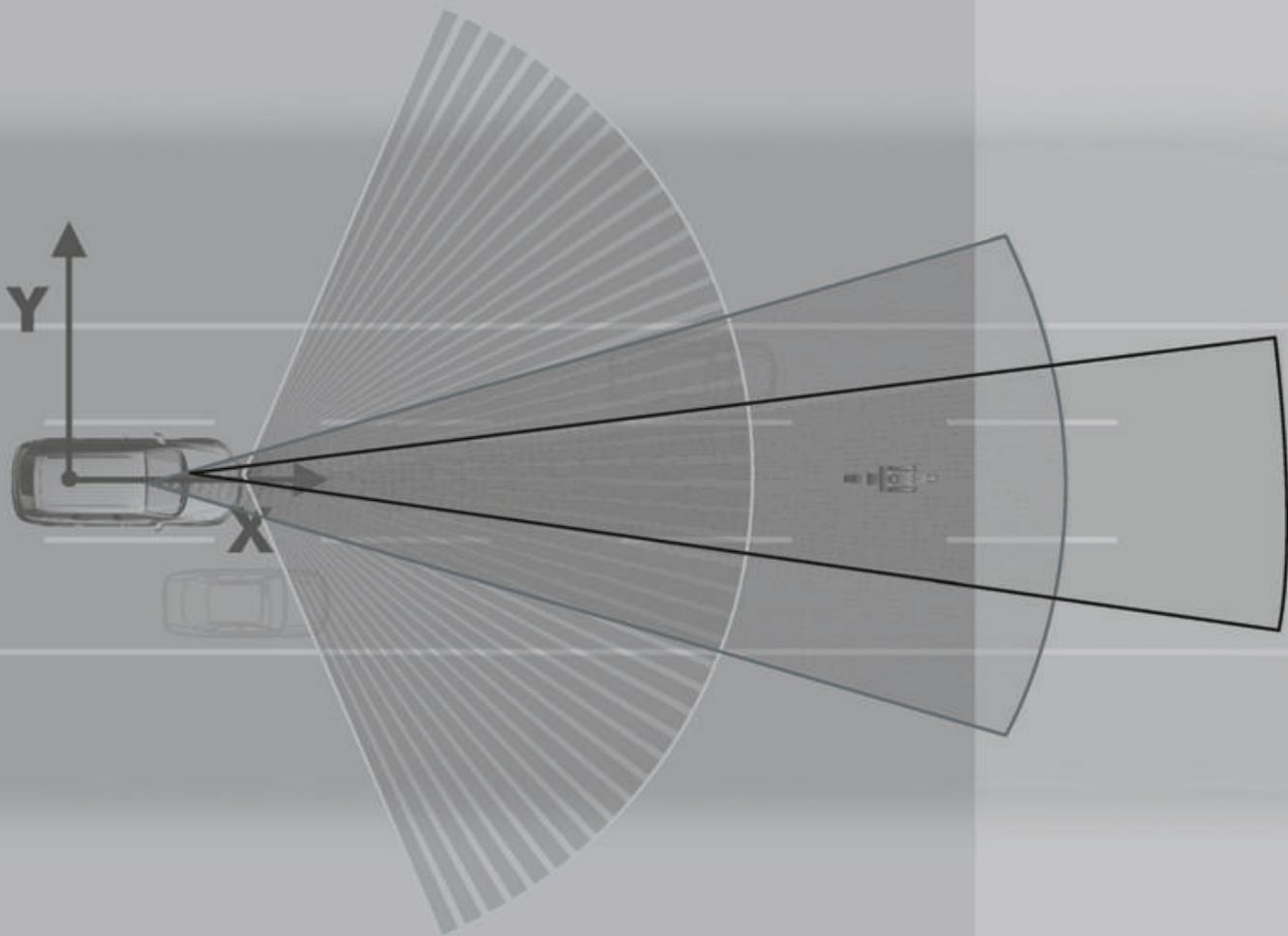




Videobasierte modell- gestützte Objekterkennung für Fahrerassistenzsysteme

Dr.-Ing. Johannes Speth



Videobasierte modellgestützte Objekterkennung für Fahrerassistenzsysteme

Dipl.-Ing. Univ. Johannes Speth

Lehrstuhl für Realzeit-Computersysteme

**Videobasierte modellgestützte Objekterkennung
für Fahrerassistenzsysteme**

Dipl.-Ing. Univ. Johannes Speth

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. techn. J. A. Nossek

Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. G. Färber, i. R.

2. Univ.-Prof. Dr.-Ing. K. Diepold.....

Die Dissertation wurde am 09.06.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 18.11.2009 angenommen.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

1. Aufl. - Göttingen: Cuvillier, 2010
Zugl.: (TU) München, Univ., Diss., 2009

978-3-86955-317-7

Audi Dissertationsreihe, Band 28

© CUVILLIER VERLAG, Göttingen 2010
Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage 2010
Gedruckt auf säurefreiem Papier.

978-3-86955-317-7

Danksagung

Diese Arbeit entstand im Rahmen meiner Forschungstätigkeit für die *Ingolstadt Institute der Technischen Universität München* (INI.TUM) als Doktorand am Lehrstuhl für Realzeit-Computersysteme. Mein ausgesprochen großer Dank gilt meinem Doktorvater Professor Georg Färber, der mir die Arbeit an diesem spannenden Thema erst ermöglicht hat und mir durch sein Vertrauen und seinen fundierten fachlichen Rat stets den nötigen Rückhalt gegeben hat. Danke auch an Prof. Diepold für die Übernahme der Aufgabe als Zweitprüfer und an Prof. Nosseck für die Übernahme des Vorsitzes der Prüfungskommission.

Vielen Dank für die gute fachliche Betreuung und den unerschöpflichen Ideenreichtum von Dr. Ulrich Hofmann. Ich werde die intensiven und bereichernden Diskussionen vermissen – sie waren der Schlüssel zum Erfolg dieser Arbeit. Bei Dr. Karl-Heinz Siedersberger bedanke ich mich ganz herzlich für die freundschaftliche und zielorientierte Führungsarbeit im Projekthaus und in unserem Team. Es war eine sehr bereichernde Erfahrung in einem so sympathischen und motivierten Team zu arbeiten. Dafür möchte ich mich auch bei meinen Kollegen Andreas Siegel, Michael Reichel, Dr. Rolf Dubitzky und Sayed Bouzouraa bedanken. Für ihre ausgezeichneten Leistungen und die angenehme Zusammenarbeit bei ihren studentischen Arbeiten danke ich auch Johannes Schels und Peter Bergmiller.

Im Rahmen parallel laufender INI.TUM-Projekte habe ich viele nette und inspirierende Kollegen kennengelernt. Es war immer sehr erfrischend und hat mir große Freude bereitet mit Rosa-Linde Fischer, Dr. Andreas Muigg, Dr. Michael Botsch, Peter Knauer, Dr. Thomas Bock und Tilo Koch zusammenzuarbeiten aber auch ausserfachliches zu besprechen und privat etwas zu unternehmen. Vielen Dank für die abwechslungsreiche Zeit.

Bei meinen Kollegen am Lehrstuhl möchte ich mich ganz herzlich bedanken für die Unterstützung, die ich als halb-externer in Uni-Angelegenheiten bekommen habe. Herausgreifen möchte ich dabei Dr. Sonja Karg, die mich bereits als Student betreute und bei Stephan Neumaier, der mir durch sein weitreichendes fachliches Wissen häufig weiterhelfen konnte.

Vielen Dank auch für die organisatorische Leitung der INI.TUM-Projekte an Dr. Uwe Koser, der sich immer für uns eingesetzt hat.

Mein herzlicher Dank gilt meinen guten Freunden Bastian Hartig und Sebastian Büschel, die immer für mich da sind, wenn ich sie brauche und es durch gemeinsame extreme Unternehmungen in vielerlei Hinsicht immer wieder schaffen mich aus dem Alltagstrott herauszuholen. Das ist eine sehr wichtige Energiequelle für mich und somit auch für diese Arbeit.

Bei Bettina Alvarez und Dr. Manuel Alvarez bedanke ich mich sehr herzlich für die intensive moralische Unterstützung und für das Korrekturlesen dieser Arbeit.

Dafür, dass ich in der Lage war diese Arbeit fertig zu stellen, habe ich einzig meiner lieben Frau Katrin zu danken. Sie sorgt dafür, dass ich noch ein Leben neben der Arbeit habe und die wichtigen Dinge nicht aus dem Blick verliere. Unserem Sohn Julian danke ich dafür, dass er für mich jeden Tag die Sonne scheinen lässt.

Der größte Dank auch über diese Arbeit hinaus gilt meinen Eltern dafür, dass sie meine Schulausbildung und mein Studium ermöglicht haben und mich bei jedem einzelnen Schritt meines Lebenswegs mit voller Kraft unterstützen. – Ohne Euch wäre ich nicht hier.

München, 10.01.2010

Inhaltsverzeichnis

Verzeichnis der verwendeten Symbole	ix
1 Einleitung	1
1.1 Motivation	1
1.2 Beiträge dieser Arbeit	4
1.3 Struktur dieser Arbeit	5
2 Stand der Technik	7
2.1 Fahrerassistenzsysteme mit maschineller Wahrnehmung	7
2.2 Forschungsprojekte zur vorausschauenden Fahrerassistenz	14
2.2.1 Historie	14
2.2.2 Aktuelle Forschungsprojekte	15
3 Beschreibung des Gesamtsystems	20
3.1 Aufbau des Versuchsfahrzeugs	20
3.2 Physikalische Beschreibung der Sensoren	21
3.2.1 Einordnung von Sensortypen	22
3.2.2 Radar	23
3.2.3 Laserscanner	27
3.2.4 Kamera	30
3.3 Verwendete Sensorkonfiguration	36
3.4 Koordinatensysteme und Transformationen	37
3.4.1 Beschreibung der verwendeten Koordinatensysteme	38
3.4.2 Homogene Koordinatentransformation	42
3.5 Softwareentwicklungsumgebung ADTF	44
4 Modellbasierte Objekterkennung mittels Bildverarbeitung	46
4.1 Grundlagen der Bildverarbeitung	47
4.1.1 Kantendetektionsoperatoren	48
4.1.2 Hauptachsentransformation	51
4.2 Merkmalspezifische Operatoren	55
4.2.1 Gradientenhäufungsoperator	55
4.2.2 Konturbewertungsoperator	56
4.2.3 Symmetriepoperatoren	57
4.3 Detektoren	60
4.3.1 Reifendetektor	60
4.3.2 Lichtdetektor	61

4.3.3	Bodenreflexionsdetektor	64
4.3.4	Dachkantendetektor für Fahrzeugheckansichten bei Nacht	66
4.4	Modellbasierte Objekterkennung	69
4.4.1	Modelle der Objektgestalt	70
4.4.2	Modelle der Objektdynamik	73
4.4.3	Rekursive Zustandsschätzung nach dem 4D-Ansatz	79
4.5	Zusammenfassung zur modellbasierten Objekterkennung	83
5	Klassifikation durch Maschinenlernverfahren	84
5.1	Random-Forest-Klassifikator	84
5.1.1	Entscheidungsbäume	85
5.1.2	Ensemble-Lernverfahren	88
5.2	Generalisierte Radiale-Basis-Funktionen-Netze	89
5.3	Merkmalsselektion	91
6	Fahrzeugdetektion bei Tag und Nacht	93
6.1	Philosophie	93
6.2	Herausforderungen	96
6.3	Verfahren zur Fahrzeugdetektion bei Tag	98
6.3.1	Trainingsdatenerzeugung	98
6.3.2	ROI-Generierung	100
6.3.3	Objekterkennung mittels Klassifikation	106
6.3.4	Bewertung des Verfahrens	110
6.4	Verfahren zur Fahrzeugdetektion bei Nacht	117
6.4.1	Trainingsdatenerzeugung	117
6.4.2	ROI-Generierung	118
6.4.3	Objektdetektion mittels Klassifikation	126
6.4.4	Bewertung des Verfahrens	131
6.5	Zusammenfassung zur Fahrzeugdetektion	133
7	Strategien zur robusteren Objektverfolgung	134
7.1	Fahrspurtracking	134
7.1.1	Angepasste Vorausschau	135
7.1.2	Gruppierung von Markierungsmerkmalen	137
7.1.3	Situationsabhängige Trackerdynamik	145
7.1.4	Gierratenoffsetschätzung	148
7.2	Multisensorielles Fahrzeugtracking	150
8	Zusammenfassung und Ausblick	153
A	Grundlagen zu Maschinenlernverfahren	156
A.1	Kompromiss zwischen Bias und Varianz	158
A.2	Bewertung und Validierung eines Klassifikators	159
B	Parameter für Fahrzeugerkennung bei Nacht	161

Inhaltsverzeichnis

C	Bewertungsmaße	162
C.1	Detektionsraten	162
C.2	Konfusionsmatrix	163
D	Algorithmus zur Verkettung von Spurmarkierungsmerkmalen	165
E	Grundlagen zur Kalman-Filterung	168
F	Diskretisierung von Differenzialgleichungssystemen	170
	Literaturverzeichnis	171

Verzeichnis der verwendeten Symbole

RCS	Lehrstuhl für Realzeit-Computersysteme
EGO	Versuchsfahrzeug
LRR	Long Range Radar
SRR	Short Range Radar
FAS	Fahrerassistenzsystem
ACC	Adaptive Cruise Control
LDW	Lane Departure Warning
PDC	Park Distance Control
ADTF	Automotive Data and Time Triggered Framework
KOS	Koordinatensystem
FMCW	Frequency Modulated Continuous Wave
GRBF	Generalisierte Radiale-Basis-Funktion
MLP	Multi Layer Perceptron
PCA	Principle Component Analysis
BV	Bildverarbeitung
KOS_{Konst}	Konstruktionskoordinatensystem
KOS_K	Kamerakoordinatensystem
KOS_R	Radarkoordinatensystem
KOS_L	Laserkoordinatensystem
KOS_A	Fahrzeugaufbaukoordinatensystem
KOS_U	Umfeldkoordinatensystem
KOS_F	Fahrzeugkoordinatensystem
KOS_S	Fahrspurkoordinatensystem
I	Einheitsmatrix
Θ_L	Öffnungswinkel des Lidarsensors
N_K	Anzahl der Kanäle des Lidarsensors
θ_K	Abtastwinkel des Lidarsensors
δ_{Hor}	Horizontale Defokussierung des Lidarsensors
δ_{Ver}	Vertikale Defokussierung des Lidarsensors
ϑ_{n_K}	Winkel des n_K -ten Lidarkanal
r_{n_K}	Gemessene Distanz im n_K -ten Lidarkanal
\underline{P}_{n_K}	Gemessener 3D-Punkt im n_K -ten Lidarkanal
$b_{n_K,Hor}$	Horizontale Defokussierung des Laserstrahls für Messpunkt \underline{P}_{n_K}

Verzeichnis der verwendeten Symbole

g	Gegenstandsweite
b	Bildweite
f_{LK}	Brennweite des Lochkameramodells
u_{HP}	Spaltenindex des Kamerahauptpunktes
v_{HP}	Zeilenindex des Kamerahauptpunktes
U	Anzahl der Spalten im Videobild
V	Anzahl der Zeilen im Videobild
s_x	Ausdehnung eines Pixel in Bildzeilenrichtung
s_y	Ausdehnung eines Pixel in Bildspaltenrichtung
\mathbf{R}	Rotationsmatrix
φ	Rotationswinkel um X -Achse
θ	Rotationswinkel um Y -Achse
ψ	Rotationswinkel um Z -Achse
\mathbf{B}	Matrix des Bildinhaltes
u	Spaltenindex eines Pixels
v	Zeilenindex eines Pixels
$i[u, v]$	Intensitätswert des Pixels an der Position (u, v)
\mathbf{F}	Faltungsmatrix / Faltungskern
C	Anzahl der Spalten eines Faltungskerns
R	Anzahl der Zeilen eines Faltungskerns
$f[c, r]$	Wert der Faltungsmatrix an der Bildposition (c, r)
$i_{\mathbf{F}}[u, v]$	Ergebniswert einer Faltung an der Bildposition (u, v)
G	Anzahl an Beispielen in einem Datensatz
N	Dimension eines Merkmalsvektors
\underline{x}	Merkmalsvektor
$\underline{\mu}$	Mittelwertvektor
$\underline{\mathbf{C}}$	Kovarianzmatrix
E	Erwartungsfunktion
\underline{u}_i	i -ter Eigenvektor
λ_i	i -ter Eigenwert
\mathbf{U}	Matrix der Eigenvektoren
$\underline{x}_{\text{PCA}}$	PCA-Transformierter Merkmalsvektor
U_{PCA}	Breite des interpolierten Bildausschnitts für PCA
V_{PCA}	Höhe des interpolierten Bildausschnitts für PCA
$x_{\text{PCA},k}$	k -ter Eintrag von $\underline{x}_{\text{PCA}}$
$\underline{x}_{\text{Bild}}$	Bildausschnitt in Spaltenvektorform
$\underline{x}_{\text{Bild}}^*$	Rücktransformiertes Eigenface von $\underline{x}_{\text{Bild}}$
HI_i	i -tes Helligkeitsintervall
\mathbf{F}_{HI}	Faltungsmaske für Helligkeitsintervallsuche
τ_{pos}	Positiver Schwellwert für Intervallbeginn
τ_{neg}	Negativer Schwellwert für Intervallende
u_i	Spaltenindex des i -ten Helligkeitsintervalls
$v_{a,i}$	Zeilenindex des n -ten Intervallanfangs
$v_{e,i}$	Zeilenindex des n -ten Intervallendes

h_i	Grauwertsumme des n -ten Intervalls
LB_l	l -ter Lichtblob
L	Anzahl detektierter Lichtblobs
$u_{a,l}$	minimaler Spaltenindex von LB_l
$u_{e,l}$	maximaler Spaltenindex von LB_l
$v_{a,l}$	minimaler Zeilenindex von LB_l
$v_{e,l}$	maximaler Zeilenindex von LB_l
h_l	Grauwertsumme von LB_l
f_l	Fläche von LB_l
P	Anzahl der Punkte eines Formmodells
K	Anzahl der Kanten eines Formmodells
F	Anzahl der Flächen eines Formmodells
$P^{(k)}$	Anzahl der Punkte auf der k -ten Kante
$\underline{P}_p^{(k)}$	p -ter Punkt der k -ten Kante
$K_k^{(f)}$	k -te Kante der f -ten Fläche
$\underline{n}^{(f)}$	Normalenvektor der f -ten Fläche
c_0	Klothoidenkrümmung im Ursprungspunkt
c_1	Koeffizient der Krümmungsänderung einer Klothoide
$\chi(l)$	Bahnazimut-Winkel des Fahrspurmodells
$\beta(t)$	Schwimmwinkel des Einspurmodells
$\Delta y(t)$	Ablage des EGO-Fahrzeugs zur Fahrspurmitte
$\psi_G(t)$	Gierwinkel des EGO-Fahrzeugs zur Fahrspurtangente
$c_{0hm}(t)$	gemittelte horizontale Krümmung des Spurmodells
$c_{1hm}(t)$	Koeffizient der gemittelten horizontalen Krümmungsänderung
$c_{1h}(t)$	Koeffizient der horizontalen Krümmungsänderung
$\dot{\psi}_{\text{off}}(t)$	Gierratenoffset
$\lambda(t)$	Lenkwinkel am Rad des EGO-Fahrzeugs
$v(t)$	Betrag des Geschwindigkeitsvektors
L_{\min}	minimale Vorausschauweite
$L_{\max}(t)$	maximale Vorausschauweite
k_r	Seitenkraftbeiwert des Reifens
m_{Fzg}	Fahrzeugmasse
a_{Achse}	Achsabstand
$\dot{\psi}(t)$	Gierrate des EGO-Fahrzeugs
$\theta(t)$	Nickwinkel des EGO-Fahrzeugs gegenüber Bodenebene
ω_0	Eigenfrequenz des Fahrzeugaufbaus
ϕ_Y	Trägheitsmoment des Fahrzeugaufbaus um Y -Achse
\underline{S}	Fahrzeugschwerpunkt
γ	Dämpfungskonstante für Nickwinkelschätzung
$B_0(t)$	Breite des Fahrspurmodells am Ursprung
$B_1(t)$	Koeffizient der Breitänderung des Fahrspurmodells
$\psi_F(t)$	relativer Gierwinkel eines Fremdfahrzeugs im Umfeldkoordinatensystem
$X_1(t)$	Längsbewegung eines Fremdfahrzeugs
$v_1(t)$	Längsgeschwindigkeit eines Fremdfahrzeugs

Verzeichnis der verwendeten Symbole

$a_l(t)$	Längsbeschleunigung eines Fremdfahrzeugs
$Y_q(t)$	Querbewegung eines Fremdfahrzeugs
$v_q(t)$	Quergeschwindigkeit eines Fremdfahrzeugs
$L_F(t)$	Länge eines Fremdfahrzeugs
$B_F(t)$	Breite eines Fremdfahrzeugs
$H_F(t)$	Höhe eines Fremdfahrzeugs
\underline{x}_k^*	prädizierter Zustandsvektor auf den diskreten Abtastzeitpunkt t_k
$\hat{\underline{x}}_k$	innovierter Zustandsvektor
$x'_{m k}$	variiertes m -ter Zustand zum Zeitpunkt t_k
δx_m	Wert der Zustandsvariation des m -ten Zustands
\underline{v}	Sensorrohdatenvektor
$\underline{z}_{\text{KF}}$	Merkmalsvektor bei Kalmanfilterung
$\underline{z}_{\text{KF}}^*$	Prädizierter Merkmalsvektor
$\underline{z}'_{\text{KF}}$	variiertes Merkmalsvektor
δz_n	Wert der Merkmalsvariation des n -ten Merkmals
r_n	Modelliertes Messrauschen der n -ten Messung
Δz_n	Residuum der n -ten Messung
Δx_m	Innovationsbeitrag für m -ten Zustand
\mathbf{P}_k^*	prädizierte Schätzfehlerkovarianzmatrix
$\hat{\mathbf{P}}_k$	innovierte Schätzfehlerkovarianzmatrix
\mathbf{V}_k	Innovationskovarianzmatrix
\mathbf{H}_k	linearisierte Jakobi'sche Messmatrix
\mathbf{Q}_k	Kovarianzmatrix der Systemunsicherheit
\mathbf{R}_k	Kovarianzmatrix der Messunsicherheit
$\hat{s}_{\text{opt}}(t)$	Optimaler Split am Knoten t eines Entscheidungsbaums
$i(t)$	Unreinheitsfunktion für Entscheidungsbaum
$\Delta i(\hat{s}, t)$	Abnahme der Unreinheit durch Split \hat{s} im Knoten t
c_k	k -te Klasse
$h(\underline{x})$	Transferfunktion eines künstlichen Neurons
$I(\underline{x})$	Eingangsfunktion eines künstlichen Neurons
$O(I(\underline{x}))$	Ausgangsfunktion eines künstlichen Neurons
$h_{\text{GRBF}}(\underline{x})$	Transferfunktion eines GRBF-Netzes
C	Anzahl an Klassen
M	Anzahl an Prototypen
$d_k(\underline{x})$	Zugehörigkeitsfunktion des Zufallsvektors \underline{x} zur Klasse c_k
$\alpha_{m,k}$	Gewichtungsfaktor des m -ten Prototypen für k -te Klasse
$f_{\text{GRBF}}(\underline{x})$	Klassifikationsfunktion eines GRBF-Netzes
\mathbf{C}_m	Kovarianzmatrix für m -ten Prototypen
$\underline{\mu}_m$	Mittelwertvektor (Zentrum) des m -ten Prototypen
σ	Standardabweichung
FI	Feature Importance
B	Anzahl der Bäume des Random-Forests
H_{min}	minimale Fahrzeughöhe
H_{max}	maximale Fahrzeughöhe

$Z_{L,\min}$	minimale Einbauhöhe der Rückleuchten
$Z_{L,\max}$	maximale Einbauhöhe der Rückleuchten
B_{\min}	minimaler Abstand der Rückleuchten
B_{\max}	maximaler Abstand der Rückleuchten
$X_{\min}^{(SB)}$	minimaler Distanz des 3D-Suchbereichs
$X_{\max}^{(SB)}$	maximaler Distanz des 3D-Suchbereichs
$Z_{\min}^{(SB)}$	minimale Höhe über Boden des 3D-Suchbereichs
$Z_{\max}^{(SB)}$	maximale Höhe über Boden des 3D-Suchbereichs
$B^{(SB)}$	halbe Breite des 3D-Suchbereichs
ε	Toleranzwinkel zur Bildung eines Lichterpaars
Δy_{\max}	maximaler vertikaler Versatz eines Lichterpaars in Pixel
N_{LB}	Anzahl an Lichtblobs
N_{LP}	Anzahl an Lichterpaaren
$\beta_{i,j}^{(h)}$	Helligkeit-Ähnlichkeitsmaß für Lichterpaar
$\beta_{i,j}^{(f)}$	Flächen-Ähnlichkeitsmaß für Lichterpaar
$\Delta u_{i,j}$	Abstand der Lichter des Lichterpaars aus LB_i und LB_j im Bild
$v_{i,j,\min}^{(UK)}$	minimaler Zeilenindex für Unterkantensuchbereich
$v_{i,j,\max}^{(UK)}$	maximaler Zeilenindex für Unterkantensuchbereich
$v_{i,j}^{(UK)}$	Zeilenindex der gefundenen Unterkante
$u_{i,j,\min}^{(OK)}$	minimaler Spaltenindex für Oberkantensuchbereich
$u_{i,j,\max}^{(OK)}$	maximaler Spaltenindex für Oberkantensuchbereich
$v_{i,j,\min}^{(OK)}$	minimaler Zeilenindex für Oberkantensuchbereich
$v_{i,j,\max}^{(OK)}$	maximaler Zeilenindex für Oberkantensuchbereich
$v_{i,j}^{(OK)}$	Zeilenindex der gefundenen Oberkante
x_{Nacht}	Merkmalsvektor für Klassifikator bei Nacht
x'_{Nacht}	Merkmalsvektor der ausgewählten Merkmale
$\mu_{i,j}$	Winkel der Verbindungslinie eines Lichtpaars mit Bildzeile
l_{\min}	minimale Vorausschau der Spurerkennung
l_{\max}	maximale Vorausschau der Spurerkennung

Zusammenfassung

Um den steigenden Anforderungen zukünftiger Fahrerassistenzsysteme an die Qualität der Informationen über das lokale Umfeld des eigenen Fahrzeugs zu entsprechen, werden in dieser Arbeit Algorithmen zur videobasierten Fahrzeug- und Fahrspurerkennung entwickelt. Die Verfahren zur Fahrzeugdetektion kombinieren Expertenwissen mit modernen Maschinenlernverfahren in einem Gesamtsystem, das an verschiedene Problemstellungen zur videobasierten Objekterkennung angepasst werden kann. Durch den Einsatz spezieller Sensormodelle wird das anschließende multisensorielle Verfolgen von Fahrzeugen in einem Kalman-Filter basierten Objekttracker ermöglicht.

Ein Algorithmus zur videobasierten Fahrspurverfolgung wird durch robustheitssteigernde Maßnahmen erweitert, so dass eine automatische Spurführung des Versuchsfahrzeugs auch bei komplexen Fahrszenarien ermöglicht wird.

In order to meet the increasing requirements of future driver assistance systems in respect to the quality of information about the local environment of one's own vehicle, this thesis develops algorithms for the video-based perception of vehicles and traffic lanes. The methods for vehicle detection combine expert knowledge with state-of-the-art machine learning algorithms to form an overall system, which can be adapted to different video-based object recognition problems. Specialized sensor models subsequently facilitate the Kalman-Filter based tracking of vehicles using a multi-sensor arrangement.

An algorithm for the video-based tracking of traffic lanes is extended by measures to increase its robustness, allowing for automated lane-keeping control of the experimental vehicle even in complex driving scenarios.

1 Einleitung

Wäre es nicht schön, bei der täglichen Fahrt zur Arbeit die Zeitung lesen zu können? — Wäre es nicht schön, den Familienurlaub schon während der Fahrt zum Reiseziel beginnen zu lassen? — Wäre es im Autobahnstau nicht schön, sich einfach auszuruhen, etwas zu lesen oder einen Film anzuschauen?

Für die meisten Autofahrer hören sich diese Wünsche nach Szenen aus einem Science-Fiction-Film an. Viele Leute können heute aber auch noch nichts mit den Begriffen „Adaptive Cruise Control“, „Heading Control“, „Side Assist“ oder „Lane Departure Warning“ anfangen. Diese Begriffe sind lediglich ein Auszug aus dem breiten Spektrum an Fahrerassistenzsystemen (FAS), die bereits in Oberklasse-Fahrzeugen verfügbar sind. Die neuen Entwicklungen im Bereich der Fahrerassistenz werden unter dem Begriff Advanced Driver Assistance Systems (ADAS) zusammengefasst.

Eher bekannt sind die konventionellen FAS, wie z. B. das ABS und das ESP sowie die informierenden FAS, zu denen z. B. Radiostaumeldungen (RDS), Warnanzeigen in den Armaturen und Navigationssysteme gezählt werden. Zu den ersten Vertretern von FAS zählen bereits die Servolenkung, der Bremskraftverstärker und selbst die Geschwindigkeitsanzeige im Fahrzeug [64]. Die Systeme der ADAS zeichnen sich im Gegensatz zu den konventionellen FAS dadurch aus, dass sie durch den Einsatz umfelderfassender Sensorik neue Möglichkeiten im Automobil schaffen. Diese Arbeit beleuchtet einen Teil dieser Möglichkeiten und zeigt Entwicklungen zur maschinellen Wahrnehmung der Fahrzeugumgebung auf.

Im Folgenden wird in Unterkapitel 1.1 die Entwicklung neuer Fahrerassistenzsysteme motiviert, in 1.2 die Beiträge dieser Arbeit dargestellt, und abschließend wird in Kapitel 1.3 der Aufbau dieser Arbeit erläutert.

1.1 Motivation

Fahrerassistenzsysteme sollen den Fahrer bei der Führung seines Fahrzeugs entlasten und dafür sorgen, dass er entspannter am Ziel seiner Fahrt ankommt und auf dem Weg dorthin weniger Unfälle verursacht. Den Sinn der Entwicklung dieser Systeme zeigt die Statistik der Unfalltoten der letzten Jahrzehnte. Zusammen mit den Verbesserungen der Infrastruktur im Straßenverkehr haben die Weiterentwicklungen der Sicherheitssysteme im Fahrzeug dazu beigetragen, dass seit 1970 die Anzahl der getöteten Personen im Straßenverkehr stetig sinkt. Und das obwohl der Verkehr auf Deutschlands Straßen dichter wird

1 Einleitung

und die Fahrleistung zunimmt, was sich in einer immer höheren Anzahl an Unfällen pro Jahr widerspiegelt [59].

Die Systeme der ADAS setzen verstärkt Sensoren ein, die Informationen über das nahe Umfeld des Fahrzeugs liefern, so dass Aktionen vorausschauend geplant werden können. Dazu gehören auch die oben genannten ADAS aus dem Bereich der Komfort- bzw. warnenden Systeme sowie die neuesten Entwicklungen im Bereich der aktiven Sicherheitssysteme (z. B. Bremsassistent). Aktuell kooperieren und konkurrieren viele Forschungsinstitute und Automobilhersteller weltweit intensiv miteinander bei der weiteren Entwicklung moderner FAS, um das Automobil seiner ursprünglichen Bestimmung näher zu bringen – der autonomen Fortbewegung[44][34].

Vor dem Hintergrund der aktuellen Diskussionen um CO₂-Ausstoß und dessen Folgen für den Klimawandel und der allgemeinen Rohstoffknappheit, müssen sich auch die Entwickler von Fahrerassistenzsystemen die Frage stellen, wie sie konstruktiv auf die ökonomischen und ökologischen Probleme reagieren können. Im Bereich der Fahrerassistenz gibt es dazu viele Möglichkeiten. Beispielsweise kann durch eine optimierte Routenführung die Fahrzeit reduziert werden oder das Längsführungssystem ACC so ausgelegt werden, dass es unter Nutzung von digitalen Kartendaten vorausschauender und energiesparender fährt, als es der Mensch je könnte[5][75].

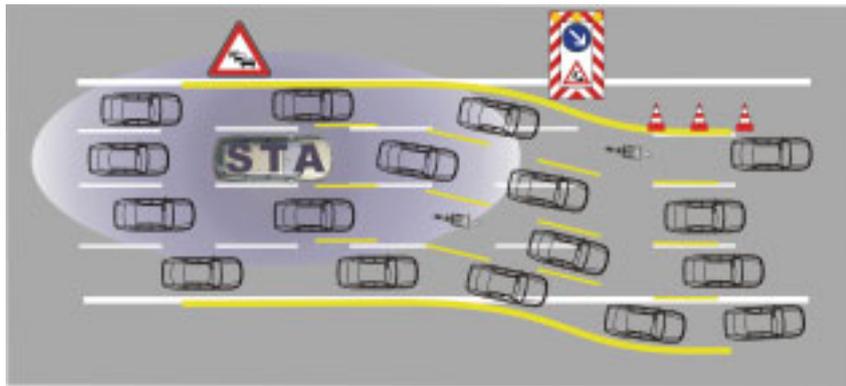


Bild 1.1: Stauassistenten

Eine weitere Möglichkeit wäre ein System, das das Fahrzeug automatisch durch einen Stau steuern kann – der so genannte Stauassistent (STA). Laut unabhängiger Studien kosten „Staus [...] täglich [...] 33 Millionen Liter Kraftstoff, 13 Millionen Stunden verlorene Zeit – und [...] 250 Millionen Euro [...]“ [25]. Abbildung 1.2 zeigt die Zeitanteile, die ein Fahrer statistisch in verschiedenen Geschwindigkeitsbereichen verbringt.

Gerade die verlorene Zeit und die daraus resultierenden wirtschaftlichen Kosten könnten dadurch gemindert werden, dass sich der Fahrer nicht vollständig auf die monotone Fahrzeugführung konzentrieren muss, sondern bereits in seinem mobilen Büro im Fahrzeug, mit der Unterstützung drahtloser Kommunikationswege, seiner Arbeit nachgehen kann. Auch im Hinblick auf eine älter werdende Gesellschaft kann das Automatisieren

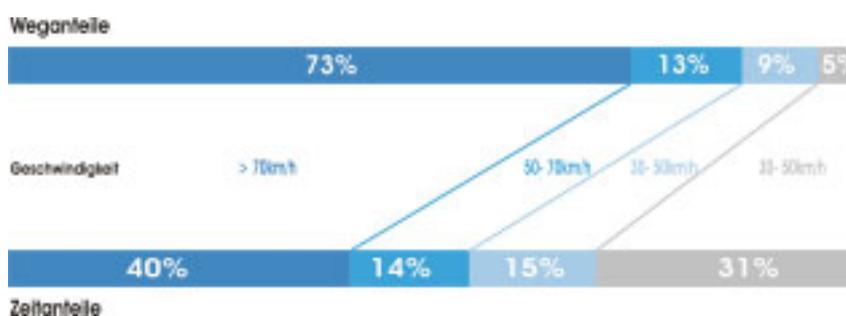


Bild 1.2: Zeitanteile in verschiedenen Geschwindigkeitsbereichen (nach [25])

der Fahrzeugführung zu einer verbesserten Mobilität im Alter und zu einer Erhöhung der Sicherheit auf den Straßen beitragen.

Wieso gibt es noch keine autonomen Fahrzeuge? Die heute verfügbaren Assistenzsysteme lassen die langfristige Vision erkennen, dass sich Fahrzeuge in den kommenden Jahrzehnten zu autonomen Systemen weiterentwickeln. Diese werden sich von den heute in der Entwicklung befindlichen automatischen Systemen dadurch unterscheiden, dass der Fahrer nicht einmal mehr die Tätigkeit als Überwacher des Systems übernehmen muss. Autonome Systeme, wie auch automatische Systeme, bergen das nicht zu vernachlässigende Risiko eines Fehlverhaltens durch fehlerhafte Sensorinformationen oder -interpretation. Dies könnte z. B. das unerwünschte Verlassen der Fahrspur oder den Zusammenstoß mit einem anderen Fahrzeugs nach sich ziehen. Wegen der noch ungeklärten Frage nach der Haftbarkeit für solche Fehlerfälle müssen alle heute verfügbaren vorausschauenden Assistenzsysteme noch vom Fahrer überwacht werden und übersteuerbar sein. Das bedeutet, dass der Fahrer letztendlich noch die volle Verantwortung trägt. Der große Nutzen autonomer Systeme ist also zugleich die große Herausforderung. Zusätzlich werden, im Unterschied zu Fehlern herkömmlicher Systeme, die Fehler (teil-)autonomer Fahrerassistenzsysteme vom Menschen erst viel später oder auch gar nicht erkannt, da dieser aus der Fahraufgabe herausgenommen ist. Eine höhere Fremdbeschäftigung bewirkt unmittelbar längere Reaktionszeiten des Fahrers auf unvorhergesehene Ereignisse. Dies ist schon heute beim Führen wichtiger Telefonate während des Fahrens feststellbar.

Die Grundlage automatisch agierender Systeme ist somit eine fehlerlose Situationsanalyse mit anschließender Aktionsplanung und -durchführung, sowie eine ständige Selbstüberwachung, die es ermöglicht, den Fahrer frühzeitig zu warnen.

Abbildung 1.3 zeigt den prinzipiellen Aufbau eines automatischen Fahrerassistenzsystems mit maschineller Wahrnehmung. Um die automatische Aktion durchführen zu können, muss zunächst die Szene richtig interpretiert und daraus ein adäquates Verhalten abgeleitet sein. Die Grundlage dafür bietet ein konsistentes Umfeldmodell, das auf der Merkmalsextraktion aus den Sensordaten mit anschließender Modellbildung basiert.

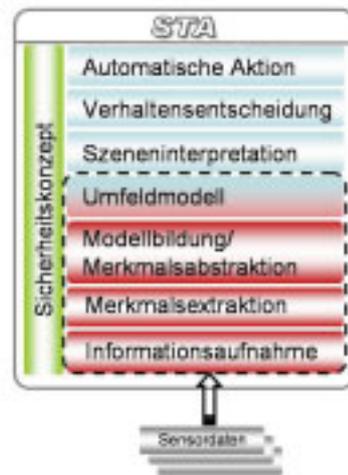


Bild 1.3: Systemaufbau der Stauassistentenfunktion

Somit ist die langfristige Entwicklung autonomer Assistenzsysteme von der Entwicklung robuster Wahrnehmungsalgorithmen, die speziell an die Anforderungen der autonomen Fahrzeugführung angepasst sind, abhängig.

1.2 Beiträge dieser Arbeit

Die Ziele dieser Arbeit wurden in dem sehr dynamischen Umfeld, bei der Entwicklung des Stauassistenten, an die sich ergebenden Problemstellungen ausgerichtet. Als grundsätzliches Ziel steht die Entwicklung von videobasierten Algorithmen im Vordergrund, welche der Erfassung des Umfeldes eines mit Sensorik ausgestatteten Fahrzeugs dienen. Die entwickelten Algorithmen müssen in C++-Modulen für die Entwicklungsumgebung ADTF¹⁾ gekapselt werden, welche dann den nachgeschalteten Teilsystemen zur Situationsanalyse, Verhaltensentscheidung und Regelung die nötige Information über Fahrzeuge und Fahrspuren im Fahrzeugumfeld liefert. Im Speziellen haben sich drei Entwicklungsschwerpunkte für diese Arbeit herauskristallisiert:

1. Erkennen und Verfolgen von anderen Fahrzeugen
2. Erkennen und Verfolgen von Fahrspurverläufen
3. Bewerten der entstandenen Verfahren

Aus den früheren Forschungsarbeiten von V. von Holt [49] und U. Hofmann [48] ist im Rahmen des Kooperationsprojekt FORBIAS [7] ein Kalman-Filter basierter Zustandsschätzer (im Folgenden auch „Tracker“ genannt) entstanden, welcher Fahrspuren und Fahrzeuge mittels Bildverarbeitung verfolgen kann. Die verwendete Kalmanfilter-

¹⁾ Automotive Data and Time Triggered Framework (siehe Kapitel 3.5)

Implementierung basiert auf der sequentiellen Methode nach [11]. Dieses System wird durch diese Arbeit um folgende Eigenschaften erweitert:

- Detektion neuer Fahrzeuge bei Tag und Nacht zum Initialisieren und Verifizieren des Fahrzeugtrackers
- Robusteres Verfolgen von Fahrspuren im Bereich komplexer Verkehrssituationen (z. B. Stau, Baustellenbereich)
- Stützung des videobasierten Fahrzeugtrackers durch andere Sensoren
- Detektion von Fahrspuren zum Initialisieren des Fahrspurtrackers

Um die Möglichkeiten der videobasierten Objekterkennung zu erweitern, werden spezielle Bildverarbeitungsoperatoren entwickelt. Diese werden dazu genutzt, um spezifische Objektmerkmale für die Erkennung von Fahrzeugen bei Tag- und Nachtszenarien zu extrahieren. Die Eignung dieser spezifischen Merkmale zur Fahrzeugdetektion wird statistisch untersucht und die wichtigsten Merkmale zu einem echtzeitfähigen und robusten Detektor kombiniert. Des Weiteren wird für die multisensorielle Fahrzeugverfolgung ein spezielles Sensormodell zur Verwendung des Laserscanners implementiert.

Zur letztendlichen Bewertung der Qualität des Zustandsschätzers bzw. des Detektors, sind verschiedene Verfahren umgesetzt:

- Softwaremodul zum Erzeugen von Referenzdaten im Videobild
- Vergleich der Schätzerangaben mit Ground-Truth-Daten aus Referenzsensorik

1.3 Struktur dieser Arbeit

Nach der Hinführung zum Thema der vorausschauenden Fahrerassistenzsysteme in diesem Kapitel, gibt die vorliegende Arbeit in Kapitel 2 einen Überblick über aktuelle Forschungs- und Entwicklungstätigkeiten auf diesem Gebiet. Anschließend wird in Kapitel 3 der Aufbau des Versuchsfahrzeugs mit den verwendeten Sensoren beschrieben. Im gleichen Kapitel wird auch auf die Koordinatensysteme und die Softwareentwicklungsumgebung eingegangen.

Kapitel 4 führt den Leser zunächst in die theoretischen Grundlagen zum Thema der Bildverarbeitung ein und erklärt die Funktionsweise der verwendeten Operatoren und Detektoren. Der zweite Teil des Kapitels behandelt die Objektmodelle zur modellbasierten Objektverfolgung. In Kapitel 5 werden die zur Klassifikation verwendeten Maschinenlernverfahren beschrieben.

Die folgenden Teile der Arbeit beschreiben die entwickelten praktischen Anwendungen auf Basis der zuvor beschriebenen Verfahren. Dabei werden in den Kapiteln 6.3 und 6.4 die entstandenen Verfahren zur Fahrzeugdetektion für Tag- und Nachtsequenzen behandelt. Kapitel 7 stellt die Erweiterungen an ein System zur Verfolgung der Fahrspur und anderer Fahrzeuge dar.

1 Einleitung

Abschließend werden in Kapitel 8 die Schwerpunkte dieser Arbeit zusammengefasst, einige Problempunkte angesprochen und das Weiterentwicklungspotenzial der Algorithmen kurz diskutiert.

2 Stand der Technik

Insgesamt ist die sensorgestützte Fahrzeugführung eine sehr interdisziplinäre Anwendung und integriert mehrere unterschiedliche Forschungsgebiete: Optik, Hochfrequenztechnik, Signalverarbeitung (Bildverarbeitung, Klassifikation, Filterung, etc.), Informatik, Regelung und Steuerung, Fahrzeugtechnik, Ergonomie, etc. Die aktuellen Entwicklungen aller Teilgebiete können nicht im Detail dargestellt werden. Im Fokus dieses Überblicks liegt die sensorgestützte Wahrnehmung des Fahrzeugumfeldes. Da diese keinen Selbstzweck erfüllt, sondern stets als Informationslieferant für eine nachgeschaltete Funktion dient, wird zunächst im Abschnitt 2.1 ein Überblick über die aktuell verfügbaren Fahrerassistenzsysteme geschaffen und dabei die zugrunde liegenden Sensorkonzepte erläutert. Anschließend werden in 2.2 die derzeit wichtigsten Forschungsprojekte im Bereich der autonomen Fahrzeugführung mit deren Konzepten zur maschinellen Wahrnehmung vorgestellt.

2.1 Fahrerassistenzsysteme mit maschineller Wahrnehmung

Die Grafik 2.1 zeigt die Entwicklungsschritte, die Fahrerassistenzsysteme bisher durchschritten haben und gibt einen Ausblick auf die Visionen in der Zukunft.

Nach [83] lassen sich Fahrerassistenzsysteme funktionsgetrieben in die Gruppen *Komfortsysteme*, *Fahrerinformationssysteme*, *Autonome Systeme* und *Effizienzsteigernde Systeme* einteilen. In dieser Arbeit wird eine Einteilung hinsichtlich der Auswirkung auf die Fahrzeugführung unternommen. Aus Fachbeiträgen und Fachbüchern[23][38][65] oder aktuellen Dissertationen aus dem Bereich der vorausschauenden FAS[15][20][49][48][59][86] kann ein Überblick über derzeitig verfügbare und in der Entwicklung befindliche Fahrerassistenzsysteme erlangt werden. Tabelle 2.1 fasst diese zusammen. Die Buchstaben A, B und C repräsentieren unterschiedliche mögliche Sensorvarianten eines Assistenzsystems. Buchstaben in Klammern bedeuten, dass diese Sensorkombinationen zwar noch nicht erhältlich sind, jedoch kurz vor der Serieneinführung stehen.

Im Folgenden werden die Systeme kurz vorgestellt. Um auf dem aktuellen Stand von Neuentwicklungen im Bereich der sensorgestützten Fahrerassistenz zu bleiben, sei auf die Internetseiten der zur Zeit führenden Automobilhersteller im Bereich Fahrerassistenz (z. B. Audi AG [2], Volkswagen AG[91], Daimler AG[30], BMW AG[14] oder Toyota [89]) sowie ihrer Systemlieferanten (z. B. Bosch AG[18], Hella KGaA Hueck&Co.[47] oder Continental[27]) verwiesen.

2 Stand der Technik

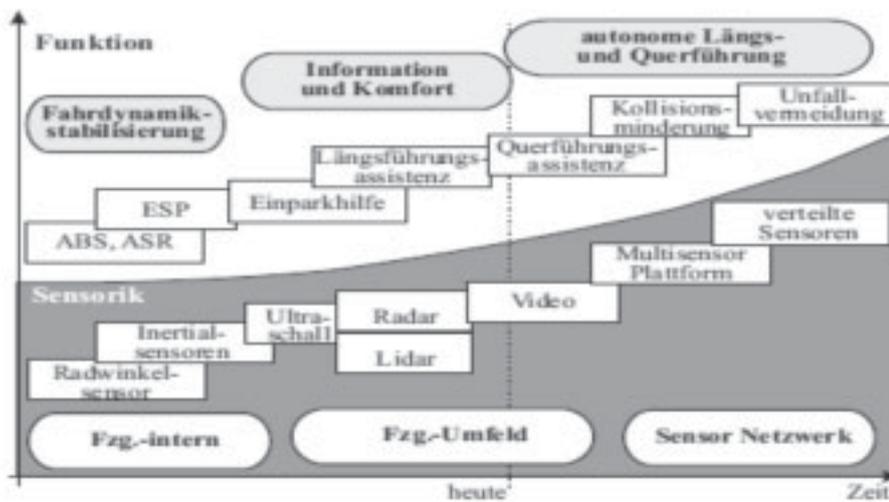


Bild 2.1: Entwicklung von Fahrerassistenzsystemen (Quelle: [83])

Funktion \ Sensor		Ultraschall	Radar	Lidar	Video
Längsführungssysteme					
Adaptive Cruise Control (ACC)			A	B	
ACC-Stop&Go		A,B	A	B	
Bremsassistent			A	B	(A,B)
Automatische Notbremse (ANB)			A	B	(A,B)
Querführungssysteme					
Spurverlassenswarner					A
Spurhalteassistent					A
Spurwechselassistent			A		
Längs- und Querführungssysteme					
Einparkassistent		A	B		C
Night Vision					A
Fußgängerschutz ¹⁾			A	B	A,B
Collision Avoidance ¹⁾		A,B	A	B	A,B
Stauassistent ¹⁾		A	A	A	A
sonstige Systeme					
Verkehrszeichenerkennung					A
Fernlichtassistent					A
Müdigkeitserkennung					A

Tabelle 2.1: FAS-Systeme und Sensoren

¹⁾ Diese Systeme befinden sich noch im Entwicklungsstadium

ACC

Die Funktionsweise von ACC stellt sich wie folgt dar: Wie bei einer gewöhnlichen Geschwindigkeitsregelanlage (GRA bzw. Tempomat) wird über einen Bedienhebel am Lenkrad eine Setzgeschwindigkeit eingestellt, die das Fahrzeug selbstständig einhalten soll. Als Erweiterung zu einer gewöhnlichen GRA kommt beim ACC hinzu, dass ein Sensor genutzt wird, um die Geschwindigkeit vorausfahrender Verkehrsteilnehmer zu messen. Diese Information wird dann im Geschwindigkeitsregler dazu verwendet, die Setzgeschwindigkeit, falls nötig, so weit zu reduzieren, dass ein Mindestabstand zum Vorderfahrzeug stets eingehalten wird. Abbildung 2.2 visualisiert den typischen Einsatzbereich von ACC.



Bild 2.2: Anwendungsgebiet ACC (Quelle: Audi AG [2])

ACC ist seit 1999, durch die Einführung in der S-Klasse von Daimler-Benz, für Serienfahrzeuge in Europa erhältlich [83]. Dieses ACC-System verfügte über einen Radarsensor mit Steuergerät, die von der Firma A.D.C gefertigt wurden. Auch andere Hersteller in Europa setzen verstärkt auf die Radartechnologie. Jedoch sind auf dem japanischen Markt vereinzelt Systeme mit Lidar-Sensoren im Einsatz. Lidar bietet sich für Systeme im Mittelklassesegment an, da die Herstellungskosten der Sensoren im Vergleich zu Radarsensoren um einiges günstiger sind. Durch technische Probleme mit der Robustheit auch bei widrigen Umständen hat sich die Lidartechnologie in diesem Bereich jedoch nicht durchsetzen können.

Im Zuge der Verbesserung von Sensorik und Aktorik hat ACC seit der ersten Generation eine stetige Weiterentwicklung erfahren: In der ersten Generation konnte ACC im Geschwindigkeitsbereich von 30 bis 180 km/h aktiviert werden. Die zweite Generation ab 2004 erweiterte den Geschwindigkeitsbereich bereits auf 30 bis 200 km/h. Das nächste Ziel, ACC bis zum Stillstand verfügbar zu machen, wurde 2005 in Serie gebracht. Dabei spricht man häufig von ACC-Stop&Go. Diese Erweiterung der Systemgrenzen erhöht den Komfort im Innerstädtischen sowie in stauähnlichen Situationen erheblich. ACC ist heute unter anderem in Fahrzeugen aller großen Automobilhersteller verfügbar.

Bremsassistent

Das ACC war auch die Grundlage für neue Sicherheitsfunktionen, die sich allgemein unter dem Begriff Bremsassistent zusammenfassen lassen. Dabei reichen Ausprägungsvarianten von rein warnenden Systemen [47] bis hin zu teilautonomen Systemen, die eine Bremsung bereits einleiten und den Fahrer beim Aufbau des maximalen Bremsdrucks unterstützen, wenn der Wunsch nach einer Vollbremsung identifiziert wird [29]. Dadurch können unnötige Verzögerungen beim Einleiten der Vollbremsung durch den Fahrer minimiert werden.

Bei Audi werden diese Bremsassistentensysteme mit dem Begriff *Audi Braking Guard* bezeichnet. Die bislang letzte Generation des Audi Braking Guards, die von 2006 im Q7 in Serie gebracht wurde, ist um die Funktionalität des *Warnrucks* erweitert. Dieser kurze aber spürbare Bremseingriff lenkt die Aufmerksamkeit eines abgelenkten Fahrers intuitiv zurück auf den Straßenverkehr vor ihm, wenn ein Auffahrunfall kurz bevor steht. Die aktuellsten Systeme leiten, auch wenn der Fahrer nicht reagiert, sogar eine autonome Vollbremsung ein, wenn der Unfall nicht mehr verhindert werden kann. Dabei spricht man von Systemen zur Unfallschwereminderung (Collision Mitigation).

Als Informationslieferant werden die Sensoren des jeweiligen ACC-Systems mitgenutzt. Wegen der schlechten lateralen Auflösung des Radarsensors ist es zu riskant die Systeme noch autonom auszulegen. So gibt es bisher noch kein System, das autonom eine Vollbremsung (ANB) einleitet, um einen Unfall tatsächlich zu vermeiden (Collision Avoidance). Dazu müsste die Lage und Ausdehnung des Kollisionsobjektes gerade im Nahbereich sehr genau bekannt sein.

Einparkassistent

Auch bei diesem Systemtyp gibt es verschiedene Ausprägungen. Die bekannteste ist wohl die Einparkhilfe, die schon bei vielen Mittel- und Oberklassefahrzeugen zur serienmäßigen Ausstattung gehört. Diese vermisst während des Einparkens mittels Ultraschallsensoren im Front und Heck des Fahrzeugs den Abstand zum nächsten Hindernis und gibt dem Fahrer visuelle oder akustische Rückmeldungen. Ihren Ursprung haben diese Systeme in den so genannten „Heckflossen“ bei Großfahrzeugen in den 60er Jahren, mit denen der Fahrer das Ende des eigenen Fahrzeugs besser einschätzen kann. Die heutigen aktiven Systeme werden in zwei Varianten bezüglich der Informationsaufnahme angeboten: ultraschall- und radarbasiert[23]. Abbildung 2.3 zeigt den Aufbau und das Anwendungsgebiet der Einparkhilfe auf Ultraschallbasis.

Die erste funktionierende Ultraschall-Einparkhilfe wurde im VW-Konzern entwickelt und von der Firma Hella zur Serienreife gebracht[98]. Die Automobilfirmen führen die Einparkhilfssysteme unter den Namen *APS* (Acoustic Parking System, Audi), *PDC* (Park Distance Control, BMW), *ParkTronic* (Daimler) oder *ParkPilot* (Volkswagen). Heute wird das System bereits bis zu Geschwindigkeiten von 20 km/h verwendet, um z. B. ACC bei der Messung des Abstandes zum vorausfahrenden Fahrzeug zu unterstützen und den

2.1 Fahrerassistenzsysteme mit maschineller Wahrnehmung

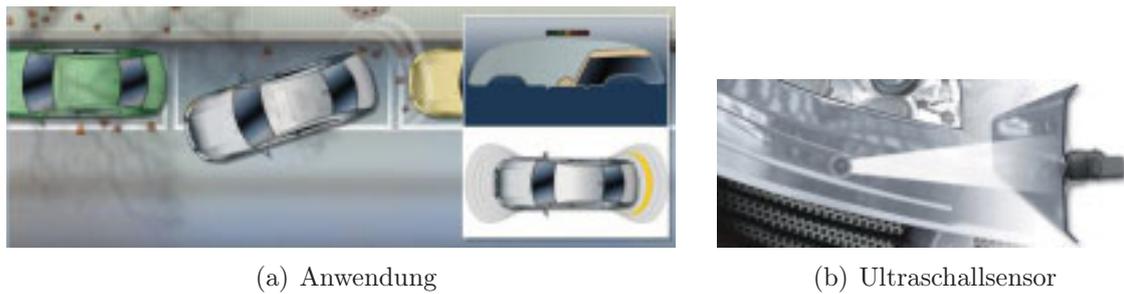


Bild 2.3: Einparkhilfe mit Ultraschallsensoren (Quelle: Bosch AG [18])

Sichtbereich im Nahfeld zu vergrößern. Nachteilig wirkt sich die Fehleranfälligkeit des Ultraschallsensors bei vorhandenen Störquellen aus – wie Presslufthammer oder LKW-Luftventile.

Im Zusammenhang mit ACC brachte Daimler 2006 ein radarbasiertes Einparkhilfssystem auf den Markt. Dadurch können die Kosten der Ultraschallsensoren gespart, eine höhere Reichweite erreicht und die Störanfälligkeit reduziert werden. Allerdings ist dieses System nur in Kombination mit der ACC-Variante DISTRONIC-Plus von Daimler zu haben [38].

Gerade bei großen Fahrzeugen wird häufig eine Rückfahrkamera dazu eingesetzt, um die Übersichtlichkeit bei Einparkmanövern zu steigern. Diese Anwendung ist in Abbildung 2.4 dargestellt.

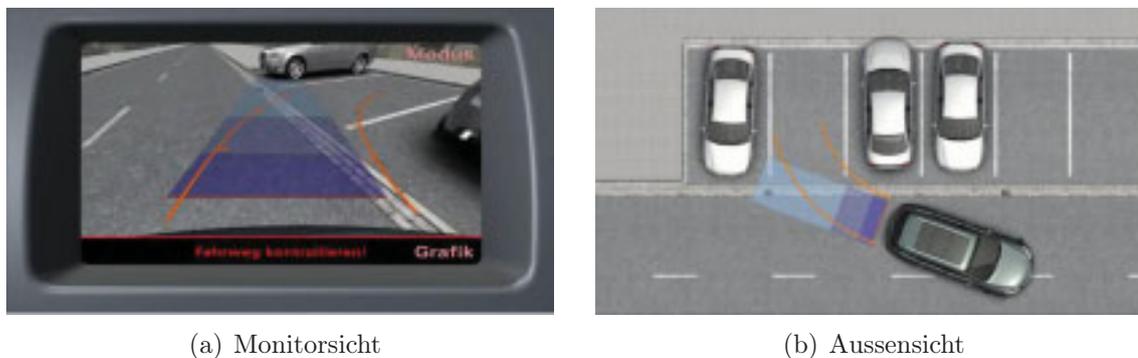


Bild 2.4: Einparkhilfe durch Rückfahrkamera (Quelle: Audi AG [2])

Mittlerweile gibt es Einparkassistenten, welche den informierenden und warnenden Charakter der Einparkhilfe um aktive Unterstützung erweitern. Sie vermessen beim Vorbeifahren die Größe der Parklücke und führen die nötigen Lenkmanöver automatisch durch. Dazu wird das System um zwei zusätzliche Ultraschallsensoren erweitert, die an der Fahrzeugseite angebracht sind. Der Fahrer hat weiterhin die Kontrolle über die Längsregelung des Fahrzeugs und bleibt somit verantwortlich über die Gesamtsituation [91].

Spurwechselassistent

Der Spurwechselassistent arbeitet mit Radarsensoren, die den rückwärtigen Bereich des Fahrzeugs überwachen. Wenn ein Fahrzeug im toten Winkel des Rückspiegels fährt, weisen LED-Leuchten am Außenspiegel den Fahrer auf die Gefahrensituation hin. Setzt der Fahrer dennoch den Blinker, wird die Warnung durch Blinken der LEDs verstärkt. Manche Systeme setzen zusätzlich das Vibrieren des Lenkrads als Warnung ein.

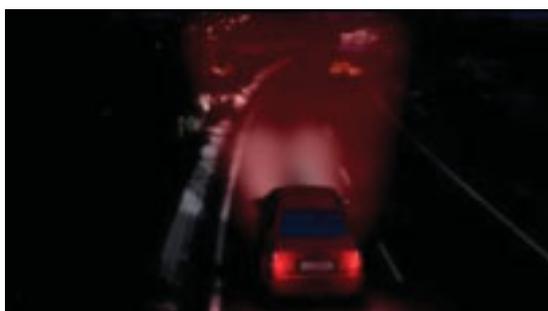


Bild 2.5: Spurwechselassistent (Quelle: Audi AG [2])

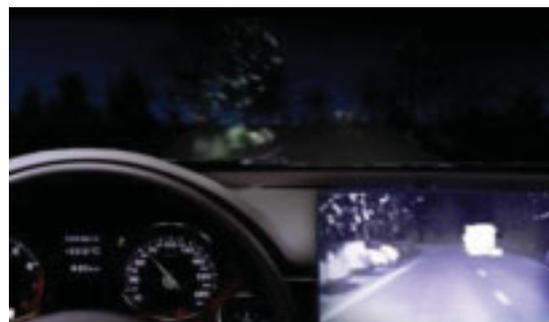
Abbildung 2.5 zeigt einen typischen Anwendungsfall für den Spurwechselassistenten.

Kamerabasierte Systeme

Neben der Rückfahrkamera der Einparkhilfe erhalten verstärkt kamerabasierte Systeme Einzug ins Fahrzeug, die den Bereich vor dem Fahrzeug analysieren sollen. Dabei werden echtzeitfähige Bildverarbeitungsalgorithmen eingesetzt, um die benötigte Information aus dem Bild zu extrahieren (mehr zum Thema Bildverarbeitung kann in Kapitel 4 nachgelesen werden).



(a) Night-Vision-Simulation



(b) Night-Vision-Bild

Bild 2.6: Beispiel für Night-Vision-System (Quelle: Bosch AG [18])

Das erste System dieser Art war das *Night Vision*-System (siehe Abbildung 2.6). Dieses nutzt die Infrarotsensibilität des Kamerasensors, um dem Fahrer bei Nacht eine bessere

2.1 Fahrerassistenzsysteme mit maschineller Wahrnehmung

Fernsicht zu gewährleisten. Dazu wird der Bereich vor dem Fahrzeug durch IR-Licht ausgeleuchtet, die IR-Reflexionen von der Kamera aufgenommen und in den sichtbaren Bereich transformiert. Die entstehenden Videobilder werden über ein Display im Fahrzeuginneren angezeigt. Das ergibt ein kontrastreiches Bild der Umgebung, ohne dass andere Verkehrsteilnehmer geblendet werden. Somit wird der Sichtbereich in der Nacht von ca. 40 auf bis zu 150 Metern erweitert.

Weitere wichtige Aufgaben, die ausschließlich von Kamerasystemen erledigt werden können, sind die Erkennung des Spurverlaufs, die Verkehrszeichenerkennung, die Steuerung des Fernlichtassistenten sowie die Müdigkeitserkennung des Fahrers.

Das prinzipielle Vorgehen der Spurerkennung wird in dieser Arbeit in Kapitel 4.4 beschrieben. Es gibt bereits heute Systeme in Serienfahrzeugen, die in einfachen Situationen auf Landstraßen und Autobahnen die Spurmarkierung erkennen und dem Fahrer beim Halten der eigenen Fahrspur unterstützen. Dabei wird wieder zwischen warnenden und aktiv eingreifenden Systemen unterschieden. Die Spurverlassenswarnung zeigt durch Vibrationen am Lenkrad an, dass die Markierung der eigenen Fahrspur unbeabsichtigt überfahren wird. Die aktiven Spurhaltesysteme werden häufig als *Heading Control* bezeichnet und nutzen die Klothoideninformation der Spurerkennung, um den richtigen Lenkwinkel einzustellen.

Der Fernlichtassistent passt die Leuchtweite des Fernlichts an die gegebenen Umgebungsbedingungen an. Dabei wird die Position anderer Verkehrsteilnehmer an deren Fahrzeuglichtern erkannt und das Fernlicht für diesen Raumwinkelbereich lokal zurückgenommen.

Dadurch ergeben sich auch bei Nacht Sichtweiten bis zu 300 Metern, ohne dass entgegenkommender Verkehr geblendet wird.

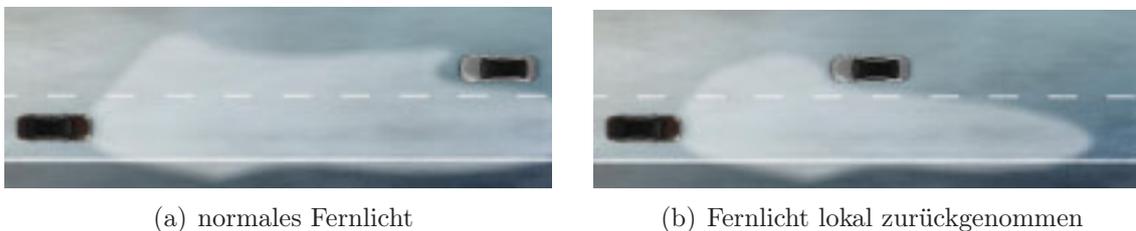


Bild 2.7: Fernlichtassistent (Quelle: Daimler AG [30])

2.2 Forschungsprojekte zur vorausschauenden Fahrerassistenz

Weltweit arbeiten viele Automobilhersteller und Forschungseinrichtungen an Systemen für die Zukunft des Automobils. Gerade der Bereich der Erkennung und Interpretation des Fahrzeugumfeldes hat in den letzten Jahren einen großen Innovationsschub erfahren. Einen Überblick über aktuelle Forschungsergebnisse, speziell im Bereich der Wahrnehmung, kann aus den Veröffentlichungen der IEEE-Konferenzen „Intelligent Vehicles Symposium“ , „Conference on Intelligent Transportation Systems“ , „Pattern Analysis and Machine Intelligence“ und „Transactions on Image Processing“ erlangt werden. Auch im Rahmen der Konferenz „Aktive Sicherheit durch Fahrerassistenz“ , die vom TÜV-Süd und der TU-München regelmäßig organisiert wird, werden neue Entwicklungen vorgestellt. Das Sammelwerk der seit 2004 regelmäßig abgehaltenen Konferenz „Workshop Fahrerassistenzsysteme“ in Walting fasst die aktuellen Probleme und Lösungsstrategien verschiedener Forschungseinrichtung und Automobilkonzerne zusammen [65]. Für allgemeine Informationen zu Neuentwicklungen im Bereich der Fahrerassistenz sei auf die Tagung „FISITA - World Automotive Congress“ hingewiesen.

2.2.1 Historie

1986 wurde das Projekt *PROMETHEUS*²⁾ ins Leben gerufen. Darin waren 13 Automobilhersteller und 19 Europäische Staaten beteiligt. Eine wichtige Komponente dieses Projektes war die Kooperation zwischen der Universität der Bundeswehr München (UBM) und der Daimler-Benz-Forschung. In Deutschland stellte die UBM 1987 das Versuchsfahrzeug für autonome Mobilität und Rechnersehen (VaMoRs) vor. Damit wurde eine vollkommen automatische längs- und quergeregelte Fahrt auf einem 20 Kilometer langen, unbefahrenen Autobahnstück mit einer Geschwindigkeit bis zu 96 km/h durchgeführt. Dadurch wurde die Machbarkeit von Systemen zur automatischen Fahrzeugführung erstmals gezeigt. Die später entwickelten Versuchsträger VITA³⁾ von Daimler-Benz und VaMP (kurz für VaMoRs-Pkw) von der UBM [76][8] demonstrierten eine Fahrt vom Flughafen Charles-de-Gaulle in Paris nach Lille. Dabei wurden bereits Verkehrszeichen berücksichtigt und Spurwechsel vollzogen. Bei einer Versuchsfahrt von München nach Odense in Dänemark konnten 95% der 1600 Kilometer langen Strecke ohne Eingriff des Sicherheitsfahrers bewältigt werden [63]. Die Forschungsarbeiten aus jener Zeit bilden heute noch die Grundlagen im Bereich der maschinellen Wahrnehmung für Fahrerassistenzsysteme wie sie heute auf dem Markt verfügbar sind [69][36][58][100].

Der Forschungsdrang in Japan wurde durch die Tätigkeiten in Europa ausgelöst. Das japanische Ministerium für Handel und Industrie (MITI⁴⁾) und die Firmen Nissan und Fujitsu haben Anfang der 90er Jahre gemeinsam das Projekt „Personal Vehicle System“ ins Leben

²⁾ PROgramMme for a European Traffic of Highest Efficiency and Unprecedented Safety

³⁾ VIsion Technology Application

⁴⁾ Ministry of International Trade and Industry

gerufen. Daraus ergab sich 1996 die Advanced Cruise-Assist Highway System Research Association (AHSRA), die eine Verbindung zwischen japanischen Forschungseinrichtungen und der Automobilindustrie herstellte [88]. Diese Aktivitäten und die Öffentlichkeitsarbeit haben dazu geführt, dass der japanische Markt einer der innovativsten und häufig der Einstiegsmarkt für neue Fahrerassistenzsysteme ist.

Die Carnegie Mellon Universität (CMU) hat die Vorreiterrolle in den USA übernommen. Bereits 1984 wurden die ersten automatischen Fahrzeugsysteme aufgebaut. Über die Jahre wurden die Versuchsfahrzeuge Navlab1 bis Navlab11 entwickelt, womit verschiedenartige Fahrscenarien umgesetzt wurden. So konnte 1995 Navlab5 mit automatischer Querregelung von der amerikanischen West- zur Ostküste fahren, wobei 98% der Strecke ohne Fahrereingriff bewältigt wurden[48]. Erst 1995 wurde das National Automated Highway System Consortium (NAHSC) von der Regierung eingerichtet, welches 1997 die Intelligent Vehicle Initiative (IVI) startete. Das brachte den nötigen finanziellen Anschlag für amerikanische Unternehmen, wie Ford und GM [88], die Entwicklung neuer Fahrerassistenzsysteme ebenfalls anzutreiben.

2.2.2 Aktuelle Forschungsprojekte

Das Ziel aller Forschergruppen auf der Welt ist das Gleiche: durch die fortschreitende Automatisierung des Autofahrens soll das Autofahren angenehmer und sicherer werden. Ein besonderes Ziel der Europäischen Union ist die Halbierung der Unfalltoten bis 2010 im Vergleich zum Jahr 2000. Um dieses Ziel in erreichbare Nähe zu bringen sind in den letzten Jahren verstärkt Forschungsk Kooperationen zwischen Industrie und Hochschulen entstanden. In diesem Umfeld wird auch die sensorgestützte Fahrzeugumfeldwahrnehmung weiter erforscht.

AKTIV

Die Forschungsinitiative AKTIV [44] steht für „Adaptive und Kooperative Technologien für den Intelligenten Verkehr“ . An der deutschen Forschungsinitiative sind 28 Partner aus der Industrie und Forschung beteiligt. Dabei werden die Schwerpunkte „Verkehrmanagement“ , „Aktive Sicherheit“ und „Cooperative Cars“ verfolgt [42].

Das Themengebiet „Aktive Sicherheit“ umfasst die Weiterentwicklung der Fahrerassistenzsysteme „Aktive Gefahrenbremsung (AGB)“ , „Integrierte Querführung (IQF)“ , „Kreuzungsassistenz“ , „Sicherheit für Fußgänger und Radfahrer“ und „Fahrtsicherheit und Aufmerksamkeit“ [43]. Hierbei werden speziell an die Anforderungen der Systeme angepasste Algorithmen zur lokalen Umfelderkennung entwickelt. Bei AGB entsteht eine Sensorfusionsplattform zwischen Radar- und Videodaten, um eine Automatische Notbremsung möglichst früh einleiten zu können. Bei IQF wird die Spurerkennung dazu verwendet das Fahrzeug aktiv in der Spur zu halten. Im Unterschied zu herkömmlichen Systemen soll das auch im Niedergeschwindigkeitsbereich und bis zum Stillstand funktionieren. Die

2 Stand der Technik

Kreuzungsassistenten verbindet Radar- und Lasersensorik mit den Möglichkeiten der Car-To-Car-Kommunikation. Dadurch sollen Unfälle im Kreuzungsbereich vermieden werden. Die Bereiche „Sicherheit für Fußgänger und Radfahrer“ und „Fahrsicherheit und Aufmerksamkeit“ setzen verstärkt auf Videosensorik und Bildverarbeitung, um Informationen über den Innenraum und das Umfeld des Fahrzeugs zu erhalten. Spezielle Klassifikationsalgorithmen erkennen Fußgänger und Radfahrer und stellen somit die Grundlage für Warn- und Sicherheitmechanismen dar. Die Innenraumkamera hat den Fahrer selbst im Blickfeld und detektiert den Lidschlag und die Blickrichtung des Fahrers, um eine Einschätzung des Aufmerksamkeitsgrades zu ermöglichen.

Im Jahr 2005 wurde das Vorgängerprojekt INVENT abgeschlossen [25]. Dieses wies eine ähnliche Organisationsstruktur wie AKTIV auf. Dabei wurden bereits die Themengebiete „Fahrerassistenz/Aktive Sicherheit“ (FAS/AS), „Verkehrsmanagement 2010“ und „Verkehrsmanagement in Transport and Logistik“ untersucht. Im Teilprojekt FAS/AS wurden die Themen „Fahrumgebungserfassung und Interpretation“ , „Vorausschauende Aktive Sicherheit“ und „Stauassistenten“ betrachtet. Eine Akzeptanzstudie ergab, dass die Fahrer eines Fahrzeugs ein Assistenzsystem wie den Stauassistenten positiv empfinden würden. Es wurden Versuchsfahrzeuge von Audi, Bosch, Opel, Siemens VDO und BMW zur Umsetzung verschiedener Funktionsausprägungen aufgebaut[52]. Die Fahrzeuge wurden dafür mit Radar-, Video- und Lidarsensoren ausgestattet, um den Fern- und Nahbereich abdecken zu können. Die Sensordaten wurden von einer eigens entwickelten Sensorfusionssoftware aufbereitet und der Objektbildung zur Verfügung gestellt[51]. Auf Basis des entwickelten Sensorpakets konnten bereits stauähnliche Situationen automatisch abgefahren werden [60].

Im Umfeld dieser Entwicklungen hat die Daimler-Forschung mit der Forschungsgruppe um Dr. U. Franke ein Bildverarbeitungssystem aufgebaut, das für zukünftige Fahrerassistenzsysteme einen enormen Informationszugewinn darstellen könnte. Es ermöglicht, ähnlich wie die menschliche visuelle Wahrnehmung, die gleichzeitige Auswertung der Entfernung- und Bewegungsinformation [45]. Dabei werden 3D-Merkmalpunkte mittels Stereobildverarbeitung gewonnen und deren Bewegung über die Zeit mittels Kalman-Filterung verfolgt. Die Bezeichnung 6D-Vision leitet sich aus der Kombination der 3D-Ortsinformation mit der 3D-Bewegungsinformation ab.

Abbildung 2.8 zeigt die geschätzten Bewegungsrichtungsvektoren der verfolgten Merkmalspunkte. Die Vektoren ergeben sich aus der Prädiktion der Bewegung auf einen Zeitpunkt 0.5 Sekunden in der Zukunft. Der Ansatz zeigt viel versprechende Möglichkeiten zur Detektion und Verfolgung verschiedenartiger Objekte. Allerdings wird für den Einsatz des Systems eine Stereokameraanordnung benötigt. Aus ökonomischen und designtechnischen Gründen ist der Einsatz von Stereokamerasystemen bei Serienfahrzeugen jedoch noch nicht absehbar.



Bild 2.8: Messergebnisse des 6D-Vision-Ansatzes (Quelle: [45])

Kognitive Automobile

Auch unabhängig von der Industrie machen es sich Forschungseinrichtungen zur Aufgabe, das autonome Fahren der Realität näher zu bringen. Im Jahr 2006 wurde von der Deutschen Forschungsgemeinschaft der Sonderforschungsbereich (SFB) Transregio Kognitive Automobile gegründet. Dabei arbeiten Wissenschaftler der Karlsruher Universität, des Forschungszentrums und des Fraunhofer IITB eng zusammen mit Forschern der Technischen Universität und der Universität der Bundeswehr in München. Im Vordergrund steht die Entwicklung von Systemen, die dem Fahrzeug kognitive Fähigkeiten verleihen, so dass sich ein Fahrzeug autonom durch den Straßenverkehr bewegen kann [34]. Dabei werden zum einen Systeme zur lokalen Wahrnehmung entwickelt, zum anderen aber auch kooperative Systeme, um den Sichthorizont der einzelnen Fahrzeuge zu erweitern[84]. Dazu kommen Video-, Radar- und Lidarsensoren zum Einsatz, die eine Rundumsicht um das Fahrzeug ermöglichen. Abbildung 2.9 zeigt das Fahrzeug „AnniWAY“ der Universität Karlsruhe mit der Visualisierung der Daten des Velodyne-Laserscanners[85].

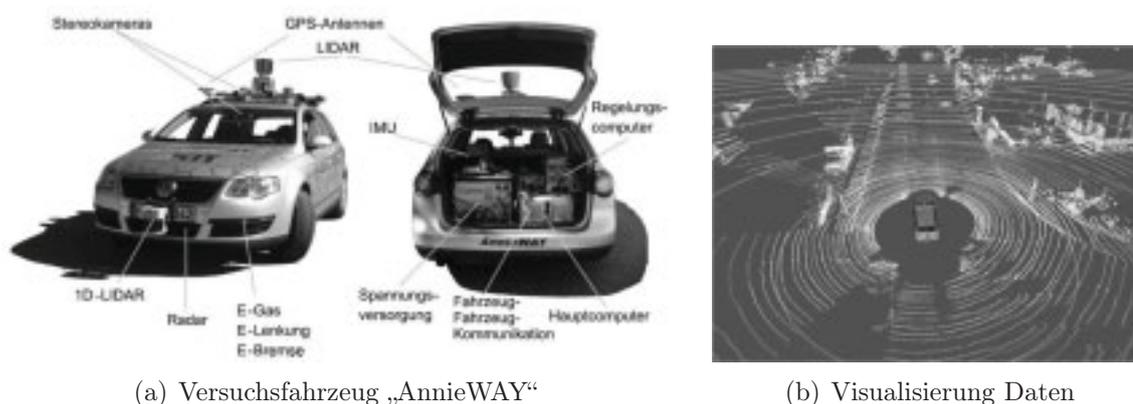


Bild 2.9: Kognitive Automobile (Quelle: [85])

Angelehnt an das menschliche Vorbild kommt der videobasierten Umfelderkennung hierbei eine große Bedeutung zu. So ist ein aktives multifokales Kamerasystem im Einsatz, das

durch eine Aufmerksamkeitssteuerung bestimmte Bereiche gezielt genauer untersuchen kann. Die aktive Kameraausrichtung ermöglicht zudem mittels Inertialsensorik ein stabilisiertes Kamerabild, so dass die starken Nickbewegungen des Fahrzeugs relativ zur Straße keine negative Auswirkung auf die Wahrnehmungsgüte mehr hat.

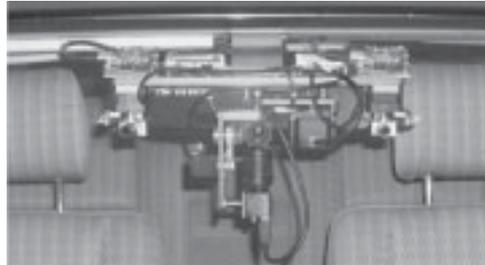


Bild 2.10: Aktives multifokales Kamerasystem (Quelle: [85])

Das Fahrzeug „AnniWAY“ kam auch beim internationalen Wettbewerb „Urban Challenge“ zum Einsatz und konnte seine Fähigkeiten durch das Erreichen des Finales unter Beweis stellen.

DARPA Grand/Urban Challenges

Im März 2004 startete das US Ministerium *Defense Advanced Research Projects Agency (DARPA)* die Initiative „Grand Challenge“ [31]. Hier sollten 15 Fahrzeuge vollständig autonom eine 400 Kilometer lange Strecke durch die Wüste in einer bestimmten Zeit durchfahren. Dadurch wollte das US Militär neue Möglichkeiten für unbemannte Operationen mit Militärfahrzeugen schaffen[102]. Es waren lediglich GPS-Wegpunkte vorgegeben, unvorhergesehene Behinderungen auf der Strecke mussten jedoch vom System selbstständig erkannt und umfahren werden, ohne dass ein Sicherheitsfahrer oder ein Entwickler die Situation beobachten oder in das Geschehen eingreifen konnte. Es war ein Preisgeld von 1 Million Dollar ausgeschrieben. Das „Red Team“ der CMU war mit 7 bewältigten Meilen das beste Team [88].

Dadurch wurde ein beispielloser Konkurrenzkampf für die „Grand Challenge 2005“ im darauf folgenden Jahr entfacht. Dieser konnte gleich von mehreren Teams erfolgreich durchfahren werden, wobei das Team der Universität Stanford in Kooperation mit der Volkswagen AG mit dem gemeinsamen Fahrzeug „Stanley“ die beste Zeit erreichte [88].

Die DARPA „Urban Challenge 2007“ stand unter einem neuen Erwartungsdruck. Auch waren die Anforderungen diesmal, wie die Namensänderung schon andeutet, anders als in den Jahren zuvor. Sie sollte „den Fortschritt und die Leistungsfähigkeit autonomer Fahrzeuge öffentlich darstellen und beschleunigen“ [85]. Das Ziel war die unfallfreie Bewältigung mehrerer Fahrscenarien, wie Überholen und Dreipunkt-Wendemanöver, in einer vorstadtähnlichen Umgebung mit anderen Verkehrsteilnehmern. Dabei mussten die kalifornischen Verkehrsregeln beachtet werden. 11 der insgesamt 89 angetretenen Teams standen am 3. November 2007 in Victorville (Kalifornien) im Finale.

2.2 Forschungsprojekte zur vorausschauenden Fahrerassistenz



(a) Grand Challenge 2005: „Stanley“ (Quelle: [32])



(b) Urban Challenge 2007: „Boss“ (Quelle: [33])

Bild 2.11: Gewinner der DARPA-Challenges

Abbildung 2.11 zeigt die Gewinnerfahrzeuge der Jahre 2005 („Stanley“, Stanford University) und 2007 („Boss“, Carnegie Mellon University). Die Kosten der Sensoren übersteigen dabei leicht die Kosten der Fahrzeuge. Auch die serienreife Integrationsfähigkeit der Aufbauten scheint unmöglich.

Im Gegensatz zu vielen Forschungsarbeiten auf dem Gebiet der autonomen Fahrzeugführung, wo häufig der Schwerpunkt auf dem Zeigen der Machbarkeit liegt, muss im seriennahen Entwicklungsumfeld bedacht werden, dass sich die Kosten für ein Fahrerassistenzsystem im Rahmen halten müssen. Auch die Integrationsfähigkeit der Module ins Gesamtfahrzeug muss gegeben sein. Davon sind die autonom fahrenden Forschungsfahrzeuge heute noch weit entfernt.

3 Beschreibung des Gesamtsystems

Dieses Kapitel beschreibt die technischen Komponenten des Versuchsfahrzeugs und erarbeitet die Grundlagen für deren weitere Verwendung. In Abschnitt 3.1 wird der Aufbau des Versuchsfahrzeugs geschildert. In 3.2 wird auf die zum Einsatz kommenden Sensoren mit den verwendeten Sensormodellen eingegangen. Die verwendete Sensorkonfiguration wird dann in 3.3 zusammengefasst dargestellt. Unterkapitel 3.4 beschreibt die Grundlagen zum Umgang mit Koordinatensystemen und gibt einen Überblick über die in dieser Arbeit verwendeten Koordinatensysteme. Abschließend wird in Abschnitt 3.5 die zur Algorithmenentwicklung eingesetzte Softwareentwicklungsumgebung ADTF (Automotive Data and Time Triggered Framework) beschrieben.

3.1 Aufbau des Versuchsfahrzeugs

Zum Aufnehmen von Sensordaten, zum Evaluieren von Sensoren sowie zum Testen von entwickelten Softwarekomponenten im realen Betrieb, wurde ein Fahrzeug des Typs A6 Limousine mit mehreren Rechnern, einer zusätzlichen Stromversorgung und den zur Verfügung stehenden Sensoren ausgestattet. Die folgenden Sensoren werden in dieser Arbeit verwendet:

- **Long Range Radar der zweiten Generation (LRR2) von Bosch:** Das Radar ist bereits durch den Serieneinsatz für ACC im Fahrzeug verbaut und nach in Fahrtrichtung ausgerichtet. Dieser Sensor verfügt intern über ein Trackingverfahren, das die Radarrohdaten bereits zu Objekten zusammenfasst. Diese Objektbeschreibungen werden in einer Botschaft über den CAN-Bus übertragen.
- **Graubild-Videokamera von Bosch:** Die Kamera ist nach vorne ausgerichtet. Über einen speziellen Framegrabber können die Bilder der Kamera ausgelesen werden und für die eigene Bildverarbeitung genutzt werden.
- **Laser-Scanner im Prototypenstatus des Ingenieurbüros Spies:** Der Laserscanner schaut ebenfalls nach vorne, und es kann ebenfalls über CAN auf die Rohdaten zugegriffen werden. Der Sensor liefert für jeden Laserkanal die Entfernung zu reflektierenden Hindernissen.

Abbildung 3.1 zeigt das Fahrzeug mit einer Übersicht über die eingebauten Sensoren.

Zusätzlich zu den Sensoren ist im Kofferraum des Fahrzeugs ein Box PC verbaut, der mit zwei CAN-Karten und der Framegrabberkarte für die Kamera ausgestattet ist. Auf diesem

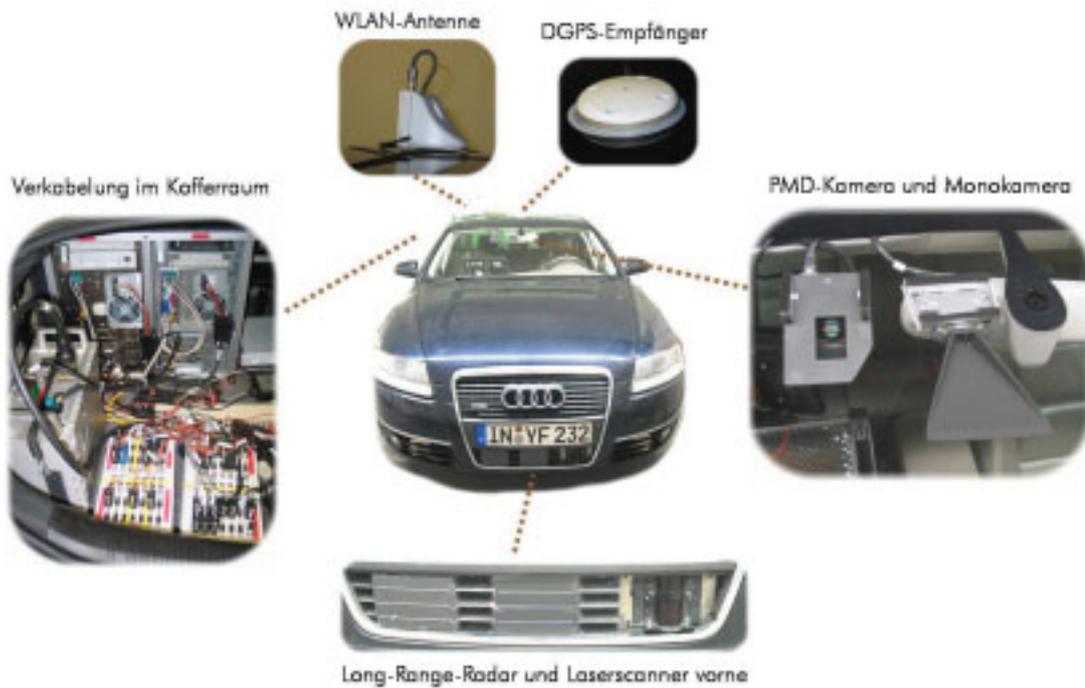


Bild 3.1: Aufbau Versuchsträger

Rechner können die Daten aufgezeichnet werden und auch neue Algorithmen während der Fahrt getestet werden. Als Rahmensystem kommt dafür die Entwicklungsumgebung ADTF zum Einsatz, das anschließend beschrieben wird.

3.2 Physikalische Beschreibung der Sensoren

Ein Sensor beschreibt eine physikalische Einheit, welche Informationen über eine Eingangsschnittstelle empfängt, diese intern verarbeitet und die verarbeiteten Daten über eine Ausgangsschnittstelle bereitstellt. Bedingt durch die unterschiedlichen physikalischen Eigenschaften der Mess- und Verarbeitungsprinzipien verschiedener Sensortypen unterscheidet sich die Information, die man von ihnen über die Umwelt erhalten kann stark von einander. Bei der Weiterverarbeitung der Sensorausgangsdaten (im Folgenden Sensordaten genannt) ist es daher von großer Bedeutung ein genaues Modell der physikalischen Eigenschaften der Datenquelle zu haben. Gleichzeitig müssen auch die speziellen Bedingungen, die am Einsatzort das Sensorverhalten beeinflussen, berücksichtigt werden. Deshalb wird im Abschnitt 3.2.1 zunächst ein Überblick über die, für Fahrerassistenzsysteme verwendeten, Sensoren gegeben. Anschließend werden die in dieser Arbeit verwendeten Sensoren (Radar, Lidar, Mono-Video) hinsichtlich ihres Funktionsprinzips und des ver-

3 Beschreibung des Gesamtsystems

wendeten Sensormodells beschrieben. Die sensorinterne Vorverarbeitung der Sensorrohdaten (Diodenströme, Strahlungsintensitäten, Phasenverschiebungen ...) für die Berechnung der interessierenden Informationen (Helligkeiten, Entfernungen, Geschwindigkeiten, ...) werden in dieser Arbeit nicht näher untersucht und in den Sensormodellen nur stark vereinfacht berücksichtigt.

Ein Sensormodell spiegelt das Wissen über die physikalischen und algorithmischen Eigenschaften des gesamten Sensorsystems wider, um die ausgegebenen Daten richtig interpretieren zu können. Es beschreibt den Übergang der physikalischen Eingangsdaten zu den interessierenden Sensordaten. In das Sensormodell können geometrische und messprinzipbedingte Effekte eingehen. Je nach Anspruch an die Genauigkeit der Messauswertung muss ein Sensormodell mit entsprechender Güte benutzt werden. Durch Invertierung des Sensormodells können die vom Sensor ausgegebenen Daten in die eigentlich interessierende physikalische Information zurückgerechnet werden. Das inverse Sensormodell kann auch dazu verwendet werden, eine Messerwartung zu generieren, was für die Assoziation und die Plausibilisierung von Sensordaten genutzt werden kann.

Da der Hauptfokus dieser Arbeit auf Bildverarbeitung liegt, wird das Sensormodell der Videokamera in Unterkapitel 3.2.4 detaillierter betrachtet.

3.2.1 Einordnung von Sensortypen

Die im Automobilbereich eingesetzten Sensoren können grundsätzlich in die Kategorien „aktiver“ bzw. „passiver“ Sensor eingeordnet werden. Dabei werden als aktive Sensoren diejenigen bezeichnet, die zusätzlich zu der Empfangseinheit eine Sendeeinheit haben, um die Umwelt „aktiv“ wahrnehmbar zu machen. Tabelle 3.1 zeigt die Kategorisierung der Sensoren im Überblick.

Passive Sensoren	Aktive Sensoren
Mono-Video	Radar
Stereo-Video	Ultraschall
Fern-Infrarot	Lidar (Laser)
GPS	Nah-Infrarot
Inertial	PMD ¹⁾

Tabelle 3.1: Einteilung der Sensoren für FASe nach aktiv / passiv

Bei den Sensoren, die der Vorausschau dienen (Video, Radar, Ultraschall, Lidar, Infrarotkamera), lässt sich ein Überhang an aktiven Sensoren erkennen. Dies kann wohl damit erklärt werden, dass diese Sensortypen eine geringere Abhängigkeit der Informationsqualität von den Umgebungsbedingungen zeigen. Da die Umgebung im Automobilbereich je nach Einsatzort, -zeit und Witterung stark unterschiedlich sein kann, ist diese Eigenschaft

¹⁾ Kameraähnlicher Sensor aus der Familie der Photomischdetektoren (engl.: *Photonic Mixer Device*). Durch ausgesendetes moduliertes Infrarotlicht kann jedem Pixel ein Distanzwert zugewiesen werden.

aktiver Sensoren ein großer Vorteil gegenüber passiven. Ein weiterer Vorteil ist zudem, dass durch das Wissen über den Sendezeitpunkt mit diesen Sensoren meist die Entfernung zu Fremdobjekten direkt gemessen werden kann. Macht man sich nun bewusst, dass der Mensch ohne Einsatz jeglicher aktiver Sensorik auch komplexe Fahrsituationen gut meistern kann, wird man sich erst klar über dessen großartige kognitive Leistung.

Eine weitere Einteilung lässt sich hinsichtlich der Art der auszuwertenden Information vornehmen, sprich mit welchem „Sinn“ der Sensor arbeitet. Hierbei fällt auf, dass fast alle Sensoren aus Tabelle 3.1 mit elektromagnetischen Wellen arbeiten. Allein der Wellenlängenbereich ist unterschiedlich. In [16] kann ein Überblick über aktuelle, für automotiv taugliche, optische Sensoren gewonnen werden. Nur der Ultraschallsensor arbeitet, wie der Name schon vermuten lässt, mit akustischer Information. Das bedeutet, dass in Analogie mit dem Menschen das Fahrzeug bis heute die Sinne „Hören“ und „Sehen“ erworben hat. Wenn man die Informationsverarbeitung von Infrarotlicht als Wärmedetektion auffasst, könnte man das „Fühlen“ (im Sinne von Wärme fühlen) noch mit aufführen.

Die in dieser Arbeit verwendeten Sensoren benutzen ausschließlich elektromagnetische Wellen als Informationsträger. Die Sensoren arbeiten jedoch in unterschiedlichen Frequenzbereichen, was sich wiederum auf die sinnvollen Einsatzbereiche und die auszuwertende Information auswirkt. Bild 3.2 zeigt den Arbeitsbereich der eingesetzten Sensortypen.

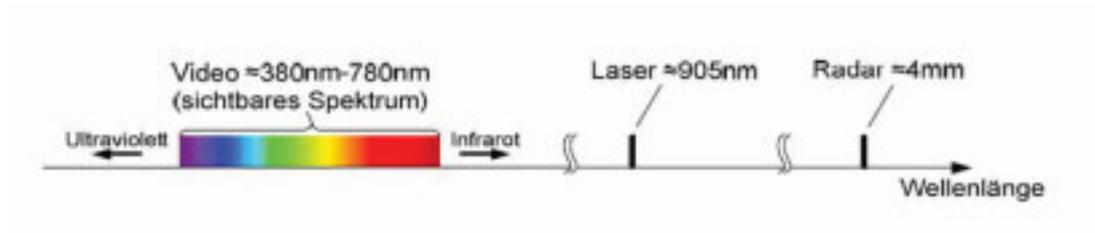


Bild 3.2: Wellenlängenspektrum der Sensoren

3.2.2 Radar

Der Radarsensor wird seit langem sehr stark im militärischen Bereich, wie auch in der zivilen Luftfahrt zur Ortung von Flugzeugen eingesetzt. Das Akronym „Radar“ stand deshalb früher für „Radio Aircraft Detection And Ranging“. Wegen der Erweiterung der Einsatzgebiete außerhalb des Flugzeugbereichs wird heute eher die ausgeschriebene Form „RADIO Detection And Ranging“ verwendet. Dieser Sensor gehört ebenfalls in die Kategorie der aktiven, entfernungsmessenden Sensoren.

Funktionsprinzip

Der Radarsensor nutzt elektromagnetische Wellen als Informationsquelle. Diese werden über Antennen ausgesendet und wieder empfangen. Durch Laufzeitmessung kann auf die Entfernung des reflektierenden Objektes geschlossen werden.

3 Beschreibung des Gesamtsystems

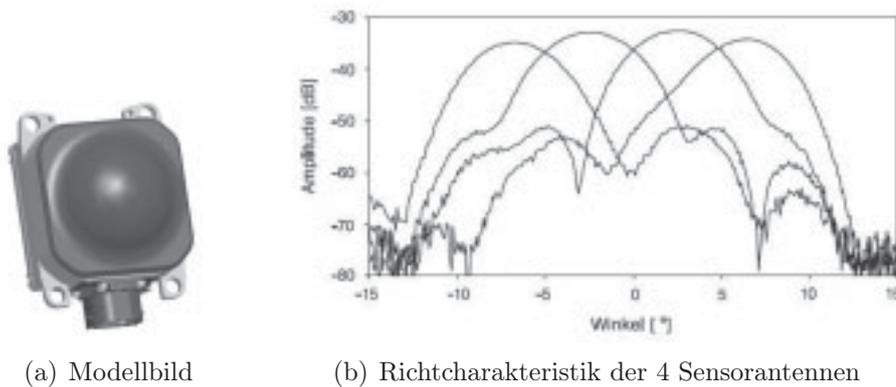


Bild 3.3: LRR2-Sensor von Bosch

Abbildung 3.3(a) zeigt ein Modell des verwendeten Long-Range-Radar-Sensors der zweiten Generation (LRR2) der Firma Bosch. Ein Beispiel für die Richtcharakteristik der vier internen Antennen ist in der Grafik 3.3(b) abgebildet.

Die vom Radargerät ausgesendeten Wellen werden an elektrisch leitenden Oberflächen (metallische Gegenstände) oder dielektrischen Grenzflächen (Erdboden, Wasser, usw.) reflektiert und können vom Empfangsteil des Radars wieder aufgenommen werden. Zusätzlich zur Laufzeitmessung kann der Dopplereffekt zur direkten Geschwindigkeitsmessung des reflektierenden Objektes genutzt werden.

Laufzeitmessung Gleichung 3.1 beschreibt die Errechnung des Abstands zum reflektierenden Objekt aus der Laufzeit der elektromagnetischen Welle. Dabei muss beachtet werden, dass die Welle den doppelten Weg zurücklegen muss.

$$D_{r,LRR} = 0.5 \cdot \Delta t \cdot c_{\text{Licht}} \quad \text{mit } c_{\text{Licht}}: \text{Lichtgeschwindigkeit} \quad (3.1)$$

Dopplereffekt Der Dopplereffekt beruht auf der Tatsache, dass die wahrgenommene Frequenz einer elektromagnetischen Welle von der relativen Geschwindigkeit des Senders abhängt. Das bedeutet, dass die reflektierte Welle durch die Relativgeschwindigkeit des reflektierenden Objektes eine abweichende Frequenz zu der ausgesendeten Welle aufweist. Gleichung 3.2 beschreibt den physikalischen Zusammenhang.

$$\Delta f_{LRR} = \frac{(-2 \cdot f \cdot v_{\text{rel}})}{c_{\text{Licht}}} \quad (3.2)$$

Winkelmessung Durch den Einsatz mehrerer Antennen in einem Radarsystem kann zusätzlich zur Entfernung- und Geschwindigkeitsinformation der Raumwinkel des Objektes ermittelt werden (siehe Bild 3.4).

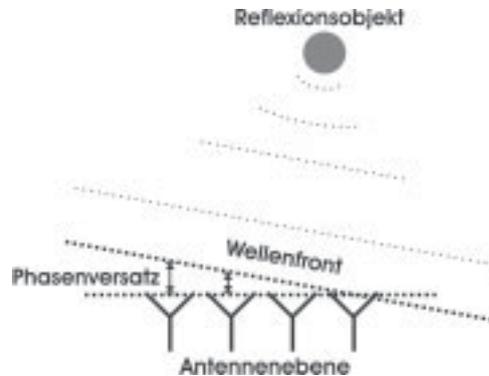


Bild 3.4: Bestimmung des Raumwinkels

Da im Fernbereich ($D_{T,LRR} \gg \lambda$) die Radarwellenfront näherungsweise als linear angenommen werden kann, geschieht dies über die Phasenverschiebung der empfangenen Welle an den verschiedenen Antennen.

Messverfahren Bei Radarsystemen wird zwischen zwei Messverfahren unterschieden: Pulsverfahren und FMCW²⁾-Verfahren. Beim Pulsverfahren wird die Zeit zwischen Ausenden und Eintreffen direkt gemessen, um auf die Entfernung zu schließen. Beim FMCW-Verfahren wird kontinuierlich eine Welle ausgesendet, deren Wellenlänge über die Zeit variiert wird (meist linear). Durch Auswertung des Differenzsignals zwischen Sendefrequenz und Empfangsfrequenz kann gleichzeitig auf die Entfernung und die relative Geschwindigkeit des reflektierenden Objektes geschlossen werden. Da beim Pulsverfahren wegen der hohen geforderten Abtastrate sehr hochwertige Komponenten nötig sind, werden vermehrt FMCW-Radare im Automobilbereich eingesetzt. Auch der in diesem Projekt eingesetzte Radar der Firma Bosch (LRR2) greift auf dieses Verfahren zurück. Auf eine detaillierte Beschreibung der Verfahren wird in dieser Arbeit verzichtet und auf einschlägige Literatur der Radartechnologie verwiesen, z. B. [67] und [50].

Sensormodell

Das hier eingesetzte Radar ist im Zuge der Entwicklung von ACC entstanden, arbeitet mit einer Wellenlänge von ca. 3.8 mm (77 GHz) und kann für eine Vorausschau von bis zu 200 Metern eingesetzt werden[18]. Der Sensor verfügt über ein internes Steuergerät zur Datenauswertung. Dieses führt die Signalverarbeitung durch und verfügt bereits über eine Objektbildungs- und Trackingeinheit. Die intern getrackten Objekte werden über eine Datenschnittstelle auf dem CAN-Bus übertragen. Dabei können maximal 32 Objekte

²⁾ Frequency Modulated Continuous Wave

3 Beschreibung des Gesamtsystems

Bezeichnung	Datentyp	Beschreibung
ID	<i>int</i>	Eindeutiger Identifikationsnummer für die Lebensdauer eines Objektes
ΔX	<i>double</i>	X-Koordinate des Referenzpunktes des Fremdobjektes im Sensor-KOS
ΔY	<i>double</i>	Y-Koordinate des Referenzpunktes des Fremdobjektes im Sensor-KOS
$V_{X,rel}$	<i>double</i>	X-Komponente des Geschwindigkeitsvektors des Fremdobjektes im Sensor-KOS
$V_{Y,rel}$	<i>double</i>	Y-Komponente des Geschwindigkeitsvektors des Fremdobjektes im Sensor-KOS
$A_{X,rel}$	<i>double</i>	Beschleunigung des Fremdobjektes entlang der X-Achse des Sensor-KOSs

Tabelle 3.2: LRR-Objektspezifizierung

gleichzeitig verfolgt werden. Jedes LRR-Objekt besitzt eine Objektbeschreibung, deren wichtigste Elemente in Tabelle 3.2 beschrieben sind.

Da Fahrzeuge zu einem Großteil aus metallischen Materialien bestehen, stellen Sie für den Radar ein Objekt mit vielen Einzelreflexionen dar. Reflexionen mit sehr ähnlicher Distanz und Geschwindigkeit, die sich zusätzlich im gleichen Raumwinkelsegment befinden, werden von der internen Objektbildungslogik des LRR-Steuergeräts zu einem Objekt zusammengefasst. Da die Einzelheiten dieser internen Rohdatenverarbeitung des LRR-Steuergeräts nicht bekannt sind, wird in dieser Arbeit ein sehr vereinfachtes Sensormodell angenommen.

Wegen der Position des Sensors in der Mitte der vorderen Stoßstange des Fahrzeugs und dem relativ kleinen Öffnungswinkel von $\pm 8^\circ$ [62], kann davon ausgegangen werden, dass bewegte Radarobjekte dem Heck eines voraus fahrenden Fahrzeugs entsprechen. Zusätzlich wird die Annahme getroffen, dass der Referenzpunkt des Radarobjektes die Mitte des realen Fahrzeughecks beschreibt. In der Realität kann beobachtet werden, dass diese Annahme unter erschwerten Bedingungen nur bedingt als richtig bezeichnet werden kann. So schwankt die Position je nach Betrachtungswinkel und Umgebungsbedingungen über die gesamte Breite der Heckansicht.

In Abbildung 3.5) wird dieser Effekt deutlich. Dabei sind die Radarobjekte als Dreiecke visualisiert. Da über Ausdehnung und Höhe des Objektes über Boden keine Information vorhanden ist, muss die Mitte der Unterkante des Dreiecks als Referenzpunkt im Bild betrachtet werden. Die Dreiecke werden mit einer virtuellen Breite und Höhe von einem Meter ins Bild projiziert. Über die Lage der Bodenkante und die Längen der Kanten des abgebildeten Dreiecks kann eine Abschätzung der Entfernung durch den Betrachter vollzogen werden.

Bei der späteren Verwendung der Radardaten werden die beschriebenen Effekte berücksichtigt und die Fehler des einfachen Sensormodells als Unsicherheiten der Daten inter-

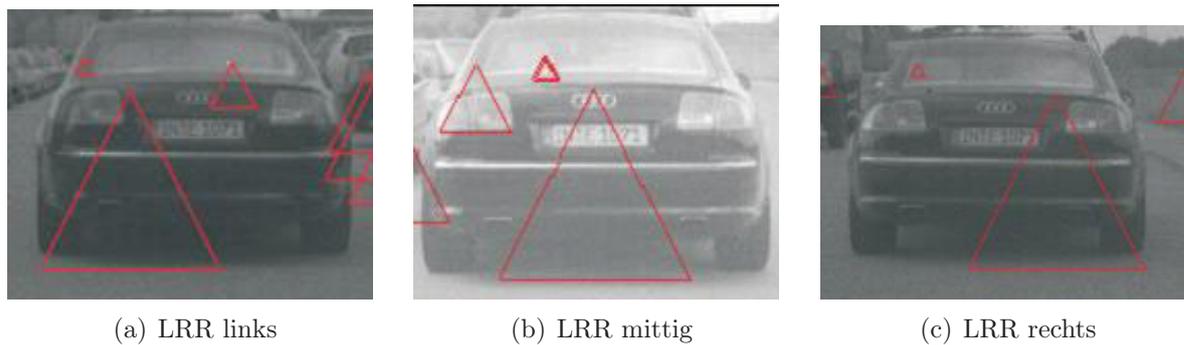


Bild 3.5: Unsicherheit des LRR-Referenzpunktes

pretiert (z. B. bei der Parametrisierung von Bildverarbeitungsoperatoren oder beim Objekttracking mittels Kalman-Filterung).

3.2.3 Laserscanner

Lasersensoren, ebenfalls aktive und entfernungsmessende Sensoren, sind im Automobilbereich noch nicht so etabliert wie der zuvor vorgestellte Radarsensor. Das Akronym „Laser“ steht für „Light Amplification by Stimulated Emission of Radiation“. Immer mehr Hersteller sind allerdings auf der Suche nach Möglichkeiten, diesen Sensortyp an Stelle des Radars ins Fahrzeug zu bekommen. Das liegt an den günstigeren Herstellungskosten und an den neuen Möglichkeiten, die dieser Sensor durch seine genaue Winkelauflösung in Kombination mit der direkten Entfernungs-messung mit sich bringt. Angelehnt an das Radar-Akronym wird heute für Lasersensoren auch der Begriff „Lidar“ (LIght Detection And Ranging) verwendet.

Lasersensoren senden monochromatische und scharf gebündelte Lichtstrahlen (kohärentes Licht) aus. Die heute meist verwendeten Diodenlaser arbeiten im Wellenlängenbereich zwischen $0.4 \mu\text{m}$ und $4.0 \mu\text{m}$, sprich zwischen 750 THz und 75 THz , worin auch der Bereich des sichtbaren Lichts (siehe Bild 3.2) enthalten ist. Bei Lasersensoren unterscheidet man zwischen einzeiligen und mehrzeiligen Sensoren, die wiederum entweder auf dem Multibeamprinzip oder einem Scanverfahren basieren. Da in dieser Arbeit ein einzeiliger Laserscanner eingesetzt wird, wird im Folgenden das Funktionsprinzip und Sensormodell für diesen speziellen Sensortyp beschrieben. Für weiterführende Literatur zum Thema Lasersensoren und deren Anwendung sei auf [86], [67], [49], [55], [80] und [28] verwiesen.

Funktionsprinzip

Ein Lasersensor besteht aus einem Sende- und einem Empfangsteil. Den Laserscanner unterscheidet vom Multibeamlidar, dass es nur einen einzelnen Laserstrahl gibt, der durch einen Ablenkungsmechanismus in verschiedene Raumrichtungen geschickt wird. Dies ge-

3 Beschreibung des Gesamtsystems

schiebt entweder über eine rotierende Sendeeinheit oder durch einen rotierenden Spiegel, der den Laserstrahl in die gewünschte Richtung ablenkt. Mit dem gleichen Mechanismus wird der Empfänger in die richtige Raumrichtung ausgerichtet und registriert das von einem Hindernis in diesem Raumwinkel reflektierte Laserlicht. Während der Rotation werden in kurzen Abständen Lichtimpulse ausgesendet und von der Empfangseinheit detektiert. Die diskreten Senderichtungen werden in der Lasertechnik häufig als „Kanal“ bezeichnet. Durch Laufzeitmessung kann somit die Umgebung, bzw. der Freiraum bis zum ersten Hindernis, diskret abgetastet werden. Die Laufzeitmessung kann wiederum mit Gleichung 3.1 beschrieben werden, da es sich auch um elektromagnetische Wellen handelt.

Der verwendete Sensor besitzt ebenso wie der Radarsensor eine Auswerteeinheit, die allerdings in diesem Fall lediglich die Steuerung des Senders, die Auswertung der Empfangsamplitude und die Laufzeitmessung durchführt. Somit steht für jeden Kanal jeweils eine Distanzmessung zur Verfügung. Dabei muss die auftretende Divergenz bzw. Defokussierung des Laserstrahls berücksichtigt werden: je weiter die Messdistanz, desto größer ist die angestrahlte Fläche. Dadurch sinkt die Aussagekraft über den wirklichen Aufenthaltsort eines reflektierenden Objektpunktes mit steigender Entfernung.

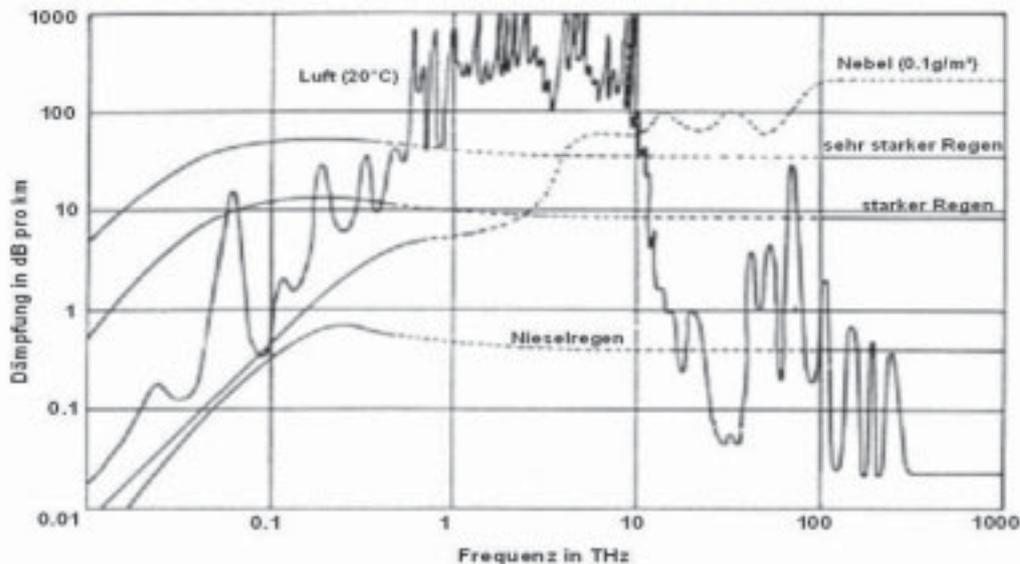


Bild 3.6: Atmosphärische Dämpfung elektromagnetischer Wellen(Quelle: [54])

Ein Nachteil der Lasersensoren gegenüber der Radartechnologie ist die größere atmosphärische Dämpfung. Bild 3.6 zeigt die Abhängigkeit der Dämpfung des ausgesendeten Signals von den Witterungsbedingungen und dem Frequenzbereich. Die hier verwendeten Lidarsensoren arbeiten mit Licht bei einer Wellenlänge von ca. 905nm. Das entspricht einer Frequenz von ca. 331 THz. Im Vergleich zur Radarfrequenz bei 77 GHz wird der Nachteil besonders bei schlechter Witterung (Regen, Nebel) deutlich.

Sensormodell

Die Strahlkonfiguration des Lidarsensoren ist in Tabelle 3.3 beschrieben³⁾.

	Sensor nach vorne
Öffnungswinkel (Θ_L)	$90^\circ (\pm 45^\circ)$
Anzahl Kanäle (N_K)	45
Kanalschrittswinkel (θ_K)	1.97°
hor. Defokussierung (δ_{Hor})	$1.5 \circ (\pm 0.75^\circ)$
vert. Defokussierung (δ_{Ver})	$3^\circ (\pm 1.5^\circ)$

Tabelle 3.3: Strahlkonfiguration des Lidarsensoren (Quelle: [79])

Der Kanalschrittswinkel θ_K berechnet sich dabei aus Θ_L , δ_{Hor} und N_K zu

$$\theta_K = \frac{\Theta_L - \delta_{\text{Hor}}}{N_K}$$

Diese Daten können nun für das Erzeugen des inversen Sensormodells genutzt werden, so dass die Entfernungsinformation, die aus einem Kanal ausgelesen werden kann, in einen 3D-Punkt rückgerechnet werden kann. Dabei ergibt sich für eine Umlaufrichtung der Sende- und Empfangseinheit in mathematisch positiver Richtung der Winkel $\vartheta_{N_K, \theta_K, n_K}$ des n_K -ten Kanals in der X-Y-Ebene des Sensor-KOSs zu

$$\vartheta_{N_K, \theta_K, n_K} = \left(-\frac{N_K}{2} + n_K\right) \cdot \theta_K. \quad (3.3)$$

Aus Gründen der Übersichtlichkeit wird im Folgenden $\vartheta_{N_K, \theta_K, n_K}$ mit ϑ_{n_K} abgekürzt. Mittels Gleichung 3.4 kann nun der Winkel ϑ_{n_K} , in Kombination mit der im n_K -ten Kanal gemessenen radialen Entfernung r_{n_K} , in einen 3D-Punkt \underline{P}_{n_K} im Lidarsensor-KOS umgerechnet werden.

$$\underline{P}_{n_K} = \begin{pmatrix} r_{n_K} \cdot \cos(\vartheta_{n_K}) \\ r_{n_K} \cdot \sin(\vartheta_{n_K}) \\ 0 \end{pmatrix} \quad (3.4)$$

Um die Defokussierung des Sensors zu berücksichtigen kann der Querschnitt durch einen Kanal als Kugeloberflächensegment aufgefasst werden. So steigt die von einem Kanal ausgeleuchtete Fläche quadratisch mit der radialen Entfernung ($\sim r^2$). Der horizontale Divergenzwinkel δ_{Hor} erzeugt im Abstand r_{n_K} eine horizontale Breite $b_{n_K, \text{Hor}}$ des Kanals von

$$b_{n_K, \text{Hor}} = 2 \cdot \tan(0.5 \cdot \delta_{\text{Hor}}) \cdot r_{n_K}. \quad (3.5)$$

³⁾ Abbildung 3.15 aus Kapitel 3.3 zeigt eine winkelgetreue Darstellung des Abtastbereichs.

3 Beschreibung des Gesamtsystems

Somit erzeugt beispielsweise ein Divergenzwinkel von $\delta_{\text{Hor}} = 1.5^\circ$ im Abstand von $r_{n_K} = 20 \text{ m}$ eine horizontale Messunsicherheit von

$$b_{n_K, \text{Hor}} = 0.524 \text{ m}.$$

Das gleiche gilt ebenfalls für die vertikale Divergenz des Laserstrahls.

3.2.4 Kamera

In dieser Arbeit werden Kameras zum Erkennen der Fahrspur sowie des voraus fahrenden Fahrzeugs verwendet. Abbildung 3.7 zeigt die dazu eingesetzte Kamera.

Funktionsprinzip

Eine Kamera unterscheidet sich von den in Kapitel 3.2.2 und 3.2.3 vorgestellten Sensoren dadurch, dass sie eine höhere Raumwinkelauflösung besitzt und die Winkelbestimmung horizontal und vertikal, also zweidimensional, durchgeführt werden kann. Ein weiterer wichtiger Unterschied besteht darin, dass die Kamera ein passiver Sensor ist, der auf das Umgebungslicht angewiesen ist. Abbildung 3.7 zeigt die verwendete Kamera.

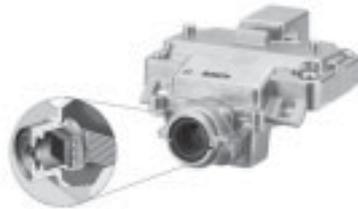


Bild 3.7: Kamera von Bosch (Quelle: [92])

Das Funktionsprinzip ähnelt stark dem des menschlichen Auges: Eine Linse mit der Brennweite f fokussiert das von einem Gegenstand G abgestrahlte Licht so, dass sich die Lichtstrahlen, die von einem Objektpunkt kommen, in der Bildebene wieder treffen. Dadurch erscheint ein am Hauptpunkt gespiegeltes Abbild B des Gegenstands auf der Bildebene hinter der Linse. Der Hauptpunkt ist zugleich der Ursprung des Kamera-KOSs. In Abbildung 3.8 ist der Strahlengang des ausgesendeten Lichts von einem Punkt des Gegenstands G hin dessen Abbild B zu sehen.

In der Bildebene befindet sich der Sensorchip. Auf diesem Chip sind kleine meist quadratische Fotodioden (Fotozellen) in einer zweidimensionalen Matrix angeordnet. Auf jede Diode fällt somit nur noch Licht aus einem bestimmten Raumwinkelsegment. Von der Fotodiode wird ein zur Lichtmenge proportionaler Strom erzeugt, der abgegriffen werden kann. Ein Nachgeschalteter A/D-Wandler ordnet dem Strom einen diskreten Wert in einem definierten Wertebereich zu. Je nach Bittiefe ergeben sich Werte zwischen 0 und 255 (bei 8 Bit) oder 0 und 4095 (bei 12 Bit). Dieser Wert, auch Grau- oder Helligkeitswert

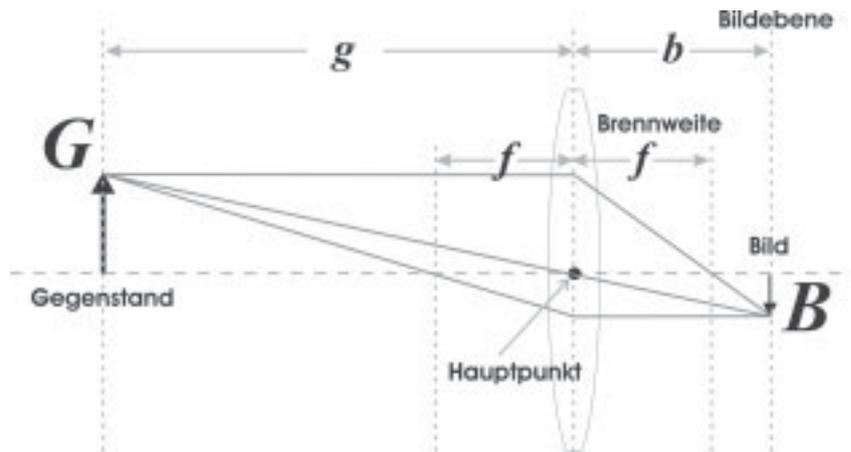


Bild 3.8: Vereinfachte Darstellung der Kameraprojektion

genannt, kann für jede Fozelle des Kamerachips ausgelesen werden. Im Zusammenhang mit der Weiterverarbeitung des Helligkeitswertes hat sich als Bezeichnung einer einzelnen Fozelle der Begriff „Pixel“ durchgesetzt. Dieses Akronym basiert auf dem Begriff „Picture Element“ (deutsch: „Bild Elemente“), was die Information einer Fotodiode auch darstellt. Eine gängige Dimension der Fozellenmatrix ist 640×480 ($d_u = 640$ Spalten, $d_v = 480$ Zeilen) und entspricht der Auflösung des VGA-Standards. Die Größe der Pixel kann mit heutigen Herstellungsverfahren sehr konstant gehalten werden.

Die verwendete Kamera benutzt einen *Active Pixel Sensor* (APS), der auf der CMOS⁴⁾-Technologie basiert. Diese ermöglicht im Vergleich zum *CCD*⁵⁾-Sensor auf MOSFET⁶⁾-Technologie

- einen verbrauchsärmeren Betrieb,
- das pixelweise oder segmentweise Auslesen der Helligkeitsinformation,
- die Integration des A/D-Wandlers auf dem Chip und
- eine höhere Dynamik des Helligkeitsbereichs.

Bei CMOS-Farbkameras kommt noch der Vorteil der besseren Farbtreue im Unterschied zu CCD-Kameras hinzu.

Sensormodell

Zur Modellierung des Kamerasystems, bestehend aus der Linse und dem CMOS-Chip, wird auf das idealisierte Lochkameramodell zurückgegriffen. Abbildung 3.9 veranschaulicht den Aufbau und die Funktionsweise einer Lochkamera.

⁴⁾ Complementary Metal Oxide Semiconductor

⁵⁾ Charged Coupled Device

⁶⁾ Metall-Oxid-Halbleiter-Feldeffekttransistor

3 Beschreibung des Gesamtsystems

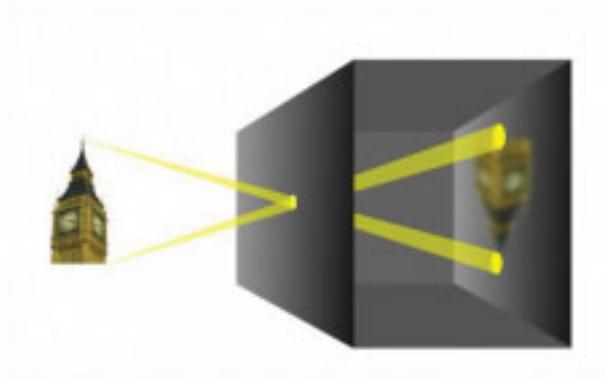


Bild 3.9: Lochkamera (Quelle: Wikipedia [97])

Hier kann Licht nur durch eine kleine Öffnung (Blende) in das Innere der Kamera gelangen. Dabei entsteht ebenfalls eine Punktspiegelung der Lichtstrahlen in diesem Öffnungspunkt, so dass wiederum ein gespiegeltes Abbild des Gegenstands auf der Bildebene entsteht. Der Abstand der Bildebene von der Lochblende wird für die Lochkamera als virtuelle Brennweite f_{LK} definiert.

Die Idealisierung des Modells besteht darin, dass Beugungs- und Streuungseffekte, die in der Realität an der Lochblende entstehen, vernachlässigt werden. So können die Projektionseigenschaften der idealisierten Lochkamera auf das mathematische Konstrukt der Zentralprojektion zurückgeführt werden. Die Theorie der Zentralprojektion wird von dem „Zeichner der Laute“ im abgebildeten Holzschnitt praktisch angewandt. Dabei werden die Lichtstrahlen, die von einem Objektpunkt ausgehen, durch ein gespanntes Seil imitiert, so dass der Zeichner den projizierten Punkt an der richtigen Stelle markieren kann.

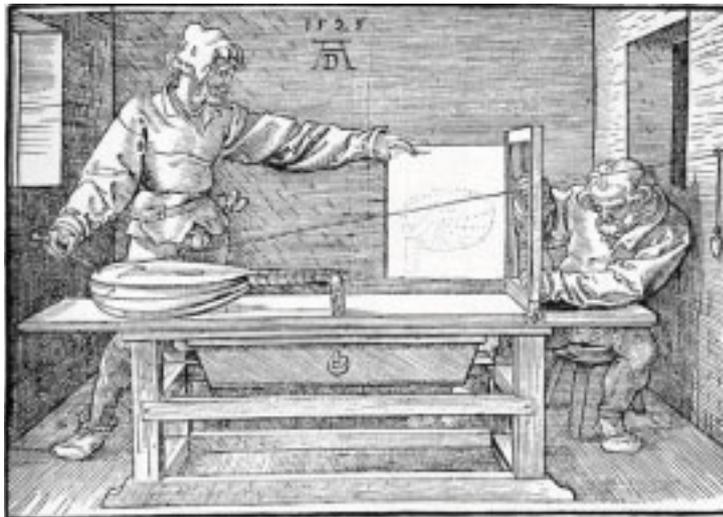


Bild 3.10: Holzschnitt: „Der Zeichner der Laute“ von Dürer

Die Zentralprojektion (auch als perspektivische Projektion bekannt) beschreibt die Projektion von 3D-Punkten in eine 2D-Bildebene, wobei die Projektionsstrahlen alle durch

3 Beschreibung des Gesamtsystems

$$u = \frac{x_B}{s_x} + u_{HP} \quad (3.8)$$

$$v = \frac{y_B}{s_y} + v_{HP} \quad (3.9)$$

Den Ursprung des Bildkoordinatensystems \underline{HP} , auch Hauptpunkt genannt, stellt dabei der Schnittpunkt der X_K -Achse mit der Bildebene dar und hat die Pixelkoordinaten $\underline{HP} = (u_{HP}, v_{HP})$ (siehe Abbildung 3.11). Die Parameter s_x und s_y bezeichnen die physikalische Breite bzw. Höhe eines Pixels in Meter. Es ist zu beachten, dass der Hauptpunkt in der Regel nicht mit dem Mittelpunkt des Sensorchips \underline{MP} übereinstimmt. Die Position des Hauptpunktes wird ebenfalls während der Kalibrierung bestimmt.

Führt man eine Normierung der Kamerakoordinaten nach deren X -Komponente durch, kann die Umrechnung von Kamerakoordinaten in Pixelkoordinaten in einem Schritt durchgeführt werden. Die Kombination der Gleichungen 3.6, 3.7, 3.8 und 3.9 sind als lineares Gleichungssystem in Matrix-Vektor-Notation in Formel 3.10 zu sehen.

$$\underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\text{Pixelpunkt } \underline{p}_P} = \underbrace{\begin{pmatrix} \frac{f_{LK}}{s_x} & 0 & u_{HP} \\ 0 & \frac{f_{LK}}{s_y} & v_{HP} \end{pmatrix}}_{\text{intr. Kameramatrix } \mathbf{K}_{\text{int}} = {}^P\mathbf{T}_B} \cdot \underbrace{\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix}}_{\text{Trafo-Matrix } {}^B\mathbf{T}_K} \cdot \underbrace{\begin{pmatrix} 1 \\ Y_K/X_K \\ Z_K/X_K \end{pmatrix}}_{\text{normierter Kamerapunkt } \underline{P}'_K} \quad (3.10)$$

Die Transformationsmatrix ${}^B\mathbf{T}_K$ ist nötig, um zunächst von Kamerakoordinaten in Standard-Bildkoordinaten überzugehen. Mit Hilfe der intrinsischen Kameramatrix \mathbf{K}_{int} erfolgt die Projektion, die Ursprungsverschiebung und die Pixelnormierung in einem Schritt, so dass als Ergebnis Punkte in Pixelkoordinaten entstehen.

Die Auflösung der hier verwendeten Kamera beträgt 640×480 Pixel und die Pixelausdehnung in Zeilen- sowie in Spaltenrichtung hat den Wert $s_x = s_y = 8 \cdot 10^{-6} \text{ m}$. Bei der Kalibrierung, die im Vorfeld der Verwendung der Kamera durchgeführt werden muss, ergeben sich Werte für die Brennweite des Lochkameramodells von $f_{LK} = 0.006656 \text{ m}$. Für den Hauptpunkt wurden die Pixelkoordinaten $\underline{HP} = (316.0, 236.0)^T$ ermittelt. Somit erhält man die intrinsische Kameramatrix

$$\mathbf{K}_{\text{int}} = \begin{pmatrix} 832 & 0 & 316.0 \\ 0 & 832 & 236.0 \end{pmatrix}. \quad (3.11)$$

Da bei der Projektion der Kamerapunkte in die Bildebene die Entfernungsinformation verloren geht, kann ein Bildpunkt in Pixelkoordinaten durch Invertieren der Kameramatrix lediglich in einen 3D-Vektor im Bild-KOS umgerechnet werden. Bei den durch die *Rückprojektion* entstehenden Vektoren wird häufig der Begriff *Sichtstrahlen* verwendet. Die eigentliche Position des projizierten Punktes kann nur über zusätzliche Zwangsbedingungen

(Annahmen) ermittelt werden. Eine Möglichkeit besteht darin, eine zweite Kamera für ein Stereosystem hinzuzufügen, so dass mittels Triangulierung die 3D-Information errechnet werden kann. Durch den starken Kostendruck in der Automobilindustrie, ist nicht absehbar, dass mehrere Kameras im gleichen Erfassungsbereich zum Einsatz kommen werden. Daher haben sich in dieser Domäne zwei andere Verfahren durchgesetzt:

Bewegungsstereo:

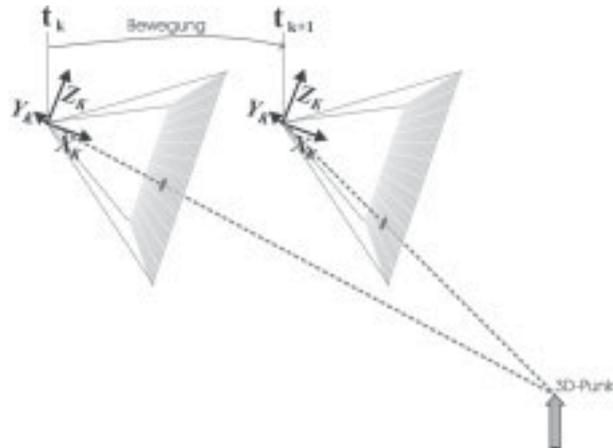


Bild 3.12: Bewegungsstereo

Bei bekannter Bewegung (Translation und Rotation) der Kamera zwischen zwei Zeitpunkten t_k und t_{k+1} kann, unter der Annahme einer statischen Umwelt, durch Triangulation auf die Tiefeninformation geschlossen werden. Der Unterschied zu normalen Stereosystemen besteht darin, dass die gleiche Kamera von zwei unterschiedlichen Positionen die Szene aufnimmt und die Information dann so interpretiert wird, als wären die Aufnahmen mit zwei separaten Kameras aufgenommen worden. Da Kameras in Fahrzeugen meist in Bewegung sind, lässt sich das Verfahren gut anwenden. Probleme entstehen dadurch, dass sich andere Objekte im Fahrzeugumfeld auch bewegen können und dadurch die Tiefeninformation verfälschen. Abbildung 3.12 zeigt schematisch die Vorgehensweise. Das Verfahren des Bewegungsstereos findet in dieser Arbeit nur implizit, durch die zeitliche Zuordnung von Messungen zu einem Modellpunkt innerhalb des 4D-Ansatzes (Kapitel 4.4.3), Verwendung.

Bodenebenenannahme:

Eine weitere Vorgehensweise zur Entfernungsschätzung mit einer Monokamera nutzt das Vorwissen über die Fahrzeugumgebung durch die so genannte „Bodenebenenannahme“. Dabei wird davon ausgegangen, dass der Boden im Nahbereich um das Fahrzeug einer Ebene entspricht. Mit dieser Ebene können die Sichtstrahlen unterhalb des Horizonts geschnitten werden, um auf die Entfernung eines Punktes zu schließen. Bei Weltpunkten, die tatsächlich in der Bodenebene liegen, wie Spurmarkierungen oder Bodenberührungspunkten von Objekten (z. B. Reifen bei Fahrzeugen), entspricht dieser Schnittpunkt der tatsächlichen Position des 3D-Punktes. Probleme bei dieser Methode entstehen durch unausgeglichene

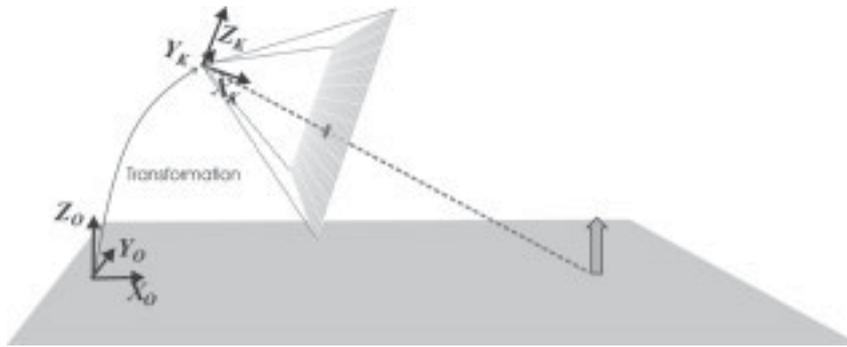


Bild 3.13: Bodenebenenannahme

Nickbewegungen des Fahrzeugaufbaus gegenüber der Bodenebene, fehlerhafte Merkmalsuche im Bild und vertikale Fahrbahnkrümmungen bzw. unebene Straßen. Abbildung 3.13 veranschaulicht die geometrische Modellannahme. Die Bodenebenenannahme spielt eine große Rolle bei den Verfahren zur Fahrzeugdetektion und der Fahrspurerkennung.

3.3 Verwendete Sensorkonfiguration

Nachdem nun die physikalischen Prinzipien der verschiedenartigen Sensoren erklärt sind, fasst dieses Unterkapitel die Eigenschaften der verwendeten Sensoren zusammen. Die technischen Daten dieser Sensoren sind in Tabelle 3.4 näher spezifiziert.

Beschreibung	Hersteller	Spezifikation			
		Zykluszeit	Horizontaler Öffnungswinkel	Vertikaler Öffnungswinkel	Sonstiges
LRR2	Bosch	ca. 130 ms	$\pm 8^\circ$		4 Antennen 77 GHz
Kamera	Bosch	40 ms	$\pm 21^\circ$	$\pm 16^\circ$	640 × 480 Pixel S/W CMOS-Chip Farbtiefe 12 Bit
Lidar Vorne	Spies	ca. 60 ms	$\pm 67.5^\circ$	$\pm 1^\circ$	45 Kanäle 1 Ebene $\pm 1^\circ$ Divergenz

Tabelle 3.4: Eigenschaften der Sensoren (Quellen: [79], [15] und [18])

In Abbildung 3.14 ist eine winkelgetreue Darstellung der Auflösungen und Sichtbereiche der in dieser Arbeit verwendeten Sensoren in der X-Z-Ebene (von rechts) zu sehen. Abbildung 3.15 zeigt die X-Y-Ebene (von oben). Dabei ist zu beachten, dass für den Videosensor aus Gründen der Darstellbarkeit, der Winkelbereich, den zehn benachbarte Pixel einnehmen würden, jeweils zu einem Gesamtbereich zusammengefasst ist.

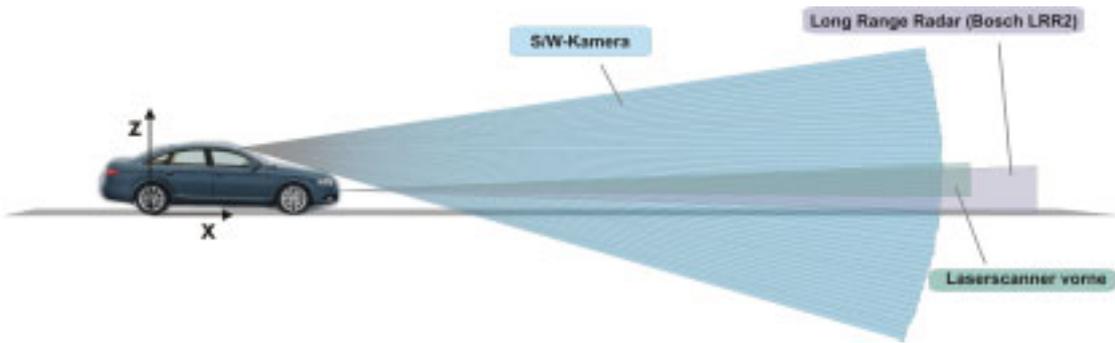


Bild 3.14: Sensorsichtbereiche seitlich

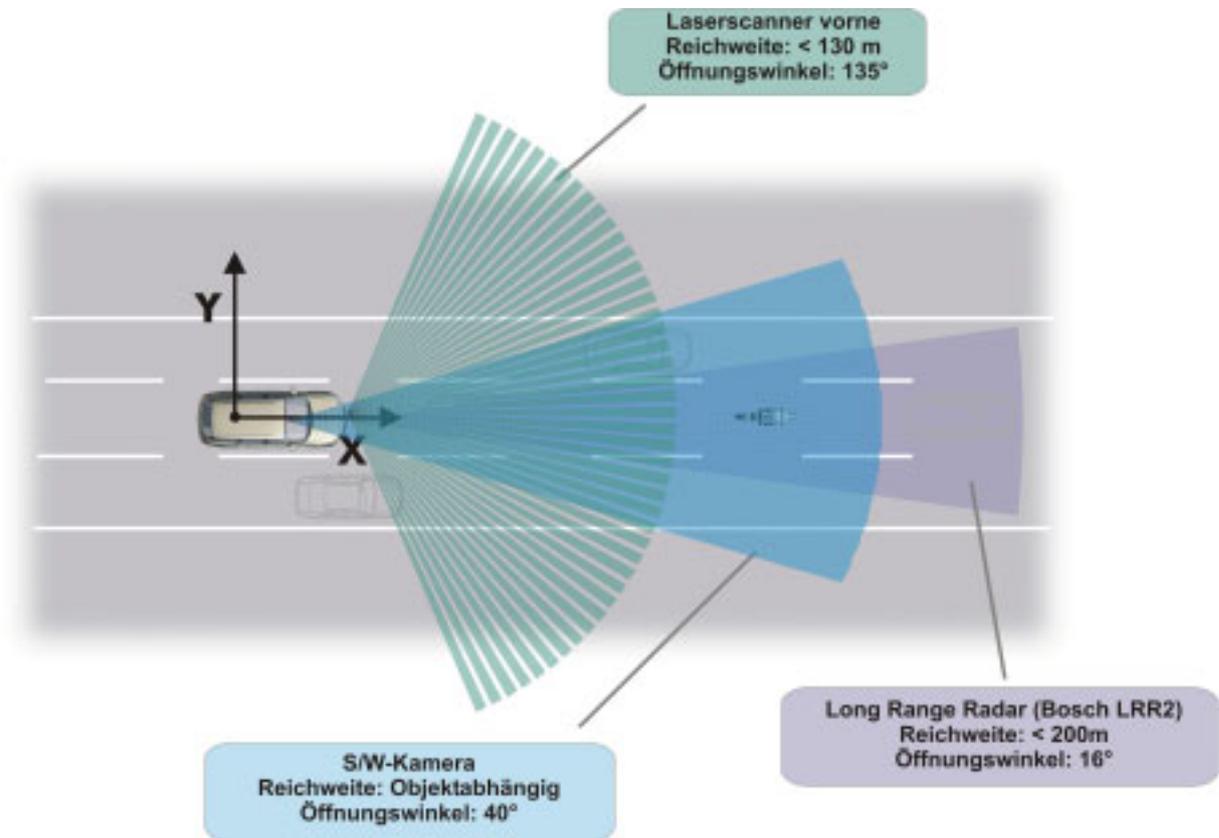


Bild 3.15: Sensorsichtbereiche von oben

3.4 Koordinatensysteme und Transformationen

Bei der Erkennung und Verfolgung von Objekten mit Sensoren müssen verschiedene Koordinatensysteme und die Transformation zwischen diesen definiert werden, um die Objekte mit ihrer Umwelt in Beziehung setzen zu können. Im Folgenden werden zunächst die wichtigsten Koordinatensysteme und das Prinzip der homogenen Koordinatentransformation beschrieben.

3.4.1 Beschreibung der verwendeten Koordinatensysteme

Bei den Koordinatensystemen muss grundsätzlich zwischen aufbaufesten, fahrzeuginternen und eigenbewegungskompensierten, umfeldbeschreibenden Koordinatensystemen unterschieden werden. Die Einteilung der verwendeten Koordinatensysteme wird in Tabelle 3.5 beschrieben. Alle hier beschriebenen Koordinatensysteme sind kartesische Rechtssysteme.

Fahrzeuginterne KOSe (aufbaufest)	Umfeldbeschreibende KOSe (eigenbewegungskompensiert)
Konstruktions-KOS (KOS_{Konst})	Umfeld-KOS (KOS_U)
Fahrzeugaufbau-KOS (KOS_A)	Fahrzeug-KOS (KOS_F)
Kamera-KOS (KOS_K)	Fahrspur-KOS (KOS_S)
Radar-KOS (KOS_R)	
Lidar-KOS (KOS_L)	

Tabelle 3.5: Einteilung der Koordinatensysteme

Die Abbildungen 3.16 und 3.17 zeigen die relativen Lagen der verschiedenen Koordinatensysteme zueinander. Die Koordinatensysteme werden dann in Kapitel 4.4 zur Beschreibung der Form- und Dynamikmodelle verwendet. Über die Subscriptbezeichnungen können die Koordinatenachsen (z. B. X_U) den passenden Koordinatensystemen (z. B. KOS_U) zugeordnet werden.

Nun werden zunächst die fahrzeuginternen Koordinatensysteme beschrieben und anschließend wird auf die umfeldbeschreibenden Koordinatensysteme eingegangen.

Konstruktionskoordinatensystem (KOS_{Konst}) Eines der fahrzeuginternen Koordinatensysteme ist das Konstruktionskoordinatensystem. Es ist in der ISO-Norm 4130 definiert und wird für Konstruktionsarbeiten am Fahrzeug vorgesehen. Die Achsen des Koordinatensystems werden durch die Schnittlinien der drei Bezugsebenen Null-X-Ebene, Null-Y-Ebene und Null-Z-Ebene aufgespannt. Bei der Fahrzeuglängsmittlebene bzw. Null-Y-Ebene handelt es sich gemäß ISO 4130 um die Symmetrieebene, die durch die Spurmitte von Vorder- und Hinterachse verläuft. Die Null-X-Ebene wird auch als Vertikalebene (bzw. senkrechte Ebene) und die Null-Z-Ebene als Horizontalebene (bzw. waagrechte Ebene) bezeichnet. Die Koordinatenachsen sind somit wie folgt definiert:

- X_{Konst} -Achse: Schnittgerade der Null-Y-Ebene mit der Null-Z-Ebene; Ausrichtung ist entgegen der Fahrtrichtung.
- Y_{Konst} -Achse: Steht senkrecht auf der Null-Y-Ebene und weist in Fahrtrichtung gesehen nach rechts.
- Z_{Konst} -Achse: zeigt nach oben und vervollständigt das orthogonale Rechtssystem.

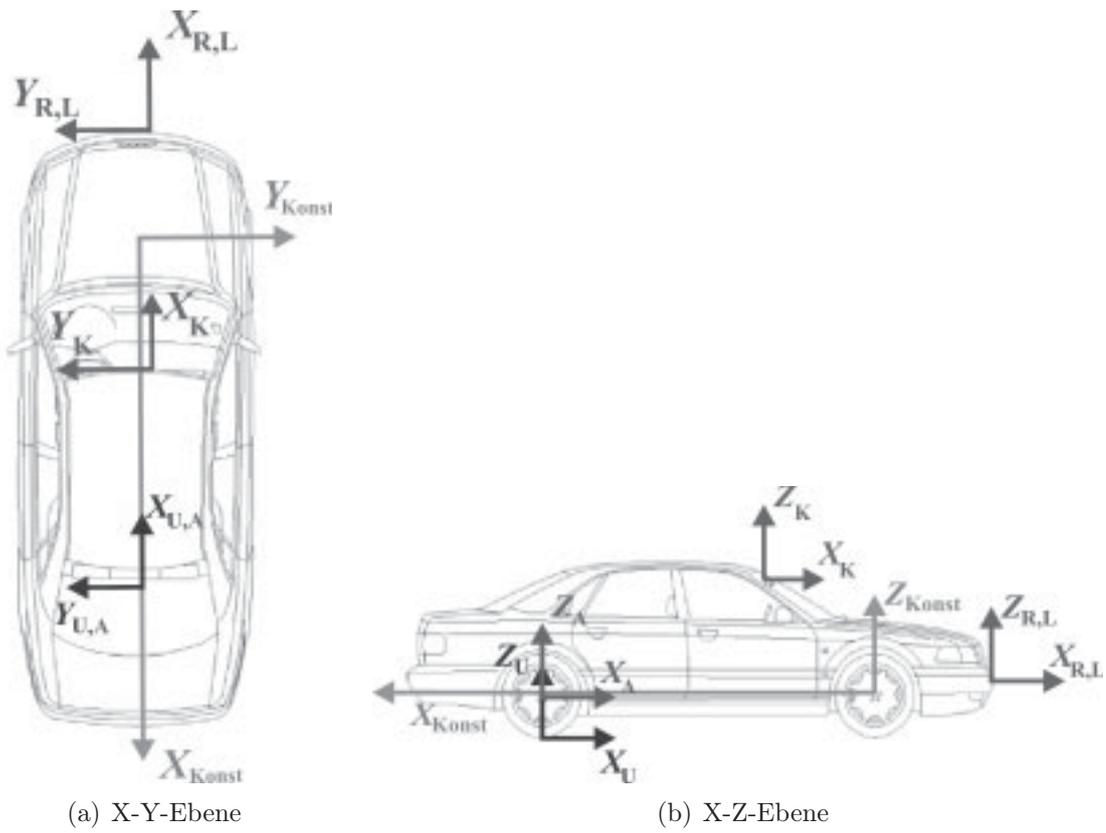


Bild 3.16: Schematische Darstellung fahrzeuginterner Koordinatensysteme

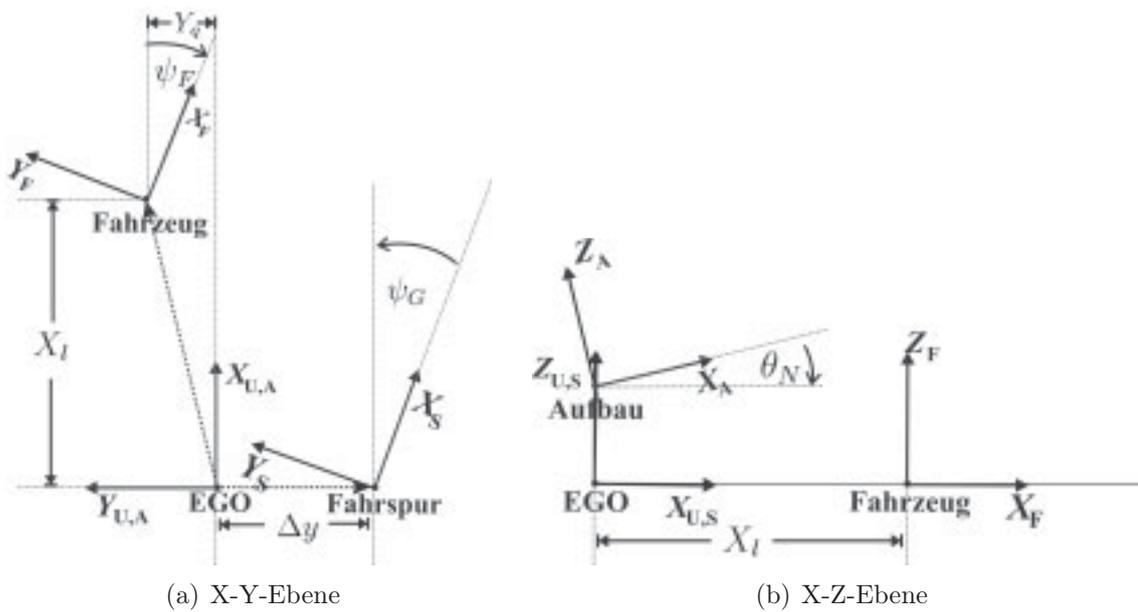


Bild 3.17: Schematische Darstellung umfeldbeschreibenden Koordinatensysteme

3 Beschreibung des Gesamtsystems

Die Lage der Null-X-Ebene und der Null-Z-Ebene und damit einhergehend die Lage des Koordinatensystemursprungs ist gemäß ISO 4130 frei wählbar. So liegt es nahe, die Einbaulage von fest an der Fahrzeugkarosserie angebrachten Sensoren, die für Versuchszwecke eingebaut sind, ebenfalls hinsichtlich dieses Koordinatensystems zu beschreiben.

Fahrzeugaufbau-KOS (KOS_A) Das Fahrzeugaufbaukoordinatensystem ist ein Zwischenkoordinatensystem, das zur Aufnahme der Bewegung der Fahrzeugkarosserie über der Bodenebene dient. Dieses Koordinatensystem ist für die Verwendung im Fahrdynamikbereich optimiert und in der ISO-Norm 8855 beschrieben. Der Ursprung dieses Koordinatensystems liegt üblicherweise im Schwerpunkt des Fahrzeugs. Die Achsen \underline{X}_A und \underline{Z}_A liegen in der Fahrzeuglängsmittalebene und \underline{Y}_A zeigt aus Fahrersicht nach links. In dieser Arbeit ist die relative Lage zum Konstruktionskoordinatensystem so definiert, dass die \underline{Z}_A -Achse bei einem voll getankten und ansonsten leerem Fahrzeug senkrecht zur Bodenebene nach oben zeigt. Dieser Zustand stellt die definierte Ruhelage des Fahrzeugs dar, so dass Abweichungen von der Ruhelage durch Änderungen im Beladungszustand bzw. durch die Fahrdynamik relativ zur Bodenebene beschrieben werden können. In dieser Arbeit wird der Ursprung \underline{O}_A von KOS_A , abweichend von der Norm, nicht in den Schwerpunkt, sondern auf die Mitte der Hinterachse des Fahrzeugs gelegt. Dadurch werden die kleinen Verschiebungen der Schwerpunktlage im Fahrzeugaufbau vernachlässigt. Diese Vereinfachung hat sich bei Fahrten unterhalb des fahrdynamischen Grenzbereichs bewährt.

Sensorkoordinatensysteme (KOS_K , KOS_R , KOS_L) Da ein Sensor zunächst alle Daten aus der eigenen Sicht bereitstellt, müssen die Messungen für die anschließende Verarbeitung richtig interpretiert werden. Dazu ist die Festlegung eines für jeden Sensortypen speziell definierten Koordinatensystems nötig. In diesem Projekt wurde für alle Sensoren die Vereinbarung getroffen, dass die X-Achse eines Sensor-KOSs in die Hauptsichtachse des Sensors zeigt:

Kamera:

Bei der Kamera ist die X_K -Achse gleich dem Lot des Brennpunktes der Linse auf den Sensorchip und zeigt durch die Linse aus der Kamera. Die Y_K -Achse ist so gewählt, dass sie entgegen der Bildzeilenrichtung des Chips, also in Sichtrichtung nach links zeigt. Dies führt dazu, dass die Z_K -Achse bei horizontal ausgerichteter Kamera nach oben zeigt. Dies ermöglicht ein leichtes Interpretieren der Daten für den Menschen in Bezug auf das Objektkoordinatensystem für Plausibilisierungen.

Radar:

Beim Radarsensor steht die X_R -Achse im Lot auf der Antennenebene. Da der Radarsensor nur Winkel in der Horizontalebene auflösen kann, wird die X_R -Achse stets als parallel zur Fahrbahnebene angenommen. Die Ausrichtung der Z_R -Achse wird wiederum nach oben definiert, woraus sich die Richtung der Y_R -Achse ableitet.

Laserscanner:

Beim Laserscanner zeigt die X_L -Achse senkrecht zur Rotationsachse der Sende- bzw. Empfangseinheit. Zusätzlich ist die X_L - Z_L -Ebene so definiert, dass sie entlang der Mitte des mittleren Laserkanals liegt. Da für eine erfolgreiche Messung der Reflektionspunkt im Sendebereich des Sensors liegen muss, ist der Ursprung der X_L -Achse im Drehzentrum des Senders gewählt. Die Z_L -Achse zeigt entlang der Drehachse des Sensors und erfüllt die Bedingung der „Rechten-Hand-Regel“. Die Y_L -Achse vervollständigt das Rechtssystem.

Umfeld-KOS (KOS_U) Die Informationen über andere Objekte im Fahrzeugumfeld, die benötigt werden, um eigene Aktionen an die aktuelle Situation anzupassen, können folgendermaßen zusammengefasst werden:

1. Die relative Position und Ausrichtung
2. Die absolute Dynamik über Grund
3. Die aktuellen Formparameter

Um die ersten beiden Punkte interpretierbar beschreiben zu können, ist es nötig, das Umfeld-KOS einzuführen. Dessen X_U - Y_U -Ebene entspricht einer gemittelten Ebene durch die Bodenfläche in der unmittelbaren Umgebung um das EGO-Fahrzeug. Das Umfeld-KOS ist also die Projektion des Fahrzeugaufbau-KOSs auf eine perfekt planare Fahrbahn. Dabei werden die Abweichungen der Nick-, Wank- und Hubbewegung des EGO-Aufbaus von der definierten Ruheposition durch die Transformation ${}^U\mathbf{T}_A$ aufgenommen. Dadurch werden die Bewegungen des EGO-Aufbaus von der Dynamik der Objekte im Fahrzeugumfeld entkoppelt, wodurch erst eine sinnvolle Beschreibung der Umfeldobjekte ermöglicht wird. Aus Gründen der Fahrdynamik bietet es sich an, den Ursprung dieses KOSs in den auf die Bodenebene projizierten Mittelpunkt der Hinterachse zu legen.

Fremdobjekt-KOSe (KOS_F und KOS_S) Mit Hilfe der bisher beschriebenen Koordinatensysteme lassen sich Umfeldobjekte aus Sicht des EGO-Fahrzeugs beschreiben. Will man nun jedoch die Position eines Fahrzeugs innerhalb einer Fahrspur wissen, oder die Lage zweier Fahrzeuge zueinander, benötigt man ein zusätzliches Koordinatensystem für jedes Fremdobjekt. Die Abbildung 3.17 zeigt schematisch das typische Zusammenspiel der verschiedenen Koordinatensysteme.

Der Ursprung \underline{O}_S des Fahrspur-KOSs liegt in der Mitte zwischen der linken und rechten Fahrspurbegrenzung. Die \underline{X}_S -Achse zeigt entlang der Tangente an die Skelettlinie der Fahrspur im Ursprungspunkt. Die X_S - Y_S -Ebene ist zudem parallel zur X_U - Y_U -Ebene. Der Ursprung des Fahrspurmodells wird mit der Bewegung des EGO-Fahrzeugs so mitbewegt, dass die X-Komponente der Position des Umfeld-KOS-Ursprungs im Fahrspur-KOS stets Null ist.

Der Ursprung \underline{O}_F des Fahrzeug-KOSs ist auf die geometrische Mitte der Bodenfläche des einhüllenden Quaders um das Fremdfahrzeug festgelegt. Wie beim EGO-Fahrzeug zeigt auch hier die \underline{X}_F -Achse entlang der Fahrzeuglängsachse und die \underline{Z}_F -Achse ist entlang der

3 Beschreibung des Gesamtsystems

Normalen der Bodenebene am Punkt des Fahrzeugobjektes ausgerichtet. Im einfachsten Fall, unter der Annahme eines planaren Fahrbahnverlaufs, liegt die X_F - Y_F -Ebene ebenfalls parallel zur X_U - Y_U -Ebene.

3.4.2 Homogene Koordinatentransformation

Um die verschiedenen Koordinatensysteme ineinander überzuführen, bedarf es Koordinatentransformationen. Dabei hat sich die Darstellung von Punkten als so genannte „Homogene Koordinaten“ bewährt. Diese unterscheidet sich von der normalen Darstellung in Vektorform lediglich durch das Hinzufügen einer zusätzlichen Zeile, welche den Wert 1 enthält. Für den dreidimensionalen Fall also:

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} \rightarrow \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ 1 \end{pmatrix}$$

Gleichzeitig müssen die Matrizen um eine Zeile und eine Spalte erweitert werden, um für konsistente Matrix-Vektor-Operationen zu sorgen. Mittels dieser Schreibweise lässt sich die affine Transformation eines Punktes \underline{P} vom KOS_A nach KOS_B durch Gleichung 3.12 beschreiben.

$$\underline{P}_B = {}^B \mathbf{T}_A \cdot \underline{P}_A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \underline{P}_A \quad (3.12)$$

Affine Transformationen beinhalten folgende Transformationstypen: Translation, Rotation, Skalierung, Scherung.

Soll durch die Transformation lediglich die Translation und Rotation zweier Koordinatensysteme zueinander beschrieben werden, kann die allgemeine Form der affinen Transformation zu

$$\underline{P}_B = {}^B \mathbf{T}_A \cdot \underline{P}_A = \begin{pmatrix} r_{00} & r_{01} & r_{02} & t_{03} \\ r_{10} & r_{11} & r_{12} & t_{13} \\ r_{20} & r_{21} & r_{22} & t_{23} \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \underline{P}_A = \begin{pmatrix} \mathbf{R} & \underline{t} \\ \underline{0}^T & 1 \end{pmatrix} \cdot \underline{P}_A \quad (3.13)$$

vereinfacht werden. Die Matrix \mathbf{R} beschreibt die Rotation zwischen den beiden Koordinatensystemen, während der Vektor \underline{t} durch die homogene Vektorform bei der Multiplikation als Verschiebungsvektor wirkt. Die letzte Zeile der homogenen Transformationsmatrix ${}^B \mathbf{T}_A$ bewirkt lediglich, dass als Ergebnis wiederum ein homogener Vektor entsteht.

Rotation Der rotatorische Anteil der Transformation kann ebenfalls durch die drei Winkel ψ , θ und φ und die Achsen \underline{Z}_A , \underline{Y}_A und \underline{X}_A beschrieben werden. Wird die Rotation nur um eine dieser Achsen ausgeführt so ergibt sich je nach Achse eine der Rotationsmatrizen in Gleichung 3.16.

$$\mathbf{R}_X(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \quad (3.14)$$

$$\mathbf{R}_Y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad (3.15)$$

$$\mathbf{R}_Z(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.16)$$

Mathematisch lässt sich die kombinierte Drehung um mehrere Achsen durch die Multiplikation der Matrizen \mathbf{R}_X , \mathbf{R}_Y und \mathbf{R}_Z darstellen. Da Matrixmultiplikationen nicht kommutativ sind, spielt die Multiplikationsreihenfolge eine Rolle bei der letztendlich erzeugten Rotationsmatrix. Deshalb muss definiert werden, wie die Multiplikationsreihenfolge ist, damit das Ergebnis der Matrix nach mehreren Transformationen wieder mit den richtigen Winkeln um die jeweiligen Achsen interpretiert werden kann. Bei den Arbeiten im Bereich der Fahrzeugtechnik wird der Aufbau der Rotationsmatrix an die Definition in DIN 70000 angelehnt. Dadurch ergibt sich folgende Form für die Rotationsmatrix \mathbf{R} .

$$\mathbf{R} = \begin{pmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi & \sin\theta \\ \cos\varphi\sin\psi + \sin\varphi\sin\theta\cos\psi & \cos\varphi\cos\psi - \sin\varphi\sin\theta\sin\psi & -\sin\varphi\cos\theta \\ \sin\varphi\sin\psi - \cos\varphi\sin\theta\cos\psi & \sin\varphi\cos\psi + \cos\varphi\sin\theta\sin\psi & \cos\varphi\cos\theta \end{pmatrix}$$

mit $\mathbf{R} = \mathbf{R}_X(\varphi) \cdot \mathbf{R}_Y(\theta) \cdot \mathbf{R}_Z(\psi)$

Durch die Festlegung dieser Definition, kann über die Gleichungen 3.17 aus den Einträgen der Matrix \mathbf{R} eindeutig auf die zugrunde liegenden Drehwinkel geschlossen werden, wobei $r_{i,j}$ der Eintrag in der i-ten Zeile und j-ten Spalte der Matrix \mathbf{R} ist.

$$\begin{aligned} \theta &= \operatorname{atan2} \left(-r_{2,0}, \sqrt{(r_{0,0})^2 + (r_{1,0})^2} \right) \\ \psi &= \operatorname{atan2} \left(\frac{r_{1,0}}{\cos(\theta)}, \frac{r_{0,0}}{\cos(\theta)} \right) \\ \varphi &= \operatorname{atan2} \left(\frac{r_{2,1}}{\cos(\theta)}, \frac{r_{2,2}}{\cos(\theta)} \right) \end{aligned} \quad (3.17)$$

3.5 Softwareentwicklungsumgebung ADTF

Die Bezeichnung ADTF steht für „Automotive Data and Time Triggered Framework“ . ADTF wird von der Audi Electronics Venture (AEV) entwickelt und bietet dem Entwickler zahlreiche nützliche Funktionen bei der Entwicklung von Softwaresystemen im Automobilbereich. ADTF basiert überwiegend auf den Prinzipien des *Component Object Model* (COM) von Microsoft®. Dabei wird eine Komponente als Abstraktion einer Klasse verstanden, welche den Zugriff auf die Funktionalität einer Klasse durch eine Standardschnittstelle zur Verfügung stellt. Im Folgenden werden diese Komponenten als Filter bezeichnet. Der Hauptvorteil von ADTF besteht darin, dass über diese Basisschnittstelle, unabhängig voneinander entwickelte Filter, miteinander kommunizieren können. Zusätzlich zu den durch die COM-Schnittstelle zur Verfügung gestellten Funktionalitäten eines Filters, ist im ADTF auch ein Kommunikationsmechanismus definiert, der es erlaubt, einen Filter mit so genannten „Pins“ zu versehen. Kompatible Ein- und Ausgangspins verschiedener Filter können auf der Konfigurations-GUI mittels Drag-And-Drop-Mechanismen miteinander verbunden werden, so dass die Filter definierte Datenpakete, so genannte „Mediasamples“ , senden und empfangen können. Ein Mediasample ist dadurch ausgezeichnet, dass es einen void-Pointer auf eine Speicheradresse und einen zugehörigen Zeitstempel transportiert. Dadurch können beliebige Daten zusammen mit dem Aufnahmezeitpunkt zwischen den Filtern ausgetauscht werden. Eine weitere Eigenschaft von ADTF besteht darin, dass die Berechnungen der einzelnen Filter durch bestimmte Ereignisse angestoßen werden können. Zu diesen Ereignissen zählt das Anlegen neuer Daten an einem Pin oder das Ablaufen eines Timers. Deshalb bezeichnet sich ADTF auch als Daten- bzw. Zeitgetriggert. Darüber hinaus kann ein Filter bestimmte „Properties“ besitzen. Dies sind Werte die im Rahmen des ADTF-Frameworks durch den Nutzer gesetzt werden können und zur Parametrisierung eines Filters dienen. Abschließend sei noch erwähnt, dass ADTF als synchron ablaufendes System entwickelt ist. Zum Vermeiden von Dead-Locks ist der Programmierer verantwortlich. Weitere Informationen zu ADTF können im ADTF-Entwicklerhandbuch [3] nachgelesen werden.

Durch den intensiven Gebrauch von ADTF in den Entwicklungsabteilungen bei AUDI sind viele Standard-Filter entstanden, auf die man als Entwickler zurückgreifen kann. Die Wiederverwendbarkeit von vorhandenen Filtern wirkt sich sehr positiv auf die Entwicklungszeit und auf die Robustheit beim Entwickeln des eigenen Filters aus und wurde auch für die in dieser Arbeit entstandenen Algorithmen verwendet.

In Tabelle 3.6 werden kurz die Filter beschrieben, die im Verlauf dieser Arbeit genutzt wurden. Alle Standardfilter, die einer ADTF-Installation beiliegen, können im ADTF-Referenzhandbuch [4] nachgelesen werden.

ADTF-Filter	Beschreibung
<i>Harddisk-Recorder</i>	„Der Harddisk-Recorder dient zur Aufzeichnung der Messdaten auf Festplatte.“ ([4], S. 4) Die Daten verschiedener Quellen werden in der Reihenfolge ihres Eintreffens an der Rechnerschnittstelle mit dem zugehörigen Messzeitstempel aufgezeichnet. Die entstandene Datei wird als DAT-File bezeichnet und hat die Endung <code><*.dat></code> .
<i>Harddisk-Player</i>	„Der Harddisk-Player spielt die mit dem Harddisk-Recorder aufgezeichneten Messdaten-Files ab und speist diese Daten ins Framework ein.“ ([4], S. 6) Dabei werden die Daten in der gleichen Reihenfolge wie beim Aufzeichnen ausgelesen. Das hat zur Folge, dass die Anwendung (ein ADTF-Filter) nicht unterscheiden kann, ob es sich um eine Aufzeichnung handelt oder um einen Livebetrieb.
<i>BoschFrameGrabberDevice</i>	Dieser Filter stellt die Funktionalität zum Einlesen der Videobilder von der Boschkamera zur Verfügung.
<i>VectorCanDevice</i>	„Der VectorCanDevice-Filter dient zum Einlesen von CAN-Roh-Daten über die jeweilige Vector Hardware ins ADTF.“ ([4], S. 7
<i>CanConfigCodec</i>	„Das CanConfigCodec-Filter decodiert und encodiert CAN-Rohdaten.“ ([4], S. 9) Dazu liest es Daten aus dem VectorCanDevice aus und rechnet diese in die dazugehörige SI-Einheit um, welche im dbc-File angegeben ist. Der Filter kann über eine XML-Datei konfiguriert werden. Entsprechend dieser Konfiguration werden Datenstrukturen in CAN-Messages umgesetzt und umgekehrt.
<i>VideoDisplay</i>	„Das VideoDisplay-Filter dient zur Anzeige von Videobildern“ ([4], S. 23). Es stellt zusätzlich alle Funktionalitäten zur Verfügung, um eigene Markierungen über das Videobild zu legen (Overlays).
<i>SampleStorage</i>	„Das SampleStorage Filter dient zur Pufferung der Daten, um während des Betriebs des Frameworks auf eine Daten-Historie zugreifen zu können. Zusätzlich bietet es die Möglichkeit durch so genannte <i>StorageTrigger</i> andere Filter/Applikationen zu benachrichtigen, wenn die jeweils von ihnen benötigten Daten neu im Storage eingetroffen sind.“ ([4], S. 4)

Tabelle 3.6: Verwendete ADTF-Filter

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

Bei der täglichen Fahrt zur Arbeit ist es dem Fahrer häufig gar nicht bewusst, wie komplex die Aufgabe ist, die er dabei durchführt. Es müssen Informationen auf unterschiedlichen Ebenen verarbeitet, Entscheidungen getroffen und Aktionen ausgeführt werden. Das beginnt bei der Informationsaufnahme über die verschiedenen Sinne. Beim Führen eines Fahrzeugs spielen die visuelle Wahrnehmung und der Gleichgewichtssinn die größte Rolle. Aber auch die auditive Wahrnehmung und der Tastsinn helfen dem Fahrer bei der Bewältigung seiner Aufgabe. Die Komplexität der Verarbeitung der Gesamtinformation lässt sich erahnen, wenn man bedenkt, dass der menschliche Lernprozess 18 Jahre dauert, bis er ein Fahrzeug sicher im Straßenverkehr bewegen kann.



Bild 4.1: Vergrößerte Heckansicht

Die verwendete Kamera liefert Bilder mit einer Grauwerttiefe von 12 Bit pro Pixel. Die Pixelinformation wird in Blöcken zu je zwei Byte im Rechner zur Verfügung gestellt. Somit ergibt sich durch die Auflösung von 640×480 Pixel bei einer Bildrate von 25 Frames pro Sekunde eine Datenrate von 15 360 000 Bytes pro Sekunde. Aus dieser großen Datenmenge lässt sich der Informationsgehalt, welche ein Videobild zur Verfügung stellt, erahnen. Leider ist diese Information zunächst in den Helligkeitswerten der Pixel verschlüsselt. Dies soll mittels Abbildung 4.1 verdeutlicht werden. Das vergrößerte Fahrzeug ist ca. 55 Meter entfernt. Dennoch werden einzelne Merkmale des Fahrzeugs, wie die Heckleuchten

und das Nummernschild, nur noch durch wenige Pixel dargestellt. Im Originalbild mit Bezug zur Szene hat der Betrachter zunächst den Eindruck noch jedes Detail erkennen zu können. Dabei kommt die große Menge an Vorwissen des Menschen zum Einsatz.

Ziel der Bildverarbeitung ist es diese Helligkeitswerte mittels Vorwissen und Modellannahmen, ähnlich wie das visuelle Zentrum im menschlichen Gehirn, in die gesuchte Information zu transformieren. Im Unterschied zu der massiven parallelen Informationsverarbeitung im Gehirn arbeiten heutige Computer noch sequentiell. Somit stoßen Standard-PCs selbst mit ihren hohen Taktraten bei komplexen Verarbeitungsschritten schnell an die Grenzen der Echtzeitfähigkeit. Deshalb spielt ein optimiertes Design und die effiziente Implementierung beim Entwickeln von Bildverarbeitungsoperatoren eine große Rolle. Die im Laufe dieser Arbeit entstandenen Bildverarbeitungs-Operatoren sind in C++ implementiert und durch den Einsatz von Pointerarithmetik und dem Vermeiden unnötiger Multiplikationen auf kurze Rechenzeiten optimiert.

Das Unterkapitel 4.1 beschreibt allgemein die Anwendung von Bildverarbeitungsoperatoren am Beispiel eines Kantenoperators. In 4.1.2 wird die Hauptachsentransformation (PCA) und deren Anwendung im Bereich der Bildverarbeitung erklärt. Anschließend werden in Kapitel 4.2 und 4.3 die speziellen Bildverarbeitungsverfahren beschrieben, die zur Erkennung von Fahrzeugheckansichten in dieser Arbeit zum Einsatz kommen. Anschließend werden in Kapitel 4.4 die Modelle der Objektdynamik und -gestalt zur Objektverfolgung aufgezeigt und die Grundzüge des 4D-Ansatzes beschrieben.

4.1 Grundlagen der Bildverarbeitung

Zunächst soll die verwendete Notation erläutert werden: Das Bild wird als zweidimensionale Matrix \mathbf{B} mit der Dimension $U \times V$ aufgefasst und beinhaltet die Menge \mathcal{B} aller Bildpunkte \underline{p}_p im Pixelkoordinatensystem. Aus Gründen der Übersichtlichkeit wird ab hier in diesem Kapitel bei Bildpunkten auf die Angabe des Referenzkoordinatensystems verzichtet ($\underline{p}_p \rightarrow \underline{p}$). $i[u, v]$ bezeichnet den Grauwert bzw. die Intensität des Pixels mit den Koordinaten (u, v) .

Um die gewünschte Information aus dem Bild zu extrahieren, kommen Bildverarbeitungsoperatoren zum Einsatz. Die einzelnen in den Operatoren ausgeführten Operationen können den Gruppen Punkt-, lokale und globale Operationen zugeordnet werden [82]. Unter Punktoperationen versteht man dabei Berechnungen, die jedes Pixel einzeln betrachten und modifizieren. Dazu gehört beispielsweise die Binärisierung oder eine Modifikation der Pixelgrauwerte (Normalisierung, Logarithmisierung). Das Gegenteil von Punktoperationen bilden die globalen Operationen, bei denen ein ganzes Bild oder ein Bildausschnitt komplett transformiert wird. Zu dieser Gruppe gehören beispielsweise die 2D-Wavelet- oder auch die 2D-Fouriertransformation.

Die meisten in dieser Arbeit verwendeten Operatoren führen allerdings lokale Operationen aus. Diese gewinnen Informationen über eine gewählte Bildposition, indem sie einen bestimmten Punkt in Relation zu seiner Nachbarschaft betrachten. Dazu wird meist das

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

Mittel der linearen diskreten Faltungsoperation herangezogen. Eine zweidimensionale diskrete Faltungsoperation ist dabei in Gleichung 4.1 definiert.

$$i_{\mathbf{F}}[u, v] = \mathbf{F} * i[u, v] = \sum_{r=-\infty}^{\infty} \sum_{c=-\infty}^{\infty} (f[c, r] \cdot i[u - c, v - r]) \quad (4.1)$$

Dabei ist $i_{\mathbf{F}}[u, v]$ das Ergebnis der diskreten Faltung des Originalbildes um den Punkt mit den Koordinaten (u, v) mit dem Faltungskern \mathbf{F} . $f[c, r]$ ist der Eintrag der Faltungsmatrix \mathbf{F} in der c -ten Spalte und r -ten Zeile.

Ziel der Faltungsoperation ist es, die Position eines erwarteten Merkmals zu finden. Dabei wird der Faltungskern so ausgelegt, dass er mit dem Bildbereich um das erwartete Merkmal ein maximales Ergebnis erzielt. Das entspricht dem Prinzip des signalangepassten Filters („Matched Filter“) und kann auch als Korrelationsanalyse interpretiert werden. Führt man nun die Faltungsoperation an allen Stellen aus, an denen das erwartete Merkmal vorhanden sein könnte, wird man ein Maximum an der Position des wirklich vorhandenen Merkmals finden.

Wegen dem in der Realität vorhandenen Rauschanteil des informationstragenden Signals, ist es in der Praxis häufig nicht eindeutig ob ein Maximum in der Filterantwort der Position des gesuchten Merkmals entspricht. Auch können Merkmale mehrfach auftreten oder überhaupt nicht vorhanden sein. Deshalb spielt bei dieser Art von Merkmalsextraktion das richtige Einstellen von Schwellwerten eine große Rolle, um die Fehlerraten bei der Detektion möglichst gering zu halten. Auch die richtige Parametrierung der Faltungsmaske \mathbf{F} an sich ändernde Bedingungen ist eine große Herausforderung.

Da die Dimension eines gesuchten Merkmals im Bild auf R Zeilen und C Spalten beschränkt ist, können die Elemente des Faltungskerns, die außerhalb dieser Dimension liegen zu Null gesetzt werden. Dadurch ergeben sich bei der Faltungsoperation nur noch maximal $R \times C$ Summanden mit einem Wert ungleich Null. Somit ändert sich die allgemeine Faltungsformel zu der implementierbaren Rechenvorschrift 4.2.

$$i_{\mathbf{F}}[u, v] = \mathbf{F} * i[u, v] = \sum_{r=-\frac{R-1}{2}}^{\frac{R-1}{2}} \sum_{c=-\frac{C-1}{2}}^{\frac{C-1}{2}} (f[c, r] \cdot i[u - c, v - r]) \quad (4.2)$$

mit $R, C \in \{1, 3, 5, \dots\}$

R und C geben die Anzahl der Zeilen bzw. Spalten der Faltungsmaske an. Diese müssen ungerade sein, damit eine symmetrische Maske um den Maskenmittelpunkt entsteht.

4.1.1 Kantendetektionsoperatoren

Ein Standardmerkmal, das bei nahezu allen Bildverarbeitungsaufgaben verwendet wird, sind Objektkonturen und Helligkeitsübergänge (bei Fahrzeugen z.B. Außenkanten und Schattenübergänge). Beide Merkmale bilden sich im Graustufenbild als hohe Helligkeitssprünge in benachbarten Pixeln ab. Mathematisch betrachtet bedeutet das einen hohen

Wert in der ersten Ortsableitung in Richtung des Helligkeitssprungs bzw. senkrecht zur Übergangskante (siehe Abbildung 4.2).

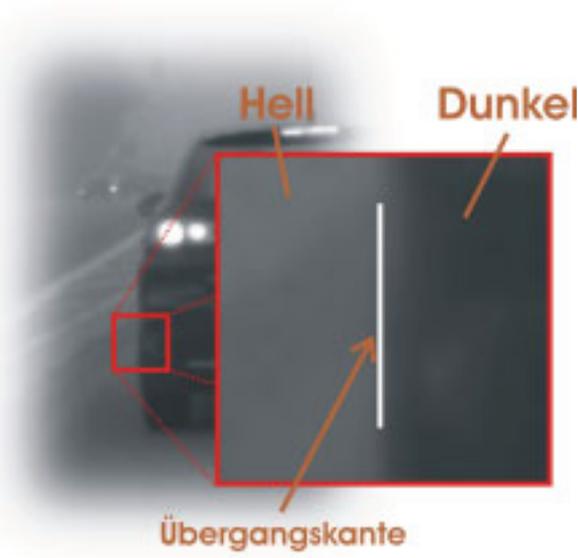


Bild 4.2: Beispiel einer Helligkeitskante

Allgemein lässt sich die Ableitung $f'(n)$ einer Funktion $f(n)$ im Diskreten durch

$$f'(n) = f(n) - f(n - 1) \quad (4.3)$$

formulieren. Als Bildverarbeitungsoperation ausgedrückt ergeben sich hieraus für die Ableitung in Zeilen- bzw. Spaltenrichtung die Rechenvorschriften 4.4 bzw. 4.5.

$$i'_{\text{Hor}}[u, v] = i[u, v] - i[u - 1, v] \quad (4.4)$$

$$i'_{\text{Ver}}[u, v] = i[u, v] - i[u, v - 1] \quad (4.5)$$

Diese Differenzbildung lässt sich auch als Faltung des Originalbildes mit den Faltungsmasken \mathbf{F}_{Hor} und \mathbf{F}_{Ver} darstellen.

$$\mathbf{F}_{\text{Hor}} = \begin{pmatrix} -1 & +1 \end{pmatrix} \quad (4.6)$$

$$\mathbf{F}_{\text{Ver}} = \begin{pmatrix} -1 \\ +1 \end{pmatrix} \quad (4.7)$$

Wie man sieht, sind diese Masken nicht symmetrisch um ein zentrales Pixel aufgebaut, so dass das Ergebnis der Ableitung *zwischen* zwei Pixeln entspricht. Um unsymmetrische Masken mit Gleichung 4.2 darstellen zu können, müssen sie virtuell um eine „0“ erweitert werden, welche keinen Einfluss auf das Ergebnis hat. Zusätzlich muss bedacht werden, dass die Position des Maximums des Faltungsergebnisses um den Wert -0.5 in die jeweilige Richtung verschoben werden muss, um auf die richtige Position zwischen den Pixeln zu kommen.

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

$$\mathbf{F}_{\text{Hor}} = \begin{pmatrix} -1 & +1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 & +1 & 0 \end{pmatrix} \quad (4.8)$$

$$\mathbf{F}_{\text{Ver}} = \begin{pmatrix} -1 \\ +1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ +1 \\ 0 \end{pmatrix} \quad (4.9)$$

Eine häufig verwendete symmetrische Maske zur Bildung der diskreten Ableitung lässt sich am Beispiel der horizontalen Maske \mathbf{F}_{Hor} durch die Kombination des vorlaufenden und nachlaufenden Gradienten herleiten.

$$\begin{aligned} i'_{\text{Hor}}[u, v] &= \frac{1}{2}((i[u, v] - i[u - 1, v]) + (i[u + 1, v] - i[u, v])) \\ &= \frac{1}{2}(i[u + 1, v] - i[u - 1, v]) \end{aligned} \quad (4.10)$$

Die zugehörigen ternären Masken ergeben sich also zu

$$\mathbf{F}_{\text{Hor,t}} = \begin{pmatrix} -1 & 0 & +1 \end{pmatrix} \quad \text{und} \quad (4.11)$$

$$\mathbf{F}_{\text{Ver,t}} = \begin{pmatrix} -1 \\ 0 \\ +1 \end{pmatrix}. \quad (4.12)$$

Eine weitere Herleitungsmöglichkeit kommt durch das Fitten einer Parabel durch die Helligkeitswerte des zentralen und der beiden angrenzenden Pixel zustande[81]. Die örtliche Ableitung dieser Parabel ausgewertet am Ort des zentralen Pixels ergibt ebenfalls diese Berechnungsvorschrift. Da meist nur das Maximum des Verlaufs und nicht der wahre Wert des Gradienten interessiert, kann auf den Faktor $\frac{1}{2}$ aus Effizienzgründen verzichtet werden.

Da Bilder im Allgemeinen einen Rauschanteil besitzen, ist es sinnvoll den Bereich in dem die Helligkeitskante gefunden werden soll zunächst zu glätten. Daher haben sich Filtermasken etabliert, die eine Tiefpassfilterung mit der Gradientenberechnung vereinen. Ein einfaches und weit verbreitetes Beispiel hierfür ist der *Prewitt-Operator*[101]. Dieser Operator führt eine Mittelung parallel zur erwarteten Kantenrichtung durch. Die zugehörigen Faltungsmasken sind

$$\mathbf{F}_{\text{Hor,P}} = \begin{pmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{pmatrix} \quad \text{und} \quad \mathbf{F}_{\text{Ver,P}} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{pmatrix}. \quad (4.13)$$

Diese Masken sind nur optimal für perfekt horizontal oder vertikal verlaufende Kanten im Bild.

Ausgehend vom Prewitt-Operator haben sich die richtungsselektiven Masken entwickelt, welche über die Parameter

- Suchpfadrichtung,
- Kantenrichtung,
- Mittelungstiefe in erwartete Kantenrichtung (Maskenhöhe),
- Mittelungstiefe in Suchpfadrichtung (Maskenbreite) und
- Kantenunschärfe (Anzahl der Nullen)

optimal für den erwarteten Helligkeitsübergang eingestellt werden können.

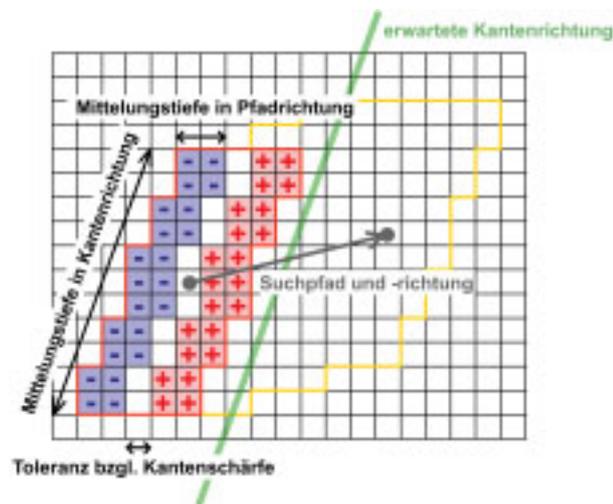


Bild 4.3: Parametrisierungsmöglichkeiten des richtungsselektiven Gradientenoperators

Abbildung 4.3 zeigt den Einfluss der Parameter auf die Maskengeometrie. In [48] werden die Einflüsse der Parameter auf das Operatorergebnis genauer untersucht.

4.1.2 Hauptachsentransformation

Die Hauptachsentransformation, auch als Principle Components Analysis (PCA) bekannt, ist ein effizientes, informationstheoretisch motiviertes Merkmalsextraktionsverfahren, das häufig zur Datenkompression eingesetzt wird. Sie gehört zur Gruppe der globalen Operatoren, da hierbei ein gesamter Bildausschnitt transformiert wird. Dabei wird der ursprüngliche Datenraum mit Hilfe einer linearen Transformation in den so genannten Eigenraum überführt, in dem die Achsen in Richtung der maximalen Streuung der Muster vorliegen [93]. Eine ausführliche Herleitung der PCA ist in [73] zu finden. An dieser Stelle soll lediglich das Berechnungsverfahren der PCA sowie deren Eigenschaften erläutert werden.

Berechnungsverfahren der PCA

Ausgangspunkt der PCA ist ein Datensatz von G Merkmalsvektoren $\underline{x} \in \mathcal{R}^N$, der für die Trainingsphase der Verfahren verwendet wird. Aus dem Datensatz wird zunächst der Mittelwertvektor $\underline{\mu}$ gebildet:

$$\underline{\mu} = \frac{1}{G} \sum_{g=1}^G \underline{x}_g \quad (4.14)$$

Mittels $\underline{\mu}$ kann nun die Kovarianzmatrix $\mathbf{C} \in \mathcal{R}^{N \times N}$ des Datensatzes nach Gleichung 4.15 bestimmt werden.

$$\mathbf{C} = E_g\{(\underline{x}_g - \underline{\mu})(\underline{x}_g - \underline{\mu})^T\} = \frac{1}{G} \sum_{g=1}^G ((\underline{x}_g - \underline{\mu})(\underline{x}_g - \underline{\mu})^T) \quad (4.15)$$

Die Eigenvektoren der Kovarianzmatrix \mathbf{C} werden als Hauptachsen des Datensatzes bezeichnet. Die Eigenvektoren \underline{u}_i können mittels Eigenwertzerlegung durch Lösen der Gleichung 4.16 berechnet werden, wobei die Eigenwerte λ_i der Kovarianzmatrix \mathbf{C} durch Lösen von Gleichung 4.17 bestimmt werden können.

$$\mathbf{C} \cdot \underline{u}_i = \underline{u}_i \lambda_i \quad (4.16)$$

$$\det(\mathbf{C} - \lambda \cdot \mathbf{I}) = 0 \quad (4.17)$$

Dabei ist \mathbf{I} die Einheitsmatrix mit der Dimension $N \times N$.

Auf Basis der Eigenvektoren \underline{u}_i und der zugehörigen Eigenwerte λ_i kann die Matrix \mathbf{U} aufgestellt werden. Die Eigenvektoren \underline{u}_i werden nach der abfallenden Größe des zugehörigen Eigenwertes λ_i sortiert und als Spalten in der Matrix \mathbf{U} eingetragen.

$$\mathbf{U} = (\underline{u}_1, \dots, \underline{u}_i, \dots, \underline{u}_N) \quad (4.18)$$

mit $\lambda_i > \lambda_{i+1} \quad \forall i \in \{1, \dots, N-1\}$

Die Spalten der Matrix \mathbf{U} spannen nun den so genannten *Eigenraum* des Trainingsdatensatzes auf. Die Transformation eines unbekanntes Datenmusters in diesen Eigenraum entspricht der Hauptachsentransformation. Abbildung 4.4 zeigt schematisch den Effekt der Hauptachsentransformation.

Man sieht, dass der Mittelwertvektor $\underline{\mu}$ den Ursprung eines neuen Koordinatensystems darstellt. Die Achsen des neuen Koordinatensystems werden durch die Richtungen der Eigenvektoren \underline{u}_i aufgespannt. Die Achse u_1 , die dem Eigenvektor mit dem größten zugehörigen Eigenwert entspricht, zeigt in die Richtung der größten Varianz des Trainingsdatensatzes. Die Achse u_2 liegt senkrecht dazu. Bei höherdimensionalen Problemen zeigen die weiteren Achsen immer in Richtung des Maximums der verbleibenden Varianz, unter der Nebenbedingung, dass ein orthogonaler Raum aufgespannt wird.

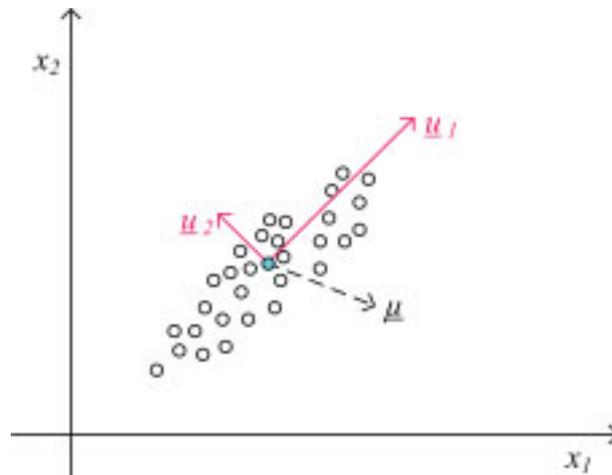


Bild 4.4: Schematische Darstellung des Effektes der Hauptachsentransformation

Die Überführung eines unbekanntes Datenmusters \underline{x} in den Eigenraum der Trainingsdaten wird mittels Gleichung 4.19 durchgeführt.

$$\underline{x}_{\text{PCA}} = \mathbf{U}^T \cdot (\underline{x} - \underline{\mu}) \quad (4.19)$$

Ein transformiertes Datenmuster $\underline{x}_{\text{PCA}}$ hat nun die Eigenschaft, dass der größte Anteil der klassenspezifischen Information, im Sinne der Hauptachsentransformation, in den ersten Einträgen des Vektors zu finden ist. Das bedeutet auch, dass man durch Vernachlässigung der letzten Einträge in $\underline{x}_{\text{PCA}}$ nur wenig Information verliert. Es hat also eine Informationskomprimierung auf die ersten Einträge des Vektors stattgefunden.

Anwendung in der Bildverarbeitung

Bei der Verwendung der PCA in der Bildverarbeitung zur Analyse eines Bildausschnitts, muss häufig ein Vorverarbeitungsschritt erfolgen, damit Ausschnitte verschiedener Größe verglichen werden können. Eine Voraussetzung der PCA ist nämlich, dass alle Datenmuster die gleiche Dimension $R_{\text{PCA}} \times C_{\text{PCA}}$ besitzen. Um das zu erreichen, kann das Verfahren der bilinearen Interpolation verwendet werden. Damit können Bildausschnitte der Größe $U' \times V'$ in interpolierte Bilder der Dimension $U_{\text{PCA}} \times V_{\text{PCA}}$ umgerechnet werden, unter der Bedingung, dass gilt:

$$U' \geq U_{\text{PCA}} \wedge V' \geq V_{\text{PCA}}.$$

Somit müssen für dieses Verfahren U_{PCA} und V_{PCA} so gewählt werden, dass die Bedingungen für alle Bildausschnitte erfüllt werden können. Wegen der hohen Auflösung der Kamera und des hohen Informationsgehaltes der schon in kleinen Bildausschnitten vorhanden ist, funktioniert dieser Ansatz für viele Anwendungsfälle ausreichend gut.

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

Die entstehenden Bildausschnitte der geforderten Größe können dann in den von der PCA erwarteten Spaltenvektor $\underline{x}_{\text{Bild}}$ umgeschrieben werden, in dem die Spalten des interpolierten Bildausschnitts zu einem Gesamtvektor aneinander gehängt werden. Dieses Vorgehen wird schematisch in Abbildung 4.5 beschrieben.

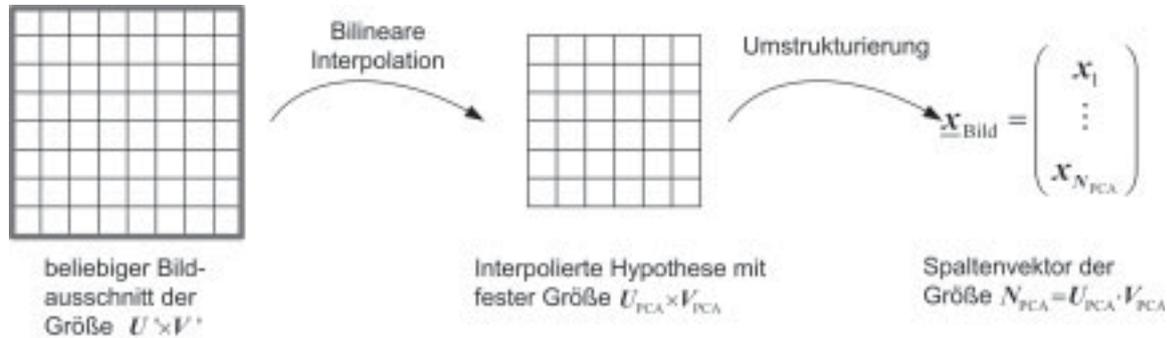


Bild 4.5: Bilineare Interpolation und Umstrukturierung für PCA

Somit kann die PCA auf beliebige Bildausschnitte angewendet werden.

Der Eigenface-Ansatz Der Eigenface-Ansatz ist einer der ältesten Ansätze zur Detektion von Gesichtern mittels Bildverarbeitung[93]. Dabei werden die mittels Hauptachsentransformation in den Eigenraum überführten Datenmuster $\underline{x}_{\text{Bild}}$ durch die Linearkombination aus Gleichung 4.20 wieder in den Ursprungsraum zurück transformiert.

$$\underline{x}_{\text{Bild}}^* = \begin{pmatrix} x_{\text{Bild},1}^* \\ \vdots \\ x_{\text{Bild},N_{\text{Bild}}}^* \end{pmatrix} = \sum_{k=1}^K x_{\text{Bild},k} \cdot \underline{u}_k + \underline{\mu} \quad (4.20)$$

Allerdings werden dabei nur die ersten K Eigenvektoren u_k und deren entsprechende Einträge $x_{\text{Bild},k}$ berücksichtigt. Unter der Annahme, dass das unbekannte Datenmuster tatsächlich der erwarteten Klasse (z. B. PKW-Heckansicht) entspricht, müsste das rücktransformierte Datenmuster $\underline{x}_{\text{Bild}}^*$ seinem ursprünglichen Pendant $\underline{x}_{\text{Bild}}$ sehr ähnlich sein. Datenmuster (z. B. LKW-Heckansicht), die nicht aus der erwarteten Klasse stammen, ähneln nach der Rücktransformation trotzdem der erwarteten Klasse (PKW-Heckansicht) und ähneln somit dem Originalbild (LKW-Heckansicht) nicht mehr.

Daraus folgt, dass die Klassenzuordnung unter Einführung eines Ähnlichkeitsmaßes zwischen dem Original- und dem rekonstruierten Datenmuster durchgeführt werden kann. Das hier verwendete Ähnlichkeitsmaß ist der Korrelationskoeffizient ρ nach Gleichung 4.21.

$$\begin{aligned}
\rho(x, x^*) &= \frac{E((x - E(x)) \cdot (x^* - E(x^*)))}{\sigma_x \cdot \sigma_{x^*}} \\
&= \frac{\sum_{n=1}^{N_{\text{PCA}}} (x_n - \bar{x})(x_n^* - \bar{x}^*)}{\sqrt{\sum_{n=1}^{N_{\text{PCA}}} (x_n - \bar{x})^2 \cdot \sum_{n=1}^{N_{\text{PCA}}} (x_n^* - \bar{x}^*)^2}} \quad (4.21) \\
\text{mit } \bar{x} &= \frac{1}{N_{\text{PCA}}} \sum_{n=1}^{N_{\text{PCA}}} x_n, \quad \bar{x}^* = \frac{1}{N_{\text{PCA}}} \sum_{n=1}^{N_{\text{PCA}}} x_n^* \\
\text{und } N_{\text{PCA}} &= U_{\text{PCA}} \cdot V_{\text{PCA}}
\end{aligned}$$

4.2 Merkmalspezifische Operatoren

Bildverarbeitungsoperatoren werden dazu verwendet, aus der Helligkeitsinformation der einzelnen Pixel die gewünschte Information zu extrahieren. Dabei wird häufig auf Basis eines Abbildungsmodells eines realen Objektmerkmals eine Faltungsmaske erzeugt, welche auf ausgewählte Bildbereiche angewendet wird. Um ein gewünschtes Merkmal zu detektieren, kann es manchmal nötig sein, Operationen auf den Ergebnissen anderer Operationen auszuführen. Meist lässt sich durch eine geschickte Berechnungsreihenfolge oder vorab ausgeführte Standardberechnungen die Anzahl der Rechenoperationen und somit die benötigte Rechenzeit eines Operators verkürzen. Durch diese optimierte Kombination einzelner Operationen entstehen sehr spezialisierte und komplexe Operatoren. Die in dieser Arbeit verwendeten Standardoperatoren und die im Rahmen dieser Arbeit entwickelten werden im Folgenden beschrieben.

4.2.1 Gradientenhäufungsoperator

Basierend auf dem parametrisierbaren Gradientenoperator ist ein neuer Operator für die Aufgabe der Fahrzeugentstehung entstanden. Der Gradientenhäufungsoperator berechnet für jeden Punkt entlang seines Suchpfades in einer ausgewählten Region die Summe der Absolutbeträge aller Gradienten senkrecht zu seiner Laufrichtung. Dadurch liefern Objekte mit einer Häufung von Kanten parallel zur Suchrichtung ein markantes Maximum im Operatorverlauf und können darüber detektiert werden. Dieser Operator hat sich bei der Grobdetektion von Fahrzeugen im Videobild sehr bewährt. Die wesentlichen Schritte bei der Berechnung des Ergebnisarrays sind in Abbildung 4.6 dargestellt.

Im Fall der Suche nach einer horizontalen Kantenhäufung (vertikale Gradientenhäufung), wie sie bei Fahrzeugheckansichten auftreten, wird zunächst an jeder Position des zu untersuchenden Bildbereichs der vertikale Gradient berechnet (Schritt 1). Die Mittelungstiefe des Kantenoperators wird an die erwartete Breite der Abbildung eines Fahrzeugs angepasst. Danach werden alle Gradientenbeträge aller Spalten über den gesamten untersuchten Bereich aufsummiert und in einem Zwischenspeicher abgelegt (Schritt 2). Dieser wird

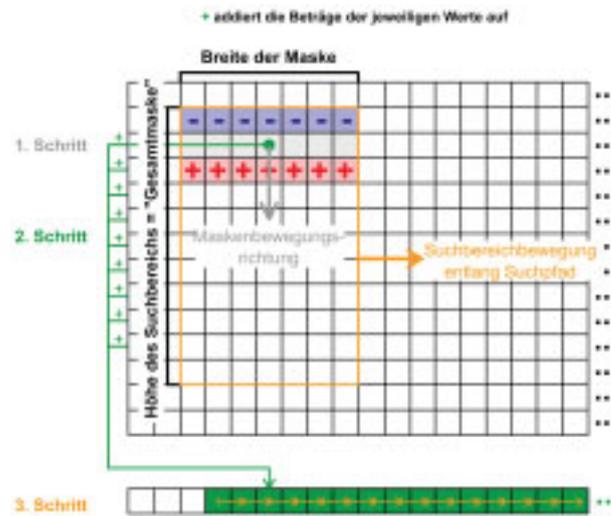


Bild 4.6: Arbeitsweise des Gradientenhäufungsoperators

abschließend auf vorhandene Maxima mit einer entsprechenden Ausprägung untersucht (Schritt 3).

4.2.2 Konturbewertungsoperator

Der „Konturbewertungsoperator“ wird dazu verwendet zu überprüfen wie geschlossen eine angenommene Kontur tatsächlich ist. Abbildung 4.7 zeigt den Aspekt, den der Konturbewertungsoperator anhand eines Kennwerts bewertet. Je deutlicher sich eine Kontur von der Umgebung abhebt und je besser diese Kontur zur Erwartung passt, desto größer wird der resultierende Gütewert.

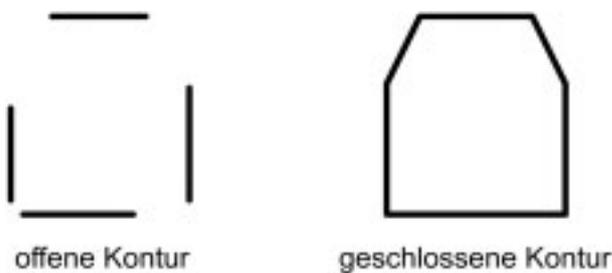


Bild 4.7: Veranschaulichung der Zielsetzung des Konturbewertungsoperators

Der Konturbewertungsoperator ist in dieser Arbeit speziell für die Fahrzeugheckerkennung implementiert. Als Konturen stehen lediglich die eines Standard-Fahrzeughecks sowie die eines Rechtecks zur Auswahl. Da der Operator sehr genau zur erwarteten Heckansicht passen muss, ist viel Vorwissen zur richtigen Parametrisierung notwendig. Deshalb bietet sich dieser Operator dafür an, eine bestimmte vorgegebene Stelle im Bild zu bewerten.

Im später beschriebenen Anwendungsfall wird er dazu verwendet die Positionen potentieller Fahrzeuge, die aus einer vorgeschalteten Erkennungsstufe resultieren, genauer zu untersuchen.

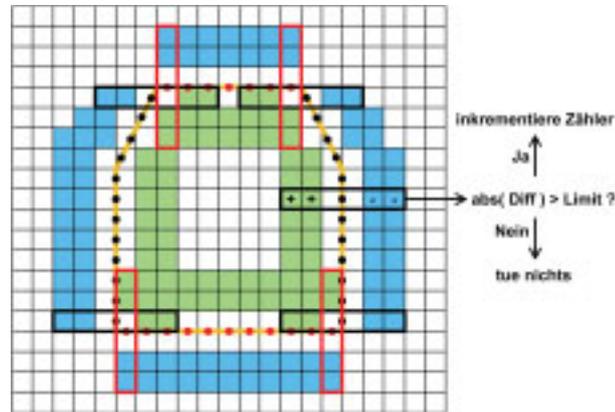


Bild 4.8: Arbeitsweise Konturoperator

Zunächst werden schrittweise die in Abbildung 4.8 dargestellten inneren und äußeren Bereiche voneinander subtrahiert. Anschließend wird an jeder Stelle der Kontur überprüft, ob der Betrag des berechneten Gradienten eine festgelegte Rauschschwelle überschreitet. Durch die Betragsbildung wird eine Unabhängigkeit von der Art des Helligkeitsübergangs (Hell-Dunkel oder Dunkel-Hell) erreicht. Bei Überschreitung der Schwelle wird ein interner Zähler des Operators inkrementiert. Als Ergebnis liefert der Operator das Verhältnis des Zählerstands zur Anzahl der Untersuchten Positionen entlang des Konturwegs zurück. Wie beim Standard-Gradientenoperator ist auch hier die Wahl der richtigen Anzahl an Nullzeilen und die Größe der zur Differenzberechnung herangezogenen Positiv- und Negativblöcke wichtig.

4.2.3 Symmetrieoperatoren

Wie der Name sagt, werden diese Operatoren eingesetzt, um besonders symmetrische Objekte zu finden. Typischerweise sind vom Menschen erbaute Gegenstände häufig symmetrisch; so auch zum Großteil Fahrzeugheckansichten. Dabei sind zwei Typen von Symmetrieoperatoren implementiert. Zum einen der Symmetrieoperator nach U. Regensburger und V. Graefe (hier als Helligkeitssymmetrieoperator bezeichnet), der pixelweise basierend auf den tatsächlichen Grauwerten im Bild Symmetrien berechnet [72]. Zum anderen ein neu entstandener Operator, der die Struktursymmetrie basierend auf dem Gradientenbild untersucht.

Helligkeitssymmetrieoperator

Der Helligkeitssymmetrieoperator nach Regensburger bildet jeweils die Differenz zwischen zwei symmetrisch zur untersuchten Achse liegenden Pixelwerten (Schritt 1 in Abbildung 4.9) und addiert die Beträge der Differenzen im gesamten Maskenbereich auf (Schritt 2 in Abbildung 4.9). Diese Summe wird als Ergebnis für die aktuelle Wegposition im Ergebnisarray abgelegt und der Suchbereich um eine Position weiter geschoben (Schritt 3 in Abbildung 4.9).

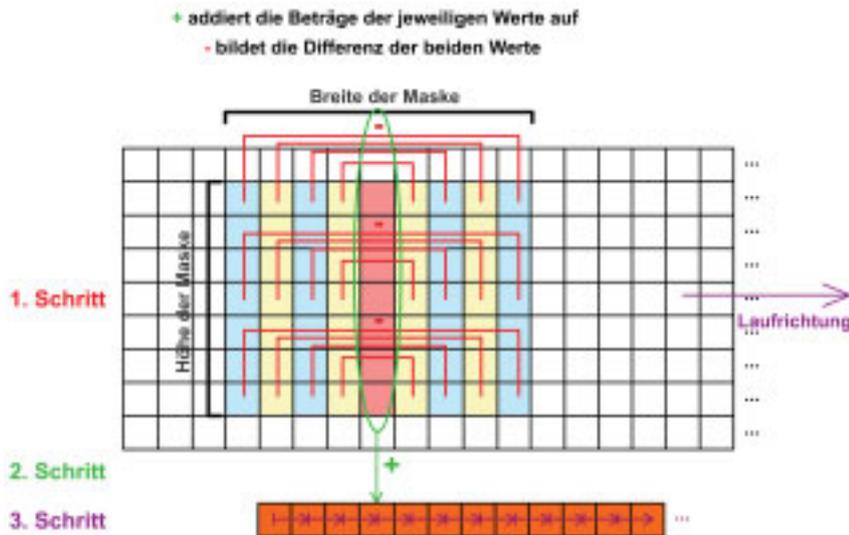


Bild 4.9: Arbeitsweise des Helligkeitssymmetrieoperators nach [72]

Der so erhaltene Verlauf enthält minimale Werte an Positionen an denen die Helligkeitswerte im Bild maximal symmetrisch zur erwarteten Symmetrieachsenrichtung sind.

Gradientensymmetrieoperator

Der Gradientensymmetrieoperator bildet zunächst die Gradienten parallel zur Laufrichtung im untersuchten Bereich (Schritt 1 in Abbildung 4.10). Anschließend werden die Absolutbeträge der Gradienten entlang parallel zur erwarteten Symmetrieachsenrichtung gewählter Bahnen aufsummiert und in einem temporären Vektor abgelegt. Auf diesem Vektor (Schritt 2 in Abbildung 4.10) wird dann eine Symmetriebetrachtung durch paarweise Differenzbildung um die Symmetrieposition und anschließendes Aufaddieren der Differenzbeträge durchgeführt (Schritt 3 in Abbildung 4.10). Das hier erhaltene Ergebnis wird als Wert für die Symmetrie der Struktur an der aktuellen Position gespeichert und die Maske um eine Position weiter geschoben (Schritt 4 in Abbildung 4.10).

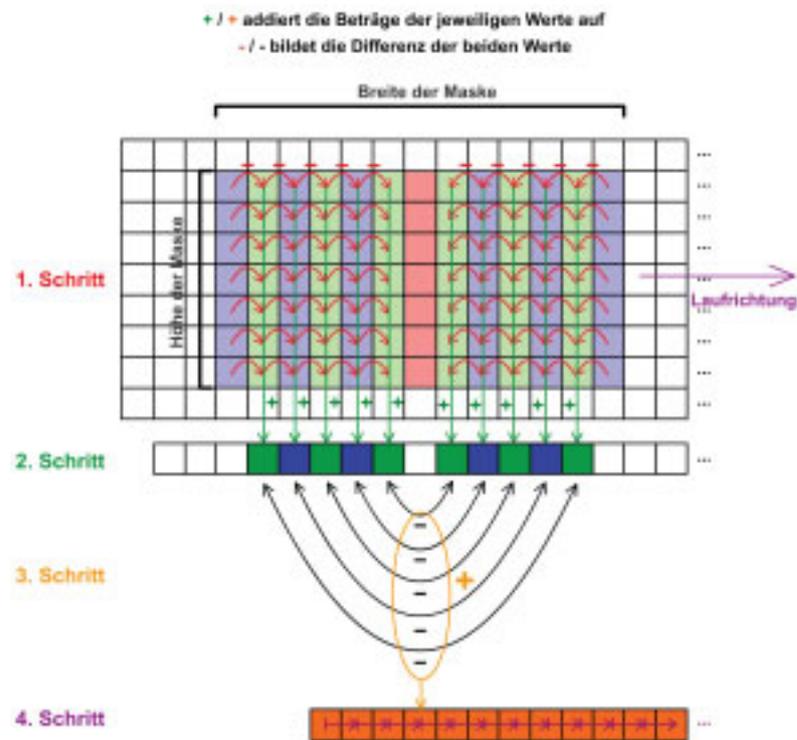


Bild 4.10: Arbeitsweise des Gradientensymmetrieoperators

Vergleich der Verfahren

Beide Verfahren haben Vor- und Nachteile, die dazu führen, dass je nach Problemstellung der eine oder der andere Operator besser geeignet ist. Die Gradientensymmetrie ist robuster gegen einseitige Beleuchtung von Objekten und Schlagschatten, da sich ein harter Hell-Dunkel-Übergang innerhalb des untersuchten Bereichs durch die Verwendung des Gradienten zur Symmetrieberechnung nur wenig auf das Ergebnis auswirkt. Bei der Berechnung der Symmetrie auf Basis der Grauwerte der Pixel führt eine solche Situation zu einer Asymmetrie, da beispielsweise die eine Fahrzeughälfte durch schrägen Sonneneinfall deutlich heller ist, als die andere. Die Rechenzeitanforderungen des Gradientensymmetrieoperators sind, bedingt durch bessere Optimierungsmöglichkeiten bei der Implementierung, geringer als die des Intensitätssymmetrieoperators. Durch eine vorgeschaltete Unterabtastung des Originalbildes kann auch bei der Berechnung der Helligkeitssymmetrie Rechenzeit eingespart werden. Gemäß dem Grundsatz, dass der Rauschanteil eines Signals durch dessen Ableitung verstärkt wird, ist der Signalverlauf des Gradientensymmetrieoperators weniger eindeutig als der des Intensitätssymmetrieoperators. Das Rauschproblem des gradientenbasierten Operators wird zwar durch die Mittelung über die Maskenhöhe entschärft, dennoch reicht die Robustheit und Stabilität nicht an die des Symmetrieoperators auf Grauwertbasis heran. In beiden Fällen ist eine gute Kenntnis des aufzufindenden Objekts von Vorteil, da nur mit einer an die erwartete Größe angepassten Maske wirklich gute Ergebnisse erreicht werden können.

4.3 Detektoren

Als Detektoren werden in dieser Arbeit Algorithmen bezeichnet, deren Komplexität über die einfache Transformation eines Bildausschnitts in ein Ergebnisarray hinaus geht. Die nachfolgenden Detektoren können, wie Operatoren, auf Bildbereichen ausgeführt werden. Dabei werden jedoch mehrere Operatoren in einer gewissen Strategie ausgeführt, eine Bewertung der Ergebnisse durchgeführt und mögliche Objektpositionen (Objekthypothesen) zurückgegeben.

4.3.1 Reifendetektor

Der in diesem Kapitel vorgestellte Reifendetektor basiert auf dem in der Dissertation von U. Hofmann [48] beschriebenen parametrisierbaren Reifendetektor. Der Ansatz basiert auf den Verfahren des Template Matching, das ebenfalls eine Korrelationsanalyse zwischen einer Maske, die dem gesuchten Muster möglichst gut ähnelt und einem gewählten Bildausschnitt durchführt. In diesem Fall werden ganze 3D-Rädermodelle für eine erwartete Position generiert und in das 2D-Bild projiziert. Die daraus entstehenden Masken werden dann für die Berechnungen verwendet.

Da im hier verfolgten Anwendungsfall später der Fokus auf der Erkennung von Fahrzeugheckansichten liegt, kann auf ein einfaches Reifenmodell zurückgegriffen werden. Dies hat den Vorteil, dass nur noch ein Parameter, nämlich die Breite des Reifenmodells, in Abhängigkeit von der Größe der Fahrzeughypothese eingestellt werden muss. Dank der relativ guten Robustheit des Modells liefert der Detektor auch bei leicht schräg betrachteten Reifen noch gute Ergebnisse. Der genaue Aufbau der Maske ist in Abbildung 4.11 dargestellt. Dort wird auch die Vorgehensweise zur Berechnung des Korrelationswertes für eine Bildposition illustriert.

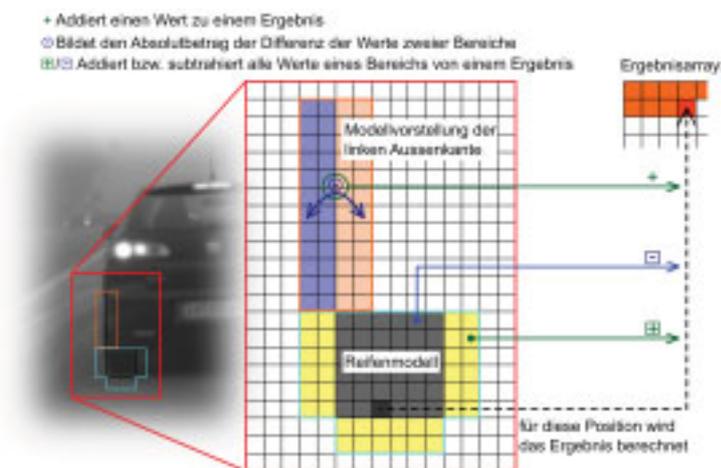


Bild 4.11: Arbeitsweise des Reifendetektor

Der untere Teil der Maske symbolisiert die Ansicht eines Teils der Lauffläche des Reifens. Es wird davon ausgegangen, dass die Lauffläche des Reifens dunkler ist als seine Umgebung. Daher wird die Differenz zwischen den innen liegenden Pixelwerten und den außen liegenden berechnet. Die zusätzliche Berechnung der absoluten horizontalen Gradienten an den verlängerten Außenseiten beruht darauf, dass Reifen normalerweise an der äußeren vertikalen Kante eines Fahrzeugs liegen. Der Bereich modelliert also die Außenkante des Fahrzeugs. Je nachdem ob nach Reifen auf der rechten oder linken Seite des Fahrzeugs gesucht wird, wird die Maske entsprechend gespiegelt verwendet. Damit sich die Ergebnisse der Maske für die Außenkante mit den Ergebnissen der Maske für das Reifenmodell verstärken, sind die Vorzeichen der Reifenmaske so gewählt, dass an einer Stelle mit guter Übereinstimmung hohe Werte, bei niedriger Übereinstimmung niedrige Korrelationswerte erhalten werden.

Die einzelnen Ergebnisse für die Bildpositionen werden in einem Ergebnisarray abgelegt. Da die beschriebene Suchmaske in der Praxis in horizontaler und vertikaler Richtung über einen ausgewählten Bereich geschoben wird, handelt es sich dabei um ein zweidimensionales Array. Aus diesem Grund wird direkt nach den Operatorberechnungen die Auswertung der Ergebnisse in Form einer Suche nach den höchstens drei besten Maximalwerten innerhalb des Ergebnisarrays gestartet. Diese null bis drei Werte werden dann als Reifenhypothesen zurückgegeben.

4.3.2 Lichtdetektor

Einen weiteren Detektor stellt der Lichtdetektor dar, der für die Detektion von Rückleuchten, die sich im Videobild als helle Flächen variabler Größe und Form darstellen (im Folgenden als *Lichtblob* bezeichnet), entwickelt ist. Durch das Fehlen von Farbinformation im Grauwertbild könnte der Detektor auch zur Erkennung von Frontscheinwerfern und sonstigen hellen Punkten im Bild verwendet werden. Das Verfahren hat sich gut für die Detektion von Rückleuchten bei Nacht bewährt.

Das Verfahren lässt sich in zwei Schritte unterteilen:

1. Detektion von Helligkeitsintervallen in den Bildspalten.
2. Verkettung der benachbarten Intervalle zu so genannten Lichtblobs.

Detektion der Helligkeitsintervalle

Da die Helligkeitsübergänge zwischen Lichtblob und Umgebung meist sehr scharf sind und die Lichtblobs in größerer Entfernung nur wenige Pixel ausgedehnt sein können, wird bei der gewählten Gradientenmaske auf die mittlere Null verzichtet. Der Aufbau der Korrelationsmaske \mathbf{F}_{HI} ist daher wie in Abbildung 4.12 gewählt.

Die Maske \mathbf{F}_{HI} wird spaltenweise über das Bild geschoben. Dabei ergeben sich hohe positive Korrelationswerte an Übergängen von dunklen zu hellen Segmenten (Dunkel-Hell-Übergang) und entsprechend hohe negative Werte an Hell-Dunkel-Übergängen. Es ent-

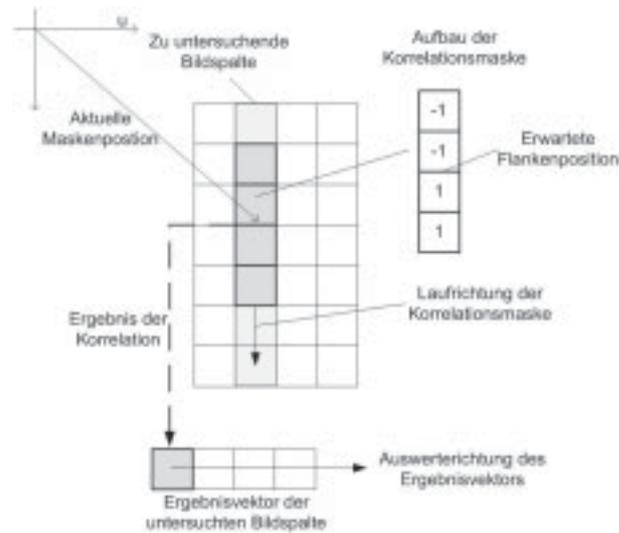


Bild 4.12: Helligkeitsintervallsuche in einer Bildspalte

steht das Ergebnisarray \mathbf{B}_{FH} . Dieses wird wiederum spaltenweise durchlaufen und die Einträge mit den Schwellen τ_{pos} und τ_{neg} verglichen. Überschreitet der Gradientenwert den Schwellwert τ_{pos} , stellt die entsprechende Position in der Spalte des Ergebnisarrays den Beginn eines Helligkeitsintervalls dar. Das Ende dieses Helligkeitsintervalls wird detektiert, wenn in der gleichen Spalte der Korrelationswert die negative Schwelle τ_{neg} unterschreitet. Das i -te Helligkeitsintervall HI_i wird mit der Datenstruktur gemäß Tabelle 4.1 gespeichert.

Bezeichnung	Datentyp	Beschreibung
u_i	<i>int</i>	Spaltenindex von HI_i
$v_{a,i}$	<i>int</i>	Zeilenindex des Anfangs von HI_i
$v_{e,i}$	<i>int</i>	Zeilenindex des Endes von HI_i
h_i	<i>int</i>	Gesamthelligkeit von HI_i , d.h. Summe der Grauwerte
f_i	<i>bool</i>	Flag: verkettet= <i>true</i> ; unverkettet= <i>false</i>

Tabelle 4.1: Beschreibung eines Helligkeitsintervalls

Verkettung der Helligkeitsintervalle

Die detektierten Helligkeitsintervalle in den Bildspalten werden im zweiten Schritt gemäß der Vorschrift aus Abbildung 4.13 zu so genannten Lichtblobs verkettet. Dabei wird für jedes der unverketteten Intervalle der nachfolgend beschriebene Verkettungsalgorithmus durchlaufen (siehe Ablaufdiagramm 4.13). Die Informationen über den l -ten Lichtblob LB_l werden in einer Datenstruktur nach Tabelle 4.2 gespeichert.

Bezeichnung	Datentyp	Beschreibung
$u_{a,l}$	<i>int</i>	minimaler Spaltenindex von LB_l
$u_{e,l}$	<i>int</i>	maximaler Spaltenindex von LB_l
$v_{a,l}$	<i>int</i>	gemittelter minimaler Zeilenindex von LB_l
$v_{e,l}$	<i>int</i>	gemittelter maximaler Zeilenindex von LB_l
h_l	<i>int</i>	Gesamthelligkeit von LB_l , d.h. Summe der Grauwerte
f_l	<i>int</i>	Gesamtfläche von LB_l , d.h. Anzahl der Pixel

Tabelle 4.2: Beschreibung eines Lichtblobs

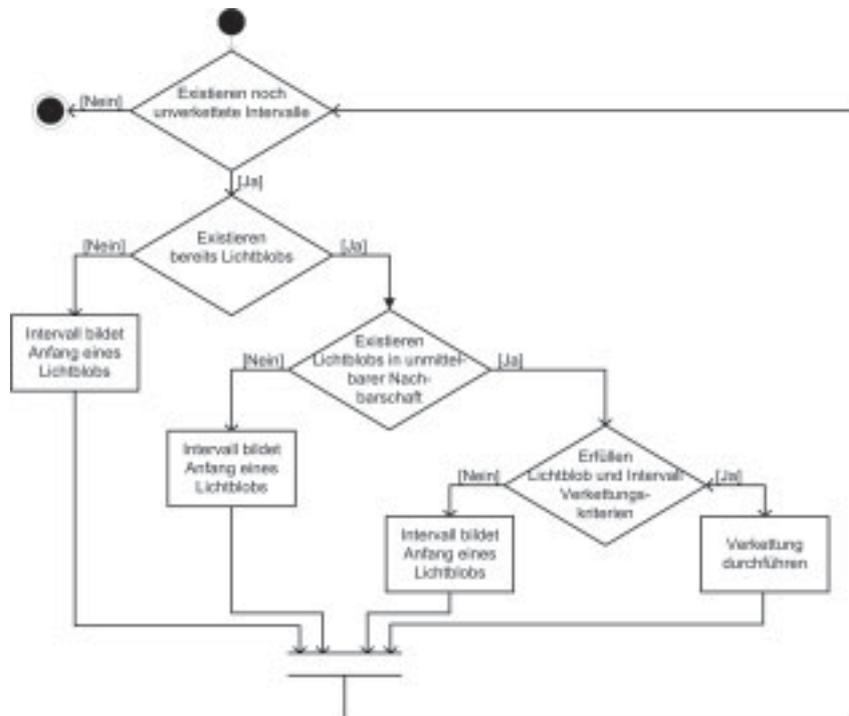


Bild 4.13: Ablaufdiagramm Lichtblobverkettung

Zunächst werden für das aktuelle Helligkeitsintervall mehrere Verkettungskriterien mit Bezug auf alle L bereits vorhandenen Lichtblobs überprüft.

$$u_i - u_{e,l} = 1 \quad (4.22)$$

$$|v_{a,i} - v_{a,l}| < \tau_a \quad (4.23)$$

$$|v_{e,i} - v_{e,l}| < \tau_e \quad (4.24)$$

$$\tau_{\Delta h, \min} < \frac{h_i}{v_{e,i} - v_{a,i} + 1} \cdot \frac{f_l}{h_l} < \tau_{\Delta h, \max} \quad (4.25)$$

τ_a und τ_e legen dabei die maximal zulässige Differenz der Indices der betrachteten Zeilenanfänge bzw. Zeilenenden fest. $\tau_{\Delta h, \min}$ und $\tau_{\Delta h, \max}$ bestimmen die Grenzen für den Vergleich der mittleren Helligkeiten.

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

Existieren noch keine Lichtblobs ($L = 0$) oder erfüllt kein vorhandener Lichtblob alle Kriterien, bildet das aktuell betrachtete Helligkeitsintervall HI_i den Anfang eines neu gebildeten Lichtblobs LB_{L+1} . Dabei berechnen sich die Einträge der Beschreibungsstruktur folgendermaßen:

$$u_{a,L+1} = u_i \quad (4.26)$$

$$u_{e,L+1} = u_i \quad (4.27)$$

$$v_{a,L+1} = v_{a,i} \quad (4.28)$$

$$v_{e,L+1} = v_{e,i} \quad (4.29)$$

$$h_{L+1} = h_i \quad (4.30)$$

$$f_{L+1} = v_{e,i} - v_{a,i} + 1 \quad (4.31)$$

Erfüllt ein vorhandener Lichtblob LB_l jedoch alle Bedingungen, wird das betrachtete Helligkeitsintervall HI_i zum Lichtblob hinzugefügt. Daraus ergibt sich folgende Beschreibung des erweiterten Lichtblobs LB'_l :

$$u'_{a,l} = u_{a,l} \quad (4.32)$$

$$u'_{e,l} = u_i \quad (4.33)$$

$$v'_{a,l} = \frac{M}{M+1}v_{a,l} + \frac{1}{M+1}v_{a,i} \quad (4.34)$$

$$v'_{e,l} = \frac{M}{M+1}v_{e,l} + \frac{1}{M+1}v_{e,i} \quad (4.35)$$

$$h'_l = h_l + h_i \quad (4.36)$$

$$f'_l = f_l + v_{e,i} - v_{a,i} + 1 \quad (4.37)$$

$$\text{mit } M = u_{e,l} - u_{a,l} + 1.$$

Der Spaltenanfang des neuen Lichtblobs ist somit gleich dem Spaltenanfang des bereits bestehenden Lichtblobs. Das Spaltenende des neuen Lichtblobs wird durch den Spaltenindex des Helligkeitsintervalls bestimmt. Der neue Zeilenanfang bzw. das neue Zeilenende wird durch eine rekursive Mittelung von Lichtblob und Intervall bestimmt. Für die Gesamthelligkeit und Gesamtfläche des neuen Lichtblobs werden die entsprechenden Größen des Helligkeitsintervalls und des Lichtblobs addiert.

4.3.3 Bodenreflexionsdetektor

Ein stark ausgeprägtes und fast immer vorhandenes Merkmal zur Erkennung von Fahrzeugen in Nachtszenen ist, neben den Rückleuchten der Fahrzeuge selbst, die Reflexion ihrer Lichtstrahlung am Boden (siehe Abbildung 4.14). Diese diffuse Lichtreflexion kann zur Detektion der Fahrzeugunterkante und dadurch zur Abschätzung der Entfernung eines Fahrzeugs bei Nacht sehr gut verwendet werden.



Bild 4.14: Bodenreflexionen der Rückleuchten

Zur Reflexionsdetektion wurde ein spezieller Detektor entwickelt, der bei der Detektion von Fahrzeugen bei Nacht in Kapitel 6.4 entscheidende Rolle spielt. Der Bodenreflexionsdetektor besteht aus den Stufen

1. Zeilenweise Mittelung
2. Korrelation
3. Maximumsuche

Diese sind in Abbildung 4.15 schematisch dargestellt.

Zeilenweise Mittelung:

Im ersten Schritt wird der Bereich innerhalb der übergebenen ROI in Spaltenrichtung durchlaufen und für jede Bildzeile der mittlere Helligkeitswert aller ebenfalls innerhalb der ROI befindlichen Pixel berechnet. Daraus resultiert der „Ergebnisvektor der zeilenweisen Mittelung“.

Korrelation:

Auf dem entstandenen Ergebnisvektor wird nun die Korrelationsanalyse mit der signalangepassten Korrelationsmaske durchgeführt. Bei dieser hochgradig unsymmetrischen Maske ist abermals eine Umformung nötig, um die Korrelation mittels Gleichung 4.2 mathematisch darstellen zu können. Die Gründe für diese unsymmetrische Auslegung der Maske sind

- die beschränkte Größe des dunklen Bildbereichs oberhalb eines vorhandenen Reflexionsbereichs,
- der unscharfe Übergang hin zum helleren Reflexionsbereich und
- die große vertikale Ausdehnung des Reflexionsbereichs.

Das Ergebnis dieser Korrelation wird im „Ergebnisvektor der Korrelation“ abgespeichert. Da in der Korrelationsmaske wiederum auf eine zentrale „0“ verzichtet wird, liegt die

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

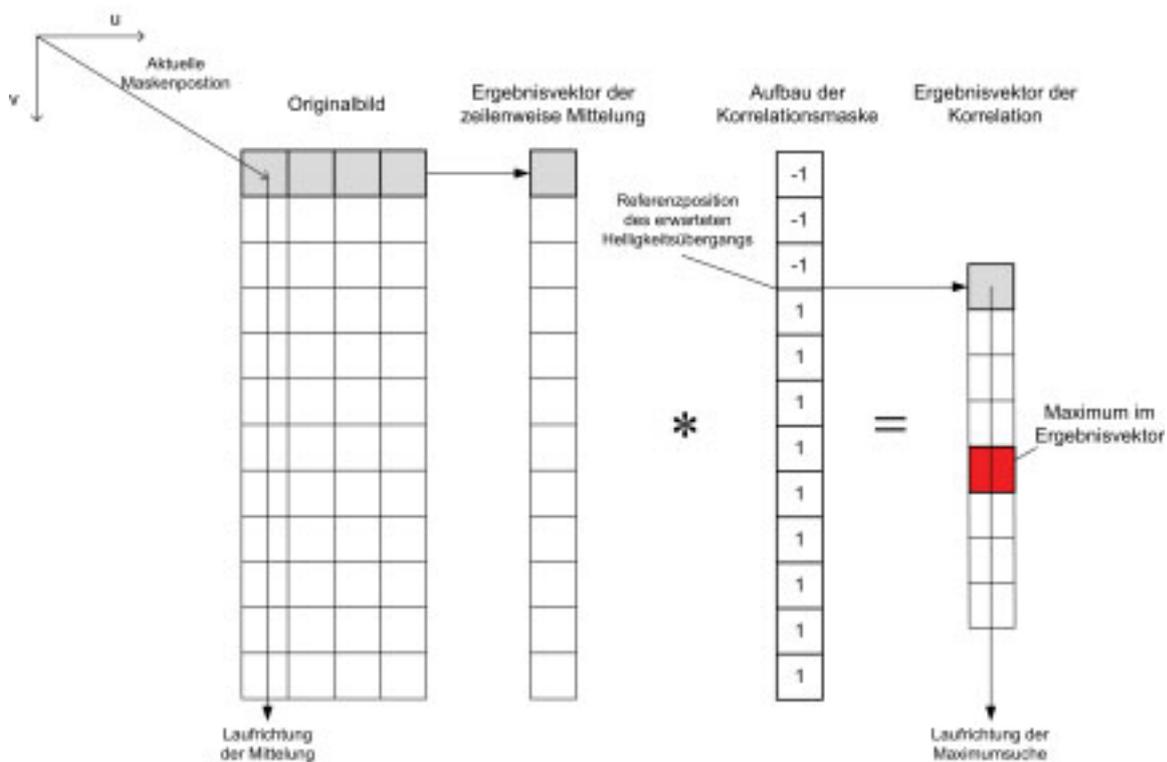


Bild 4.15: Schematische Darstellung der Vorgehensweise zur Detektion von Bodenreflexionen

Referenzposition für den erwarteten Helligkeitsübergang genau zwischen zwei Pixelpositionen im Originalbild bzw. Ergebnispositionen der zeilenweisen Mittelung. Folglich muss das halbe Pixel bei der Bestimmung der Merkmalsposition im Originalbild berücksichtigt werden. In Abbildung 4.15 ist dies durch die Verschiebung des Ergebnisvektors der Korrelation um ein halbes Pixel zum Ausdruck gebracht.

Maximumsuche:

Den Abschluss der Bodenreflexionsdetektion bildet eine Maximumsuche in diesem Ergebnisvektor. Die Position des Maximums mit dem höchsten Signal-Rausch-Abstand wird als Position der Übergangskante vom unbeleuchteten zum diffus reflektierenden Bodenbereich interpretiert.

4.3.4 Dachkantendetektor für Fahrzeugheckansichten bei Nacht

Bei der genauen Betrachtung von Fahrzeugheckansichten bei Nacht fällt auf, dass sich der schmale Bereich, der von der Dachfläche gesehen wird, als homogener Bereich darstellt. Zur Verdeutlichung wurde in Abbildung 4.16 im Bereich der Oberkante des abgebildeten Fahrzeugs ein lokaler Histogrammausgleich durchgeführt.

Wie man sieht, hebt sich der Übergang vom Hintergrund zur Fahrzeugoberkante einerseits durch die Homogenität der Dachfläche und andererseits durch einen klaren verti-



Bild 4.16: Verdeutlichung des homogenen Bereichs am Fahrzeugdach

kalen Grauwertsprung von seiner Umgebung ab. Die Detektion der Fahrzeugoberkante wird deshalb durch die Kombination eines Laplace-Operators mit einem Kantendetektor durchgeführt. Eine schematische Darstellung der Korrelationsmasken und der Verrechnung der Teilergebnisse ist in Abbildung 4.17 zu sehen.

Die Maske zur Kantendetektion besitzt den gleichen Aufbau wie die Maske zur Detektion der Helligkeitsintervalle in Kapitel 4.3.2. Der Laplace-Operator besitzt den bekannten symmetrischen Aufbau. In homogenen Bildbereichen wird als Korrelationsantwort des Laplace-Operators ein Wert nahe Null erwartet.

Bei der Ausführung der Detektion werden die beiden Masken vertikal aneinander gehängt, so dass der Laplace-Operator örtlich vor dem Kantenoperator entlang des Suchpfades läuft. Für jede Position in der übergebenen ROI wird der Quotient der beiden Operatorantworten berechnet, wobei die Laplace-Antwort im Nenner steht. Somit ergeben sich sehr hohe Werte für starke Helligkeitssprünge mit anschließendem homogenen Bereich, so wie es für eine Dachkante erwartet wird. In den zu einer Bildspalte gehörenden Ergebniswerten wird anschließend eine Maximumsuche durchgeführt. Die entsprechenden Bildzeilenindices¹⁾ der resultierende Maxima werden im Ergebnisvektor \underline{e}_M gespeichert (siehe Abbildung 4.18).

Den Abschluss der Oberkantendetektion bildet eine Mediansuche²⁾ im Vektor \underline{e}_M . Der ermittelte Medianwert wird als Bildzeilenindex für die Dachkantenposition im Bild zurückgegeben.

¹⁾ Die zu den Einträgen im Ergebnisvektor zugehörigen Merkmalspositionen im Originalbild entsprechen jeweils der erwarteten Flankenposition im Kantenoperator. Das muss bei der Auswertung der Ergebnisse berücksichtigt werden.

²⁾ Definition nach [17]: Der Median kann bei einer ungeraden Anzahl von N Messwerten bestimmt werden, indem die Messwerte der Größe nach geordnet werden und die unteren $(N - 1)/2$ abgezählt werden. Der nächstgrößere Wert ist dann der Medianwert. Ist N geradzahlig, werden die unteren 50% der geordneten Fälle abgezählt. Der Mittelwert zwischen dem größten der zu den unteren 50% gehörenden Werte und dem darauf folgenden Wert kennzeichnet den Medianwert.

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

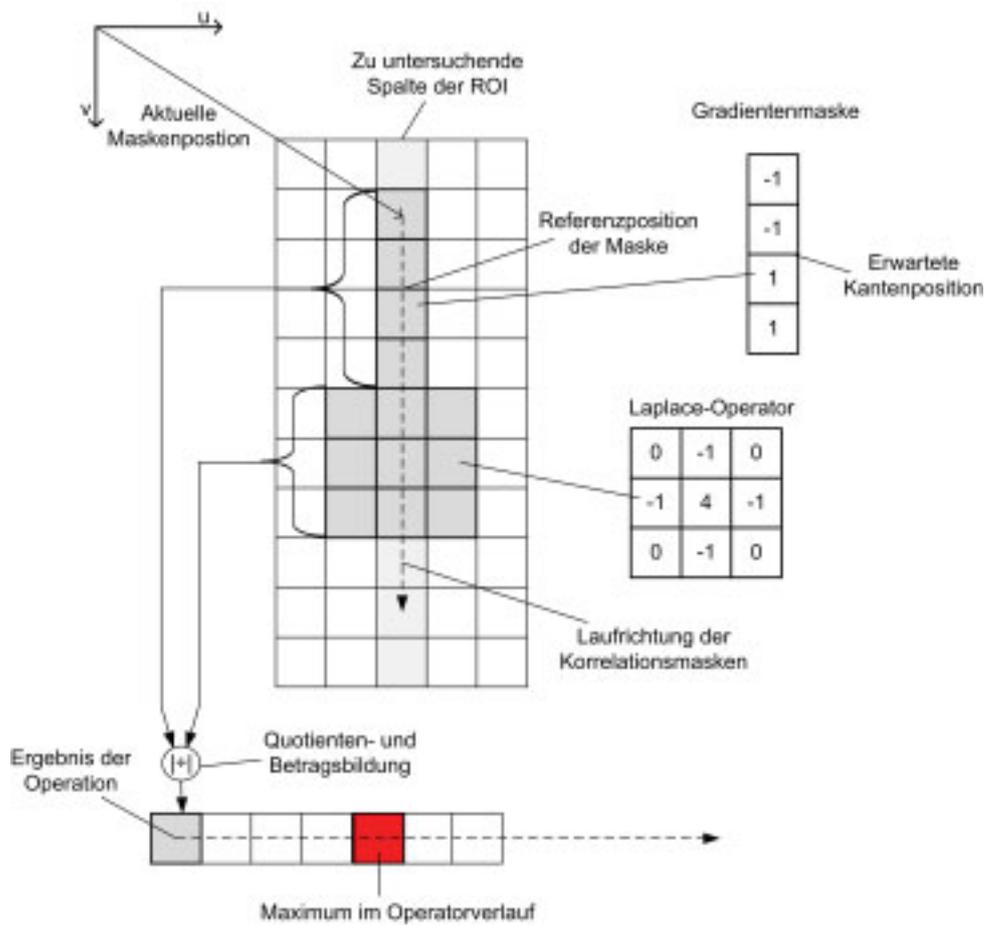


Bild 4.17: Schematische Darstellung der Vorgehensweise zur Detektion der Fahrzeugoberkante bei Nacht

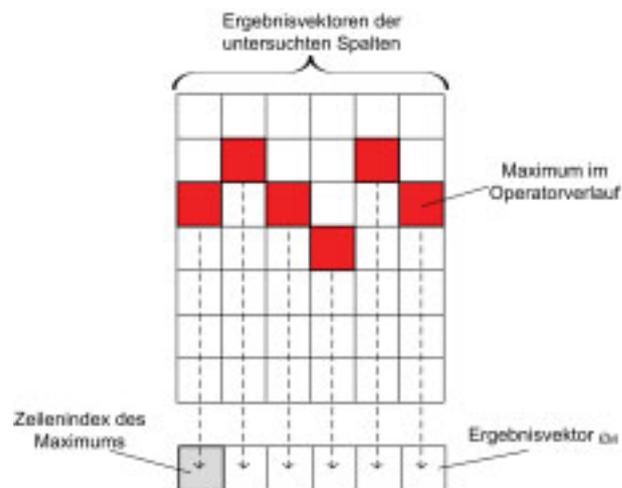


Bild 4.18: Mediansuche im Zeilenindexvektor

4.4 Modellbasierte Objekterkennung

Modelle spielen in der Datenverarbeitung im Allgemeinen eine große Rolle. Die Aussage: „Da ist ein PKW.“ beruht auf der Tatsache, dass in den Daten des verwendeten Sensorsets eine Merkmalskonstellation gefunden werden kann, die auf die Eigenschaften der Modellvorstellung für ein Objekt der Klasse „PKW“ schließen lässt. Menschen sind sehr gut darin, verschiedenartige Objekte voneinander zu unterscheiden. Das liegt daran, dass sie von Geburt an ihre Umwelt in Kategorien einteilen und Modelle für verschiedene Objektklassen entwickeln[39]. Dabei nutzt auch der Mensch seine „Sensoren“, sprich Sinne, um Objekteigenschaften durch Sinnesmerkmale zu erkennen. Dass dabei ein Lernprozess und Modelle eine Rolle spielen, kann durch ein einfaches Beispiel verständlich gemacht werden: Wenn man verschiedene Menschen aus einem fremden Kulturkreis (z. B. Asien) von einander unterscheiden soll, fällt es einem viel schwerer als bei Menschen aus dem eigenen Kulturkreis. Das Modell für den eigenen Kulturkreis ist also hinsichtlich der wichtigsten Unterscheidungsmerkmale viel reifer. Deshalb ist es auch bei der maschinellen Objekterkennung wichtig, möglichst aussagekräftige, also klassenspezifische Merkmale zu generieren. Dafür müssen Modelle zum Einsatz kommen, die das Wissen über die physikalischen Eigenschaften des zu klassifizierenden Objektes, im folgenden *Objektmodell* genannt, widerspiegeln. Einige Kategorien für diese Eigenschaften wären beispielsweise: die Form, die Farbe, der Geruch, das Material, usw. eines Objektes. Durch den Einsatz von *Sensormodellen* (siehe 3.2) können daraus klassenspezifische Merkmale abgeleitet werden, welche aus den Sensordaten extrahiert werden können. Je nach Messverfahren eines Sensors wirken sich bestimmte Objekteigenschaften mehr oder weniger stark in den Messwerten des Sensors aus.

Die Objektform spielt bei der Objekterkennung eine außerordentlich wichtige Rolle. Das liegt daran, dass alle sichtbaren Merkmale auf den nach außen weisenden Oberflächen und Kanten der Objekte zu finden sind. Die verwendeten Modelle der Objektformen für Fahrzeuge und Fahrspurmarkierungen werden in Abschnitt 4.4.1 beschrieben. Diese sind in dieser Arbeit aus früheren Arbeiten übernommen[7][48][70] und werden lediglich zum besseren Verständnis der nachfolgenden Teile beschrieben.

Bisher wird in dieser Arbeit nur die Detektion von Objekten zu einem bestimmten Zeitpunkt behandelt. Da Objekte in ihrer Erscheinungsform, sowie der relativen Position zum Beobachter eine ausgeprägte Kontinuität aufweisen, macht es Sinn eine weitere Art von Modell einzuführen. Diese sollen die zeitliche Entwicklung der Objekteigenschaften, sowie die relative Positionsänderung für ein begrenztes Zeitintervall möglichst gut approximieren. Das zusätzliche Vorwissen, das diese so genannten *Dynamikmodelle* einbringen, kann dazu genutzt werden den Suchraum bei zeitlich aufeinander folgenden Detektionen für dasselbe Objekt einzuschränken. Dadurch wird der Übergang von der Einzeldetektion hin zum Verfolgen (Tracken) von Objekten möglich. Durch die zeitliche Zuordnung von Messwerten können zusätzlich zu den direkt messbaren Objektzuständen (z. B. Position) auch implizit beobachtbare Zustände des Objektes (z. B. Geschwindigkeit) geschätzt werden. Die verwendeten Dynamikmodelle für die Fahrspur- und Fahrzeugverfolgung werden im Abschnitt 4.4.2 erläutert. Auch diese sind im Wesentlichen aus den oben erwähnten

Arbeiten übernommen, wobei in einigen Teilen eine Anpassung an neue Problemstellungen vorgenommen wurde (siehe Kapitel 7.1). Die rekursive Vorgehensweise zur Schätzung der Zustandsgrößen der Modelle wird in Abschnitt 4.4.3 behandelt.

4.4.1 Modelle der Objektgestalt

Die Abbildungen 4.20 und 4.19 zeigen die 3D-Formmodelle, wie sie im Fahrspur- und Fahrzeugtracker verwendet werden. Zunächst sind diese Modelle lediglich sehr vereinfachte Drahtgittermodelle mit einer für die verfolgten Objekte typischen Form.

Die Modelle sind aus P Punkten, K Kanten und F Flächen aufgebaut. Der Punkt \underline{P}_p , die Kante K_k und die Fläche F_f werden durch deren Indices $p \in \{0, \dots, P - 1\}$, $k \in \{0, \dots, K - 1\}$ und $f \in \{0, \dots, F - 1\}$ beschrieben. Ein Punkt \underline{P} ist ein Vektor, der im lokalen Koordinatensystem des jeweiligen Formmodells beschrieben ist, und stellt ein elementares Bauteil des Formmodells dar. Kanten verbinden mindestens zwei beliebige Punkte zu einem allgemeinen Polygonzug. In der entstehenden hierarchischen Struktur aus Punkten, Kanten und Flächen bezeichnet $\underline{P}_p^{(k)}$ den p -ten Punkt der k -ten Kante und $K_k^{(f)}$ die k -te Kante der f -ten Fläche. Der Normalenvektor $\underline{n}^{(f)}$ der f -ten Fläche wird durch das Kreuzprodukt der Vektoren zwischen Anfangs- und Endpunkt der beiden ersten Kanten berechnet.

$$\underline{n}^{(f)} = (\underline{P}_{P-1}^{(0)} - \underline{P}_0^{(0)}) \times (\underline{P}_{P-1}^{(1)} - \underline{P}_0^{(1)})$$

Diese parametrisierbaren Drahtgittermodelle stellen die Grundlage für die Sichtbarkeits- und Verdeckungsüberprüfung³⁾ aus Sicht der Sensoren dar. Des Weiteren können die Formmodelle als „Träger“ für sensor- und objektspezifische Merkmale genutzt werden und bilden somit das Gerüst für die merkmalsbasierte Objektverfolgung. Im Folgenden werden die Formmodelle für Fahrzeuge und die Fahrspur detaillierter beschrieben. Welche Merkmale in welcher Weise in das Modell integriert sind, wird in Kapitel 7 erläutert.

Fahrzeugmodell

Das Formmodell des Fahrzeugtrackers ist aus 12 Modellpunkten aufgebaut, welche durch 20 Kanten so miteinander verbunden sind, dass ein quaderähnliches Drahtgittermodell mit 10 nach außen weisenden Flächen entsteht. Der Ursprungspunkt liegt, wie in Abbildung 4.19 zu sehen ist, in der geometrischen Mitte der Bodenebene der Fahrzeugmodells.

Das Modell lässt sich über die Größen Höhe H , Breite B und Länge L wie ein Quadermodell parametrisieren. Um reale Fahrzeugformen besser annähern zu können ist zusätzlich zur Boden- und Dachebene eine horizontale Zwischenebene in das Modell integriert. Dadurch lässt sich die nach oben hin schmaler werdende Form von PKWs gut nachbilden.

³⁾ Auf diese Überprüfungen wird nicht gesondert eingegangen, da sie Standardverfahren aus der Computergrafik darstellen und in anderen Arbeiten bereits beschrieben sind[48].

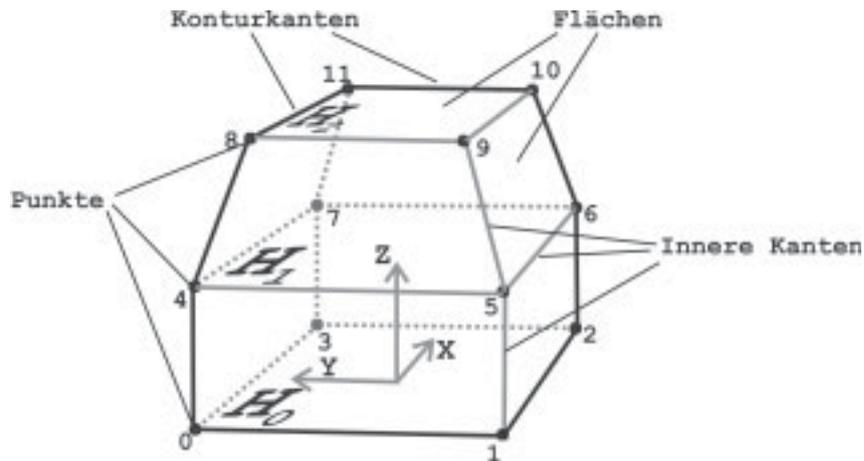


Bild 4.19: Darstellung des Formmodells des Fahrzeugtrackers

Die Punkte des Modells erhält man durch Variieren der Vorzeichen in der X- und Y-Komponente und der Variablen e im Vektor \underline{P}_F .

$$\underline{P}_F = \begin{pmatrix} \pm \xi_{X,e} \cdot L/2 \\ \pm \xi_{Y,e} \cdot B/2 \\ e \cdot H/2 \end{pmatrix} \quad \text{mit } e \in \{0, 1, 2\} \quad (4.38)$$

Der Parameter e ist der Index der horizontalen Ebene H_e des Fahrzeugmodells (siehe Abbildung 4.19). Im abgebildeten Modell sind die Parameter $\xi_{X,2} = 0.85$ und $\xi_{Y,2} = 0.70$ gesetzt.

Fahrspurmodell

Das Fahrspurmodell ist für eine einzelne Fahrspur dadurch charakterisiert, dass es durch zwei parallel zueinander verlaufende Polygonzüge begrenzt ist. Diese repräsentieren die linke bzw. rechte Begrenzung einer Fahrspur. Die Stützpunkte der beiden Polygonzüge werden symmetrisch zu einer virtuellen, mittleren Skelettlinie gesetzt. Das für die Skelettlinie zugrunde liegende mathematische Konstrukt ist ein Klothoidensegment, die nach Gleichung 4.39 definiert ist.

$$c(l) = c_0 + l \cdot c_1 \quad (4.39)$$

c_0 ist dabei die Krümmung der Klothoide im Ursprungspunkt des lokalen Modellkoordinatensystems und c_1 der Änderungskoeffizient der Krümmung in Abhängigkeit von der Lauflänge l entlang des Klothoidensegments. Die Krümmung c_0 ist nach Gleichung 4.40 über den Ursprungsradius r_0 definiert und hat die Einheit $1/m$.

$$c_0 = \frac{1}{r_0} \quad (4.40)$$

Gleichung 4.41 zeigt die Beziehung des Parameters c_1 zum so genannten Klothoidenparameter A .

$$c_1 = \frac{1}{A^2} \quad (4.41)$$

Klothoiden beschreiben also spiralförmig enger oder weiter werdende Kurvensegmente, wobei ein regulärer Kreis den Sonderfall einer Klothoide mit $c_0 \neq 0$ und $c_1 = 0$ darstellt. Klothoiden finden beim Straßenbau auf allen Straßen Anwendung. Sie sollen laut Richtlinie für die Anlage von Straßen (RAS-L) [41]

- „...eine kontinuierliche Änderung der bei der Kurvenfahrt auftretenden Zentrifugalbeschleunigung ermöglichen“ ,
- „als Übergangsegmente für die Fahrbahnverwindung dienen“ und
- „eine optisch befriedigende Linienführung bewirken“ .

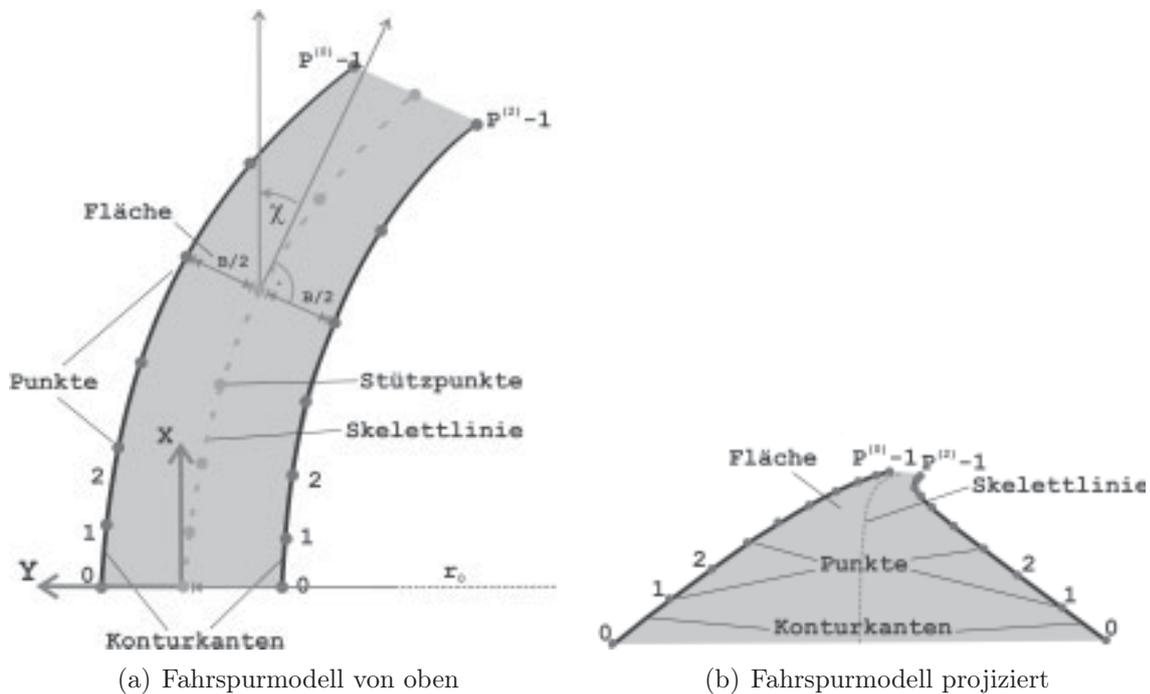


Bild 4.20: Darstellung des Formmodells des Fahrspurtrackers

Nach [69] berechnet sich der Bahnazimut-Winkel $\chi(l)$ zwischen der Tangente der Skelettlinie in einem bestimmten Punkt und der X-Achse des lokalen Modellkoordinatensystems KOS_S zu

$$\chi(l) = \int_0^l c(\lambda) d\lambda = l \cdot c_0 + \frac{l^2}{2} \cdot c_1 \quad (4.42)$$

Für kleine Winkel können daraus die Komponenten $X(l)$ und $Y(l)$ eines Punktes mit der Lauflänge l entlang der Klothoide mit den Näherungsformeln 4.43 und 4.44 angegeben werden.

$$X(l) = \int_0^l \cos(\chi(\lambda)) d\lambda \approx \int_0^l 1 d\lambda = l \quad (4.43)$$

$$Y(l) = \int_0^l \sin(\chi(\lambda)) d\lambda \approx \int_0^l \chi(\lambda) \lambda = \frac{l^2}{2} \cdot c_0 + \frac{l^3}{6} \cdot c_1 \quad (4.44)$$

Die Position des Modellpunktes $\underline{P}_{l/r,p}$ der linken bzw. rechten Konturkante wird unter Verwendung des p -ten Stützpunktes in der Lauflängentfernung l_p wie folgt berechnet:

$$\underline{P}_{l,p} = \begin{pmatrix} X(l_p) - \sin(\chi(l_p)) \cdot B(l_p)/2 \\ Y(l_p) + \cos(\chi(l_p)) \cdot B(l_p)/2 \\ 0 \end{pmatrix} \quad (4.45)$$

$$\underline{P}_{r,p} = \begin{pmatrix} X(l_p) + \sin(\chi(l_p)) \cdot B(l_p)/2 \\ Y(l_p) - \cos(\chi(l_p)) \cdot B(l_p)/2 \\ 0 \end{pmatrix} \quad (4.46)$$

Ausgehend von den Stützpunkten auf der Skelettlinie werden die Positionen der Modellpunkte unter Berücksichtigung des Bahnazimut-Winkels χ im Abstand $B(l_p)/2$, was der Hälfte der Breite der Fahrspur entspricht, zur Skelettlinie gesetzt.

4.4.2 Modelle der Objektdynamik

Das Verhalten dynamischer Prozesse über die Zeit kann durch das Aufstellen einer Differentialgleichung (DGL) modelliert werden. Eine DGL erster Ordnung stellt die erste Ableitung $\dot{x}(t)$ einer Variable in Abhängigkeit von $x(t)$ dar. Dies lässt sich auf physikalische Systeme mit mehreren Zustandsvariablen, die in dem Zustandsvektor $\underline{x}(t)$ zusammengefasst sind, erweitern.

$$\dot{\underline{x}}(t) = f(\underline{x}(t)) \quad (4.47)$$

Die DGL wird häufig als *Dynamikmodell* des zu beschreibenden Prozesses bezeichnet. Ziel eines Dynamikmodells ist es, auf Basis des aktuellen Zustands $\underline{x}(t_0)$, eine möglichst gute Vorhersage (Prädiktion) $\underline{x}^*(t)$ über den Zustand eines Systems zu einem in der Zukunft liegenden Zeitpunkt $t > t_0$ zu treffen. Beim Verfolgen von Objekten spielen Dynamikmodelle eine entscheidende Rolle im Hinblick auf die Robustheit und die Genauigkeit

der Zustandsschätzgrößen $\hat{x}(t)$ des Trackers. Je besser die Prädiktion $\underline{x}^*(t)$ mit der Realität übereinstimmt, umso kleiner wird das Fehlerrisiko bei der Assoziation gemessener Merkmale mit den erwarteten.

In diesem Abschnitt werden die verwendeten Dynamikmodelle in Form von zeitkontinuierlichen Differenzialgleichungssystemen für den Fahrzeug- und Fahrspurtracker beschrieben. Die mathematischen Schritte, die beim Übergang zu einer zeitdiskreten Formulierung nötig sind, werden im Anhang F aufgezeigt.

Fahrzeugmodell

Das unterlegte Dynamikmodell des Fahrzeugtrackers besteht aus drei Teilmodellen. Das Teilmodell „Fahrzeuggröße“ schätzt die Parameter Länge L , Breite B und Höhe H des Fahrzeugformmodells. Die Teilmodelle „Fahrzeuglängsbewegung“ und „Fahrzeugquerbewegung“ schätzen die Komponenten des Bewegungsvektors des Fahrzeugs im eigenen Umfeldkoordinatensystem. Dabei ist kein komplexes Fahrdynamikmodell hinterlegt, sondern die Längs- und Querdynamik entkoppelt. So können die Bewegungsgleichungen für bewegte Massenpunkte verwendet werden.

Im Prädiktionsschritt des Kalmanfilters wird die geschätzte Positionsänderung des Fremdfahrzeugs seit der letzten Prädiktion in die Transformationsmatrix eingetragen, die die relative Lage des Fremdfahrzeugs im Umfeldkoordinatensystem beschreibt. Auf die gleiche Weise kann auch der relative Gierwinkel $\psi_F(t)$ des Fremdfahrzeugs eingetragen werden. Da dieser nur sehr schlecht aus den vorliegenden Sensordaten direkt gemessen werden kann, wird auf ein eigenes Dynamikmodell dafür verzichtet. Es sind jedoch zwei Varianten implementiert, um den Gierwinkel zu berücksichtigen. Dazu kann zum einen die Ausrichtung direkt aus den geschätzten Werten der Längs- und Quergeschwindigkeit $v_l(t)$ und $v_q(t)$ abgelesen werden. Die zweite Variante kommt durch Kopplung mit dem Spurmodell zustande. Dazu wird der Bahnazimutwinkel $\chi(l)$ des Fahrspurmodells in der Entfernung des Fremdfahrzeugs direkt übernommen.

Fahrzeuggröße Die Form von Fahrzeugen, wie auch von anderen Objekten im Fahrzeugumfeld, kann in der Regel als starr angenommen werden. Da sich die Formen verschiedener Fahrzeuge jedoch sehr wohl von einander unterscheiden, müssen die Parameter L_F , B_F und H_F des Formmodells während der Initialisierungsphase des Fahrzeugtrackers noch einen gewissen Variationsspielraum besitzen, um sich an die aktuellen Gegebenheiten anpassen zu können. Wegen der angenommenen Starrheit der getrackten Fahrzeuge sagt das Dynamikmodell der Fahrzeuggrößenschätzung lediglich aus, dass der prädizierte Wert gleich dem alten sein soll. Daraus ergibt sich das folgende triviale Differenzialgleichungssystem:

$$\begin{pmatrix} \dot{B}_F(t) \\ \dot{H}_F(t) \\ \dot{L}_F(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} B_F(t) \\ H_F(t) \\ L_F(t) \end{pmatrix}. \quad (4.48)$$

Fahrzeuginnenbewegung Dieses Teildynamikmodell dient zur Prädiktion der Fahrzeugbewegung entlang der \underline{X}_F -Achse. Dabei wird, wie eingangs dieses Abschnittes erwähnt, die Längs- und Querdynamik des Fahrzeugs entkoppelt betrachtet. Zur Modellierung der Längsbewegung des Fahrzeugs kommt die Differenzialgleichung für ein Masseilchen unter Einfluss einer konstanten Kraft zum Einsatz. Diese kann über das zweite Newton'sche Gesetz ($\underline{F} = m \cdot \underline{a}$) hergeleitet werden und hat die Form

$$\underline{X}_1(t) = \frac{1}{2} \cdot \underline{a}_1(t) \cdot t^2 + \underline{v}_1(t) \cdot t + \underline{X}_0(t). \quad (4.49)$$

Daraus ergibt sich das äquivalente DGL-System erster Ordnung zu

$$\begin{pmatrix} \dot{X}_1(t) \\ \dot{v}_1(t) \\ \dot{a}_1(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & t \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_1(t) \\ v_1(t) \\ a_1(t) \end{pmatrix}. \quad (4.50)$$

Hierbei wird auf die Vektorschreibweise verzichtet, da die Richtung aller Vektoren in diesem Modell entlang der \underline{X}_F -Achse zeigen.

Fahrzeugquerbewegung Dieses Teildynamikmodell dient zur Prädiktion der Fahrzeugbewegung entlang der \underline{Y}_F -Achse. Diese Bewegung wird ähnlich zur Fahrzeuginnenbewegung modelliert, mit dem Unterschied, dass die Annahme einer konstanten Geschwindigkeit getroffen wird, d. h. $\underline{a}_q(t) = 0$.

$$\underline{Y}_q(t) = \underline{v}_q(t) \cdot t + \underline{Y}_0(t). \quad (4.51)$$

Daraus ergibt sich das äquivalente DGL-System erster Ordnung zu

$$\begin{pmatrix} \dot{Y}_q(t) \\ \dot{v}_q(t) \end{pmatrix} = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} Y_q(t) \\ v_q(t) \end{pmatrix}. \quad (4.52)$$

Auch hier wird auf die Vektorschreibweise verzichtet, da die Richtung aller Vektoren in diesem Modell entlang der \underline{Y}_F -Achse zeigen.

Fahrspurmodell

Die Gleichungen sind stark an die Ergebnisse der Arbeiten von Dickmanns[36][35][37] angelehnt. Die Herleitung der Gleichungen kann zu großen Teilen in der Dissertation von Mysliwetz nachvollzogen werden[69]. Das Dynamikmodell der Fahrspur ist aus drei unabhängigen Teilmodellen aufgebaut, deren DGLs im Folgenden beschrieben werden. Als Besonderheit muss hierbei beachtet werden, dass sich die Fahrbahn in Wirklichkeit nicht bewegt. Trotzdem ändern sich die Fahrspurbegrenzungeigenschaften relativ zum Fahrzeug dadurch, dass sich das Fahrzeug über die Fahrbahn bewegt. Diese relativen Änderungen werden im Dynamikmodell des Fahrspurtrackers modelliert.

Horizontale Krümmung Dieses Teilmodell führt hauptsächlich eine örtliche Prädiktion des horizontalen Krümmungsparameters c_{0h} durch. Da sich beim Entlangfahren der Straße deren Krümmung unmittelbar auf den zukünftigen Gierwinkel ψ des Fahrzeugs zur Fahrspurtangente und dadurch wiederum auf die Ablage Δy des Fahrzeugs zur Mitte der Fahrspur auswirkt, ist es sinnvoll diese Größen zusammen mit den Größen für den Krümmungsverlauf der Klothoide zu schätzen. Deshalb ist dieses Dynamikmodell mit dem aus der Fahrdynamik bekannten „Einspurmodell“ gekoppelt. Auf eine detaillierte Erklärung des Einspurmodells wird in dieser Arbeit verzichtet und auf einschlägige Literatur verwiesen[53][38]. Zur Darstellung der Änderung des Parameters c_{1h} über die Zeit, müssen wegen auftretender Unstetigkeiten an Übergängen angrenzender Klothoidensegmente die Größen c_{0hm} und c_{1hm} eingeführt werden. Diese stellen nach Mysliwetz[69] eine gute Näherung für die realen Werte c_{0h} und c_{1h} dar. Das Gleichungssystem 4.53 beschreibt das entstehende Gesamtsystem.

$$\begin{pmatrix} \dot{\beta}(t) \\ \Delta \dot{y}(t) \\ \dot{\psi}_G(t) \\ \dot{c}_{0hm}(t) \\ \dot{c}_{1hm}(t) \\ \dot{c}_{1h}(t) \\ \dot{\psi}_{off}(t) \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} \beta(t) \\ \Delta y(t) \\ \psi_G(t) \\ c_{0hm}(t) \\ c_{1hm}(t) \\ c_{1h}(t) \\ \psi_{off}(t) \end{pmatrix} + \underline{b} \cdot u_\lambda \quad \text{mit} \quad (4.53)$$

$$\mathbf{A} = \left(\begin{array}{ccc|ccc|c} a_F & 0 & 0 & 0 & 0 & 0 & 0 \\ v(t) & 0 & v(t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -v(t) & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & v(t) & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{3v(t)}{L_{max}(t)} & \frac{3v(t)}{L_{max}(t)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right),$$

$$\underline{b} = \begin{pmatrix} b_F \\ 0 \\ c_F \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad u_\lambda = \lambda(t),$$

$$a_F = -2 \frac{k_r}{m_{Fzg} \cdot v(t)}, \quad b_F = -\frac{k_r}{m_{Fzg}} + c_F \quad \text{und} \quad c_F = \frac{v(t)}{a_{Achse}}$$

In Tabelle 4.3 sind die auftretenden Variablen des Gleichungssystems genauer beschrieben.

Das Originalmodell wurde um den geschätzten Zustand $\dot{\psi}_{off}(t)$, mit $\dot{\psi}_G(t) = \dot{\psi}(t) - \dot{\psi}_{off}(t)$, erweitert. Da der aktuelle Lenkwinkel $\lambda(t)$ als Steuergröße implizit über die Gierrate

Variable	Kurzbeschreibung	Einheit
$\beta(t)$	Schwimmwinkel	<i>rad</i>
$\Delta y(t)$	Ablage zur Fahrspurmitte	<i>m</i>
$\psi_G(t)$	Gierwinkel zur Fahrbahntangente	<i>rad</i>
$c_{0hm}(t)$	gemittelte horizontale Krümmung	$1/m$
$c_{1hm}(t)$	gemittelte horizontale Krümmungsänderung	$1/m^2$
$c_{1h}(t)$	horizontale Krümmungsänderung	$1/m^2$
$\dot{\psi}_{off}(t)$	Gierratenoffset	<i>rad/s</i>
$\lambda(t)$	Lenkwinkel am Rad	<i>rad</i>
$v(t)$	Betrag des Fahrzeuggeschwindigkeitsvektors	<i>m/s</i>
$L_{max}(t)$	maximale Vorausschau	<i>m</i>
k_r	Seitenkraftbeiwert des Reifens	<i>N/rad</i>
m_{Fzg}	Fahrzeugmasse	<i>kg</i>
a_{Achs}	Achsabstand	<i>m</i>

Tabelle 4.3: Beschreibung der Variablen des Krümmungsmodells

$\dot{\psi}(t)$, mit $\lambda(t) = \psi(t) * a_{Achs}/v$, berechnet wird, übertragen sich die Ungenauigkeiten des Gierratensensors im Prädiktionsschritt auf die Zustandsschätzgrößen. Deshalb ist es sinnvoll den Offset des Sensors mit vergleichsweise niedriger Dynamik mitzuschätzen und in der Prädiktion zu berücksichtigen.

Nickwinkel des Fahrzeugaufbaus Dieses Teilmodell ist zur Kompensation des Nickwinkels $\theta(t)$ des Fahrzeugaufbaus gegenüber der Bodenebene gedacht. Nickwinkel treten bei Fahrten im Straßenverkehr ständig, z. B. durch Unebenheiten in der Fahrbahnoberfläche oder durch Beschleunigen des Fahrzeugs, auf. Dieser Winkel spielt für Sensoren mit vertikaler Winkelauflösung eine sehr wichtige Rolle, da der Nickwinkel des Fahrzeugs durch die starre Kopplung mit den Sensoren einen direkten Einfluss auf die Ausrichtung eines Sensors gegenüber dem Fahrzeugumfeld hat. Dadurch variiert die Blickrichtung des Sensors und somit auch die Bedeutung der Daten, die er ausgibt[49]. Um die Daten richtig interpretieren zu können, muss daher der relative Nickwinkel gegenüber der Bodenebene zum Zeitpunkt der Datenaufnahme möglichst genau bekannt sein.

Vor allem bei Videosensoren wirkt sich ein falscher angenommener Nickwinkel stark aus. Das liegt zum einen an der hohen vertikalen Winkelauflösung und zum anderen daran, dass der Sensor selbst keine Entfernungsmessung durchführen kann. Die Entfernungsmessung beruht häufig auf der Bodenebenenannahme, wie in Abschnitt 3.2.4 beschrieben. Bei der Rückprojektion eines gefundenen Merkmals, das als in der Bodenebene liegend angenommen ist, führt ein falscher angenommener Nickwinkel bei der Transformation des Sichtstrahls in das Umfeldkoordinatensystem zu einer falschen Sichtstrahlrichtung. Dies führt wiederum zu einem falschen Schnittpunkt mit der angenommenen Bodenebene. Abbildung 4.21 zeigt für die verwendete Kamera die Abhängigkeit der X-Komponente des Bodenschnittpunktes eines Sichtstrahls vom Nickwinkel. Dabei wird die Annahme getrof-

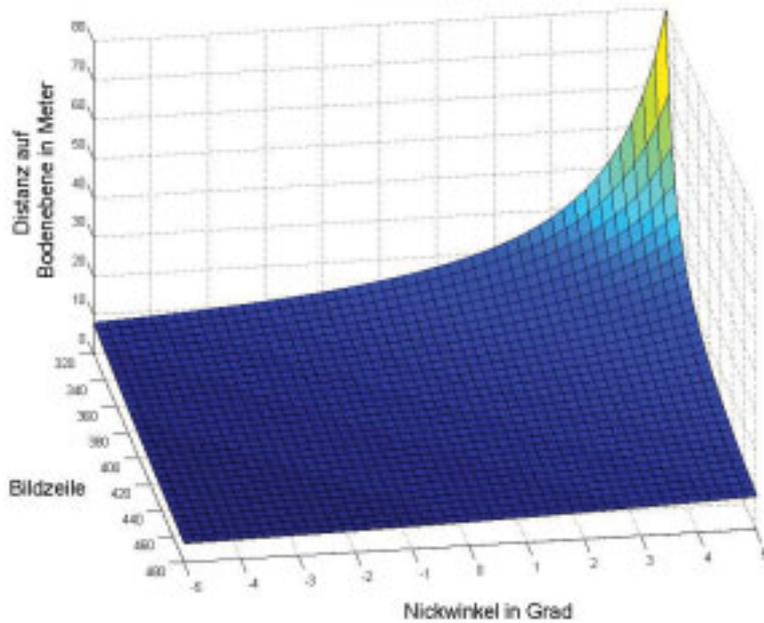


Bild 4.21: Gemessene Entfernungen des Videosensors in Abhängigkeit des Nickwinkels

fen, dass die X-Achse der Kamera in Ruhelage des Fahrzeugs parallel zur Bodenebene zeigt.

Die hohe vertikale Winkelauflösung der videobasierten Erkennung kann im Fall der Spurerkennung als Vorteil ausgenutzt werden. Durch das Wissen, dass Spurbegrenzungen ortsfeste Bezugspunkte darstellen, kann unter Verwendung der Bodenebenenannahme und der Parallelitätsannahme für Begrenzungsmarkierungen (Fahrspurmarkierungen, Leitplanken, usw.) der Nickwinkel direkt aus den Bildmerkmalen geschätzt werden. Deshalb ist der Nickwinkelschätzer als Teilmodell des Spurtrackers implementiert.

Die Nickbewegung des Fahrzeugaufbaus kann als gedämpfter harmonischer Oszillator mit Anregung modelliert werden. Mathematisch kann das Verhalten vereinfacht über das Momentengleichgewicht im Schwerpunkt des Fahrzeugs mit der DGL 4.54 beschrieben werden.

$$\ddot{\theta}(t) = -2\gamma\dot{\theta}(t) - \omega^2\theta(t) + \frac{M_{\text{ext}}(t)}{\phi_Y} \quad (4.54)$$

Dabei ist ω_0 die ungedämpfte Eigenfrequenz oder Kreisfrequenz des oszillierenden Systems (bestimmt durch die Masse und die Federkonstanten der Dämpfer des Fahrzeugs), ϕ_Y das Trägheitsmoment des Fahrzeugaufbaus um die $\underline{Y}_{\text{Fzg}}$ -Achse im Fahrzeugschwerpunkt \underline{S} und γ die Dämpfungskonstante. Das Moment M_{ext} steuert die Anregung des Systems und wird durch Gleichung 4.55 unter Berücksichtigung der Beschleunigung $\ddot{x}(t)$, der Fahrzeugmasse m_{Fzg} und der Schwerpunkthöhe Z_S berechnet [57][40].

$$M_{\text{ext}}(t) = Z_S \cdot m_{\text{Fzg}} \cdot \ddot{x}(t). \quad (4.55)$$

Unter Berücksichtigung eines unbekanntes Nickwinkeloffsets θ_{off} , mit $\theta_N(t) = \theta(t) - \theta_{\text{off}}(t)$, ergibt sich nach [8] für das Gesamtsystem das DGL-System 4.56.

$$\begin{pmatrix} \dot{\theta}_N(t) \\ \ddot{\theta}_N(t) \\ \dot{\theta}_{\text{off}}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -\omega^2 & -2\gamma & \omega^2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta_N(t) \\ \dot{\theta}_N(t) \\ \theta_{\text{off}}(t) \end{pmatrix} (t) + \begin{pmatrix} 0 \\ (Z_S \cdot m_{\text{Fzg}} \cdot \ddot{x}(t))/\phi_Y \\ 0 \end{pmatrix} \quad (4.56)$$

Breite der Fahrspur Das dritte Teilmodell ist für die Prädiktion der Fahrspurbreiten-schätzung verantwortlich. Dabei wird, analog zum Krümmungsmodell, die Breite $B_0(t)$ der Fahrspur am Ursprung des Fahrspurkoordinatensystems sowie die Änderung der Fahrspurbreite über den Vorausschaubereich, der sich von einer minimalen Distanz L_{min} bis zu einer maximalen Distanz L_{max} erstreckt, mit dem Faktor $B_1(t)$ berücksichtigt.

$$B_S(t) = B_0(t) + l \cdot B_1(t) \quad \text{mit} \quad L_{\text{min}} \leq l \leq L_{\text{max}} \quad (4.57)$$

Ebenfalls analog zum Krümmungsmodell ergeben sich auch hier Unstetigkeiten bei der Differenzierung des Breitenänderungsparameter $B_1(t)$ nach der Zeit. Deshalb kommt auch hier ein über den Vorausschaubereich gemitteltes Modell zum Einsatz. Dieses wird durch die DGL 4.58 beschrieben.

$$\begin{pmatrix} \dot{B}_{0m} \\ \dot{B}_{1m} \\ \dot{B}_1 \end{pmatrix} = \begin{pmatrix} 0 & v & 0 \\ 0 & -\frac{v}{L_{\text{max}}} & \frac{v}{L_{\text{max}}} \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} B_{0m} \\ B_{1m} \\ B_1 \end{pmatrix} \quad (4.58)$$

B_{0m} und B_{1m} sind die über die Lauflänge gemittelten Näherungswerte für die gewünschten Schätzgrößen B_0 und B_1 .

4.4.3 Rekursive Zustandsschätzung nach dem 4D-Ansatz

Der 4D-Ansatz ist ein von E. D. Dickmanns erstmals 1987 beschriebenes Verfahren zum effizienten Tracken von Objekten mittels Bildverarbeitung [36][35]. Dabei kommen 3D-Formmodelle und Dynamikmodelle, wie die aus Kapitel 4.4.1 und 4.4.2, zum Einsatz. Die drei Raumdimensionen zusammen mit der Zeitkomponente der Dynamikmodelle stellen die vier Dimensionen dar. Der Kern des Verfahrens ist ein erweitertes Kalman-Filter, das auf Basis des a-priori Wissens zum diskreten Abtastzeitpunkt t_{k-1} , in Form des Zustandsschätzvektors $\hat{\mathbf{x}}_{k-1} \in \mathbb{R}^M$, unter Hinzunahme des Informationsgehaltes der aktuellen Messungen $\mathbf{z}_{\text{KF}} = (\mathbf{z}_{\text{KF},0}, \mathbf{z}_{\text{KF},1}, \dots, \mathbf{z}_{\text{KF},N-1}) \in \mathbb{R}^N$, Schätzwerte $\hat{\mathbf{x}}_k = (\hat{x}_{0|k}, \hat{x}_{1|k}, \dots, \hat{x}_{M-1|k})$

für die Zustände zum aktuellen Zeitpunkt t_k generiert. Die grundlegenden Rechenschritte eines erweiterten Kalman-Filters können im Anhang E nachvollzogen werden.

Im Folgenden werden die einzelnen Schritte, die zur Durchführung eines Rekursionsschrittes des 4D-Ansatzes nötig sind, beschrieben. Ausgangspunkt ist das Anliegen neuer Sensorrohdaten \underline{v} an einem Sensor zum Messzeitpunkt t_k :

1. **Prädiktion:** Der Vektor $\hat{\underline{x}}_{k-1}$ der Zustandsschätzwerte und die Schätzfehlerkovarianzmatrix $\hat{\mathbf{P}}_{k-1}$ zum Zeitpunkt t_{k-1} müssen mittels Dynamikmodell auf den Messzeitpunkt t_k prädiziert werden. Daraus entstehen der prädizierte Vektor \underline{x}_k^* sowie die prädizierte Schätzfehlerkovarianzmatrix \mathbf{P}_k^* .
2. **Merkmalerzeugung:** Auf Basis von \underline{x}_k^* werden unter Verwendung eines Formmodells und eines Sensormodells eine Erwartung für die zum Sensor passenden Merkmale des Objektes erzeugt. Daraus entsteht der prädizierte Merkmalsvektor $\underline{z}_{\text{KF}}^*$. Bei der Erzeugung von Merkmalen muss für jeden einzelnen Sensor die Messbarkeit des erwarteten Merkmals überprüft werden, da die Sensoren unterschiedliche Sichtrichtungen und Messprinzipien haben. Dabei können Verdeckungs- aber auch andere sensorspezifische Eigenschaften eine Rolle spielen. Diese müssen in den Sensormodellen abgebildet sein.
3. **Merkmalsextraktion:** Aus den Sensorrohdaten \underline{v}_k müssen die Merkmale $\underline{z}_{\text{KF}}$ gewonnen werden, die zu den prädizierten Merkmalen $\underline{z}_{\text{KF}}^*$ passen. In Ausnahmefällen entsprechen die Sensorrohdaten bereits dem erwarteten Merkmalstyp (z. B. Distanzmessung des Radarsensors kann zur Entfernungsschätzung direkt verwendet werden). Davon kann aber in der Regel nicht ausgegangen werden. Grundsätzlich muss zunächst eine Sensordatenvorverarbeitung stattfinden, um die erwarteten Merkmale in den Rohdaten zu finden, vorausgesetzt sie sind vorhanden. Für Videomerkmale kommen dabei ähnliche Verfahren wie die in Kapitel 4 beschriebenen zum Einsatz. In diesem Schritt hat das Prinzip des 4D-Ansatzes seinen großen Vorteil. Die Erwartung, die durch die prädizierten Merkmale erzeugt wird, erlaubt es den Suchbereich in den Sensorrohdaten lokal zu begrenzen. Es müssen also nicht alle Merkmale im Rohdatenvektor \underline{v}_k gefunden werden, sondern nur diejenigen, die sich im lokalen Umfeld der prädizierten Merkmale befinden. Dadurch kann die Rechenzeit für die Sensordatenvorverarbeitung minimal gehalten werden.
4. **Linearisierung der Messgleichung:** Für jedes prädizierte Merkmal $z_{\text{KF},n}^*$ muss nun mit Hilfe des Sensor- und Objektmodells der Einfluss einer Messabweichung $\Delta z_n = z_{\text{KF},n} - z_{\text{KF},n}^*$ auf die prädizierten Zustände $\Delta \underline{x}_m = \hat{\underline{x}}_{m|k} - \underline{x}_{m|k}^*$ des Objektmodells berechnet werden. Dieser Zusammenhang kann in Form einer Matrix beschrieben werden, die im Fall von nichtlinearen Beziehungen zwischen Mess- und Zustandsraum für jeden Zeitpunkt im aktuellen Arbeitspunkt neu berechnet werden muss (Jakobimatrix \mathbf{H}_k). Dies kann zum einen analytisch geschehen, in dem man die mathematische Beschreibung des Sensor- und Objektmodells linearisiert.

Eine andere Möglichkeit besteht in der numerischen Bestimmung der Jakobimatrix. Dabei werden die prädizierten Objektzustände $\underline{x}_{m|k}^*$ einzeln gezielt variiert, wodurch

die variierten Zustandsgrößen $\mathbf{x}'_{m|k}$ entstehen. Die Variation $\delta x_m = \mathbf{x}'_{m|k} - \mathbf{x}^*_{m|k}$ im Zustandsraum bewirkt eine Veränderung $\delta z_n = \mathbf{z}'_{\text{KF},n} - \mathbf{z}^*_{\text{KF},n}$ der prädizierten Merkmale. Die Änderung im Merkmalsraum kann nun ins Verhältnis zur verursachenden Änderung im Zustandsraum gesetzt werden. Dieses Verhältnis ist zugleich der Eintrag der Jakobimatrix \mathbf{H}_k und beschreibt den linearisierten Proportionalitätsfaktor $\frac{\delta z_n}{\delta x_m}$ zwischen den Messraumdifferenzen Δz_n und den Zustandsraumdifferenzen Δx_m . Die Jakobimatrix hat die Form

$$\mathbf{H}_k = \begin{pmatrix} \frac{\delta z_0}{\delta x_0} & \frac{\delta z_0}{\delta x_1} & \dots & \frac{\delta z_0}{\delta x_M} \\ \frac{\delta z_1}{\delta x_0} & \frac{\delta z_1}{\delta x_1} & \dots & \frac{\delta z_1}{\delta x_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta z_N}{\delta x_0} & \frac{\delta z_N}{\delta x_1} & \dots & \frac{\delta z_N}{\delta x_M} \end{pmatrix}. \quad (4.59)$$

Die Transformation zwischen Merkmals- und Zustandsraum geschieht mittels Gleichung 4.60.

$$\Delta \underline{z}_{\text{KF}} = \mathbf{H}_k \cdot \Delta \underline{x}. \quad (4.60)$$

5. **Assoziation:** Ziel der Assoziation ist es jedem prädizierten Merkmal $\mathbf{z}^*_{\text{KF},n}$ das beste dazu passende gemessene Merkmal $\mathbf{z}_{\text{KF},n}$ zuzuordnen. Für jedes prädizierte Merkmal kann unter Berücksichtigung der Unsicherheiten im Dynamik-, Sensor- und Messmodell aller beteiligten Komponenten die so genannte Innovationskovarianzmatrix \mathbf{V}_k^* im Messraum berechnet werden[49]. Deren Werte beschreiben die zu erwartende Varianz der Merkmale. Davon kann eine Schwelle für die maximale Abweichung der gültigen Messung von der Erwartung abgeleitet werden.

Von allen gültigen Merkmalen muss diejenige Merkmalskombination gewählt werden, die gesamtheitlich am besten zum prädizierten Modell passt. Dabei können auch die Daten mehrerer Sensoren gemeinsam betrachtet werden (Sensordatenfusion). Es muss also eine globale Assoziationsfunktion optimiert werden. Im einfachsten Fall wird für jedes prädizierte Merkmal eine Distanz zu jedem passenden gemessenen Merkmal im Gültigkeitsbereich berechnet, und der Messwert mit der kleinsten Distanz („nearest neighbor“) wird dem prädizierten Merkmal zugewiesen. Dabei ist darauf zu achten, dass diese Zuweisung eineindeutig sein muss. Kein gemessenes Merkmal darf mehreren prädizierten Merkmalen zugewiesen werden, und keinem prädizierten Merkmal dürfen mehrere gemessene Merkmale zugewiesen werden. Jedes prädizierte Merkmal, dem ein gemessenes Merkmal zugewiesen wurde, wird nun als gültiges Merkmal bezeichnet.

6. **Ermitteln der Residuen:** Für jedes gültige Merkmal wird nun die euklidische Distanz zwischen Prädiktion und Messung im Merkmalsraum berechnet. Diese Distanz wird auch als Residuum bezeichnet.
7. **Innovation / Kalman-Update:** Die Residuen Δz_n werden mittels Kalman-Verstärkungsfaktors \mathbf{K}_k in eine Zustandsänderung Δx_m umgerechnet. Die Zustandsänderung wird auf den prädizierten Zustand addiert, so dass der aktuelle

4 Modellbasierte Objekterkennung mittels Bildverarbeitung

Schätzwert $\hat{x}_{m|k} = x_{m|k}^* + \Delta x_m$ entsteht. Sind alle Zustände innoviert, entsteht der innovierte Zustandsvektor \hat{x}_k . Dieser stellt beim Eintreffen neuer Daten den Ausgangspunkt für den 1. Schritt dar.

Praxisrelevante Anmerkungen

Zu den Schritten 3. und 5. sei angemerkt, dass es je nach Anwendung und Vorwissen sinnvoll sein kann, mehr Aufwand in die Merkmalsextraktion zu stecken und dafür Aufwand in der Assoziation zu sparen oder umgekehrt. So könnte in einem Extremfall bereits in der Merkmalsextraktion aus den Messungen der einzelnen Sensoren sehr komplexe und abstrahierte Merkmale unter Hinzunahme einer Modellannahme gewonnen werden. Im Fall der Fahrzeugverfolgung würde das bedeuten, dass bereits hier eine Hypothese für ein Fahrzeugmodell erstellt wird. Bei der Assoziation müsste lediglich eine Zuordnung auf Zustandsebene erfolgen. Im anderen Extremfall werden die Sensorrohdaten gar nicht oder nur minimal vorverarbeitet. Dann bleibt dem Assoziationsschritt die Aufgabe die beste Kombinationsmöglichkeit der einzelnen Merkmale zu finden. Generell kann man sagen, dass es sinnvoller ist in die Merkmalsextraktion so viel Wissen wie möglich zu integrieren. Allerdings können die Möglichkeiten der Merkmalsextraktion aus Gründen der Effizienz, aus Mangel an Vorwissen oder wegen Synchronisationsschwierigkeiten bei Verwendung mehrerer Sensoren begrenzt sein.

So ist zu beachten, dass der Messzeitpunkt t_k in der Praxis nicht beliebig genau bekannt ist. Bei Kamerasystemen, zum Beispiel, sind die Daten zum Zeitpunkt $t_{k+} = t_k + \Delta t_B + \Delta t_L + n_L(t_k)$, zu dem sie im System verfügbar sind, schon mindestens so alt wie die Dauer Δt_B der Belichtung der Kamera. Dazu kommt noch, dass das Aussenden der Daten meist erst verzögert nach einer gewissen Latenzzeit Δt_L erfolgen kann. In Δt_L sind auch die Übertragungszeiten der Daten auf dem Weg vom Sensor zum System enthalten. Bei Nicht-Echtzeitsystemen, wie sie in dieser Arbeit eingesetzt werden (Betriebssystem WindowsXP[®]), sind die Systemübertragungszeiten zusätzlich mit einem Jitter $n_L(t_k)$ behaftet, der nur im Mittel bestimmt werden kann. Das heißt, im Prädiktionsschritt (Schritt 1.) muss eine Synchronisation der Sensordaten mit der Systemzeit erfolgen. Das geschieht meist dadurch, dass der für die Prädiktion relevante Messzeitpunkt über

$$t_k = t_{k+} - \Delta t_B - \Delta t_L - E(n_L(t))$$

berechnet wird. Eine der Wahrnehmung nachgeschaltete Funktion muss dann noch eine Prädiktion der innovierten Daten (Schritt 7.) mittels Dynamikmodell auf den Zeitpunkt t_{k+} durchführen, um möglichst aktuelle Daten zu erhalten. Diese Problematik ist bei allen Sensoren vorhanden und muss bei hohen Objektdynamiken, wie sie im Fahrerassistenzbereich vorhanden sind, berücksichtigt werden.

Beispiel:

Ein entgegenkommendes Fahrzeug auf einer Landstraße mit einer Geschwindigkeit von 100km/h erzeugt bei einer Eigengeschwindigkeit von ebenfalls 100km/h eine relative Distanzänderung von

$$\Delta X = \Delta t * 200\text{km/h} = \Delta t * 55\text{m/s}$$

Daraus ergibt sich ohne Berücksichtigung von Belichtungs- und Latenzzeit (unter Vernachlässigung des unbekanntes Jitters), für $\Delta t_B = 40\text{ms}$ und $\Delta t_L = 60\text{ms}$, eine falsche Positionsschätzung von

$$\Delta X = 100\text{ms} * 55\text{m/s} = 5.5\text{m}.$$

So ein Fehler in der Prädiktion wirkt sich unmittelbar auf die Qualität der Assoziation (Schritt 4.) und somit auch auf die letztendlich geschätzten Zustandsgrößen aus.

4.5 Zusammenfassung zur modellbasierten Objekterkennung

Dieses Kapitel umfasst die theoretischen Grundlagen zur modellbasierten Objekterkennung mittels Bildverarbeitung. Nachdem in Abschnitt 4.1 die grundlegenden Verfahren von Bildverarbeitungsoperatoren beschrieben sind, wird in Abschnitt 4.2 und 4.3 auf die wichtigsten in dieser Arbeit verwendeten Operatoren eingegangen. Während der Entwicklung der Module zur erstmaligen, rein videobasierten Objektdetektion für Tag- und Nachtsequenzen, die in Kapitel 6 beschrieben werden, sind einige Operatoren und Detektoren entstanden, die spezielle Objektmerkmale im Videobild extrahieren können:

Der *Gradientenhäufungsoperator* wird zur Auffindung einer Häufung horizontaler Strukturen verwendet, während der *Konturbewertungsoperator* eine Qualitätszahl dafür zurückliefert, wie gut ein Bildausschnitt zur Kontur einer Fahrzeugheckansicht passt. Um das Merkmal der Symmetrie noch robuster im Bild zu finden, ist der *Gradientensymmetrieoperator* entstanden, der einen Kennwert für die Symmetrie der Strukturen im Bild erzeugt. Die Information des Gradientenhäufungsoperators und des Gradientensymmetrieoperators lassen sich gut und effizient kombinieren, um einen Hinweis auf mögliche Fahrzeuge in der Umgebung zu bekommen (siehe Bild 6.8 in Kapitel 6.3.2). Der *Lichtdetektor* und der *Bodenreflexionsdetektor* sind im Zuge der Verfahren zur Fahrzeugerkennung bei Nacht (Kapitel 6.4) entstanden. Sie liefern Positionen markanter Merkmale für Fahrzeugheckansichten bei Nacht zurück.

Des Weiteren werden in Abschnitt 4.4 die hinterlegten Modelle zur Verfolgung von Fahrspuren und Fahrzeugen beschrieben und abschließend die grundlegenden Schritte zur Objektverfolgung mittels 4D-Ansatz nach Dickmanns[36] erläutert.

5 Klassifikation durch Maschinenlernverfahren

Der Term „Maschinenlernen“ wurde von dem Elektrotechnik-Ingenieur Arthur Samuel in den 50er Jahren geprägt, der als Erster ein selbst lernendes DAME-Computerspiel entwickelte[96]. Ursprünglich sollte der Begriff Maschinenlernen allgemein intelligentes Verhalten des Menschen, das auf eine Maschine übertragen wurde, bezeichnen. Tatsächlich beschäftigt sich der Bereich des Maschinenlernens seither hauptsächlich mit der Erkennung und Extraktion von Strukturen aus Daten, was nur einem sehr kleinen Ausschnitt intelligenten Verhaltens entspricht. Seit ihrem ersten Einsatz haben die Maschinenlernverfahren eine außergewöhnlich schnelle Weiterentwicklung erfahren[77]: Der erste große Durchbruch stellt das 1943 von McCulloch und Pitts entwickelte neuronale Netz dar. Dieses war ursprünglich biologisch motiviert. Das menschliche Gehirn umfasst etwa 10^{11} Neuronen, die über 10^{14} Verbindungen mit einer Gesamtlänge von ca. $5.8 \cdot 10^9$ Metern vernetzt sind[99]. Dieses Schema sollte durch ein künstliches Netz vieler einfacher, parallel arbeitender Übertragungsfunktionen imitiert werden. Die Weiterentwicklung dieser neuronalen Netze zum so genannten *Perceptron* markiert 1951 den nächsten Meilenstein[68]. Ein Perceptron ist in der Lage lineare Zusammenhänge nachzubilden und konnte bereits einfache Mustererkennungsaufgaben bewältigen. Aufbauend darauf entwickelten sich in den 80er Jahren die *Multi-Layer-Perceptronen* (MLP). Diese Form neuronaler Netze ist bei entsprechender Auslegung in der Lage alle möglichen mathematischen Zusammenhänge nachzubilden. Sie sind damit die ersten Stellvertreter der Gruppe universaler Approximatoren. In den 90er Jahren wurden die *Support-Vektor-Maschinen* entwickelt, die bis heute zu den mächtigsten Verfahren zählen. Erst um die Jahrtausendwende entstanden die neuesten Vorgehensweisen, die so genannten *Ensemble-Lernverfahren*, zu denen auch der hier eingesetzte Random-Forest-Klassifikator gehört.

Einige Grundlagen zu Maschinenlernverfahren können im Anhang A nachgelesen werden. Im Folgenden werden lediglich die verwendeten Verfahren – der Random-Forest-Klassifikator (RF-Klassifikator) in Kapitel 5.1 und generalisierte Radiale-Basis-Funktionen-Netze (GRBF-Netze) in Kapitel 5.2 – detaillierter vorgestellt.

5.1 Random-Forest-Klassifikator

Der RF-Klassifikator ist ein Ensemble-Lernverfahren, das auf dem Kombinieren der Entscheidung vieler Entscheidungsbäume basiert. Deshalb wird im Folgenden kurz die Funk-

tionsweise von Entscheidungsbäumen und Ensemble-Lernverfahren diskutiert. Für weiterführende theoretische Informationen sei auf die Grundlagenarbeit von Leo Breiman verwiesen[24].

5.1.1 Entscheidungsbäume

Entscheidungsbäume sind eine spezielle Darstellungsform einer komplexen Entscheidungsregel. Dabei werden Einzelentscheidungen hierarchisch angeordnet und dadurch eine Gesamtregel erzeugt. Alle Entscheidungsbäume setzen sich aus den Bestandteilen *Wurzelknoten*, *interne Knoten*, *Endknoten* und *Kanten* zusammen. Die Endknoten werden häufig auch als *Blätter* des Entscheidungsbaums bezeichnet. Abbildung 5.1 zeigt den schematischen Aufbau eines Entscheidungsbaums. Tabelle 5.1 gibt eine kurze Beschreibung seiner Bestandteile.

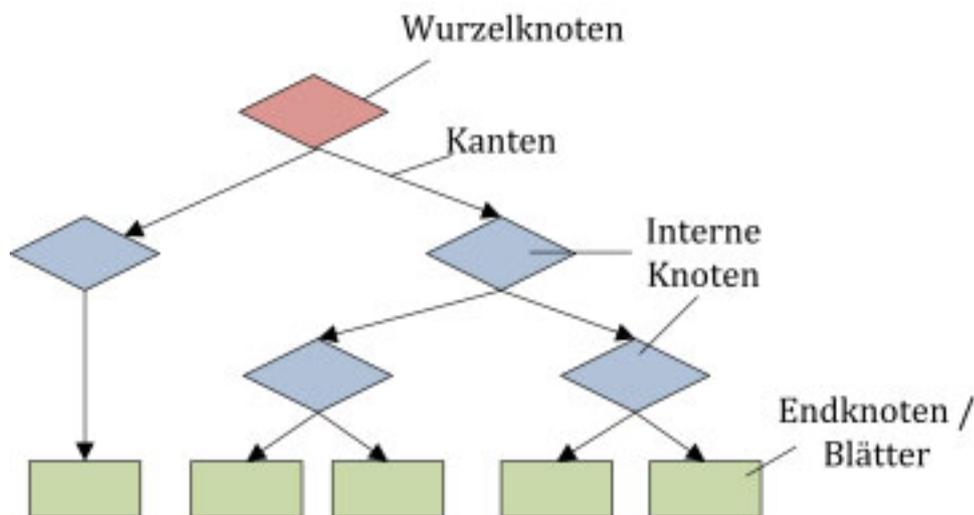


Bild 5.1: Entscheidungsbaum

An internen Knoten erfolgt ein Teilen der Daten in zwei oder mehr Untermengen. Entscheidungsbäume, bei denen jeder interne Knoten die eingehenden Daten in nur zwei Subkategorien einteilt, werden als *binäre Entscheidungsbäume* bezeichnet.

Erzeugen von Entscheidungsbäumen Die einfachste Möglichkeit einen Entscheidungsbaum zu erzeugen ist, einen Experten eine entsprechende „if... then... else“-Struktur entwerfen zu lassen. Für einfache, niederdimensionale Probleme weisen Expertensysteme in der Regel recht gute Ergebnisse auf. Sie sind aber immer daran gekoppelt, dass Experten für das entsprechende Gebiet verfügbar sind. Bei komplexen, hochdimensionalen Optimierungsaufgaben sind jedoch auch diese meist überfordert und können nicht für den minimal möglichen Klassifikationsfehler garantieren. Aus diesem Grund wurde eine ganze Reihe von Algorithmen entwickelt, mit deren Hilfe Entscheidungsbäume maschinell generiert

Bezeichnung	Beschreibung
<i>Wurzel</i>	Die Wurzel stellt den Startpunkt des Entscheidungsbaums dar. Hier werden neue Daten eingespeist.
<i>Interne Knoten</i>	Interne Knoten sind Entscheidungspunkte in einem Baum welche ein Aufsplitten der Daten bewirken und nicht Wurzel des Baums sind. Interne Knoten besitzen immer nachfolgende so genannte „Kinderknoten“ .
<i>Blätter</i>	Blätter sind eindeutig einer endgültigen Entscheidung (Klasse) zugeordnet. Gleichzeitig können mehrere Blätter der selben Klasse entsprechen. Blätter besitzen keine Kinderknoten.
<i>Kanten</i>	Eine Kante bezeichnet die Verbindung von einem Knoten zu seinem Kindknoten

Tabelle 5.1: Bestandteile eines Entscheidungsbaums

werden können. Als Ausgangspunkt erhalten solche Verfahren einen Satz von Trainingsdaten, bei denen für jedes Datenmuster zusätzlich die zugehörige Klasse gespeichert ist. Ein solches Datentupel wird als *gelabelt* bezeichnet. Die grundlegende Vorgehensweise eines solchen Verfahrens, sieht dabei folgendermaßen aus:

1. Schritt Wurzelknoten erzeugen.
2. Schritt Merkmalskombinationen mit Schwellwerten für die Aufteilung der Eingangsdaten wählen.
3. Schritt Eingangsdaten in Submengen (interne Knoten) aufteilen.
4. Schritt Abbruchbedingung für jede Submenge überprüfen.
5. Schritt Für jede Submenge, die die Abbruchbedingung erfüllt den 6. Schritt ausführen. Für jede verbleibende Submenge, wieder im 2. Schritt beginnen, wobei als Eingangsdaten die Submenge verwendet wird.
6. Schritt Weise den Endknoten der am häufigsten vorkommenden Klasse, unter den in diesem Knoten gelandeten Trainingsdaten zu.

Ein Beispiel für einen Algorithmus zum automatisierten Erzeugen von Entscheidungsbäumen ist der CART¹⁾-Algorithmus nach L. Breiman [24], welcher in der Lage ist, binäre Entscheidungsbäume zu generieren. Dabei wird in jedem Knoten t eine optimale Entscheidung für die Klassentrennung hinsichtlich einer Unreinheitsfunktion $i(t)$ in den Kinderknoten getroffen.

¹⁾ Classification And Regression Trees

Beim Finden des optimalen Splits $\hat{s}_{\text{opt}}(t)$ der im Knoten t enthaltenen Datenmuster, zum Aufspalten in die Kinderknoten t_L und t_R , wird eine Maximierung der *Abnahme der Unreinheitsfunktion* $\Delta i(\hat{s}, t)$ angestrebt. Dies kann mittels Gleichung 5.1 mathematisch formuliert werden.

$$\begin{aligned} \hat{s}_{\text{opt}}(t) &= \arg \max_{\hat{s}} \{\Delta i(\hat{s}, t)\} \\ \text{mit } \Delta i(\hat{s}, t) &= i(t) - \hat{p}(\underline{x} \in t_L | \underline{x} \in t) \cdot i(t_L) - \hat{p}(\underline{x} \in t_R | \underline{x} \in t) \cdot i(t_R), \end{aligned} \quad (5.1)$$

wobei $\hat{p}(\underline{x} \in t_L | \underline{x} \in t)$ und $\hat{p}(\underline{x} \in t_R | \underline{x} \in t)$ die bedingten Wahrscheinlichkeiten sind, dass ein Datenmuster \underline{x} , welches im Knoten t liegt, dem linken bzw. den rechten Kinderknoten zugeordnet wird. Diese sind wie folgt definiert:

$$\hat{p}(\underline{x} \in t_{L/R} | \underline{x} \in t) = \frac{\hat{p}(\underline{x} \in t_{L/R})}{\hat{p}(\underline{x} \in t)}. \quad (5.2)$$

Dabei sind $i(t_L)$ und $i(t_R)$ die verbleibenden Werte der Unreinheitsfunktion in den Kinderknoten nach dem Split.

Beim Erzeugen der Bäume im hier verwendeten RF-Klassifikator wird die geschätzte Unreinheitsfunktion $\hat{i}(t)$ verwendet:

$$\begin{aligned} \hat{i}(t) &= \sum_{k=1}^C \hat{p}(y = c_k | \underline{x} \in t) (1 - \hat{p}(y = c_k | \underline{x} \in t)) \\ \text{mit } \hat{p}(y = c_k | \underline{x} \in t) &= \frac{M_k(t)}{M(t)}. \end{aligned} \quad (5.3)$$

Die Herleitung dieser Funktion kann in [19] nachvollzogen werden. Dabei ist $M(t)$ die Anzahl aller Datenmuster, die zu Knoten t gehören und $M_k(t)$ die Anzahl der Datenmuster in der Untermenge, welche im Knoten t zur Klasse c_k gehören. Wenn alle Datenmuster in einem Knoten zur gleichen Klasse gehören, ist die Unreinheit mit $\hat{i}_{\text{min}}(t) = 0$ minimal. Maximal wird die Unreinheit, wenn alle Klassen in einem Knoten gleich häufig auftreten.

Als Eigenheit bei der Suche nach dem optimalen Split in den Knoten der Bäume eines RF-Klassifikators muss noch darauf hingewiesen werden, dass nicht alle N verfügbaren Features berücksichtigt werden. Es findet eine zufällige Auswahl von N_{RF} Features statt, mit $N_{\text{RF}} \ll N$.

Klassifikation mit Entscheidungsbäumen Neu zu klassifizierende Datenmuster werden an der Wurzel des Baums eingespeist und „fallen“ anschließend durch den Baum, bis sie schließlich in einem der Blätter landen. Das Datenmuster wird derjenigen Klasse zugeordnet, der das erreichte Blatt während der Trainingsphase zugewiesen wurde.

5.1.2 Ensemble-Lernverfahren

Der Gesamtfehler, den ein Klassifikator beim Lösen der Klassifikationsaufgabe macht, ergibt sich aus den Bestandteilen *Bias* und *Varianz* des zugrunde liegenden Maschinenlernverfahrens. Ensemble-Lernverfahren eignen sich besonders dafür den Varianzwert zu verringern, ohne dabei den Biaswert zu erhöhen, was in der Regel zu einem kleineren Gesamtfehler des erzeugten Klassifikators führt. Dazu kombinieren sie mehrere Teilklassifikatoren zu einem Gesamtklassifikator, was unter mit Begriff Ensemble-Lernverfahren bezeichnet wird. Der optimale Einsatz von Ensemble-Techniken ist Bestandteil aktueller Forschungen im Bereich der Klassifikation[20]. Es kann gezeigt werden, dass durch die Kombination vieler flexibler Basislernalgorithmen, wie Neuronale Netze oder Entscheidungsbäume, der niedrige Biaswert dieser Basislernalgorithmen erhalten bleibt, jedoch der Varianzwert stark sinkt. Das heißt zum einen, dass sich die Gefahr des Übertrainierens für den resultierenden Gesamtklassifikator verringert und gleichzeitig das Fehlerrisiko des Klassifikators sinkt. Im Anhang A wird nochmal auf die Bias-Varianz-Problematik eingegangen.

Ensemble-Lernverfahren werden in zwei Kategorien unterteilt: parallele und sequenzielle Verfahren. Sequenzielle Verfahren zeichnen sich dadurch aus, dass die Entwicklung eines Teilklassifikators von anderen Teilklassifikatoren abhängt. Zu den sequenziellen Methoden zählen Boosting- und Arcing²⁾-Techniken (ein bekanntes Verfahren ist der AdaBoost³⁾-Algorithmus). Ein sehr simples und zugleich mächtiges Werkzeug zur Verbesserung von Klassifikatoren stellt das parallel arbeitende Verfahren des Bagging (**Bootstrap Aggregating**) dar. Dabei werden mehrere gleichartige Klassifikatoren mit unterschiedlichen Datensätzen trainiert. Da die Größe des Trainingsdatensatzes meist begrenzt ist, wird hier zur Generierung der unterschiedlichen Datensätze die Methode des Bootstrapping genutzt. Dabei werden aus dem Gesamtdatensatz durch „Ziehen mit Zurücklegen“ Teildatensätze erzeugt, mit denen dann die Basislernverfahren trainiert werden. Als Klassifikationsergebnis wird die Mehrheitsentscheidung der einzelnen Klassifikatoren genommen. Einen guten Überblick über die genannten Themen mit weiterführenden Informationen lässt sich in [21], [21], [13], [77] und [93] erhalten.

Die positive Auswirkung von Ensemble-Lernverfahren ist bei Basislernverfahren mit hoher Varianz bezogen auf verschiedene Trainingsdatensätze besonders groß. Deshalb ist gerade die Kombination mit Entscheidungsbäumen sehr sinnvoll, da hier der Varianzwert im Allgemeinen sehr hoch ist, jedoch der Biaswert durch die hohe Flexibilität sehr klein. Der große Nachteil, den Ensemble-Techniken mit sich bringen, ist der Verlust der Interpretierbarkeit des Basislernverfahrens: Ist die Entscheidung eines einzelnen binären Entscheidungsbaums in der Regel noch nachvollziehbar, so lässt sich ein Ensemble aus hunderten Entscheidungsbäumen vom Menschen nicht mehr interpretieren.

²⁾ **Adaptive Reweighting and Combining**

³⁾ **Adaptive Boosting**

5.2 Generalisierte Radiale-Basis-Funktionen-Netze

Generalisierte Radiale-Basis-Funktionen-Netze (GRBF-Netze) gehören zur Kategorie der gewichteten linear-kombinierten Basisfunktionen. Die Netztopologie eines GRBF-Netzes ist in Abbildung 5.2 dargestellt.

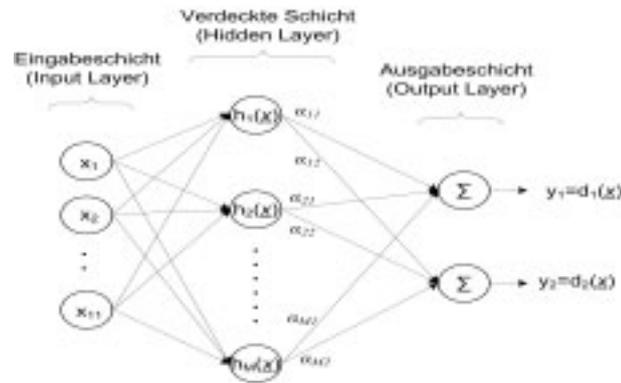


Bild 5.2: Topologie eines typischen GRBF-Netzes

Die Basisbestandteile der einzelnen Schichten werden als Neuronen bezeichnet. Der Aufbau eines einzelnen Neurons ist in Abbildung 5.3 dargestellt.

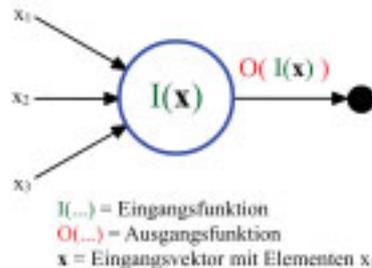


Bild 5.3: Allgemeiner Aufbau eines Neurons

Jedes Neuron besitzt eine bestimmte Transferfunktion $h(\underline{x})$, die sich durch das Zusammenspiel von Eingangsfunktion $I(\underline{x})$ und Ausgangsfunktion $O(I(\underline{x}))$ ergibt. Diese Funktion bestimmt, in welcher Weise das Neuron auf ein Eingangssignal reagiert. Die Transferfunktion ist somit verantwortlich für das Verhalten des gesamten Netzes. Sind alle Transferfunktionen lineare Funktionen, wird auch der Merkmalsraum mittels linearer Funktionen in Subräume unterteilt; liegen nichtlineare Übertragungsfunktionen vor, können komplexere Grenzen realisiert werden.

Bei GRBF-Netzen entspricht die Anzahl der Neuronen in der Eingabeschicht der Dimension des Eingangsraums bzw. des Merkmalsvektors (N Merkmale $\Rightarrow N$ Neuronen). Die verdeckte Schicht besteht aus M Neuronen, welche eine radiale Basisfunktion als Transferfunktion $h_{\text{GRBF}}(\underline{x})$ mit dem Zentrum $\underline{\mu}_m$ aufweisen. In der hier verwendeten Variante

5 Klassifikation durch Maschinenlernverfahren

stellen die Neuronen in der verdeckte Schicht die prototypischen Vertreter der zu unterscheidenden Klassen dar. Die Prototypen werden in der Trainingsphase des Netzes bestimmt. Die Anzahl der Neuronen in der Ausgangsschicht wird durch die Anzahl C der zu unterscheidenden Klassen vorgegeben.

Die Zugehörigkeitsfunktion $d_k(\underline{x})$ des Merkmalsvektors \underline{x} zur Klasse $y = c_k$ berechnet sich durch

$$d_k(\underline{x}) = \sum_{m=1}^M \alpha_{m,k} h_{\text{GRBF},m}(\underline{x}). \quad (5.4)$$

Für die Klassifikation wird zugunsten der Klasse c_k mit dem größten Zugehörigkeitswert $d_k(\underline{x})$ entschieden:

$$f_{\text{GRBF}}(\underline{x}) = \arg \max_{c_k} \{d_k(\underline{x})\}. \quad (5.5)$$

Als Vertreter der radialen Basisfunktionen werden in dieser Arbeit modifizierte Gauß-Funktionen verwendet. Diese sind im mehrdimensionalen Fall für den m -ten Prototypen durch den Mittelpunktvektor $\underline{\mu}_m$, die inverse Kovarianzmatrix \mathbf{C}_m^{-1} und einen allgemeinen Gewichtungsfaktor γ wie folgt definiert:

$$h_{\text{GRBF},m}(\underline{x}) = e^{-\gamma(\underline{x}-\underline{\mu}_m)^T \mathbf{C}_m^{-1}(\underline{x}-\underline{\mu}_m)}. \quad (5.6)$$

Für den eindimensionalen Fall reduziert sich die Kovarianzmatrix \mathbf{C} zu dem Wert σ^2 , d. h. $\mathbf{C}^{-1} = 1/\sigma^2$. Mit $\mu = 50$, $\sigma = 20$ und $\gamma = 1$ ergibt sich der in Abbildung 5.4 dargestellte Verlauf einer Gaußglocke.

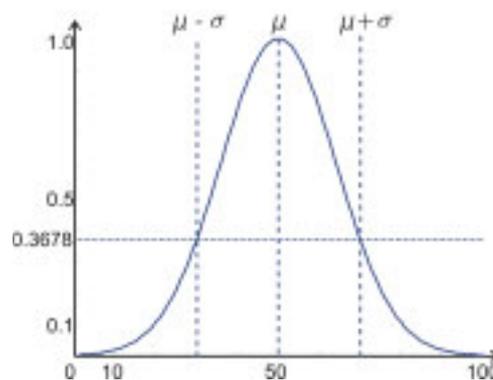


Bild 5.4: Ausgangsfunktion eines GRBF-Neurons (hier: Gaußfunktion)

Die Parameter eines GRBF-Netzes mit Gauss-Kernel sind also:

- die Zentren der radialen Basisfunktionen $\underline{\mu}_m$

- die Maße für die Ausdehnung der Hyperellipsoide in die einzelnen Raumrichtungen, festgelegt durch die inverse Kovarianzmatrix \mathbf{C}_m^{-1} des m -ten Prototypen
- die Gewichtungsfaktoren $\alpha_{m,k}$ für die Überlagerung der Ergebnisse der Hidden Layer in den Neuronen der Ausgangsschicht.
- der globale Gewichtungsfaktor γ

Eine sinnvolle Möglichkeit, diese Größen zu bestimmen, besteht darin statistische Auswertungen auf dem Trainingsdatensatz zu vollziehen. In dieser Arbeit wird dazu ein von M. Botsch speziell entwickeltes Verfahren verwendet, das ein Ähnlichkeitsmaß für Trainingsbeispiele aus der Trainingsphase eines Random-Forest-Klassifikators berechnet [20]. Anhand dieses Ähnlichkeitsmaßes können die Zentren μ_m der Prototypen und die Varianzen der Merkmalswerte für die zu diesem Prototypen ähnlichen Trainingsbeispiele errechnet werden. Dadurch können trotz Verwendung des GRBF-Klassifikators die Generalisierungsfähigkeit und die Möglichkeiten der Merkmalsselektion aus der Trainingsphase des RF-Klassifikators genutzt werden (siehe Kapitel 5.3).

Die Überführung in ein GRBF-Netz bringt noch zusätzliche Vorteile für den hier vorliegenden Anwendungsfall: Zum einen können GRBF-Netze sehr effizient implementiert werden. Zusätzlich bringen sie die verloren gegangene Interpretierbarkeit des RF-Klassifikators zurück, wenn sich die Dimension des Merkmalsvektors in Grenzen hält. So können zum Beispiel die entstehenden Prototypen vom Entwickler auf Plausibilität überprüft werden. Ein weiterer Vorteil der GRBF-Netze besteht darin, dass durch die mit der Distanz zum Zentrum abfallenden Werte der Radialen-Basis-Funktionen lokale Rückweisungsschwellen eingeführt werden können. Dadurch können Datenmuster aus einem noch nie gesehenen Bereich des Merkmalsraums gesondert betrachtet werden. Das ermöglicht es einem Algorithmus, der das Ergebnis des GRBF-Klassifikators weiterverwendet, eine Warnung bei unbekanntem Mustern zurückzuliefern, anstatt eine falsche Entscheidung zu treffen.

5.3 Merkmalsselektion

Der Entwickler eines Klassifikationssystems unternimmt große Anstrengungen, um möglichst gute Merkmale zu erhalten. Dabei ist das Ziel mit einer minimalen Anzahl an Merkmalen genug Information über das zu klassifizierende Objekt zu erhalten. Eine Möglichkeit dies zu erreichen besteht darin, zunächst viele Merkmale $\underline{x} \in \mathbb{R}^N$ zu extrahieren und mittels Merkmalsselektion die wichtigsten Merkmale zu identifizieren, um den Merkmalsvektor $\underline{x}^* \in \mathbb{R}^{N'}$ mit $N' \ll N$, zu erzeugen.

Es gibt Filter-, Embedded- und Wrapper-Methoden, um die Wichtigkeit eines Merkmals, im Folgenden als *Feature Importance* (FI) bezeichnet, in Bezug auf die enthaltene klassentrennende Information, zu bestimmen[21]. Filtermethoden benutzen einfache Korrelationsmaße zwischen Merkmal und Klassifikationsergebnis, um die Wichtigkeit eines Merkmals zu bestimmen. Im Gegensatz dazu verwenden Wrapper- und Embedded-Methoden dafür Lernverfahren. Diese können auch Abhängigkeiten von Merkmalen zuein-

ander erkennen, was ein erheblicher Vorteil gegenüber Filtermethoden ist. Der Unterschied zwischen Embedded- und Wrapper-Methoden besteht darin, dass Embedded-Methoden Wichtigkeitsmaße verwenden, die speziell an den verwendeten Lernalgorithmus angepasst sind. Wrapper-Methoden behandeln den Lernalgorithmus als „Black Box“ und können somit auf beliebige Lernverfahren angewendet werden.

Laut [21] entstehen durch den Schritt der Merkmalsselektion folgende Vorteile:

- Geringere Fehlerrate
- Weniger Ressourcen
 - Weniger Aufwand bei der Implementierung des Klassifikators
 - Schnellere Ausführungszeit
- Besseres Verständnis des Eingangsdatenvektors

Die hier verwendete Wrapper-Methode basiert auf der folgenden Vorgehensweise: Zunächst wird mittels *Bootstrapping* ein erwartungstreuer Schätzwert $\hat{R}(f^{(b)})$ für den Gesamtfehler bzw. das Risiko der Klassifikation durch den b -ten Entscheidungsbaum des RF-Klassifikators generiert. Dabei wird ein Testdatensatz der von dem Baum im Training noch nicht gesehenen Daten herangezogen. Um die Wichtigkeit des n -ten Merkmals in diesem Baum zu bestimmen, wird in einem zweiten Schritt die klassentrennende Information, die das n -te Merkmal in sich trägt, aus dem Trainingsdatensatz entfernt. Dazu werden die Werte des n -ten Merkmals der einzelnen Testbeispiele im Testdatensatz zufällig permutiert. Danach wird erneut der Gesamtfehler $\hat{R}^{(n,\text{perm})}(f^{(b)})$ des Klassifikators bestimmt, der auf dem permutierten Datensatz erzeugt wurde. Die Feature Importance des n -ten Merkmals im b -ten Baum berechnet sich dann zu

$$\text{FI}^{(b)}(\mathbf{x}_n) = \hat{R}^{(n,\text{perm})}(f^{(b)}) - \hat{R}(f^{(b)}). \quad (5.7)$$

Dieses Vorgehen wird für alle Merkmale durchgeführt. Schließlich findet noch eine Normierung von $\text{FI}^{(b)}(\mathbf{x}_n)$ auf den Wert des wichtigsten Merkmals statt:

$$\text{FI}^{(b)}(\mathbf{x}_n) = \frac{100 \cdot \text{FI}^{(b)}(\mathbf{x}_n)}{\max_{n'} \{\text{FI}^{(b)}(\mathbf{x}_{n'})\}}. \quad (5.8)$$

Diese Vorgehensweise wird auf alle B Entscheidungsbäume des RF-Klassifikators angewendet. Durch Mittelung von $\text{FI}^{(b)}(\mathbf{x}_n)$ über alle B Bäume erhält man einen endgültigen Wert für die Wichtigkeit eines Merkmals im Hinblick auf die Klassifikationsaufgabe im Trainingsdatensatz:

$$\overline{\text{FI}}_{\text{ges}}(\mathbf{x}_n) = \frac{1}{B} \sum_{b=1}^B \text{FI}^{(b)}(\mathbf{x}_n). \quad (5.9)$$

6 Fahrzeugdetektion bei Tag und Nacht

Ziel dieser Arbeit war es ein Gesamtsystem zum Erkennen und Verfolgen von Fahrzeugen und Fahrspuren weiter zu entwickeln. Zum Tracken der Objekte lag bereits eine gute Variante eines Fahrspur- und Fahrzeugtrackers vor. Zum Initialisieren eines Trackers muss jedoch vorher das noch unbekannte Objekt detektiert werden. Das heißt, es muss erkannt werden, ob sich ein neues Objekt im Sensorerfassungsbereich befindet, wo es sich befindet und zu welcher Objektklasse es gehört. Erst mit diesen Informationen kann das weitere modellbasierte Verfolgen des Objektes angestoßen werden. Vor diesem Hintergrund sind in dieser Arbeit komplexe Algorithmen zum Detektieren von Fahrzeugheckansichten für Tag- und Nachtszenarien entstanden.

Die bei der Entwicklung verfolgte Philosophie wird im Unterkapitel 6.1 erläutert. Danach wird im Abschnitt 6.2 kurz auf die Herausforderungen dieser Aufgabe eingegangen, bevor dann in den Unterkapiteln 6.3 und 6.4 die entstandenen Algorithmen im Detail beschrieben werden.

6.1 Philosophie

Die Ziele bei der Entwicklung waren geprägt von der Anforderung mit möglichst wenig Rechenzeit eine gute Erkennungsleistung für PKW-Heckansichten bei Fahrten auf Autobahnen und ausgebauten Landstraßen zu erzielen. Beim Experimentieren mit verschiedenen Operatoren zur Extraktion intuitiv relevanter Merkmale und deren Kombination zur Objektbildung ergeben sich stets zwei Probleme, die es zu lösen gilt:

1. **Unbeherrschbare Kombinationsvielfalt:** Führt man die Merkmalsextraktion im gesamten Bild aus, ergeben sich Unmengen an Merkmalen, von denen die meisten nicht vom eigentlich gesuchten Objekt stammen. Das liegt daran, dass meist einfache Operatoren eingesetzt werden, die lediglich Primitive des Objektes erkennen sollen und sehr wenig Rechenzeit benötigen. Solche Bildprimitive (wie Kanten, Ecken, usw.) kommen in unstrukturierten Umgebungen leider auch in der Umgebung des eigentlich gesuchten Objektes vor. Trotzdem bietet sich diese Vorgehensweise deshalb an, da die Bildprimitive die relevante Bildinformation komprimiert darstellen. Eine Objekterkennung auf diesen Primitiven ist daher viel effizienter als auf den Rohdaten selbst. Auch beim menschlichen Sehen spielen Vorverarbeitungen und Merkmalsextraktion eine große Rolle. Heutige Rechner kommen jedoch immer

noch an ihre Grenzen hinsichtlich Echtzeitfähigkeit der Algorithmen, wenn mehrere Primitive in einem voll aufgelösten VGA-Bild extrahiert und alle Kombinationsmöglichkeiten auf eine Objektklassenzugehörigkeit hin in Echtzeit überprüft werden sollen.

2. **Suboptimale Parametrisierung:** Bei der eben angesprochenen Kombination der Merkmale ist es häufig sinnvoll, dass die Merkmalswerte unterschiedlich stark gewichtet oder nichtlinear kombiniert werden, um eine Gesamtqualität einer Merkmalskombination zu berechnen und diese mit einer anderen Kombinationsmöglichkeit zu vergleichen. Auch können Schwellwerte zum Einsatz kommen, um bestimmte Merkmale als besonders gut oder schlecht passend zu deklarieren. Diese Gewichtungsfaktoren und Schwellwerte müssen vom Entwickler bei der Erzeugung des Detektors zunächst als sinnvoll erkannt und dann auf einen möglichst optimalen Wert gesetzt werden. Auch ist die Reihenfolge der Entscheidungsausführung wichtig für die spätere Detektionsleistung. Bei dieser Aufgabe benötigt man einen sehr erfahrenen Experten und selbst dieser kann sich nicht sicher sein, ob Ausführungsreihenfolge und die gesetzten Werte für die vorhandenen Merkmale optimal sind. Deshalb kann man bei so genannten Expertensystemen in der Regel von einer suboptimalen Parametrisierung ausgehen.

Um diesen Effekten entgegen zu wirken, ist ein zweistufiges Detektionsverfahren entstanden, das in einer ersten Stufe auf Basis einer Vorverarbeitung ROIs generiert, welche dann an die Objekterkennung in der zweiten Stufe weitergegeben werden (siehe Abbildung 6.1).

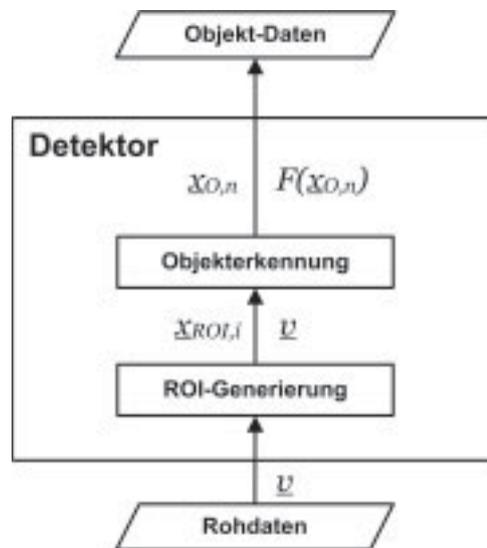


Bild 6.1: Zweistufiger Detektor

Durch die Vorverarbeitung in der ersten Stufe (ROI-Generierung) wird die Kombinationsvielfalt extrem eingeschränkt, da die Merkmalskombination nicht ROI-übergreifend durchgeführt werden muss. Sie ist so ausgelegt, dass mittels weniger und einfacher Primitive Regionen im Bild gefunden werden, wo möglicherweise ein Objekt sein könnte (Objekthypothese). Dabei spielt objektspezifisches Vorwissen eine bedeutende Rolle. Die Schwellen

und Gewichtungsfaktoren werden in diesem Schritt immer zu Gunsten der Objekthypothese ausgelegt, so dass nicht aus Versehen wirklich vorhandene Objekte ausgeschlossen werden. Dieser Teil kann gut durch ein Expertensystem umgesetzt werden, da es nicht auf eine optimale Parameterfindung ankommt. Trotzdem gibt es im Bild viele Bereiche, die diese einfachen Voraussetzungen für das Vorhandensein des gesuchten Objektes nicht erfüllen können und somit für die weitere Suche ausgeschlossen werden können.

In der zweiten Stufe (Objekterkennung) werden die positiven ROIs mit dem eigentlichen Detektor untersucht. Dieser extrahiert in den ROIs des Videobildes objektspezifische Merkmale. Sinnvolle Merkmalskombinationen werden dann mit Hilfe eines durch ein Maschinenlernverfahren generierten Klassifikators einer Objektklasse zugewiesen. Das in der Trainingsphase des Klassifikators eingesetzte Lernverfahren stellt sicher, dass auf der Grundlage der zur Verfügung gestellten Objektmerkmale und des Trainingsdatensatzes, eine optimale Parametrisierung hinsichtlich Fehlerrate und Generalisierungsfähigkeit gefunden wird.

Beide Stufen laufen nach dem in Abbildung 6.2 dargestellten Prinzip ab: Auf dem Datenvektor \underline{v} wird eine Merkmalsextraktion (MME) durchgeführt. Die Hypothesengenerierung (HG) sucht in \underline{x} nach Merkmalskombination, die auf ein vorhandenes Objekt schließen lassen können. Für jede Objekthypothese entsteht der zugehörige Merkmalsvektor $\underline{x}_{h,i}$. Dieser wird an das Entscheidungsmodul weitergegeben, welches den Merkmalsvektor $\underline{x}_{h,i}$ der wahrscheinlichsten Klasse c_k zuweist. In der Trainingsphase ist das Entscheidungsmodul durch ein Referenzsystem¹⁾ (RS) und in der späteren Anwendung durch ein Klassifikationssystem²⁾ (KS) realisiert wird.

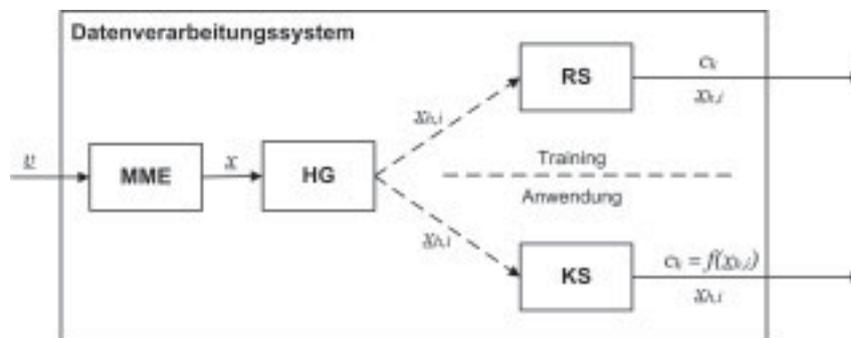


Bild 6.2: Datenverarbeitung

Das Verfahren zeichnet sich dadurch aus, dass es ermöglicht, sehr viel Vorwissen des Entwicklers und gleichzeitig die Vorteile eines statistischen Klassifikators mit einzubringen. Es ist also eine gute Kombination aus Expertensystem und Maschinenlernverfahren, wodurch sehr effiziente, nachvollziehbare und robuste Algorithmen entstehen.

¹⁾ Das Referenzsystem muss die geforderte Aufgabe mindestens so gut lösen können, wie es in der späteren Anwendung gefordert wird. Für diese Aufgabe ist der Entwickler selbst ein gutes Referenzsystem, da er die Bildausschnitte mit den gefundenen Merkmalskombinationen einer Klasse zuweisen kann.

²⁾ Das Klassifikationssystem kann wiederum entweder ein Expertensystem oder ein statistischer Klassifikator sein.

6.2 Herausforderungen

Mit der videobasierten Objekterkennung beschäftigen sich Forschergruppen weltweit seit mehreren Jahrzehnten. Jedoch war der Einsatz im Bereich der Fahrzeugführung bis vor wenigen Jahren lediglich eine Vision. Erst die gestiegenen Rechenleistungen und Speicherkapazitäten in Steuergeräten und die günstiger, robuster und qualitativ hochwertiger werdende Herstellung von CMOS-Chips für Videokameras brachte diese Vision in den letzten Jahren dem Serieneinsatz für Fahrerassistenzsysteme einige Schritte näher und wird derzeit bereits in den Fahrzeugen mehrerer Hersteller zur Erkennung der Fahrspur serienmäßig eingesetzt (siehe 2.1). Die heutigen Verfahren der Bildverarbeitung sind jedoch noch nicht robust genug, um auch für sicherheitskritische Funktionen eingesetzt zu werden. Das hat folgende Gründe:

Beschränkte Sensordatenqualität:

Wie auch den Abschnitten 4.4.2 und 4.4.3 angesprochen wird, muss die Sensordatenverarbeitung im Automobil noch mit großen Problemen bei der Genauigkeit der Sensordaten kämpfen. So sind Sensordaten häufig offsetbehaftet und der genaue Messzeitpunkt der Sensordaten ist nicht bekannt. Dies hat auch mit den extremen Bedingungen, die im Fahrzeug vorherrschen, zu tun. Die Sensoren sind großen Vibrationen, Temperaturunterschieden und sonstige Umwelteinflüssen ausgesetzt. Auch ist der finanzielle Spielraum bei der Fahrzeugproduktion sehr klein und die Anforderungen an eine harmonische Integrationsfähigkeit der Sensoren ins Fahrzeuggesamt-konzept aus Sicht des Fahrzeugdesigns sehr hoch. Das erfordert häufig eine Kompromisslösung bei der Qualität der Sensordaten.

Unstrukturierte Umgebung:

Unter einer unstrukturierten Umgebung versteht man in der Objekterkennung einen Einsatzort eines Erkennungsalgorithmus, dessen Umgebungsbedingungen nicht bekannt sind und sich dynamisch ändern können. Robotersysteme in Fabriken arbeiten meist in einer strukturierten Umgebung. Das heißt, sie haben ein Modell der Umgebung am Einsatzort hinterlegt. Da sich diese Umgebung niemals ändert, können sie sich sicher in der Umgebung bewegen. Auch bei der Objekterkennung ist eine strukturierte Umgebung vorteilhaft, da zum Beispiel die Beleuchtungsbedingungen für videobasierte Algorithmen optimal eingestellt werden können. Im Einsatzbereich für Fahrerassistenzsysteme muss jedoch von einer hochgradig unstrukturierten Umgebung ausgegangen werden. Diese zeichnet sich durch folgende Bedingungen aus:

- Die zu erkennenden Objekte können nah und fern sein.
- Der Hintergrund kann hell oder dunkel sein und ...
- ... viel oder wenig Struktur aufweisen.
- Es können viele Objekte oder auch kein Objekt im Erfassungsbereich sein.
- Es können die unterschiedlichsten Witterungs- und Beleuchtungsbedingungen vorherrschen (z. B. tiefstehende Sonne/Monsunregen/Schnee/Hagel/Nebel).

- Es können (Teil-)Verdeckungen von Objekten auftreten.
- Im Hinblick auf den Radarsensor können störende Reflexionen (Geisterobjekte) auftreten oder nicht.

Variierende Objektmerkmale:

Die eben genannten Eigenschaften der unstrukturierten Umgebung haben gravierende Auswirkungen auf die Merkmalsbeschaffenheit der zu detektierenden Objekte. Es herrscht eine hohe, situationsabhängige *Intra-Objekt-Variation* der Merkmale. So weist ein Objekt im Nahbereich durch die höhere Detailauflösung andere Merkmale auf, als im Fernbereich. Verschmutzte oder schneebedeckte Fahrbahnen weisen andere Merkmale auf als freie Straßen bei schönem Wetter. Fahrzeuge können ihre Beleuchtung je nach Umgebungshelligkeit ein- oder ausgeschaltet haben und Nummernschilder beleuchtet oder verschmutzt sein. Je nach Witterung und Tageszeit verursachen Objekte Schlagschattenmerkmale oder nicht.

Zusätzlich dazu haben die im realen Verkehr vorkommenden Fahrzeuge und auch sonstige Hindernisse eine große Variantenvielfalt. Hinsichtlich der Objektmerkmale ergibt sich daraus eine hohe *Inter-Objekt-Variation*. Diese erhöht sich noch mehr, wenn man mehrere Kulturkreise betrachtet. Das beginnt bei unterschiedlichen Straßenbelägen (z. B. Asphalt/Schotter) und Fahrspur-Markierungstypen (z. B. weiß durchgezogen/gelb doppelt-gestrichelt/keine), führt über unterschiedliche Fahrzeugtypen (z. B. PKWs/Rikschas/Radfahrer) hin zu unterschiedlichen Verkehrsleitzeichen (z. B. Straßenschilder/Schilderbrücken/Polizisten).

6.3 Verfahren zur Fahrzeugdetektion bei Tag

Für die videobasierte Fahrzeugdetektion existiert eine Reihe von Algorithmen, wobei sich jedoch ein Großteil dieser Algorithmen auf die Detektion bei Tag konzentriert bzw. beschränkt. Aktuelle Beiträge in den IEEE-Konferenzen bzw. -Sammelwerke „Intelligent Vehicles“ , „Intelligent Transportation System“ , „Pattern Analysis and Machine Intelligence“ und „Image Processing“ verschaffen einen ersten Überblick über aktuell erforschte Detektionsstrategien für Tagansichten von Fahrzeugen [88]. Bei den merkmalsbasierten Verfahren spielt das Symmetriemerkmal, die Aussenkanten und die Schattenkante häufig eine entscheidende Rolle[61][78]. Die templatebasierten Verfahren setzen häufig statistische Methoden ein, um einen robusten Korrelationswert zwischen dem aktuellen Bildausschnitt und den Prototypen für eine Heckansicht zu berechnen[87][94].

In diesem Abschnitt wird nun aufbauend auf den theoretischen Kapiteln 4 und 5 die Umsetzung des Fahrzeugdetektors für Tagsequenzen im Detail beschrieben. Dabei wird zunächst in Abschnitt 6.3.1 auf die Erzeugung der Trainingsdatenbank eingegangen, bevor in den Unterkapiteln 6.3.2 und 6.3.3 die Teilmodule „ROI-Generierung“ und „Hauptdetektor“ beschrieben werden. Abschließend wird in Abschnitt 6.3.4 die Qualität des Erkennungssystems bewertet und diskutiert.

6.3.1 Trainingsdatenerzeugung

Die Trainingsdatengenerierung für Tagszenarien ist in zwei Schritte unterteilt: Speichern von Bildausschnitten und Labeln von Merkmalskombinationen. Die dafür entwickelten ADTF-Filter „ImageExtraction-Filter“ und „LabelTool-Filter“ werden nun kurz beschrieben.

1. Schritt: Rohdatenerzeugung:

Abbildung 6.3 zeigt einen Screenshot des ImageExtraction-Filter’s im ADTF-Framework. Der Filter ist von der Basisimplementierung eines ADTF-Videodisplays abgeleitet und über einen Pin an den Harddisk-Player angeschlossen (siehe Abschnitt 3.5). So kann das Video an der gewünschten Position pausiert werden.

Im Display werden auch die Projektion der Radarsignale als blaue bzw. zyanfarbene Dreiecke dargestellt. Zyan bedeutet dabei, dass dieses Objekt tatsächlich durch den Radar gemessen wurde, blau deutet auf einen in diesem Zyklus rein durch Tracking erhaltenen Wert hin. Um die Zuordnung der Radarsignale zu Fahrzeugen zu erleichtern, haben die Dreiecke jeweils eine Höhe und Breite von einem Meter und stehen in der als eben angenommenen Straße. Die beiden Zahlen unterhalb des Dreiecks bezeichnen zum einen die vom Radarsensor vergebene Objekt-ID, zum anderen die auf Ganzzahlen gerundete Entfernung des Fahrzeugs.

Im angehaltenen Videobild kann mit der Maus ein Rahmen erzeugt werden (weißer Rahmen in Abbildung 6.3). Pro Region kann ein zugehöriges Radarsignal markiert werden. Zur

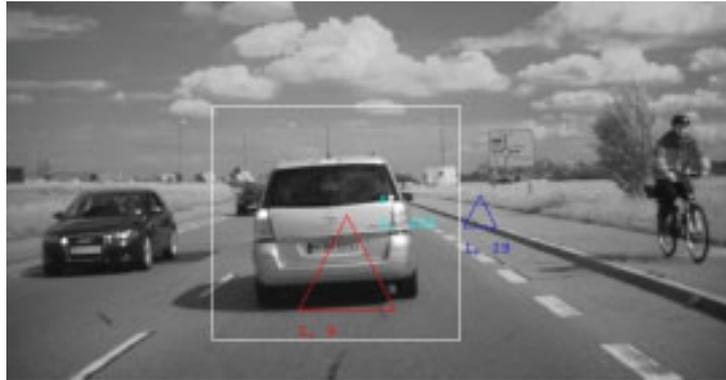


Bild 6.3: Screenshot des ADTF-Displays zum Erzeugen der Rohdatenbasis

Bestätigung färbt sich das Dreieck rot. Durch Drücken der Taste „S“ wird der ausgewählte Bildbereich zusammen mit den optional ausgewählten Radardaten in ein festgelegtes Verzeichnis gespeichert. Dabei entsteht für jede Auswahlregion eine .bmp-Datei und ein neuer Eintrag in einer Textdatei mit den nötigen Zusatzinformationen. Eine Zeile der Textdatei ist für den Fall der Fahrzeugheckerkennung bei Tag nach Tabelle 6.1 aufgebaut.

Bildname	Name der .bmp-Datei des zugehörigen Bildausschnittes.
X_{ROI}	x -Position der linken Kante der ROI im Originalbild.
Y_{ROI}	y -Position der Oberkante der ROI im Originalbild.
X_{LRR,rel}	Relative x -Position des projizierten Radarobjektes im Bildausschnitt.
Y_{LRR,rel}	Relative y -Position des projizierten Radarobjektes im Bildausschnitt.
t_{LRR}	Can-Zeitstempel des Radarsignals (nicht Messzeitpunkt!).
X_{LRR}	X -Position aus Radarbotschaft.
Y_{LRR}	Y -Position aus Radarbotschaft.
V_{LRR}	Relativgeschwindigkeit aus Radarbotschaft.

Tabelle 6.1: Datenstruktur der Zusatzinformationen in Textdatei

Aus den zur Verfügung stehenden Landstraßen- und Autobahnszenarien wurde eine Datenbank mit 716 PKW- und 92 LKW-Heckansichten bei unterschiedlichen Bedingungen generiert. Abbildung 6.4 zeigt eine kleine Auswahl an Ansichten der erzeugten Datenbasis.



Bild 6.4: Beispiele für Fahrzeugheckansichten bei Tag

2. Schritt: Labeln:

Im Fall der Trainingsdatengenerierung muss darauf geachtet werden, dass die sonstige Datenverarbeitung genauso wie in der späteren Anwendung abläuft. Da der spätere Einsatz in der Entwicklungsumgebung ADTF gedacht ist, ist ein ADTF-Filter entstanden („Label-Tool“), das dem Referenzsystem (RS) der Trainingsphase aus Abbildung 6.2 entspricht. Der Filter lädt die vorher erzeugten Bildausschnitte und startet für jeden Ausschnitt die Merkmalsextraktion. Der Filter ermöglicht ein stufenweises Labeln der Merkmale, da die Darstellung von mehreren Merkmalstypen mit vielen Merkmalskandidaten sonst zu unübersichtlich werden kann.



Bild 6.5: Anwendungsbeispiel für Labeltool

Die Ergebnisse (Merkmalspositionen) der Berechnungen der einzelnen Stufen werden dem Nutzer in Form von kleinen Punkten im Bild angezeigt. Ein Beispiel hierfür ist in Abbildung 6.5 aufgezeigt. Der Nutzer hat in jeder Stufe die Möglichkeit per Mausklick die beste Merkmalskombination auszuwählen. Die relevanten Informationen der ausgewählten Merkmalskombinationen werden zusammen mit dem Typ des zu klassifizierenden Objektes abgespeichert. So entsteht eine Liste von Trainingsdatenpaaren bestehend aus den Merkmalsvektoren \underline{x}_i und der zugehörigen Klasse c_k .

6.3.2 ROI-Generierung

Die ROI-Generierung stellt die erste Stufe des in Abbildung 6.1 gezeigten Gesamtprozessablaufs dar. Hier werden Regionen im Bild bestimmt, innerhalb derer sich wahrscheinlich Fahrzeugheckansichten befinden. Es wurden zwei verschiedene Module zur ROI-Generierung entwickelt: ein radarbasierter und ein rein videobasierter Ansatz.

Da das radarbasierte Modul lediglich eine Projektion von Radarobjekten in das 2D-Bild vornimmt und anschließend einen groben Auswahlrahmen um das punktförmige Radarobjekt erzeugt, wird dieses Modul hier nicht weiter erläutert. Der videobasierte Vorfilter hingegen ist ein relativ komplexes Subsystem, das gleichzeitig aufgrund des größeren Öffnungswinkels des Videosensors im Vergleich zum Radar vor allem im Nahbereich bessere Ergebnisse als der radarbasierte Vorfilter produziert.

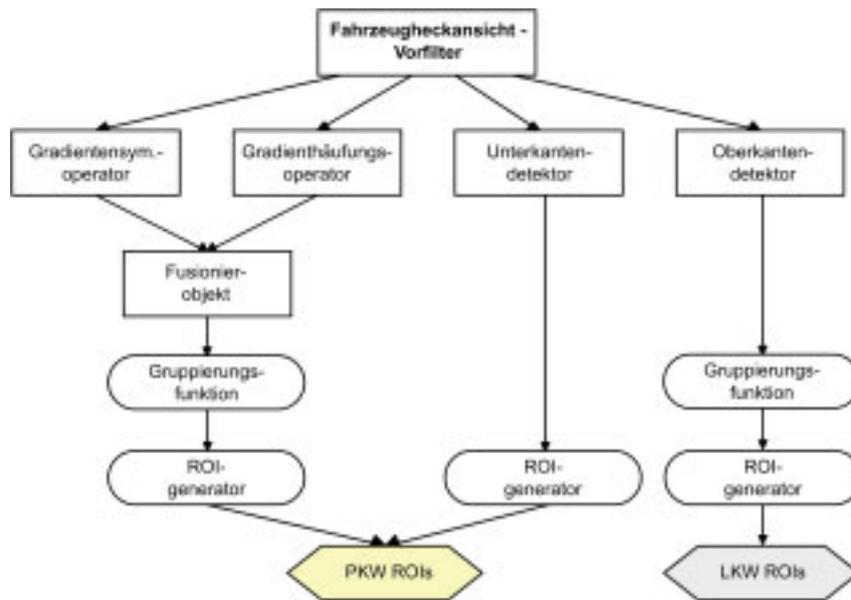


Bild 6.6: Kontrollflußdiagramm des Moduls zur ROI-Generierung

Abbildung 6.6 zeigt das Ablaufdiagramm des Gesamtsystems zur ROI-Generierung. Die Bezeichnung „Gruppierungsfunktion“ kennzeichnet dabei die Zusammenfassung mehrerer Hypothesen zu einer Hypothese, basierend auf einer Überdeckungsanalyse. Zur Generierung der PKW- bzw. LKW-ROIs kommen unterschiedliche Strategien zum Einsatz. Diese Aufspaltung hat ihren Grund in den unterschiedlichen Merkmalstypen, die sich als erfolgversprechend zur PKW- bzw. LKW-Grobdetektion erwiesen haben. Die beiden Vorgehensweisen werden nun detaillierter beschrieben.

PKW-ROIs

Bei der PKW-Grobdetektion werden zunächst die Grundmerkmale (Schatten, Unterkante, Symmetrie, Horizontale Kanten) im Bild gesucht. Im Folgenden werden die Vorgehensweisen zur Extraktion der Einzelmerkmale genauer beschrieben.

Merkmalskombination 1: Unterkante und Schatten

Als erster wichtiger Punkt wird die Kante beim Übergang von Teer zum Fahrzeug eingeführt. In direkter Kombination damit wird nach dem Schatten unterhalb eines Fahrzeugs gesucht. Ein ähnliches Vorgehen hat sich bereits in früheren Arbeiten (z. B. [?]) als sinnvoll erwiesen. Diese Vorgehensweise ist in Abbildung 6.7 dargestellt.

Dabei wird entlang vieler Suchpfade von der Unterkante des Videobildes nach oben hin nach Kanten gesucht. Bei jeder Kante wird eine Unterscheidung nach der Art des Helligkeitsübergangs (Vorzeichen des Gradientenwertes) getroffen (Hell-Dunkel oder Dunkel-Hell). Unter Berücksichtigung des Vorwissens, dass sich der Übergang von Straße zu Schatten des Fahrzeugs stets als Hell-Dunkel-Übergang darstellt, können anschließend

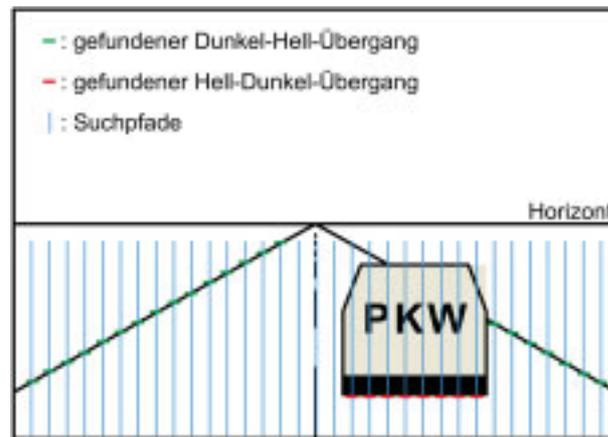


Bild 6.7: Schematische Darstellung der Unterkantendetektion für PKW-Heckansichten

Gruppierungen der entlang der einzelnen Pfade gefundenen Kanten vorgenommen werden. Als zusätzliches Wissen bei der Gruppierung fließt ein, dass die Unterkante eines Fahrzeugs näherungsweise horizontal im Videobild abgebildet werden sollte. Darüber hinaus kann aus der Position im Bild zusammen mit der Ebenenannahme auf die Entfernung der Schattenkante geschlossen werden. Daraus lässt sich eine grobe Abschätzung über die maximale und minimale ins Bild projizierte Fahrzeugbreite und damit über die erwartete Länge der Schattenkante ableiten. Mit steigender Anzahl an Pfaden steigt die Qualität, da Ausreißer bei der Gruppierung detektiert werden können und die räumliche Auflösung durch den geringeren Abstand zwischen den Suchpfaden sich verbessert. Negativ wirkt sich hingegen aus, dass mit steigender Anzahl an Pfaden die Rechenzeit ansteigt. Auch ist die maximale Anzahl der besten Kantenmerkmalspunkte, die pro Pfad für die Verkettung verwendet werden, eine wichtige Größe. Eine Erhöhung hier bewirkt einerseits ebenfalls eine Verbesserung der Ergebnisqualität, vor allem in Umgebungen mit inhomogener Fahrbahnoberfläche (z.B. schlechter Teerbelag mit ausgebesserten Elementen). Aufgrund der exponentiell steigenden Kombinationsmöglichkeiten bei der Gruppierung der Einzelmerkmale zu Kanten, führt das andererseits zu einem enormen Rechenzeitanstieg. Als guter Kompromiss zwischen Rechenzeit und Qualität hat sich bei einem Videobild der Größe 640×480 eine Wahl von 40 äquidistant verteilten Pfaden mit maximal drei Ergebnissen pro Pfad herausgestellt.

Merkmalskombination 2: Symmetrie und horizontale Kanten

Als Merkmale drei und vier werden die Symmetrie und die Häufung horizontaler Kanten herangezogen. Die Auswertung dieser beiden Merkmale ist aufgrund von Synergieeffekten kombiniert. Als Operatoren kommen der in 4.2.3 beschriebene Gradientensymmetrieoperator und der Gradientenhäufungsoperator aus Kapitel 4.2.1 zum Einsatz. Beide Operatoren zeichnen sich durch die gute Optimierbarkeit bei der Implementierung und die damit verhältnismäßig niedrige Berechnungszeit aus. Gleichzeitig liefern beide Operatoren nur bedingt genaue Ergebnisse, was für die Vorfilterstufe mit der ohnehin groben Parametrisierung der Operatoren jedoch ausreichend ist.

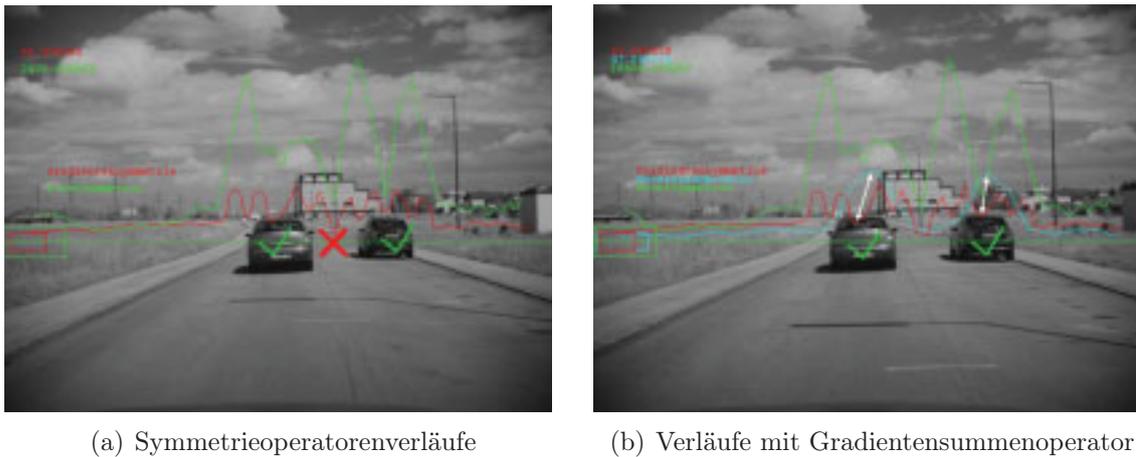


Bild 6.8: Synergieeffekte durch Operatorkombination

Die Synergieeffekte sind in Abbildungen 6.8 veranschaulicht. Die linke Abbildung zeigt die Verläufe der Symmetrieoperatoren auf Basis von Helligkeitswerten (grün) und Gradientenwerten (rot). Beide Verläufe suggerieren gerechtfertigt das Vorhandensein der beiden wirklich vorhandenen Fahrzeuge im Bild. Zusätzlich entsteht aber eine weitere Hypothese genau in der Mitte zwischen den beiden vorhandenen Fahrzeugen. Das lässt sich dadurch begründen, dass der Fahrbahnbelag als sehr homogene Fläche abgebildet wird. Deshalb sind die Unterschiede der Helligkeitswerte links und rechts zu einer angenommenen Symmetrieachse sehr klein. Auch die Unterschiede der symmetrisch zu der Achse liegenden Gradientenwerte sind sehr gering, da durch die homogene Fläche schon die Absolutbeträge der Gradienten sehr klein sind.

Durch Hinzufügen des Gradientensummenoperators aus Kapitel 4.2.1 lässt sich diese Fehlhypothese vermeiden, wie das rechte Bild (Abbildung 6.8(b)) zeigt. Durch das gehäufte Vorhandensein von horizontalen Strukturen auf Fahrzeughecks ist von einem Maximum im Verlauf des Gradientensummenoperatorverlaufs auszugehen, wenn in einem Bildausschnitt wirklich eine Fahrzeugheckansicht zu sehen ist. Wie erwartet bestätigt der gezeigte Verlauf des Operators (cyan) die beiden wirklich vorhandenen Fahrzeughypothesen aus dem Symmetrieoperatorverläufen. Die Fehlhypothese kann allerdings verworfen werden, da für diese Hypothese kein Maximum im Verlauf des Gradientensummenoperators gefunden werden kann.

Da sich Fahrzeuge in unterschiedlichen Entfernungen befinden können und sich somit unterschiedlich groß im Videobild abbilden, wird eine Suchstrategie nach Abbildung 6.9 verfolgt. Entsprechend der Ebenenannahme für die Straße werden entlang horizontaler Suchpfade die Operatorergebnisse für Symmetrie und Gradientenhäufung ausgewertet. Die Größe der Operatormasken wird, wie in Abbildung 6.9 dargestellt, an die Erwartung der Größe eines Durchschnitts-PKW in der durch den Suchpfad abgedeckten Entfernung angepasst. Die dabei detektierten Positionen für maximale Symmetrie und Gradientenhäufung werden zu vorläufigen PKW-Hypothesen kombiniert.

Bei der Verdeckungsanalyse wird überprüft ob einzelne Hypothesen durch andere, räumlich näher am Ego-Fahrzeug befindliche Hypothesen, verdeckt werden. Dieser Effekt tritt relativ häufig auf, da innerhalb der Heckansicht eines im Bild großen, also nahe am Ego-Fahrzeug befindlichen Fahrzeugs, oft Fehlhypothesen für kleinere Fahrzeuge generiert werden. Diese kleinen Fehlhypothesen können basierend auf der Verdeckungsbetrachtung verworfen werden.

LKW-ROIs

Die LKW-Vorauswahl stützt sich ausschließlich auf die markante Oberkante eines Kasten-LKWs. Andere Merkmale wie Symmetrie, Häufung horizontaler Kanten oder Unterkante mit Schatten weisen bei LKWs deutlich schlechtere Ergebnisse auf als bei PKWs. Dies begründet sich vor allem in den sehr unterschiedlichen Aufbauten und Bodenfreiheiten der LKW Typen. Die Oberkante hingegen zeichnet sich aus Sicht eines PKWs, aufgrund der Lage im Bild oberhalb des Horizonts, meist sehr deutlich gegenüber dem Hintergrund ab. Diese Kante wird analog zur Unterkantenbestimmung bei PKWs, wie in Abbildung 6.7 dargestellt, detektiert.

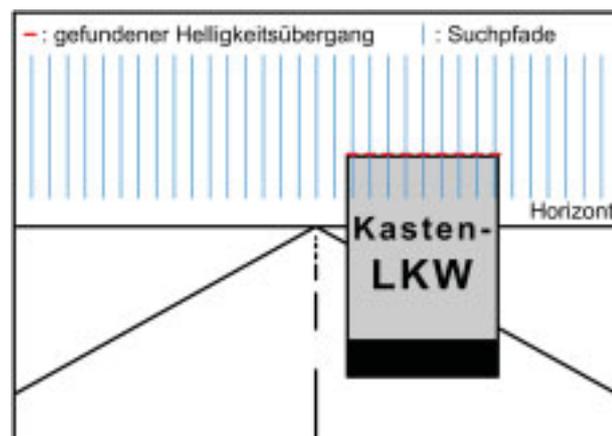


Bild 6.11: Schematische Darstellung der Oberkantendetektion für LKW-Heckansichten

Damit können bei freier Fahrt auf der Autobahn oder Landstraße sehr gute Ergebnisse erzielt werden. Probleme ergeben sich bei Autobahnunterführungen, Schilderbrücken oder in Städten, wenn Häuser im Hintergrund den LKW überragen. Hier steigt die Anzahl der Fehlhypothesen deutlich an. Wie zuvor bei den PKWs wird auch für die LKW-ROIs eine Überdeckungsanalyse durchgeführt.

ROI-Plausibilisierung

Ist die vorläufige Hypothesenbildung für LKWs und PKWs abgeschlossen, werden diejenigen Hypothesen ausgewählt, um die eine ROI generiert werden soll, welche an den nachgeschalteten Objektdetektor weitergegeben wird. Diese Auswahl wird durch einen

einfachen, dafür aber sehr robusten Klassifikator durchgeführt. Es wird ein Expertensystem eingesetzt, das basierend auf bestimmten Plausibilitätsgrenzen für Höhe und Breite der ROI unter Berücksichtigung des Kameramodells zu unrealistische Hypothesen verwirft. Die Grenzen werden dabei sehr tolerant gewählt, um sicherzustellen, dass keinesfalls ROIs verworfen werden, in denen sich tatsächlich Fahrzeuge befinden. Bei der ROI-Auswahl findet keine Überdeckungsrechnung zwischen LKW- und PKW-ROIs statt. Bei der Durchfahrt unter einer Brücke kann es passieren, dass fehlerhafte LKW-ROIs über korrekte PKW-ROIs gelegt werden. Eine Verdeckungsanalyse würde die PKW-Hypothese fälschlicherweise verwerfen, was nicht passieren darf. Nach der Plausibilisierung sollten nun alle wirklich vorhandenen Fahrzeuge in einer ROI liegen. Zusätzlich gibt es einige ROIs, welche keine Fahrzeuge enthalten. Deshalb muss ein nachgeschalteter Klassifikator die letztendliche Entscheidung treffen.

6.3.3 Objekterkennung mittels Klassifikation

Die nach den Beschreibungen des vorangehenden Kapitels entstehenden ROIs werden, wie Abbildung 6.1 verdeutlicht, an das Objekterkennungsmodule weitergereicht. Dieses geht wiederum nach dem Schema in Abbildung 6.2 vor. Es wird eine speziell auf die Parametrisierung des nachgeschalteten Klassifikationssystems abgestimmte Merkmalsgenerierung durchgeführt. Aus den Merkmalskombinationen werden Hypothesen generiert, welche dann von dem Klassifikator auf ihre Gültigkeit hin überprüft werden. Entsprechend gliedern sich die weiteren Erläuterungen in mehrere Abschnitte. Zunächst werden die vom Klassifikator unterscheidbaren Klassen nochmals aufgezeigt. Daraufhin werden die zur Speisung des Klassifikators verwendeten Merkmale beschrieben. Anschließend wird der entstandene Klassifikator mit den Werten der wesentlichen Parameter für den Anwendungsfall beschrieben.

Klassen

Das Objekterkennungsmodule muss basierend auf einer gegebenen ROI feststellen, ob sich in der ausgewählten Region des Bildes ein Fahrzeug („PKW“ oder „Kasten-LKW“) befindet. Ist dies nicht der Fall, unterscheidet der Klassifikator darüber hinaus die beiden Ergebnisse „irrelevant“ und „keine Aussage“. „irrelevant“ kennzeichnet im Gegensatz zu „keine Aussage“ den Fall, dass die augenblicklich bewertete Merkmalskombination einem Beispiel aus der Trainingsphase ähnelt, das explizit der Klasse „irrelevant“ zugeordnet wurde. Diese Klasse beinhaltet Negativbeispiele, welche trotzdem viele der gesuchten Merkmale ausgeprägt enthalten. Der Klassifikator kann diese Beispiele nutzen um die Lage der Merkmalsvektoren im Merkmalsraum, die zu wirklich vorhandenen Hypothesen gehören, noch besser von Merkmalsvektoren von Fehlhypothesen zu unterscheiden. Im Fall „keine Aussage“ ist die gemessene Kombination keiner der drei Klassen „PKW“, „LKW“ und „irrelevant“ eindeutig zuzuordnen. Das heißt: ein ähnliches Beispiel ist im Trainingsdatensatz nicht vorhanden. Es kann sinnvoll sein, in solchen Fällen der nachgeschalteten Anwendung eine Warnung auszugeben. Für die spätere Auswertung wurden jedoch die

Bereiche „irrelevant“ und „keine Aussage“ zu einer neuen Gruppe „keine Detektion“ zusammengefasst. Die Klasse „irrelevant“ hat dennoch ihre Berechtigung, da die Abgrenzung von PKW oder Kasten-LKW zum Restraum durch das Lernen einiger Negativbeispiele verbessert werden kann.

Merkmale

Als Eingangsgrößen des Klassifikators kamen in einer ersten Version die zehn Merkmale aus Tabelle 6.2 zum Einsatz. Zunächst wurde der RF-Klassifikator unter Verwendung aller Merkmale trainiert. Um den Rechenaufwand in der späteren Anwendung zu minimieren, wurde auf Basis der ermittelten *Feature Importance* FI nach Gleichung 5.9 aus Kapitel 5.1 eine Merkmalsselektion durchgeführt, so dass die Anzahl auf die wichtigsten Merkmale reduziert werden konnte. Wie die Grafik 6.12 zeigt, weisen die vier Merkmale „Proportion“, „Konturwert“, „Breitenschätzung“ und „Gradientenverhältnis“ den größten FI-Wert auf.

Merkmale	Berechnungsgrundlage
1. Proportion	Verhältnis von Höhe zu Breite
2. Konturwert	Konturbewertungsoperator aus Kapitel 4.2.2
3. Breitenschätzung	Bodenebenenannahme nach Abbildung 3.13
4. Gradientenverhältnis	Kantendetektionsoperatoren aus Kapitel 4.1.1
5.&6. Reifenposition (u,v) links	Reifendetektor aus Kapitel 4.3.1
7.&8. Reifenposition (u,v) rechts	Reifendetektor aus Kapitel 4.3.1
9. Häufung horizontaler Kanten	Gradientenhäufungsoperator aus Kapitel 4.2.1
10. Symmetrie	Symmetrieoperatoren aus Kapitel 4.2.3

Tabelle 6.2: Übersicht über Klassifikationsmerkmale

Die Vorgehensweise zur Generierung der wichtigsten vier Klassifikationsmerkmale wird nun im Detail beschrieben:

1. Suche nach Seitenkantenhypotesen:

Die Detektion der Fahrzeugkanten geschieht mittels Gradientenoperator nach Abbildung 4.3 und anschließender Suche nach Maximalwerten im generierten Operatorverlauf. Die horizontalen und vertikalen Kanten werden dabei schrittweise bestimmt. Zunächst werden

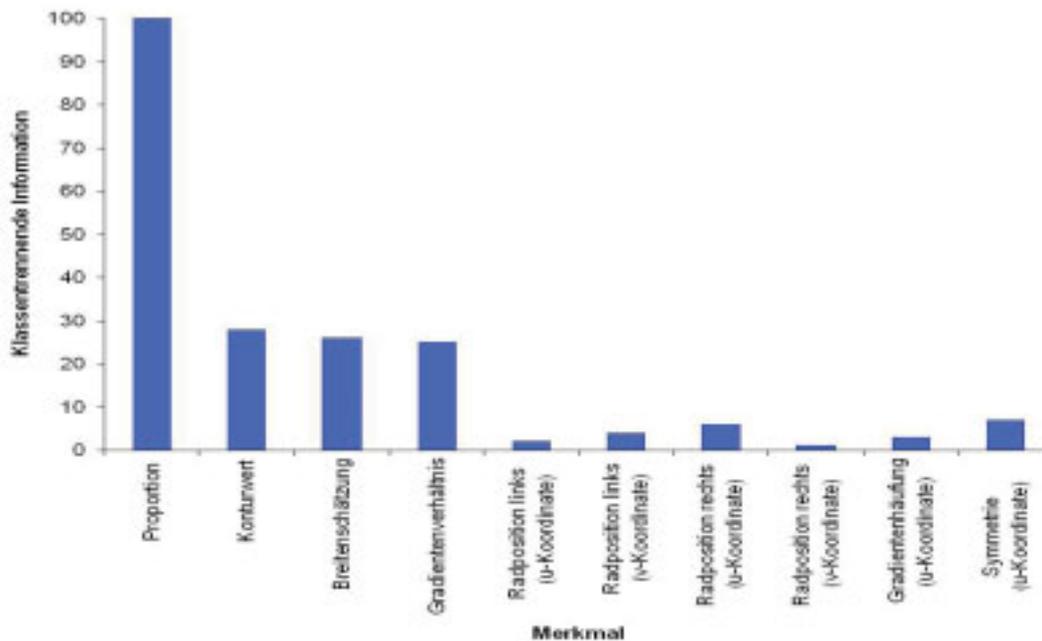


Bild 6.12: (FI-Wert) der Merkmale

die vertikalen Kanten innerhalb der übergebenen ROI identifiziert. Die Operatorsuchmaske wird dabei sehr tolerant bezüglich der erwarteten Kantenlänge parametrisiert, da aus der Größe der ROI nur eine sehr grobe Abschätzung über die Größe des enthaltenen Fahrzeugs abgeleitet werden kann. Im Anwendungsfall bedeutet dies, dass die Mittelungstiefe senkrecht zum Suchpfad relativ klein gewählt wird, um auch kleine Autos mit kurzen Aussenkanten in der ROI detektieren zu können. Die Suchpfade für die linke und rechte Kante beginnen dabei jeweils im Zentrum der ROI und verlaufen in Richtung des linken bzw. rechten Randes. Pro Kante werden die besten N_{AK} Außenkanten-Hypothesen für die weitere Verwendung gespeichert (hier: $N_{AK} = 2$). Daraus leiten sich $2 \cdot 2 = 4$ Kombinationsmöglichkeiten ab.

2. Suche nach Ober- und Unterkantenhypothesen:

Für jedes mögliche Seitenkantenpaar wird nun nach horizontalen Kanten gesucht. Die Maskenhöhe des Gradientenoperators wird dabei zwar an den Abstand der vertikalen Kanten angepasst, aufgrund von Toleranz bezüglich verschiedener Fahrzeugbauformen aber etwas geringer gewählt. Für die Oberkante werden die besten $N_{OK} = 2$ Hypothesen und für die Unterkante die besten $N_{UK} = 4$ Hypothesen gespeichert. Im Fall der LKW-ROIs wird als Oberkante direkt die Oberkante der ROI übernommen, da die Suche nach markanten Oberkanten bei der ROI-Generierung bereits sehr gründlich durchgeführt wird.

3. Feinjustierung und Hypothesengenerierung:

Für jedes Ober- und Unterkantenpaar wird zur Feinjustierung der Kontur in einem kleinen Bereich um die bisher festgestellten zugehörigen Seitenkanten erneut nach vertikalen

Kanten gesucht. Diesmal werden allerdings die Suchmasken entsprechend des Abstandes zwischen Ober- und Unterkantenhypothese parametrisiert. Zusätzlich wird der Suchpfad zur Seitenkantendetektion in die untere Hälfte der Fahrzeughypothese verlegt, da sich hier, durch den Übergang der dunklen Reifen zum Teerbelag der Straße, die vertikalen Kanten sehr ausgeprägt darstellen. Auch hier wird eine LKW-ROI insofern anders behandelt, als dass auf die erneute Suche nach den vertikalen Kanten zur Feinjustierung verzichtet wird. Diese können aufgrund ihrer starken Ausprägtheit bei Kasten-LKWs bereits im ersten Lauf gut genug bestimmt werden.

Jede Kombinationsmöglichkeit der gefundenen Kantenhypothesen wird anschließend als Heckhypothese gespeichert und als Grundlage für die Parametrisierung der Operatoren für die weiteren Berechnungen verwendet.

4. Berechnung des Merkmals „Proportion“ :

Das Merkmal „Proportion“ lässt sich direkt über das Höhe-zu-Breite-Verhältnis der erzeugten Hypothesen berechnen.

5. Berechnung des Merkmals „Konturwert“ :

Die Parameter des Konturbewertungsoperators aus Kapitel 4.2.2 werden so gewählt, dass die identifizierten Kantenstücke Teil des verfolgten Pfades sind. Dadurch ergeben sich hohe Ergebniswerte, wenn sich die Fahrzeugkontur auch in den Bereichen fortsetzt, die durch die Kantendetektion nicht berücksichtigt werden. Dies ist in Abbildung 6.13 dargestellt.

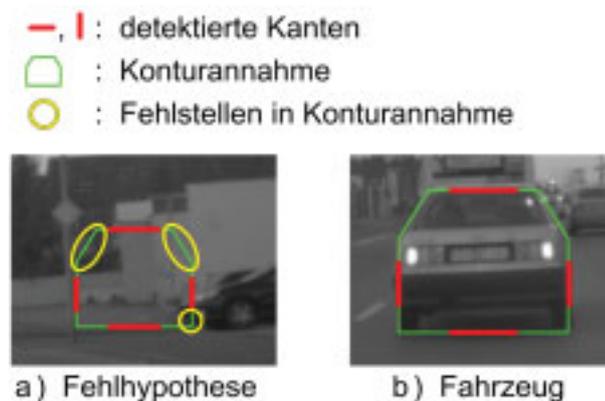


Bild 6.13: Realistisches Beispiel für Bedeutung des Konturbewertungsoperators

Für PKW-Hypothesen wird die Heckansicht des Fahrzeug-Formmodells aus Kapitel 4.4.1 verwendet. Durch die Wahl von drei Nullzeilen zu beiden Seiten des Konturpfades ist eine ausreichende Toleranz gegenüber Abweichungen von der gemittelten Konturannahme gegeben.

Die Kontur von Kasten-LKWs wird durch einen rechteckigen Kasten angenähert. Dabei wird allerdings die Oberkante nicht mehr berücksichtigt, da diese im Vorfilter bereits detailliert untersucht wurde und aufgrund der Länge der Kanten durch den Verzicht auf die erneute Berechnung Rechenzeit eingespart werden kann. Zusätzlich haben sich bei

6 Fahrzeugdetektion bei Tag und Nacht

der Ergebnisqualität mit und ohne Berücksichtigung der Oberkante keine erkennbaren Unterschiede ergeben.

5. Berechnung des Merkmals „Breitenschätzung“ :

Die geschätzte reale Breite der Fahrzeughypothese wird mit Hilfe der Bodenebenenannahme analog zu Abbildung 3.13 berechnet. Zur Durchführung der Berechnung werden als Merkmalspunkte im Bild die Schnittpunkte der vertikalen Aussenkanten mit der horizontalen Unterkante des Fahrzeugs verwendet.

6. Berechnung des Merkmals „Gradientenverhältnis“ :

Für dieses Merkmal kommt eine spezielle Variante eines Kantenoperators zum Einsatz. Innerhalb der untersuchten Region wird je ein Zwischenbild der horizontalen und vertikalen Gradientenbeträge erzeugt. In diesen Zwischenbildern wird die Anzahl N_{Hor} und N_{Ver} der horizontalen und vertikalen Gradientenwerte bestimmt, die einen bestimmten Schwellwert überschreiten. Das Verhältnis $\frac{N_{\text{Ver}}}{N_{\text{Hor}}}$ der vertikalen zu horizontalen starken Gradientenmerkmale wird als Ergebnis des Operators zurückgegeben. Dies entspricht dem Längenverhältnis der horizontalen zu vertikalen Kanten in der Region.

Die Auswahlregion zur Gradientenberechnung wird entsprechend der gemessenen Außenkanten festgelegt. Dabei ist zu beachten, dass die seitlichen Kanten nicht direkt auf die Seitenkantenhypothesen des Fahrzeugs gelegt werden, sondern innerhalb des Fahrzeughecks platziert werden. Dies beruht darauf, dass ein Fahrzeugheck zwar prinzipiell bereits ein charakteristisches Übergewicht der horizontalen gegenüber den vertikalen Gradienten aufweist, durch das bewusste Aussparen der dominanten vertikalen Aussenkanten, dieser Effekt aber nochmals verstärkt werden kann. Bei LKWs wird die Auswahlregion zusätzlich auf den unteren Bereich um die Stoßstange eingeschränkt, um dominante vertikale Konturen im Bereich des Aufbaus, wie beispielsweise durch Ladetüren und Beschriftung bedingt, auszusparen.

6.3.4 Bewertung des Verfahrens

Als Grundlagen für die im Weiteren dargelegten Entscheidungen dient ein Satz von 19 Fahrscenarien, in denen die Bildpositionen der im Erfassungsbereich der Kamera fahrenden Fahrzeuge per Hand gelabelt wurden. Darunter befinden sich 7 Aufnahmen von Autobahnscenarien, 5 Aufnahmen von Stadtfahrten, 5 Aufnahmen von Landstraßenfahrten sowie 2 Aufnahmen einer Kombination aus Stadt und Landstraßenfahrt. Damit ist bereits ein recht ausgewogenes Spektrum an Landschaften abgedeckt. Hinzu kommen unterschiedliche Beleuchtungsverhältnisse in den Szenarien, die von heiter über neblig bis hin zu fortgeschrittener Dämmerung reichen.

Abbildung 6.14 zeigt die Visualisierung der händisch generierten Referenzdaten auf den Testdatensätzen. Bei der Erzeugung werden die relevanten Fahrzeugheckansichten mittig mit dem Mauszeiger verfolgt und die Mauspositionen zusammen mit dem Bildzeitstempel abgespeichert. Bei der Auswertung gilt ein Fahrzeug als erkannt, wenn die Mitte der

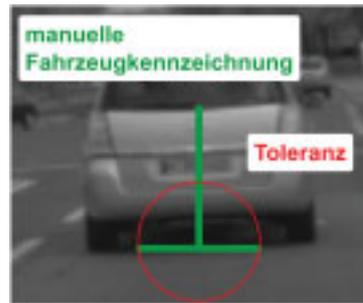


Bild 6.14: Schematische Darstellung der Labeltoleranz

Unterkante der Hypothese weniger als 30% der ins Bild projizierten angenommenen Fahrzeugbreite vom Referenzpunkt entfernt liegt. Dadurch werden auch Ungenauigkeiten bei der Erzeugung der Referenzpositionen im Bild berücksichtigt.

Vorbemerkungen

Vor Darstellung der Ergebnisse muss noch erwähnt werden, dass zur Berechnung der Detektionsraten nur Detektionen innerhalb eines Toleranzkorridors vor dem Eigenfahrzeug berücksichtigt sind. Der Toleranzkorridor ist nötig um die Aussagekraft des Wertes hinsichtlich der Systemspezifikation zu verbessern. Üblicherweise werden Fahrzeuge nur in diesem eingegrenzten Bereich tatsächlich von hinten wahrgenommen. Andere Bildbereiche liegen somit außerhalb der Systemspezifikation und müssen vernachlässigt werden. Durch die Breite des Korridors von 6 Metern (3 Meter zu jeder Seite) ist sichergestellt, dass im Fall einer Geradeausfahrt auf keinen Fall eine Kollision stattfinden kann. Die Voraussicht ist auf 100 Meter begrenzt, da ab dieser Entfernung der verwendete optische Vorfilter ohnehin nur noch unzuverlässige Ergebnisse liefert, die Systemgrenzen also erreicht sind. Die gewählten Bewertungsmaße werden im Anhang C.1 genauer beschrieben. Die Erkennungsraten werden im Weiteren in Prozent angegeben.

Zur Visualisierung der Messergebnisse wird das in Abbildung 6.15 erläuterte Schema gewählt. Die in der ROI-Generierung ermittelten Regionen werden durch die mit dicken Strichen eingezeichneten Kästen gekennzeichnet. PKW-ROIs sind dabei in gelb, mit einer zugeordneten Qualitätszahl gekennzeichnet. LKW-ROIs werden weiß ohne Qualitätszahl dargestellt. Die Qualitätszahl der PKW-ROIs zeigt an, ob eine ROI nur durch eine der bei der ROI-Generierung verwendeten Merkmalskombinationen (Symmetrie&Gradientenhäufung bzw. Unterkante&Schatten) oder durch beide bestätigt ist. Je nach Anzahl der Quellen für eine ROI wird folglich der Wert „1“ bzw. „2“ dargestellt. ROIs der Qualität „2“ werden zusätzlich durch einen dünnen roten Strich innerhalb des dicken gelben Strichs gekennzeichnet.

Zur Darstellung des Ergebnisses des Klassifikators werden positiv klassifizierte LKW-Hypothesen mit einem orangen Rechteck und PKW-Hypothesen mit einer roten Kontur gekennzeichnet.

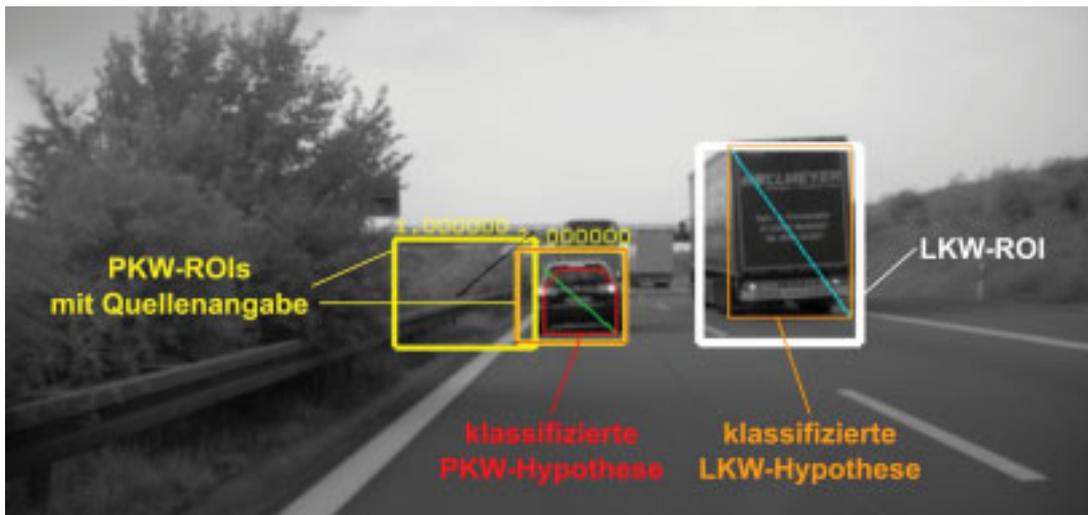


Bild 6.15: Farbschema zur Darstellung der Detektionsergebnisse des Gesamtsystems

Vergleich verschiedener Klassifikatorauslegungen

Als Klassifikationssystem wurde ein GRBF-Netz eingesetzt, das analog zu Kapitel 5.2 aufgebaut ist. Die Parameter des Netzes wurden basierend auf den Trainingsergebnissen eines RF-Klassifikators gesetzt (siehe Kapitel 5.1). Die sich ergebenden Zentren des GRBF-Netzes stellen jeweils „Prototypen“ für die entsprechenden Klassen dar. Das Zentrum eines zur Klasse PKW gehörigen Hyperellipsoids ist somit eine gemittelte Merkmalskonstellation für einen „Standard-PKW“. Tabelle 6.3 zeigt einen Vergleich der untersuchten Klassifikatorkonfigurationen auf einer Testdaten-Stichprobe, um die geeignetste Konfiguration auszuwählen. Die Bezeichnung $3M3Z$ beschreibt beispielsweise die Ergebnisse des Klassifikators der auf drei Merkmalen basiert und nur drei Zentren besitzt. Je ein Zentrum ist folglich den Klassen „irrelevant“, „PKW“ und „Kasten-LKW“ zuzuordnen. Der $3M3Z$ -Klassifikator ist, wie zu erwarten, der schnellste der untersuchten Systeme. Gleichzeitig weist er die niedrigste Fehlerrate auf. Der $4M3Z$ -Klassifikator erreicht zwar den höchsten D/B Wert, ist allerdings aufgrund der vier zu berechnenden Merkmale um 40% langsamer und liefert gleichzeitig um 28% mehr Fehlhypothesen als der $3M3Z$ -Klassifikator. Das im Vergleich zum $3M3Z$ -Klassifikator zusätzliche Merkmale des Gradientenverhältnisses weist also tendenziell einen positiven Effekt bei der Identifikation von vorhandenen Fahrzeugen auf, verursacht aber gleichzeitig zusätzliche Fehlhypothesen. Einen Mittelweg zwischen den beiden Klassifikatoren $3M3Z$ und $4M3Z$ stellt der $3M7Z$ -Klassifikator dar. Dieser muss zwar auch nur drei Merkmale auswerten, durch die Verwendung von sieben Zentren sind bei der Online-Klassifikation allerdings Matrizen der Dimensionen 7×7 an Stelle von 3×3 zu multiplizieren. Dies schlägt sich deutlich in der Rechenzeit³⁾ nieder.

Insgesamt betrachtet, rechtfertigt der Zugewinn von ca. 1% bei der Detektionsrate durch

³⁾ Die Berechnungszeiten beziehen sich auf einen Pentium 4 Prozessor mit 3 GHz Taktrate und Hyperthreading-Technologie auf einem WindowsXP Betriebssystem im Entwicklungsrahmensystem ADTF

Klassifikator	Mittl. Berechnungszeit in μs	D/B	F/B
3M3Z	4776	85.6784	18.14
4M3Z	6456	86.7108	23.17
3M7Z	5736	86.4147	19.66

Tabelle 6.3: Vergleich der getesteten Klassifikatoren

den Übergang von dem *3M3Z*-Klassifikator auf einen der beiden anderen Klassifikatoren nicht den deutlichen Rechenzeitzuwachs von 20% bzw. sogar 40%. Die frei werdenden Ressourcen könnten beispielsweise für ein Tracking der Hypothesen sinnvoller verwendet werden. Als zusätzlicher Punkt ist die im Vergleich niedrige Fehldetektionsrate des *3M3Z*-Klassifikators zu sehen. Aus diesen Gründen wird der *3M3Z*-Klassifikator für die weiteren Betrachtungen herangezogen.

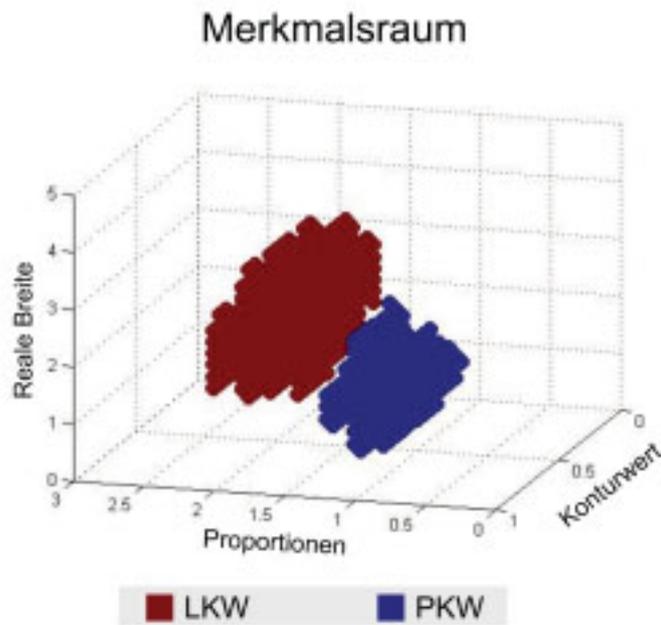


Bild 6.16: Klassifikatorentscheidungen im Merkmalsraum

Durch Zusammenfassung der Regionen für unbekannte und irrelevante Bereiche im Merkmalsraum zu einem Bereich „Nichts“ ergibt sich für den Fall des *3M3Z*-Klassifikators eine räumlichen Darstellung der Klassifikationsergebnisse im Merkmalsraum nach Abbildung 6.16. Zur Generierung dieser Veranschaulichung wurde der Merkmalsraum äquidistant mit virtuellen Merkmalsvektoren abgetastet und die Entscheidung des Klassifikators visualisiert.

Bewertung der gewählten Konfiguration

In diesem Abschnitt werden einige zusätzliche Auswertungen für den im vorhergehenden Abschnitt ausgewählten *3M3Z* Klassifikator durchgeführt. Dabei wird vor allem auf die Problemfälle bei der Detektion eingegangen und deren Ursachen analysiert.

D/B	79.35
D/O	73.02
F/B	16.70

Tabelle 6.4: Gemittelte Detektions- und Fehlerraten des Gesamtsystems

Betrachtet man zunächst die in Tabelle 6.4 dargestellten über alle Testszenarien gemittelten Werte (die individuellen Werte der Einzelszenarien sind gleich gewichtet), zeigt sich bereits, dass der mittlere *D/O*-Wert deutlich niedriger ist, als der mittlere *D/B*-Wert. Dies lässt sich damit begründen, dass vor allem Fahrzeuge, die überholt werden und nur kurz am Rand des Erfassungsbereichs des Systems auftauchen, schlechte Werte erzielen. Aufgrund der nur kurzen Zeit im Erfassungsbereich des Systems haben solche Fahrzeuge nur wenig Einfluss auf den über die gesamte Szene ermittelten *D/B*-Wert. Im *D/O*-Wert hingegen erhalten diese Fahrzeuge das gleiche Gewicht wie die länger vorhandenen, direkt vorausfahrenden Fahrzeuge. Da das System insbesondere auf Heckansichten abzielt, liegt der Systemschwerpunkt auf verfolgten oder leicht schräg versetzten Fahrzeugen. Dennoch wurden auch „grenzwertig schräge“ Heckansichten berücksichtigt, um die Systemgrenzen besser beurteilen zu können.

In den betrachteten Stadtszenarien fällt auf, dass jeweils die über einen Großteil der Szene hinweg verfolgten Fahrzeuge sehr gut erkannt werden. Die über diese Fahrzeuge gemittelte Detektionsrate ergibt sich zu $D/O = 90.43\%$. Damit werden diese Fahrzeuge signifikant besser erkannt als der Gesamtdurchschnitt. Dies begründet sich darin, dass sich die Fahrzeuge in niedriger bis mittlerer Entfernung vor dem Fahrzeug befinden. Handelt es sich zusätzlich um eine Verfolgungsfahrt mit gleichmäßiger Geschwindigkeit ist die Ebenenannahme für die Straße gut erfüllt. Damit erzielen sowohl Vorfilter als auch das Klassifikationssystem optimale Ergebnisse.

Betrachtet man die mittlere Fehlerrate von $F/B = 33.18\%$ der Stadtszenen im Vergleich zum Gesamtmittel, erkennt man, dass dieser Wert deutlich über dem Gesamtdurchschnitt liegt. Dies wird durch mehrere Effekte verursacht. Zum einen sind aus Aufwandsgründen gerade in Stadtszenen nicht alle auftretenden Heckansichten mit einem Referenzpunkt gekennzeichnet. Gerade kurz auftauchende Hecks am Rand des Wahrnehmungsbereichs (z. B. parkende Fahrzeuge) sind meist nicht referenziert. Werden diese Fahrzeuge richtig erkannt, erhöht sich dadurch die Fehlerrate. Zusätzlich werden auch viele Frontansichten von Fahrzeugen des Gegenverkehrs als Hecks identifiziert. Das wäre für viele Anwendungen, in denen das beschriebene Detektionssystem als Informationslieferant eingesetzt wird, unproblematisch.

Um den deutlichen Einfluss dieser Aspekte aufzuzeigen, wurde eine ausgewählte Szene vollständig gelabelt und nochmal ausgewertet. In dieser Szene reduziert sich die ursprüng-



Bild 6.17: Gegenverkehr erzeugt Fehldetektion

liche Fehlerrate von $F/B = 19.9\%$ auf $F/B = 6.1\%$. Es kann also eine Reduktion der Fehlerraten um ca. 70% durch Tolerierung der Detektionen im Gegenverkehr abgeschätzt werden. Überträgt man diese Abschätzung auf alle Stadtszenarien, verringert sich auch hier die Fehlerrate auf einen Wert von ungefähr $F/B = 11\%$.

Probleme bei normalen Folgefahrten auf Landstraßen und Autobahnen ergeben sich in den getesteten Szenarien nur, wenn die Ebenenannahme der Fahrbahnoberfläche schlecht erfüllt und der Abstand zum Vorderfahrzeug sehr groß ist. Einen Extremfall stellt beispielsweise das Einnicken aufgrund starker Bodenwellen innerorts dar. Hier werden teilweise sogar Fahrzeuge im mittleren Abstand nicht detektiert.



Bild 6.18: Fehlklassifikationen innerhalb von Heckansichten

Insbesondere bei der Detektion von LKWs muss noch ein weiterer Punkt betrachtet werden, der zur Erhöhung der Fehlerrate beiträgt. So kommt es vor, dass Fehlhypothesen innerhalb der eigentlichen LKW-Heckansicht detektiert werden. Diese Hypothesen können unterschiedlichen Charakter aufweisen. Entweder sie blitzen sporadisch an unterschiedlichen Positionen innerhalb der Heckansicht auf oder werden analog zum Fahrzeug in beinahe jedem Frame mitdetektiert. In Abbildung 6.18 finden sich Beispiele für solche „Mit-

6 Fahrzeugdetektion bei Tag und Nacht

fahrhypothesen“ . Diese könnten bei einem intelligenten Tracking nach einer Assoziation aufgrund gleicher Bewegung im Raum sogar als zusätzliche Stützen für LKW-Hypothesen verwendet werden, falls der LKW in einzelnen Frames nicht detektiert wird.

Abschließend ist noch anzumerken, dass Fremdfahrzeuge, die sich unmittelbar vor dem EGO-Fahrzeug befinden, mit der momentanen Auslegung des Moduls zur ROI-Generierung nicht detektiert werden können. In diesem Fall wäre eine andere Strategie erforderlich, da beispielsweise die markante Unterkante des Fahrzeugs nicht mehr wahrgenommen werden kann. So könnte in diesen Fällen ein zusätzlicher Nahbereichssensor (z. B. Ultraschall, Radar) eingesetzt werden, um die Videoerkennung zu stützen oder es muss auf andere Videomerkmale (z. B. Nummernschild) zur Aufmerksamkeitssteuerung zurückgegriffen werden.

6.4 Verfahren zur Fahrzeugdetektion bei Nacht

Da autonome Fahrerassistenzsysteme die gleichen Anforderungen an die Erkennung von Objekten bei Nacht wie auch bei Tag stellen, wird in dieser Arbeit ebenfalls untersucht, wie sich Bildverarbeitungsalgorithmen zur Erkennung von Fahrzeugheckansichten bei Nacht einsetzen lassen. Für den Anwendungsfall der Fahrzeugheckdetektion bei Nacht mit standard Videobildverarbeitung existieren bislang nur sehr wenige Ansätze. In [26] wird beispielsweise ein rein regelbasierter Detektionsansatz vorgestellt, wohingegen sich der Ansatz von [1] auf Support-Vector-Maschinen stützt. Der hier verwendete Ansatz geht wieder nach dem in den Abbildungen 6.1 und 6.2 dargestellten Prinzip vor. Auch hier werden, wie schon im vorangehenden Kapitel zur Fahrzeugdetektion bei Tag, die Vorteile einer Vorverarbeitung basierend auf Expertenwissen mit den Vorteilen eines Klassifikators auf Basis eines Maschinenlernverfahrens kombiniert.

6.4.1 Trainingsdatenerzeugung

Im Unterschied zum vorhergehenden Kapitel 6.3.1 werden hier keine aufwändigen Tools eingesetzt, um die Trainingsdaten zu generieren. Das liegt daran, dass die ROI-Generierung, die nachfolgend in Kapitel 6.4.2 beschrieben wird, auf Merkmalen aufbaut, die für eine plausible Fahrzeughypothese keine so hohe Anzahl an Kombinationsmöglichkeiten zulassen. Deshalb kann die ROI-Generierung selbst zur Generierung der Rohdatenbasis verwendet werden. Die extrahierten ROIs müssen dazu lediglich vom Benutzer den Klassen „Fahrzeugheck“ und „Nichts“ zugewiesen werden. Dies kann in der ADTF-Umgebung sehr leicht mit Hilfe des Video-Displays umgesetzt werden. Dabei wird pro Videoframe durch alle gefundenen ROIs iteriert. Das Bild wird zusammen mit der nötigen Zusatzinformation über die Lage der ROI, die extrahierten Merkmale und der zugewiesenen Klasse abgespeichert. Dies ergibt die Datenbasis für die Trainingsphase des Maschinenlernverfahrens.

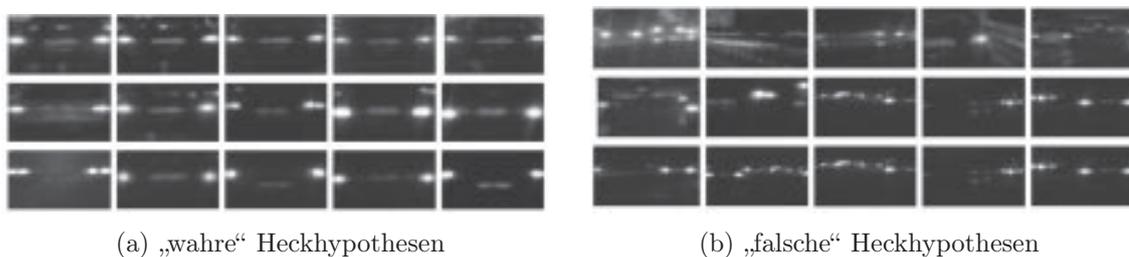


Bild 6.19: Beispielbilder aus Trainingsdatenbank

Der verwendete Trainingsdatensatz enthält 457 verschiedene Heckansichten und 331 Fehlhypothesen bei Nacht, die aus der ROI-Generierung stammen. Bild 6.19 zeigt einen kleinen Auszug der abgespeicherten Bilder. Mit diesem Datensatz findet dann das Training der PCA sowie des RF-Klassifikators statt.

6.4.2 ROI-Generierung

Wie schon im Fall der Heckerkennung bei Tag, soll mit Hilfe der ROI-Generierung zunächst eine Lokalisierung von Bildbereichen stattfinden, welche wahrscheinlich ein Fahrzeugheck enthalten. Die ROI-Generierung basiert ausschließlich auf dem Vorwissen über die Fahrzeuggeometrie in Kombination mit der Ebenenannahme. Die Bezeichnungen, die dabei eine wichtige Rolle spielen, sind in Abbildung 6.20 dargestellt. Tabelle 6.5 gibt eine kurze Beschreibung der verwendeten geometrischen Kennwerte.

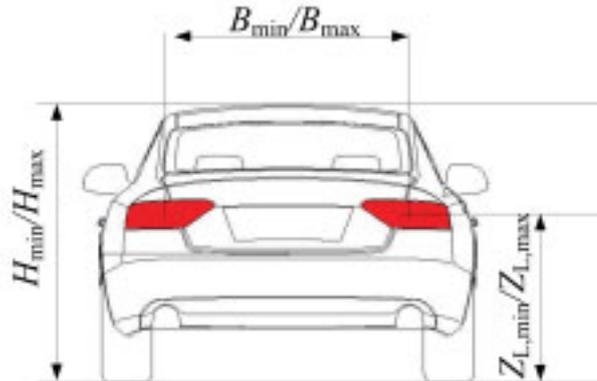


Bild 6.20: Darstellung der geometrischen Bezeichnungen eines Fahrzeughecks

Bezeichnung	Beschreibung
H_{\min}	minimale Fahrzeughöhe
H_{\max}	maximale Fahrzeughöhe
$Z_{L,\min}$	minimale Einbauhöhe der Rückleuchten
$Z_{L,\max}$	maximale Einbauhöhe der Rückleuchten
B_{\min}	minimaler Abstand der Rückleuchten
B_{\max}	maximaler Abstand der Rückleuchten

Tabelle 6.5: Beschreibung der geometrischen Kennwerte für Fahrzeughecks

Der Bezugspunkt für die Kennwerte $Z_{L,\min}$, $Z_{L,\max}$, B_{\min} und B_{\max} liegen jeweils im Mittelpunkt der Rückleuchten.

Die einzelnen Schritte zur Erzeugung der ROIs sind:

1. Einschränkung des Suchbereichs
2. Rücklichtdetektion
3. Einschränkung des Assoziationsbereichs
4. Bildung gültiger Lichtpaare
5. Detektion der Unterkante

6. Detektion der Oberkante

7. Plausibilisierung

Im Folgenden werden diese Schritte detailliert beschrieben.

Schritt 1: Einschränkung des Suchbereichs

Um nicht unnötig Rechenzeit zu verschenden, wird zuerst ein Fahrkorridor vorgegeben, in dem sich Fahrzeuge vollständig befinden müssen, damit sie vom System detektiert werden. Alle Fahrzeuge außerhalb des Suchbereichs werden von der Verarbeitungskette ignoriert. Der 3D-Suchbereich wird über die folgenden Parameter in Tabelle 6.6 definiert. In Abbildung 6.21 ist der relevante Suchbereich grün markiert.

Bezeichnung	Beschreibung
$X_{\min}^{(SB)}$	minimaler Abstand
$X_{\max}^{(SB)}$	maximaler Abstand
$Z_{\min}^{(SB)}$	minimale Höhe über Boden
$Z_{\max}^{(SB)}$	maximale Höhe über Boden
$B^{(SB)}$	maximaler lateraler Versatz zu beiden Seiten

Tabelle 6.6: Parameter des 3D-Suchbereichs

Schritt 2: Rücklichtdetektion

Zur Detektion der Rücklichter kommt der in Kapitel 4.3.2 beschriebene Ansatz zum Einsatz. Als Ergebnis der Lichtdetektion entsteht eine Liste heller Bildregionen, die potentiellen Lichtern von Fahrzeugen entsprechen. Diese werden im weiteren Verlauf als Lichtblobs LB bezeichnet und sind durch die Datenstruktur aus Tabelle 4.2 beschrieben. In Abbildung 6.21 sind die gefundenen Lichtblobs rot umrandet.

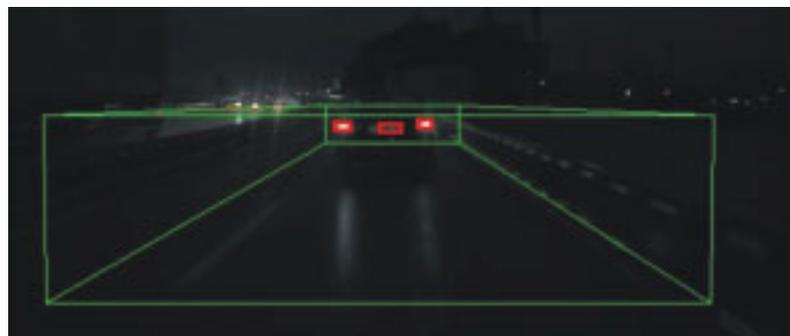


Bild 6.21: Detektierte Rücklichthypothesen im Suchbereich

6 Fahrzeugdetektion bei Tag und Nacht

Schritt 3: Einschränkung des Assoziationsbereichs

Grundlage für diesen Verarbeitungsblock bildet das Vorwissen, dass zwei korrespondierende Rückleuchten eines Fahrzeugs in der Bildebene keine beliebige Lage zueinander besitzen können. Ziel ist es, ausgehend von einem gefundenen Lichtblob, den Gültigkeitsbereich für einen dazu passenden Lichtblob einzuschränken, damit diese ein gültiges Lichtpaar ergeben können. Dazu wird das Vorwissen über den minimalen und maximalen Abstand von Fahrzeugrückleuchten (B_{\min} und B_{\max} in Abbildung 6.20), derer minimalen und maximalen Einbauhöhe $Z_{L,\min}$ bzw. $Z_{L,\max}$ im Fahrzeug zusammen mit der Kenntnis über die Einbaulage und die intrinsischen Parameter der Kamera (vgl. Kapitel 3.2.4) genutzt.

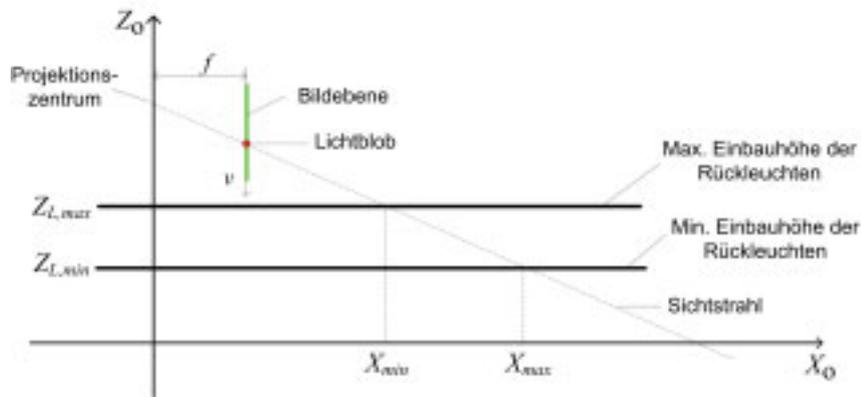
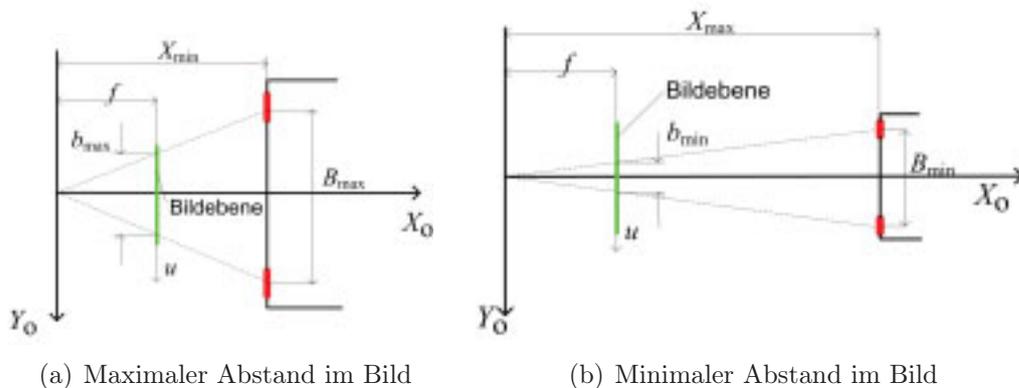


Bild 6.22: Einschränkung des plausiblen Distanzbereichs für eine Bildzeile v

Zur Bildung des Assoziationsbereichs für den zweiten Lichtblob, werden die Mittelpunkte der detektierten Lichtblobs unter Verwendung der Ebenenannahme in das Objektkoordinatensystem rückprojiziert. Als Besonderheit wird in diesem Fall nicht die Bodenebene als Schnittebene zur Entfernungsbestimmung verwendet, sondern die parallel verschobenen Ebenen $Z = Z_{L,\min}$ bzw. $Z = Z_{L,\max}$. Durch diese Rückprojektion, welche in Abbildung 6.22 schematisch dargestellt ist, ergeben sich die möglichen Extremwerte X_{\min} und X_{\max} der X-Koordinate einer zugehörigen Fahrzeughypothese.



(a) Maximaler Abstand im Bild

(b) Minimaler Abstand im Bild

Bild 6.23: Minimaler und maximaler Abstand gültiger Lichtpaare im Bild

Unter Verwendung der ermittelten Größen X_{\min} und X_{\max} lassen sich mit Hilfe von B_{\min} und B_{\max} , durch die Gleichungen 6.1 und 6.2, für jeden detektierten Lichtblob der minimale und maximale laterale Versatz b_{\min} bzw. b_{\max} eines zugehörigen Lichtblobs in Pixel berechnen.

$$b_{\min} = \frac{B_{\min} \cdot f}{X_{\max} \cdot s_u} \quad (6.1)$$

$$b_{\max} = \frac{B_{\max} \cdot f}{X_{\min} \cdot s_u} \quad (6.2)$$

Dabei ist s_u die Breite eines Pixels und durch f die Brennweite der Kamera in Meter.

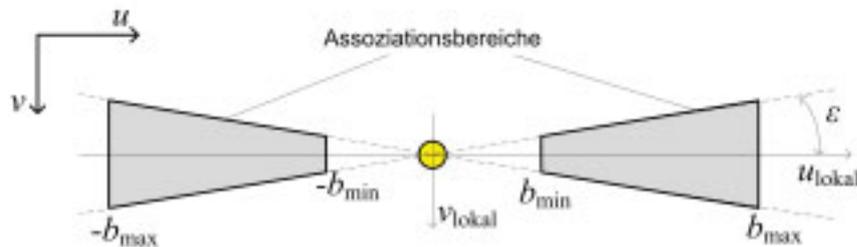


Bild 6.24: Schematische Darstellung des endgültigen Assoziationsbereichs

Ein weiteres Kriterium zur Festlegung des Assoziationsbereichs für ein zweites Licht eines Lichtpaares, ist der maximal zulässige vertikale Versatz der Lichter zueinander. Unter Einführung eines Toleranzwinkels ε lässt sich der maximal zulässige vertikale Versatz Δy_{\max} für den zuvor berechneten Abstandsbereich zwischen b_{\min} und b_{\max} , ausgehend von einem Referenzlicht, durch Gleichung 6.3 berechnen.

$$\Delta y_{\max} = b \cdot \sin(\varepsilon) \quad (6.3)$$

mit $b_{\min} < b < b_{\max}$

Überträgt man die entstehenden Grenzen in die vier Quadranten des lokalen Lichtblobkoordinatensystems, ergibt sich für jeden Lichtblob ein lokaler Assoziationsbereich für einen potentiellen Partner zur Bildung eines Lichtpaares. Dieser ist in Abbildung 6.24 dargestellt.

Schritt 4: Bildung gültiger Lichtpaare

Nach dem letzten Verarbeitungsschritt kann grundsätzlich jeder Lichtblob mit jedem anderen ein potentielles Lichtpaar bilden. Dadurch ergeben sich nach Gleichung 6.4 bei insgesamt N_{LB} relevanten Lichtblobs N_{LP} potentielle Lichtpaare.

$$N_{\text{LP}} = \frac{N_{\text{LB}}^2 - N_{\text{LB}}}{2} \quad (6.4)$$

6 Fahrzeugdetektion bei Tag und Nacht

Zwei Lichtblobs LB_i und LB_j , die sich innerhalb ihres Toleranzbereichs befinden, müssen, um ein gültiges Lichtpaar zu bilden, zusätzlich eine gewisse Ähnlichkeit zueinander besitzen. Dafür werden zwei Ähnlichkeitsmaße $\beta_{i,j}^{(h)}$ und $\beta_{i,j}^{(f)}$ mit den zugehörigen Bedingungen

$$\beta_{\min}^{(h)} < \beta_{i,j}^{(h)} < \beta_{\max}^{(h)} \quad \text{und} \quad \beta_{\min}^{(f)} < \beta_{i,j}^{(f)} < \beta_{\max}^{(f)} \quad (6.5)$$

eingeführt. $\beta_{i,j}^{(h)}$ und $\beta_{i,j}^{(f)}$ berechnen sich zu

$$\beta_{i,j}^{(h)} = \frac{h_{LB_i}}{h_{LB_j}} \quad (6.6)$$

$$\beta_{i,j}^{(f)} = \frac{f_{LB_i}}{f_{LB_j}}. \quad (6.7)$$

h_{LB_i} und f_{LB_i} bzw. h_{LB_j} und f_{LB_j} stehen dabei für die in Kapitel 4.3.2 beschriebenen Werte der Helligkeit und der Fläche des i -ten bzw. j -ten Lichtblobs. Paare, welche nicht beide Bedingungen der Ungleichungen 6.5 erfüllen, werden verworfen. Somit verbleiben nach diesem Schritt nur noch $N_{LP,red} \leq N_{LP}$ gültige Lichterpaare.

Schritt 5: Detektion der Unterkante

Als Verifizierung der verbleibenden Lichtpaarhypothesen wird für jedes Lichtpaar nach einer Unterkante des Fahrzeugs gesucht. So wie bei Tag die Fahrzeugunterkante sehr gut durch den Übergang von den dunklen Laufflächen der Fahrzeugreifen zum Teerboden gefunden werden kann, ist es bei Nacht möglich, die Fahrzeugunterkante durch die Reflexionen der Rückleuchten am Boden zu detektieren. Dafür kommt der Detektor für Bodenreflexionen aus Kapitel 4.3.3 zum Einsatz.

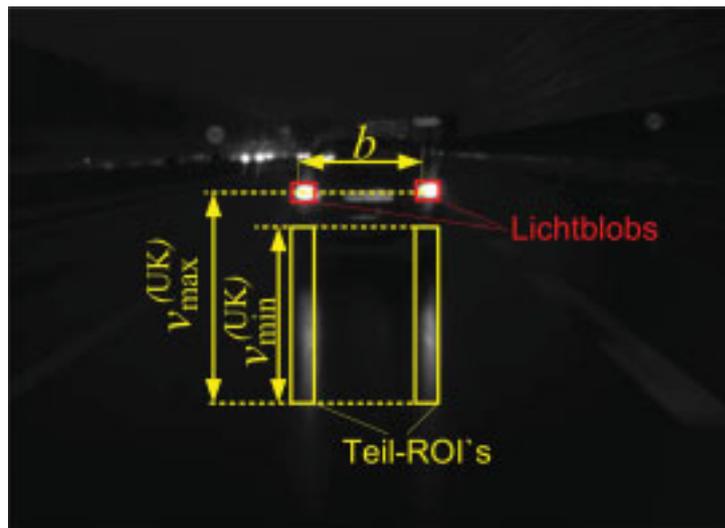


Bild 6.25: ROIs zur Unterkantensuche

Dafür müssen zunächst, basierend auf den Ergebnissen der vorherigen Schritte, geeignete ROIs für die Reflexionsdetektion aufgespannt werden. Für die Festlegung der ROI-Grenzen in Zeilenrichtung für den i -ten und j -ten Lichtblob werden die Variablen $u_{a,i}$ und $u_{e,i}$ bzw. $u_{a,j}$ und $u_{e,j}$, der Anfangs- und Endindizes der Bildspalten (aus Tabelle 4.2), übernommen. Die oberste und unterste Bildzeile der beiden ROIs, also die Grenzen in Spaltenrichtung, werden für beide ROIs gleich gesetzt und über folgende Abschätzungsformel ermittelt:

$$\Delta v_{i,j,\min}^{(\text{UK})} = \frac{\Delta u_{i,j}}{B_{\max}} \cdot Z_{L,\min} \quad (6.8)$$

$$\Delta v_{i,j,\max}^{(\text{UK})} = \frac{\Delta u_{i,j}}{B_{\min}} \cdot H_{\max} \quad (6.9)$$

$$(6.10)$$

$\Delta u_{i,j}$ ist der Abstand der beiden Lichtblobs des Lichtpaars. Die Brüche $\frac{\Delta u_{i,j}}{B_{\max}}$ und $\frac{\Delta u_{i,j}}{B_{\min}}$ stellen eine gezielte Unter- bzw. Überschätzung der Einheit $\frac{\text{Pixel}}{\text{Meter}}$ dar. Durch Multiplikation mit der minimalen Einbauhöhe der Rücklichter $Z_{L,\min}$ bzw. der maximalen Fahrzeughöhe H_{\max} ergibt sich ein für diese Anwendung optimaler Toleranzbereich.

$$v_{i,j,\min}^{(\text{UK})} = \min(v_{a,i}, v_{a,j}) + \Delta v_{i,j,\min}^{(\text{UK})} \quad (6.11)$$

$$v_{i,j,\max}^{(\text{UK})} = \max(v_{e,i}, v_{e,j}) + \Delta v_{i,j,\max}^{(\text{UK})} \quad (6.12)$$

$$(6.13)$$

Ausgehend vom obersten und untersten Punkt der beiden zugehörigen Lichtblobs kann durch die Gleichungen 6.11 und 6.12 sichergestellt werden, dass $v_{i,j,\min}^{(\text{UK})}$ noch überhalb und $v_{i,j,\max}^{(\text{UK})}$ schon innerhalb des Reflexionsbereichs im Bild liegt.

Somit ergeben sich die linke und rechte Teil-ROI in Abbildung 6.25 für den erwarteten linken und rechten Reflexionsbereich. Die Berechnungsvorschriften der Teil-ROIs sind nochmal in Tabelle 6.7 zusammengefasst. Dabei wird davon ausgegangen, dass der linke Lichtblob den Index i und der rechte den Index j besitzt.

	ROI _l	ROI _r
$u_{ROI,\min}$	$u_{a,i}$	$u_{a,j}$
$u_{ROI,\max}$	$u_{e,i}$	$u_{e,j}$
$v_{ROI,\min}$	$v_{i,j,\min}^{(\text{UK})}$	
$v_{ROI,\max}$	$v_{i,j,\max}^{(\text{UK})}$	

Tabelle 6.7: Berechnungsvorschriften der Teil-ROIs

Die Bereiche der beiden Teil-ROIs werden nun zu einer Gesamt-ROI zusammengefasst, welche dann das Eingangsdatenmuster des Bodenreflexionsdetektors darstellt. Die Schritte

6 Fahrzeugdetektion bei Tag und Nacht

zur Bestimmung der Unterkantenposition können in Kapitel 4.3.3 nachvollzogen werden. Der Bildzeilenindex $v_{i,j}^{(\text{UK})}$ der detektierten Unterkante wird für die weitere Verwendung gespeichert. Wird keine Unterkante gefunden, so wird das Lichtpaar als ungültig deklariert.

Schritt 6: Detektion der Oberkante

Für die Detektion der Oberkante ist ebenfalls ein spezieller Detektor entstanden. Dieser wird in Kapitel 4.3.4 genauer beschrieben. Für dessen Anwendung muss wiederum der Suchbereich für eine erwartete Dachkante auf Basis von Vorwissen eingegrenzt werden.

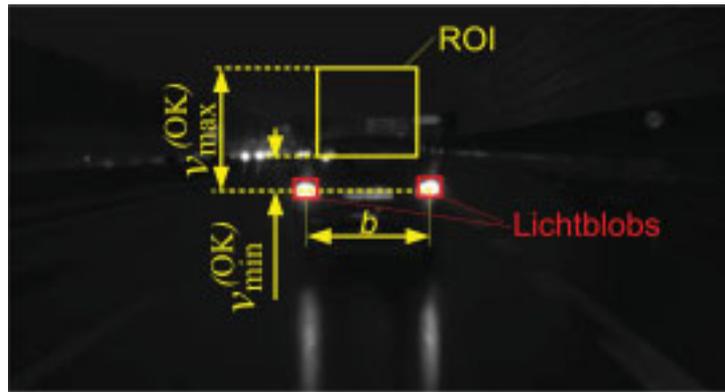


Bild 6.26: ROIs zur Oberkantensuche

Als Begrenzung in Zeilenrichtung werden die Innenkanten der zugehörigen Lichtblobs LB_i und LB_j genommen.

$$u_{i,j,\min}^{(\text{OK})} = u_{e,i} \quad (6.14)$$

$$u_{i,j,\max}^{(\text{OK})} = u_{a,j} \quad (6.15)$$

Zum Aufspannen des Suchbereichs in Spaltenrichtung wird wiederum eine Abschätzung durchgeführt.

$$v_{i,j,\min}^{(\text{OK})} = \min(v_{a,i}, v_{a,j}) - \frac{\Delta u_{i,j}}{B_{\min}} \cdot (H_{\max} - E_{\min}) \quad (6.16)$$

$$v_{i,j,\max}^{(\text{OK})} = \max(v_{a,i}, v_{a,j}) - \frac{\Delta u_{i,j}}{B_{\max}} \cdot (H_{\min} - E_{\max}) \quad (6.17)$$

Hierbei kommt nochmals die Unter- bzw. Überabschätzung der Einheit $\frac{\text{Pixel}}{\text{Meter}}$ durch den Quotienten aus gemessener Lichtabstand $\Delta u_{i,j}$ und maximaler bzw. minimaler angenommener Fahrzeugbreite B_{\max} bzw. B_{\min} zum Einsatz. Die anderen Variablen können aus den Tabellen 4.2 und 6.5 entnommen werden. Innerhalb den so erhaltenen ROI wird dann

der Dachkantendetektor aus Kapitel 4.3.4 ausgeführt und der Bildzeilenindex $v_{i,j}^{(\text{OK})}$ der Oberkante gemerkt. Wenn keine Oberkante gefunden werden kann, wird das Lichtpaar verworfen.

Schritt 7: Plausibilisierung

Mit Hilfe der detektierten Fahrzeugunterkante wird im letzten Modul der regelbasierten Hypothesengenerierung eine abschließende Plausibilisierung der verbleibenden Lichtpaare bzw. Fahrzeugheckhypothesen durchgeführt.

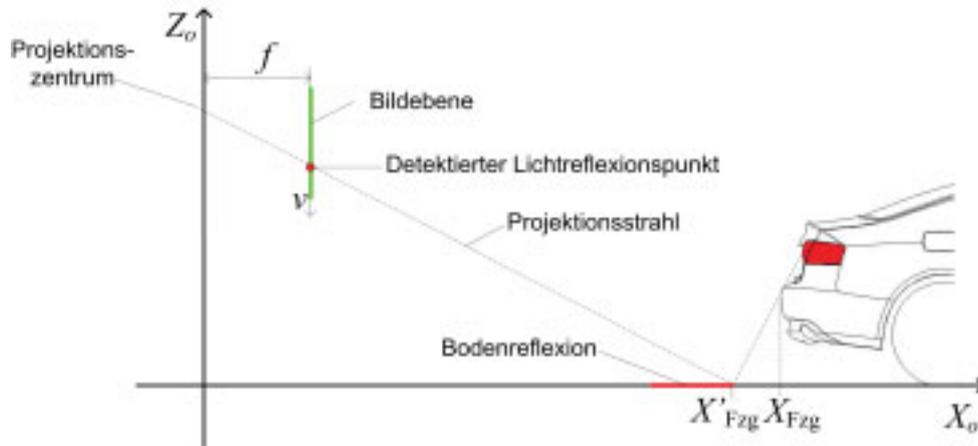


Bild 6.27: Entfernungsabschätzung mittels Lichtreflexion

Dazu wird wiederum das Mittel der Bodenebenenannahme zu Hilfe genommen. Über das Kameramodell wird der Sichtstrahl durch die in Schritt 5 gefundene Fahrzeugunterkante, mit der Bodenebene geschnitten. Daraus ergibt sich eine Schätzung des Abstandes X'_{Fzg} zum vorausfahrenden Fahrzeug. Wie Abbildung 6.27 zeigt, verursacht der Abschattungsbereich am Heck des Fahrzeugs eine Differenz zum wirklichen Abstand X_{Fzg} des Fahrzeugs. Dieser ist jedoch für die vorliegende Anwendung vernachlässigbar klein. Deshalb wird im Weiteren von

$$X_{\text{Fzg}} = X'_{\text{Fzg}}$$

ausgegangen.

Der gemessene Abstand $\Delta u_{i,j}$ eines Lichtblobpaares im Bild kann nun in den Abstand $B_{i,j}$ des Lichtpaares im Objektkoordinatensystem umgerechnet werden. Dazu wird $\Delta u_{i,j}$ über das Kameramodell mittels Gleichung 6.18 in die gemessene Entfernung X_{Fzg} projiziert.

$$B_{i,j} = \frac{\Delta u_{i,j} \cdot s_u}{f} \cdot X_{\text{Fzg}} \quad (6.18)$$

Dabei ist s_u die Pixelausdehnung in Bildzeilenrichtung und f die Brennweite des Lochkameramodells.

6 Fahrzeugdetektion bei Tag und Nacht

Als plausible Lichtpaare gelten nur solche, die die Bedingung der Ungleichung 6.19 erfüllen.

$$B_{\min} \leq B_{i,j} \leq B_{\max} \quad (6.19)$$

Ein Beispiel für die Verwerfung einer unplausiblen Hypothese ist in Abbildung 6.28 gezeigt.

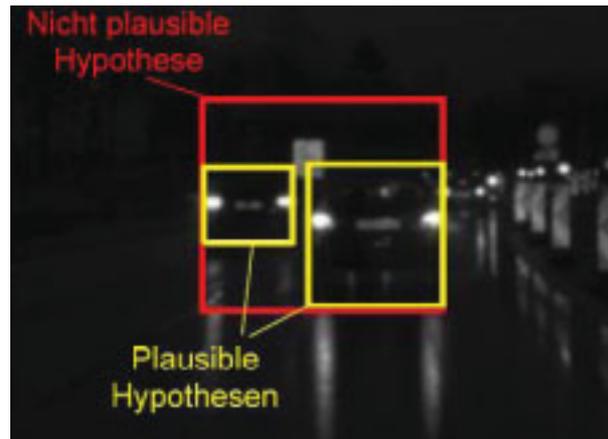


Bild 6.28: Unterscheidung zwischen plausiblen und unplausiblen Hypothesen

Durch diese Plausibilisierung reduziert sich Anzahl der $N_{LP,red}$ gültigen Lichtblobpaare zu

$$N_{LP,plaus} \leq N_{LP,red}$$

plausiblen Lichtpaaren für Fahrzeugheckansichten.

6.4.3 Objektdetektion mittels Klassifikation

Das Vorfiltermodul zur Hypothesengenerierung ist so ausgelegt, dass möglichst alle Rücklichtpaare als plausibel erkannt werden, die auch in Wirklichkeit einem Rücklichtpaar entsprechen. Die True-Positive-Rate soll also möglichst nahe an 100% liegen. Durch die tolerante Auslegung der Rückweisungsschwellen in der Vorfilterung, werden teilweise Bildausschnitte als Fahrzeugheck interpretiert, welche in Wirklichkeit keinem Fahrzeugheck entsprechen. Deshalb wird auch hier das Maschinenlernverfahren aus Kapitel 5.1 eingesetzt, um auf Basis eines höherdimensionalen Merkmalsvektors die Fehlerraten zu minimieren.

In diesem Fall soll lediglich die Klasse „PKW“ von der Klasse „kein PKW“ unterschieden werden. Hierbei wird wiederum der Fall „keine Aussage“ der Entscheidung „kein PKW“ zugewiesen.

Die Merkmale

Zunächst werden auf Basis der zuvor generierten plausiblen Fahrzeugheckhypothesen Merkmale generiert, von denen eine gute Klassentrennung zu erwarten ist. Die erzeugten Merkmale lassen sich prinzipiell in zwei Teile untergliedern. Der erste Teil beschreibt rein geometrische Merkmale, welche sich direkt aus den erzeugten Fahrzeugheckhypothesen der ROI-Generierung ableiten lassen. Der zweite Teil umfasst ansichtsbasierte Merkmale, die mittels Hauptachsentransformation gewonnen werden (siehe Kapitel 4.1.2).

Auch hier wird zur Optimierung der Komplexität des Klassifikators und zur Reduzierung der Rechenzeit eine Merkmalsselektion über die Feature Importance des Random-Forest-Klassifikators durchgeführt. Abbildung 6.29 zeigt die relativen Gewichtungen der einzelnen Merkmale zueinander.

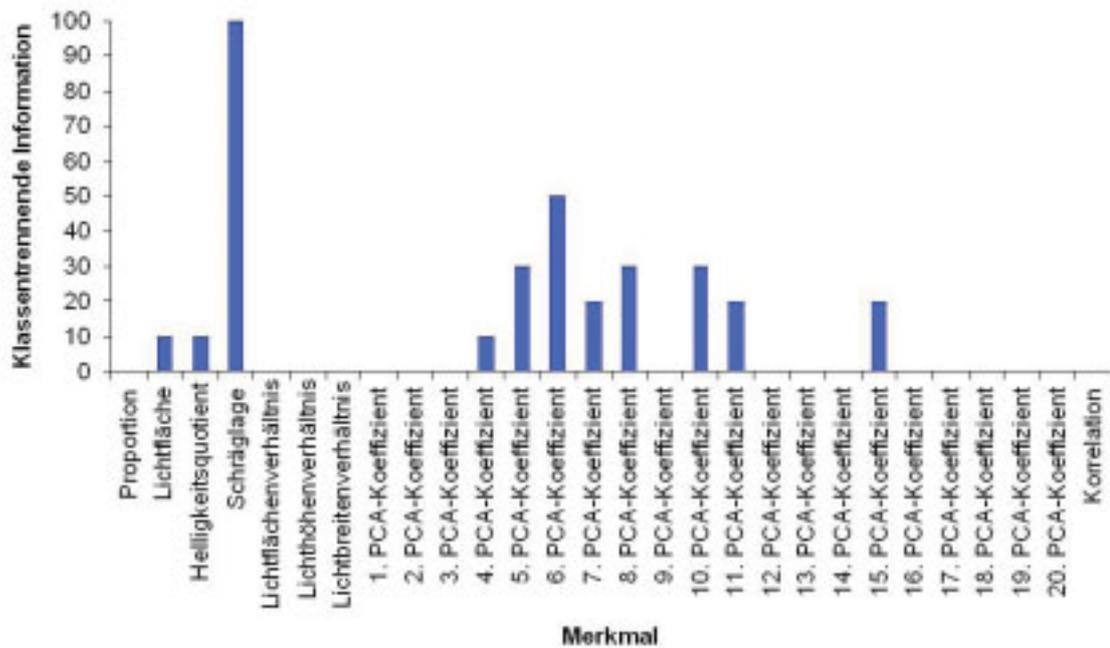


Bild 6.29: Feature Importance der generierten Merkmale

Für den Einsatz im Klassifikator kann so ein reduzierter Merkmalsvektor x'_{Nacht} aus den besten Merkmalen erzeugt werden.

$$x'_{\text{Nacht}} = (x_2, x_3, x_4, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{22})^T \quad (6.20)$$

Die einzelnen Merkmale des ursprünglich erzeugten, vollständigen Merkmalsvektors \underline{x} werden im Folgenden erläutert.

6 Fahrzeugdetektion bei Tag und Nacht

Merkmal „Proportion“ (x_1):

Der erste Eintrag im Merkmalsvektor ist das Verhältnis von Breite und Höhe der erzeugten Hypothesen in Beziehung. Die Höhe wird über die Zeilendifferenz der Ober- und Unterkante berechnet.

$$x_1 = \frac{\Delta u_{i,j}}{v_{i,j}^{(\text{UK})} - v_{i,j}^{(\text{OK})}}$$

Merkmal „Lichtfläche“ (x_2):

Merkmal x_2 normiert die gemittelte Fläche der beiden Lichtblobs auf die geschätzte Entfernung.

$$x_2 = \frac{f_i + f_j}{2 \cdot X_{Fzq}}$$

Merkmal „Helligkeitsquotient“ (x_3):

Merkmal x_3 setzt die Grauwertsumme der beiden Lichtblobs ins Verhältnis.

$$x_3 = \frac{h_i}{h_j}$$

Merkmal „Schräglage“ (x_4):



Bild 6.30: Merkmal „Schräglage“ der Hypothese

Merkmal x_4 ist der Betrag des Winkels zwischen der Verbindungslinie der Lichtblobs und der u -Achse des Bildkoordinatensystems (siehe Abbildung 6.30).

$$x_4 = \mu_{i,j} = \tan^{-1} \left(\left| \frac{v_{LB_i} - v_{LB_j}}{u_{LB_i} - u_{LB_j}} \right| \right)$$

Merkmal „Lichtflächenverhältnis“ (x_5):

Merkmal x_5 ist das Verhältnis der Flächen der beiden Lichtblobs zueinander.

$$x_5 = \frac{f_i}{f_j}$$

Merkmal „Lichthöhenverhältnis“ (x_6):

Merkmal x_6 betrachtet den Unterschied der Ausdehnung der beiden Lichtblobs in Spaltenrichtung.

$$x_6 = \frac{v_{e,i} - v_{a,i}}{v_{e,j} - v_{a,j}}$$

Merkmal „Lichtbreitenverhältnis“ (x_7):

Merkmal x_7 betrachtet den Unterschied der Ausdehnung der beiden Lichtblobs in Zeilenrichtung.

$$x_7 = \frac{u_{e,i} - u_{a,i}}{u_{e,j} - u_{a,j}}$$

Merkmale „PCA-Koeffizienten“ ($x_8 - x_{27}$):

Die Berechnungsvorschrift der PCA findet sich in Kapitel 4.1.2 und deren Anwendung für Bildverarbeitungszwecke wird in Kapitel 4.1.2 erläutert. Die Matrix \mathbf{U} und der Mittelwertvektor $\underline{\mu}$ werden basierend auf einem unabhängigen Trainingsdatensatz erzeugt. Die Bildausschnitte, welche zwischen der linken, rechten, oberen und unteren Kante der ROIs für plausible Fahrzeugheckhypothesen liegen, werden auf die Größe 30×30 Pixel interpoliert und in den Eigenraum unter Verwendung der Gleichung 4.19 transformiert. Die Werte der ersten 20 Einträge des Vektors \underline{x}_{PCA} werden den Merkmalen x_8 bis x_{27} zugewiesen. In diesen sollte der größte Anteil der klassenspezifischen Information komprimiert enthalten sein und somit eine gute Grundlage zur Unterscheidung von Ansichten anderer Objektklassen gegeben sein.

$$\begin{pmatrix} x_8 \\ \vdots \\ x_{27} \end{pmatrix} = \begin{pmatrix} x_{PCA,1} \\ \vdots \\ x_{PCA,20} \end{pmatrix}$$

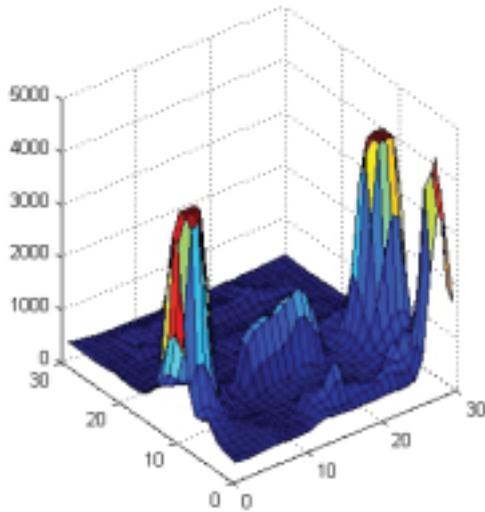
Merkmal „Eigenface“ (x_{28}):

Merkmal x_{28} enthält den Wert des Korrelationskoeffizienten aus Gleichung 4.21 zwischen dem Originalbild der Fahrzeugheckhypothese und der über die Hauptachsen rekonstruierten Variante.

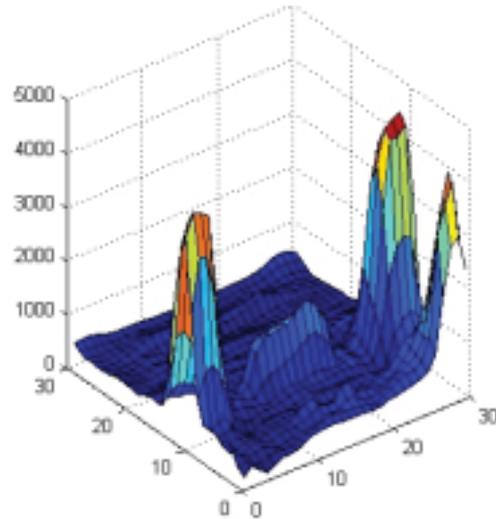
$$x_{28} = \varrho(\underline{x}_{\text{Bild}}, \underline{x}_{\text{Bild}}^*)$$

Dabei wird die Vorgehensweise des Eigenface-Ansatzes direkt auf Fahrzeugheckansichten übertragen. Zur Rekonstruktion des Bildes kommt die Gleichung 4.20 zum Einsatz unter Verwendung der ersten $K = 20$ Werte von \underline{x}_{PCA} .

6 Fahrzeugdetektion bei Tag und Nacht



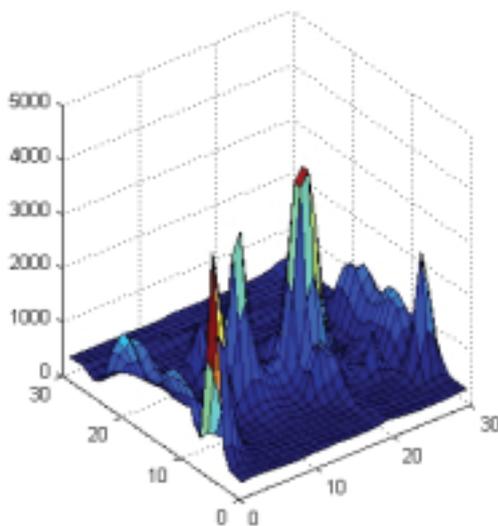
(a) Interpolierte Fahrzeugheckansicht



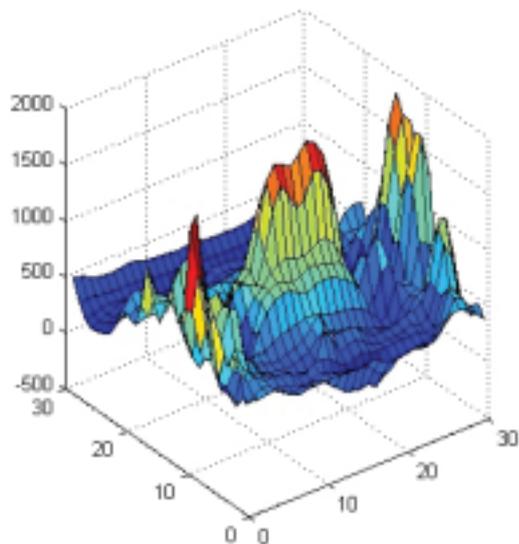
(b) Rekonstruktion mit $K = 20$

Bild 6.31: Vergleich zwischen Fahrzeugheckansicht und der Rekonstruktion

Abbildungen 6.31(a) zeigt die auf eine Größe von 30 Pixel interpolierte Version der Originalansicht eines Fahrzeughecks, das durch die ROI-Generierung gefunden wurde. In Abbildung 6.31(b) ist die rekonstruierte Variante über die ersten $K = 20$ Hauptachsen des Eigenraums dargestellt. Wie man sieht, ist die Ähnlichkeit zwischen Originalbild und Rekonstruktion sehr groß.



(a) Interpolierte Nicht-Fahrzeugheckansicht



(b) Rekonstruktion mit $K = 20$

Bild 6.32: Vergleich zwischen Nicht-Fahrzeugheckansicht und der Rekonstruktion

Bild 6.32(a) zeigt hingegen den interpolierten Ausschnitt eine fälschlicherweise generierten ROI. Abbildung 6.32(b) zeigt ebenfalls die rekonstruierte Variante über die ersten 20 Hauptachsen des Eigenraums. Hier fällt die Ähnlichkeit eher gering aus.

Der Klassifikator

Den Klassifikator für den Echtzeitbetrieb stellt, wie bei der Fahrzeugheckerkennung bei Tag, ein GRBF-Netz analog zu Kapitel 5.2 dar. Die Parameter werden wiederum aus den Ergebnissen der Trainingsphase des RF-Klassifikators gesetzt.

Auch hier besitzt das endgültig verwendete Netz jeweils ein Zentrum pro Klasse, also einen Prototypen für die Klasse „Fahrzeugheck“ und einen Prototypen für „kein Fahrzeugheck“. Als Entscheidungsgrundlage wird der Quotient der Mahalanobis-Distanzen $d_{\text{Heck}}(x'_{\text{Nacht}})$ und $d_{\overline{\text{Heck}}}(x'_{\text{Nacht}})$ des unbekanntes Datenmusters zu den beiden Prototypen der Klassen „Heck“ bzw. „Kein Heck“ berechnet. Der erhaltene Wert wird mit der Schwelle μ_{GRBF} verglichen, die es ermöglicht, das Verhalten des Klassifikators zu beeinflussen.

$$f_{\text{Nacht}}(\underline{x}_{\text{Nacht}}) = \begin{cases} \text{„Heck“} & \text{falls } \frac{d_{\text{Heck}}(\underline{x}_{\text{Nacht}})}{d_{\overline{\text{Heck}}}(\underline{x}_{\text{Nacht}})} > \mu_{\text{GRBF}} \\ \text{„Kein Heck“} & \text{sonst} \end{cases} \quad (6.21)$$

6.4.4 Bewertung des Verfahrens

Die regelbasierte ROI-Generierung hat zum Ziel, im betrachteten Bild Bereiche zu lokalisieren, welche potentiell ein Fahrzeugheck enthalten. Dabei ist wiederum wichtig, zunächst keine wirklich vorhandenen Fahrzeughecks zu verwerfen. Deshalb sind die Parameter aller Verarbeitungsblöcke bewusst tolerant gewählt. Die Tabellen in Anhang B fassen die verwendeten Parameter der verschiedenen Verarbeitungsblöcke zusammen. Zur Bewertung des Gesamtsystems werden die in Anhang C.2 angesprochenen Konfusionsmatrizen verwendet. Die Auswertung ist nicht so umfangreich wie die der Erkennung bei Tag. Dennoch haben die Ergebnisse durch die Wahl einer eher anspruchsvollen Teststrecke eine hohe Aussagekraft hinsichtlich der minimal zu erwartenden Erkennungsgüte im normalen Autobahnbetrieb. Die ausgewertete Szene ist eine Verfolgungsfahrt bei Nacht auf einer Autobahn in einer komplexen Baustellenumgebung. Die Szene beinhaltet 682 Einzelbilder, wobei das Vorderfahrzeug den für das System spezifizierten Bereich nie verlässt und somit in jedem Frame erkannt werden müsste.

ROI-Generierung

Die Tabelle 6.8 zeigt, dass das Fahrzeugheck tatsächlich in jedem Frame detektiert wird.

	Heck	Kein Heck	<i>Accuracy</i> =0.652
Als Heck erkannt	682	364	<i>Recall</i> =1.00
Als kein Heck erkannt	0	0	<i>Precision</i> =0.652

Tabelle 6.8: Konfusionsmatrix nach ROI-Generierung

Zusätzlich zu diesen 682 „wahren“ Hypothesen werden noch weitere 364 plausible Heckhypothesen erzeugt, die allerdings keiner Heckansicht entsprechen. Diese stellen Fehlhypothesen dar, welche durch Reflektoren an Baken und entgegenkommenden Verkehr erzeugt werden. Die Anzahl der Fehlhypothesen lässt die Komplexität der gewählten Szene erkennen. Für das Vorverarbeitungsmodul ergibt sich somit eine relativ niedrige Erkennungsrate von 0.652. Ausschlaggebend für die regelbasierte ROI-Generierung ist vor allem der Recall-Wert. Im Optimalfall hat dieser den Wert 1.0, was bedeutet, dass für jedes vorhandene Fahrzeugheck auch eine Hypothese erzeugt wird. Dieser Optimalfall wird in dieser Szene sogar erreicht. Es ist nun die Aufgabe des GRBF-Klassifikators, den Precision-Wert bei idealerweise gleich bleibendem Recall-Wert zu steigern.

Klassifikation

Im Klassifikationsteil ist zu beachten, dass neben dem RF-Klassifikator auch die PCA trainiert werden muss. Dabei muss darauf geachtet werden, dass kein Datensatz von beiden Klassifikatoren gleichzeitig gesehen wird, da sonst die PCA-Merkmale dem RF-Klassifikator zu übermäßig guten Ergebnissen verhelfen würden. In diesem Fall wurde die PCA auf Basis von 100 positiven Fahrzeugheckansichten trainiert und die verbleibenden Beispiele für das Training des RF-Klassifikators verwendet. Das aus dem RF-Klassifikator mit der Methode nach M. Botsch [20] gewonnene GRBF-Netz wird dann zur Klassifikation der Hypothesen verwendet.

Dadurch ergibt sich für die Daten aus den ROI-Generierung mit dem Klassifikationsschwellwert $\mu_{\text{GRBF}} = 0.5$ (siehe Gleichung 6.21), die Konfusionsmatrix aus Tabelle 6.9.

	Heck	Kein Heck	<i>Accuracy</i> =0.9378
Als Heck erkannt	659	42	<i>Recall</i> =0.9662
Als kein Heck erkannt	23	322	<i>Precision</i> =0.9400

Tabelle 6.9: Konfusionsmatrix nach Klassifikation

Es zeigt sich, dass mit 322 ein Großteil der insgesamt 364 Fehlhypothesen nach der ROI-Generierung nun korrekterweise der Klasse „kein Heck“ zugeordnet wird. Allerdings werden 23 generierte Hypothesen, welche in Wirklichkeit ein Fahrzeugheck darstellen, falsch klassifiziert, was zu einer Erkennungsrate von 93.78% führt.

6.5 Zusammenfassung zur Fahrzeugdetektion

In diesem Kapitel werden die entwickelten Verfahren zur bildbasierten Detektion von Fahrzeugheckansichten für Tag- und Nachtsequenzen detailliert beschrieben und einer statistischen Bewertung unterzogen. Beide Verfahren sind nach dem gleichen Grundsche-ma aufgebaut: Dabei werden in einem ersten Schritt auf Basis einfacher Primitive Re-gionen im Bild identifiziert, die wahrscheinlich ein Fahrzeugheck enthalten. Aus diesen Bildregionen werden in einem zweiten Schritt weitere Merkmale extrahiert und die ent-stehenden Merkmalskombinationen von einem vorher trainierten GRBF-Klassifikator der wahrscheinlichsten Klasse zugeordnet. Zur Merkmalsextraktion wird auf die in Kapitel 4 beschriebenen Operatoren und Detektoren zurückgegriffen.

Auf Basis einer Trainingsdatenbank wird ein RF-Klassifikator trainiert, der zugleich eine Merkmalsselektion durchführt, so dass nur die wichtigsten Merkmale bei der späteren Anwendung des Gesamtsystems berechnet werden müssen. Das zur Online-Klassifikation eingesetzte GRBF-Netz wird auf Basis des trainierten RF-Klassifikators erzeugt.

Die entstehenden Gesamtsysteme werden auf unbekanntem Testdatensätzen, die bei Real-fahrten mit dem Versuchsfahrzeug aufgezeichnet wurden, angewendet. Zum Bewerten der Gesamtsysteme werden Referenzdaten in den Testdatensätzen erzeugt und mit den vom Detektionssystem erzeugten Daten verglichen. Dabei stellt die Referenzdatenerzeugung eine große Herausforderung an den Entwickler dar. Es hat sich gezeigt, dass die erzeugten Referenzdatensätze nicht vollständig genug sind, um eine endgültige Aussage über die De-tektionsqualität treffen zu können. Jedoch kann eine gute Abschätzung der Qualität und der Möglichkeiten des Systems auf Basis der verwendeten Merkmale generiert werden.

7 Strategien zur robusteren Objektverfolgung

Die in Kapitel 6 erarbeiteten Verfahren können zum erstmaligen Detektieren von Fahrzeugheckansichten verwendet werden. Wie schon in Kapitel 4.4 angesprochen, können mit dem Wissen über den aktuellen Aufenthaltsort und der Objektklasse klassen- und situationsspezifische Dynamikmodelle verwendet werden, um bereits erkannte Objekte über die Zeit sehr effizient zu verfolgen. Dazu wird der in Kapitel 4.4.3 beschriebene 4D-Ansatz nach Dickmanns zusammen mit den Formmodellen aus Kapitel 4.4.1 und den Dynamikmodellen aus Kapitel 4.4.2 verwendet.

Im Folgenden wird lediglich auf die Erweiterungen des bereits bestehenden Systems eingegangen. Zunächst werden in Kapitel 7.1 die Erweiterungen des Fahrspurtrackers beschrieben. In Kapitel 7.2 wird auf die Erweiterungen des Fahrzeugtrackers zur multisensoriellen Objektverfolgung eingegangen.

7.1 Fahrspurtracking

Beim Verfolgen von Fahrspurmarkierungen auf Bundesstraßen und Bundesautobahnen ergeben sich in vielerlei Hinsicht große Herausforderungen an die Objekterkennung. So können...

- ... zum Teil sehr enge Kurven auftreten,
- ... Mehrdeutigkeiten durch andere Strukturen auf der Fahrbahn auftreten,
- ... Markierungen einseitig wegfallen und
- ... Verdeckungen durch andere Fahrzeuge auftreten.

Im Folgenden werden intuitive Lösungsstrategien vorgestellt, die dabei helfen, mit diesen Problemfällen besser umzugehen. Das Ziel der beschriebenen Vorgehensweisen ist stets, Fehlmessungen möglichst zu vermeiden und in den Fällen, wo Fehlmessungen auftreten, ihren Einfluss möglichst gering zu halten.

7.1.1 Angepasste Vorausschau

In engen Kurven¹⁾ kann das Problem auftreten, dass durch eine zu große Vorausschau der Bildverarbeitung Messpunkte auf Nebenspuren oder auf Hintergrundstrukturen liegen, die einer Spurmarkierung ähneln. Daraus ergeben sich Bildmerkmale, die nicht zur eigentlich verfolgten Spurmarkierung gehören und somit zu einer verfälschten Modellannahme über den Verlauf der Fahrspur führen. Abbildung 7.1 zeigt ein Beispiel einer Fehllassoziation durch den eben beschriebenen Problemfall.



Bild 7.1: Beispiel einer zu großen Vorausschauweite in einer engen Kurve

So liegt es nahe, die Vorausschauweite l an die aktuell vorliegende Krümmung $c_0(t_k)$ zum aktuellen diskreten Abtastzeitpunkt t_k anzupassen. Dazu wird zunächst eine minimale und maximale Vorausschauweite

$$l_{\min} = 20m \quad \text{und} \quad l_{\max} = 70m$$

festgelegt. Unter der Annahme, dass bei Kurvenradien unter $r_{\min} = 50$ Meter nur noch die minimale Vorausschau verwendet werden kann, wird der folgende krümmungsabhängige Interpolationsfaktor zwischen l_{\min} und l_{\max} berechnet.

$$i_c(t_k) = \begin{cases} 1 - \left| \frac{c_0(t_k)}{c_{0,\max}} \right| & \text{für } |c_0(t_k)| < c_{0,\max} \\ 0 & \text{sonst} \end{cases} \quad (7.1)$$

$$\text{mit} \quad c_{0,\max} = \frac{1}{r_{\min}} \quad (7.2)$$

Daraus ergibt sich die krümmungsabhängige Vorausschauweite $l_c(c_0)$ zu

$$l_c(t_k) = l_{\min} + i_c(t_k) \cdot (l_{\max} - l_{\min}). \quad (7.3)$$

¹⁾ Kurven mit großem Krümmungswert c_0 ($\frac{1}{c_0} \leq 200$) aus Gleichung 4.39 in Kapitel 4.4.1

7 Strategien zur robusteren Objektverfolgung

Zusätzlich zur Anpassung der Vorausschau über die Krümmung hat sich die Anpassung über die Geschwindigkeit als sinnvoll erwiesen. Das Fahren bei niedrigen Geschwindigkeiten ist meistens durch eine vorliegende komplexe Fahrsituation gegeben (z. B. Staufahrt, Ampelannäherung, kurvige Straße, ...). Auch in diesen Situationen führt eine zu weite Vorausschau zu einer erhöhten Messfehlergefahr. Daher wird zusätzlich ein geschwindigkeitsabhängiger Interpolationsfaktor $i_v(v)$ eingeführt.

$$i_v(t_k) = \begin{cases} 1 - \frac{v_{\max} - v(t_k)}{v_{\max} - v_{\min}} & \text{für } v_{\min} \leq v(t_k) \leq v_{\max} \\ 1 & \text{für } v(t_k) > v_{\max} \\ 0 & \text{für } v(t_k) < v_{\min} \end{cases} \quad (7.4)$$

Daraus wird die geschwindigkeitsabhängige Vorausschauweite $l_v(t_k)$ mittels Gleichung 7.5 berechnet.

$$l_v(t_k) = l_{\min} + i_v(t_k) \cdot (l_{\max} - l_{\min}). \quad (7.5)$$

Um das Risiko der Fehlmessung insgesamt zu minimieren, wird der kleinere der beiden Werte $l_c(t_k)$ und $l_v(t_k)$ als neue „Ziel-Vorausschauweite“ $l^*(t_k)$ gewählt.

$$l^*(t_k) = \min(l_c(t_k), l_v(t_k)) \quad (7.6)$$

Dieser Wert wird jedoch nicht direkt als neue Vorausschauweite $l(t_k)$ übernommen, da sich für $l_c(t_k)$ im Übergangsegment ($c_0 = 0$ in Abbildung 7.2) von S-Kurven starke Wertsprünge ergeben.

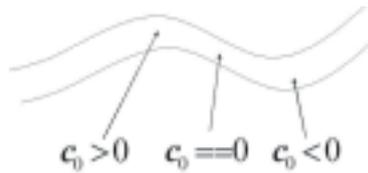


Bild 7.2: Nulldurchgang von c_0 im Übergangsegment einer S-Kurve

Deshalb wird noch eine Begrenzung der Änderung von der letzten verwendeten Vorausschauweite $l(t_{k-1})$ zur aktuellen $l(t_k)$ eingeführt. $l(t_k)$ berechnet sich somit nach der Fallunterscheidung aus Gleichung 7.7.

$$l(t_k) = \begin{cases} l(t_{k-1}) - \Delta l_{\text{neg,max}} & \text{für } l^*(t_k) - l(t_{k-1}) < -10 \\ l(t_{k-1}) + \Delta l_{\text{pos,max}} & \text{für } l^*(t_k) - l(t_{k-1}) > +3 \\ l^*(t_k) & \text{sonst} \end{cases} \quad (7.7)$$

Dabei sind die Werte für $\Delta l_{\text{neg,max}} = 10$ und $\Delta l_{\text{pos,max}} = 3$ gewählt. Dies trägt der gewünschten vorsichtigeren Umgehensweise mit der Vergrößerung der Vorausschauweite – z. B. im Bereich von S-Kurven – im Vergleich zur Zurücknahme der Vorausschauweite – z. B. bei der plötzlichen Einfahrt in eine enge Kurve – Rechnung.

7.1.2 Gruppierung von Markierungsmerkmalen

Eine weitere Möglichkeit zur Reduzierung von Fehlmessungen während des Trackings ist die Einführung einer modellangepassten Messwertgruppierung, bevor die Einzelmessungen im Innovationsschritt des Kalman-Filters verwendet werden. Somit wird die Assoziation zwischen prädizierten und gemessenen Merkmalen nicht mehr einzeln für jedes Merkmal durchgeführt, sondern auf der gesamten Merkmalsgruppe. Die Einzelmessungen werden also immer im Kontext mit anderen Messungen betrachtet, so dass für die Messwertauswahl mehr Information zur Verfügung steht. Gerade bei der Fahrspurerkennung kann dieses Vorgehen sehr gut eingesetzt werden, da der Fahrspurverlauf als weitestgehend stetig angenommen werden kann und somit nicht beliebig örtlich zueinander liegende Messungen möglich sind. Im Folgenden wird zunächst kurz die bestehende Vorgehensweise der videobasierten Merkmalsextraktion im vorliegenden Fahrspurtracker erklärt, um anschließend die Vorgehensweise zur Messwertgruppierung besser beschreiben zu können.

Bestehendes Messverfahren

Entlang des Fahrspurmodells aus Kapitel 4.4.1 wird an jedem Modellpunkt eine lokale Messung mit dem parametrisierbaren Kantenoperator aus Kapitel 4.1.1 durchgeführt. Die Parameter werden so gewählt, dass die Spaltenrichtung der Operatormaske parallel zur erwarteten Spurtangente zeigt und die Tiefpasseigenschaften des Operators optimal an den Entfernungsbereich des Modellpunktes angepasst sind. Für jede Messung um einen Modellpunkt erhält man also einen Ergebnisvektor der Operatorantworten (im Folgenden Operatorverlauf genannt). Aus diesem muss dann noch das eigentliche Markierungsmerkmal extrahiert werden. Dazu werden die Maxima und Minima im Operatorverlauf gesucht und deren Position im Originalbild gespeichert. Ein Maximum entspricht somit einem ausgeprägten positiven Grauwertgradienten und stellt den Dunkel-Hell-Übergang (DH-Übergang) von einem dunklen zu einem hellen Bildbereich dar. Ein Minimum hingegen entspricht einem ausgeprägten negativen Grauwertgradienten und somit einem Hell-Dunkel-Übergang (HD-Übergang).

Für Spurmarkierungsmerkmale wird nun erwartet, dass ein HD-Übergang direkt auf DH-Übergang folgt. Das ist an den Übergang vom dunklen Fahrbahnbelag zur weißen Spurmarkierung gefolgt vom Übergang zurück auf den Bodenbelag angelehnt. Die Merkmalsposition für ein Spurmarkierungsmerkmal (DHD-Merkmal) wird dann in die Mitte zwischen den beiden Kantenmerkmalen gelegt. Dabei darf kein anderer Extremwert dazwischen liegen, da von einer gleichmäßigen weißen Markierungsfarbe ausgegangen wird. Abbildung 7.3 stellt die Merkmale schematisch dar.

7 Strategien zur robusteren Objektverfolgung

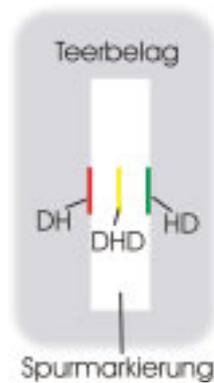
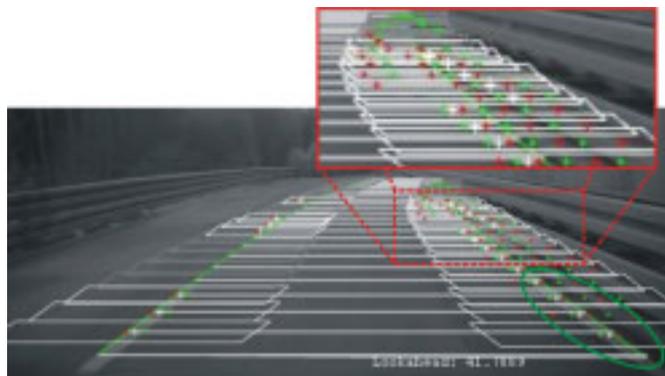
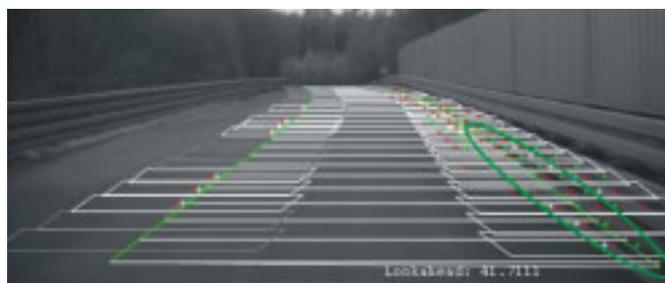


Bild 7.3: Bildmerkmale zur Spurerkennung

Im Bereich von inhomogenen Fahrbahnbelägen, ausgeprägtem Schattenwurf oder komplexen Fahrspurverläufen kann es vorkommen, dass pro Messfenster mehrere solcher Fahrspurmerkmale oder ausschließlich Fehlmerkmale gefunden werden, so dass die richtigen Merkmale ausgewählt und/oder falsche Merkmale verworfen werden müssen. Bei der Merkmalsselektion werden häufig diejenigen Fahrspurmerkmale mit den ausgeprägtesten Gradientenübergänge ausgewählt oder einfach diejenigen, welche am nächsten zum prädierten Merkmal liegen. Wie in Abbildung 7.4(a) zu sehen ist, führen diese Herangehensweisen nicht immer zum optimalen Ergebnis.



(a) Ausgangssituation



(b) Resultierende Situation

Bild 7.4: Messfehler bei Verwendung von Einzelmessungen

Abbildung 7.4 zeigt ein Beispiel für mehrdeutige Merkmale pro Messfenster. Die grünen durchgezogenen Linien in Richtung des Fahrbahnverlaufs sind die Projektion der linken und rechten Begrenzung der prädizierten Fahrspurklothoide. Die weißen Parallelelogramme stellen die Messfenster dar, in denen nach Kantenmerkmalen gesucht wird. Grüne Kreuze repräsentieren DH-Merkmale und rote HD-Merkmale. Weiße Kreuze stellen die ausgewählten Markierungsmerkmale dar, die in der Mitte zwischen zwei DH- und HD-Merkmalen liegen. In der Ausgangssituation in Bild 7.4(a) sieht man bereits im rot umrahmten Bereich, dass einige Merkmale falsch zugeordnet werden. Für einzelne Messungen werden manchmal die Strukturen in den Leitplanken eher als Spurmarkierungsmerkmal interpretiert als die Spurmarkierung selbst. Im grünen Kreis ist die resultierende Abweichung der Klothoide feststellbar. In Abbildung 7.4(b) führt das zu noch größeren Abweichungen des geschätzten Fahrspurverlaufs von der Realität, woraus sich noch mehr Fehllassoziationen ergeben.

Messwertgruppierung

Um diese Fehllassoziationen zu vermeiden, wird das Erkennungssystem um eine, speziell an die Fahrspurerkennung angepasste, Merkmalsgruppierungsfunktion erweitert. Dabei sollen nur Messungen als gültige Messungen angesehen werden, die einer Merkmalskette angehören, welche zum prädizierten Fahrspurmodell passt. Die dafür entwickelte Verkettungsstrategie für Spurmerkmale wird im Absatz *Verkettungsstrategie* und die anschließenden Merkmalsauswahl in Absatz *Onlinebewertung der Hypothesen* beschrieben.

Verkettungsstrategie Im Folgenden wird nur eine Seite des zweiseitigen Fahrspurmodells betrachtet, da die Vorgehensweise für beide Seiten identisch ist. Ziel der Verkettung ist die Gruppierung von einzelnen Spurmarkierungsmerkmalen zu Merkmalsketten, die einer Fahrspurmarkierung entsprechen könnten. Zudem sollen auch lange Messpunktketten eines bestimmten Typs (Dunkel-Hell bzw. Hell-Dunkel) erzeugt werden, die bei fehlenden Spurmarkierungen als alternative Merkmale für den weiteren Spurverlauf genutzt werden können.

Ausgangspunkt der Verkettung ist das Vorliegen der $N_p^{(DH)}$ Maxima (Dunkel-Hell-Übergang) und $N_p^{(HD)}$ Minima (Hell-Dunkel-Übergang) in den Ergebnisvektoren des Kantenoperators für alle P Modellpunkte \underline{P}_p des Spurmodells. Diese werden direkt als Merkmale $M_{p,n_{DH}}^{(DH)}$ bzw. $M_{p,n_{HD}}^{(HD)}$ weiter verwendet, mit $n_{DH} \in \{1, \dots, N_p^{(DH)}\}$ und $n_{HD} \in \{1, \dots, N_p^{(HD)}\}$. $i_{p,n_{DH}}^{(DH)}$ bzw. $i_{p,n_{HD}}^{(HD)}$ bezeichnen die Messwertindizes der Extremwerte vom Typ DH bzw. HD im Ergebnisvektor aus der Messung um den p -ten Modellpunkt. $p_{p,n_{DH}}^{(DH)}$ bzw. $p_{p,n_{HD}}^{(HD)}$ stehen für die zugehörigen 2D-Bildpositionen.

Im ersten Schritt werden alle Extrema durchlaufen um daraus neue Messpositionen $p_{p,n_{DHD}}^{(DHD)}$ für Markierungsmerkmale zu generieren. Dabei werden die Bildpunkte der Extrema über das Kameramodell (Kapitel 3.2.4) und die Bodenebenenannahme (Abbildung 3.13) in

7 Strategien zur robusteren Objektverfolgung

Positionen $\underline{P}_{p,n}$ im Objektkoordinatensystem umgerechnet. Für das neu erzeugte Markierungsmerkmal $M_{p,n_{\text{DHD}}}^{(\text{DHD})}$ an der Position

$$\underline{P}_{p,n_{\text{DHD}}}^{(\text{DHD})} = 0.5 \cdot (\underline{P}_{p,n_{\text{DH}}}^{(\text{DH})} + \underline{P}_{p,n_{\text{HD}}}^{(\text{HD})}),$$

müssen folgende Bedingungen erfüllt sein:

$$\begin{aligned} i_{p,n_{\text{HD}}}^{(\text{HD})} > i_{p,n_{\text{DH}}}^{(\text{DH})} \quad \wedge \quad B_{\min}^{(\text{DHD})} < B_{p,n_{\text{DHD}}} < B_{\max}^{(\text{DHD})} \\ \text{mit} \quad B_{p,n_{\text{DHD}}} &= \|\underline{P}_{p,n_{\text{HD}}}^{(\text{HD})} - \underline{P}_{p,n_{\text{DH}}}^{(\text{DH})}\|. \end{aligned} \quad (7.8)$$

Um die möglichen Fälle von Autobahnausfahrten bis hin zu schlecht markierten Landstraßen abzudecken, sind die Werte $B_{\max}^{(\text{DHD})} = 0.05$ und $B_{\min}^{(\text{DHD})} = 0.5$ Meter gesetzt.

Im zweiten Schritt wird die eigentliche Verkettung der Merkmalstypen „DH“, „HD“ und „DHD“ durchgeführt. Diese ist als doppelt verkettete Liste implementiert, so dass von jedem Merkmal zu seinem Nachfolger und Vorgänger in der eigenen Kette gesprungen werden kann. Der Anfang bzw. das Ende einer Kette sind dadurch gekennzeichnet, dass kein Vorgänger bzw. Nachfolger in der Liste existiert.

Die Kette wird von Nah nach Fern aufgebaut, das heißt, es wird am Modellpunkt \underline{P}_0 begonnen, welcher am nächsten am Ego-Fahrzeug liegt. Punkte mit einer X-Komponente von $X_p < X_{\max}$ werden auf die Bedingungen für einen Kettenstart überprüft. Modellpunkte mit einer größeren Distanz werden als möglicher Kettenstart vernachlässigt, da davon ausgegangen werden kann, dass Spurmarkierungen und sonstige markante Strukturen auch im Nahbereich sichtbar sein müssen und in größeren Entfernungen eher unzuverlässig sind. Die gefundenen Kettenstarts werden dann in einem rekursiven Verfahren unabhängig vom prädierten Formmodell weiter verkettet. Die Schritte zur Kettenerzeugung werden nun kurz erläutert. Im Anhang D wird der implementierte Algorithmus mittels Pseudocode genau beschrieben. Dabei wird das Winkelkriterium aus Gleichung 7.9 verwendet.

$$\text{QUALDELTAANGLE}(\alpha_1, \alpha_2) = \begin{cases} 100 - \frac{|\alpha_2 - \alpha_1|}{\Delta\alpha_{\max}} \cdot 100 & \text{für } |\alpha_2 - \alpha_1| < \Delta\alpha_{\max} \\ 0 & \text{sonst} \end{cases} \quad (7.9)$$

Bei der Suche nach Kettenanfängen ist α_1 der Winkel der Verbindungslinie der beiden Messpunkte mit der X-Achse des Objektkoordinatensystems und α_2 der Winkel der Verbindungslinie der zugehörigen prädierten Modellpunkte, wie in Abbildung 7.5(a) abgebildet. Hierbei ist die Toleranzgrenze $\Delta\alpha_{\max} = 5^\circ$ gewählt. Daraus ergeben sich Paare von gemessenen Merkmalen, die als Kettenstart verwendet werden können.

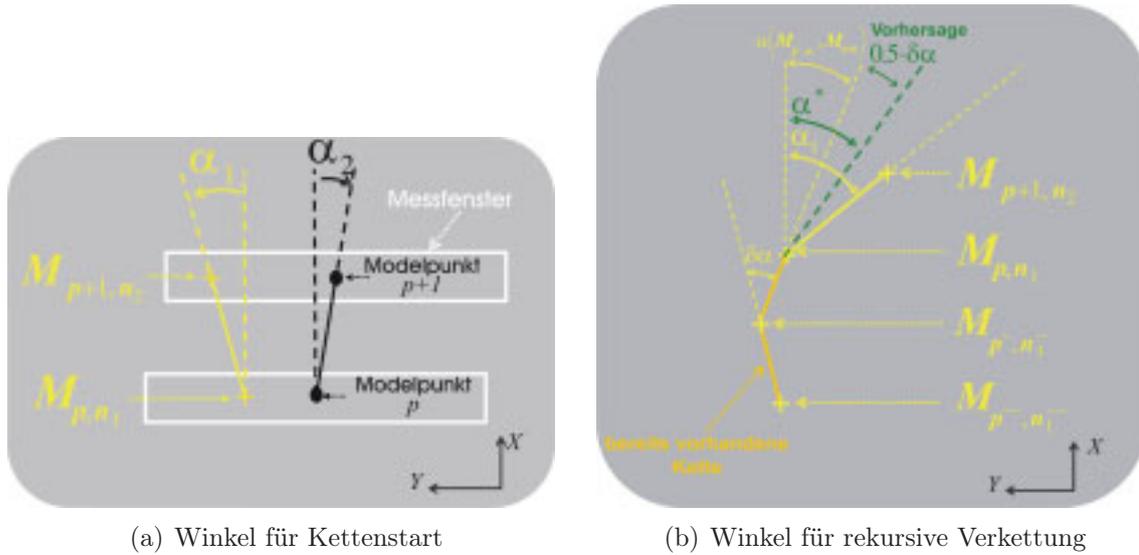


Bild 7.5: Betrachtete Winkel bei Verkettung

Ausgehend von den gefundenen Merkmalspaaren für einen Kettenstart werden die nachfolgenden Modellpunkte nach weiteren passenden Messpunkten durchsucht. Dieses Vorgehen ist in Algorithmus D.2 beschrieben. Dabei wird wiederum das Qualitätsmaß nach Gleichung 7.9 herangezogen. α_2 wird jedoch durch den prädizierten Winkel α^* ersetzt. Die Berechnung dieses Winkels ist schematisch in Abbildung 7.5(b) dargestellt und in Gleichung 7.10 mathematisch formuliert.

$$\alpha^* = \alpha \langle M_{p^-,n_1}^{(\text{type})}, M_{p,n_1}^{(\text{type})} \rangle + 0.5 \cdot \delta\alpha \quad (7.10)$$

$\alpha \langle \cdot, \cdot \rangle$ bezeichnet den Winkel der Verbindungslinie der beiden in spitzen Klammern stehenden Merkmalen mit der X-Achse. $M_{p,n_1}^{(\text{type})}$ ist das letzte Merkmal einer bereits bestehenden Kette vom Typ type am p -ten Modellpunkt. $M_{p,n_1}^{(\text{type})}$ und $M_{p,n_1^-}^{(\text{type})}$ sind die beiden Vorgängermerkmale in der Kette. Bei der rekursiven Verkettung wird, wie schon bei der Erzeugung des Kettenstarts, das beste Nachfolgemerkmal $M_{p+1,n_2}^{(\text{type})}$ mit dem aktuellen Merkmal $M_{p,n_1}^{(\text{type})}$ verbunden.

Onlinebewertung der Hypothesen Die im vorhergehenden Schritt entstehenden N_C Ketten C_c , mit $c = 1, \dots, N_C$, und die zugehörigen M_c Merkmale $M_{m_c}^{(C_c)}$, mit $m_c = 1, \dots, M_c$, werden nachfolgend einer statistischen Analyse unterzogen. Dabei entsteht die Datenstruktur nach Tabelle 7.1 zur Beschreibung einer Kette C_c .

Basierend darauf werden die beiden Gütemaße nach Gleichungen 7.11 und 7.12 berechnet. P ist dabei die Anzahl der Messpunkte entlang der Spurmodellkante.

7 Strategien zur robusteren Objektverfolgung

Bezeichnung	Beschreibung
N_c	Anzahl an Messpunkten, die zu einer Kette zusammengefügt sind.
l_c	X-Komponente des Vektors zwischen Anfangs- und Endpunkt der Kette in Meter
$\overline{\Delta Y}_c$	Mittelwert der seitlichen Ablage der Messpunkte zu den Modellpunkten
$\overline{\Delta x}_c$	Mittelwert der Absolutbeträge der Residuen der Messpunkte im Bild normiert auf die Messfensterbreite
\overline{B}_c	Mittelwert der gemessenen Merkmalsbreiten (nur für DHD-Merkmale)

Tabelle 7.1: Beschreibung der Datenstruktur einer Merkmalskette

$$Q_{\text{Count},c} = \frac{N_c}{P} \cdot 100 \quad (7.11)$$

$$Q_{\text{Res},c} = (1 - \overline{\Delta x}_c) \cdot 100 \quad (7.12)$$

Diese werden noch zu einer heuristisch ermittelten Gesamtgüte Q'_c für die c -te Kette gemäß Gleichung 7.13 zusammengeführt.

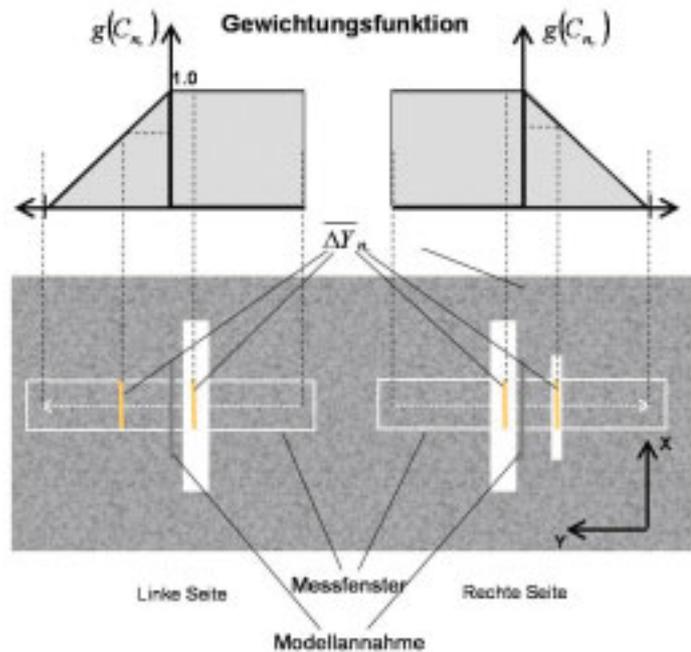
$$Q'_c = \frac{2 \cdot Q_{\text{Count},c} + Q_{\text{Res},c}}{3} \quad (7.13)$$

Da es bei schlecht ausgebauten Straßen häufig viele Strukturen im Randbereich, außerhalb der eigentlichen Spurbegrenzungen, gibt, hat sich in praktischen Tests die Vorgehensweise als sinnvoll herausgestellt, Strukturen, die näher an der Skelettlinie des Spurmodells liegen (also weiter innen), stärker zu gewichten. Das hilft auch für den in Abbildung 7.4 dargestellten Problemfall. Deshalb wird abschließend die Güte Q'_c mit dem Wert der Gewichtungsfunktion $g(C_c)$ multipliziert, der von der relativen Lage der Kette zur Referenzklothoide der Modellannahme abhängt. Dies ist in Abbildung 7.6 schematisch dargestellt.

Zur Berechnung der Funktion $g(C_c)$ wird die Variable $\overline{\Delta Y}_c$ aus Tabelle 7.1 herangezogen, so dass sich der endgültige Gütewert Q_c nach Gleichung 7.14 berechnet.

$$Q_c = g(C_c) \cdot Q'_c \quad (7.14)$$

Die Gütefunktion befolgt dabei die Regel 7.15.

Bild 7.6: Schematische Darstellung der Gewichtungsfunktion $g(C_c)$

$$g(C_c) = \begin{cases} 1.0 - \frac{\overline{\Delta Y_c}}{\Delta Y_{\max}} & \text{für } \overline{\Delta Y_c} > 0 \wedge \text{linke Seite} \\ 1.0 + \frac{\overline{\Delta Y_c}}{\Delta Y_{\max}} & \text{für } \overline{\Delta Y_c} < 0 \wedge \text{rechte Seite} \\ 1.0 & \text{sonst} \end{cases} \quad (7.15)$$

Hierbei ist eine Unterscheidung zwischen der linken und rechten Seite zu machen, da das Vorzeichen von $\overline{\Delta Y_c}$ korrekt berücksichtigt werden muss.

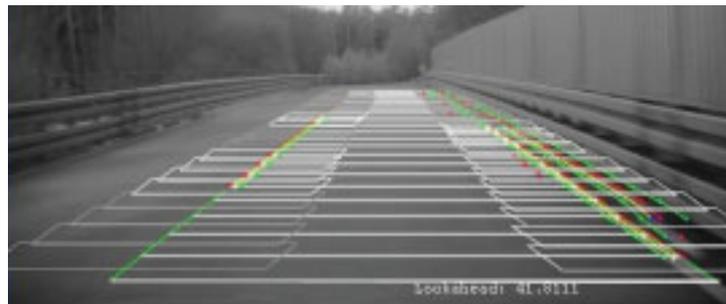
Ergebnisse

Die Algorithmen zur Merkmalsgruppierung mit anschließender Bewertung zu Merkmalsketten erzeugt ein robusteres Verhalten des Spurtrackers in Situationen mit vielen Störmerkmalen. Durch die Auswahl der besten Merkmale, nach den Kriterien des vorangehenden Unterkapitels, werden alle falschen Messungen verworfen auch wenn die einzelnen falschen Messungen sehr gut zu einem erwarteten Markierungsmerkmal passen. Somit kann die eingangs erwähnte Situation durch die beschriebenen Maßnahmen so gelöst werden, dass der Spurverlauf durchgehend verfolgt werden kann. Dadurch wird erstmals eine durchgehende automatische Regelung auf der Teststrecke ermöglicht. Abbildung 7.7 zeigt das Ergebnis der Verkettung für die verschiedenen Merkmalstypen. Die letztendlich ausgewählten Merkmale sind in Bild 7.7(b) mit einem weißen Kreuz markiert.

7 Strategien zur robusteren Objektverfolgung



(a) Merkmalsketten

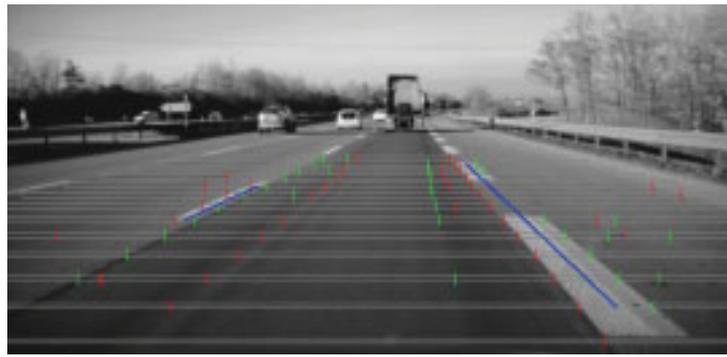


(b) Ausgewählte Merkmale

Bild 7.7: Merkmalsauswahl durch Verkettung

Der Ansatz zur Merkmalsverkettung kann auch in der Initialisierungsphase des Trackers genutzt werden. Hier besteht die Herausforderung darin, ohne Vorwissen ein Modell für die Fahrspur zu generieren und den Fahrspurtracker damit zu initialisieren. Unter der Annahme, dass sich das eigene Fahrzeug zum Zeitpunkt der Aktivierung des Systems parallel zur relevanten Fahrspurmarkierung bewegt, hat sich der Ansatz bewährt die Skelettlinie der Fahrspur mittig vor das Fahrzeug zu legen, bei einer Vorausschauweite von maximal 30 Meter. Die Messfenster zur Suche nach Markierungsmerkmalen mittels Bildverarbeitung werden in der Initialisierungsphase direkt auf die Skelettlinie gelegt. Die Suchbereiche der einzelnen Messfenster werden viel größer gewählt (z. B. doppelte erwartete Spurbreite) als in der Trackingphase. Da die Spaltenrichtung der Gradientenmaske ohne Vorwissen nicht an die Ausrichtung der Markierung im Bild angepasst werden kann, ist es wichtig, dass die Maskenhöhe kleiner gewählt wird als im angepassten Fall. Dadurch werden eventuelle Mittelungseffekte bei eher horizontal abgebildeten Markierungen (also Markierungen mit extremer seitlicher Ablage oder quer durch das Bild laufende Markierungen) minimiert.

Die Verwendung des gleichen Verkettungsansatzes wie während des Trackings liefert zu allen Markierungen im Bild die äquivalenten Ketten mit den dazugehörigen Beschreibungen nach Tabelle 7.1. Diese Daten können dazu verwendet werden, um z. B. den Anfang und Endpunkt der Ketten zu verbinden und daraus ein lineares Fahrspurmodell zur Initialisierung des Fahrspurtrackers abzuleiten. Bild 7.8(a) zeigt den Anwendungsfall für die Auffahrt auf eine Autobahn, wo zunächst der Verlauf der Markierungen und die relative



(a) Autobahnauffahrt



(b) Spurwegfall

Bild 7.8: Fahrspurdetektion durch Verkettung

eigene Lage dazu unbekannt sind. Die gefundenen Markierungshypothesen sind mit der Farbe blau markiert. In Bild 7.8(b) ist eine komplexere Situation dargestellt. Hier befindet sich das eigene Fahrzeug vor dem Ende einer Fahrspur und muss die Spur wechseln. Auch für diesen Fall liefert das Verkettungsverfahren die relevanten Markierungen der Nachbarspur und filtert korrekt die meisten möglichen Störmerkmale des Sperrbereichs.

7.1.3 Situationsabhängige Trackerdynamik

Verdeckungen der Fahrspurmarkierungen durch andere Verkehrsteilnehmer spielen vor allem bei niedrigen Geschwindigkeiten eine große Rolle. Gerade in Stausituationen auf Autobahnen, wenn man sich auf der mittleren Spur befindet, ist der Verdeckungseffekt sehr ausgeprägt zu beobachten. Das liegt zum einen an den extrem kleinen Abständen, die Fahrzeuge im Stau bei niedrigen Geschwindigkeiten zueinander haben. Zum anderen liegt dies aber auch an den großen Lücken zwischen den Teilstücken einer gestrichelten Markierungslinie auf der Autobahn. Diese können bis zu 20 Meter betragen. Befindet man sich auf einer mittleren Spur einer mindestens dreispurigen Fahrbahn können die Markierungen der eigenen Fahrspur sogar zeitweise vollständig durch ein Vorderfahrzeug verdeckt, also nicht sichtbar, sein.



Bild 7.9: Verdeckung der Fahrspurmarkierung

In diesem Fall ist die Wahrscheinlichkeit einer Fehlmessung bei der Suche nach Markierungsmerkmalen im Bild sehr hoch, da auch in Fahrzeugheckansichten markierungsähnliche Merkmale auftreten können.

Eine Möglichkeit die Auswirkungen einer solchen Fehlmessung zu minimieren ist die Anpassung der Modellunsicherheit an die Geschwindigkeit des EGO-Fahrzeugs. Zur Veranschaulichung der Sinnhaftigkeit dieser Vorgehensweise, kann folgendes Gedankenexperiment herangezogen werden: Man muss sich vorstellen, man fährt mit 100 km/h auf der Autobahn und schließt die Augen. Wie lange würde man sich sicher fühlen bevor man die Augen wieder öffnet, bzw. wie lange ist man sich noch sicher, dass die interne Vorstellung über den Fahrspurverlauf mit der Realität übereinstimmt? – Nun stellt man sich vor man macht das gleiche bei 50 km/h, 30 km/h oder im Stillstand.

Je nach Geschwindigkeit würde man wohl unterschiedlich lange auf eine „Messung“ verzichten können. Das heißt im Umkehrschluss, dass die Zunahme an Unsicherheit pro Zeiteinheit, zwischen zwei Messungen, von der eigenen Geschwindigkeit abhängt. Man kann das auch so interpretieren, dass die Unsicherheit über den Verlauf der Fahrspurmarkierung nicht von der vergangenen Zeit zwischen zwei Messungen, sondern vielmehr von dem gefahrenen Weg zwischen zwei Messungen abhängt. Die Übernahme dieses Gedankenganges in das bestehende System wird nun erläutert. Dazu müssen die Kalman-Gleichungen analysiert werden.

Die Systemfehlerkovarianzmatrix \mathbf{Q}_k besitzt die Werte der Modellfehlervarianzen für die einzelnen Zustandsgrößen. Sie drückt die Unzulänglichkeiten des verwendeten Dynamikmodells gegenüber der Realität aus. Sie wird im bestehenden System mittels Gleichung 7.16 aus einer zeitlich normierten \mathbf{Q}_0 -Matrix berechnet.

$$\mathbf{Q}_k = \mathbf{Q}_0 * (t_k - t_{k-1}) \quad (7.16)$$

Dies ist nötig, da eine konstante Framerate nicht garantiert werden kann, so dass die Zeiträume zwischen zwei Messungen variieren können. Die Matrix \mathbf{Q}_0 spiegelt die Zunahme der Schätzunsicherheit pro Zeiteinheit wider. Unter Verwendung der Gleichung $\Delta s_k = v_{k-1} \cdot (t_k - t_{k-1})$ kann nun die Gleichung 7.16 in die Gleichung 7.17 übergeführt

werden. Die Matrix \mathbf{Q}'_0 muss dabei die auf den Weg normierten Werte der Zustandsfehlervarianzen enthalten.

$$\mathbf{Q}'_k = \mathbf{Q}'_0 \cdot v_{k-1} \cdot (t_k - t_{k-1}) = \mathbf{Q}'_0 \cdot \Delta s_k \quad (7.17)$$

Bei dieser Vorgehensweise würden jedoch die Effekte, welche durch unterschiedliche Straßenführungen und unterschiedliches Fahrerverhalten mit in das System eingehen, außer Acht gelassen. Deshalb wird diese Variante nochmals um eine heuristische Komponente erweitert, die zusätzliche Freiheitsgrade ermöglicht. Dazu werden die Hilfsmatrizen $\mathbf{Q}'_{0,\min}$ und $\mathbf{Q}'_{0,\max}$ eingeführt. Diese bestehen aus minimal und maximal erlaubten Werten der auf den Weg normierten Modellfehlervarianzen. Gleichzeitig wird die Matrix \mathbf{Q}'_0 aus Gleichung 7.17 durch eine geschwindigkeitsabhängige Version $\mathbf{Q}'_0(v_{k-1})$ ersetzt. Diese berechnet sich, nach Gleichung 7.18, durch eine geschwindigkeitsabhängige Interpolation zwischen den Werten der Matrizen $\mathbf{Q}'_{0,\min}$ und $\mathbf{Q}'_{0,\max}$.

$$\mathbf{Q}'_0(v_{k-1}) = \mathbf{Q}'_{0,\min} + i(v_{k-1}) \cdot (\mathbf{Q}'_{0,\max} - \mathbf{Q}'_{0,\min}) \quad (7.18)$$

Daraus ergibt sich die endgültige Gleichung 7.19 zur Berechnung der Systemfehlervarianzen zwischen zwei Messzeitpunkten.

$$\mathbf{Q}'_k = \mathbf{Q}'_0(v_{k-1}) * \Delta s_k \quad (7.19)$$

Der Verlauf des Interpolationskoeffizienten $i(v_{k-1})$ ist heuristisch ermittelt und in Abbildung 7.10 dargestellt.

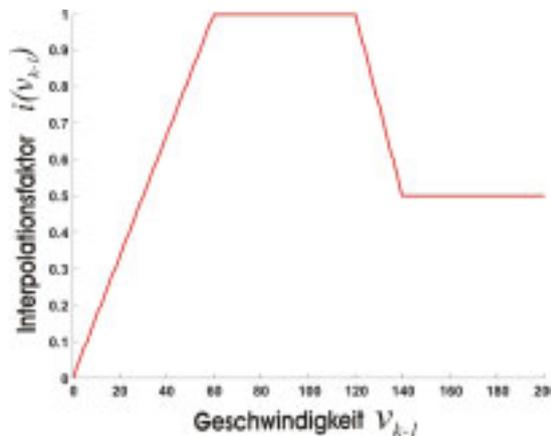


Bild 7.10: Verlauf des geschwindigkeitsabhängigen Interpolationsfaktors

Daraus ergibt sich für \mathbf{Q}'_k ein Verlauf, der bei einer Geschwindigkeit von 0 km/h ebenfalls nur Nullen aufweist. Bei steigenden Geschwindigkeiten erhöhen sich ebenfalls die Unsicherheitswerte. Der Abfall des Interpolationskoeffizienten bei sehr hohen Geschwindigkeiten ist davon abgeleitet, dass die Straßenführung auf Autobahnen der hohen Geschwindigkeit

7 Strategien zur robusteren Objektverfolgung

angepasst ist und sich daher wieder kleinere Werte für die Unsicherheit pro gefahrener Wegstrecke ergeben.

Die Auswirkungen dieser Vorgehensweise kann über die Kalman-Gleichungen klar gemacht werden. Die Einträge der Schätzfehlerkovarianzmatrix P_k werden, wie auch die Zustandsschätzwerte x_k selbst, rekursiv bestimmt. Dabei wird im Prädiktionsschritt des Kalmanfilters die Fehlerfortpflanzung vom Zeitpunkt t_{k-1} auf den Zeitpunkt t_k nach Gleichung 7.20 beschrieben.

$$\mathbf{P}_k^* = \Phi_{k-1} \cdot \hat{\mathbf{P}}_{k-1} \cdot \Phi_{k-1}^T + \mathbf{Q}_k \quad (7.20)$$

Wie man sieht, wird die Zunahme der Werte in der Schätzfehlerkovarianzmatrix \mathbf{P}_k^* durch die Addition mit der Systemfehlerkovarianzmatrix \mathbf{Q}_k mitbestimmt. Dies hat unmittelbar Auswirkung auf den Kalman-Verstärkungsfaktor \mathbf{K}_k in Gleichung 7.21.

$$\mathbf{K} = \mathbf{P}_k^* \cdot \mathbf{H}_k^T \cdot (\mathbf{H}_k \cdot \mathbf{P}_k^* \cdot \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (7.21)$$

\mathbf{H}_k ist dabei die Jakobimatrix der Messgleichungen im prädizierten Arbeitspunkt. Der Kalman-Verstärkungsfaktor legt fest, wie stark sich die Abweichungen zwischen prädiziertem und gemessenem Merkmal auf die Änderung (Innovation) der Zustandswerte auswirken. Dies bringt Gleichung 7.22 zum Ausdruck.

$$\hat{\underline{x}}_k = \underline{x}_k^* + \mathbf{K}_k \cdot (\underline{y}_k - \underline{y}_k^*) \quad (7.22)$$

Dabei enthält \underline{y}_k^* die prädizierten und \underline{y}_k die gemessenen Werte der Merkmale.

Für große Werte in \mathbf{P}_k nähert sich \mathbf{K}_k der Matrix \mathbf{H}_k^{-1} an, das heißt die Messabweichung wird direkt in die äquivalente Zustandsänderung umgerechnet. Für kleine Werte hingegen gehen die Einträge in \mathbf{K}_k gegen Null, was bedeutet, dass eine Messabweichung keine Auswirkung auf die Zustandsgrößen hat.

Somit hat die dynamische Berechnung von \mathbf{Q}_k aus Gleichung 7.20 mit $\mathbf{Q}_k = \mathbf{Q}'_k$ einen maßgeblichen Einfluss auf die Auswirkung einer Fehlmessung auf die geschätzten Größen, was auch als Dynamik bzw. Flinkheit des Trackers bezeichnet wird.

7.1.4 Gierratenoffsetschätzung

Wie schon im Kapitel 4.4.2 angesprochen, wurde das ursprüngliche Dynamikmodell zur Schätzung des Fahrspurverlaufs um die Komponente $\dot{\psi}_{off}(t)$ erweitert, die es erlaubt den dynamischen Offset des Gierratensensors zu schätzen. Die Berücksichtigung des geschätzten Wertes für $\dot{\psi}_{off}(t)$ wirkt sich vorteilhaft bei der Prädiktion der relativen EGO-Position zum Fahrspurverlauf aus.

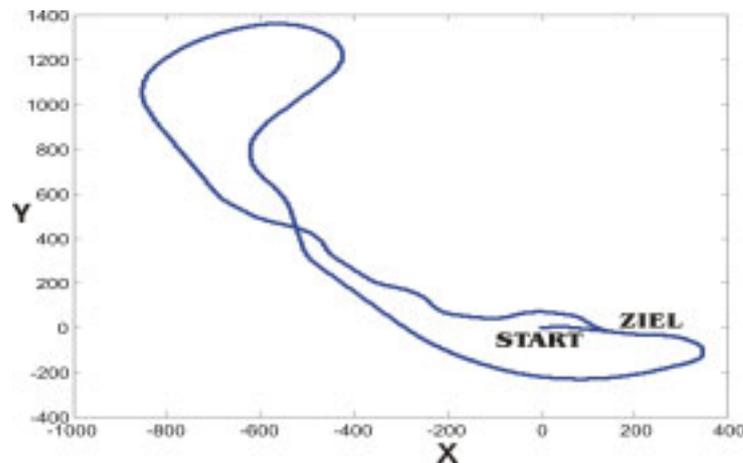


Bild 7.11: Streckenverlauf aus DGPS Referenz

In Abbildung 7.11 ist der rekonstruierte Verlauf der Versuchsstrecke zu sehen, der aus den Daten einer Fahrt mit Referenzsensorik gewonnen werden kann. Die Verläufe in den Grafiken 7.12(a) und 7.12(b) sind aus den geschätzten Größen des Spurtrackers generiert. Dabei werden im Fall 7.12(a) die fahrzeuginternen Größen Gierrate $\dot{\psi}(t)$ und die Geschwindigkeit $v(t)$ sowie der aktuell geschätzte Schwimmwinkel $\beta(t)$ dazu genutzt, eine relative Positionsänderung $(\Delta X, \Delta Y)$ des EGO-Fahrzeugs zwischen zwei Zeitschritten zu berechnen. Diese Änderungen werden aneinandergehängt, so dass der Plot 7.12(a) entsteht. Im Fall 7.12(b) wird zusätzlich der Gierratenoffset mitgeschätzt und bei der Berechnung der relativen Positionsänderungen berücksichtigt.

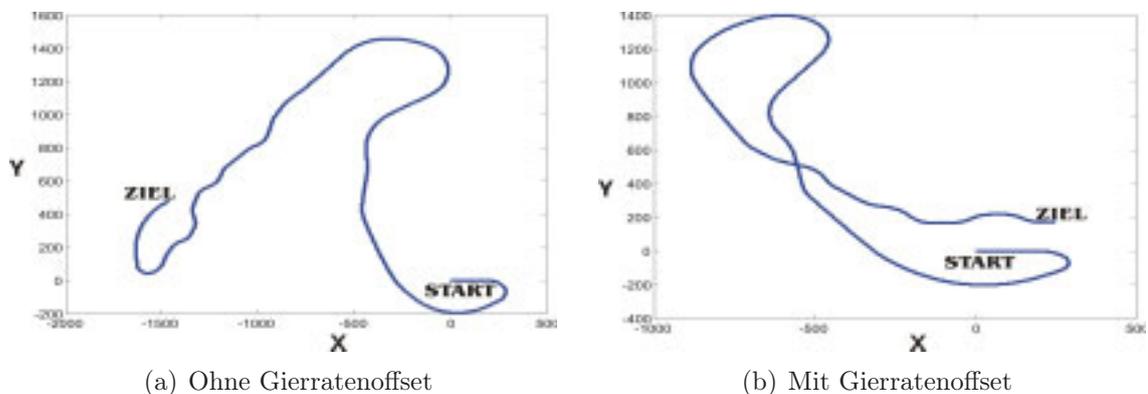


Bild 7.12: Vergleich der geschätzten Streckenverläufe

Durch die Aneinanderreihung der relativen Positionsänderungen pflanzen sich die kleinen Fehler aus den einzelnen Prädiktionsschritten fort. Im Fall 7.12(a) ist durch die Fehlerfortpflanzung der wahre Streckenverlauf nicht mehr erkennbar. Durch Hinzunahme des Schätzwertes für $\psi_{off}(t)$ verringert sich der Fehler so weit, dass der rekonstruierte Streckenverlauf dem Originalverlauf aus Abbildung 7.11 schon sehr ähnlich ist und der Startpunkt im zweiten Durchlauf beinahe wieder getroffen wird.

7.2 Multisensorielles Fahrzeugtracking

Der Schwerpunkt im Bereich der Fahrzeugerkennung liegt in dieser Arbeit auf den bereits vorgestellten Verfahren zur erstmaligen Detektion von Fahrzeugheckansichten. Dennoch wurden auch Erweiterungen an der bestehenden Variante des Fahrzeugtrackers durchgeführt. So ist das ursprünglich rein auf Videomerkmale basierende System um Merkmale aus Radar- und Lidarsicht erweitert worden. Die Information dieser distanzmessenden Sensoren wird ebenfalls nach dem 4D-Ansatz verwertet. Dabei kommen die Sensormodelle zum Einsatz, die bereits in den Kapiteln 3.2.2 und 3.2.3 beschrieben sind.



Bild 7.13: Sensorrohdaten ins Bild projiziert

Bild 7.13 zeigt die Rohdaten der verwendeten Radar- und Lidar-Sensoren. Die gemessenen Positionen werden dabei über das Kameramodell aus Kapitel 3.2.4 in das Bild projiziert. Die Punktobjekte des Radarsensors sind als grüne, auf dem Boden stehende, virtuelle Dreiecke mit einer Breite und Höhe von jeweils 1 Meter dargestellt. Die grüne Zahl unter dem Dreieck gibt die radarsensorinterne Objekt-ID wieder. Die Visualisierung der Lidarreflexe ergibt sich aus der Kopplung des Kameramodells mit dem Sensormodell für den Lidarsensor aus Kapitel 3.2.3. Die von einem Kanal ausgestrahlten Flächen werden aus der Einbaulage des Sensors und der Divergenz der Laserstrahlen in der gemessenen Entfernung berechnet. Diese Flächen sind durch rote Kästchen umschlossen. Die weiße Zahl über der Fläche gibt die gemessene Entfernung im Objektkoordinatensystem an, die Zahl darunter die Nummer des entsprechenden Laserkanals. Die gemessene Rückstrahlenergie ist quantisiert über den Grauwert im Inneren des Kästchens dargestellt. Um die Entfernungsinformation des Laserscanners besser interpretieren zu können, wird ein virtueller Bodenberührungspunkt berechnet und ebenfalls ins Videobild projiziert. Dieser ist über ein auf dem Kopf stehendes „T“, das mit dem Kasten verbunden ist, visualisiert. Wie man sieht, ist der Öffnungswinkel des nach vorne ausgerichteten Laserscanners viel größer als der Sichtbereich des Videosensors. Dies kann gut zur frühzeitigen Detektion von Fahrzeugen genutzt werden, so dass der Lidarsensor eine Alternative zu den in Kapitel 6 vorgestellten videobasierten Algorithmen darstellt.



Bild 7.14: Detektion über Lidarsignale

Ein Beispiel hierfür ist in Abbildung 7.14 gezeigt. Hier werden einzelne Messungen über den Verkettungsalgorithmus nach J. Gutmann[46] zu Linien verkettet. Die entstehenden Segmente werden über mehrere Zeitpunkte zueinander assoziiert. Nach drei Zeitschritten wird eine Objekthypothese mit den abgebildeten Informationen erzeugt. Dabei lässt sich eine initiale Schätzung für die Position, die Geschwindigkeit, die Breite und die Ausrichtung des Objektes ermitteln. Wie man am rechten Bildrand erkennen kann, lassen sich auch für den Fall der laserbasierten Objekterkennung Fehlhypothesen nicht vermeiden, so dass letztendlich die Kopplung aller Sensoren zu optimalen Ergebnissen führen muss.

Einmal detektierte Fahrzeuge können dann, unter Verwendung des 4D-Ansatzes, zusammen mit den Modellen für die Fahrzeugform aus Kapitel 4.4.1 und für die Fahrzeugdynamik aus Kapitel 4.4.2, multisensoriell getrackt werden. Abbildung 7.15 zeigt die Information, die dem Gesamtsystem zur Verfügung steht. Das 3D-Fahrzeugformmodell ist aus weißen Linien aufgebaut. Die weißen Kästen stellen die Operatormasken für die videobasierte Kanten- und Symmetriesuche dar. Das grüne Dreieck mit der ID „4“ ist die Radarinformation, die dem Modell zugeordnet ist. Die Lidarstrahlen, welche die der Kamera zugewiesenen Fahrzeugflächen schneiden, sind als grüne Kästen abgebildet. Die Distanzmessungen des Laserscanners werden im 4D-Ansatz als zusätzliche Merkmale verwendet. Für die Assoziation der Lidarmessungen zum Fahrzeugmodell muss vorher aus dem Sensormodell des Laserscanners eine Erwartung für die Lasermesspunkte errechnet werden, um daraus die nötigen prädierten Lidarmerkmale zu erzeugen.

Bild 7.16 zeigt noch einmal das Bild des getrackten Fahrzeugs. Hier werden im speziellen die Ergebnisse der Suche nach den erwarteten Bildmerkmalen visualisiert. Die weißen Verläufe stellen das Operatorergebnis der Kantensuche für die horizontalen und vertikalen Kanten dar. Der Verlauf des Symmetrieoperators ist grün dargestellt.

7 Strategien zur robusteren Objektverfolgung



Bild 7.15: Multisensorielles Tracken des Fahrzeugs



Bild 7.16: Operatorverläufe der Bildverarbeitung des Fahrzeugtrackers

8 Zusammenfassung und Ausblick

Der Erkennung von Objekten im Fahrzeugumfeld wird bei neuartigen Fahrerassistenzsystemen eine immer größere Bedeutung beigemessen. Durch genauere Informationen über die lokale Umgebung können Fahrerassistenzsysteme besser auf die aktuelle Situation reagieren, und es lassen sich neue Funktionsumfänge realisieren. Die vorliegende Arbeit gibt einen Überblick über aktuell verfügbare Fahrerassistenzsysteme und schildert den Einsatz von Sensoren zur Gewährleistung ihrer Funktionalität. Zusätzlich werden wichtige Forschungsgruppen im Bereich der autonomen Fahrzeugführung mit ihren Zielen vorgestellt.

Der Einsatz von Videosensorik mit Algorithmen zur Bildverarbeitung hat in den neu entwickelten Fahrerassistenzsystemen der letzten Jahre erheblich zugenommen. Dies lässt sich auf die reduzierten Herstellungskosten der Kamerachips sowie auf immer leistungsfähigere Steuergeräte in den Fahrzeugen zurückführen. So gibt es erst seit kurzem kamerabasierte Fahrerassistenzsysteme auf dem Markt, die den Fahrer dabei unterstützen, das eigene Fahrzeug in der Spur zu halten. Mittel- bis langfristig wird die Kamera im Fahrzeug die Erkennungsleistung des menschlichen Fahrers immer besser imitieren. Große Herausforderungen an die Bildverarbeitung ergeben sich durch die ständig wechselnden, extremen Umgebungsbedingungen sowie durch die große Vielfalt an relevanten Objekttypen.

Die videobasierte Erkennung von Fahrzeugheckansichten kann für viele Anwendungen im Bereich der Fahrerassistenz eine wichtige Informationsquelle bieten. Objekte, die bereits von einem anderen Sensor detektiert wurden, können durch die hohe Informationsdichte des Videobildes einer Objektklasse zugewiesen werden. Auch gibt es derzeit noch keinen entfernungs-messenden Sensor, der an die Winkelauflösung des Videosensors auch nur annähernd heranreicht. Somit stellen die in dieser Arbeit entwickelten Algorithmen zur Fahrzeugheckerkennung einen wichtigen Baustein für die Entwicklung eines Fahrerassistenzsystems dar, das auf andere Fahrzeuge im lokalen Umfeld reagieren muss.

Der verfolgte Ansatz einer zweistufigen Fahrzeugerkennung, bestehend aus ROI-Generierung und Klassifikation, weist vielversprechende Merkmale auf. Die regelbasierte ROI-Generierung ermöglicht es, die aufwändigere Erzeugung spezieller Fahrzeugmerkmale auf wenige kleine Bildbereiche zu reduzieren. Die dadurch sinkende Kombinationsvielfalt der Merkmale ermöglicht die letztendliche Klassifikation auf einem handelsüblichen PC in Echtzeit. Das Hinzufügen oder Vernachlässigen von Merkmalen ist in der Implementierung mit wenig Aufwand möglich, wodurch verschiedener Klassifikatorvarianten verglichen werden können.

Zur Klassifikation wird ein Maschinenlernverfahren verwendet, das auf Basis eines RF-Klassifikators ein GRBF-Netz erzeugt. Die Möglichkeit der Merkmalsselektion in der Trainingsphase des RF-Klassifikators bewirkt eine merkliche Effizienzsteigerung des erzeugten Gesamtsystems, da nur die wichtigsten Merkmale für die Klassifikationsaufgabe generiert und bewertet werden müssen. Das zur Laufzeit verwendete GRBF-Netz zeichnet sich durch eine leichte Implementierbarkeit, eine kurze Berechnungszeit sowie eine gute Interpretierbarkeit und Generalisierungsfähigkeit aus. Somit kann das verwendete Klassifikationsverfahren auch für andere Aufgaben zur bildbasierten Objekterkennung empfohlen werden.

Das zweistufige Detektionsverfahren ist für die beiden Anwendungsfälle der Detektion von Fahrzeugheckansichten in Tag- und Nachtsequenzen implementiert. Die dabei zugrunde gelegten und zum Teil eigens entwickelten Bildverarbeitungsverfahren zeichnen sich durch ihre hohe Anpassungsfähigkeit an die vorliegende Szene und durch die kurzen Berechnungszeiten aus. Sie ermöglichen die Extraktion von klassenspezifischen Merkmalen, die bei der ROI-Generierung und der danach folgenden Klassifikation verwendet werden.

Die Bewertung der entstehenden Gesamtsysteme zur Fahrzeugdetektion erweist sich als äußerst anspruchsvoll, da die vorhandenen Fahrzeuge in den Videosequenzen von Hand markiert werden müssen. Der hohe Aufwand der Referenzdatenerzeugung bewirkt, dass die Datenmenge zur Analyse des Detektionssystems sehr beschränkt und zum Teil nicht vollständig ist. Daher lassen sich lediglich Abschätzungen über die Erkennungsqualität des Systems machen. Die Detektionsraten der entworfenen rein videobasierten Systeme erfüllen noch nicht die Anforderungen, die Fahrerassistenzsysteme im Sicherheitsbereich an die Wahrnehmung stellen würden. Dennoch könnte durch Kopplung der videobasierten Klassifikation mit distanzmessenden Radar- oder Lidarsensoren ein sehr robustes Gesamtsystem für neuartige Komfortsysteme entstehen. Weiteres Verbesserungspotenzial der Algorithmen könnte auch durch eine bessere Kompensation der Bewegung des Fahrzeugaufbaus, durch weitere Bildmerkmale (z. B. durch den Einsatz einer Farbkamera) und durch einen größeren Trainingsdatensatz erreicht werden.

Neben der videobasierten Fahrzeugdetektion behandelt diese Arbeit auch die Themen der videobasierten Spurerkennung und der multisensoriellen Fahrzeugverfolgung. Dabei wird auf ein bestehendes Tracking-System mit Kalman-Filter-Kern zurückgegriffen. Die unterlegten Form- und Dynamikmodelle des Spur- und Fahrzeugtrackers werden detailliert beschrieben.

Die Erweiterungen an das bestehende System ermöglichen ein robusteres Verhalten des Fahrspurtrackers durch

- eine situationsadaptive Trackerdynamik,
- eine der Fahrsituation und Streckenführung angepasste Vorausschauweite und
- die Verwendung eines Modells zur Verkettung einzelner Spurmerkmale zu Merkmalsketten.

Die Erweiterungen im Bereich der Fahrzeugverfolgung sind durch das Einbringen neuer Merkmale der Lidar- und Radarsensoren gekennzeichnet. Die zur Generierung der Mess-

erwartung nötigen Modelle der verwendeten Sensoren sind ebenfalls detailliert beschrieben.

Die in dieser Arbeit entstandenen Bildverarbeitungsalgorithmen, die Sensormodelle, die Fahrzeugdetektoren sowie die erweiterten Systeme zum Verfolgen von Fahrzeugen und Fahrspurmarkierungen stehen der Entwicklungsabteilung für Fahrerassistenzsysteme bei Audi in einer lauffähigen Variante im Rahmensystem ADTF zur Verfügung. Die Algorithmen werden teilweise schon für Demonstrationszwecke eines prototypischen Fahrerassistenzsystems eingesetzt und durch Nachfolgearbeiten weiterentwickelt.

Durch das erweiterte Nutzen der Daten eines Kamerasensors und deren Vernetzung mit anderen Sensoren im Fahrzeug werden die Möglichkeiten zukünftiger Fahrerassistenzsysteme deutlich über denen heutiger liegen. So leistet diese Arbeit einen kleinen Beitrag dazu, dass die Vision des autonomen und unfallfreien Verkehrs in den kommenden Jahrzehnten in greifbare Nähe rücken wird.

A Grundlagen zu Maschinellenlernverfahren

Maschinellenlernverfahren werden dazu verwendet, basierend auf statistischem a-priori Wissen, Entscheidungen zu treffen. Das „Lernen“ findet in der sogenannten *Trainingsphase* statt. Dabei werden dem Algorithmus bestimmte Beispieldatenmuster zusammen mit der zugehörigen Klasse präsentiert. Die Datenmuster werden auch als Merkmalsvektoren $\underline{x} \in \mathbb{R}^N$ mit N Merkmalen bezeichnet. Findet man sehr aussagekräftige Merkmale lässt sich die klassentrennende Information durch wenige Merkmale ausdrücken. Eine reduzierte Dimension des Merkmalsraums verringert den Trainingsaufwand des Klassifikators und führt gleichzeitig zu einer besseren Klassifikationsleistung. Abbildung A.1 zeigt den prinzipiellen Datenfluss in einem Klassifikationssystem.

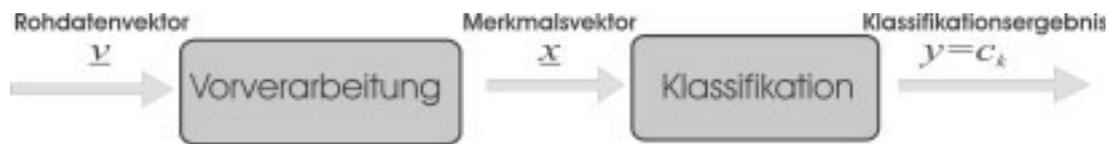


Bild A.1: Prinzipieller Datenfluss in einem Klassifikationssystem

Die auf einem Datenmuster \underline{x} basierende Entscheidung c_k des Systems wird in der Größe y abgelegt. Im Unterschied zu Regressionssystemen, bei denen y beliebige reelle Werte annehmen kann, lassen Klassifikationssysteme nur eine begrenzte Anzahl diskreter Werte zu, die den zu unterscheidenden Klassen entsprechen. In der Trainingsphase lassen sich verschiedene Arten des Lernens unterscheiden:

1. Unsupervised Learning, z. B. *Clustering*- oder *Self-Organization*-Algorithmen
2. Supervised Learning, z. B. *SVM* oder *MLPs*

Der in dieser Arbeit angewendete Lernalgorithmus greift auf das Prinzip des *Supervised Learning* zurück. In diesem Fall wird zu jedem Eingangsmuster \underline{x}_n auch ein Sollergebnis y_n präsentiert. Im Laufe der Trainingsphase versucht das Lernsystem nun eine Zuordnung der Eingangsgrößen auf die Ausgangsgrößen so zu realisieren, dass in möglichst vielen Fällen der Sollwert und der generierte Ausgabewert übereinstimmen. Das nachzubildende System wird bei Maschinellenlernverfahren meist als Black-Box betrachtet. Der Lernalgorithmus hat ein eigenes mehr oder weniger flexibles und komplexes Modell hinterlegt, dessen Parameter modifiziert werden, um das gewünschte Systemverhalten zu erlangen.

Um den Klassifikator bestmöglich an die Sollvorgaben anzupassen, wird von den meisten Verfahren eine Fehlerminimierung durchgeführt. Als Synonym für Fehler wird in diesem Zusammenhang auch oft der Begriff Risiko gebraucht. Um diese Optimierung durchführen zu können, ist zunächst eine Definition des verwendeten Risikofunktional $R(f)$ erforderlich. Die folgenden mathematischen Beschreibungen sind stark an die Arbeit von M. Botsch angelehnt[21][20].

Das Risikofunktional für die Klassifikation lautet

$$R(f) = E_{\underline{x}, y} \{ L(y, f(\underline{x})) \} = \int_{\underline{x} \in \mathbb{R}^N} \sum_{k=1}^C L(c_k, f(\underline{x})) p(\underline{x} = \underline{x}, y = c_k). \quad (\text{A.1})$$

$f(\underline{x})$ stellt die vom Lernsystem erzeugte Funktion dar. Die Gesamtheit der möglichen erzeugbaren Funktionen eines Lernsystems fasst man unter dem Begriff „Modellraum“ zusammen. Des Weiteren bezeichnet $p(\underline{x} = \underline{x}, y = y)$ die gemeinsame Wahrscheinlichkeitsdichtefunktion der Variablen \underline{x} und y . Die Funktion $L(\cdot)$ ist die Verlustfunktion („Loss-Function“), die es zu minimieren gilt. Sie gibt an, wann ein Sollwert als nicht erreicht gilt und wie sich dies auf das Risiko auswirkt.

Entsprechend werden der Funktion der Sollwert und das Ergebnis des Klassifikators übergeben. Eine häufig verwendete und anschauliche Realisierung der Verlustfunktion ist bei der Klassifikation

$$L(y, f(\underline{x})) = 1 - I(y, f(\underline{x})) = \begin{cases} 0 & \text{für } y = f(\underline{x}) \\ 1 & \text{sonst} \end{cases}. \quad (\text{A.2})$$

Wird die Funktion $f(\underline{x})$ so gewählt, dass das Risikofunktional A.1 minimiert wird, bezeichnet man $f(\underline{x})$ auch als Bayes-Klassifikator $f_B(\underline{x})$. Leider sind in der praktischen Anwendung die Wahrscheinlichkeitsdichtefunktionen in den meisten Fällen nicht bekannt. Folglich weicht man zwangsläufig auf eine andere Risikodefinition aus. Dieses neue Risiko wird als empirisches Risiko bezeichnet und beruht auf „gemessenen“ Datenmuster für die auch Sollwerte vorliegen:

$$R_{emp}(f, \mathcal{S}) = \frac{1}{G} \sum_{m=1}^G L(y_g, f(\underline{x}_g)). \quad (\text{A.3})$$

Die Formel A.3 berechnet für einen Trainingsdatensatz \mathcal{S} mit G Elementen den Mittelwert über alle Abweichungen der G Klassifikationsergebnisse von den Sollwerten. Die Bezeichnung $R_{emp}(f, \mathcal{S})$ sagt dabei aus, dass die Funktion f auf die Datenmenge \mathcal{S} angewendet wurde. Bei der Minimierung des empirischen Risikos ist zu beachten, dass ein sehr kleiner Wert von $R_{emp}(f, \mathcal{S})$ auf den Trainingsdaten nicht zwangsläufig bedeutet, dass das tatsächliche Risiko ebenfalls niedrig ist. Der niedrige Wert kann auch bedeuten,

dass die Trainingsdaten sehr gut auswendig gelernt wurden. Dabei ist dann keine Aussage enthalten, wie sich das System bei der Klassifikation von noch nie gesehenen, neu einzuordnenden Daten verhält. In der Regel nimmt ab einer bestimmten unteren Grenze des empirischen Risikos die Generalisierungsfähigkeit des Systems ab, d.h. die erfolgreiche Anwendbarkeit auf unbekannte Daten wird schlechter. Deshalb gibt es verschiedene Verfahren, um die Gefahr des Übertrainierens eines Klassifikationssystems zu minimieren. Mit der erfolgreichen Minimierung des Risikos ist die Trainingsphase abgeschlossen und das resultierende Klassifikationssystem kann zur Lösung von Klassifikationsaufgaben eingesetzt werden.

Der Zielkonflikt zwischen *Varianz* und *Bias* spielt bei der Auslegung eines Klassifikators eine zentrale Rolle und wird deshalb im nachfolgenden Abschnitt A.1 aufgegriffen. In Abschnitt A.2 wird das gängige Verfahren der Kreuzvalidierung erklärt.

A.1 Kompromiss zwischen Bias und Varianz

Um die Güte eines Maschinenlernverfahrens quantifizieren zu können, werden die beiden Kennwerte Bias und Varianz herangezogen. Dabei misst der Biaswert, wie gut der mittlere von einem Lernverfahren produzierte Klassifikator den Bayes-Klassifikator $f_B(\underline{x})$ annähert. Somit ist der Biaswert ein Maß dafür, ob der Modellraum im Allgemeinen flexibel genug ist, um die wahren Systemeigenschaften zu beschreiben. Im Gegensatz dazu drückt die Varianz aus, wie sehr sich die auf verschiedenen Datensätzen durch den gleichen Lernalgorithmus erzeugten Funktionen voneinander unterscheiden. Je kleiner der Biaswert ist, desto größer ist die Anpassungsfähigkeit des Modells und desto größer auch die Gefahr Datensätze auswendig zu lernen („Overfitting“). Je kleiner der Varianzwert ist, desto unabhängiger ist der entstehende Klassifikator von dem speziellen zur Erzeugung verwendeten Datensatz. Für einen möglichst gut angepassten Klassifikator, der nicht nur auf den bereits gesehenen Daten gut funktioniert, ist es notwendig beide Werte möglichst klein zu halten.

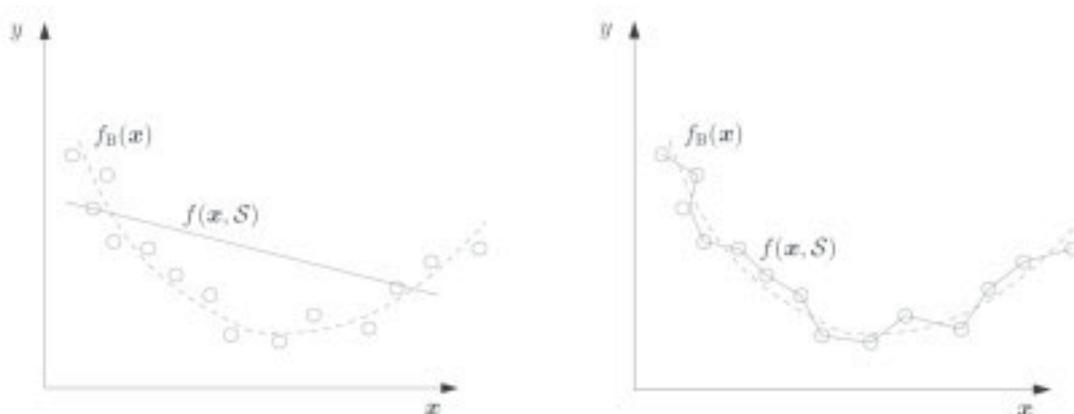


Bild A.2: Darstellung des Bias-Varianz-Problematik nach Bishop[12]

Eine anschauliche grafische Darstellung der Bedeutung von Bias und Varianz für den Fall eines Regressionssystems ist in Abbildung A.2 zu sehen. Im linken Bild wird deutlich, dass der parabolische Verlauf der zugrunde liegenden verrauschten Funktion durch eine lineare Funktion nur sehr ungenügend angenähert werden kann. Daraus resultiert ein hoher Biaswert. Hier sind komplexere Modelle erforderlich um gute Ergebnisse zu erzielen. Bei hoher Anpassungsfähigkeit der Nachbildungsfunktion kann jedoch ein sehr hoher Varianzwert resultieren. $f_B(\underline{x})$ bezeichnet in der Grafik den optimalen Bayes-Klassifikator.

Der Bayes-Klassifikator ist nach dem englischen Mathematiker Thomas Bayes benannt. Im Allgemeinen minimiert der Bayes-Klassifikator das Risikofunktional $R(f)$. Für den speziellen Fall einer *Loss-Function* $L(\cdot)$ nach Gleichung A.2, weist der Bayes-Klassifikator ein Eingangsmuster \underline{x} derjenigen Klasse c_k zu, die die höchste a-posteriori Wahrscheinlichkeit $p(y = c_k | \underline{x} = \underline{x})$ besitzt.

$$f_B(\underline{x}) = \max_{c_k} \{p(y = c_k | \underline{x} = \underline{x})\}. \quad (\text{A.4})$$

A.2 Bewertung und Validierung eines Klassifikators

Lernalgorithmen versuchen auf ganz unterschiedliche Weisen das Verhalten des realen Systems anzunähern. Dabei haben die vom Algorithmus erzeugten Beziehungen oft keinerlei Zusammenhang mit den tatsächlich ablaufenden physikalischen Prozessen des untersuchten Systems. Dies macht die von Maschinenlernalgorithmen erzeugten Klassifikatoren oder Regressionssysteme für den Anwender meist sehr schlecht nachvollziehbar bzw. durchschaubar. Da deshalb eine „Plausibilitätsprüfung“ durch den Entwickler kaum möglich ist, ist die Evaluation und Validierung der erzeugten Klassifikations- oder Regressionsergebnisse von großer Bedeutung. Deshalb sollen die Verfahren zur Überprüfung der Güte des erzeugten Klassifikators kurz erläutert werden. Ausführlicher werden die Themen in [24] und [21] beschrieben.

Eine zentrale Idee bei der Validierung eines Entscheidungssystems ist die Verwendung unabhängiger Datensätze für die Erzeugung des Klassifikators und dessen Überprüfung. Da die Erzeugung von Referenzdaten in vielen Fällen mit einem hohen Aufwand verbunden ist, soll gleichzeitig versucht werden, die vorhandenen Datensätze möglichst optimal zu nutzen. Teilt man den verfügbaren Datensatz zufällig in zwei Mengen und verwendet die eine Menge ausschließlich zum Training des Systems, die andere zur Bewertung, hat man die Unabhängigkeit der Daten sichergestellt. Allerdings verliert man dadurch einen erheblichen Teil der Daten, die für das Training des Systems nicht mehr verwendet werden können. Da tendenziell die Qualität eines Klassifikators mit steigender Anzahl an Trainingsdaten steigt, wählt man oft spezielle Vorgehensweisen, wie die V-fache Kreuzvalidierung.

V-fache Kreuzvalidierung

Bei der V-fachen Kreuzvalidierung (engl.: V-Fold Cross Validation) wird die Menge an verfügbaren Daten in V ungefähr gleich große Gruppen unterteilt. Das System wird jeweils mit $V - 1$ Gruppen trainiert und das Risikofunktional für den entstandenen Klassifikator $f^{(\mathcal{S} \setminus \mathcal{S}_v)}$ auf dem jeweils verbleibenden Datenset \mathcal{S}_v ausgewertet. Diesen Vorgang wiederholt man für alle verfügbaren Tupel. Anschließend bildet man den Mittelwert aus den Risiken der einzelnen Versuche. Der so erhaltene Schätzwert $R_{emp,CV}(f, \mathcal{S}, V)$ hat eine sehr hohe Aussagekraft hinsichtlich des realen Fehlers eines auf dem gesamten Datensatz trainierten Klassifikators.

$$R_{emp,CV}(f, \mathcal{S}, V) = \frac{1}{V} \sum_{v=1}^V R_{emp}(f^{(\mathcal{S} \setminus \mathcal{S}_v)}, \mathcal{S}_v) \quad (\text{A.5})$$

Zusätzlich lassen sich Rückschlüsse auf die Generalisierungsfähigkeiten des Systems ziehen. In der Praxis haben sich die 5- und 10-fache Kreuzvalidierung für die Abschätzung des realen Fehlers als besonders gut erwiesen.

B Parameter für Fahrzeugerkennung bei Nacht

Die hier abgebildeten Tabellen beinhalten die Parameter, die bei der Auswertung des Algorithmus zur Fahrzeugerkennung bei Nacht aus Kapitel 6.4 eingestellt waren. Einige Parameter beziehen sich auf den Lichtdetektor aus Kapitel 4.3.2.

$H_{\min} = 1.0m$	$H_{\max} = 2.3m$
$E_{\min} = 0.35$	$E_{\max} = 1.20m$
$B_{\min} = 1.10m$	$B_{\max} = 2.00m$

Tabelle B.1: Parameterset für Fahrzeugmodell aus Tabelle 6.5

$X_{\min}^{SB} = 10.00m$	$X_{\max}^{SB} = 30.00m$
$Z_{\min}^{SB} = 0.35m$	$Z_{\max}^{SB} = 1.20m$
$B^{SB} = 4.00m$	

Tabelle B.2: Parameterset für den Suchbereich aus Tabelle 6.6

$\tau_{\text{pos}} = 896$	$\tau_{\text{neg}} = -320$
$\tau_{va} = 5$	$\tau_{ve} = 5$
$\tau_{\Delta h, \min} = 0.65$	$\tau_{\Delta h, \max} = 2.50$

Tabelle B.3: Parameterset des Lichtdetektors aus Kapitel 4.3.2

$\varepsilon = 10^\circ$

Tabelle B.4: Parameter für Toleranzwinkel nach Abbildung 6.24

$\beta_{\min}^{(h)} = 0.70$	$\beta_{\max}^{(h)} = 1.30$
$\beta_{\min}^{(f)} = 0.60$	$\beta_{\max}^{(f)} = 1.40$

Tabelle B.5: Parameterset für gültige Lichtpaare nach Gleichung 6.5

C Bewertungsmaße

Dieser Teil erklärt die zur Auswertung der entwickelten Detektoren verwendeten statistischen Maße. Bei der Bewertung des videobasierten Detektionsverfahrens für Tagsequenzen aus Kapitel 6.3 kommen die Detektionsraten aus C.1 zum Einsatz. Der Algorithmus zur Erkennung von Heckansichten bei Nacht, aus Kapitel 6.4, wird auf Basis der Konfusionsmatrix bewertet, welche in C.2 beschrieben wird.

C.1 Detektionsraten

Zur Bewertung des Detektionssystems zur Erkennung von Fahrzeugheckansichten bei Tag werden verstärkt Detektionsraten analysiert. Dabei wird zwischen „Detektionen pro Bild“ (D/B), „Detektionen pro Fahrzeug“ (D/F) und „Fehldetektionen pro Bild“ (F/B) unterschieden. Hier soll kurz die Notwendigkeit dieser Unterscheidung verdeutlicht werden.

D/B bezeichnet das Verhältnis der Anzahl positiver Detektionen zur maximal möglichen Anzahl an positiven Detektionen in der Gesamtszene.

$$D/B = \sum_{n=1}^{N_{\text{Fzg}}} \frac{B_n}{B_{\text{Ges}}} \cdot \frac{D_n}{B_n} = \frac{1}{B_{\text{Ges}}} \sum_{n=1}^{N_{\text{Fzg}}} D_n \quad (\text{C.1})$$

$$\text{mit } B_{\text{Ges}} = \sum_{n=1}^{n=N_{\text{Fzg}}} B_n \quad (\text{C.2})$$

In einer Szene, in der insgesamt N_{Fzg} verschiedene Fahrzeuge zu sehen sind, ist B_n die Anzahl der Einzelbilder, in denen das n -te Fahrzeug im Gesamtvideo zu sehen ist. D_n gibt die Anzahl der Einzelbilder wieder in denen das n -te Objekt richtig erkannt wurde. Hierbei geht die Dauer des Vorhandenseins eines Objektes mit in die Bewertung ein. Objekte, welche sich nur wenige Einzelbilder im Sichtbereich der Kamera befinden, werden weniger stark gewichtet als Objekte, welche eine längere Dauer im Sichtbereich vorhanden sind.

D/O bezeichnet den Mittelwert der jeweils für ein Fahrzeug einzeln ermittelten D/B -Werte aller in einer Szene vorkommenden Fahrzeuge unabhängig voneinander.

$$D/O = \frac{1}{N_{\text{Fzg}}} \sum_{n=1}^{N_{\text{Fzg}}} \frac{D_n}{B_n} \quad (\text{C.3})$$

Hier werden die Detektionsraten für die einzelnen Objekte unabhängig voneinander berechnet und dann gemittelt. Dadurch geht jedes Objekt gleichgewichtet in die Bewertung ein und länger vorhandene werden nicht bevorzugt. Der Unterschied zum D/B -Wert lässt sich anhand des folgenden Beispiels veranschaulichen.

Beispiel: Ein Fahrzeug ist in 30 Einzelbildern sichtbar und wird in 26 davon vom System erkannt. Daraus ergeben sich die Werte:

$$D/B = D/O = 26/30 = 0.8667.$$

Ist in der gleichen Szene ein weiteres Fahrzeug über 10 Frames vorhanden, welches 7 mal vom System richtig erkannt wird ergeben sich hingegen folgende Werte:

$$D/B = \frac{1}{40} \cdot (26 + 7) = 0.825$$

und

$$D/O = \frac{1}{2} \cdot \left(\frac{26}{30} + \frac{7}{10} \right) \approx 0.78$$

F/B bezeichnet die Anzahl von Fehldetektionen bezogen auf die Gesamtanzahl an Bildern im Video.

$$F/B = \frac{1}{N_B} \sum_{n=1}^{N_B} F_n \quad (\text{C.4})$$

F_n ist die Anzahl der Fehldetektionen im n -ten Bild einer Szene mit N_B Einzelbildern.

C.2 Konfusionsmatrix

Im folgenden wird von einem Zwei-Klassen-Problem ausgegangen, wie es im Fall der Fahrzeugerkennung bei Nacht in Kapitel 6.4 existiert. Dabei repräsentiert die Klasse c_1 die gesuchte Objektklasse und c_2 allen anderen vorkommenden Klassen. Für diesen Fall können bei der Evaluierung die folgenden vier Fälle auftreten:

True-Positive (TP) Ein Muster \underline{x} der Klasse c_1 wird als Klasse c_1 erkannt.

False-Positive (FP) Ein Muster \underline{x} der Klasse c_2 wird als Klasse c_1 erkannt.

False-Negative (FN) Ein Muster \underline{x} der Klasse c_1 wird als Klasse c_2 erkannt.

True-Negative (TN) Ein Muster \underline{x} der Klasse c_2 wird als Klasse c_2 erkannt.

Dies lässt sich übersichtlich in Form der so genannten Konfusionsmatrix darstellen (vgl. Tabelle C.1), wobei die Einträge N_{TP} , N_{FP} , N_{FN} und N_{TN} für die Anzahl der in die jeweilige Kategorie fallenden Muster stehen.

C Bewertungsmaße

	Klasse c_1	Klasse \bar{c}_1
Als Klasse c_1 erkannt	N_{TP}	N_{FP}
Als Klasse \bar{c}_1 erkannt	N_{FN}	N_{TN}

Tabelle C.1: Konfusionsmatrix für Zwei-Klassen-Problem

Anhand dieser Matrix lassen sich nun mehrere Evaluierungsmaße bzw. Gütemaße ableiten, welche in dieser Arbeit verwendet werden:

$$Accuracy = \frac{N_{\text{TP}} + N_{\text{TN}}}{N_{\text{TP}} + N_{\text{FP}} + N_{\text{FN}} + N_{\text{D}}} \quad (\text{C.5})$$

$$Recall = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FN}}} \quad (\text{C.6})$$

$$Precision = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FP}}} \quad (\text{C.7})$$

Mit den Gleichungen C.5, C.6 und C.7 lassen sich folgendermaßen beschreiben:

- Die *Erkennungsrate* bzw. der *Accuracy*-Wert wird durch den Quotienten aus korrekt klassifizierten Mustern zu allen Mustern des Testdatensatzes definiert.
- Die *Sensitivität* bzw. der *Recall*-Wert beschreibt den Anteil der richtig klassifizierten Muster von allen Mustern, die in Wahrheit zur Klasse c_1 gehören.
- Die *Relevanz* bzw. der *Precision*-Wert beschreibt den Anteil der richtig klassifizierten Muster von allen Mustern, die der Klasse c_1 zugeordnet werden.

D Algorithmus zur Verkettung von Spurmarkierungsmerkmalen

Die in Pseudo-Code beschriebenen Algorithmen D.1 und D.2 werden zur Merkmalsverkettung im Kapitel 7.1.2 zur robusteren Fahrspurerkennung verwendet.

Algorithmus D.1: Initialisierung der Verkettung

```

Types = {DH, HD, DHD}
for each type in Types do
  set  $p_1 = 0, p_2 = 0$ 
  while  $p_1 < P - 2$  and  $X_{p_1} < X_{\max}$  do
     $p_2 = p_1 + 1$ 
    set  $\alpha_M = \text{atan2}(Y_{p_2} - Y_{p_1}, X_{p_2} - X_{p_1})$ 
    for  $n_1 = 1, \dots, N_{p_1}^{(\text{type})}$ 
      continue if  $M_{p_1, n_1}^{(\text{type})}$  has next or previous
      set  $Q_{\max} = 0$ 
      for  $n_2 = 1, \dots, N_{p_2}^{(\text{type})}$ 
11:         continue if  $M_{p_2, n_2}^{(\text{type})}$  has next or previous
12:         continue if type == DHD and  $|B_{p_1, n_1} - B_{p_2, n_2}| > \beta \cdot \max(B_{p_1, n_1}, B_{p_2, n_2})$ 
13:         set  $\alpha = \text{atan2}(Y_{p_2, n_2}^{(\text{type})} - Y_{p_1, n_1}^{(\text{type})}, X_{p_2, n_2}^{(\text{type})} - X_{p_1, n_1}^{(\text{type})})$ 
14:         set  $Q = \text{QUALDELTAANGLE}(\alpha, \alpha_M)$ 
           if  $Q > Q_{\max}$ 
16:             CONNECT( $M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})}$ )
17:             set  $Q\langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle = Q$ 
18:             set  $\alpha\langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle = \alpha$ 
           set  $Q_{\max} = Q$ 
           end if
       end for
     end for
   end for
23:   RECURSIVECONNECT( $p_2, \text{type}$ )
      $p_1 = p_1 + 1$ 
   end while
end for

```

Algorithmus D.1 beschreibt die Vorgehensweise, um einen Kettenstart zu erzeugen. Dazu müssen zwei aufeinander folgende Merkmale vom gleichen Typ existieren und bestimmte

D Algorithmus zur Verkettung von Spurmarkierungsmerkmalen

Kriterien erfüllen. Im Fall von DHD-Merkmalen wird in Zeile 11 des Algorithmus sichergestellt, dass sich die Breiten der beiden Merkmale nicht zu sehr unterscheiden. Das Winkelkriterium in Zeile 12 berechnet sich durch Gleichung 7.9. Dieses wird verwendet, um eine Güte für zwei aufeinander folgende Messungen zu berechnen. Für jedes gemessene Merkmal $M_{p_1, n_1}^{(\text{type})}$ um den Modellpunkt \underline{P}_{p_1} wird das nachfolgende Merkmale $M_{p_2, n_2}^{(\text{type})}$ mit der besten Qualität ermittelt. Das beste Paar wird mit der Funktion CONNECT in Zeile 14 verbunden und die Qualität $Q\langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle$ zwischen den beiden Merkmalen $M_{p_1, n_1}^{(\text{type})}$ und $M_{p_2, n_2}^{(\text{type})}$ sowie der Winkel $\alpha\langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle$ der Verbindungslinie der beiden Merkmale mit der Bildzeile gespeichert. Die Bezeichnung $\langle \cdot, \cdot \rangle$ drückt dabei die Beziehung in der doppelt verketteten Liste aus. Dabei ist der erste Eintrag der Vorgänger und der zweite Eintrag der Nachfolger innerhalb der Liste. Die Navigation durch die Liste wird im Folgenden durch die hiermit eingeführten Operationen $next(\cdot)$ und $previous(\cdot)$ beschrieben. Diese geben das Index-Tupel $[p^+, n^+]$ des Nachfolgers bzw. $[p^-, n^-]$ des Vorgängers in der Liste eines übergebenen Merkmals $M_{p, n}^{(\text{type})}$ zurück.

In Zeile 21 wird die weitere Verkettung der besten Kettenstarts aus Sicht des Merkmals $M_{p_1, n_1}^{(\text{type})}$ mit der Funktion RECURSIVECONNECT(p_2, type) durchgeführt. Diese wird nun unabhängig von der aktuellen Modellannahme vollzogen, damit sich die Modellfehler in größeren Entfernungen nicht auf die Messungen auswirken können. Die Vorgehensweise ist in Algorithmus D.2 beschrieben.

Zeile 4 gibt vor, dass Merkmalslücken nur für den Fall von DHD-Merkmalen auftreten dürfen. Diese dürfen laut Zeile 5 jedoch den Wert $\Delta X_{\text{Gap, max}}$ nicht überschreiten. Das liegt daran, dass Markierungen auch gestrichelt vorkommen können und dennoch eine zusammenhängende Kette bilden sollen. Sonstige spurbegrenzende Strukturen, wie Leitplanken oder Grasnaben, müssen in jedem Messfenster erkannt werden, um eine gültige Kette zu bilden. Für zwei Modellpunkte, deren Messungen die eben genannten Bedingungen erfüllen, werden nun wiederum die besten Merkmalspaare gesucht. DHD-Merkmale müssen dazu erneut in Zeile 9 eine maximale Breitenänderung unterschreiten. Das Gütemaß für die Verbindung zweier Messpunkte berechnet sich ebenfalls nach Gleichung 7.9. α_1 wird dabei wiederum über die Verbindungslinie der beiden zu verkettenden Messpunkte bestimmt. $\alpha_2 = \alpha^*$ berechnet sich nach den Regeln in den Zeilen 12–14. Dabei werden die Winkel zwischen etwaigen Vorgängern in der Kette für eine Prädiktion des erwarteten Winkels α^* nach Gleichung 7.10 genutzt. In Zeile 15 wird die Qualität der beiden gefundenen Messpunkte $M_{p_1, n_1}^{(\text{type})}$ und $M_{p_2, n_2}^{(\text{type})}$ berechnet. Falls das neue Merkmal $M_{p_2, n_2}^{(\text{type})}$ hinsichtlich dieses Qualitätsmaßes besser zu $M_{p_1, n_1}^{(\text{type})}$ passt als ein eventuell bereits vorhandener Nachfolger von $M_{p_1, n_1}^{(\text{type})}$ oder Vorgänger von $M_{p_2, n_2}^{(\text{type})}$, werden die beiden Messpunkte in Zeile 18 miteinander verbunden und der Winkel $\alpha\langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle$ sowie die Qualität $Q\langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle$ gespeichert. Wurde eine passende Konstellation zur Verlängerung der Kette gefunden, wird der Punkt mit dem Index p_2 der neue Ausgangspunkt für den Algorithmus ab Zeile 3 und p_2 wird inkrementiert. Ansonsten bleibt der Ausgangspunkt auf dem ursprünglichen Punkt mit dem Index p_1 und p_2 wird zur Überbrückung von Markierungslücken inkrementiert.

Algorithmus D.2: Modellunabhängige Verkettung (RECURSIVECONNECT)

input : Startindex Modellpunkt: p , Merkmalstyp: type
set $p_1 = p, p_2 = p + 1$
3: **while** $p_2 \leq P - 1$
4: **break if** $N_{p_2}^{(\text{type})} == 0$ **and** $\text{type} \neq \text{DHD}$
5: **break if** $X_{p_2} - X_{p_1} > \Delta X_{\text{Gap,max}}$
 set $\text{hit} = 0$
 for $n_1 = 1, \dots, N_{p_1}^{(\text{type})}$
 continue if $\text{previous}(M_{p_1, n_1}^{(\text{type})})$ does not exist
 for $n_2 = 1, \dots, N_{p_2}^{(\text{type})}$
10: **continue if** $\text{type} == \text{DHD}$ **and** $|B_{p_1, n_1} - B_{p_2, n_2}| > \beta \cdot \max(B_{p_1, n_1}, B_{p_2, n_2})$
 set $\alpha = \text{atan2}(Y_{p_2, n_2}^{(\text{type})} - Y_{p_1, n_1}^{(\text{type})}, X_{p_2, n_2}^{(\text{type})} - X_{p_1, n_1}^{(\text{type})})$
12: **set** $\alpha^* = \alpha \langle M_{p_1, n_1}^{(\text{type})}, M_{p_1, n_1}^{(\text{type})} \rangle$
13: **if** $\text{previous}(M_{p_1, n_1}^{(\text{type})})$ exists
14: **set** $\alpha^* = \alpha^* + 0.5 \cdot (\alpha \langle M_{p_1, n_1}^{(\text{type})}, M_{p_1, n_1}^{(\text{type})} \rangle - \alpha \langle M_{p_1, n_1}^{(\text{type})}, M_{p_1, n_1}^{(\text{type})} \rangle)$
 end if
16: **set** $Q = \text{QUALDELTAANGLE}(\alpha, \alpha^*)$
17: **continue if** $\text{previous}(M_{p_2, n_2}^{(\text{type})})$ exists **and** $Q < Q \langle M_{p_2, n_2}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle$
18: **continue if** $\text{next}(M_{p_1, n_1}^{(\text{type})})$ exists **and** $Q < Q \langle M_{p_1, n_1}^{(\text{type})}, M_{p_1, n_1}^{(\text{type})} \rangle$
19: CONNECT($M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})}$)
20: **set** $Q \langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle = Q$
21: **set** $\alpha \langle M_{p_1, n_1}^{(\text{type})}, M_{p_2, n_2}^{(\text{type})} \rangle = \alpha$
 set $\text{hit} = \text{hit} + 1$
 end for
 end for
 if $\text{hit} > 0$
26: **set** $p_1 = p_2$
 end if
 set $p_2 = p_2 + 1$
 end while

E Grundlagen zur Kalman-Filterung

Dieses Kapitel dient der Darlegung der theoretischen Grundlagen für das Unterkapitel 4.4.3 sowie Teilen des Kapitels 7. Hier werden die Grundlagen zur Kalman-Filterung kurz beschrieben. Weiterführende Informationen können in [6], [56], [66] und [95] gefunden werden. Die nachfolgende Beschreibung stützt sich im Wesentlichen auf die Arbeit von Welch und Bishop [95].

Das Kalman-Filter wird dazu verwendet, den Zustandsvektor $\underline{x} \in \mathbb{R}^n$ eines linearen zeitdiskreten Prozesses zu schätzen. Dieser kann durch die Gleichung E.1 beschrieben werden.

$$\underline{x}_k = \mathbf{\Phi}_k \cdot \underline{x}_{k-1} + \mathbf{G}_k \cdot \underline{u}_{k-1} + \underline{w}_{k-1} \quad (\text{E.1})$$

Die daraus entstehende Messung $\underline{z} \in \mathbb{R}^m$ wird durch Gleichung E.2 beschrieben.

$$\underline{z}_k = \mathbf{H}_k \cdot \underline{x}_k + \underline{v}_k \quad (\text{E.2})$$

Dabei repräsentieren die Zufallsvektoren \underline{w}_k und \underline{v}_k das Prozess- bzw. Messrauschen. Das Rauschen wird als weißes Gauss'sches Rauschen $N(0, \mathbf{Q}_k)$ bzw. $N(0, \mathbf{R}_k)$ modelliert, wobei \mathbf{Q}_k die Prozessfehlerkovarianzmatrix und \mathbf{R}_k die Messfehlerkovarianzmatrix darstellt.

Die Transitionsmatrix $\mathbf{\Phi}_k$ aus Gleichung E.1 prädiziert den Zustandsvektor vom letzten Zeitpunkt t_{k-1} auf den aktuellen Zeitpunkt t_k . Die Matrix \mathbf{G}_k führt den Steuervektor \underline{u}_k in den Zustandsraum über. Dadurch kann die Prädiktion durch Steuergrößen beeinflusst werden. Die Matrix \mathbf{H}_k stellt die Beziehung zwischen Messvektor (Merkmalsvektor) \underline{z}_k und Zustandsvektor \underline{x}_k her.

Das Kalman-Filter führt eine Schätzung des zugrundeliegenden Prozesses aus. Dazu wird eine rekursive Schleife durchlaufen, bei der in jedem Zeitpunkt der geschätzte Zustandsvektor zum letzten Zeitpunkt $\hat{\underline{x}}_{k-1}$ mit Hilfe von verrauschten Messungen \underline{z}_k korrigiert wird. Die dem Kalmanfilter zugrundeliegende *Predictor-Corrector*-Struktur wird anhand der nachfolgenden, sogenannten Kalman-Gleichungen, demonstriert.

$$\underline{x}_k^* = \mathbf{\Phi}_k \cdot \hat{\underline{x}}_{k-1} + \mathbf{G}_k \cdot \underline{u}_{k-1} \quad (\text{E.3})$$

$$\mathbf{P}_k^* = \mathbf{\Phi}_k \cdot \hat{\mathbf{P}}_{k-1} \cdot \mathbf{\Phi}_k^T + \mathbf{Q}_{k-1} \quad (\text{E.4})$$

Gleichungen E.3 und E.4 beschreiben die Prädiktion des geschätzten Zustandsvektors und der Schätzfehlerkovarianzmatrix auf den aktuellen Zeitpunkt.

$$\mathbf{K} = \mathbf{P}_k^* \cdot \mathbf{H}_k^T \cdot (\mathbf{H}_k \cdot \mathbf{P}_k^* \cdot \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (\text{E.5})$$

$$\hat{\underline{x}}_k = \underline{x}_k^* + \mathbf{K}_k \cdot (\underline{z}_k - \mathbf{H}_k \cdot \underline{x}_k^*) \quad (\text{E.6})$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \cdot \mathbf{H}_k) \cdot \mathbf{P}_k^* \quad (\text{E.7})$$

Mittels Gleichung E.5 wird die Kalmanverstärkung \mathbf{K}_k berechnet. Diese bestimmt den Einfluss der Messung \underline{z}_k auf die Änderung des zuletzt prädizierten Zustandsvektors \underline{x}_k^* . Der Messvektor \underline{z}_k wird aus den Daten des Prozesses erzeugt. Das dabei auftretende Messrauschen wird durch die Kovarianzmatrix \mathbf{R}_k abgebildet. Die Gleichungen E.6 und E.7 erzeugen die neuen Schätzwerte $\hat{\underline{x}}_k$ für den Zustandsvektor bzw. $\hat{\mathbf{P}}_k$ für die Schätzfehlerkovarianzmatrix.

Der zweite Teil der Berechnungsformel der Kalmanverstärkung beinhaltet die erwartete Innovationskovarianz \mathbf{V}_k^* .

$$\mathbf{V}_k^* = \mathbf{H}_k \cdot \mathbf{P}_k^* \cdot \mathbf{H}_k^T + \mathbf{R}_k \quad (\text{E.8})$$

Aus dieser kann der Toleranzbereich für die Assoziation von Messwerten \underline{z}_k zur generierten Messerwartung $\underline{z}_k^* = \mathbf{H}_k \cdot \underline{x}_k^*$ abgeleitet werden.

Eine Erweiterung des standard Kalman-Filter-Ansatzes stellt das erweiterte Kalman-Filter (EKF) dar. Dieses muss verwendet werden, wenn der nachzubildende Prozess nicht durch lineare Gleichungssysteme beschrieben werden kann. Dies ist beispielsweise bei der nichtlinearen perspektivischen Projektion der Fall[49]. Hier muss die nichtlineare Messgleichung

$$\underline{z}_k = g_k(\underline{x}_k) + \underline{v}_k \quad (\text{E.9})$$

im aktuellen Arbeitspunkte \underline{x}_k^* um die Nominaltrajektorie linearisiert werden. Dabei entsteht die linearisierte Messgleichung

$$\Delta \underline{z}_k = \mathbf{H}_k \cdot \Delta \underline{x}_k + \underline{v}_k \quad (\text{E.10})$$

$$\text{mit } \mathbf{H}_k = \left. \frac{\partial g_k(\underline{x})}{\partial \underline{x}} \right|_{\underline{x}=\underline{x}_k^*}. \quad (\text{E.11})$$

In Kapitel 4.4.3 wird die numerische Erzeugung der Mess-Jakobimatrix \mathbf{H}_k beschrieben.

F Diskretisierung von Differentialgleichungssystemen

Um die zeitkontinuierlichen Dynamikmodelle aus Kapitel 4.4.2 im vorliegenden zeitdiskreten Schätzer verwenden zu können, müssen die Differentialgleichungen in äquivalente zeitdiskrete Ausdrücke übergeführt werden. Zur Diskretisierung wird nach [69] auf das Mittel der Laplace-Transformation zurückgegriffen. Dabei berechnet sich die diskretisierte Variante von \mathbf{A} zu $\Phi(T)$ in Gleichung F.1.

$$\Phi(T) = L^{-1} ((s \cdot \mathbf{I} - \mathbf{A})^{-1}) \quad (\text{F.1})$$

Die diskrete Form der Eingangsverstärkungsmatrix \mathbf{B} wird basierend auf Φ durch Gleichung F.2 erzeugt.

$$\mathbf{G}(T) = \int_0^T \Phi(\tau) \cdot \mathbf{B} d\tau \quad (\text{F.2})$$

Literaturverzeichnis

- [1] ALCANTARILLA, P., L. BERGASA, P. JIMENEZ, M. SOTELO, I. PARRA und D. FERNANDEZ: *Night Time Vehicle Detection for Driving Assistance LightBeam Controller*. In: IEEE (Herausgeber): *Intelligent Vehicles Symposium*, Seiten 291–296. 2008. 117
- [2] AUDI AG: *Audi Lexikon*. <http://www.audi.de/de/brand/de/tools/advice/glossary.html>. Zuletzt geprüft am 19.05.2009. 7, 9, 11, 12
- [3] AUDI ELECTONIC VENTURE GMBH: *ADTF Entwicklerhandbuch*, 2006/2007. 44
- [4] AUDI ELECTONIC VENTURE GMBH: *ADTF Referenzhandbuch*, 2006/2007. 44, 45
- [5] BALDESSARI, R., B. BÖDEKKER, M. DEEGENER, A. FESTAG, W. FRANZ, C. KELLUM, T. KOSCH, A. KOVACS, M. LENARDI, C. MENIG, T. PEICHL, M. RÖCKL, D. SEEBERGER, M. STRASSBERGER, H. STRATIL, H. VÖGEL, B. WEYL und W. ZHANG: *Car-2-Car Communication Consortium: Manifesto*, 2007. 2
- [6] BAR-SHALOM, YAAKOV und E. THOMAS FORTMANN: *Tracking and Data Association*, Band 179 der Reihe *Mathematics in science and engineering*. Academic Press, Boston [u.a.], 1988. 168
- [7] BAYERISCHE FORSCHUNGSSTIFTUNG: *Forbias: Bioanaloge Sensomotorische Assistenz*. <http://www.forbias.de/home.html>, 2004. Zuletzt geprüft am 09.01.2009. 4, 69
- [8] BEHRINGER, R.: *Visuelle Erkennung und Interpretation des Fahrspurverlaufes durch Rechnersehen für ein autonomes Straßenfahrzeug: Dissertation*. Dissertation, Universität der Bundeswehr München, München, 1996. 14, 79
- [9] BERGMILLER, P.: *Videobasierte Fahrzeugheckansichtserkennung im Fahrzeugumfeld mittels Random-Forest-Klassifikation und Radialen-Basis-Funktionen-Netzen: Semesterarbeit*. München, 28.01.2008.
- [10] BERGMILLER, P., M. BOTSCH, J. SPETH und U. HOFMANN: *Vehicle Rear Detection in Images with Generalized Radial-Basis-Function Classifiers*. In: IEEE (Herausgeber): *Intelligent Vehicles Symposium*. 2008.
- [11] BIERMAN, J. G.: *Factorization methods for discrete sequential estimation*, Band 128 der Reihe *Mathematics in science and engineering*. Acad. Press, New York u.a., 1977. 4

- [12] BISHOP, M. C.: *Neural networks for pattern recognition*. Clarendon Press and Oxford University Press, Oxford , New York, 1995. 158
- [13] BISHOP, M. C.: *Pattern Recognition and Machine Learning*. Springer Science & Business Media LLC, New York, NY, 1. Aufl. Auflage, 2008. 88
- [14] BMW AG: *BMW Techniklexikon: Sicherheit*. http://www.bmw.de/de/de/insights/technology/technology_guide/index_category.html?code=4&view=Sicherheit. Zuletzt geprüft am 19.05.2009. 7
- [15] BOCK, T.: *Vehicle in the Loop - Test- und Simulationsumgebung für Fahrerassistenzsysteme: Dissertation*. Dissertation, TU München, München, 2008. 7, 36
- [16] BÖRNER, A.: *Optische Sensoren für Fahrzeuge*. In: *VDI-Berichte*, Band Vol. 1944, Seiten 37–48. VDI-Verl., 2006. 23
- [17] BORTZ, JÜRGEN: *Statistik für Sozialwissenschaftler: mit 247 Tabellen*. Springer-Lehrbuch. Springer, Berlin u.a., 5., vollst. überarb. und aktualisierte Aufl. Auflage, 1999. 67
- [18] BOSCH AG: *Kraftfahrzeugtechnik Erstausrüstung*. <http://rb-k.bosch.de/de/kraftfahrzeugtechnikerstausruestung/uebersicht/index.html>. Zuletzt geprüft am 19.05.2009. 7, 11, 12, 25, 36
- [19] BOTSCH, M.: *Machine Learning Techniques for Time Series Classification*. Doktorarbeit, TU München, München, 2009. 87
- [20] BOTSCH, M. und J. A. NOSSEK: *Construction of Interpretable Radial Basis Function Classifiers Based on the Random Forest Kernel*. In: *World Conference on Computational Intelligence*, Seiten 220–227. IEEE, 2008. 7, 88, 91, 132, 157
- [21] BOTSCH, M. und J. A. NOSSEK: *Feature Selection and Ensemble Learning in Time-Series Classification Tasks*. München, Januar 2007. 88, 91, 92, 157, 159
- [22] BOUGUET, Y J.: *Camera Calibration Toolbox for Matlab*. http://www.vision.caltech.edu/bouguetj/calib_doc/, 02.06.2008. Zuletzt geprüft am 23.05.2009. 33
- [23] BRAESS, H.-H.: *Vieweg Handbuch Kraftfahrzeugtechnik: mit 64 Tabellen*. ATZ-MTZ-Fachbuch. Vieweg, Braunschweig u.a., 2., verb. Aufl. Auflage, 2001. 7, 10
- [24] BREIMAN, L.: *Classification and regression trees*. Chapman & Hall/CRC, Boca Raton, 1st CRC press repr. Auflage, 1998. 85, 86, 159
- [25] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG: *invent: Intelligenter Verkehr und Nutzergerechte Technik*. <http://www.invent-online.de/index.html>, 2005. Zuletzt geprüft am 26.04.2009. 2, 3, 16
- [26] CHEN, Y.-L., Y.-H. CHEN, C.-J. CHEN und B.-F. WU: *Nighttime Vehicle Detection for Driver Assistance and Autonomous Vehicles*. In: IEEE (Herausgeber): *International Conference on Pattern Recognition (ICPR'06)*, Band 18. 2006. 117

- [27] CONTINENTAL AG: *Fahrerassistenzsysteme*. http://www.conti-online.com/generator/www/de/de/continental/automotive/themes/passenger_cars/chassis_safety/adas/ov1_adas_de.html, 2009. Zuletzt geprüft am 19.05.2009. 7
- [28] CRACKNELL, P ARTHUR und LADSON HAYES: *Introduction to remote sensing*. CRC Press, Boca Raton, Fla., 2. ed. Auflage, 2007. 27
- [29] DAIMLER AG: *Real Life Safety: Ein Stern genügt*. Stuttgart, 2008. 10
- [30] DAIMLER AG: *Daimler*. www.daimler.com, 2009. Zuletzt geprüft am 19.05.2009. 7, 13
- [31] DARPA: *DARPA Grand Challenge*. <http://www.darpa.mil/grandchallenge04/>, 2004. Zuletzt geprüft am 27.04.2009. 18
- [32] DARPA: *DARPA Grand Challenge*. <http://www.darpa.mil/grandchallenge05/>, 2005. Zuletzt geprüft am 27.04.2009. 19
- [33] DARPA: *DARPA Urban Challenge*. <http://www.darpa.mil/grandchallenge/index.asp>, 2007. Zuletzt geprüft am 27.04.2009. 19
- [34] DEUTSCHE FORSCHUNGSGEMEINSCHAFT (DFG): *Cognitive Automobiles: Trans-regional Collaborative Research Center 28*. <http://www.kognimobil.org/index.php>, 2009. Zuletzt geprüft am 20.05.2009. 2, 17
- [35] DICKMANN, D.: *Rahmensystem für visuelle Wahrnehmung veränderlicher Szenen durch Computer*. Doktorarbeit, Universität der Bundeswehr München, München, 1997. 75, 79
- [36] DICKMANN, D. E.: *4-D Szenenanalyse mit integralen raum/zeitlichen Modellen*. In: E. PAULUS (Herausgeber): *Informatik-Fachberichte*, Band 149 der Reihe *Musterrerkennung*, Seiten 257–271. Springer-Verlag, 1987. 14, 75, 79, 83
- [37] DICKMANN, D. E.: *Dynamic Vision for Perception and Control of Motion*. Springer London, 2007. 75
- [38] DIETSCH, K.-H.: *Kraftfahrtechnisches Taschenbuch*. Vieweg, Wiesbaden, 26., überarb. und erg. Aufl.. Auflage, 2007. 7, 11, 76
- [39] EYSENCK, M. und M. KEANE: *Cognitive psychology: A student's handbook*. Psychology Press, Hove, New York, 4. ed., repr. Auflage, 2004. 69
- [40] FÄRBER, G. und T. BRANDT: *FORBIAS - Forschungsverbund: Jahresbericht 2006 an die Bayerische Forschungstiftung*, 2007. 78
- [41] FORSCHUNGSGESELLSCHAFT FÜR STRASSEN- UND VERKEHRSWESSEN: *Richtlinien für die Anlage von Straßen: RAS-L*, 1995. 72
- [42] FORSCHUNGSINITIATIVE AKTIV: *AKTIV: Gemeinsam die Zukunft erleben*. <http://www.aktiv-online.org/deutsch/Downloads/Broschueren/aktiv%20image-deutsch.pdf>, 2006. Zuletzt geprüft am 20.05.2009. 15

- [43] FORSCHUNGSINITIATIVE AKTIV: *Aktive Sicherheit – AS: Sichterheitstechnologien aktivieren.* <http://www.aktiv-online.org/deutsch/Downloads/Broschueren/Aktiv%20AS-d.pdf>, 2006. Zuletzt geprüft am 20.05.2009. 15
- [44] FORSCHUNGSINITIATIVE AKTIV: *aktiv - adaptive und kooperative Technologien für den intelligenten Verkehr.* <http://www.aktiv-online.org/>, 2009. Zuletzt geprüft am 27.04.2009. 2, 15
- [45] FRANKE, U., C. RABE, H. BADINO und S. GEHRIG: *6D-Vision: Fusion of Stereo and Motion for Robust Environment Perception.* In: *27th Annual Meeting of the German Association for Pattern Recognition.* Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), 2005. 16, 17
- [46] GUTMANN, J.-S.: *Robuste Navigation autonomer mobiler Systeme.* Doktorarbeit, Univ. Freiburg im Breisgau, Freiburg, 2000. 151
- [47] HELLA KGAA HUECK & CO.: *Fahrerassistenz-Systeme: Technische Informationen.* http://www.hella.com/hella-de-de/assets/media_global/Autoindustry_ti_fas_d.pdf, 2009. Zuletzt geprüft am 17.05.2009. 7, 10
- [48] HOFMANN, U.: *Zur visuellen Umfeldwahrnehmung autonomer Fahrzeuge.* Doktorarbeit, Universität der Bundeswehr, München, 2004. 4, 7, 15, 51, 60, 69, 70
- [49] HOLT, V.: *Integrale multisensorielle Fahrumgebungserfassung nach dem 4D-Ansatz.* Doktorarbeit, Universität der Bundeswehr, München, 2004. 4, 7, 27, 77, 81, 169
- [50] HUDER, B.: *Einführung in die Radartechnik.* Teubner, Stuttgart, Leipzig, 1999. 25
- [51] INVENT: *INVENT "erfahren" – mobil mit 8 Sinnen: Ergebnisbericht – Teilprojekt Fahrumgebungserfassung und Interpretation – FUE.* <http://www.invent-online.de/index.html>, 2005. Zuletzt geprüft am 26.04.2009. 16
- [52] INVENT: *INVENT "erfahren" - mobil mit 8 Sinnen: Ergebnisbericht - Teilprojekt Stauassistenz - STA.* <http://www.invent-online.de/index.html>, 2005. Zuletzt geprüft am 26.04.2009. 16
- [53] ISERMANN, R.: *Fahrdynamik-Regelung: Modellbildung, Fahrerassistenzsysteme, Mechatronik.* <http://dx.doi.org/10.1007/978-3-8348-9049-8>, 2006. Zuletzt geprüft am 19.05.2009. 76
- [54] JOHANSEN, E.: *Millimeter-Wave Radar.* In: ACCETTA, S. J. und L. D. SHUMAKER (Herausgeber): *The Infrared and electro-optical systems handbook*, Seiten 115–242. SPIE Optical Engineering Press, Michigan, Washington, 1993. 28
- [55] KÄMPCHEN, N.: *Feature-level fusion of laser scanner and video data for advanced driver assistance systems.* Doktorarbeit, Universität Ulm, Ulm, 2007. 27
- [56] KAY, M. S.: *Fundamentals of statistical signal processing: Estimation theory*, Band Vol. 1. PTR Prentice Hall, Englewood Cliffs, NJ, 16 Auflage, 1993. 168

- [57] KUHL, P.: *Anpassung der Lichtverteilung des Abblendlichtes an den vertikalen Straßenverlauf*. Doktorarbeit, Universität Paderborn, Paderborn, 2006. 78
- [58] KUHNERT, K.-D.: *Zur Echtzeit-Bildfolgenanalyse mit Vorwissen*. Doktorarbeit, Universität der Bundeswehr München, Neubiberg, 1988. 14
- [59] LANGE, C.: *Wirkung von Fahrerassistenz auf der Führungsebene in Abhängigkeit der Modalität und des Automatisierungsgrades*. Doktorarbeit, TU München, München, 2007. 2, 7
- [60] LINDBERG, T., T. SCHALLER und B. GRADENEGGER: *Stauassistentz - Unterstützung des Fahrers durch Übernahme der Quer- und Längsführung im Stau: Hoch automatisiertes Fahren in Stop&Go- und Stausituationen*. VDI-Berichte, (2015):29–42, 2007. 16
- [61] LIU, W., X. WEN, B. DUAN, H. YUAN und N. WANG: *Rear Vehicle Detection and Tracking for Lane Change Assist*. In: IEEE (Herausgeber): *Intelligent Vehicles Symposium*, Seiten 252–257. 2007. 98
- [62] LUH, S.: *Untersuchung des Einflusses des horizontalen Sichtbereichs eines ACC-Sensors auf die Systemperformance*. Doktorarbeit, Technischen Universität Darmstadt, Darmstadt, 2006. 26
- [63] MAURER, M.: *Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen*, Band 443 der Reihe *Fortschritt-Berichte / VDIReihe 12, Verfahrenstechnik, Fahrzeugtechnik*. VDI Verl., Düsseldorf, Als Ms. gedr. Auflage, 2000. 14
- [64] MAURER, M.: *Einführung in Fahrerassistenzsysteme*. Materialien zur Vorlesung, TU München, 2004. 1
- [65] MAURER, M. und C. STILLER (Herausgeber): *Fahrerassistenzsysteme mit maschineller Wahrnehmung*. Springer-Verlag Berlin Heidelberg, 2004. 7, 14
- [66] MAYBECK, S. P.: *Stochastic models, estimation, and control*. Mathematics in science and engineering. Academic Press, New York, 1979. 168
- [67] MEINECKE, M.-M.: *Zum optimierten Sendesignalentwurf für Automobilradare*. Doktorarbeit, TU Hamburg-Harburg, Hamburg, 2001. 25, 27
- [68] MINSKY, M.: *Neural Nets and the Brain Model Problem*. Doktorarbeit, Princeton University, 1954. 84
- [69] MYSLIWETZ, B.: *Parallelrechner-basierte Bildfolgen-Interpretation zur autonomen Fahrzeugführung*. Doktorarbeit, Universität der Bundeswehr München, Neubiberg, 1990. 14, 72, 75, 76, 170
- [70] NEUMAIER, S. und G. FÄRBER: *Videobasierte 4D-Umfelderfassung für erweiterte Assistenzfunktionen: Vision-Based 4D-Environmental Perception for Advanced Assistance Functions*. it – Information Technology, (1):33–39, 2007. 69

- [71] NEUMANN-COSEL, K. VON, M. NENTWIG, D. LEHMANN, J. SPETH und A. KNOLL: *Preadjustment of a Vision-Based Lane Tracker: Using Virtual Test Drive Within a Hardware in the Loop Simulator*. In: *Driving Simulator Conference*. 2009.
- [72] REGENSBURGER, U. und V. GRAEFE: *Visual Recognition of Obstacles on Roads*. In: ELSEVIER B.V. (Herausgeber): *Selections of the International Conference on Intelligent Robots and Systems*, Seiten 73–86. 1995. 57, 58
- [73] RUSKE, G.: *Datenanalyse und Informationsreduktion*. Skriptum zur Vorlesung, Lehrstuhl für Mensch-Maschine-Kommunikation, TU München, 2007. 51
- [74] SCHELS, J.: *Videobasierte Fahrzeugerkennung bei Nacht: Diplomarbeit*. München, 29. Dezember 2008.
- [75] SCHULZE, M., T. MÄKINEN, J. IRION, M. FLAMENT und T. KESSEL: *PReVENT - Final report*. <http://www.prevent-ip.org/>, 2008. Zuletzt geprüft am 27.04.2009. 2
- [76] SCHWARZINGER, M.: *Flexible Fahrzeugerkennung für das Rechnersehen in einem Automobil*, Band 375 der Reihe *Fortschritt-Berichte VDI : Reihe 10, Informatik, Kommunikation*. VDI-Verl., Düsseldorf, 1995. Stand der Technik; Faltung; Eckenfindung; Polygonzug Extraktion; Deformierbare elastische Netze;. 14
- [77] SHAW-TAYLOR, J. und N. CRISTIANINI: *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge, Reprint. Auflage, 2005. 84, 88
- [78] SONG, G., K. LEE und J. LEE: *Vehicle detection by edge-based candidate generation and appearance-based classification*. In: IEEE (Herausgeber): *Intelligent Vehicles Symposium*. 2008. 98
- [79] SPIES, M.: *Technische Daten Precrash-Sensor NBS-45/90-106B*. 2006. 29, 36
- [80] SPIES, M. und H. SPIES: *Automobile Lidar Sensorik: Stand, Trends und zukünftige Herausforderungen*. In: *Advances in Radio Science*, Band vol. 4, Seiten 99–104. U.R.S.I. Landesausschuss in der Bundesrepublik Deutschland e.V., 2006. 27
- [81] STEGER, C., M. ULRICH und C. WIEDEMANN: *Machine vision algorithms and applications*. WILEY-VCH, Weinheim, 2008. 50
- [82] STEINMÜLLER, J.: *Bildanalyse: Von der Bildverarbeitung zur räumlichen Interpretation von Bildern*. Springer-Verlag, Berlin, Heidelberg, 2008. 47
- [83] STILLER, C.: *Fahrerassistenzsysteme - Von realisierten Funktionen zum vernetzt wahrnehmenden, selbstorganisierenden Verkehr*. In: MAURER, M. und C. STILLER (Herausgeber): *Fahrerassistenzsysteme mit maschineller Wahrnehmung*, Seiten 1–20. Springer-Verlag Berlin Heidelberg, 2004. 7, 8, 9
- [84] STILLER, C. und J. GAYKO: *Kognitive Automobile*. *Automatisierungstechnik - at*, (11):551–553, 2008. 17

- [85] STILLER, C., S. KAMMEL, I. LULCHEVA und J. ZIEGLER: *Probabilistische Methoden in der Umfeldwahrnehmung Kognitiver Automobile*. Automatisierungstechnik - at, (11):562–574, 2008. 17, 18
- [86] STÜKER, D.: *Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straßenverkehr*. Doktorarbeit, Carl von Ossietzky-Universität, Oldenburg, 17. November 2003. 7, 27
- [87] SUN, Z., G. BEBIS und R. MILLER: *Monocular Precrash Vehicle Detection: Features and Classifiers*. In: IEEE (Herausgeber): *Transactions on Image Processing*, Band 7, Seiten 2019–2034. 2006. 98
- [88] SUN, Z., G. BEBIS und R. MILLER: *On-Road Vehicle Detecion: A Review*. In: *Transactions on Pattern Analysis and Machine Intelligence*, Band 5, Seiten 694–711. IEEE, 2006. 15, 18, 98
- [89] TOYOTA DEUTSCHLAND GMBH - DIVISION LEXUS: *Innovations and Technologies*. <http://www.lexus.com/innovation/>, 2009. Zuletzt geprüft am 19.05.2009. 7
- [90] TSAI, R.: *A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses*. In: *Journal of Robotics and Automation*, Band Vol. 3, Seiten 323–344. IEEE, 1987. 33
- [91] VOLKSWAGEN AG: *Innovation & Technik: Assistenzsysteme*. http://www.volkswagen.de/vwcms/master_public/virtualmaster/de3/unternehmen/innovation__technik/assistenzsysteme.html, 2009. Zuletzt geprüft am 19.05.2009. 7, 11
- [92] WALLENTOWITZ, H. und G. REIF: *Handbuch Kraftfahrzeugelektronik: Grundlagen, Komponenten, Systeme, Anwendungen; mit zahlreichen Tabellen*. Vieweg, Wiesbaden, 1. Aufl. Auflage, 2006. 30
- [93] WALLHOFF, F.: *Entwicklung und Evaluierung neuartiger Verfahren zur automatischen Gesichtsdetektion, Identifikation und Emotionserkennung*. Doktorarbeit, TU München, München, 2006. 51, 54, 88
- [94] WANG, C. und J. LIEN: *Automatic Vehicle Detection Using Local Features - A Statistical Approach*. In: IEEE (Herausgeber): *Transactions on Intellegent Transportation Systems*, Band 7. 2008. 98
- [95] WELCH, G. und G. BISHOP: *An Introduction to the Kalman Filter*. <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>, 2006. Zuletzt geprüft am 22.04.2009. 168
- [96] WIEDERHOLD, G., J. MCCARTHY und E. FEIGENBAUM: *Memorial Resolution Arthur L. Samuel*. <http://histsoc.stanford.edu/pdfmem/SamuelA.pdf>, 1990. Zuletzt geprüft am 23.05.2009. 84
- [97] WIKIPEDIA: *Lochkamera*. <http://de.wikipedia.org/wiki/Lochkamera>, 06.05.2009. Zuletzt geprüft am 23.05.2009. 32

Literaturverzeichnis

- [98] WIKIPEDIA: *Einparkhilfe*. <http://de.wikipedia.org/wiki/Einparkhilfe>, 2009. Zuletzt geprüft am 17.05.2009. 10
- [99] WIKIPEDIA: *Gehirn*. <http://de.wikipedia.org/wiki/Gehirn>, 2009. Zuletzt geprüft am 23.05.2009. 84
- [100] WÜNSCHE, H.-J.: *Bewegungssteuerung durch Rechnersehen: Ein Verfahren zur Erfassung und Steuerung räumlicher Bewegungsvorgänge in Echtzeit*, Band 20 der Reihe *Fachberichte Messen - Steuern - Regeln*. Springer, Berlin, 1988. Als Disseratation: "Erfassung und Steuerung von Bewegungen durch Rechnersehen", Universität der Bundeswehr München. 14
- [101] YOUNG, I., J. GERBRANDS und L. VAN VLIET: *Fundamentals of Image Processing*. Delft PH Publications, Delft, 1998. 50
- [102] ZWIJNENBERG, H. und ET AL.: *Benchmarking study on activities in promoting and deploying Intelligent vehicle safety systems in the EU: TNO Report 2007-D-R0674/B*. 18

