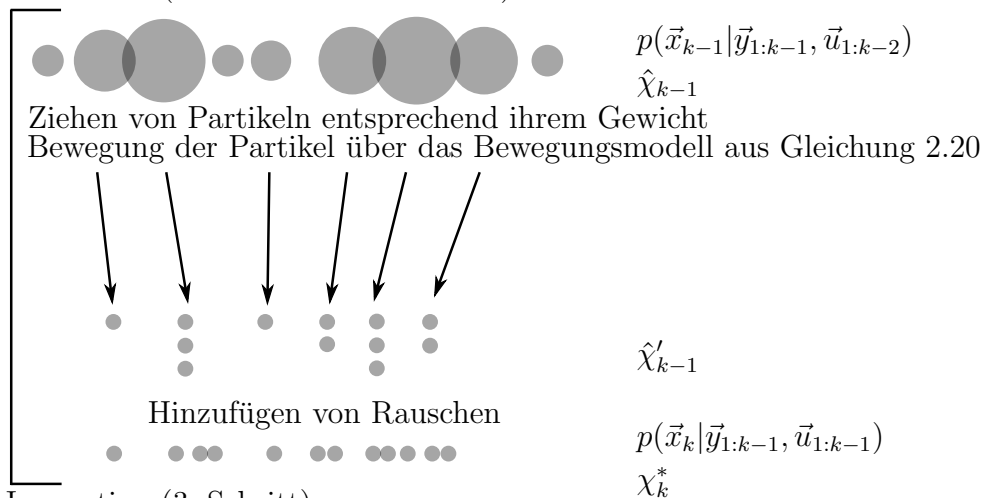
3. Schritt

Bringe die aktuelle Messung mit ein und gewichte die Partikel in der Partikelmenge χ_k^* . Je nach Art der Gewichtung ist am Ende eine Normierung notwendig, sodass die Bedingung aus Gleichung 5.4 erfüllt ist. Es entsteht die Partikelmenge $\hat{\chi}_k$.

Abbildung 5.1 fasst die drei Schritte des Condensation-Algorithmus grafisch zusammen.

Weitere Beschreibungen des Partikel-Filters finden sich beispielsweise bei Sebastian Thrun et al. [88], Michael Isard et al. [50], Dieter Fox [31], Mirko Mählisch [59] oder bei M. Sanjeev Arulampalam et al. [3].

Prädiktion (1. Schritt und 2. Schritt)



Innovation (3. Schritt)



Abbildung 5.1: Grafische Zusammenfassung des Condensation-Algorithmus (Grafik in Anlehnung an entsprechendes Schaubild in [50])

5.3 Objekt-lokale Gittermodelle

Ein objekt-lokales dreidimensionales Gittermodell G repräsentiert jeweils die Form eines Objektes. Es setzt sich aus einzelnen Würfeln zusammen. Im Weiteren sollen die Würfel als Zellen bezeichnet werden, welche jedoch die Eigenschaft eines Würfels, dass alle Kanten gleich lang sind, beibehalten. Die Kantenlänge wird durch κ beschrieben und hier als $\kappa = 0.1m$ angenommen. Es wird vorausgesetzt, dass die Länge l , die Höhe h und die Breite b des Gittermodells jeweils ein Vielfaches der Kantenlänge κ darstellen. Die mathematische Definition eines Gittermodells ist in Gleichung 5.5 beschrieben.

$$G : \Gamma \rightarrow \mathbb{R}^3 \quad (5.5)$$

$$\Gamma = \{(i\kappa, j\kappa, q\kappa) \mid i \in N_x, j \in N_y, q \in N_z\},$$

$$N_x = \{1, \dots, n_x\}, N_y = \{1, \dots, n_y\}, N_z = \{1, \dots, n_z\}$$

mit

$$n_x = \frac{l}{\kappa}, n_y = \frac{b}{\kappa}, n_z = \frac{h}{\kappa} \quad (5.6)$$

Das Gittermodell G definiert jeweils die Mitte einer jeden Zelle.

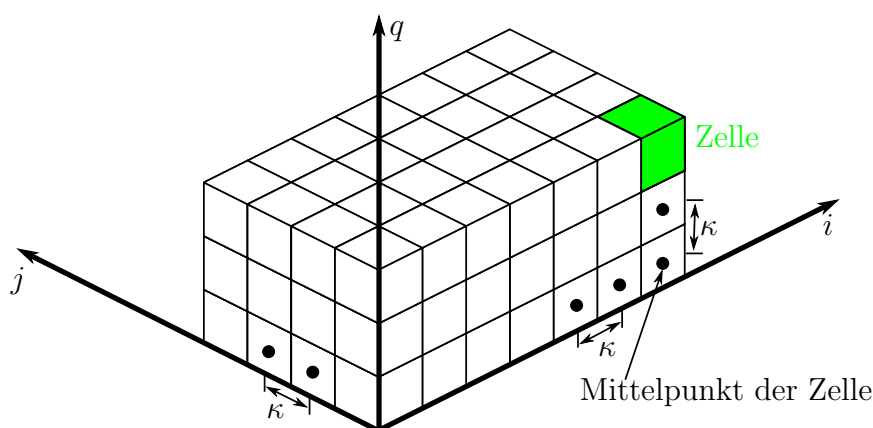


Abbildung 5.2: Definition eines Gittermodells

Die Funktion G zählt die Anzahl der Messungen, die in eine Zelle fallen. Bei der Verwendung eines Lasersensors kann dieser Ansatz gewählt werden, wenn die Kantenlänge κ gegenüber der Strahlaufweitung eines Laserstrahls als genügend groß angenommen werden kann. Zu Beginn sind alle Werte in G auf Null gesetzt.

Der in '5.2 Partikel-Filter' beschriebene Partikel-Filter wird um die Funktionalität erweitert, dass ein Partikel nicht nur den Zustand des Objekts über den Zustandsvektor repräsentiert, sondern auch die Form des Objekts über ein Gittermodell speichert. Demnach entspricht ein Partikel einem Gittermodell. Jedes Gittermodell stellt damit eine Hypothese der Position, Orientierung und Form eines Objektes dar.

5.4 Kostenfunktionen

Wenn neue Messungen vorliegen, werden diese in die objekt-lokalen Gittermodelle integriert.

Bei der Integration können unterschiedliche Kostenfunktionen berechnet werden, die im späteren Verlauf im Partikel-Filter für die Gewichtung der Partikel verwendet werden. In den folgenden Unterkapiteln werden fünf, im Rahmen dieser Arbeit entwickelte, Kostenfunktionen erläutert und verglichen, um die in diesem Fall effizienteste Gesamtkostenfunktion zu ermitteln. Bei den Kostenfunktionen handelt es sich um: Integrationsverhältnis, Besetzungsverhältnis, Besetzungsgradverhältnis, Besetzungsdistanz und Schwerpunktdistanz. Alle Kostenfunktionen werden auf das Intervall $[0, 1]$ skaliert. Ein Wert nahe Null entspricht demnach geringen Kosten, wobei ein Wert nahe Eins mit hohen Kosten verbunden ist.

Die Kostenfunktionen werden sowohl mathematisch als auch über eine schematische Zeichnung in den folgenden Abschnitten dargestellt. Die schematische Zeichnung ist, um die Visualisierung möglichst einfach zu halten, als zweidimensionale Projektion ausgeführt. Alle Kostenfunktionen beachten jedoch das dreidimensionale Gittermodell.

5.4.1 Integrationsverhältnis

Bei einem objekt-lokalen Gittermodell wird von einem sich in der Größe nicht ändernden Modell ausgegangen. Aus diesem Grund wird eine Kostenfunktion definiert, die die Anzahl der Messungen, die im Bereich des Gittermodells liegen, in das Verhältnis mit der Gesamtzahl der Messungen setzt. Gleichung 5.7 beschreibt die Kostenfunktion mathematisch. Dabei ist nI die Anzahl der Messungen, die innerhalb des objekt-lokalen Gittermodells liegen, und N die Gesamtzahl der Messungen.

$$c_{IV} = 1 - \frac{nI}{N} \quad (5.7)$$

Abbildung 5.3 zeigt die Kostenfunktion des Integrationsverhältnisses graphisch.

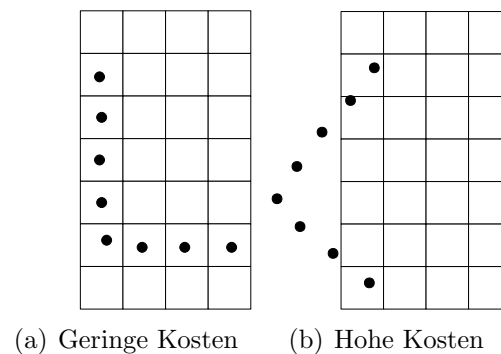


Abbildung 5.3: Kostenfunktion: Integrationsverhältnis. Die zu integrierende Messungen sind als schwarze Punkte dargestellt.

5.4.2 Besetzungsverhältnis

In aufeinanderfolgenden Messungen werden ähnliche Stellen am Objekt gemessen. Diese Überlegung führt zur Kostenfunktion des Besetzungsverhältnisses. Übertragen auf das Gittermodell bedeutet dies, dass in zwei aufeinanderfolgenden Messungen die neu zu integrierenden Messungen in schon vom vorhergehenden Schritt besetzte Zellen fallen müssen. Je mehr Messungen in schon besetzte Zellen fallen, desto besser ist das Resultat und desto geringer sind die Kosten.

Um einen vergleichbaren Wert über alle Hypothesen herstellen zu können, bedarf es eines Gewichtungsfaktors λ . λ ist die maximale Anzahl an schon besetzten Zellen, über alle Hypothesen hinweg, die von neuen Messungen getroffen werden. nO ist die Summe über alle Messungen, die in der aktuellen Hypothese in schon besetzte Zellen fallen.

Die Kostenfunktion ergibt sich nach Gleichung 5.8.

$$c_{BV} = \frac{nO}{\lambda} \quad (5.8)$$

Abbildung 5.4 veranschaulicht die Kostenfunktion des Besetzungsverhältnisses.

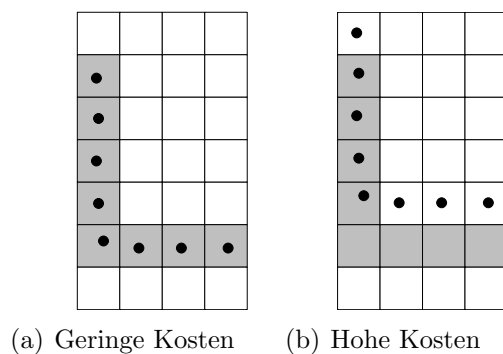


Abbildung 5.4: Kostenfunktion: Besetzungsverhältnis. Grau hinterlegte Zellen symbolisieren schon durch vorherige Messungen besetzte Zellen.

5.4.3 Besetzungsgradverhältnis

Die Annahme, dass in aufeinanderfolgenden Messungen ähnliche Stellen am Objekt gemessen werden, wird wie bei '5.4.2 Besetzungsverhältnis' auch für das Besetzungsgradverhältnis angewendet. Zusätzlich zu der Forderung, dass eine Zelle bereits besetzt ist, erwartet die Kostenfunktion, dass diese Zelle mit möglichst vielen Messungen besetzt ist.

sI ist die Summe über alle Messungen, die sich bereits in den Zellen befinden. NI ist die maximale Anzahl der bereits in eine Zelle integrierten Messungen über alle Hypothesen hinweg. Die Kostenfunktion ist in Gleichung 5.9 definiert.

$$c_{BGV} = 1 - \frac{sI}{NI} \quad (5.9)$$

Abbildung 5.5 zeigt die Kostenfunktion des Besetzungsgradverhältnisses schematisch.

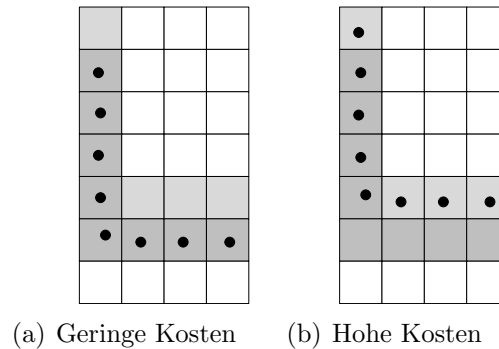


Abbildung 5.5: Kostenfunktion: Besetzungsgradverhältnis. Dunkelgrau hinterlegte Zellen zeigen an, dass bereits mehr Messungen in diese als in die hellgrau hinterlegten Zellen gefallen sind.

5.4.4 Besetzungsdistanz

Im Gegensatz zu den Kostenfunktionen Integrationsverhältnis ('5.4.1 Integrationsverhältnis'), Besetzungsverhältnis ('5.4.2 Besetzungsverhältnis') und Besetzungsgradverhältnis ('5.4.3 Besetzungsgradverhältnis'), die fordern, dass eine Messung exakt in eine Zelle fallen muss, beachtet die Besetzungsdistanz die weitere Umgebung um die Messung herum. Für die Kostenfunktion der Besetzungsdistanz werden die kürzesten Abstände ($dMin$) zwischen den Messungen und den nächsten besetzten Zellen summiert. Die Gewichtung erfolgt über die maximale gefundene Distanz ($dMax$) über alle Hypothesen. Die Kostenfunktion ergibt sich nach Gleichung 5.10.

$$c_{BD} = \frac{dMin}{dMax} \quad (5.10)$$

Abbildung 5.6 zeigt die Kostenfunktion der Besetzungsdistanz grafisch.

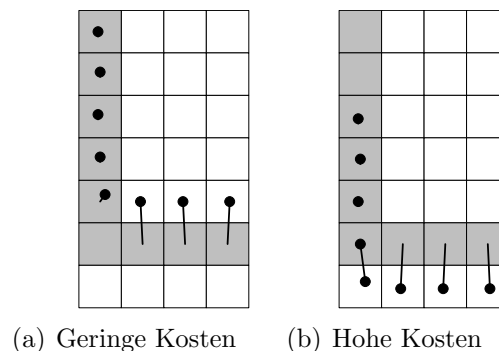


Abbildung 5.6: Kostenfunktion: Besetzungsdistanz. Grau hinterlegte Zellen zeigen an, dass sie bereits durch Messungen belegt sind. Die schwarzen Verbindungen zwischen den Zellen und den Messungen visualisieren die Distanz.

5.4.5 Schwerpunktdistanz

Die Kostenfunktion der Schwerpunktdistanz stellt eine Verstärkung des Integrationsverhältnisses ('5.4.1 Integrationsverhältnis') dar. Es wird die Forderung gestellt, dass sich der Schwerpunkt der Messungen $cog = (\frac{1}{N} \sum_{i=1}^N x_i, \frac{1}{N} \sum_{i=1}^N y_i, \frac{1}{N} \sum_{i=1}^N z_i)$ innerhalb des Gittermodells befindet. Dabei ist N die Anzahl der Messungen und die Vektoren $\vec{x}, \vec{y}, \vec{z} \in \mathbb{R}^N$ speichern die Messdaten entlang der zugehörigen Achsen. Die Funktion $inGrid(\cdot)$ gibt den Wert 1 zurück, falls sich der Schwerpunkt innerhalb des Gittermodells befindet, und den Wert 0, falls er außerhalb liegt.

$$c_{SD} = \begin{cases} 1 & \text{wenn } inGrid(cog) == 0 \\ 0 & \text{sonst} \end{cases} \quad (5.11)$$

Abbildung 5.7 zeigt die Kostenfunktion der Schwerpunktdistanz grafisch.

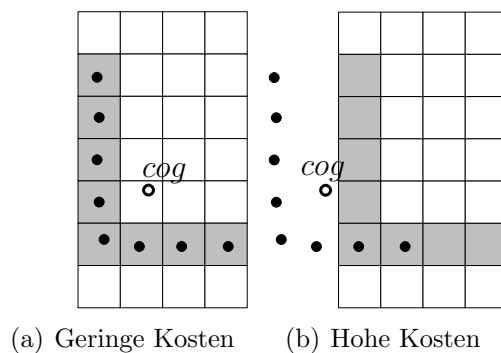


Abbildung 5.7: Kostenfunktion: Schwerpunktdistanz. Grau hinterlegte Zellen zeigen an, dass sie bereits durch Messungen belegt sind. Der Schwerpunkt ist durch einen nicht ausgefüllten schwarzen Kreis dargestellt.

5.4.6 Vergleich und Auswahl der Kostenfunktionen

Die zuvor beschriebenen Kostenfunktionen folgen alle der Forderung, dass keine Informationen über das zu verfolgende Objekt vorhanden sein müssen. Zu beachten ist jedoch, dass auch Kostenfunktionen beschrieben wurden, die in ihren Eigenschaften ähnlich sind.

Abbildung 5.8 zeigt für eine Iteration die Kostenfunktionen. Der gezeigte Verlauf ist repräsentativ für alle Iterationen im später untersuchten Datensatz in '5.9 Auswertung'. Für das Beispiel sind 400 Partikel verwendet worden, die verschiedene Hypothesen für den Zustand des zu verfolgenden Objekts repräsentieren. Nach der Integration von mehreren Messungen ergeben sich für jedes Partikel entsprechend den Kostenfunktionen fünf Werte, die die jeweiligen Kosten darstellen. Für die Darstellung wurden die fünf Werte, die sich aus den Kostenfunktionen für jedes Partikel ergeben haben, aufsummiert. Die aufsummierten Werte wurden daraufhin aufsteigend sortiert. Um eine einfache Darstellung zu haben, wurde auf die Zuordnung der ursprünglichen und eindeutigen Partikelidentifikation verzichtet. Somit hat jedes Partikel eine neue Identifikation nach der Sortierung erhalten.

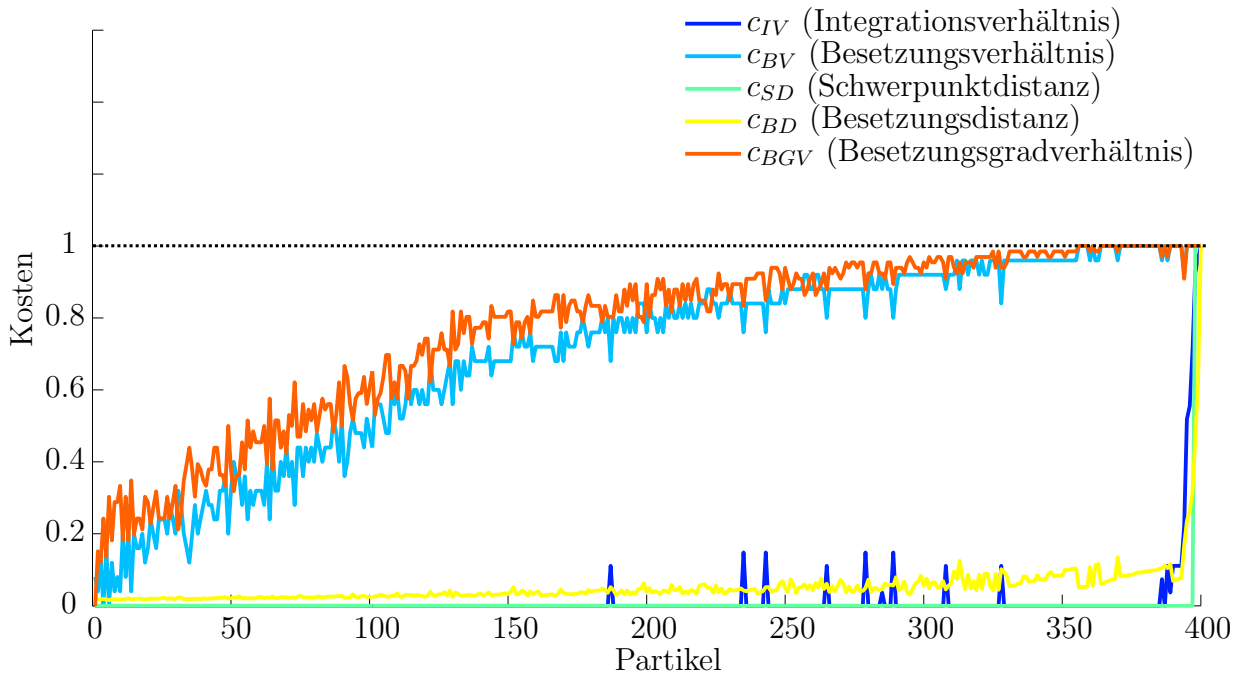


Abbildung 5.8: Vergleich der Kostenfunktionen

Der Vergleich der Kostenfunktionen zeigt, dass Ähnlichkeiten in der Ausprägung der Kostenfunktionen vorhanden sind. Die Kosten für das Besetzungsgradverhältnis (’5.4.3 Besetzungsgradverhältnis’) und das Besetzungsverhältnis (’5.4.2 Besetzungsverhältnis’) entwickeln sich nahezu identisch. Die übrigen Kostenfunktionen weisen kaum mittlere Kosten auf. Bei ihnen lässt sich eine Tendenz zu Extremwerten erkennen.

Aufgrund dieser Daten wird die Gesamtkostenfunktion aus den Kostenfunktionen Besetzungsgradverhältnis, Besetzungsverhältnis und Integrationsverhältnis gebildet. Sie decken den gesamten Wertebereich im Kostenintervall $[0, 1]$ ab. Der Einfluss des Integrationsverhältnisses erweist sich zwar als nicht sehr groß, aber die entsprechende Kostenfunktion wird dennoch in die Gesamtkostenfunktion aufgenommen, da die Anzahl der integrierten Messungen ein wichtiger Hinweis auf möglicherweise schlechte Zustandsschätzungen ist. Die Schwerpunktdistanz wird nicht verwendet, da sie im Allgemeinen nur extreme Ausreißer filtert. Die Besetzungsdistanz entfällt aufgrund des hohen Rechenaufwands für die Distanzberechnungen.

Die Gesamtkostenfunktion, die im weiteren Verlauf Verwendung findet, ergibt sich nach Gleichung 5.12.

$$c_G = \frac{c_{BGV} + c_{BV} + c_{IV}}{3} \tag{5.12}$$

5.5 Bewegungsmodell

Alle Gittermodelle sind mit einem Bewegungsmodell verbunden. Um den jeweils nächsten Zustand von jedem objekt-lokalen Gittermodell entsprechend der Zykluszeit zu präzisieren, wird ein Einspurmodell verwendet, das einer Approximation der Bewegung eines Fahrzeugs entspricht. Dabei wird davon ausgegangen, dass sich das Fahrzeug auf einer Kreisbahn bewegt. Abbildung 5.9 zeigt das Einspurmodell mit den beschreibenden Größen.

Für die Bewegung des Modells werden zwei Eingangswerte verwendet. Der erste Wert v_R gibt die Tangentialgeschwindigkeit des Hinterrades an, der zweite Wert δ ist der Lenkwinkel. Der Punkt mit der Bezeichnung M stellt den Momentanpol dar, um den sich das Fahrzeug in einer Zykluszeit dreht. $C(x_C, y_C)$ ist der Schwerpunkt des Fahrzeugs in globalen Koordinaten und wird in der Mitte des objekt-lokalen Gittermodells angenommen. Die Geschwindigkeit des Fahrzeugschwerpunktes $C(x_C, y_C)$ wird mit v_C bezeichnet. Der Orientierungswinkel des objekt-lokalen Gittermodells wird im globalen Koordinatensystem mit Ψ bezeichnet. Bei der Bewegung der Objekte wird angenommen, dass sie sich mit einer konstanten Geschwindigkeit bewegen. Die Beschleunigung wird durch ein Rauschen s_{ar} modelliert. Für den Lenkwinkel gilt eine ähnliche Annahme. Es wird angenommen, dass der Lenkwinkel konstant bleibt. Die Abweichung davon wird über ein Rauschen s_δ modelliert.

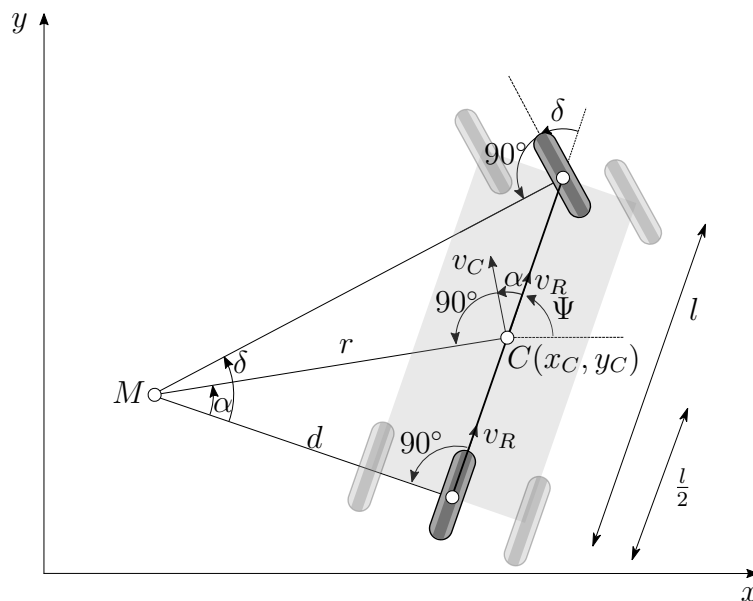


Abbildung 5.9: Einspurmodell

Für die Berechnung der statischen Größen des Einspurmodells gelten die folgenden Zusammenhänge:

$$d = \frac{l}{\tan(\delta)} \quad (5.13)$$

$$\alpha = \arctan\left(\frac{l}{2d}\right) \quad (5.14)$$

$$r = \sqrt{d^2 + \left(\frac{l}{2}\right)^2} = d\sqrt{1 + \left(\frac{l}{2d}\right)^2} = d\sqrt{1 + \tan^2(\alpha)} \quad (5.15)$$

$$v_C = \frac{v_R}{d}r = v_R\sqrt{1 + \frac{\tan^2(\delta)}{4}} \quad (5.16)$$

Die dynamischen Größen ergeben sich wie im Folgenden dargestellt. Dabei entsprechen \dot{x} und \dot{y} den Geschwindigkeiten in x- und y-Richtung, v_R der Beschleunigung in tangentialer Richtung, $\dot{\delta}$ der Lenkgeschwindigkeit und $\dot{\Psi}$ der Winkelgeschwindigkeit um die z-Achse.

$$\dot{x} = v_C \cos(\Psi + \alpha) = v_R \sqrt{1 + \frac{\tan^2(\delta)}{4}} \cos\left(\Psi + \arctan\left(\frac{\tan(\delta)}{2}\right)\right) \quad (5.17)$$

$$\dot{y} = v_C \sin(\Psi + \alpha) = v_R \sqrt{1 + \frac{\tan^2(\delta)}{4}} \sin\left(\Psi + \arctan\left(\frac{\tan(\delta)}{2}\right)\right) \quad (5.18)$$

$$v_R = s_{ar} \quad (5.19)$$

$$\dot{\delta} = s_{\delta} \quad (5.20)$$

$$\dot{\Psi} = \frac{v_r \tan(\delta)}{l} \quad (5.21)$$

5.6 Fahrzeugdynamikanalyse

Zur Bestimmung der Ausprägung des Modellrauschens, das bereits in '5.5 Bewegungsmodell' eingeführt wurde, werden im Folgenden typische Beschleunigungen und Lenkwinkelgeschwindigkeiten bestimmt. Die zu Grunde liegenden Daten stammen aus einer Datenbank mit verschiedenen Strecken. Die Strecken ergeben eine Gesamtlänge von mehreren hundert Kilometern, die sich hauptsächlich aus ausgebauten Landstraßen zusammensetzen. Innerstädtische Straßen sind in geringem Maße ebenfalls vertreten. Die Strecken decken die Länder Deutschland, Italien und Österreich ab.

Die Beschleunigungen werden bezüglich einer Sensorzykluszeit von 0.1s ausgewertet. Dies entspricht der Messfrequenz des Velodyne HDL-64E, der für die weitere Auswertung genutzt werden wird. Bei gemäßigter Fahrt ist zu erwarten, dass sich die Beschleunigungen in positive und negative Richtung ähnlich verhalten werden. Des Weiteren wird die Stärke der jeweiligen Beschleunigungen mit steigender Geschwindigkeit stark abnehmen. In einigen Arbeiten, wie beispielsweise bei Michael Manz et al. [62], wird die Verteilung daher als normalverteilt angenommen. Entsprechend Abbildung 5.10 bestätigt sich die Annahme der Normalverteilung. In jeder Zeile der Grafik ist ein Histogramm über die

Wahrscheinlichkeit entsprechend der aktuellen Geschwindigkeit aufgetragen. Die Summe der Werte in jeder Zeile ergibt den Wert eins.

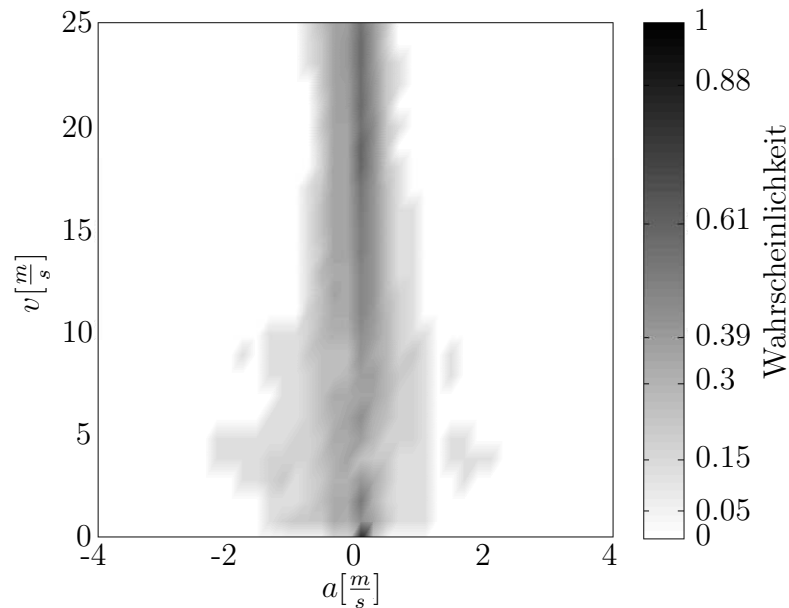


Abbildung 5.10: Verteilung der Beschleunigungen

Dieselben Annahmen wie zuvor bei den Beschleunigungen treffen auch auf die Lenkwinkelgeschwindigkeit zu. Es wird ebenfalls eine Normalverteilung erwartet. Für die Auswertung wird die laterale Beschleunigung a_L verwendet, um den Lenkwinkel δ des Einspurmodells aus Abbildung 5.9 zu bestimmen. Der Lenkwinkel ergibt sich mit Gleichung 5.13 wie folgt:

$$a_L = \frac{v_R^2}{d} \quad (5.22)$$

$$a_L = \frac{v_R^2 \tan(\delta)}{l} \quad (5.23)$$

$$\tan(\delta) = \frac{a_L l}{v_R^2} \quad (5.24)$$

$$\delta = \arctan\left(\frac{l}{v_R^2} a_L\right) \quad (5.25)$$

Abbildung 5.11 zeigt die Verteilung der Lenkwinkelgeschwindigkeit für die jeweilig passende Geschwindigkeit. Wie auch bei der Verteilung der Beschleunigung gilt, dass jede Zeile sich zu eins summiert. Es bestätigt sich auch hier die Annahme der Normalverteilung, die für jede Geschwindigkeitszeile gilt.

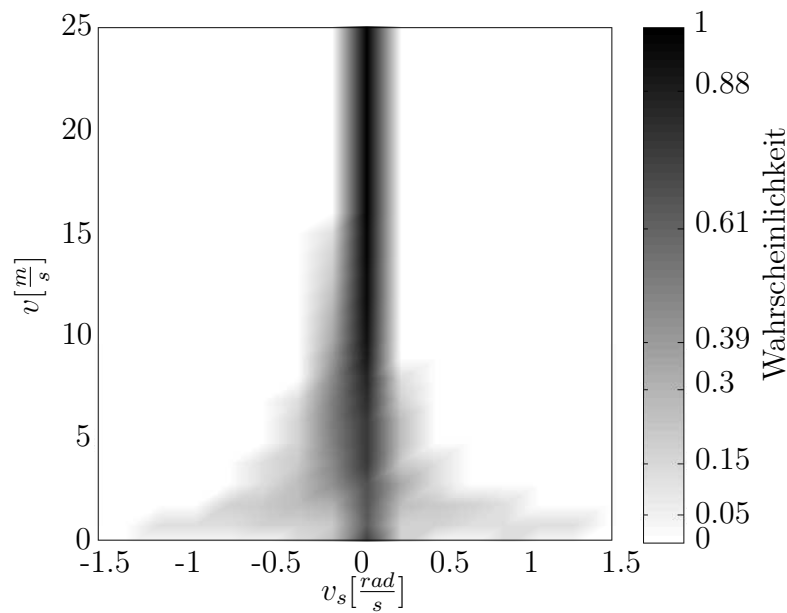


Abbildung 5.11: Verteilung der Lenkgeschwindigkeiten

5.7 Überblick über den Algorithmus

Der Algorithmus zur Objektverfolgung mit einem objekt-lokalen Gittermodell besteht aus verschiedenen einzelnen Komponenten, die in den vorherigen Abschnitten beschrieben wurden. Im Folgenden werden die Komponenten in ihrer Funktionsweise zusammengefasst und der gesamte Ablauf des Algorithmus wird veranschaulicht.

Abbildung 5.12 zeigt den Ablauf des Algorithmus in Form eines Blockschaltbildes. Zu Beginn erfolgt der Initialisierungsprozess. Dabei werden die Blöcke 1, 2, 3, 4, 5, 6 in der angegebenen Reihenfolge durchlaufen. Im ersten Block erfolgt die Sensordatengewinnung durch den Sensor. Diese Daten werden im zweiten Block über Methoden, die in Kapitel '3 Segmentierung' vorgestellt wurden, segmentiert. Der dritte Block führt eine Datenassoziation durch, diese wird jedoch im Initialisierungsprozess nicht durchgeführt, da noch kein Objekt zum aktuellen Zeitpunkt verfolgt wird. Im vierten Block werden die Daten so aufbereitet, dass sie an den Partikel-Filter übergeben werden können. Der fünfte Block führt den Initialisierungsschritt aus. Dabei werden die Partikel (Gittermodelle) erzeugt und an die Stelle der ersten Messung bewegt. Im sechsten Block erfolgt die Zustandsschätzung durch eine Mittelwertbildung über die Zustandsvektoren der einzelnen Partikel.

Nach dem Initialisierungsprozess beginnt der Algorithmus in einer Schleife zu laufen. Dabei werden jeweils die Blöcke in folgender Reihenfolge durchlaufen: 6, 7, 1, 2, 3, 4, 8, 9. Block sechs führt die Zustandsschätzung durch. In Block sieben werden die Partikel, bzw. die Gittermodelle entsprechend dem Bewegungsmodell in die Zukunft prädictiert. Danach folgen die bereits beim Initialisierungsprozess beschriebenen Blöcke eins, zwei

und drei. Im dritten Block erfolgt die Datenassoziation. Dabei wird in den segmentierten Daten nach dem globalen nächsten Nachbarn zum aktuellen Zustandsvektor gesucht. Diese Daten werden im folgenden Block (Block vier) für die Weitergabe aufbereitet. Es folgt in Block acht die Gewichtung der Partikel, bzw. Gittermodelle über die eingebrachte neue Messung. Dabei werden die neuen Messungen nach der Bewertung über die Kostenfunktionen in das Gittermodell integriert. Über einen einstellbaren Parameter ist es zusätzlich möglich Zellen mit wenigen Messungen über die Zeit löschen zu lassen. Dieser Parameter kann als Vergessens-Parameter aufgefasst werden und dient dazu Messungen, die über eine falsche Hypothese integriert wurden, zu entfernen. In Block neun werden die besten ν Prozent der Partikel, bzw. Gittermodelle ausgewählt und für den weiteren Verlauf des Algorithmus weiterverwendet. ν stellt dabei eine frei wählbare Konstante dar. Die übrigen Partikel werden gelöscht und es werden auf Grundlage der besten ν Prozent neue Partikel erzeugt. Der Fall der Partikelverarmung (Engl. impoverishment), bei dem es passieren kann, dass die Partikelmenge auf einige wenige Partikel degeneriert, wird hier nicht weiter untersucht. Ein Überblick zu Lösungen des Problems findet sich bei Li et al. [56].

Das Ergebnis der Zustandsschätzung aus Block sechs stellt die Eingabe für eine darauf aufbauende Anwendung dar.

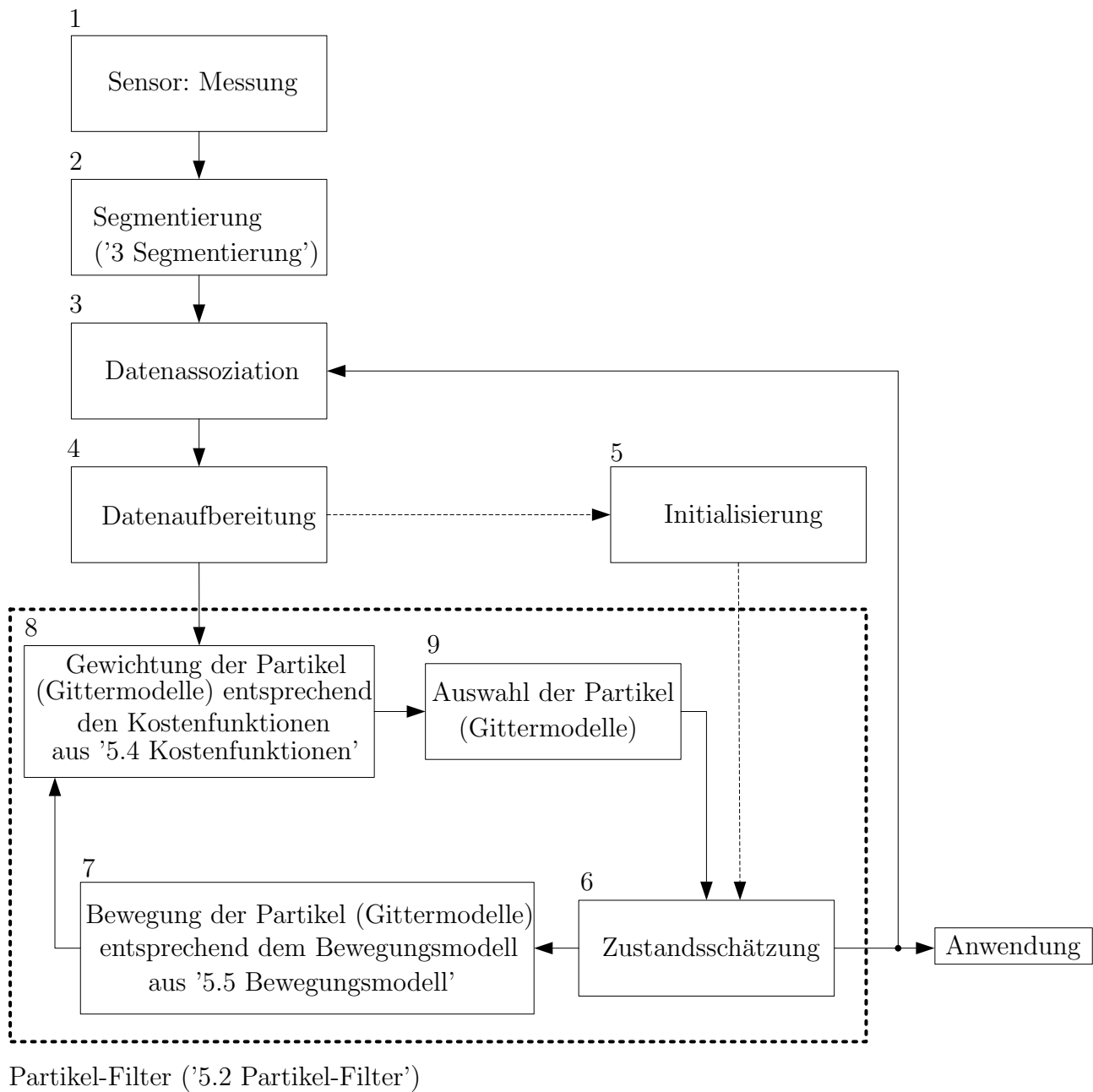


Abbildung 5.12: Ablauf des Algorithmus zur Objektverfolgung mit einem objekt-lokalen Gittermodell

5.8 Akkumulation von Messdaten

Im Verlauf der Objektverfolgung über die lokalen Gittermodelle werden mit jedem Zeitschritt neue Messungen integriert. Dies führt dazu, dass sich in jedem lokalen Gittermodell eine Hypothese über eine dreidimensionale Form des verfolgten Objekts aufbaut. Im nachfolgenden Abschnitt ('5.9 Auswertung') erfolgt die Auswertung des vorgestellten Algorithmus. Um die Akkumulation von Messdaten zu veranschaulichen, soll hier auf

ein Teilergebnis vorgegriffen werden. Abbildung 5.13 zeigt beispielhaft ein objekt-lokales Gittermodell nach der Integration von mehreren Messungen eines Fahrzeugs, das vom Sensor aus verschiedenen Perspektiven gesehen wurde. Die frei gebliebenen Stellen zwischen den rot und blau markierten Bereichen sind die Fenster des Fahrzeugs von denen keine Reflexionen zurück gekommen sind.

Weiterführend ist auch ein Verfeinerungsschritt bei der Integration der Messdaten in das lokale Gittermodell denkbar. Hierzu kann beispielsweise der Iterative-Closest-Point-Algorithmus (Kapitel '4.15.3.1 (Weighted) Iterative Closest Point Algorithmus ((W)ICP)') eingesetzt werden. Untersuchungen haben jedoch ergeben, dass dadurch die Ergebnisse teilweise verschlechtert werden. In einigen Fällen kommt es vor, dass die zu integrierenden Messungen mit ihrer initialen Positions- und Orientierungsschätzung um einige zehn Grad um die z-Achse durch den ICP-Algorithmus gedreht werden. Grund ist die Kostenfunktion des ICP-Algorithmus, die auf der Minimierung der quadratischen Abstände beruht. In dem geschilderten Fall kommt es zu einer Optimierung in Richtung minimaler Kosten, die jeweils aber lediglich ein lokales Minimum darstellen.

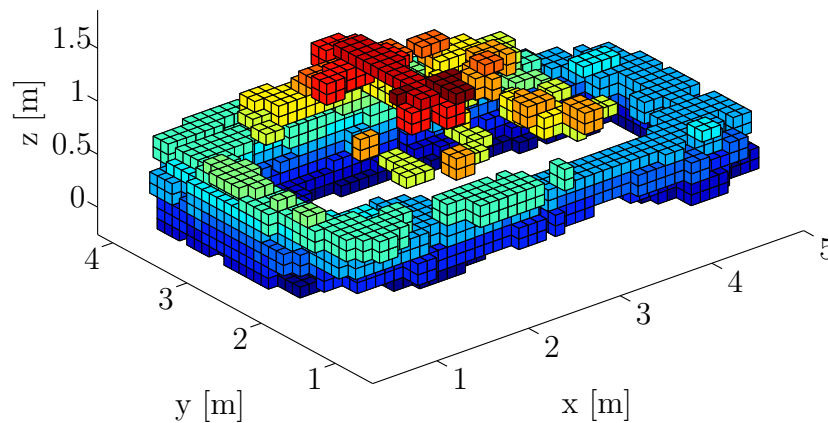


Abbildung 5.13: Akkumulierte Messungen in einem objekt-lokalen Gittermodell. Die farbliche Kodierung zeigt die Höhe der Zellen an.

5.9 Auswertung

Die Auswertung des Algorithmus zur Objektverfolgung mit objekt-lokalen Gittermodellen erfolgt auf den gleichen Daten, die auch zur Auswertung der Außenkonturbestimmung in '4.15.3.3 Auswertung' verwendet wurden. Die detaillierte Beschreibung zur Erzeugung der Referenzdaten, sowie die Eigenschaften der Messungen, finden sich im genannten Abschnitt. Für einen Vergleich der Ergebnisse mit aktuellen Verfahren sei auf die Arbeit von Matthias Schmid [80] verwiesen, der auf die gleichen Messdaten bei der Auswertung zurückgegriffen hat.

Zusammengefasst wurden die Messungen durch unterschiedliche Störquellen beeinflusst, die sich in der Qualität der Messungen niedergeschlagen haben. Die Route, die das zu

messende Fahrzeug gefahren ist, ist als versetzte Kreisfahrt mit Zu- und Abfahrt definiert. Diese ist in Abbildung 4.46 dargestellt.

Die Auswertung erfolgt mit 200 Partikeln. Die Länge, Breite und Höhe der einzelnen Gittermodelle wird auf $5m$, $4m$ und $2m$ festgelegt. Die Größe des zu verfolgenden Objekts (Mercedes-Benz E-Klasse W212) ist: $4.8m$ (Länge), $1.8m$ (Breite), $1.4m$ (Höhe). Die jeweiligen Längen der Gittermodelle wurden bewusst nicht direkt auf die Längen des zu verfolgenden Objekts angepasst, um zu zeigen, dass dabei Toleranzen möglich sind, falls die Größe des zu verfolgenden Objekts zu Beginn nicht bekannt ist. Dies ist notwendig, da aktuell keine Möglichkeit vorgesehen ist das Gittermodell in der Größe zu verändern. Die Anfangsgeschwindigkeit wird mit $5\frac{m}{s}$ angenommen.

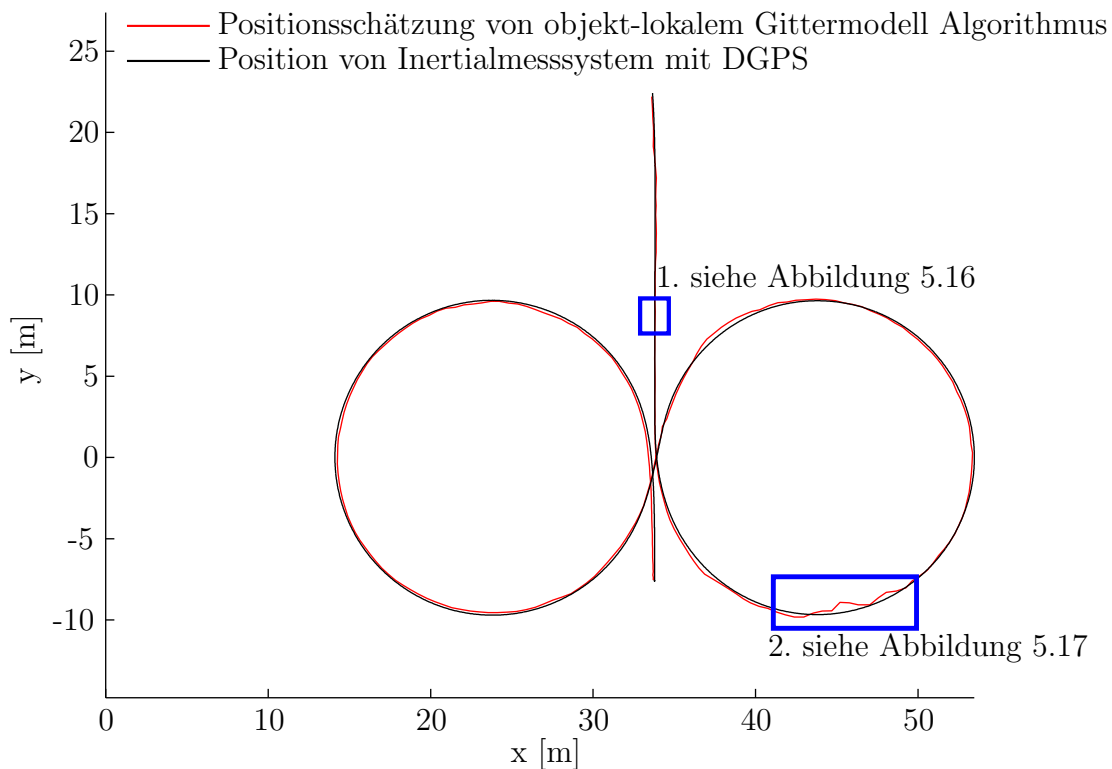


Abbildung 5.14: Vergleich der geschätzten Position des Objektes durch den objekt-lokalen Gittermodell Algorithmus und der Position aus einem Inertialmesssystem mit DGPS

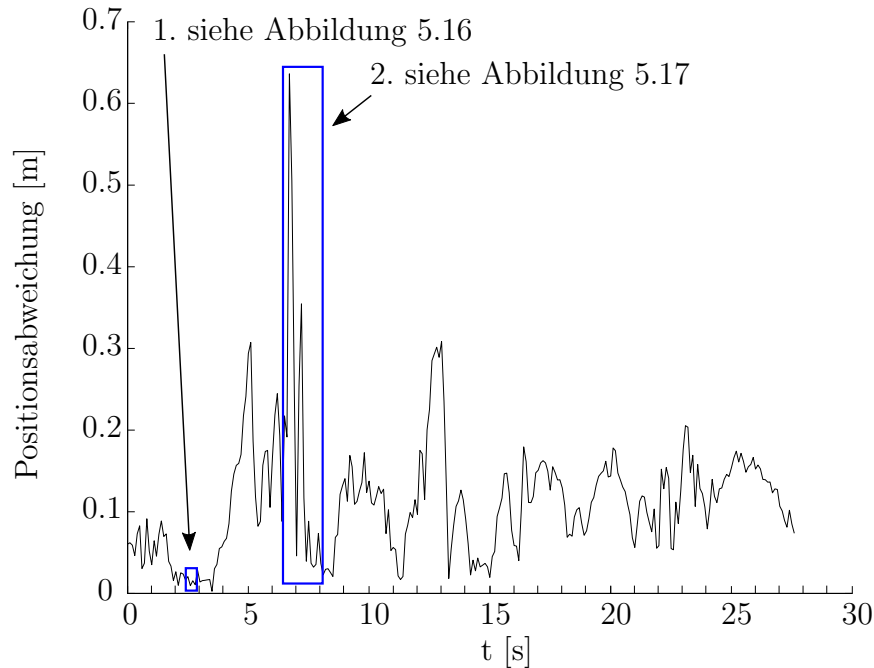


Abbildung 5.15: Positionsabweichung zwischen der realen Position und der über das objekt-lokale Gittermodell bestimmten Position

Abbildung 5.14 zeigt den Vergleich zwischen der geschätzten Position des Objektes durch den objekt-lokalen Gittermodell Algorithmus und der Position aus einem Inertialmesssystem mit DGPS. Der Sensor befindet sich fest an der Koordinate (0, 0). Der Positionsfehler, welcher sich aus dem Vektor zwischen den X- und Y-Abweichungen ergibt, ist in Abbildung 5.15 dargestellt. Der mittlere Fehlerwert (RMSE) beträgt über die gesamte Strecke $0.13m$.

In Tabelle 5.1 ist ein Vergleich mit anderen Algorithmen zur Positionsbestimmung dargestellt. In der referenzierten Arbeit wurde auf die gleichen Messdaten zurückgegriffen. Somit lassen sich die Ergebnisse direkt vergleichen.

Im Folgenden werden zwei Bereiche näher beschrieben. Diese sind in Abbildung 5.14 und Abbildung 5.15 durch 1. und 2. gekennzeichnet.

Algorithmus	Fehler (RMSE)
Lokales Gittermodell Partikel-Filter	$0.13m$
Belegungskarte Kalman-Filter [80]	$1.14m$
Belegungskarte Partikel-Filter Gauss [80]	$1.15m$
Belegungskarte Partikel-Filter Epanechnikov [80]	$1.63m$

Tabelle 5.1: Vergleich der Fehlerwerte (RMSE) für unterschiedliche Algorithmen zur Positionsbestimmung

Die erste gekennzeichnete Stelle in Abbildung 5.14 zeigt eine nahezu perfekte Übereinstimmung zwischen den beiden Positionswerten. In Abbildung 5.16 sind die zugehörige Messung und die entsprechenden Positionshypothesen dargestellt. Die Varianz der Positionshypothesen ist sehr gering, was auf eine zuverlässige Positionsschätzung hinweist. Zur Verdeutlichung der Darstellung ist schematisch ein Fahrzeugbild eingeblendet.

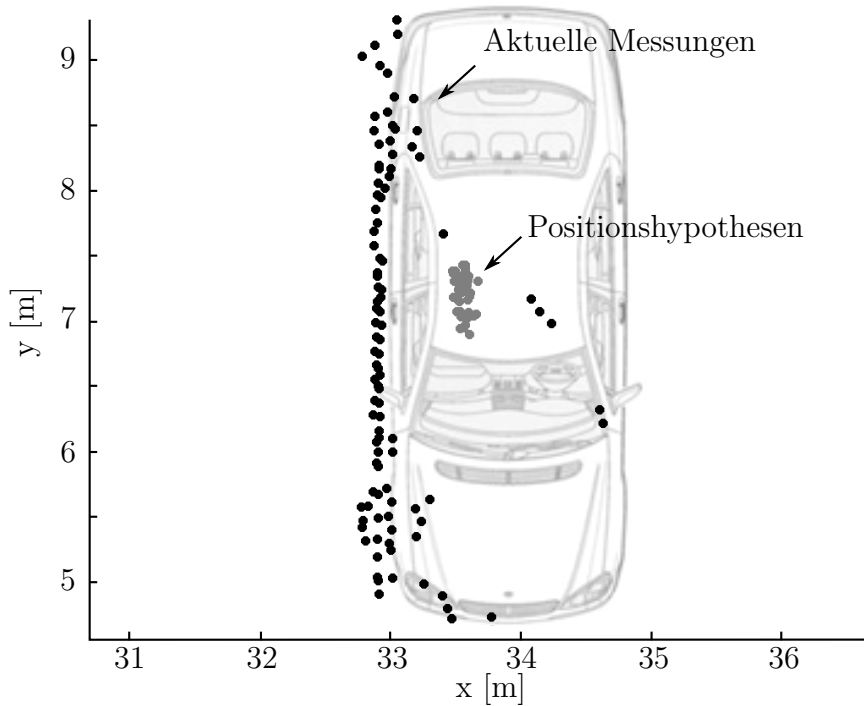


Abbildung 5.16: Messungen des Laser Sensors und Positionshypothesen zu einem Zeitschritt der im mit 1. gekennzeichneten Fenster in Abbildung 5.14 liegt.

Die zweite gekennzeichnete Stelle in Abbildung 5.14 zeigt einen Bereich mit größeren Abweichungen zwischen den beiden Positionen. Die zu integrierende Messung in Abbildung 5.17 zeigt, dass vorwiegend das Heck des Fahrzeugs gemessen wurde. Dieses lässt sich über verschiedene Hypothesen in die vorhandenen Gittermodelle integrieren. Diese Unsicherheit, wie die Messung integriert werden kann, drückt sich in einer größeren Varianz der Partikel im Vergleich zu den in Abbildung 5.16 aus. In Abbildung 5.15 ist die Positionsabweichung bei ca. $7s$ zu sehen.

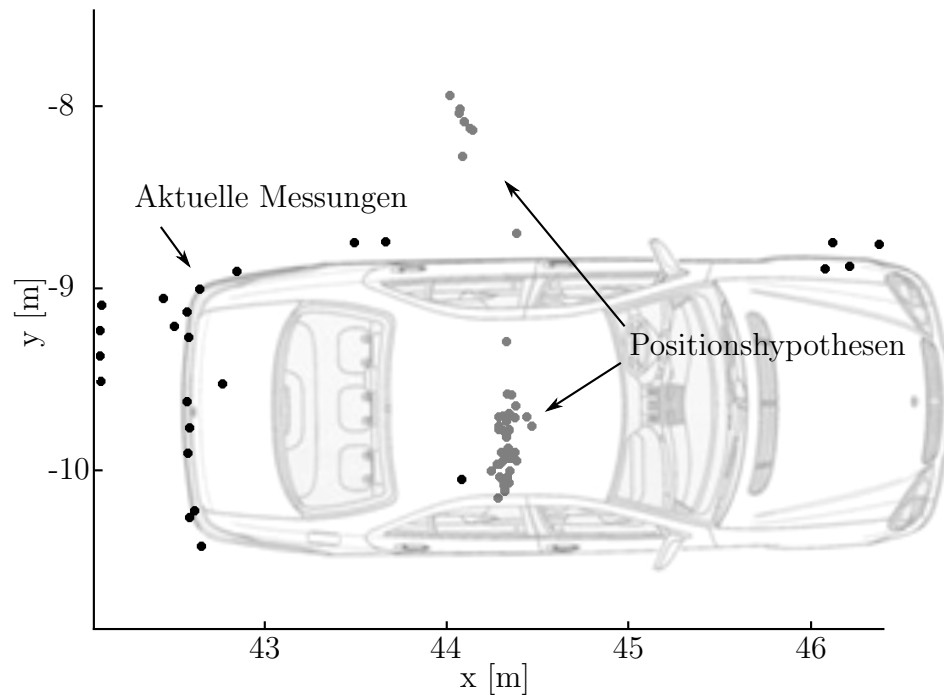


Abbildung 5.17: Messung des Laser Sensors und Positionshypothesen zu einem Zeitschritt der im mit 2. gekennzeichneten Fenster in Abbildung 5.14 liegt.

In Abbildung 5.18 ist der Vergleich zwischen der geschätzten Geschwindigkeit des objekt-lokalen Gittermodell Algorithmus und der Geschwindigkeit aus dem Inertialmesssystem mit DGPS dargestellt. Der Fehlerwert (RMSE) liegt bei $0.23 \frac{m}{s}$. Selbst im Bereich des Abbremsvorgangs treten nur geringe Abweichungen auf. Wird die Geschwindigkeit zu Beginn der Messung mit $0 \frac{m}{s}$ angenommen, ergibt sich ein Fehlerwert (RMSE) von $0.39 \frac{m}{s}$. Den Geschwindigkeitsfehler über der Zeit zeigt Abbildung 5.19.

In Tabelle 4.1 findet sich ein Vergleich mit anderen Algorithmen zur Geschwindigkeitsbestimmung.

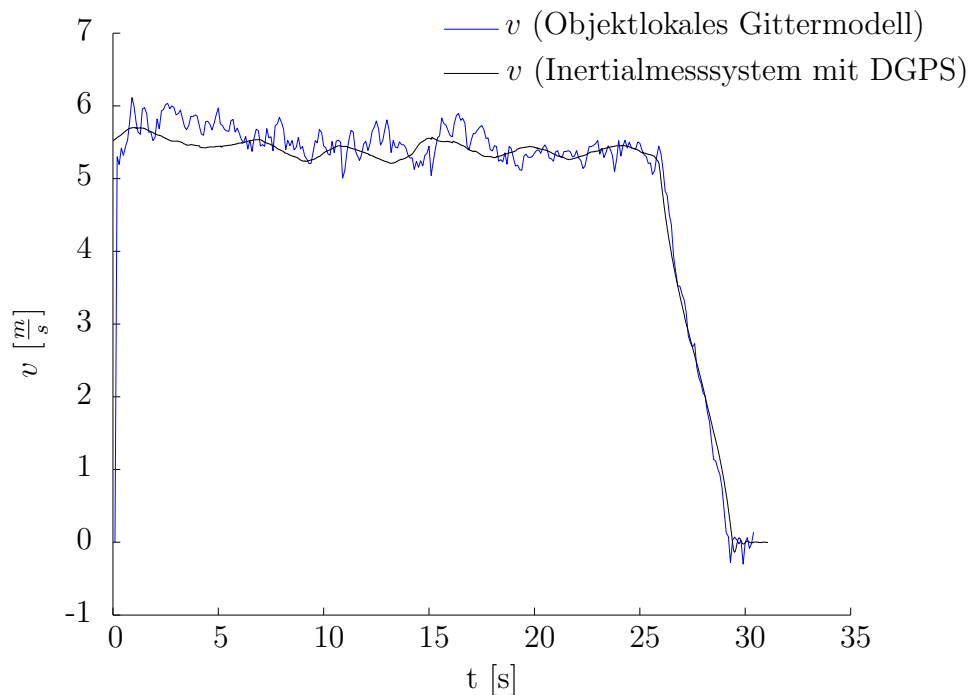


Abbildung 5.18: Vergleich der geschätzten Geschwindigkeit mit der Geschwindigkeit aus dem Inertialmesssystem mit DGPS

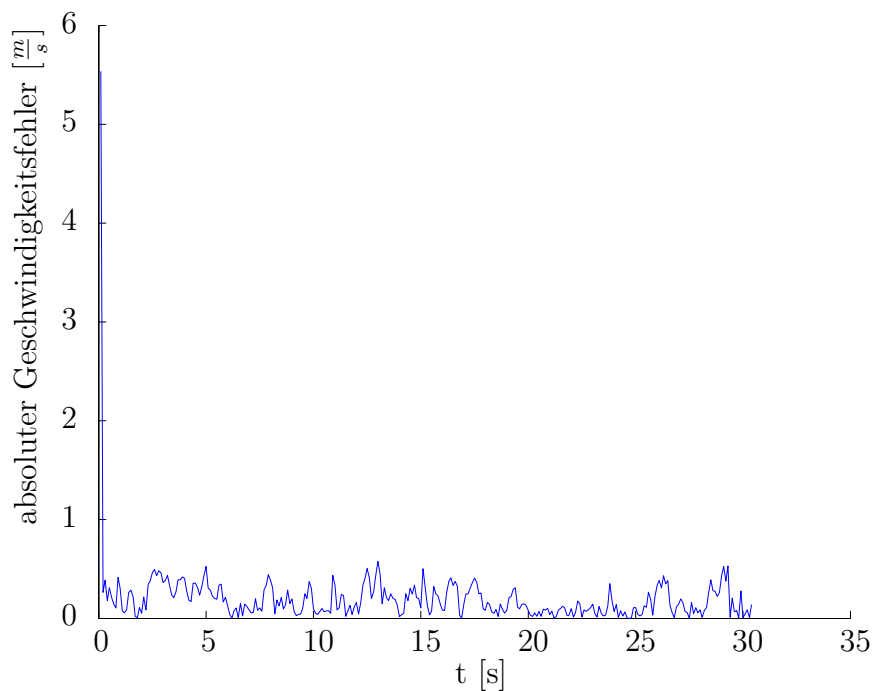


Abbildung 5.19: Absoluter Fehler der Geschwindigkeitsbestimmung

5.10 Anwendung von objekt-lokalen Gittermodellen

Die Anwendungsmöglichkeiten für die Objektverfolgung mit objekt-lokalen Gittermodellen liegt in einer unabhängigen Objektverfolgung für jegliche Arten von Fahrerassistenzsystemen. Einschränkungen entstehen jedoch durch die hohe Komplexität des Algorithmus, die zurzeit keine Echtzeitberechnung erlaubt. Untersuchungen den Algorithmus auf zweidimensionale Gittermodelle zu vereinfachen zeigen eine entsprechende Möglichkeit die Komplexität erheblich zu senken und erlauben unter Umständen eine echtzeitfähige Implementierung. Dabei ist es auch nicht mehr erforderlich Sensoren wie den Velodyne HDL-64E einzusetzen. Sensoren, die ein höheres Potential für den Serieneinsatz aufweisen, wie beispielsweise der Sick LD-MRS mit vier bis acht Laserebenen, können dann ebenso eingesetzt werden.

Mit der aktuellen Implementierung des Algorithmus zur Verarbeitung von dreidimensionalen Messdaten sind ausschließlich Berechnungen von Vergleichsmessungen möglich, die in einem nachgelagerten Schritt erfolgen und keine Echtzeitbearbeitung erfordern.

5.11 Zusammenfassung

In diesem Kapitel wurde ein neues Verfahren vorgestellt, das einerseits eine Objektbeschreibung auf Basis eines dreidimensionalen Gitters erlaubt und andererseits eine Objektverfolgung ermöglicht. Das Verfahren ergibt sich aus einer Kombination eines Partikel-Filters mit objekt-lokalen Gittermodellen. Es wurden verschiedene Kostenfunktionen vorgestellt, die die Bewegung der Partikel, die im vorliegenden Fall jeweils ein objekt-lokales Gitter repräsentieren, bewerten. Das verwendete Bewegungsmodell wurde vorgestellt und die Parameter dafür mit Hilfe einer Fahrdynamikanalyse bestimmt. Die Fahrdynamikanalyse basiert auf realen Messdaten eines Testfahrzeugs mit unterschiedlichen Fahrern und Straßentypen aus verschiedenen Ländern.

Eine Auswertung mit realen Messdaten zeigt das sich über die Zeit aufbauende dreidimensionale Abbild des gemessenen Fahrzeugs in den objekt-lokalen Gittern. Die Positions- und Geschwindigkeitsbestimmung erweist sich als sehr genau im Vergleich zu den Daten eines Referenzsystems, welches aus einer Inertialmessensorik mit DGPS besteht.

Das vorgestellte Verfahren lässt sich auf Grund des sehr hohen Speicherbedarfs und der notwendigen Rechenleistung zurzeit nicht echtzeitfähig implementieren. Untersuchungen mit zwei statt drei Dimensionen zeigen jedoch Möglichkeiten zur echtzeitfähigen Implementierung.

Das Kapitel endet mit einem kurzen Überblick über mögliche Anwendungen für den vorgestellten Ansatz.

Zusammenfassung und Ausblick

Zukünftig werden Assistenzfunktionen in Fahrzeugen den Fahrer bei seinen Fahrten immer mehr unterstützen. Zunächst werden sich weiterhin lediglich angepasste Funktionen für unterschiedliche Anwendungsfälle in den Fahrzeugen finden. So wird ein Schwerpunkt auf der Unterstützung des Fahrers in Gefahrensituationen liegen. Darunter fallen Funktionen, die Brems- oder Lenkeingriffe einleiten, um eine Gefahrensituation zu vermeiden, oder wenigstens die Folgen zu minimieren. Ein anderer Schwerpunkt, der sich in den letzten Jahren mehr und mehr ergeben hat, sind Komfortfunktionen, die den Fahrer entlasten sollen. Als Beispiel seien Funktionen wie der Totwinkelassistent, die Verkehrszeichenerkennung oder der Tempomat mit Abstandshaltung zum vorausfahrenden Fahrzeug genannt. Ein logischer Schritt und sich bereits abzeichnender Trend wird die Zusammenführung aller Einzelsysteme zu einem Gesamtsystem sein. Dieser Schritt wird das autonome Fahren ermöglichen.

Mit der Entwicklung in Richtung des autonomen Fahrens wird es notwendig sein das gesamte Umfeld des eigenen Fahrzeugs, in einer höheren Detailstufe als heute, zu erfassen und zu interpretieren. Hieraus ergibt sich die notwendige Weiter- und Neuentwicklung von Sensoren und Steuergeräten. Neue Sensoren zeichnen sich in aller Regel durch höhere Reichweiten des Erfassungsbereichs aber auch durch höhere Auflösungen der Messungen aus. Damit lässt sich die notwendige detailliertere Erfassung der Umgebung des Fahrzeugs erreichen. Mit der höheren Auflösung der Messungen ist es jedoch erforderlich die bisher genutzten einfachen Modellannahmen zu überarbeiten und anzupassen. Durch die sich ergebende Komplexität neuer Funktionen steigt auch der Aufwand die entwickelten Funktionen zu testen. Neben dem höheren Aufwand ergeben sich neue Herausforderungen, wie derartige Funktionen prinzipiell getestet werden können. Je nach Anwendung ist es erforderlich eine Unterscheidung zwischen den relevanten Objekten durchzuführen. Als grobe Gliederung kann zwischen statischen und dynamischen Objekten unterschieden werden. Diese lassen sich weiter in Unterkategorien aufteilen. Im Verlauf der Kapitel wurden speziell Fahrzeuge wie PKW und LKW aus der Kategorie der dynamischen Objekte behandelt.

Hauptteil der vorliegenden Arbeit war die Entwicklung von neuen Methoden zur Objektrepräsentation von Fahrzeugen mit Hilfe von 3D-Lasersensordaten. Dabei sollen die neuen Methoden als Grundlage für weitere Entwicklungen dienen. Für die Verarbeitung

der Daten des Lasersensors ergibt sich zunächst die Notwendigkeit einer Segmentierung. Diese wurde in Kapitel '3 Segmentierung' eingeführt. Dabei diente das Kapitel einerseits als Überblick und Evaluierung von bekannten Verfahren, zeigte jedoch auch neue Anpassungen im Bezug auf den eingesetzten Sensor. Zusätzlich wurde ein Verfahren welches auf einem dreidimensionalen Gitter Ansatz aufbaut speziell für den eingesetzten Sensor entwickelt und vorgestellt.

In Kapitel '4 Außenkonturbestimmung' wurde das erste von zwei neuen Verfahren zur Objektrepräsentation beschrieben. Das inzwischen patentierte Verfahren [98] verwendet eine Gitterfunktion als Grundmodell um die Außenkontur eines Fahrzeugs zu beschreiben. Das Gitter lässt sich über die Knotenpunkte, die die Eckpunkte der Zellen darstellen, an die segmentierten dreidimensionalen Daten des Lasersensors anpassen. Dabei wird der quadratische Abstand zwischen Knotenpunkten und Messungen minimiert. Die Neuerung stellt die Verwendung von zusätzlichen Randbedingungen dar. Diese beschreiben allgemein die geometrische Form eines Fahrzeugs. Sie sind jedoch frei formuliert, dass ohne Veränderung von Parametern die Außenkontur verschiedener PKW bis hin zu LKW beschrieben werden kann. Für die Bestimmung der Außenkontur ergibt sich ein Optimierungsproblem mit Randbedingungen. Zur Lösung wurde ein für diesen Anwendungsfall geeignetes Verfahren vorgestellt, welches zusätzlich, durch Ausnutzung spezieller Eigenschaften, so implementiert wurde, dass sich der gesamte Algorithmus in Echtzeit berechnen lässt. Anhand von Beispielen mit künstlich veränderten Messdaten wurde die Robustheit der Außenkonturbestimmung gezeigt. Mit Hilfe der dreidimensionalen Außenkontur lässt sich ebenfalls eine zweidimensionale Außenkontur bestimmen. Der Vorteil dadurch ist, falls eine zweidimensionale Außenkontur für eine Anwendung benötigt wird, dass auch in der zweidimensionalen Außenkontur indirekt Informationen aus allen dreidimensionalen Messungen enthalten sind. Ein Vergleich der realen mit der berechneten zweidimensionalen Außenkontur zeigt die zu erwartende Genauigkeit. Ein Abschnitt widmete sich der Übertragung des Verfahrens auf andere Sensoren. Dabei wurde gezeigt, dass das Verfahren ebenso ohne Anpassung von Parametern mit Messungen eines Lasersensors mit vier Ebenen und mit Messungen einer Stereokamera gleichermaßen eine Außenkontur bestimmen kann. Ein weiterer Abschnitt behandelte die möglichen Anwendungen der Außenkonturbestimmung.

Das zweite Verfahren zur Objektrepräsentation wurde in Kapitel '5 Objektverfolgung mit einem objekt-lokalen Gittermodell' beschrieben. Das Verfahren basiert auf einem dreidimensionalen objekt-lokalen Gittermodell, in dem sich Messungen eines Objekts über die Zeit akkumulieren. Es wird für jedes rigide Objekt der Umgebung ein entsprechendes Gittermodell erstellt. Durch die Kombination des objekt-lokalen Gittermodells mit einem Partikelfilter ist es möglich, gleichzeitig zur Objektrepräsentation, das Objekt über der Zeit zu verfolgen. Hierbei entspricht jedes Partikel einem objekt-lokalen Gittermodell. Das verwendete Bewegungsmodell, wie sich die Partikel, bzw. die objekt-lokalen Gittermodelle bewegen, wurde ebenso beschrieben, wie die Auswertung von realen Testfahrten zur Bestimmung der Parameter für das Bewegungsmodell. Zentrale Stelle des

Kapitels stellt der Abschnitt über die Kostenfunktionen dar. Diese werden für die Bewertung der Partikel benötigt. Dabei zeigte sich, dass sich die Kostenfunktionen Besetzungsgradverhältnis und Besetzungsverhältnis sowie das Integrationsverhältnis am besten für die vorliegende Anwendung eignen. Die Auswertung der Positionsbestimmung und der Geschwindigkeitsbestimmung zeigte in einem anspruchsvollen Szenario mit zum Teil wenigen Messdaten auf dem Objekt und vielen Aspektwinkeländerungen, sehr gute Ergebnisse. Die sich dabei ergebende dreidimensionale Beschreibung des Objekts in den objekt-lokalen Gittermodellen zeigt, obwohl das Gittermodell bewusst größer als das Objekt gewählt wurde, dass Messungen von der gleichen Stelle des Objekts auch in die gleiche Zelle des Gittermodells eingetragen werden. Die Aufbauten auf dem Dach des Objekts finden sich ebenso in der Objektbeschreibung über das Gittermodell, wie die Rundungen des Fahrzeugs an den Enden der Stoßfänger.

Das zweite Verfahren lies sich zum aktuellen Zeitpunkt aufgrund seines hohen Ressourcenbedarfs nicht wie das erste Verfahren in Echtzeit implementieren. Für den Ansatz mit objekt-lokalen Gittermodellen sind daher weitere Untersuchungen in Richtung der möglichen Implementierungen nötig. Aktuell ist der Algorithmus in Matlab implementiert, was einige Schwächen bezüglich der Laufzeit in Verbindung mit objektorientiertem Programmcode aufweist.

Die beiden vorgestellten Verfahren zur Objektrepräsentation, sowie die Verfahren zur Segmentierung stellen lediglich einen Teil eines Gesamtsystems dar. Für den Einsatz als Referenzsystem oder für die Realisierung einer Fahrerassistenzfunktion muss der hier nur kurz erwähnte Ansatz zur Klassifikation weiterentwickelt werden. Auch die Untersuchung des objekt-lokalen Gittermodells bei mehr als einem Fahrzeug steht noch aus.

Für eine möglichst automatisierte Auswertung, bei einem Referenzsystem, ist eine generische zeitliche Sensorsynchronisation empfehlenswert. Die Sensorsynchronisation kann sich jedoch beliebig komplex gestalten. Dies hängt von der Fähigkeit der Sensoren ab, ob sie eigenständig zu einer globalen Uhr einen Zeitstempel zur Messung vergeben können, oder ob eine Software auf einem Rechner dies übernimmt. Trifft der zweite Fall zu, gilt es, je nach Anforderung an die Genauigkeit, die Latenzen und den Jitter möglichst genau zu bestimmen. Muss dies für jede Sensorkombination von Hand neu erstellt werden, ist es ein sehr zeitintensiver Prozess.

Die Arbeit zeigt vielversprechende Ergebnisse für den Einsatz der vorgestellten Verfahren in der Entwicklung von Fahrerassistenzfunktionen oder bei der Implementierung von Referenzsystemen. Die steigende Rechenleistung zukünftiger Steuergeräte wird die Integration dieser Verfahren ermöglichen.

Literaturquellen

- [1] AHN, Sung J.; RAUH, Wolfgang; CHO, Hyung S. ; WARNECKE, Hans-Jürgen: Orthogonal Distance Fitting of Implicit Curves and Surfaces. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), S. 620–638
- [2] ALT, Walter: *Nichtlineare Optimierung*. Vieweg & Sohn Verlagsgesellschaft, Braunschweig/Wiesbaden, 2002
- [3] ARULAMPALAM, M. S.; MASKELL, Simon ; GORDON, Neil: A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. In: *IEEE Transactions on Signal Processing* 50 (2002), S. 174–188
- [4] BESL, Paul; MCKAY, Neil: A Method for Registration of 3-D Shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), S. 239–256
- [5] BLENDER: *Benutzerhandbuch*. <http://wiki.blender.org/index.php/Doc:2.6/Manual>
- [6] BUEHLER, Martin; IAGNEMMA, Karl ; SINGH, Sanjiv: *The 2005 DARPA Grand Challenge*. Springer Berlin Heidelberg New York, 2007
- [7] BUNDESBEHÖRDEN DER SCHWEIZERISCHEN EIDGENOSSENSCHAFT: *Übersetzung des Übereinkommens über den Strassenverkehr*. http://www.admin.ch/ch/d/sr/0_741_10/index.html. Version: 2007
- [8] BURGESS, Christopher J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. In: *Data Mining and Knowledge Discovery* 2 (1998), S. 121–167
- [9] CHEN, Zhe: Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond / McMaster University, Canada. 2007. – Forschungsbericht
- [10] COMANICIU, Dorin; MEER, Peter: Mean Shift: A Robust Approach Toward Feature Space Analysis. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), S. 603–619
- [11] CONTINENTAL AG: *Infrarot-Sensor (Pressemitteilung)*. http://www.conti-online.com/generator/www/com/de/continental/presseportal/themen/pressemitteilungen/3_automotive_group/chassis_safety/press_releases/volvo_xc60_31_03_08_de.html. Version: 2008
- [12] CONTINENTAL AG: *ARS30X Long Range Radar (Produktbeschreibung)*. http://www.conti-online.com/generator/www/de/de/continental/industrial_sensors/themes/download/ars_300_datenblatt_de.pdf. Version: 2009
- [13] COPPIN, Ben: *Artificial Intelligence Illuminated*. Jones and Bartlett Publishers, Inc., 2004

-
- [14] DAIMLER AG: *Neue S-Klasse mit Rundumblick (Pressemitteilung)*. <http://www.daimler.com/dccom/0-5-658451-49-1550372-1-0-0-0-0-12080-0-0-0-0-0-0-0.html>. Version: 2012
- [15] DAIMLER AG: *Die neue Mercedes-Benz E-Klasse: Mehr Effizienz, mehr Intelligenz, mehr Emotion (Pressemitteilung)*. <http://media.daimler.com/dcmedia/0-921-1553297-49-1570342-1-0-0-1570849-0-1-12759-614216-0-0-0-0-0-0-0.html?TS=1361280280635>. Version: 2013
- [16] DAIMLER AG: *S 500 INTELLIGENT DRIVE fährt autonom auf den Spuren von Bertha Benz (Pressemitteilung)*. <http://www.daimler.com/dccom/0-5-1625161-49-1625165-1-0-0-1625162-0-0-135-7163-0-0-0-0-0-0-0.html>. Version: 2013
- [17] DEMANT, Christian; STRICHER-ABEL, Bernd ; SPRINGHOFF, Axel: *Industrielle Bildverarbeitung: Wie optische Qualitätskontrolle wirklich funktioniert*. Springer-Verlag Berlin Heidelberg, 2011
- [18] D'ERRICO, John: Surface fitting using gridfit. In: *Matlab File Exchange*, 2005
- [19] DEUTSCHEN LANDESVERMESSUNG: *Satellitenpositionierungsdienst SAPOS*. <http://www.sapos.de/>. Version: 2015
- [20] DICKMANN, Ernst D.: *Dynamic Vision for Perception and Control of Motion*. Springer-Verlag London Limited, 2007
- [21] DODEL, Hans; HÄUPLER, Dieter: *Satellitennavigation*. Springer-Verlag Berlin Heidelberg, 2010
- [22] EFFERTZ, Jan: *Autonome Fahrzeugführung in urbaner Umgebung durch Kombination objekt- und kartenbasierter Umfeldmodelle*, Technischen Universität Carolus-Wilhelmina zu Braunschweig, Dissertation, 2009
- [23] EICHLER, Jürgen; EICHLER, Hans J.: *Laser. 7*. Springer Berlin Heidelberg, 2010
- [24] ELFES, Alberto: Using Occupancy Grids for Mobile Robot Perception and Navigation. In: *IEEE Computer* 22 (1989), S. 46–57
- [25] ELIAZAR, Austin: *DP-SLAM*, Duke University, Dissertation, 2005
- [26] ENZWEILER, Markus; GAVRILA, Dariu M.: Monocular Pedestrian Detection: Survey and Experiments. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2009), S. 2179–2195
- [27] ESTER, Martin; KRIEGEL, Hans-Peter; SANDER, Jörg ; XU, Xiaowei: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996, S. 226–231

- [28] FIJALKOWSKI, Bogdan T.: *Automotive Mechatronics: Operational and Practical Issues*. Springer, 2010 (Bd. 1)
- [29] FISCHLER, Martin A.; BOLLES, Robert C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24 (1981), S. 381–395
- [30] FORBES.COM (MOTHER NATURE NETWORK): *Self-Driving Cars Will Take Over By 2040*. <http://www.forbes.com/sites/economics/2012/09/25/self-driving-cars-will-take-over-by-2040/>. Version: 2012
- [31] FOX, Dieter: Adapting the Sample Size in Particle Filters Through KLD-Sampling. In: *International Journal of Robotics Research* 22 (2003)
- [32] FRANK SCHNEIDER: *ELROB*. <http://www.elrob.org/>
- [33] FREIE UNIVERSITÄT BERLIN: *AutoNOMOS Project*. <http://www.autonomos.inf.fu-berlin.de/>
- [34] FRIEDMAN, Jerome H.; BENTLEY, Jon L. ; FINKEL, Raphael A.: An Algorithm for Finding Best Matches in Logarithmic Expected Time. In: *ACM Transactions on Mathematics Software* 3 (1977), S. 209–226
- [35] GARATE, Virginia R.; BOURS, Roy ; KIETLINSKI, Kajetan: Numerical modeling of ADA system for Vulnerable Road Users protection based on radar and vision sensing. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2012, S. 1150 –1155
- [36] GENESYS ELEKTRONIK GMBH: *ADMA-G Produktbeschreibung*. <http://www.genesys-adma.de/file.php?id=965345642d24b3ab179b5dfc767b2b2c>. Version: Mai 2012
- [37] GESELLSCHAFT FÜR KONSUMFORSCHUNG: *Unser Auto von Morgen*. 2012
- [38] GLENNIE, Craig; LICHTI, Derek D.: Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning. In: *Remote Sensing* (2010)
- [39] GOLDFARB, Donald; IDNANI, Ashok: A NUMERICALLY STABLE DUAL METHOD FOR SOLVING STRICTLY CONVEX QUADRATIC PROGRAMS. In: *Mathematical Programming* Bd. 27, 1983
- [40] GSCHWANDTNER, Michael: *Support Framework for Obstacle Detection on Autonomous Trains*, Universität Salzburg, Österreich, Dissertation, 2013
- [41] HEERLEIN, Jörg: Die Abschaffung der Schrecksekunde - Lasersensoren im Automobil. In: *Laser+Photonik* (2009)
- [42] HEISSING, Bernd; ERSOY, Metin ; GIES, Stefan: *Fahrwerkhandbuch*. Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2011

-
- [43] HIMMELSBACH, Michael; LUETTEL, Thorsten ; WÜNSCHE, Hans-Joachim: Real-Time Object Classification in 3D Point Clouds Using Point Feature Histograms. In: *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, S. 994–1000
- [44] HIMMELSBACH, Michael; MÜLLER, André; LÜTTEL, Thorsten ; WÜNSCHE, Hans-Joachim: LIDAR-based 3D Object Perception. In: *Proceedings of 1st International Workshop on Cognition for Technical Systems*, 2008
- [45] HIMMELSBACH, Michael; VON HUNDELSHAUSEN, Felix; LÜTTEL, Thorsten; MANZ, Michael; MÜLLER, André; SCHNEIDER, Sebastian ; WÜNSCHE, Hans-Joachim: Team MuCAR-3 at C-ELROB 2009. In: *Proceedings of 1st Workshop on Field Robotics, Civilian European Land Robot Trial 2009*, 2009
- [46] HIMMELSBACH, Michael; VON HUNDELSHAUSEN, Felix ; WÜNSCHE, Hans-Joachim: Fast Segmentation of 3D Point Clouds for Ground Vehicles. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2010
- [47] HORNING, Armin; WURM, Kai M.; BENNEWITZ, Maren; STACHNISS, Cyrill ; BURGARD, Wolfram: OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. In: *Autonomous Robots* (2013)
- [48] IBEO AUTOMOTIVE SYSTEMS GMBH: *IBEO LUX (Produktbeschreibung)*. <http://www.ibeo-as.com/images/stories/pdf/neu/salesblatt%20lux.pdf>
- [49] IMAR NAVIGATION SOLUTIONS: *iTraceRT-F200 (Produktbeschreibung)*. http://www.imar-navigation.de/downloads/TraceRT-F200_de.pdf. Version: 2010
- [50] ISARD, Michael; BLAKE, Andrew: CONDENSATION - Conditional Density Propagation for Visual Tracking. In: *International Journal of Computer Vision* 29 (1998), S. 5–28
- [51] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 5., berarb. u. erw. Aufl. Springer, 2001
- [52] JOHNSTON, Cynthia; BARBARA, Philip: *Google gets first self-driven car license in Nevada*. <http://www.reuters.com/article/2012/05/08/uk-usa-nevada-google-idUSLNE84701320120508>. Version: 2012
- [53] KALMAN, Rudolph E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME—Journal of Basic Engineering* 82 (1960), Nr. Series D, S. 35–45
- [54] KALYAN, Bharath; LEE, K. W.; WIJESOMA, W. S.; MORATUWAGE, D. ; PATRIKALAKIS, Nicholas M.: A Random Finite Set based Detection and Tracking using 3D LIDAR in Dynamic Environments. In: *SMC, IEEE*, 2010, S. 2288–2292
- [55] KONRAD, Marcus; NUSS, Dominik ; DIETMAYER, Klaus: Localization in digital Maps for Road Course Estimation using Grid Maps. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2012, S. 87–92

- [56] LI, Tiancheng; SUN, Shudong; SATTAR, Tariq P. ; CORCHADO, Juan M.: Fighting against Sample Degeneracy and Impoverishment in Particle Filters: Particularly on Intelligent Choices. In: *CoRR* (2013)
- [57] LUCA DI GASPERO: *Quadprog++ 1.1*. <http://www.diegm.uniud.it/digaspero/index.php?page=software>
- [58] MACKAY, D.J.C.: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003
- [59] MÄHLISCH, Mirko: *Filtersynthese zur simultanen Minimierung von Existenz-, Assoziations- und Zustandsunsicherheiten in der Fahrzeugumfelderfassung mit heterogenen Sensordaten*, Universität Ulm, Dissertation, 2009
- [60] MAIMAN, Theodore H.: Stimulated Optical Radiation in Ruby. In: *Nature* 187 (1960), Nr. 4736, S. 493–494
- [61] MAK, Kevin: *Advanced Driver Assistance Systems Demand Forecast 2007 to 2018 DataFile, Strategy Analytics*. 2012
- [62] MANZ, Michael; LÜTTEL, Thorsten; VON HUNDELSHAUSEN, Felix ; WÜNSCHE, Hans-Joachim: Monocular Model-Based 3D Vehicle Tracking for Autonomous Vehicles in Unstructured Environment. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2011
- [63] MARKOFF, John: *Google Lobbies Nevada to Allow Self-Driving Cars*. http://www.nytimes.com/2011/05/11/science/11drive.html?_r=2&emc=eta1&. Version: 2011
- [64] MEISSNER, Daniel; DIETMAYER, Klaus: High-Precision 3D Traffic Recognition at Intersections. In: *Proceedings of the International Workshop on Intelligent Transportation*, 2011
- [65] MOOSMANN, Frank; FRAICHARD, Thierry: Motion Estimation from Range Images in Dynamic Outdoor Scenes. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010, 142–147
- [66] MOOSMANN, Frank; PINK, Oliver ; STILLER, Christoph: Segmentation of 3D Lidar Data in non-flat Urban Environments using a Local Convexity Criterion. In: *Proceedings of IEEE Intelligent Vehicles Symposium*, 2009, S. 215–220
- [67] NVIDIA CORPORATION: *CUDA*. <https://developer.nvidia.com/category/zone/cuda-zone>
- [68] OGAWA, Takashi; SAKAI, Hiroshi; SUZUKI, Yasuhiro; TAKAGI, Kiyokazu ; MORIKAWA, Katsuhiro: Pedestrian Detection and Tracking using in-vehicle Lidar for Automotive Application. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2011, S. 734–739

-
- [69] OSRAM OPTO SEMICONDUCTORS GMBH: *Datenblatt Impuls-Laserdiode SPL PL90*. <http://www.farnell.com/datasheets/316991.pdf>
- [70] PETROVSKAYA, Anna; THRUN, Sebastian: Model Based Vehicle Detection and Tracking for Autonomous Urban Driving. In: *Autonomous Robots* 26 (2009), S. 123–139
- [71] PIETZSCH, S.; AYCARD, O.; BURLET, J.; VU, Trung D.; HACKBARTH, T.; APPENRODT, N.; DICKMANN, J. ; RADIG, B.: Results of a Precrash Application Based on Laserscanner and Short Range Radars. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2008, S. 367 –372
- [72] PIETZSCH, Sylvia; APPENRODT, Nils ; RADIG, Bernd: Model-based Fusion of Laser Scanner and Radar Data for Target Tracking. In: *Proceedings of 8th International Workshop on Intelligent Transportation (WIT)*, 2011
- [73] REIF, Konrad: *Bosch Autoelektrik Und Autoelektronik: Bordnetze, Sensoren Und Elektronische Systeme*. Vieweg+Teubner, 2011 (Bosch Fachinformation Automobil)
- [74] ROBERT BOSCH GMBH: *Fernbereichsradar LRR3 (Produktbeschreibung)*. http://bosch-kraftfahrzeugtechnik.de/media/db_application/downloads/pdf/de_8/LRR3_Datenblatt_DE_2009.pdf. Version: 2009
- [75] RUSINKIEWICZ, Szymon; LEVOY, Marc: Efficient Variants of the ICP Algorithm. In: *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001
- [76] RUSSELL, Bryan C.; TORRALBA, Antonio; MURPHY, Kevin P. ; FREEMAN, William T.: LabelMe: A Database and Web-Based Tool for Image Annotation. In: *International Journal of Computer Vision* 77 (2008), S. 157–173
- [77] SAMPATH, Aparajithan; SHAN, Jie: Segmentation and Reconstruction of Polyhedral Building Roofs From Aerial Lidar Point Clouds. In: *IEEE Transactions on Geoscience and Remote Sensing* 48 (2010), S. 1554–1567
- [78] SARHOLZ, F.; MEHNERT, J.; KLAPPSTEIN, J.; DICKMANN, J. ; RADIG, B.: Evaluation of Different Approaches for Road Course Estimation using Imaging Radar. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, S. 4587 –4592
- [79] SATO, S; HASHIMOTO, M; TAKITA, M; TAKAGI, K ; OGAWA, T: Multilayer Lidar-Based Pedestrian Tracking in Urban Environments. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2010, S. 849–854
- [80] SCHMID, Matthias R.: *Umgebungserfassung für Fahrerassistenzsysteme mit hierarchischen Belegungskarten*. Neubiberg, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Dissertation, 2012

- [81] SCHMID, Matthias R.; MÄHLISCH, Mirko; DICKMANN, Jürgen ; WÜNSCHE, Hans-Joachim: Dynamic Level of Detail 3D Occupancy Grids for Automotive Use. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2010, S. 269–274
- [82] SCHNABEL, Ruwen; WAHL, Roland ; KLEIN, Reinhard: Efficient RANSAC for Point-Cloud Shape Detection. In: *Computer Graphics Forum* 26 (2007), S. 214–226
- [83] SHAPIRO, L.G.; STOCKMAN, G.C.: *Computer vision*. Prentice Hall, 2001
- [84] SICK AG: *Laserscanner LD-MRS (Produktbeschreibung)*. <https://mysick.com/saqqara/get.aspx?id=im0040378&title=Laserscanner+LD-MRS>. Version: 2009
- [85] STIMMING, Christian; AHLERS, Florian: Object Contour Tracking in Vehicle and Infrastructure Laserscanner Systems. In: *Proceedings on 7th International Workshop on Intelligent Transportation (WIT)*, 2010
- [86] TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG: *Stadtpilot*. <http://stadtpilot.tu-bs.de/>
- [87] THRUN, Sebastian: *What we're driving at*. <http://googleblog.blogspot.de/2010/10/what-were-driving-at.html>. Version: 2010
- [88] THRUN, Sebastian; BURGARD, Wolfram ; FOX, Dieter: *Probabilistic Robotics*. The MIT Press, 2005
- [89] VELODYNE LIDAR, INC.: *Velodyne HDL-64E (Produktbeschreibung)*. http://velodynelidar.com/lidar/products/brochure/HDL-64E%20S2%20datasheet_2010_lowres.pdf. Version: 2010
- [90] VIEWEG, Christof: *Zum Zeitung lesen auf die Überholspur*. <http://www.welt.de/motor/article12326230/Zum-Zeitung-lesen-auf-die-Ueberholspur.html>. Version: 2011
- [91] VIPUL, Jain; PAYAM, Heydari: *Automotive Radar Sensors in Silicon Technologies*. Springer Science+Business Media New York, 2013
- [92] WELCH, Greg; BISHOP, Gary: *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA : University of North Carolina at Chapel Hill, 1995. – Forschungsbericht
- [93] WINNER, Hermann; HAKULI, Stephan ; WOLF, Gabriele: *Handbuch Fahrerassistenzsysteme*. Vieweg+Teubner Verlag, 2012
- [94] WITTE, Carmen; ARMBRUSTER, Walter ; JÄGER, Klaus: Automatic generation of 3D models from real multisensor data. In: *11th International Conference on Information Fusion IEEE*, 2008
- [95] ZENG, Qihong; LAI, Jiazhen; LI, Xianhua; MAO, Jianhua ; LIU, Xuefeng: Simple Building Reconstruction from LIDAR Point Cloud. In: *International Conference on Audio, Language and Image Processing, IEEE*, 2008, S. 1040–1044

Eigene Veröffentlichungen

- [96] DICKMANN, Jürgen; FISCHER, Christoph; HACKBARTH, Thomas; HAHN, Markus; KLAPPSTEIN, Jens; MUNTZINGER, Marc; SAILER, Alfons ; STEINEMANN, Philipp: *Fahrzeug mit Umgebungserfassung und Interface zur Übernahme der Fahraufgabe durch den Beifahrer (Aktiver Copilot)*. Patent DE 10 2013 003 214.8, 2013
- [97] KLAPPSTEIN, Jens; STEINEMANN, Philipp: Laserscanner und deren Signalverarbeitung im automobilen Umfeld. In: *Terrestrisches Laserscanning* 106 (2011)
- [98] KLAPPSTEIN, Jens; STEINEMANN, Philipp: *Verfahren zur Außenkonturbestimmung eines Objektes, insbesondere eines Fahrzeuges*. Patent DE 10 2011 103 232, 2011
- [99] STEINEMANN, Philipp; KLAPPSTEIN, Jens; DICKMANN, Jürgen; VON HUNDELSHAUSEN, Felix ; WÜNSCHE, Hans-Joachim: Geometric-Model-Free Tracking of Extended Targets Using 3D-LIDAR-Measurements. (2012). – SPIE
- [100] STEINEMANN, Philipp; KLAPPSTEIN, Jens; DICKMANN, Jürgen; WÜNSCHE, Hans-Joachim ; VON HUNDELSHAUSEN, Felix: Determining the Outline Contour of Vehicles in 3D-LIDAR-Measurements. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2011, S. 479–484
- [101] STEINEMANN, Philipp; KLAPPSTEIN, Jens; DICKMANN, Jürgen; WÜNSCHE, Hans-Joachim ; VON HUNDELSHAUSEN, Felix: 3D Outline Contours of Vehicles in 3D-LIDAR-Measurements for Tracking Extended Targets. In: *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2012, S. 432–437

