Algorithmenauswahl für den adaptiven Sensoreinsatz an Bord unbemannter Luftfahrzeuge

Christian Hellert

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

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Mag. Dr. habil. Thomas Pany

1. Gutachter: Univ.-Prof. Dr.-Ing. Peter Stütz

2. Gutachter: Univ.-Prof. Dr. rer. nat. Wolfram Hardt

Diese Dissertation wurde am 08.11.2018 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 14.11.2018 angenommen. Die mündliche Prüfung fand am 26.04.2019 statt.

Zusammenfassung

Die Forderung nach einem gesteigerten Automatisierungsgrad von unbemannten Luftfahrzeugen betrifft auch den Betrieb und Funktionsweise der an Bord befindlichen Sensorsysteme zur Umwelterfassung. Solche Systeme dienen zum einen der navigatorischen Unterstützung (z. B. *Sense & Avoid*) und zum anderen aber vornehmlich als Nutzlast zur Durchführung von Aufklärungs- und Überwachungsmissionen.

Hierbei beeinflussen variierende Umgebungsbedingungen, wie z.B. atmosphärische Gegebenheiten oder topographische Veränderungen, die Leistung dieser Systeme. Vor diesem Hintergrund stellt diese Arbeit ein automatisiertes Verfahren zum adaptiven luftgestützten Sensoreinsatz vor, mit dem auf Veränderungen der Umgebungsbedingungen reagiert und damit einer Degradation der Systemleistung ohne kontinuierlichen Eingriff durch einen Operateur entgegengewirkt wird. Die grundlegende Methodik des adaptiven Sensoreinsatzes basiert auf einer Algorithmenauswahl, welche Modelle zum Leistungsverhalten verschiedener, an Bord zur Verfügung stehender Algorithmen zur Sensordatenauswertung nutzt. Die eingesetzten Algorithmen verwenden unterschiedliche Methoden zur Fahrzeugerkennung auf Sensordaten Sensorik. Leistungsbestimmung bildgebender Um die durchzuführen, wurden Umgebungszustände zur Beschreibung des situativen Umfeldes verwendet. Die Modellierung des Zusammenhangs zwischen Umgebungszustand und der Algorithmenleistung basiert dabei zum einen auf Expertenwissen und zum anderen auf Methoden des maschinellen Lernens. Für letzteres kam eine Simulationsumgebung zum Einsatz, um entsprechende Daten zu erheben.

Das realisierte System zum adaptiven Sensoreinsatz wurde anhand von Experimenten in der Simulationsumgebung und durch Flugexperimente evaluiert. Dabei konnte festgestellt werden, dass eine Verbesserung der Aufklärungsleistung durch Algorithmenauswahl gegenüber dem alleinigen Einsatz des zur Verfügung stehenden leistungsstärksten Algorithmus zur Objekterkennung erreicht werden konnte. Außerdem wurde gezeigt, dass eine synthetische Sensorsimulation genutzt werden kann, um die Erzeugung der Modelle zur Leistungsbestimmung zu realisieren, und dass gleichzeitig eine Verwendbarkeit in realen Szenarien möglich ist. Hierbei sich stellte sich heraus, dass die Genauigkeiten bei der Leistungsbestimmung in simulierten und realen Situationen vergleichbar sind. Abschließend konnte der implementierte adaptive Sensoreinsatz erfolgreich in realen Flugexperimenten demonstriert werden.

Schlagworte: Sensormanagement; Sensoreinsatz; UAV; Bildverarbeitung; Fahrzeugdetektion; synthetische Datengenerierung; Algorithmenauswahl; Umgebungszustand; Luftaufklärung; Leistungsmodelle; bildgebende Sensoren; Expertenwissen; Fuzzylogik; künstliche neuronale Netze;

Abstract

The demand for an increased degree of automation of unmanned aerial vehicles also affects the operation and functionality of the on-board sensor systems for environmental sensing. Such systems serve on the one hand as navigational support (e.g. sense & avoid), but on the other hand primarily as payload for conducting reconnaissance and surveillance missions.

The performance of these systems is influenced by varying environmental conditions, e.g. atmospheric conditions or topographic changes. Against this background, this thesis proposes a method for adaptive airborne sensor deployment, enabling a response to changes in environmental conditions and thus counteracting degradation of system performance without a continuous intervention by an operator. The basic methodology of adaptive sensor deployment is based on an algorithm selection which uses models for the performance behavior of the various algorithms available on board for sensor data processing. The algorithms use different methods for vehicle recognition on sensor data of imagery sensors. In order to determine the performance, environmental states were used to describe the situational environment. The modelling of the relationship between the environmental state and the algorithm performance is based on expert knowledge on the one hand and on methods of machine learning on the other. For the latter, a simulation environment was used to collect corresponding data.

The implemented system for adaptive sensor deployment was evaluated by experiments in the simulation environment as well as by flight experiments. It was found that an improvement of the reconnaissance performance could be achieved by algorithm selection compared to the sole use of the most powerful algorithm available for object recognition. Furthermore, it was shown that a synthetic sensor simulation can be used to generate the models for performance determination and at the same time be usable in real scenarios. It turned out that the accuracies of the performance determination are comparable in simulated and real situations. Finally, the implemented adaptive sensor deployment could be successfully demonstrated in real flight experiments.

Keywords: sensor management; sensor deployment; UAV; computer vision; vehicle detection; synthetic data generation; algorithm selection; environmental state; aerial reconnaissance; performance model; imaging sensors; expert knowledge; fuzzy logic; artificial neural networks;

Danksagung

"NOTHING WORTH DOING IS EVER EASY" – Theodore Roosevelt

Jeder, der an einer Dissertation gearbeitet hat oder arbeitet, sagt wahrscheinlich: Stimmt. Dass ich meine Dissertation fertigstellen konnte, liegt, außer an mir, an vielen im Hintergrund agierenden Menschen, denen ich an dieser Stelle danken möchte.

Zunächst sind es meine Eltern, denen ich danken möchte. Sie haben es mir überhaupt ermöglicht, den Weg einer Promotion einzuschlagen. Insbesondere danke ich auch meiner Ehefrau Tonja für ihre außergewöhnliche Unterstützung. Gerade während der schwierigen Phasen war sie immer für mich da und hat mir den Rücken für meine Arbeit freigehalten. Ein großes Dankeschön dafür! Du und unsere Tochter Miriam, ihr seid das Beste was mir passieren konnte.

Weiterhin möchte ich meinem Doktorvater Prof. Stütz dafür danken, dass er mich sowohl bei meinen wissenschaftlichen Arbeiten am Institut als auch bei der Themenfindung der Dissertation unterstützt hat. Er hatte stets ein offenes Ohr für mich, und ich möchte ihm auch für seine Ratschläge und vor allem für die wertvollen Diskussionen gerade in der abschließenden Phase meiner Promotion danken. Des Weiteren danke ich Prof. Hardt für sein Interesse an meiner Arbeit und die Übernahme des Koreferats.

Besonders hervorheben möchte ich meinen Kollegen Denis Smirnov, der gleichzeitig mit mir an den Lehrstuhl kam. Sowohl die Zusammenarbeit im Projekt *Sagitta* als auch die unzähligen Diskussionen zu unseren Promotionsthemen waren stets produktiv, wertschätzend und unterhaltsam. Seine offene Art, seine Gelassenheit, aber auch seine Zielstrebigkeit haben mich persönlich bereichert. Wir sind während unserer gemeinsamen Zeit an der Universität zu guten Kollegen geworden und haben diese als sehr gute Freunde verlassen. Danke für diese unvergessliche Zeit.

Ein weiterer besonderer Dank geht an Marc Schmitt und Georg Hummel für ihre andauernde Unterstützung, Diskussionsbereitschaft und die entstandenen Freundschaften. Sie haben das Labor zu einem Ort gemacht, an dem ich sehr gerne gearbeitet – man muss schon fast sagen gelebt habe. Ich werde immer an unsere gemeinsame Zeit zurückdenken. Marc, meinem besten Freund und Trauzeuge, danke ich für alles, was er für mich getan hat und wünsche ihm viel Erfolg für seine eigene Promotion. An Georg schätze ich seine besondere, verrückte Art und wünsche ihm und seiner Familie alles Gute für die Zukunft. Ich kann mich glücklich schätzen euch als Freunde zu haben.

Weiterhin möchte ich der Werkstatt für die gute Zusammenarbeit danken. Ohne die Kollegen dort wären meine Flugexperimente nie möglich gewesen. Ein großes Dankeschön auch an Alexander Schelle, der mit seiner strukturierten Art die für diese Arbeit wertvollen Flugversuche außergewöhnlich unterstützt hat. Danke auch an das Sekretariat und an die aktuellen wie ehemaligen Kollegen am Institut für die guten fachlichen Diskussionen und die vielen Erlebnisse außerhalb des Institutslebens.

Zuletzt möchte ich noch Wolfgang Müller-Tamke für das umfangreiche Korrektorat meiner Dissertation danken.

Meine Zeit am Institut für Flugsysteme hat mich persönlich sehr bereichert und Erfahrungen sammeln lassen, die ich niemals missen möchte.

Für meine Familie

Inhaltsverzeichnis

Zusamme	nfassung	iii
Abstract		V
Danksagu	ng	vii
Symbolve	rzeichnis	XV
Abkürzun	gsverzeichnis	xvii
Kapitel 1	Einleitung	1
1.1 l	Eingliederung der Arbeit	1
1.2 I	Feststellung des Forschungsbedarfs	3
1.3 I	Definition des Forschungsziels	4
1.4 I	nhaltsübersicht	5
Kapitel 2	Stand der Technik	7
2.1	Missionssensorik und deren Eigenschaften	7
2.2	Maschinelle luftgestützte Fahrzeugerkennung mit bildgebender Sensorik	9
2.2.1	Template Matching und wissensbasierte Methoden	10
2.2.2	Methoden des maschinellen Lernens	10
2.2.3	Diskussion	12
2.3 I	Einflussfaktoren auf die Sensordatenverarbeitung	13
2.3.1	Geometrische Eigenschaften von Luftbildaufnahmen	13
2.3.2	Atmosphärische Einflüsse	15
2.3.3	Bodenbedeckung und Geländeform	17
2.3.4	Diskussion	
2.4	Ansätze für den adaptiven Sensoreinsatz	19
2.4.1	Sensormanagement	19
2.4.2	Ansätze zur Adaption der Sensordatenverarbeitung	20
2.4.3	Sensor- und Perzeptionsmanagement	21
2.4.4	Diskussion	24
2.5	Algorithmenauswahl für Bildverarbeitungsroutinen	25
2.5.1	Wissensbasierte Methoden	25
2.5.2	Lernbasierte Methoden	26
2.5.3	Diskussion	27
Kapitel 3	Konzept und Methodenauswahl	31
3.1	Anforderungen	
3.1.1	Systemische Anforderungen	

3.1	2 Funktionale Anforderungen	33
3.2	Konzept zum adaptiven luftgestützten Sensoreinsatz	33
3.2.	1 Leistungsmodellierung	33
3.2	2 Algorithmenauswahl durch Anwendung der Leistungsmodelle	35
3.3	Methodenauswahl und Festlegungen	
3.3	1 Vorgehen zur Sensordaten- und Umgebungszustandserhebung	
3.3.	2 Algorithmen zur luftgestützten Fahrzeugerkennung	
3.3.	3 Umgebungszustände und deren Auswirkungen auf die Sensordaten	40
3.3	4 Methoden zur Leistungsmodellerzeugung durch maschinelles Lernen	45
3.4	Konzeptdetaillierung der Leistungsmodellierung durch Simulationseinsatz	47
3.5	Evaluierungsstrategie	49
Kapitel 4	4 Verfahren zur Vorbereitung der Leistungsmodellierung	51
4.1	Simulation der Sensordaten	51
4.2	Bestimmung der Umgebungszustände	52
4.2	1 Atmosphärische Umgebungszustände	52
4.2	2 Photogrammetrische Umgebungszustände	57
4.2	3 Topographische Umgebungszustände	59
4.3	Zusammenfassung der Umgebungszustände	61
4.4	Generierung des Modelldatensatzes	62
4.5	Perzeptionsmodule zur Fahrzeugerkennung	65
4.5	1 Segmentierung und Hypothesengenerierung	65
4.5	2 Objektdetektoren	71
4.6	Definition des Leistungsindex	
4.7	Generierung der Metadaten	87
Kapitel !	5 Verfahren zur Leistungsmodellierung	
5.1	Leistungsmodellierung mittels Expertenwissens	89
5.1	1 Aufbau des Fuzzy-Inferenzsystems	
5.1	2 Fuzzy-Inferenzprozess	90
5.1	3 Erfassung des Expertenwissens	92
5.1	4 Modellierung der Zugehörigkeitsfunktionen und Regeln	98
5.2	Leistungsmodellierung mittels ML-Verfahren	102
5.2	1 Datenaufbereitung	102
5.2	2 Leistungsmodellierung mit dem NEFPROX-Algorithmus	103
5.2	3 Leistungsmodellierung mittels künstlicher neuronaler Netze	109
Kapitel	6 Verfahren zur Algorithmenauswahl	113
6.1	Auswahl der Perzeptionsketten	113

6.2	Erweiterung des Sensor- und Perzeptionsmanagementsystems		
6.3	Umsetzung des erweiterten SPMS	116	
Kapitel 7	2 Experimente	119	
7.1	Untersuchungsgegenstände	119	
7.2	Experimentbeschreibung zur Analyse der Experten-Leistungsmodelle	120	
7.3	Experimentbeschreibung zur Analyse der ML-Leistungsmodelle	120	
7.4	Missionsplanung und -beschreibung zur Evaluierung der Algorithmenauswahl	121	
7.4.	1 Missionen in der virtuellen Umgebung	122	
7.4.	2 Flugexperimente	130	
Kapitel 8	3 Analyse und Ergebnisse	135	
8.1	Analyse und Verbesserung der Experten-Leistungsmodelle	135	
8.1.	1 Experten-Leistungsmodell des BLOB-Detektors	135	
8.1.	2 Experten-Leistungsmodell des Template Matchings	137	
8.1.	3 Experten-Leistungsmodell der Klassifizierungskaskade	139	
8.1.	4 Experten-Leistungsmodell des DPM-Detektors	141	
8.2	Analyse der Leistungsmodellierung durch ML-Verfahren	142	
8.2.	1 Verwendete Methoden zur Analyse der Leistungsmodelle	143	
8.2.	2 Analyse der Metadaten	145	
8.2.	3 Überprüfung auf Nichtlinearität	146	
8.2.	4 Vergleich der ML-Verfahren zur Leistungsmodellierung	147	
8.3	Evaluierung der Algorithmenauswahl	152	
8.3.	1 Verfahren zur Evaluierung der Algorithmenauswahl	153	
8.3.	2 Mission: Observierung	154	
8.3.	3 Mission: Unfallszenario	167	
8.3.	4 Mission: Gebietsaufklärung I	169	
8.3.	5 Mission: Gebietsaufklärung II	172	
8.3.	6 Mission: Gebietsaufklärung III	179	
Kapitel 9	Diskussion und Ausblick	187	
9.1	Zusammenfassende Diskussion der Untersuchungsergebnisse	187	
9.1.	1 Diskussion der Untersuchungsgegenstände	187	
9.1.	2 Diskussion der erreichten Forschungsziele	191	
9.2	Diskussion von Kritikpunkten	192	
9.2.	1 Unvollständige Erfassung der Umgebungszustände	192	
9.2.	2 Defizite in der Leistungsmodellierung durch ML-Verfahren	192	
9.2.	3 Einschränkungen in der Erweiterbarkeit	193	
9.2.	4 Nachteile durch Verwendung synthetischer Daten	193	

9.	3	Ausblick	194		
9.	4 '	Technisch-wissenschaftliche Beiträge	195		
Lite	Literaturverzeichnis				
Anh	ang		209		
А	Koor	dinatensysteme	209		
В	Sensor-Footprint				
С	DPM-Modelle				
D	Zugehörigkeitsfunktionen der modellierten FIS				
E	Missi	ionsabschnitte der Observierungsmission	216		

Symbolverzeichnis

- Q_r Niederschlagsmenge in [mm/h]
- ρ_f Nebeldichte
- Wo Bewölkungsgrad
- t Tageszeit
- M Monat
- *θ* Elevationswinkel in [°]
- *g* Bodenpixelauflösung in [m/px]
- *t_w* Anteil der Bodenbedeckung, welcher aus *Wasser* besteht
- *t*os Anteil der Bodenbedeckung, welcher aus *Ebene* besteht
- *t_b* Anteil der Bodenbedeckung, welcher aus *Gebäuden* besteht
- *t_s* Anteil der Bodenbedeckung, welcher aus *Straßen* besteht
- t_v Anteil der Bodenbedeckung, welcher aus *Vegetation* besteht
- *r'* Geländeform (Reliefwert)
- *a_i* Ein Algorithmus zur Sensordatenverarbeitung
- M_i^{EXP} Experten-Leistungsmodell des Algorithmus a_i
- M_i^{NEF} NEFPROX-Leistungsmodell des Algorithmus a_i
- M_i^{KNN} KNN-Leistungsmodell des Algorithmus a_i
- f_x Umgebungszustand
- *{F, P}* Metadaten zur Erzeugung der Leistungsmodelle
- $x \in X$ Menge der Sensordaten x zur Generierung der Metadaten im Problemraum X
- $f_x \in F$ Menge der Umgebungszustände der Sensordaten x im Merkmalsraum F
- $p \in P$ Menge der Leistungsindizes p im Leistungsraum P
 - A Menge der Algorithmen
 - G Perzeptionsgraph
 - *c_j* Eine Perzeptionskette im Perzeptionsgraphen
 - ${\mathcal M}$ Menge aller Leistungsmodelle
- {*X*, *F*} Modelldatensatz zur Erzeugung der Leistungsmodelle
 - $ilde{\mathcal{E}}$ Evaluierungsdaten
 - X_e^i Sensordaten der Evaluierungsdaten des Experimentes i
 - F_e^i Umgebungszustände der Evaluierungsdaten des Experimentes i
 - P_e^i Leistungsindizes der Evaluierungsdaten des Experimentes i
 - *b* Bias eines Objektklassifikators
 - f_g Skalierungsfaktor zur Normierung der Bodenpixelauflösung g
 - *f*_r Skalierungsfaktor zur Normierung des Reliefwerts *r*
 - *q*_i Anzahl der Zugehörigkeitsfunktionen einer Fuzzy-Variable
- $\mu_j^{(i)}$ Zugehörigkeitsfunktion *j* der Fuzzy-Eingangsvariable *i*
- $v_l^{(j)}$ Zugehörigkeitsfunktion *j* der Fuzzy-Ausgangsvariable *i*
- R_k Fuzzy-Regel
- A_k Bedingung einer Fuzzy-Regel R_k
- C_k Schlussfolgerung einer Fuzzy-Regel R_k

- t_p Anzahl der richtig klassifizierten positiven Beispiele
- t_n Anzahl der richtig klassifizierten negativen Beispiele
- f_p Anzahl der falsch klassifizierten positiven Beispiele
- f_n Anzahl der falsch klassifizierten negativen Beispiele
- F_1 F_1 -Maß
- *W* Breite eines Bildes in Pixeln
- *H* Höhe eines Bildes in Pixeln
- q_p Gütemaß

Abkürzungsverzeichnis

ACVS	Autonomes Computer Vision System
ANFIS	Adaptive-network-based fuzzy inference system
AP	Average precision
BKS	Bodenkontrollstation
BLOB	Binary Large OBject
BpS	Bild pro Sekunde
CC	Cascading classifier
CCL	Connected Component Labeling
CNN	Convolutional neuronal network
DG	Diskrete Gleichverteilung
DL	
DNN	Deep neuronal network
DPM	Deformable Part Model
DT	
FA	
FIS	Fuzzy-Inferenzsystem
FOV	Field of view
GEORIA	Geographic object-based image analysis
CIS	Geographie object bused image unarysis
GNC	Guidance Naviaation and Control
CSD	Cround sample distance
HMI	Human-Machine-Internface
НОС	Histogram of Orignted Cradients
	Künstliches neurongles Netz
	I ocal Pinary Dattorn
	Lincar Correlation Coofficient
	light detection and ranging
LWIR-Sensoren	Langweilige Infrarot-Sensoren Multi sum ad han dit
MAD	
MB-LBP	Multi-scale Local Binary Pattern
MCE	
MIN T	
MUM-1	Manned-Unmanned Teaming
NEFPROX	NEuro Fuzzy function apPROXimator
NMS	Non-Maximum-Suppression
РСА	Principal Component Analysis
PG	Perzeptionsgraph
РК	Perzeptionskette
PM	Perzeptionsmodul
POMDP	
ROC-Kurve	Receiver-Operating-Characteristic-Kurve
ROI	Region of Interest
SAR	Synthetic Aperture Radar
SIFT	Scalar invariant feature transform
SM	Sensormanagement
SPM	Sensor- und Perzeptionsmanagement
SPMS	Sensor und Perzeptionsmanagementsystem
ST	Straßensegmentierung
SVM	Support Vector Machine
TM	Template Matching
UAV	Unmanned aerial vehicle
VG	Vordergrundsegmentierung
	-

VIS-Sensoren	visuelle bildgebende Sensoren
WBM	Wissensbasierte Methoden

KAPITEL 1 EINLEITUNG

Unbemannte Luftfahrzeuge (engl. *unmanned aerial vehicles*, UAV) finden mittlerweile einen breiten Zuspruch sowohl im professionellen wie privaten Umfeld, wobei die Gewinnung und Auswertung von Bild- und Videomaterial für verschiedenste Zwecke im Anwendungsvordergrund stehen. Die vorliegende Arbeit behandelt den adaptiven Sensoreinsatz durch Optimierung automatisierter Sensordatenauswertung und deren Einsatz an Board unbemannter Luftfahrzeuge.

Dieses Kapitel gibt zunächst einen Einblick in typische UAV-Anwendungen. Anschließend wird auf die Herausforderungen bei der Automatisierung der Sensordatenverarbeitung bei solchen Anwendungen eingegangen und der Forschungsbedarf erfasst, um daraus die Forschungsziele zu entwickeln. Abschließend gibt dieses Kapitel einen Gesamtüberblick über diese Arbeit.

1.1 Eingliederung der Arbeit

Die letzte Dekade sah vermehrt Anstrengungen, den Automatisierungsgrad von unbemannten Luftfahrzeugen zu steigern [1]. Dies betraf sowohl Stabilisierung, Regelung und Führung der fliegenden Plattform, als auch die rechnergestützte Verarbeitung der durch die mitgeführte Sensornutzlast gewonnen Daten. Ziel ist es, die Flexibilität, Effektivität und Sicherheit beim Einsatz solcher Systeme in unterschiedlichsten Einsatzbereichen zu erhöhen.

Heutzutage werden unbemannte Luftfahrzeuge u. a. zur Verkehrsüberwachung [2], zur Fernerkundung [3], im Katastrophenschutz [4], zur Vermisstensuche [5], bei der effizienten Bewirtschaftung von Agrarflächen [6] bis hin zur Infrastrukturinspektion [7] eingesetzt. Die Darstellungen in Bild 1.1 zeigen jeweils ein Beispiel für die genannten Verwendungen. Dabei kommen unterschiedliche Sensortypen zum Einsatz, die für die entsprechende Applikation geeignet sind.

Die hauptsächlichen Vorteile von unbemannten gegenüber konventionellen Luftfahrzeugen liegen in der Kostenreduktion sowie dem flexiblen und zeitgerechten Einsatz. Des Weiteren führt die Automatisierung von UAVs zur Einsparung von Personal und erlaubt die Parallelisierung von Aufgaben, wenn mehrere UAV-Systeme zusammen betrieben werden [8]. Daneben tragen solche Systeme zu Sicherheitsaspekten für den Piloten bzw. Operateur in Gefahrensituationen bei. Weiterhin erlaubt die räumliche Trennung zwischen dem Piloten und dem UAV erweiterte Flugmanöver innerhalb der Flugenveloppe. Diese Vorteile führen dazu, dass die unbemannten Luftfahrzeuge flexibler, effizienter und sicherer eingesetzt werden können als konventionelle Luftfahrzeuge und damit auf Abruf Luftbildaufnahmen akquiriert und analysiert werden können.



Bild 1.1: Beispielbilder verschiedener UAV-Anwendungen. In (a) ist eine Verkehrsüberwachung [9], in (b) eine Wildtierbeobachtung, in (c) eine Waldbrandbekämpfung [10], in (d) eine Vermisstensuche [11], in (e) eine Agrarflächenanalyse [12] und in (f) eine Inspektion von Photovoltaikanlagen [13] zu sehen.

Beim Einsatz eines größeren UAVs sind typischerweise mehrere Operateure involviert, die das UAV bedienen. Ein charakteristisches Systemschaubild zur konventionellen Führung eines UAVs [14–16] ist in Bild 1.2 dargestellt¹. Auf der Seite des unbemannten Luftfahrzeuges befinden sich zum einen als systemisches Element die Flugsteuerung (GNC, engl. *Guidance, Navigation and Control*) mit zugehörigen Flugsensoren und zum anderen die Missionssensoren, welche als Nutzlast mitgeführt werden. Über einen Datenlink führen und überwachen die Operateure mittels Benutzerschnittstellen (HMI, engl. *Human-Machine-Interface*) in der Bodenkontrollstation (BKS) den Einsatz des UAVs. Je nach Art und Größe des UAVs sind die Rollen der Operateure auf bestimmte Aufgaben verteilt.



Bild 1.2: Systemschaubild einer konventionellen UAV Führung.

Grundsätzlich wird zwischen einem Plattform- und einem Sensoroperateur [17] unterschieden. Dabei ist der Plattformoperateur für die Flugführung zuständig und der Sensoroperateur für die Bedienung der Missionssensoren. So ist dieser z. B. mit der Sensorkonfiguration, -positionierung und -datenvorauswertung betraut. Dabei kann der Sensoroperateur bei der

¹ Das im Bild verwendete Luftfahrzeug-Abbildung wurde von Julian Herzog [164] erstellt.

Datenanalyse durch ein System zur Sensordatenverarbeitung unterstützt werden, indem Sensordaten gespeichert, annotiert oder aufbereitet werden.

Aus Gründen der Personalkosten und der Entwicklung von Multi-UAV Systemen sind Entwicklungen zu verzeichnen, die das Verhältnis der Anzahl von Operateuren zur Anzahl der geführten UAVs verringern. Im einfachsten Fall findet dabei eine Fusion des Plattform- und Sensoroperateurs statt, was zu einem 1:1-Verhältnis führt. Um die Arbeitsbelastung des Operateurs zu reduzieren und damit ein bedienbares System zu realisieren, werden bereits hierfür geeignete Automatisierungsansätze zur Flugführung und Sensordatenauswertung verfolgt [18]. Mit steigender Anzahl an zu führenden UAVs erhöht sich die Führungsspange. Dadurch ist mit einer weiter erhöhten Arbeitsbelastung des Operateurs [8] zu rechnen. Letztlich stellt das Manned-Unmanned Teaming (MUM-T) Konzept [19] höchste Anforderungen an den Operateur und die involvierten unbemannten Luftfahrtsysteme. Hierbei interagiert ein Besatzungsmitglied in einem bemannten Luftfahrzeug mit mehreren unbemannten Systemen. Damit auch hier noch ein handhabbare Führung der UAVs möglich ist, wird eine Automatisierung auf Aufgabenlevel empfohlen [20]. Damit dies möglich ist, muss die Aufgabe des Plattform- und Sensoroperateurs durch entsprechende Systeme maschinell substituiert bzw. unterstützt werden. Die vorliegende Arbeit beschäftigt sich im Folgenden mit der Automatisierung der Aufgaben des Sensoroperateurs durch adaptiven Sensoreinsatz. Insbesondere wird dabei die Aufgabe der maschinellen Sensorbildauswertung behandelt.

1.2 Feststellung des Forschungsbedarfs

Zur Automatisierung der oben genannten UAV-Anwendungen (siehe Bild 1.1) kann eine geeignete maschinelle Sensordatenverarbeitung eingebracht werden. Dadurch ist es dem Sensoroperateur möglich, perzeptive Aufgaben teilweise oder komplett an ein entsprechendes System zu delegieren. Im Anwendungsbereich der luftgestützten Bodenaufklärung sollte die Sensordatenverarbeitung robust gegenüber sich verändernden Umgebungszuständen, wie etwa Wetterbedingungen, Landschaftstypen oder Perspektiven, während einer Mission sein, um zuverlässige Ergebnisse zu ermöglichen.

Damit die Sensordatenverarbeitung (z. B. zur Bildregistrierung, Objekterkennung oder Objektverfolgung) zuverlässige Ergebnisse liefert, sind diese zum Zeitpunkt der Entwicklung für bestimmte operationelle Gegebenheiten ausgelegt. Zum einen stellt dies eine Limitierung hinsichtlich einer breiten Anwendbarkeit dar, und zum anderen können fehlerhafte Ergebnisse entstehen, wenn die Sensordatenverarbeitung außerhalb der angenommenen operationellen Grenzen eingesetzt wird. Spezifisch im Anwendungsbereich der luftgestützten Bodenaufklärung können sich Umgebungszustände, wie etwa Wetterbedingungen, Landschaftstypen oder Perspektiven während einer Mission verändern und damit die Zuverlässigkeit der Sensordatenverarbeitung beeinflussen.

Die Beeinflussung von sensordatenverarbeitenden Algorithmen durch Umweltgrößen ist bekannt und wird in der Literatur diskutiert. So stellte Gleason et al. [21] eine luftgestützte Fahrzeugerkennung mittels Farbkameras vor und kam zu der Erkenntnis, dass die Leistung des gewählten Ansatzes von topographischen Veränderungen abhängt. Dabei führen topographische Elemente wie Vegetation, Seen, Gebäude und Straßen zu einer geringeren Erkennungsgenauigkeit als gleichmäßig strukturierte Oberflächen, wie etwa Grünflächen. Des Weiteren postuliert Hoiem et al. [22], dass die Robustheit der Sensordatenverarbeitung durch perspektivisches Wissen verbessert wird. Dabei kommen Informationen über die zu betrachtende Szene, z. B. die Größenverhältnisse oder die Lage der Bodenoberfläche, auf der Objekte von Interesse befinden, zur Anwendung. Eine Übersicht sich über Datenverarbeitungsalgorithmen für die Verkehrsüberwachung findet sich bei Kanistras et al. [2]. Die Autoren kommen zu dem Schluss, dass atmosphärische Veränderungen (Witterung) die Algorithmenleistung einschränken und, je nach Situation, andere oder mehrere komplementäre Sensoren verwendet bzw. ausgewählt werden müssen.

Hinsichtlich einer situationsgerechten Auswahl von Sensoren und nachgeordneter Verarbeitungsmechanismen widmet sich das Themenfeld Sensormanagement den Fragestellungen, wann, wo und welcher Sensor unter sich verändernden Randbedingungen verwendet werden soll. Hierzu wurde der grundlegende Ansatz von Hero [23] präsentiert, welcher vertieft im Abschnitt 2.4.1 behandelt wird. Der Ansatz des Sensormanagements sieht zwar eine Bewertung der Sensordatenverarbeitung vor, um die Qualität der Ergebnisse einzuschätzen, allerdings wird diese als gegeben betrachtet bzw. modelliert (z. B. mit Wahrscheinlichkeitsverteilungen). In [24] wird außerdem festgehalten, dass, je nach Situation, die Leistungen von Objekterkennungsmethoden auf Luftbildaufnahmen variieren. Deshalb wird in [25] darauf hingewiesen, dass ein effizientes Sensormanagement die Leistung der Sensordatenverarbeitung in Abhängigkeit der Randbedingungen ermitteln muss, um diese ebenfalls optimieren zu können. Die Optimierung ist dabei sowohl durch die Parameter der Sensordatenverarbeitung, als auch durch eine Algorithmenauswahl [26] in Abhängigkeit der Randbedingungen möglich. Dabei wird in [27] ein Ansatz aus dem Bereich des maschinellen Lernens verwendet, um in Abhängigkeit geeigneter Bildmerkmale den leistungsstärksten Bildaufbereitungsalgorithmus auszuwählen. Des Weiteren macht sich Hochgeschwender et al. [28] das Verfahren der Algorithmenauswahl zunutze, um auf Basis einer Leistungsmessung zwischen verschiedenen sensordatenverarbeitenden Algorithmen adaptiv wechseln zu können. Dadurch konnte für eine Objekterkennung in schwankenden Belichtungsverhältnissen die gesamte Detektionsleistung gesteigert werden. In [29] wird ein Objekterkennungssystem vorgestellt, welches je nach Tageszeit einen Wechsel der Objekterkennungsalgorithmen durchführt, um die Systemleistung zu maximieren. In der Domäne der Luftfahrt stellt Russ et al. [30] ein wissensbasiertes System vor, welches je nach den vorherrschenden Umgebungsbedingungen und der Aufgabe, z. B. Objekterkennung oder -verfolgung, eine geeignete Sensordatenverarbeitung zusammenstellt und anwendet.

1.3 Definition des Forschungsziels

Die vorherigen vorgestellten Arbeiten nutzen kontextspezifische Informationen, wie z. B. Belichtungsverhältnisse oder Tageszeit, um damit die Sensordatenverarbeitung anzupassen. Hierbei werden allerdings entweder nur theoretische Konzepte diskutiert oder lediglich einzelne Umgebungsparameter experimentell untersucht. Daher soll diese Arbeit ein übergreifendes Konzept, inklusive zugehöriger Methodik, und dessen Umsetzung für den automatisierten optimierten Sensoreinsatz an Bord eines UAVs behandeln, unter der Berücksichtigung mehrerer situativer Umgebungsparameter. Dieses Konzept wird am Beispiel der Fahrzeugdetektion auf bildbasierter Sensorik evaluiert. Die zugrundeliegende Idee ist es, die Leistung verschiedener, spezialisierter sensordatenverarbeitender Algorithmen in Abhängigkeit von unterschiedlichen Umgebungszuständen zu modellieren, um darauf aufbauend den aktuell am geeignetsten Algorithmus zu bestimmen. Über die Ermittlung der Leistungscharakteristiken der Algorithmen kann eine Aussage über deren Einsatzbarkeit getroffen und letztlich die Sensordatenverarbeitung optimiert werden. Zur Erreichung des übergeordneten Forschungsziels werden die folgenden Zwischenziele eingeführt:

- **Ziel 1**: Entwicklung eines Konzepts zum adaptiven luftgestützten Sensoreinsatz unter Berücksichtigung wechselnden Umgebungszustände
- Ziel 2: Implementierung verschiedener sensordatenverarbeitender Algorithmen zur luftgestützten Fahrzeugerkennung mit situativ unterschiedlichen Leistungseigenschaften
- Ziel 3: Modellierung und Analyse der Einflüsse ausgewählter Umgebungszustände auf die sensordatenverarbeitenden Algorithmen hinsichtlich deren Leistungsverhaltens
- Ziel 4: Umsetzung der Algorithmenauswahl in ein prototypisches System
- Ziel 5: Evaluierung der Algorithmenauswahl anhand von repräsentativen Missionen

1.4 Inhaltsübersicht

Das folgende Kapitel fasst den Stand der Technik zu relevanten Themengebieten zusammen, um Lösungsansätze aufzudecken. Dazu wird zunächst in Abschnitt 2.1 auf die typische eingesetzte Missionssensorik eingegangen, um anschließend in Abschnitt 2.2 geeignete sensordatenverarbeitende Algorithmen zur Fahrzeugerkennung auszuwählen. Im Anschluss daran erörtert Abschnitt 2.3 mögliche Faktoren, welche das Leistungsverhalten der sensordatenverarbeitenden Algorithmen beeinflussen. Um diesen Beeinflussungen zu entgegenzuwirken, werden in Abschnitt 2.4 Ansätze vorgestellt, die ein Management des Sensoreinsatzes und der Sensordatenverarbeitung vornehmen. Abschließend wird gefolgert, dass über die Ermittlung der Algorithmenleistung anhand der Einflussfaktoren eine Optimierung der Sensordatenverarbeitung erreicht werden soll. Dazu werden in Abschnitt 2.5 Verfahren zur Auswahl von Algorithmen präsentiert.

Im Kapitel 3 wird das technisch-funktionale Konzept zur adaptiven luftgestützten Sensordatenverarbeitung mittels Algorithmenauswahl vorgestellt. Hierzu werden in Abschnitt 3.1, anhand der Erkenntnisse aus dem Stand der Technik, die funktionalen und systemischen Anforderungen abgeleitet. Anhand der Anforderungen wird das Konzept in Abschnitt 3.2 entwickelt und vorgestellt. Dabei wird zwischen der Leistungsbestimmung der sensordatenverarbeitenden Algorithmen und deren operationellem Einsatz unterschieden. Daraufhin erfolgt in Abschnitt 3.3 eine Methodensichtung und -auswahl zur Umsetzung der Komponenten des Konzepts. Anhand der ausgewählten Methoden wird das vorgestellte Konzept aus Abschnitt 3.2 in Abschnitt 3.4 detailliert. Abschließend stellt Abschnitt 3.5 die Evaluierungsstrategie vor.

Die Verfahren zur Vorbereitung der Leistungsbestimmung in Kapitel 4 behandeln die Generierung der benötigten Daten, um den Bezug zwischen Umgebungszustand und Algorithmenleistung herzustellen. Anschließend stellt Kapitel 5 die Verfahren zur Leistungsbestimmung vor, wobei zwischen Modellen mit Expertenwissen und Modellen, die durch Methoden des maschinellen Lernens erzeugt wurden, unterschieden wird. Kapitel 6 zeigt die Implementierung der Algorithmenauswahl mittels der Leistungsbestimmung und deren Integration in ein System zum adaptiven Sensoreinsatz.

Um das verwendete Verfahren zur Algorithmenauswahl zu evaluieren, stellt Kapitel 7 die aus den Forschungszielen abgeleiteten Untersuchungsgegenstände und entsprechende Experimente vor. Die Auswertung der Ergebnisse der Experimente erfolgt im anschließenden Kapitel 8. Abschließend werden in Kapitel 9 die Ergebnisse diskutiert. Außerdem wird ein Ausblick gegeben und es werden die technisch-wissenschaftlichen Beiträge dieser Arbeit zusammengefasst.

KAPITEL 2 STAND DER TECHNIK

Dieses Kapitel stellt zunächst relevante Missionssensoren sowie deren Eigenschaften vor und wählt aus diesen geeignete für die Applikation der luftgestützten Fahrzeugerkennung aus. Anschließend werden Sensordatenverarbeitungsroutinen zur Fahrzeugerkennung vorgestellt und hinsichtlich ihrer Anwendbarkeit bewertet. Um den Einfluss von Umgebungsbedingungen auf die Sensordatenverarbeitung zu identifizieren, werden wesentliche Einflussfaktoren auf die verwendete Sensorik und die Datenverarbeitung beschrieben. Zum Management von Sensorik und Sensordatenverarbeitung werden nachfolgenden signifikante Ansätze aus der Literatur wiedergegeben. Daraus wird gefolgert, dass eine Algorithmenauswahl zum adaptiven luftgestützten Sensoreinsatz praktikabel ist. Infolgedessen werden Ansätze zur Algorithmenauswahl betrachtet und bewertet.

2.1 Missionssensorik und deren Eigenschaften

Für UAV-gestützte Überwachungs-, Aufklärungs- sowie Fernerkundungsmissionen werden unterschiedliche Sensoren verwendet. Vorranging eingesetzt werden dabei bildgebende Sensoren, die im visuellen Spektrum arbeiten, da ein menschlicher Operateur die Sensordaten leicht interpretieren kann oder diese mit den aktuellen Methoden der Bildverarbeitung computergestützt ausgewertet werden können. Die bildgebenden Sensoren werden u. a. bei Suchund Rettungsdiensten [31], Infrastrukturinspektionen [32] oder im Katastrophenschutz [33] eingesetzt. Durch die zunehmend steigende optische Auflösung können zunehmend mehr Details der Umgebung erfasst werden. Daneben besitzen die Sensoren eine relativ geringe Energieaufnahme und können durch das breite Angebot des Massenmarktes kostengünstig erworben werden. Operationell weisen diese visuellen bildgebenden Sensoren (VIS-Sensoren) natürlich den Nachteil auf, dass sie ohne externe Lichtquelle kaum verwertbare Informationen liefern.

Ebenso für Tag- aber auch Nachtsicht kommen in der Fernerkundung und Überwachung Wärmebildkameras zum Einsatz, die sensitiv im mittleren bis langwelligen Infrarotspektrum sind. Sie wurden zunächst im militärischen Kontext und später auch in zivilen Anwendungen wie z. B. der Waldbrandüberwachung [34] oder Vermisstensuche [35] verwendet. In den letz-ten Jahren wurden erhebliche Fortschritte in der Miniaturisierung und Auflösung dieser Sensoren erzielt [36]. Langwellige Infrarot-Sensoren (LWIR-Sensoren) bieten ebenfalls wie VIS-Sensoren den Vorteil, dass sie eine relativ geringe Energieaufnahme besitzen. Gewicht und Energieaufnahme können durch den Einsatz von ungekühlten Mikrobolometer weiter reduziert werden. Allerdings gehen damit die Nachteile eines erhöhten Bildrauschens und die Notwendigkeit einer regelmäßigen Kalibrierung zur Temperaturdriftkompensierung mit Einher.

Das Weitern finden sich *Multispektralkameras*, die in der Lage sind mehrere dezidierte Spektralbereiche aufzunehmen. Gerade im Bereich des Präzisionsackerbaus werden solche Sensoren [37] verwendet, die für spezifische Spektralbereiche im nahen und mittleren Infrarotbereich sensitiv sind, um damit z. B. den Gesundheitszustand der Vegetation zu überprüfen. Außerdem werden *Hyperspektralkameras* eingesetzt, die mehrere zusammenhängende Spektralbereiche aufnehmen können, welche von ultravioletten bis zum langwelligen Infrarotbereich reichen können. Diese Kameras werden primär zur Analyse von Vegetationszuständen [38] verwendet. Sowohl Multispektral- als auch Hyperspektralkameras erfassen ein größeres Spektrum als VIS- und LWIR-Sensoren, haben aber den Nachteil, dass sie relativ geringe Bildwiederholraten aufweisen und keine Zoomobjektive verfügbar sind.

Neben der Stereoskopie kommen inzwischen *light detection and ranging* (LIDAR) Sensoren zur Entfernungsmessung, 3D-Rekonstruktion oder Hinderniserkennung [39] zur Anwendung. Gegenüber der Stereoskopie besitzen LIDAR Sensoren eine höhere Genauigkeit der Tiefeninformation, sind aber bzgl. Größe, Gewicht und Energieaufnahme weniger geeignet für kleine UAV Systeme. Außerdem ist die Reichweite des Sensors limitiert.

Nur für größere UAVs und Satelliten bieten sich auch Synthetic Aperture Radare (SAR) an, da sie im Vergleich mit den anderen genannten Sensoren sehr groß (Antennenlänge ca. 1 – 4 Meter) sind. Sie werden unter anderem für die Erkennung von Klimaänderungen, die fernerkundliche Veränderungsdetektion (engl. *change detection*) und Kartierungsanwendungen verwendet. Dabei bieten SAR Sensoren den Vorteil, dass sie hochaufgelöste Bilder erzeugen können, ohne dass die Sensordaten von atmosphärischen Effekten [40] beeinflusst werden. Allerdings besitzen sie aktuell gegenüber den anderen Sensoren eine geringere Bodenauflösung und höhere Verarbeitungszeiten bei der Sensordatenakquise [41].

Sensor	Anwendungen	Gewicht	Leistungsaufnahme	Limitierungen
VIS	Such- und Rettungsdienst	~ 0,2 kg	~ 3 W	Fremde Lichtquelle
	Infrastrukturinspektion			notwendig
	Katastrophenschutz			
LWIR	Tag- und Nachtsicht	~ 0,1 kg	~ 1 W	Bildrauschen
(ungekühltes	Waldbrandüberwachung			
Mikrobolometer)	Vermisstensuche			
Multispektral	Präzisionsackerbau	~ 1 kg	~ 15 W	Bildfrequenz ~ 1 BpS
Hyperspektral	Präzisionsackerbau	~ 1 kg	~ 2 - 20 W	Bildfrequenz ~ 1 BpS
LIDAR	3D-Rekonstruktion	~ 0,8 kg	~ 8 W	Reichweite < 100 m
	Hinderniserkennung			
	Entfernungsmessung			
SAR	Klimaforschung	> 2 kg	> 10 W	Auflösung 0,1 m
	Veränderungsdetektion			
	Kartierung			

Tabelle 2.1: Gegenüberstellung der Sensorarten mit Auswahl.

In Abschnitt 1.3 wurde bereits festgelegt, dass die Beispielapplikation einer luftgestützten Fahrzeugerkennung als Untersuchungsgegenstand dient. Um diese zu realisieren benötigt es einen Sensor mit einer ausreichenden Orts- und Zeitauflösung. Des Weiteren bieten sich für die Untersuchungen kommerzielle Systeme (Mini bzw. Micro UAVs [42]) an, da sie gegenüber größeren UAVs eine einfachere Handhabung ermöglichen. Tabelle 2.1 enthält einen Überblick über die verschiedenen vorgestellten Sensorarten sowie deren Eigenschaften. Wegen der Vorteile in Bezug auf Größe, Gewicht und Energieaufnahme der VIS- und LWIR-Sensoren, sind diese besonders für kleine UAVs geeignet. Außerdem gehören sie zu den Standardsensoren, um eine Fahrzeugerkennung durchzuführen. Daneben lassen sich die bildgebenden Sensoren in kardanische Aufhängungen integrieren, welche den Vorteil bieten, die Sensorik unabhängig von der Flugzeuglage zu bewegen. Aus diesen Gründen sind VIS- und LWIR-Sensoren von besonderem Interesse für die Anwendung der Fahrzeugerkennung, da sie sich hinsichtlich der

Tag- und Nachtsicht ergänzen. Infolgedessen stellt folgender Abschnitt die Verfahren zur luftgestützten Fahrzeugerkennung mittels bildgebender Sensorik vor.

2.2 Maschinelle luftgestützte Fahrzeugerkennung mit bildgebender Sensorik

Maschinelle Verfahren im Bereich der luftgestützten Fahrzeugerkennung mit bildgebender Sensorik setzten vornehmlich Verfahren aus dem Bereich des maschinellen Sehens (engl. *Computer Vision*) ein. Sie lassen sich nach [24] in vier Gruppen einteilen:

- Methoden, die auf *Template Matching* basieren
- Wissensbasierte Methoden
- Geographic object-based image analysis (GEOBIA) Verfahren
- Methoden des maschinellen Lernens

Innerhalb dieser Gruppen kann weiterhin zwischen Klassifikatoren und Objektdetektoren unterschieden werden. Eine Klassifikation behandelt dabei die Zuordnung eines Bildes bzw. Merkmals zu einer Klasse (z. B. Fahrzeug, Person oder Gebäude) und eine Detektion umfasst die Klassifikation und eine Lokalisierung ("wo befindet sich das Objekt"). Das Template Matching benötigt zunächst ein Objektmodell (Template) bevor die eigentliche Klassifizierung erfolgen kann. Das Objektmodell wird typischerweise aus einem Referenzbild des zu suchenden Objektes erzeugt. Anschließend erfolgt mittels Ähnlichkeitsmessung zwischen dem Objektmodell und einem Bild die eigentliche Klassifikation. Wissensbasierte Methoden verwenden geometrische (z. B. Größe oder Form) oder kontextbezogene Eigenschaften des Objekts (z. B. Schattenwurf), um dieses zu klassifizieren. Dabei basiert die Objektklassifikation meist auf modellierten Regeln. GEOBIA-Verfahren segmentieren das Bild in räumlich zusammenhängende Bereiche, die mittels geographischem Informationssystem (GIS) semantisch beschrieben werden. Anschließend erfolgt auf Basis der Segmentierung eine Objektklassifikation mittels Regeln, welche aus Expertenwissen abgeleitet werden. Die Methoden des maschinellen Lernens verwenden Trainingsdaten, welche Bilder mit Objektannotationen enthalten, um damit einen Klassifikator zu trainieren. Dazu werden zunächst Merkmale aus den Trainingsbeispielen extrahiert. Anschließend wird mittels überwachten Lernens ein Modell generiert, welches anhand der Merkmale z. B. zwischen Fahrzeugen und Nicht-Fahrzeugen entscheiden kann. Die Stärken und Schwächen der einzelnen Methoden sind in Tabelle 2.2 zusammengefasst.

Methode	Stärken	Limitierungen
Template Matching	Adaptierbar auf andere Objekte durch Austausch des Templates;	Rechenintensiv; Probleme mit Skalierungen, Rotationen oder perspektivischen Veränderungen
Wissensbasierte Methoden	Die Objektdetektion wird in einer hierarchischen Struktur durchgeführt, welche von groben bis zu feinen Regeln reicht	Expertenwissen zur Konstruktion der Klassifikationsregeln ist subjektiv; Sind die Regeln zu weit gefasst, können Objekte irrtümlich detektiert werden. Gleiches gilt umgekehrt für zu eng gefasste Regeln
GEOBIA Verfahren	Kontextsensitive Objektklassifikation durch die Verwendung von mehreren Informationsquellen (Bildmerkmale, GIS, Expertenwissen)	Allgemeine Lösung zur automatischen Segmentierung nicht verfügbar; Expertenwissen zur Konstruktion der Klassifikationsregeln ist subjektiv
Maschinelles Lernen	Objektmodell kann automatisch aus Trainingsdaten erzeugt werden; Detektionssystem ist skalierbar und anpassbar	Benötigt genügend Trainingsdaten; Die Detektionsgenauigkeit ist abhängig von den gewählten Trainingsdaten

Tabelle 2.2: Stärken und Limitierung der Methoden zur luftgestützten Objektdetektion.

Das GEOBIA-Verfahren wird hauptsächlich für orthogonale Satellitenbilder verwendet und ist für geringer Bildauflösungen ausgelegt, als dies eine erdgerichteten Fahrzeugdetektion nötig ist. Daher wird auf das GEOBIA-Verfahren verzichtet, und im Folgenden die drei verbleibenden Verfahren betrachtet.

2.2.1 Template Matching und wissensbasierte Methoden

In [43] wird ein Verfahren zur Extraktion von Straßennetzen aus orthogonalen Luftbildern und anschließender Fahrzeugdetektion mittels *Template Matching* (TM) vorgestellt. Hierbei wird das verwendete TM zur Objekterkennung um ein geometrisch deformierbares Modell ergänzt. Dieses sorgt für eine robuste Detektionsleistung gegenüber rotatorischen Einflüssen. Für perspektivische Luftbilder wird in [44] ein Ansatz zur Fahrzeugerkennung mit mehreren Objektvorlagen vorgestellt. Diese Objektvorlagen werden in ein 3D-Modell überführt, welches nicht nur rotatorische, sondern auch perspektivische Effekte kompensieren kann. Dabei wird ein Projektionsmodell verwendet, welches auf Basis der aktuellen Sensorpositionierung das 3D-Modell für die aktuelle Situation anpasst. Weiterhin wird in [45] ein TM auf der Grundlage von Konturen vorgestellt. Damit das Verfahren invariant gegenüber Translationen, Rotationen und Skalierungen ist, wird der *Artificial Bee Colony* (ABC) Algorithmus [46] eingesetzt.

Als klassische wissensbasierte Methode gilt die Extraktion von lokalen Formen, um damit eine Objektklassifikation durchzuführen. Diese lokalen Formen können zum Beispiel die Fläche oder der Umfang eines Objekts sein. In [47] wird eine Fahrzeugerkennung auf Luftbildaufnahmen vorgenommen, die mittels einer Binary Large OBject (BLOB) Detektion und einer Formbeschreibung der Fahrzeuge erfolgt. Der Vorteil dieser Methode ist, dass sehr kleine Objekte erkannt werden können. Damit allerdings die Formbeschreibungen der BLOBs zur Objektklassifikation valide sind, können nur Szenenaufnahmen mit einheitlicher Kamerapose sinnvoll analysiert werden. Des Weiteren wurde in [48] ein BLOB-Detektor auf Sensordaten einer langwelligen Infrarotkamera angewendet, um Fahrzeuge bei Tag- und Nachtsituationen zu erkennen. Dieses Verfahren wurde ebenfalls lediglich auf orthogonal aufgenommenen Luftbildern evaluiert, wobei auch ein Einsatz auf perspektivischen Sensordaten denkbar ist. Eine weitere wissensbasierte Methode zur Detektion von Fahrzeugen auf Parkflächen wird in [49] vorgestellt. Hierbei werden Kanten aus dem Bild extrahiert. Über ihre Lage und Geometrie wird die Klassifikation vorgenommen. Bei dieser Methode wurde festgestellt, dass bei Helligkeitsschwankungen Probleme auftreten und Rotationen der Kamera zu fehlerhaften Klassifikationen führen.

2.2.2 Methoden des maschinellen Lernens

Ursprünglich wurde von Viola und Jones [50] eine Klassifizierungskaskade (engl. *cascading classifier*, CC) vorgestellt, die gegenüber anderen Verfahren recheneffizient ist und dabei eine hohe Detektionsrate erreicht. Verwendet wurde die Methode von Xu et al. [51,52], um Fahrzeuge auf Straßen in Luftbildaufnahmen zu detektieren. Dabei kann der Detektor lediglich Fahrzeuge mit einer bestimmten Ausrichtung erkennen. Deshalb wird durch eine Vorverarbeitung die Straße segmentiert und in eine Normalorientierung gebracht, damit Fahrzeuge immer identisch ausgerichtet sind. Dabei ist dieses Verfahren nur bei möglichst geradlinigen Straßensegmenten anwendbar. Eine ähnliche Methode stellte Grabner et al. [53] vor. Der Unterschied liegt darin, dass nicht die Straße in eine Normorientierung überführt wird, sondern die extrahierten Merkmale rotiert werden. Dadurch ist der Detektor invariant gegenüber Objektorientierungen, mit der Einschränkung für orthographische Aufnahmen. In [35] und

[54] werden Klassifizierungskaskaden für verschiedene Objektorientierungen und Perspektiven verwendet, um in dynamischeren Szenen Fahrzeuge erkennen zu können.

Ein weiteres etabliertes Objekterkennungsverfahren ist die Klassifikation mittels Support Vector Machine (SVM) unter Verwendung des Histogram of Oriented Gradients (HOG) Merkmals, welches ursprünglich von Dalal und Triggs [55] vorgestellt wurde. Tuermer et al. [56] verwendet das HOG-Merkmal in Verbindung mit der oben genannten Boosting-Methode, um Fahrzeuge in urbanen Gebieten zu detektieren. Auch hier muss das Fahrzeug eine bestimmte Ausrichtung aufweisen, um detektiert werden zu können. Das gleiche Problem wird auch in [51] diskutiert, wobei hier ein hybrider Ansatz verwendet wird, der auf der vorherigen (Haarlike Merkmale und Klassifikationskaskade) und dieser Methode (HOG-Merkmal und SVM) besteht. Dabei wird das Orientierungsproblem umgangen, indem Straßen segmentiert und in eine Normorientierung gebracht werden. Anschließend wird davon ausgegangen, dass die Fahrzeuge ebenfalls in dieser Orientierung erscheinen. Des Weiteren wird von Gleason et al. [21] die Fahrzeugerkennung mittels HOG-Merkmal und SVM-Klassifikator optimiert, indem der Suchraum durch eine Vorverarbeitung auf der Basis von Farbinformationen eingeschränkt wird. In [57] wird ebenfalls der Suchraum eingeschränkt, indem eine Segmentierung des Bildes vorgenommen wird. Dabei werden solche Regionen aus dem Suchraum für den Klassifikator eliminiert, in denen die Auftrittswahrscheinlichkeit von Fahrzeugen sehr gering ist. Außerdem wird anstatt des HOG-Merkmals das lokale scalar invariant feature transform (SIFT) Merkmal verwendet.

Das *Deformable Part Model* (DPM) stellt ein relativ neues Verfahren dar, welches von Felzenszwalb et al. [58] vorgestellt wurde. Es basiert nicht wie die vorherigen Verfahren auf der Erkennung eines ganzheitlichen Objekts, sondern auf der Detektion von mehreren Teilen des Objekts, die miteinander gekoppelt sind. Außerdem können mehrere Deformationsformen, wie z. B. Orientierungen oder verschiedene Perspektiven, des Objekts erfasst werden. Damit werden die Probleme der vorherigen Verfahren beseitigt. Verwendet wurde diese Methode zur Objekterkennung in Luftbildaufnahmen in [59], um Flugzeuge und Flughäfen, und in [60], um Fahrzeuge bei Nacht, zu erkennen. Die Nachteile des *DPM*-Verfahrens im Vergleich zu den vorherigen Verfahren sind, je nach Modellgröße, die hohe Rechenzeit und die minimal notwendige Objektgröße im Bild.

Der aktuelle Trend führt zum automatischen Lernen der Merkmale mittels *Deep Learning* (DL), um beschreibungsstarke Merkmale und damit höhere Detektionsgenauigkeiten zu erhalten. In [61] wird ein Schifferkennungssystem auf Luftbildaufnahmen vorgestellt, wobei die Merkmale durch tiefe neuronale Netzwerke (engl. *deep neuronal networks*, DNN) gelernt werden. Anschließend wird die Klassifikation durch ein einschichtiges neuronales Netz durchgeführt. Dabei wird im Vergleich zu anderen gängigen Methoden festgestellt, dass durch die Verwendung von DL eine höhere Detektionsgenauigkeit erreicht und zudem der Rechenaufwand reduziert werden kann. Han et al. [62] präsentieren eine Objekterkennungsmethode, die zwei modellierte und ein durch DL erzeugtes Merkmal verwendet, um Objekte in Luftbildaufnahmen auszumachen. Dabei wird zur Klassifikation eine SVM eingesetzt. Es wird gezeigt, dass diese Methode verwendet werden kann, um Fahrzeuge, Flugzeuge und Flughäfen zu erkennen. Ein ähnlicher Ansatz wird in [63] verfolgt, um Fahrzeuge in Luftbildaufnahmen zu identifizieren. Hierbei wird ein *convolutional neuronal network* (CNN) zum Lernen der Merkmale verwendet. Anschließend wird eine SVM zur Objektklassifikation trainiert. Dabei wurde ein bereits gelerntes CNN verwendet, um die Anzahl der nötigen Trainingsdaten zu

	Pub.	Methode	Stärken	Limitierungen
	[43]	TM + Deformation	Rotationsinvariant	Nicht robust gegenüber perspektivischen Variationen
TM	[44]	TM + 3D Modell	Robust gegenüber perspektivischen Variationen; Invariant gegenüber Rotationen, Translationen und Skalierungen	Anfällig gegenüber schwankendem Kontrast
	[46]	TM mit Konturen	Invariant gegenüber Translationen, Rotationen und Skalierungen	Nur orthographische Sensorbilder
	[47]	BLOB	Detektion von relativ kleinen Objekten möglich	Nur orthographische Sensorbilder
WB	[48]	BLOB + LWIR	Objektdetektion ohne Lichtquellen möglich; Einsatz bei perspekti- vischen Variationen denkbar	Nur auf orthographische Sensorbilder evaluiert
	[49]	Kanten	Ressourcen schonend, da lediglich Kanten aus dem Bild extrahiert werden	Nur orthographische Sensorbilder; Fehleranfällig bei Helligkeits- schwankungen
	[52]	CC	Rotationsinvariant durch Orientierung der Straßensegmente	Nur orthographische Sensorbilder; Detektion von Fahrzeugen nur auf Straßensegmenten
	[53]	CC + Rotation	Rotationsinvariant	Nur orthographische Sensorbilder
	[54]	CC + Perspektiven	Robust gegenüber perspektivischen Variationen; Invariant gegenüber Rotationen, Translationen und Skalierungen	Aufwendiges Lernverfahren, da mehrere Objektdetektoren für verschiedene Perspektiven erzeugt werden müssen
	[56]	CC + HOG	Robuste Detektion in urbanen Szenerien	Nicht robust gegenüber Rotationen; Nur orthographische Sensorbilder
	[51]	CC + SVM	Rotationsinvariant durch Orientierung der Straßensegmente	Nur orthographische Sensorbilder; Detektion von Fahrzeugen nur auf Straßensegmenten
	[21]	SVM + HOG	Rotationsinvariant	Nur orthographische Sensorbilder; Nicht robust gegenüber Skalierungen
ML	[57]	SVM + SIFT	Robust gegenüber Rotationen und Skalierungen	Nur orthographische Sensorbilder; GIS nötig mit entsprechenden Informationen über asphaltierte Flächen;
	[59]	DPM	Robust gegenüber perspektivischen Variationen; Invariant gegenüber Rotationen, Translationen und Skalierungen	Die benötigte minimale Objekt- größe ist im Gegensatz zu den anderen Verfahren hoch
	[60]	DPM (Nachtzeit)	Robust gegenüber perspektivischen Variationen; Invariant gegenüber Rotationen, Translationen und Skalierungen	Externe Lichtquellen notwendig (z. B. Straßenlaterne)
	[61]	DNN	Robust gegenüber perspektivischen Variationen; Invariant gegenüber Rotationen, Translationen und Skalierungen	Hohe Anzahl an Trainingsbeispielen notwendig
	[63]	CNN + SVM	Robust gegenüber perspektivischen Variationen; Invariant gegenüber Rotationen, Translationen und Skaliorungen	Bereits trainiertes CNN notwendig; Anpassung an Problemdomäne ist daher nur limitiert möglich

reduzieren. Dem gegenüber konnte durch die Erhöhung der Anzahl der Trainingsdaten, für den Lernprozess der SVM, eine höhere Detektionsgenauigkeit erreicht werden.

Tabelle 2.3: Gegenüberstellung der Verfahren zur luftgestützten Fahrzeugerkennung. Mit Grün sind diejenigen Verfahren markiert, die für die Untersuchung der Algorithmenauswahl in Frage kommen.

2.2.3 Diskussion

Um den Forschungszielen (siehe Abschnitt 1.3) dieser Arbeit gerecht zu werden, sollen verschiedene, alternativ anzuwendende sensordatenverarbeitende Algorithmen verwendet

werden, um deren situationsgerechte Auswahl zu untersuchen. Im betrachteten Anwendungsfall der luftgestützten Fahrzeugerkennung können die Fahrzeuge grundsätzlich in unterschiedlichen Posen im Sensorbild auftreten. Daher sind vor allem Algorithmen geeignet, die robust gegenüber perspektivischen Variationen und invariant gegen Rotationen, Translationen und Skalierungen sind. Zudem sollten die Verfahren möglichst unterschiedliche Ansätze verfolgen, um eine Variabilität hinsichtlich der Leistungscharakteristik zu erhalten. Tabelle 2.3 zeigt eine Gegenüberstellung der vorgestellten Fahrzeugerkennungsmethoden mit den jeweiligen Stärken und Schwächen. Die in der Tabelle grün markiert Verfahren stellen geeignete Kandidaten für das Experimentvorhaben dar. Die abschließende Auswahl wird in Abschnitt 3.3.2 getroffen, da zuvor die Verfügbarkeit von Trainingsdaten für die Verfahren des maschinellen Lernens diskutiert werden muss.

Die vorausgewählten Verfahren unterscheiden sich hinsichtlich der verwendeten Information zur Objektklassifizierung und deren Eigenschaften. Das Template Matching nutzt die rohen Intensitätswerte der Pixel und zeigt sich damit anfälliger gegenüber Kontrast- und Helligkeitsänderungen. Der Vorteil des TM ist die einfache Austauschbarkeit des Modells. Der BLOB-Detektor nutzt zusammenhängende Regionen mit vergleichbaren Intensitätswerten und kann auf LWIR-Sensordaten eingesetzt werden, um Fahrzeuge bei Tag- und Nachtsituationen zu detektieren. Dabei kann Bildrauschen die Regionenbildung negativ beeinflussen. Die Klassifikationskaskade nutzt das Haar-Merkmal, welches die Summe der Intensitätswerte zwischen definierten Regionen gegenüberstellt. Da die Haar-Merkmale sehr klein $(2 \times 2 \text{ Pixel})$ ausfallen können, ist die Erkennung von kleineren Objekten möglich. Das DPM verwendet das HOG-Merkmal, welches auf Basis der Gradienten im Bild berechnet wird. Dadurch ist es sehr robust gegenüber Kontrast- und Helligkeitsschwankungen. Ein Nachteil ist die minimal benötigte Objektgröße. Die Ansätze DNN und CNN nutzen aus Beispieldaten erzeugte Merkmale, wodurch die Detektionsrate gegenüber allen anderen Verfahren höher sein kann. Von Nachteil sind die aufwendigen Trainingsverfahren und die benötigte Menge an Trainingsdaten.

2.3 Einflussfaktoren auf die Sensordatenverarbeitung

Um schließlich eine Algorithmenauswahl anhand der Umgebungsbedingungen zu realisieren, müssen die Einflussfaktoren der Umgebung auf die Sensordatenverarbeitung bestimmt werden. Dazu werden die Eigenschaften der bildgebenden Sensorik vorgestellt, um daraus die relevanten Einflussfaktoren auf die bildverarbeitenden Algorithmen abzuleiten.

2.3.1 Geometrische Eigenschaften von Luftbildaufnahmen

Beim Einsatz von bildgebender Sensorik beeinflussen die geometrischen Eigenschaften, entstehend durch Fluglage und Sensorausrichtung, die Abbildung der Szene auf die Bildebene des Sensors. Eine Luftbildaufnahme wird idealisiert, ohne Berücksichtigung von Verzerrungen durch die Optik, durch zentralperspektivische Abbildung erzeugt, wodurch Gelände bzw. Objektpunkte auf einer Bezugsebene radial nach außen versetzt abgebildet werden. Bild 2.1 zeigt schematisch den Effekt der radialen Punktversetzung [75]. Dabei wird die Höhe des Objektes Δh auf die Bildebene mit Δr_b projiziert. Dies führt im Sensorbild dazu, dass vertikale Objektlinien (z. B. Hauskante, Baumstämme) geneigt dargestellt werden. Der Effekt der radialen Punktversetzung wird größer, wenn der abbildende Strahl flacher wird. Deshalb ist die radiale Versetzung im Bildmittelpunkt nicht zu beobachten und wird in den Ecken des Bildes am größten.



Bild 2.1: Radiale Punktversetzung durch Höhenunterschiede.

Bei dem betrachteten Beispiel der radialen Punktversetzung wurde von der orthographischen Aufnahme einer ebenen Geländefläche ausgegangen. Sobald aber die Aufnahmerichtung geneigt wird oder die Geländefläche uneben ist, wirkt sich dies als Verzerrung [64] aus. In Bild 2.2 ist schematisch der Unterschied zwischen einem unverzerrten und verzerrten Luftbild dargestellt. Des Weiteren führt die Verzerrung zu einer Veränderung der Bodenpixelauflösung. Diese ist neben der Aufnahmerichtung von der Flughöhe und der optischen Vergrößerung (Zoom) des Sensors abhängig. Dabei ist die Bodenpixelauflösung für die Objekterkennung ein entscheidendes Element. Je nach Größe des Objektes muss eine bestimmte Bodenpixelauflösung eingehalten werden, damit das Objekt maschinell erkannt werden kann.



Bild 2.2: Schematische Verzerrung eines Luftbildes bei geneigter Aufnahmerichtung. Bei orthogonaler Aufnahmerichtung wird die Geländefläche unverzerrt (a) und bei zunehmend geneigter Aufnahmerichtung verzerrt dargestellt (b).

Mit Hilfe der geometrischen Zusammenhänge können luftgestützte Bildverarbeitungsalgorithmen robuster entwickelt werden. Hierzu nutzt Hoiem et al. [22] perspektivisches Wissen, um die Grundeben in einer perspektivischen Szene zu ermitteln. Durch Nutzung der möglichen Objektpositionen und -größen wird eine Verbesserung der Objektdetektion erreicht. Des Weitern wird in [65] eine Personenerkennung auf Luftbildaufnahmen präsentiert. Diese Methode verwendet kontextbezogene Informationen, wie Sensorpositionierung und -parameter, um damit die geometrischen Beziehungen der Szene auf das Sensorbild zu bestimmen. Diese Beziehungen werden schließlich verwendet, um den Schattenwurf in Abhängigkeit von der Tageszeit zu berechnen und damit die Personenerkennung zu optimieren. Neben dem Ansatz, Kontextinformationen zur Verbesserung der Algorithmenleistung zu verwenden, wird in [66] ein Verfahren zur Fahrzeugerkennung vorgestellt, welches invariant gegenüber geometrischen Veränderungen ist.

2.3.2 Atmosphärische Einflüsse

Die verschiedenen Sensoren und Algorithmen, die auf UAVs für bildbasierte Aufklärungszwecke zur Anwendung kommen, werden von atmosphärischen Effekten (meist abträglich) beeinflusst. Zu diesen Effekten zählen Wetterphänomene wie Bewölkung, Niederschlag (z. B. Regen oder Schnee), Dunst oder Nebel. Außerdem beeinflusst der Sonnenstand die Erscheinung der Szene.



Bild 2.3: Zusammenhänge der Winkel zwischen Sensor, Objekt und Sonne.

In Bild 2.3 sind die geometrischen Zusammenhänge zwischen der Sensorblickrichtung, dem Elevationswinkel der Sonne und dem Azimutalwinkel zwischen Sensor und Sonne dargestellt. Daraus können die Effekte Mit- und Gegenlicht, Streuung und spiegelnde Reflexionen [67] entstehen, welche schematisch in Bild 2.4 abgebildet sind. In Bild 2.4a ist der Effekt von Mitund *Gegenlicht* dargestellt. Durch die radiale Punktverzerrung (s. vorherigen Abschnitt) werden vertikale Linien am Rand des Bildes verkippt dargestellt. Je nach Sonnenstand und Sensorblickrichtung können Objekte von der Sonne belichtet sein oder aber auch im Schatten liegen. Daneben wird die Belichtung eines Objektes durch dessen Platzierung im Bild beeinflusst. Dadurch erscheint ein Baum an der Position A dunkler als ein Baum an Position B. Der Effekt hängt damit von dem Elevationswinkel der Sonne und der Objekthöhe und -platzierung ab, wobei der Effekt durch flache Elevationswinkel der Sonne verstärkt wird. Zusätzlich wird der Effekt von der Neigung des Geländes beeinflusst. Bild 2.4b zeigt den Effekt der Streuung. Die Rückstreuung der Partikel in der Atmosphäre wirkt sich als additive Beleuchtung aus, wodurch zusammen mit den Reflexionen an der Bodenoberfläche die Strahlungsintensität steigt. Damit wird vom Sensor mehr Rückstreuung durch D als durch C empfangen. In nebligen oder dunstigen Situationen wird dadurch eine unterschiedliche Belichtung über das Bild erzeugt. Weiterhin spielt der Effekt der spiegelnden Reflexion eine Rolle, welcher in Bild 2.4c dargestellt ist. Grund für diesen Effekt ist der Reflexionsgrad der Bodenoberfläche. Dabei resultiert der Effekt der spiegelnden Reflexion z. B. aus Wasserflächen. Je nach Position kommt es zu einer Über- bzw. Unterbelichtung. Im Beispiel wird die Wasserfläche an Position E heller wahrgenommen als die Wasserfläche an Position F.



Bild 2.4: Atmosphärische Effekte, welche verschiedene Beleuchtungsstärken in der Brennebene des Sensors erzeugen.

Der Effekt der spiegelnden Reflexion kann auch durch Niederschlag auf nassen Oberflächen auftreten. Kameras, die das sichtbare Spektrum aufnehmen, zeigen bei Regen eine zeitlich sowie örtlich hochfrequente Änderung der Intensität der Pixel. Daraus resultiert ein Rauschen, welches für Bildverarbeitungsalgorithmen, welche örtlich kleine Bildmerkmale (z. B. Regionen mit 2 × 2 Pixeln) auswerten, negative Auswirkung haben kann [68]. Außerdem werden bei zunehmender Regenstärke Objekte im Bild verdeckt. Dabei ist eine Minimierung dieser Effekte durch verschiedene Kameraparametereinstellungen [69], wie z. B. eine erhöhte Belichtungszeit oder eine verringerte Blendenzahl, sowie durch Methoden der Bildverarbeitung möglich. Daneben können Spiegelungen an nassen Oberflächen auftreten, wobei dieser Effekt auch bei LWIR-Sensoren zu beobachten ist. Außerdem führt regnerisches Wetter zu geringerem Kontrast [70] und beeinflusst die thermische Signatur von Objekten [71] durch die relative Temperaturänderung [72]. Daher können z. B. Personen sowohl wärmer als auch kälter als ihre Umgebung erscheinen.

Wie bereits angesprochen, führen dunstige oder neblige Situationen zu einer Streuung des Lichts, aber zusätzlich dazu wirken sich diese als *Dämpfung* der Strahlungsintensität aus. Nebel entsteht bei Saturierung der relativen Luftfeuchte in einem örtlich begrenzten Teil der bodennahen Atmosphäre. Es bilden sich sehr kleine und dicht gepackte Wassertropfen mit einem Radius von 1 bis $10 \,\mu m$ [73]. Im visuellen Spektrum führt die *Dämpfung* durch die Wasserpartikel zu einer verringerten Sättigung und die Streuung der Umgebungsbeleuchtung an den Wasserpartikeln zu einer Erhöhung der Beleuchtungsstärke [74]. Außerdem ist eine Unschärfe und Verdunklung der Objekte in der Szene festzustellen [75]. Je nach Situation ist es auch möglich, eine digitale Restauration des Bildes durchzuführen, um die Effekte von Nebel zu minimieren [76]. Aufgrund der größeren Welllänge des LWIR-Sensors ist die Streuung an den Nebelpartikeln geringer und kann damit leichten Nebel oder Dunst besser durchdringen als ein VIS-Sensor [75]. Allerdings kann es durch thermische Strahlung, z. B. von Lichtquellen, zu einer Überblendung durch den Nebel kommen, ein Effekt, der auch als *Blooming* bezeichnet wird [77].

Ein weiterer atmosphärischer Einfluss wird durch Wolken hervorgerufen, welche durch denselben physikalischen Effekt wie die Nebelbildung entsteht, sich allerdings in gewisser Distanz über dem Erdboden befinden. Je nachdem ob sich der Sensor bei der erdgerichteten Aufklärung ober- oder unterhalb der Wolkenebene befindet, kommt es zu unterschiedlichen Effekten. Befindet sich der Sensor bzw. das UAV oberhalb der Wolkenebene, kommt es zu einer partiellen oder sogar vollständigen Verdeckung der Erdoberfläche. Unterhalb der Wolkendecke führt eine aufgelockerte Bewölkung zu Umrissschatten auf dem Erdboden, der je nach Sonnenstand variiert. Des Weiteren entstehen im visuellen Spektrum bei wolkenfreien Situationen Schatten an vertikalen Objekten (z. B. Gebäuden). Objekte, die sich in diesen Schattenregionen befinden, haben durch die verringerte Helligkeit weniger ausgeprägte Kanten, wobei Methoden Korrektur Schattenregionen zur von durch Bildverarbeitungsroutinen [78] existieren. Außerdem können bei klarem Wetter durch Reflektion sog. Flares auftreten, die zu einer Übersättigung des Sensors führen. Bei steigendem Bewölkungsgrad wird zwar die globale Helligkeit durch die Dämpfung des Lichtes geringer, aber die Intensität der Schatten wird durch den Effekt der Streuung [79] reduziert oder verschwinden. Im langwelligen Infrarotspektrum führen klare Tage zu einem erhöhten Kontrast, da absorbierende Oberflächen eine thermische Signatur ausbilden. Dadurch werden im Bild auch mehr Strukturen sichtbar [35]. Außerdem sind LWIR-Sensoren robust gegenüber Blendlicht. Bei zunehmenden Bewölkungsgrad besitzen Wärmequellen, z. B. der Motor eines Fahrzeuges, tendenziell eine höhere Temperatur als die Umgebung, wodurch diese eine höhere Helligkeit aufweisen.

Um die genannten atmosphärischen Einflussfaktoren und deren Auswirkung auf die Sensordatenverarbeitung zu kompensieren, wurden verschiedenste Anstrengungen unternommen. Hierbei wird in [80] ein Verfahren zur automatischen Landung eines Luftfahrtzeuges vorgestellt, welches bei nebligen oder bewölkten Bedingungen die Horizontlinie visuell detektieren kann. Im Bereich der Fahrzeugautomatisierung stellte Radecki et al. [81] ein System zur Detektion und Verfolgung von Objekten vor, welches robust gegenüber Witterungsverhältnissen ist. Hier werden unterschiedliche Sensoren eingesetzt, um eine robuste Wahrnehmung der Umgebung unter wechselnden Wetterbedingungen zu realisieren. Verwendet wurde eine bildgebende Sensorik im visuellen Spektrum, ein LIDARund ein RADAR-Sensor. Im Bereich der unbemannten Luftfahrtzeuge wird in [82] ein unbemannter Modellhubschrauber mit bildgebenden Kameras im visuellen und langwelligen Infrarotspektrum verwendet, um Search & Rescue Missionen bei Tag und Nacht durchführen zu können. In [83] wurde zur Untersuchung von Wettereffekten und Beleuchtungssituationen ein Datensatz durch das Aufnehmen von Bildern einer Szene über fünf Monate hinweg erzeugt. Untersucht wurden verschiedene atmosphärische Effekte, wie Schattenwurf, Spiegelungen durch nasse Oberflächen, Beleuchtungswechsel und Verdeckungen entstehend durch dunstige bzw. neblige Situationen. Die aufgetretenen Effekte wurden zu Evaluierungszwecken exemplarisch analysiert und quantifiziert. Dabei werden weiterführend von Narasimhan und Nayar [73] Methoden vorgeschlagen, um durch Wettereffekte beeinflusste Bilddaten wiederherzustellen.

2.3.3 Bodenbedeckung und Geländeform

Die Bodenbedeckung beschreibt die physische Zusammensetzung (Materialien) der Erdoberfläche. Da diese Materialien unterschiedliche Reflexionsgrade in Abhängigkeit der Wellenlänge besitzen, zeigen VIS- und LWIR-Sensoren unterschiedliche Sensorbilder derselben Szene. Je nachdem, welche Bodenbedeckungen sich in der Szene befinden, kann dies Auswirkungen auf die Leistungscharakteristik der Sensordatenverarbeitung haben.

Die Auswirkungen der Bodenbedeckung auf die Sensordatenverarbeitung mit visuellen Bilddaten zeigt sich z. B. bei Verdeckungen von Objekten durch Bäume und Gebäude bei flachen Elevationswinkeln des Sensors. Außerdem nimmt die Zahl der Gradienten im Bild zu, sobald urbane Strukturen existieren. Diese können z. B. bei der zu Fehldetektionen aufgrund der rechteckigen Grundformen zu Problemen führen [84,85]. Homogene Flächen, wie Wasser und Ebenen, weisen wenig Struktur auf und können damit von Objekten (z. B. Fahrzeuge, Personen oder Gebäuden) leichter unterschieden werden. Im LWIR-Spektrum hängt die enthaltene Information in den Sensordaten nicht nur vom geographischen Szeneninhalt ab. Denn gerade in urbanen Gebieten kommt es durch Sonneneinstrahlung und anderen thermischen Quellen zu hochfrequenten Signalen in den Sensordaten. Dabei werden Wasserflächen unabhängig vom Tageszyklus homogen wahrgenommen, da Wasser ein absorbierendes Verhalten aufweist.

In einigen Verfahren zur Sensordatenverarbeitung werden die Informationen über die Bodenbedeckung genutzt, um damit die Algorithmenleistung zu erhöhen. Dabei wird in [84] ein Algorithmus zur Fahrzeugerkennung in Luftbildaufnahmen vorgestellt, der eine Minimierung der Falschdetektionsrate durch Wissen um die asphaltierten Flächen in einer Szene nutzt. Verwendet wird dazu ein Geoinformationssystem (GIS), welches die Positionen der Straßen und asphaltierten Flächen enthält. Ein allgemeines Paradigma zur Sensordatenanalyse auf Luftbildern, GEOBIA, unterstützt durch Geoinformationssysteme, wird von Blaschke et al. [86] vorgeschlagen. Dabei werden digitale Geländehöhen- und Bodenbedeckungsdaten zur Unterstützung von Klassifikationsaufgaben verwendet. Der GEOBIA-Ansatz wird hauptsächlich zur Segmentierung von Luftbildaufnahmen eingesetzt. Des Weiteren werden Systeme präsentiert, die mithilfe von Sensordaten eine Kategorisierung vornehmen [87]. Hierbei wird eine Segmentierung auf Sensordaten einer visuellen Kamera vorgestellt, die zunächst das Bild in die Kategorien Vegetation, Straßen, Gebäude und Wasser einteilt. Anschließend wird auf Basis der segmentierten Kategorien eine Plausibilitätsprüfung [88] für erkannte Objekte, wie z.B. Schiffe, Fahrräder, Personen oder Motorräder, durchgeführt. Bei der Objektdetektion werden weiterhin spezielle Bildmerkmale verwendet, welche den lokalen Kontext um die Objekte erfassen können [89]. Damit lässt sich feststellen welcher Hintergrund (z. B. Straße, Wiese oder Himmel) das Objekt umgibt, um mit dieser zusätzlichen Information Falschdetektionen zu minimieren.

2.3.4 Diskussion

In Tabelle 2.4 sind die in Abschnitt 2.3.1 bis 2.3.3 vorgestellten Anwendungen gelistet, welche Einflussfaktoren auf eine Sensordatenverarbeitung berücksichtigen. Dabei ist zunächst zu erkennen, dass lediglich einzelne Einflussfaktoren untersucht wurden. Die gelisteten Publikationen, welche photogrammetrische Einflüsse in Betracht ziehen, nutzen vor allem den Sensorelevationswinkel und Parameter, um die Bodenpixelauflösung (engl. ground sample distance, GSD) zu bestimmen. Dabei konnte gezeigt werden, dass die Algorithmen von diesen abhängig sind. Infolgedessen sollten diese bei der Beschreibung des Umgebungszustandes hinzugezogen werden. Des Weiteren werden einzelne atmosphärische Einflüsse durch Einsatz komplementärer Sensoren minimiert oder durch spezielle Sensordatenverarbeitungsroutinen eliminiert. Dabei werden u. a. sowohl bildgebende Sensorik als auch LIDAR- und RADAR-Systeme eingesetzt. Hauptsächlich wurden Einflüsse durch Bewölkungsgrad, Niederschlagsmenge, Nebeldichte und Tageszeit untersucht. Dabei konnte gezeigt werden, dass diese eine Auswirkung auf das Leistungsverhalten der Sensordatenverarbeitung besitzen und sollten deshalb berücksichtig werden.
	Pub.	Anwendung	Betrachtete Einflussfaktoren	Randbedingungen	
л.	[22]	Objekterkennung	Sensorelevationswinkel	Nur statische Szene	
gramı	[65]	Personenerkennung	Sensorposition und -lage	Schattenwurf der Personen wird zur Erkennung benötigt	
photo	[66]	Fahrzeugerkennung mit geometrisch invarianten Merkmalen		Für neue Szenen wird eine Anpassung der Algorith- menparameter empfohlen	
	[80]	Automatische Landung eines UAVs	Nebeldichte, Bewölkungsgrad	Ansatz lediglich anwendbar bei nebligen Situationen	
isch	[81]	Detektion und Verfolgung von Objekten	Nebeldichte, Bewölkungsgrad, Niederschlagsmenge (Regen und Schnee)	LIDAR, RADAR und bildgebende Sensorik werden benötigt	
sphäi	[82]	Search & Rescue	Tageszeit	Manuelle Operation mit VIS- und LWIR-Sensoren	
atmo	[83]	Datensatz zur Untersuchung von Witterungen	Nebeldichte, Bewölkungsgrad, Niederschlagsmenge	Statische urbane Szene	
	[73]	Bildwieder-herstellung	Nebeldichte, Bewölkungsgrad, Niederschlagsmenge	Witterungssituation muss bekannt sein	
	[84]	Fahrzeugerkennung	Straßensegmente	Position und Lage der Staßensegmente muss gegeben sein	
opographisch	[86]	Segmentierung	Bodenbedeckungen und Geländeformen	GIS mit entsprechenden Daten müssen verfügbar sein	
	[87]	Segmentierung und Objekterkennung	Vegetation, Gebäude, Straßensegmente, Wasserflächen	Verwendbar auf visuellen Bilddaten unter Tages- lichtsituationen	
4	[89]	Objektdetektion mit kontexterfassenden Merkmalen		Verwendbar auf visuellen Bilddaten unter Tages- lichtsituationen	

Tabelle 2.4: Untersuchte Einflussfaktoren auf Bildverarbeitungsmethoden.

Abschließend ist festzustellen, dass sich einigen Veröffentlichungen mit topographischen Einflussfaktoren befassen. Hierzu zählen vornehmlich Bodenbedeckungen, wie z.B. Vegetationen, urbane Strukturen oder Wasserflächen, und Geländeformen. Die Bodenbedeckung und Geländeform werden dabei weniger auf ihren Einfluss auf die Sensordatenverarbeitung hin untersucht, sondern verwendet, um durch Plausibilitätsprüfung vertrauenswürdigere Ergebnisse aus den bildverarbeitenden Algorithmen zu erhalten.

2.4 Ansätze für den adaptiven Sensoreinsatz

Damit die in den vorherigen Abschnitten ausgewählten Sensoren und Bildverarbeitungsalgorithmen sowie die identifizierten Einflussfaktoren gesamtheitlich untersucht werden können, fordern die Forschungsziele aus Abschnitt 1.3 die Umsetzung dieser in ein prototypisches System zum adaptiven Sensoreinsatz. Dazu werden im Folgenden geeignete Systemansätze betrachtet.

2.4.1 Sensormanagement

Der Begriff Sensormanagement (SM, engl. *sensor management*) oder Sensor-Ressourcen-Management (engl. *sensor resource management*) beschreibt einen Ansatz, der sich mit der Theorie und Anwendung von dynamischer Ressourcenallokation für Systeme mit Sensorik und Sensordatenverarbeitung beschäftigt. Grundsätzlich wird dabei auf der Basis von vorherigen Aktionen und Systemzuständen eine Entscheidung für die nächste auszuführende Aktion getroffen. Hero et al. geben dazu in [90] und [23] einen ausführlichen Überblick über die Theorien und Konzepte eines Sensormanagements. Die gängigen Theorien basieren dabei auf POMDP (engl. *partially observable Markov decision process*) [91], MAB (engl. *multi-armed* *bandit*) [92] und informationsoptimierte Entscheidungsprozesse [93]. Das dazugehörige Konzept nach Hero et al. ist in Bild 2.5 dargestellt.



Bild 2.5: Blockdiagram des Sensormanagementkonzepts nach [23]. Basierend auf dem Ausgang der *Optimization* wählt der *Sensor Selector* einen Sensor (S1, S2 oder S3) aus. Dabei hat die *Optimization* das Ziel, die Systemleistung zu verbessern und verwendet dafür geeignete Metriken, wie zum Beispiel Abschätzungen über die Leistung der Algorithmen innerhalb der *Signal Processing* Einheit.

Zunächst lässt sich dieses in eine Datenverarbeitungsebene, welche die unteren Blöcke Sensor Selector, Sensor Fusion und Signal Processing umfasst, und eine Managementebene, welche die oberen Blöcke Predict System Performance und Optimization beinhaltet, einteilen. Der Ausgang der Datenverarbeitungsebene enthält zum einen Perzepte (engl. Percepts) und zum anderen umfasst er Metriken zur Beschreibung der aktuellen Systemleistung (performance metrics). Dabei sind die Perzepte aufgabenorientierte Ergebnisse. Im Falle einer Objekterkennung kann dies die Objektposition sein, bei einer Objektverfolgung ist dies die Objekttrajektorie. Solche Perzepte und Metriken nutzt die Managementebene, um eine Optimierung durchzuführen. Anschließend verwendet der Sensor Selector das Ergebnis der Optimierung, um für die nächste Messung den optimalen Sensor auswählen. Des Weiteren unterteilen die Autoren die gewählten Metriken in zwei Gruppen: Metriken, die den physikalischen Zustand, und Metriken die den Datenzustand beschreiben. Der physikalische Zustand beinhaltet externe Informationen, was in der betrachteten Anwendung dieser Arbeit die Flugposition und -lage oder die Sensorausrichtung sein könnte. Der Datenzustand umfasst systeminterne Informationen, wie etwa die aktuelle Qualität der Ergebnisse der Sensordatenverarbeitung (Signal Processing).

Die generellen Fragestellungen, die ein *Sensormanagement* behandelt, sowie die daraus abgeleiteten Anforderungen und Aufgaben werden auf konzeptueller Ebene in [94] und [95] vorgestellt. Unter anderem wird diskutiert, dass ein solches System robust in hoch dynamischen Umgebungen agieren und sich verändernde Sensorleistungen berücksichtigen muss. Hierbei wird zunächst davon ausgegangen, dass ein menschlicher Operateur wesentliche perzeptive Teilaufgaben, wie z. B. eine Objektidentifikation oder -klassifikation, durchführt. Dabei wird in [25] diskutiert, ob und wie diese Teilaufgaben durch entsprechende automatisierte Funktionen ersetzt werden können. Um solche Funktionen in ein Sensormanagement zu integrieren, wird letztendlich geschlussfolgert, dass auch die Leistung dieser ermittelt und berücksichtigt werden muss. Damit lässt sich neben einer Sensorauswahl durch ein Sensormanagement auch eine Selektion bzw. Parametrisierung der sensordatenverarbeitenden Algorithmen durchführen.

2.4.2 Ansätze zur Adaption der Sensordatenverarbeitung

Um sich der Herausforderung der Adaption der Sensordatenverarbeitung zu stellen, wird in [96] ein *Autonomes Computer Vision System* (ACVS) vorgeschlagen. Hierbei wird der Ansatz verfolgt, dass die Sensordatenverarbeitung aus mehreren Modulen besteht, um diese geeignet

verwalten zu können. Das ACVS verfolgt das Paradigma eines *selbst-adaptiven Systems*, welches im Allgemeinen vier Fähigkeiten haben soll:

- *Selbstüberwachung* bedeutet die Fähigkeit, den internen Zustand des eigenen Systems oder der Systemkomponenten zu beobachten, hinsichtlich Zuverlässigkeit, Genauigkeit, Schnelligkeit und Datendurchsatz.
- *Selbstregulierung* beschreibt die Fähigkeit, das eigene System oder eine Systemkomponente zu regulieren, indem interne Parameter adaptiert werden.
- Die Fähigkeit *Selbstreparatur* erlaubt eine Rekonfiguration des Systems oder einer Systemkomponente in Abhängigkeit von Umgebung oder externen Anforderungen.
- *Selbsterklärung* ist die Fähigkeit, eine Beschreibung des internen Zustands, das eigene System oder einer Systemkomponente bereitzustellen.

In der Veröffentlichung werden diese Fähigkeiten anhand einer exemplarischen Objektverfolgung konzeptuell veranschaulicht, wobei das in Bild 2.6 dargestellte Konzept des ACVS als Grundlage dient. Der *Component Supervisior* (CS) überwacht die Module und kann deren Parameter in Abhängigkeit der Modulzustände adaptieren oder komplette Module austauschen. Die Module können dabei Daten prozessieren oder Ereignisse (engl. *Events*) generieren, wie z. B. die initiale Erkennung eines Objektes.



Bild 2.6: Konzeptschaubild des ACVS.

Zur Adaption der Sensordatenverarbeitung im ACVS werden Parameter wie die Prozesszeit oder die Qualität der Ergebnisse jedes Moduls vom *Component Supervisor* verwendet, um damit die Leistung dieser zu überwachen. Dadurch wird die Fähigkeit der *Selbstüberwachung* umgesetzt. Der CS hat das Ziel die Leistung des Systems zu erhalten, indem er Modulparameter verändert oder diese austauscht. Hierdurch wird die Fähigkeit *Selbstregulierung* und der *Selbstreparatur* abgebildet. Um schließlich die Fähigkeit der *Selbsterklärung* bereitzustellen, wird gefordert, dass die Module eine semantische Beschreibung besitzen.

Auch [97] empfiehlt im Sinne einer situationsgerechten Adaption der Sensordatenverarbeitung, Module auszutauschen, da diese nur bestimmte Operationsbereiche besitzen. Um eine Optimierung durch Modulaustausch zu erreichen, sei es notwendig, die Algorithmenleistung zu bestimmen. Metriken zur Bestimmung der Modulleistung oder Einflussgrößen darauf, werden in der Quelle nicht näher beleuchtet. Weiterhin werden Architekturkonzepte in [98] und [99] vorgeschlagen, um Verarbeitungsmodule zu verwalten und in Abhängigkeit der Aufgabe zu Prozessketten zusammen zu schalten.

2.4.3 Sensor- und Perzeptionsmanagement

Während die vorherigen Ansätze entweder die Auswahl der Sensoren oder die Adaption der Sensordatenverarbeitung betrachteten, wird von Russ et al. [100] ein Ansatz vorgestellt, der sowohl das Management der Sensorik als auch der Sensordatenverarbeitung behandelt. Hierzu wird das Paradigma des *Sensor- und Perzeptionsmanagements* (SPM) eingeführt, welches den auftragsbasierte und selbstkritischen Sensoreinsatz propagiert. Hierzu werden die Ressourcen (z. B. Sensoren oder Bildverarbeitungsalgorithmen) eines solchen Systems mit Beschreibungen versehen, die eine Aussage über deren Fähigkeiten, Abhängigkeiten und Limitationen erlauben. Ein vorgeschlagenes Systemkonzept des SPM-Paradigmas [30,101] zeigt Bild 2.7 und wird im Folgenden als *Sensor und Perzeptionsmanagementsystem* (SPMS) referenziert. Dieses SPMS besteht aus zwei separierten Ebenen, dem *Perzeptionsmanagement* und *Sensormanagement*. Während dem Sensormanagement die Anbindung, Verwendung und Beschreibung sensorische Hardwareressourcen obliegt, behandelt das Perzeptionsmanagement den auftragsbasierten Einsatz der Sensordatenverarbeitung.



Bild 2.7: Systemkonzept eines Sensor- und Perzeptionsmanagements nach [30,101].

Im Folgenden wird die Funktionsweise des SPMS anhand eines übergeordneten Systems (Missionsmanagement) vorgestellt, welches Missionsziele, z. B. das Erkunden eines definierten Gebietes, erhält. Das Missionsmanagement erzeugt aus übergeordneten Missionszielen einen Missionsplan, und übermittelt Teile des Plans an die entsprechenden Subsysteme. Diese Subsysteme können z. B. ein GNC-System zur Flugführung oder ein SPMS sein. Damit ein Missionsmanagement in der Lage ist einen Missionsplan zu erzeugen, muss es die *Fähigkeiten* der Subsysteme kennen. Dazu wurde von Smirnov und Stütz [102] ein *Ressourcenmodell* entwickelt, um die *Fähigkeiten* eines SPMS zu ermitteln. Dieses beschreibt auf semantischer Ebene technische Komponenten eines UAVs, welche notwendig sind, um bestimmte Aufklärungsaufgaben durchzuführen. Die Ressourcen umfassen dabei Hard- und Softwarekomponenten, wie z. B. die Sensorik, Flugplattform, Datenlinks oder sensordatenverarbeitende Algorithmen. Um eine spezifische Perzeptionsaufgabe durchzuführen, werden durch Modellierung der Beziehungen zwischen den Ressourcen und Aufklärungsaufgaben zum einen geeignete UAV-Konfigurationen ermittelt und zum anderen verschiedene mögliche Ressourcenkombinationen aufgezeigt.

Beinhaltet der Missionsplan Perzeptionsaufträge (z. B. eine Fahrzeugerkennung entlang einer Straße) werden diese vom SPMS ausgeführt. Dabei werden zunächst die sensorischen Anforderungen, die sich aus den Vorgaben der Sensordatenverarbeitung ergeben, vom Perzeptionsmanagement an das Sensormanagement übertragen. Im einfachsten Falle ist dies die Auswahl eines bestimmten Sensortyps, die Anforderungen können aber auch bestimmte Sensorparameter- und Sensorlageeinstellungen umfassen. Um z. B. auf Ressourcenausfälle reagieren zu können, liefert das Sensormanagement kontinuierlich eine Aussage über die eigenen *Fähigkeiten* an das Perzeptionsmanagement. Dabei können zur Verbesserung der Sensoreinsatzergebnisse (*Perzepte*) auch die Veränderung der Flugposition und damit der Sensorposition herangezogen werden, welche durch *Plattformanfragen* an das Missionsmanagement übermittelt werden. Schließlich prozessiert die konfigurierte Sensordatenverarbeitung die *Sensordaten* und liefert die entsprechenden *Perzepte* (z. B. die Position von detektierten Fahrzeugen) an das übergeordnete System, um diese z. B. an den Operateur zu übermitteln.

Um letztendlich auf Restriktionen durch Umgebungszustände reagieren zu können, sieht das SPMS Umgebungsbeschränkungen vor. Dabei können diese sowohl Einflüsse auf die Sensorik als auch auf die Sensordatenverarbeitung haben. Um die Einflussfaktoren der Umgebung (siehe Abschnitt 2.3) auf letzteres zu berücksichtigen, sieht der Ansatz des SPMS einen Perzeptionsgraphen vor. Dabei ist es das Ziel, möglichst viele unterschiedliche Lösungen für eine perzeptive Aufgabe bereitzustellen, um daraus die geeignetste Sensordatenverarbeitung unter den aktuellen Umgebungsbedingungen zu identifizieren. Der Perzeptionsgraph (PG) bietet eine Möglichkeit verschiedene Lösungen abzubilden, indem er Perzeptionsmodule (PM) enthält, die logisch miteinander verbunden sind. Diese besitzen ähnlich Schnittstellen wie dies die Moduldefinition aus vorherigem Abschnitt 2.4.2 beschreibt. Veranschaulichen lässt sich dies in einem gerichteten Graphen, welcher exemplarisch in Bild 2.8 dargestellt ist. Ein Pfad durch diesen Graphen, zum Beispiel {*S*1, *M*2, *M*5, *M*8, *P*}, wird als *Perzeptionskette* (PK) bezeichnet. Zur graphischen Strukturierung der Perzeptionsmodule sind in der Abbildung Verarbeitungsebenen (E1 - En) eingezeichnet. Grundsätzlich kann ein Perzeptionsmodul auch mehrere Verarbeitungsebenen umfassen. Alle Module einer Verarbeitungsebene gehören dabei einer bestimmten Kategorie an. Im Falle von bildgebender Sensorik handelt es sich um Verarbeitungsroutinen der Bildverarbeitung, wie z. B. die Datenaufbereitung, Segmentierung oder Objektklassifikation.



Bild 2.8: Beispielhafte Darstellung eines Perzeptionsgraphen.

Um eine auftragsbasierte Führung eines UAV für Aufklärungszwecke realisieren zu können, ist in Bild 2.9 die Einbettung des SPMS in das konventionelle Führungskonzept aus Bild 1.2 dargestellt. Hierbei kann ein Operateur das UAV auf Auftragslevel führen, da der Sensoroperateur durch das SPMS substituiert wurde. Durch die Integration des SPMS ergibt sich außerdem eine Automatisierungsschleife. Dies bedeutet, dass ein solches UAV nicht nur vom Operateur geführt und unterstütz werden kann, sondern auch autark, nach bekannten Zielvorgaben, agieren kann.



Bild 2.9: Einbettung des SPMS in ein Flugführungskonzept mit resultierender Automatisierungsschleife.

2.4.4 Diskussion

Der Ansatz des *Sensormanagements* wurde in Abschnitt 2.4.1 vorgestellt. Dieser konzentriert sich auf die kontinuierliche Auswahl der Sensoren, deren Parametrisierung und Ausrichtung, um die Aufklärungsleistung zu optimieren. Hierbei wird allerdings keine Optimierung der Sensordatenverarbeitung betrachtet. Der darauffolgende Abschnitt 2.4.2 stellte daher Ansätze zur Adaption der Sensordatenverarbeitung vor, die im wesentlich auf einer Modularisierung dieser beruhen. Dabei werden vor allem Metriken der Hardwareressourcen (z. B. Prozesszeit oder Speicherausleistung) herangezogen, um eine Optimierung der Datenverarbeitung zu erreichen. Einflussfaktoren der Umgebung werden hierbei weniger behandelt. Um auch diese zu berücksichtigen und den Sensoreinsatz gesamtheitlich automatisieren zu können, wurde, wie in Abschnitt 2.4.3 beschrieben, das Paradigma des SPM vorgeschlagen. Dieser Ansatz sieht sowohl ein Management der Sensorik als auch der Sensordatenverarbeitung vor. Eine Gegenüberstellung der einzelnen Ansätze zeigt Tabelle 2.5. Hierbei ist zu erkennen, dass das SPMS am geeignetsten erscheint, da es alle Komponenten vorsieht, um eine adaptive luftgestützte Sensordatenverarbeitung zu realisieren.

	SM	ACVS	SPMS
Sensorauswahl	Х	_	Х
Sensorausrichtung	Х	_	Х
Adaption der Sensordatenverarbeitung	_	Х	х
Berücksichtigung von Umgebungszuständen	—	_	Х
Tabello 2 5, Cogonüberstellung der Ansätze	für den adaptive	on Soncoroincata	

Tabelle 2.5: Gegenüberstellung der Ansätze für den adaptiven Sensoreinsatz.

Der Ansatz des SPM sieht eine Modularisierung der unterliegenden Algorithmen in Perzeptionsmodule vor, damit die Sensordatenverarbeitung geeignet verwaltet und situativ adaptiert werden kann. Die sinnvolle Kombination dieser Module wird als gerichteter Graph dargestellt. Durch den Einsatz von redundanten Modulen mit der derselben Grundfunktion, z. B. eine Objektverfolgung, welche aber durch unterschiedliche Methoden realisiert werden, entstehen verschiedenartige Lösungsmöglichkeiten, welche durch das oben genannten Ressourcenmodell ermittelt werden können. Damit bietet das SPMS bis auf die Ermittlung der geeigneten Lösungsmöglichkeit im Perzeptionsgraphen alle essenziellen Komponenten für den adaptiven Sensoreinsatz. Infolgedessen bietet sich für diese Arbeit der SPMS-Ansatz an. betrachtet Verfahren Folgenden Abschnitt zur Algorithmenauswahl, um die Lösungsmöglichkeiten im Graph zu bewerten und schließlich für den Missionseinsatz auszuwählen.

2.5 Algorithmenauswahl für Bildverarbeitungsroutinen

Die grundlegende Veröffentlichung, die sich mit dem Problem der Algorithmenauswahl (engl. *algorithm selection*) allgemeiner Art widmet, wurde von John Rice [26] verfasst und stellte ein abstraktes Modell vor. Dieses wurde in modifizierter Form zur maschinellen Algorithmenauswahl (siehe Bild 2.10) von Smith-Miles [103] vorgestellt und enthält vier essentielle Komponenten:

- *Problemraum P*, der eine Menge von Instanzen *x* der Problemklasse enthält.
- *Merkmalsraum F*, der messbare Charakteristiken der Instanzen enthält, die durch einen Merkmalsextraktionsprozess aus *P* erzeugt werden.
- *Menge an Algorithmen A*, welche alle betrachteten Algorithmen zur Problemlösung umfasst.
- *Leistungsraum Y*, der eine Zuordnung für jeden Algorithmus zu einem Satz von Leistungsmetriken beinhaltet.

Formal lässt sich damit das Problem der Algorithmenauswahl folgendermaßen ausdrücken: Für eine gegebene Probleminstanz $x \in P$, mit dem Merkmal $f(x) \in F$, wird die Zuordnung S(f(x)) gesucht, die in der Menge der Algorithmen A den Algorithmus α_i auswählt, so dass die Leistung $y(\alpha_i(x)) \in Y$ maximiert wird.



Bild 2.10: Schematisches Diagramm des Modells zur Algorithmenauswahl.

Übertragen in den Kontext dieser Arbeit, entspricht der Problemraum der durch die bildgebende Sensorik erfassten Bilder, der Merkmalsraum den Einflussfaktoren der Umgebung und die Menge der Algorithmen den bildverarbeitenden Algorithmen zur Fahrzeugerkennung. Um diese Algorithmen geeignet auswählen zu können ist ein Verfahren zur Realisierung der Zuordnungsfunktion *S* notwendig, welche eine Bestimmung der Algorithmenleistung voraussetzt. Im Folgenden werden verschiedene Ansätze zur Algorithmenauswahl für Bildverarbeitungsroutinen vorgestellt und gegenübergestellt. In der Literatur finden sich dazu sowohl *wissensbasierte Verfahren* als auch Methoden des *maschinellen Lernens*. Beide Formen werden im Folgenden beschrieben.

2.5.1 Wissensbasierte Methoden

Ein Konzept zur Auswahl von sensordatenverarbeitenden Algorithmen, welches ausschließlich Expertenwissen nutzt, wird von Tenorth und Beetz [104] vorgestellt. Dabei werden Prozessabläufe mittels einer logischen Programmiersprache definiert, um die Aufgaben eines Haushaltroboters zu beschreiben. Die spezifischen Aktionen, die zur Umsetzung einer Aufgabe bzw. eines Aufgabenplanes nötig sind, resultieren aus der Umgebungssituation, welche mittels eines semantischen Umgebungsmodells beschrieben wird. Damit ist dieser Ansatz zunächst unabhängig von aufwendigeren lernenden Verfahren. In [29] wird ein Objekterkennungssystem für Überwachungsszenarien vorgestellt, welches sich an Tag- und Nachtsituationen adaptiert. Dabei wurden verschiedene Klassifikatoren auf unterschiedlichen Datensätzen trainiert, um eine Diversität der Objekterkennung zu erreichen. Das System sucht dabei im Anwendungsfall kontinuierlich nach dem besten Algorithmus auf Basis der Detektionsanzahl pro Bild. Es wird gezeigt, dass damit ein Wechsel der Algorithmen durchgeführt werden kann und die Genauigkeit gegenüber dem besten Algorithmus steigt. Nachteil dieser Methode ist, dass darauf vertraut wird, dass die Detektoren kaum Falschdetektionen produzieren.

Robertson et al. [105] stellen ein adaptives System zur Analyse von Luftbildern vor. Dazu wird eine Wissensdatenbank verwendet, um in Abhängigkeit der Umgebungsparameter die Qualität von Segmentierungsalgorithmen zu evaluieren. Hierbei wird jedoch nicht im Detail auf die Umgebungsparameter eingegangen. Ebenso wird nicht erläutert, wie Wissenszusammenhänge der Wissensdatenbank zugeführt werden. Um den Ansatz erweitern zu können, wird eine separate Modellierung der Einflüsse der Umgebungseffekte auf die Algorithmen vorgeschlagen. In einem zweiten Schritt wird daraus automatisiert eine holistische Wissensrepräsentation erzeugt.

Für den Luftfahrtbereich präsentieren Russ et al. [30] ein weiteres Verfahren zur Algorithmenauswahl, das mittels bayesscher Netze Detektions- und Verfolgungsalgorithmen auswählt. Dabei wurden die Netze durch Expertenwissen erzeugt und bestimmen die Leistung der sensordatenverarbeitenden Algorithmen in Abhängigkeit von der verfügbaren Sensorik, dem Umgebungszustand (Tageszeit, Wetter, Temperatur und Oberflächenmaterial der zu betrachtenden Szene) und den Systemanforderungen (Speicherbedarf und Rechenzeit). Verwendet wurden hierzu bildbasierte Sensoren im visuellen und langwelligen Infrarotspektrum sowie ein LIDAR-Sensor. In einem Flugexperiment konnte u. a. gezeigt werden, dass bei hohen Umgebungstemperaturen eine Personenerkennung auf asphaltierten Flächen mittels LWIR-Sensordaten einsetzbar ist, aber auf Wiesen besser VIS-Sensordaten zur Personenerkennung genutzt werden sollten.

Ein anderer wissensbasierter Ansatz wird von Hochgeschwender et al. [99] vorgestellt. Hier wird eine Methode beschrieben, die darauf abzielt, die Detektionsrate von Markern unter verschiedenen Beleuchtungssituationen zu optimieren. Dabei wurden verschiedene Interest-Operatoren verwendet, welche in einem Evaluierungsschritt auf ihre Leistung hin überprüft werden. Immer dann, wenn eine akzeptable Algorithmenleistung vorliegt, wird das zugehörige Bildhistogramm und die eingestellten Algorithmenparameter in einer Datenbank hinterlegt. Im Anwendungsfall wird die *Kullback-Leibler* Divergenz als Ähnlichkeitsmaß zwischen den abgespeicherten Histogrammen und dem aktuellen Bildhistogramm verwendet, um den optimalen Algorithmus mit geeigneten Parametereinstellungen finden.

2.5.2 Lernbasierte Methoden

Zur automatischen Farbanpassung von unterbelichteten Bildern wird in [27] eine Algorithmenauswahl vorgeschlagen, die die Auswahl mittels Fuzzy-Inferenzsystem (FIS) vornimmt. Dabei werden die Fuzzy-Regeln auf der Basis von Farb-, Textur- und Belichtungsmerkmalen in einem Lernschritt generiert. Zusätzlich wurden verschiedene Auswahlmethoden (k-nächste-Nachbarn-Algorithmus, k-Means-Algorithmus, lineare und quadratische Diskriminanzanalyse und Fuzzy-Inferenzsysteme) untersucht, wobei die FIS mit Mamdani-Typ und Larsen-Typ am besten bewertet wurden.

In [106] wird ein adaptives System zur Objektverfolgung vorgestellt. Dabei werden in Abhängigkeit von Objekt-Kontextmerkmalen (Größe, Verdeckung, Kontrast, Varianz des Kontrastes, Fläche und Varianz der Fläche) die Parameter des Trackers ausgewählt. Die Beziehung zwischen Kontextmerkmalen und den Trackerparametern wird mit einem Fuzzy-Inferenzsystem abgebildet, wobei die Fuzzy-Regeln über Trainingsdaten extrahiert wurden.

Des Weitern stellen Zhang et al. [107] eine Algorithmenauswahl für die Segmentierung von Bildern vor. Hierbei wird auf Basis von Bildmerkmalen (globale Farb- und Texturinformationen, Anzahl der Segmente, Mittelwert der Segmentfarbe, Textur- und Konturmerkmale der Segmente) die Algorithmenleistung eines jeden betrachteten Segmentierungsalgorithmus ermittelt. Die Leistungen werden dabei durch Entscheidungsbäume (engl. *Decision Trees,* DT) bestimmt, welche mittels Trainingsdaten generiert wurden. Dadurch kann für jedes neu zu segmentierendes Bild der geeignetste Algorithmus ermittelt werden.

Einen anderen Ansatz stellt *Meta learning* [103,108] dar, mit dem das Ziel verfolgt wird, einen relativen Leistungsindex auf der Basis von sogenannten Metamerkmalen für eine Gruppe von Algorithmen zu bestimmen. Diese Metamerkmale sind oft statistischer Art, da die Anwendung in der Vorhersage der Leistung eines Algorithmus auf einer bestimmten Referenzdatenbank liegt. Daher werden statistische Variablen der Datenbank, wie z. B. die Anzahl an Instanzen oder Datenkategorien, für die Vorhersagen verwendet. Dafür wurde zuvor mittels eines Trainingsalgorithmus, der *Meta-Learner*, ein Modell abgeleitet, welches als Eingang die Metamerkmale bekommt und daraus die betrachtete Gruppe von Algorithmen nach ihrer Leistung sortiert. In diesem Kontext wurde in [109] ein künstliches neuronales Netz (KNN) verwendet, um die Leistung der Algorithmen zu berechnen. Neben statistischen Merkmalen werden auch Algorithmenparameter als Metamerkmale verwendet. Auf diese Weise konnte zwar nachgewiesen werden, dass eine sinnvolle Algorithmenauswahl möglich ist, jedoch wurde kein Vergleich mit anderen Verfahren gezogen.

2.5.3 Diskussion

In der Literatur werden beide Ansätze, wissensbasierte und solche aus dem Bereich des maschinellen Lernens (ML), zur Algorithmenauswahl erwähnt. Im Allgemeinen können wissensbasierte Methoden (WBM) mit Expertenwissen umgesetzt und damit auch auf Erfahrungswissen zurückgegriffen werden. Dazu müssen die Einflussfaktoren auf die Bildverarbeitungsroutinen auf einer semantischen Ebene beschrieben werden können, wie z. B. "Wenn der Bewölkungsgrad gering ist, dann ist die Leistung des Algorithmus hoch". Durch dieses Verfahren lassen sich fehlende Informationen zur Umgebungsbeschreibungen durch entsprechende Experten (z. B. ein Algorithmenentwickler) abfragen oder anhand von Erfahrungswissen des Operateurs erfassen. Wenn kein Expertenwissen vorliegt, aber entsprechende Datenbanken zur Ermittlung der Algorithmenleistung in Abhängigkeit des Umgebungszustandes verfügbar oder erzeugbar sind, können Methoden des maschinellen Lernens verwendet werden. Beide Verfahren sind grundsätzlich für die betrachtete Anwendung der luftgestützten Fahrzeugerkennung auf bildgebender Sensorik anwendbar.

Für beide Verfahren, WBM und ML, existieren verschiedene Methoden, welche anhand der vorherigen Literaturanalyse in Abschnitt 2.5.1 und 2.5.2, in Tabelle 2.6 mit ihren Vor- und Nachteilen aufgelistet sind. Für die wissensbasierten Methoden bieten sich Fuzzy-Inferenzsysteme an, da der Mamdani-Typ [110] besonders gut geeignet ist, Expertenwissen in Form von Regeln in das FIS-Modell zu übertragen [111]. Des Weiteren sind im Gegensatz zu

Heuristiken und Wissensdatenbanken nichtlineare Probleme abbildbar. Gegenüber bayesschen Netzen entfällt die problematische Ermittlung von Wahrscheinlichkeiten. Daher bieten sich für die Expertenmodellierung der Leistungscharakteristik der Algorithmen Fuzzy-Inferenzsystem an.

	Publikation	Methode	Vorteile	Nachteile		
	[99]	Heuristik	Anwendbar bei fehlenden Informationen	Systemfehler möglichExpertenwissen nötig		
	[29]	Heuristik	Kann aus menschlicher Erfahrung abgeleitet werden	Versuch und IrrtumFür nichtlineare Fälle ungeeignet		
WBM	[105]	WB	 Keine Trainingsdaten nötig Erklärbar Effiziente Modellierung bei linearen Abhängigkeiten 	 Verhalten muss bekannt sein Nichtlineare Probleme können unüberschaubar werden 		
	[30]	BN	 Interpretierbare Darstellung Indikation kausaler Effekte Modellierung indirekter Zusammenhänge 	 Problematische Ermittlung der Wahrscheinlichkeiten Zyklische Zusammenhänge können nicht erfasst werden Ungeeignet für hochdimensionale Probleme 		
BM	[27]	FIS	Lern- und modellierbarModelle sind erklärbar	Ungenauer als DT und KNNFür hochdimensionale Probleme		
, & WI	[106]	FIS	WissensextraktionSimple Modellierung durch Regeln	ungeeignet		
IM	[112]	FIS	Geeignet für lineare und nichtlineare Probleme			
	[107]	DT	 Interpretierbar Geeignet für lineare und nichtlineare Probleme Wenige Hyperparameter 	 Anfällig für Überanpassung Diagonale Entscheidungsgrenzen können problematisch sein 		
ML	[109]	KNN	 Für nichtlineare Probleme geeigneter als FIS und DT Kann für hochdimensionale Ein- gangsvektoren verwendet werden Geeignet für sehr große Daten- mengen 	 Black Box, nicht interpretierbar Schwerer zu trainieren als FIS und DT Viele Hyperparameter 		

Tabelle 2.6: Vor- und Nachteile der vorgestellten Methoden zur Algorithmenauswahl

Für eine nichtlineare bzw. unbekannte Abhängigkeit der Algorithmenleistung von den Einflussfaktoren der Umgebung sind aus dem Bereich des maschinellen Lernens vor allem künstliche neuronale Netze und Entscheidungsbäume geeignet. Entscheidungsbäume sind gegenüber neuronalen Netzen leichter zu interpretieren und zeigen beim Lernverfahren weniger zu wählende Hyperparameter. Dafür sind KNN besser in der Lage nichtlineare Probleme abzubilden, vor allem dann, wenn relativ viele Trainingsdaten vorliegen. Da die Verfügbarkeit der Trainingsdaten noch nicht bekannt ist, wird eine abschließende Methodenauswahl für ML-Verfahren zu einem späteren Zeitpunkt (siehe Abschnitt 3.3.3) behandelt.

Des Weiteren weisen die vorgestellten Veröffentlichungen verschiedenste Merkmale auf, anhand derer die Algorithmenauswahl vorgenommen wird. Es werden dabei u. a. Merkmale verwendet, die direkt aus den Sensordaten extrahiert werden. Dies haben allerdings den Nachteil, dass sie nicht durch einen Experten erfasst werden können. Daneben werden Merkmale verwendet, die Einflussgrößen der Umgebung enthalten, wie bereits in Abschnitt 2.3 behandelt. Dabei wird zwischen systemisch ermittelten Informationen, wie z. B. die Tageszeit, und hinterlegten Informationen, z. B. Wetterdaten, unterschieden. Da in dieser Arbeit sowohl ML-Verfahren als auch der Ansatz von wissensbasierten Verfahren geeignet erscheinen, bieten sich letztere Merkmale an, dass diese auch von einem Experten bzw. UAV-Operateur beschrieben werden können. Deshalb soll der Umgebungszustand, wenn möglich, bei fehlenden Informationen auch durch Erfahrungswissen ergänzt werden können.

KAPITEL 3 KONZEPT UND METHODENAUSWAHL

In diesem Kapitel wird das grundlegende Konzept zum adaptiven Einsatz von luftgestützter Sensordatenverarbeitung hergeleitete. Zunächst werden dazu aus dem Stand der Forschung und aus systemischen Gesichtspunkten die Anforderungen an das Konzept definiert. Anschließend erfolgen die Konzeptentwicklung und eine Analyse und Auswahl der zugehörigen Methoden. Abschließend wird das vorgestellte Konzept durch die Ergebnisse der Methodenanalyse detailliert und die Evaluierungsstrategie vorgestellt.

3.1 Anforderungen

Anhand des vorangegangenen Abschnittes zum Stand der Technik (siehe Kapitel 2), stellt dieser Abschnitt die Anforderungen an die Algorithmenauswahl auf. Hierbei wird zwischen systemischen und funktionalen Anforderungen unterschieden. Tabelle 2.1 zeigt diese in einer Zusammenfassung. Sie werden im Folgenden näher beschrieben.

	Nr.	Anforderung: Die Algorithmenauswahl soll	Aus Abschnitt
	1	bildgebenden Sensoren im VIS- und LWIR-Spektrum einsetzen.	2.1
ch	2	eine Integration der Sensoren in einen Gimbal zur Sensorausrichtung vorsehen.	2.1
mis	3	Bildverarbeitungsverfahren zur luftgestützten Fahrzeugerkennung nutzen.	2.2
Syste	4	eine Modularisierung der Bildverarbeitungsroutinen zur Leistungsbewertung verwenden.	2.4
	5	in das Systemkonzept des SPMS integriert werden.	2.4
÷	6	Einflussfaktoren der Umwelt zur Leistungsbestimmung verwenden.	2.3
unkt	7	die Algorithmenleistung durch WBM und ML bestimmen.	2.5
Ē	8	die Einflussfaktoren durch Experten- bzw. Erfahrungswissen erfassen.	2.5

Tabelle 3.1: Systemische und funktionale Anforderungen an die Algorithmenauswahl. Die letzte Spalte in der Tabelle gibt den relevanten Abschnitt aus dem Stand der Technik an.

3.1.1 Systemische Anforderungen

Um die Algorithmenauswahl letztlich in ein ablauffähiges Gesamtsystem zu integrieren und zu evaluieren, werden die folgenden systemischen Anforderungen aufgestellt.

Anforderung 1: Einsatz von bildgebenden Sensoren im VIS- und LWIR-Spektrum

In Abschnitt 1.3 wurde in den Forschungszielen festgelegt, dass als Applikation eine luftgestützte Fahrzeugerkennung betrachtet werden soll. Um hierfür geeignete Missionssensoren auszuwählen, wurden in Abschnitt 2.1 gängige Sensortypen untersucht. Hierbei konnte festgestellt werden, dass sich bildgebende Sensoren im visuellen und langwelligen Infrarotspektrum besonders gut eignen. Die Gründe hierfür sind vor allem die passende Orts- und Zeitauflösung dieser Sensoren. Deshalb sollen VIS- und LWIR-Sensoren verwendet werden.

Anforderung 2: Integration der Sensoren in einen Gimbal zur Sensorausrichtung

Zur Stabilisierung der Sensorik und Kompensation der Eigenbewegung eines UAVs können kardanische Aufhängungen eingesetzt werden. Für VIS- und LWIR-Sensoren existieren dazu sogenannte Gimbal²-Systeme. Des Weiteren bietet ein Gimbal den Vorteil, den Betrachtungswinkel auf eine Szene zu variieren, um damit bei der Evaluierung der Algorithmenauswahl diesen zu berücksichtigen. Daher sollen die VIS- und LWIR-Sensoren in einen Gimbal integriert werden.

Anforderung 3: Nutzung von Bildverarbeitungsverfahren zur luftgestützten Fahrzeugerkennung

Durch die Auswahl der bildgebenden Sensoren werden zur luftgestützten Fahrzeugerkennung Bildverarbeitungsverfahren eingesetzt. Dazu wurden in Abschnitt 2.2 sechs verschiedene Objektdetektoren (TM, BLOB, CC, DPM, DNN und CNN) als potenziell geeignet identifiziert. Diese sind robust gegenüber perspektivischen Variationen sowie invariant gegenüber Rotationen, Translationen und Skalierungen sind. Dabei verwenden vier von diesen (CC, DPM, DNN und CNN) Methoden des maschinellen Lernens und benötigen daher geeignete Trainingsdaten. Daher soll untersucht werden welche Trainingsdaten verwendet werden können, um schließlich eine Auswahl treffen zu können. Des Weiteren sollen das Template Matching und der BLOB-Detektor eingesetzt werden.

Anforderung 4: Modularisierung der Bildverarbeitungsroutinen zur Leistungsbewertung

Zur Erweiterbarkeit und Generalisierbarkeit der Algorithmenauswahl wird eine Modularisierung der Bildverarbeitungsroutinen gefordert (siehe Abschnitt 2.4.2). Dadurch können diese passend parametrisiert oder ggf. ausgetauscht werden. Aus Gründen der Erweiterbarkeit soll die Leistungsbewertung der Sensordatenverarbeitung ebenfalls auf Moduleben durchgeführt werden. Auf Basis dessen kann schließlich eine Algorithmenauswahl zur Maximierung der Gesamtleistung herangezogen werden.

Anforderung 5: Integration der Algorithmenauswahl in das Systemkonzept des SPMS

Abschnitt 2.4 stellte verschiedene Managementansätze zum adaptiven Sensoreinsatz vor. Hierbei konnte festgestellt werden, dass das Systemkonzept des SPMS eine Algorithmenauswahl der Sensordatenverarbeitung vorsieht und durch ein bereits existierendes Ressourcenmodell vorbereitet ist. Außerdem verfügt das SPMS über alle essenziellen Komponenten, um einen adaptiven Sensoreinsatz zu realisieren. Daher soll sich diese Arbeit in den Ansatz den SPMS-Ansatz eingliedern und die erwähnte Funktionslücke zur Ermittlung der geeignetsten Sensordatenverarbeitung unter verschiedenen Einflussfaktoren schließen.

² Ein Gimbal ist eine kardanische Aufhängung, in welche Sensoren montiert werden können. Solche Gimbal-Systeme bieten eine Sensorstabilisierung und -ausrichtung. Typischerweise werden dort bildgebende Sensoren im visuellen und langwelligen Infrarotspektrum sowie Sensoren zur Entfernungsmessung eingebaut.

3.1.2 Funktionale Anforderungen

Dieser Abschnitt stellt die funktionalen Anforderungen an die Algorithmenauswahl auf.

Anforderung 6: Verwendung von Einflussfaktoren der Umwelt zur Leistungsbestimmung

In Abschnitt 2.3 wurden verschiedene Einflussfaktoren auf die Sensordatenverarbeitung identifiziert. Hierbei konnten diese in drei Kategorien gegliedert werden: photogrammetrische, atmosphärische und topographische Einflussfaktoren. In dieser Arbeit sollen diese verwendet werden, um die Leistungsbestimmung der Bildverarbeitungsroutinen durchzuführen.

Anforderung 7: Bestimmung der Algorithmenleistung durch WBM und ML

Abschnitt 2.5 stellte verschiedene Verfahren vor, um die Bestimmung der Algorithmenleistung durchzuführen. Hierbei werden sowohl wissensbasierte Methoden als auch Verfahren aus dem Bereich des maschinellen Lernens eingesetzt. Beide wurden als geeignet befunden und sollen daher für die Algorithmenauswahl verwendet und untersucht werden. Als wissensbasierte Methode konnten Fuzzy-Inferenzsysteme identifiziert werden, da diese eine intuitive Integration von Expertenwissen erlauben, und deshalb auch eingesetzt werden sollen. Für ML-Verfahren konnten verschiedene geeignete Methoden (z. B. Entscheidungsbäume oder künstliche neuronale Netze) ermittelt werden. Eine abschließende Auswahl soll getroffen werden, wenn bekannt ist, welche Randbedingungen beim Erzeugen der entsprechenden Modelle gegeben sind.

Anforderung 8: Erfassung der Einflussfaktoren durch Experten- bzw. Erfahrungswissen

Wie bereits in Abschnitt 2.5.3 erläutert, sollen die Einflussfaktoren über einen Experten oder durch Erfahrungswissen erfasst werden können. Zum einen unterstützt dies die vorherige Anforderung 7, um wissensbasierte Verfahren zu realisieren, indem Expertenwissen genutzt wird, um die Abhängigkeit zwischen Einflussfaktoren und Algorithmenleistung herzustellen. Zum anderen sollen Einflussfaktoren währen des UAV-Einsatzes durch Erfahrungswissen (z. B. durch den Operateur) justiert werden können, wenn diese systemisch nicht ermittelt werden können, wie z. B. fehlende Informationen bzgl. der Wetterlage im Missionsgebiet.

3.2 Konzept zum adaptiven luftgestützten Sensoreinsatz

Dieser Abschnitt beschreibt das entwickelte Konzept zum adaptiven luftgestützten Sensoreinsatz. Dabei wird zunächst auf die Erzeugung der Leistungsmodelle eingegangen, welche den Bezug zwischen Umgebungszustand und Algorithmenleistung für jeden Bildverarbeitungsalgorithmus herstellen soll. Anschließend wird das Vorgehen zur eigentlichen Algorithmenauswahl unter Verwendung dieser Leistungsmodelle vorgestellt.

3.2.1 Leistungsmodellierung

Grundlage des hier entwickelten Konzeptes zur Leistungsmodellerzeugung bilden die Arbeiten von Rice [26] und Smith-Miles [103], welche in Abschnitt 2.5 vorgestellt und auf das Forschungsziel dieser Arbeit (siehe Abschnitt 1.3) übertragen wurde. Daraus lässt sich ein verallgemeinertes Konzept zur Erzeugung der Leistungsmodelle ableiten, wie in Bild 3.1 dargestellt.



Bild 3.1: Verallgemeinertes Konzept der Erzeugung der Leistungsmodelle.

Hierbei werden die *Leistungsindizes* der einzelnen Perzeptionsmodule durch Verarbeitung der Beispiele x der Sensordaten und vergleich mit der *Ground Truth*³ ermittelt. Für jeden Algorithmus ergibt sich damit der Leistungsraum P, der zu jedem Beispiel x einen Leistungsindex p beinhaltet. Zusätzlich dazu wird der Umgebungszustand f(x) durch Merkmalsextraktion aus den Sensordaten gewonnen. Daraus resultieren die *Metadaten* {F, P}, die ein Mapping zwischen dem Leistungsraum und Umgebungszustandsraum F bilden. Aus den Metadaten lässt sich ein Leistungsmodell M pro Perzeptionsmodul erzeugen, das aus dem Umgebungszustand den entsprechenden Leistungsindex bestimmen kann, wodurch *Anforderung* 4 (Modularisierung, siehe Abschnitt 3.1.1) erfüllt wird. Diese Leistungsmodelle werden letztendlich zur Auswahl der Perzeptionsmodule genutzt, um die potentiell leistungsfähigste Perzeptionskette im Perzeptionsgraphen (siehe Abschnitt 2.4.3) zu identifizieren, wie im folgenden Abschnitt 3.2.2 beschrieben, und im Folgenden als *Perzeptionskettenauswahl* bezeichnet wird.

In Anforderung 3 (siehe Abschnitt 3.1.1) werden die zu verwendenden Objektdetektoren zur luftgestützten Fahrzeugerkennung genannt. Diese benötigen zum Teil Trainingsdaten, um einen Klassifikator zu trainieren. Der Datensatz, der die Trainingsdaten enthält, wird im Folgenden als *Objektdatensatz* bezeichnet. Des Weiteren wird in Anforderung 7 (siehe Abschnitt 3.1.2) definiert, dass die Erzeugung der Leistungsmodelle nicht nur durch Verfahren des maschinellen Lernens aus den Metadaten erfolgt, sondern auch durch Expertenmodellierung. Zur Unterstützung der Leistungsmodellerzeugung durch WBM, sieht Anforderung 8 (siehe Abschnitt 3.1.2) die Erhebung des Umgebungszustandes durch Expertenoder Erfahrungswissen vor. Daher soll dieser nicht aus den Sensordaten extrahiert werden, sondern muss zusätzlich, in geeigneter Weise, erfasst werden können. Aus diesen Randbedingungen ergibt sich nunmehr das spezifische Konzept zur Erzeugung der Leistungsmodelle (siehe Bild 3.2), wie sie in dieser Arbeit verwendet wird.

In Abgrenzung zum *Objektdatensatz* wird hier der Begriff des *Modelldatensatzes* eingeführt, welcher X und F enthält, um die Leistungsmodelle M^{ML} zu erzeugen. Die Leistungsmodelle M^{WBM} bezeichnen jene, die durch Expertenwissen modelliert werden.

³ Im Falle einer Fahrzeugerkennung ist die *Ground Truth* die tatsächliche Position der Fahrzeuge im Sensorbild.



Bild 3.2: Spezifisches Konzept zur Erzeugung der Leistungsmodelle.

3.2.2 Algorithmenauswahl durch Anwendung der Leistungsmodelle

Bei der *Algorithmenauswahl* werden durch Anwendung der Leistungsmodelle M_i die Leistungsindizes p_i der Perzeptionsmodule PM_i in einem definierten Perzeptionsgraphen G ermittelt. Dazu wird kontinuierlich der Umgebungszustand f_x erfasst (Umgebungserfassung) und den Leistungsmodellen präsentiert. Die *Algorithmenauswahl* definiert, analog nach Rice [26], die Zuordnungsfunktion *S*:

$$c_j = S(G, \mathcal{M}, f_x) \tag{3.1}$$

Hierbei bezeichnet \mathcal{M} die Menge der Leistungsmodelle, die zu den Perzeptionsmodulen in G gehören. Außerdem ist c_j die Perzeptionskette im Perzeptionsgraphen mit der höchsten Gesamtleistung bei einem gegebenen Umgebungszustand f_x . Bestimmt wird c_j durch Ermittlung der Leistungen aller Perzeptionsketten im Graphen mit

$$p_j = \frac{1}{|c_j|} \sum_{\forall a_i \in c_j} p_i.$$
(3.2)

Dabei ist $|c_j|$ die Anzahl der Perzeptionsmodule a_i der Perzeptionskette $c_j \in G$. Die Perzeptionskette mit dem höchsten aufsummierten Leistungsindex wird schließlich ausgewählt (Selektion). Bild 3.3 zeigt ein Konzeptschaubild der Algorithmenauswahl. Hierbei ist anzumerken, dass die Selektion nur bei einem potentiellen Perzeptionskettenwechsel, die neue leistungsstärkste Perzeptionskette an die Sensordatenverarbeitung weitergibt.



Bild 3.3: Konzept der Algorithmenauswahl.

In Bild 3.4 ist ein Beispiel eines Perzeptionsgraphen mit zugehörigen Leistungsmodellen dargestellt. Dabei sind fiktive Leistungsindizes angegeben und der Pfad eingezeichnet, der den maximalen aufsummierten Leistungsindex besitzt.



Bild 3.4: Beispiel einer Ermittlung der geeignetsten Perzeptionskette im Perzeptionsgraph. Jedes Perzeptionsmodul PM besitzt ein Leistungsmodell M_i , welches aus dem Umgebungszustand f_x den Leistungsindex (blaue Werte) berechnet. Anschließend kann die Perzeptionskette mit der höchsten Gesamtleistung ermittelt werden, welche im Beispiel der blau markierte Pfad darstellt.

3.3 Methodenauswahl und Festlegungen

Dieser Abschnitt untersucht und wählt Methoden aus, die für das Konzept zur Erzeugung der Leistungsmodelle eingesetzt werden können. Zunächst werden Verfahren zur Sensordatenund Umgebungszustandserhebung, um den Modell- und Objektdatensatz zu generieren behandelt. Auf Basis des gewählten Datengenerierungsverfahrens werden die bereits gewählten Bildverarbeitungsmethoden zur Fahrzeugerkennung mittels ML bzgl. Ihrer Eignung überprüft. Anschließend werden geeignete Umgebungszustände identifiziert und ihre Auswirkungen auf die Sensordaten aufgezeigt. Abschließend werden Methoden zur Leistungsmodellerzeugung mittels Funktionsapproximation bewertet und passende ausgewählt.

3.3.1 Vorgehen zur Sensordaten- und Umgebungszustandserhebung

Für die Erzeugung der Leistungsmodelle ist ein *Modelldatensatz* notwendig, der Sensordaten $x \in X$, die das Spektrum der Umgebungszustände möglichst breit abdecken, entsprechende Annotationen und den Umgebungszustand f_x selbst enthält. Annotationen sind in diesem Zusammenhang Informationen zu den Bildinhalten, z. B. Objektpositionen (*Ground Truth*).

In der Literatur finden sich verschiedene vorgefertigte Datensätze mit Luftbildaufnahmen zum Zwecke der Fahrzeugerkennung, die zunächst auf ihre Verwendbarkeit für die vorliegende Problemstellung überprüft wurden. Tabelle 3.2 gibt einen Überblick über die Datensätze und deren annotierte Umgebungszustände. Bild 3.5 zeigt zu jedem vorgestellten Datensatz ein Beispielbild.

Namo	Dub	Annotierte Umgebungszustände			
Name	Pub.	photogrammetrisch	atmosphärisch	Topographisch	
VIRAT	[113]	Sensorelevationswinkel	Szenenbelichtung	Nur urbane Szenerien	
ORIDS	[114]	Nur orthogonale Aufnahmen	Keine Annotationen	Keine Annotationen	
VIVID	[115]	Keine Annotationen	Keine Annotationen	Keine Annotationen	
TAS aerial car	[116]	Nur orthogonale Aufnahmen	Keine Annotationen	Nur urbane Szenerien	
detection					
dataset					
VEDAI	[117]	Keine Annotationen	Keine Annotationen	Keine Annotationen	
Mini-Drone	[118]	Keine Annotationen	Keine Annotationen	Keine Annotationen	

Tabelle 3.2: Annotierte Kontextinformationen in verfügbaren Datensätzen.

In den gelisteten Datensätzen überwiegen orthogonale Luftbildaufnahmen, wodurch die Variabilität hinsichtlich perspektivischer Aufnahmen bzw. photogrammetrischer

Umgebungszustände limitiert ist. Ebenfalls beinhalten die Datensätze zwar unterschiedliche topographische Gegebenheiten in den Bildaufnahmen, welche aber nicht annotiert sind. Atmosphärische Veränderungen werden in keinem Datensatz berücksichtigt, bis auf die Schwankung der Szenenbelichtung.

Aus den vorhergehenden Beurteilungen lässt sich feststellen, dass die bestehenden Datensätze nicht ausreichen, um eine Untersuchung bzgl. der Einflüsse der Umgebungszustände auf die Sensordatenverarbeitung durchzuführen. Möglich wäre die Durchführung von eigenen Flugexperimenten zur weiteren Datenerhebung, um die genannten Defizite der vorgefertigten Datensätze zu schmälern. Allerdings sind eigene Flugversuche zur Datengewinnung kostenintensiv und zeitaufwendig. Außerdem bergen diese bei der Aufnahme von verschiedenen atmosphärischen Bedingungen, wie z. B. Nebel oder Regen, ein gewisses Risiko.

Aus diesen Gründen wurde eine eigene Datengenerierung durch *virtuelle Sensorsimulation* in Betracht gezogen. Diese hat den entscheidenden Vorteil, dass mit Hilfe der Simulationsumgebung beliebige Umgebungszustände erzeugt und die Bilddaten schneller als im Flugversuch erhoben werden können. Außerdem lässt sich eine automatische Annotation der Bilddaten durchführen, wodurch das zeitaufwendige manuelle Annotieren hinfällig wird.



Bild 3.5: Beispielluftbildaufnahmen aus veröffentlichten Datensätzen. Die Abbildung zeigt jeweils ein Beispielbild eines Datensatzes: (a) VIRAT, (b) ORIDS, (c) VIVID, (d) Mini-Drone, (e) VEDAI und (f) *TAS aerial car detection dataset*.

Bei der Verwendung von Sensorsimulationen stellt sich zunächst die Frage, ob die Sensordatenverarbeitung auf synthetischen Sensordaten ein gleichwertiges Verhalten wie auf korrespondierenden realen Sensordaten zeigt. Hierzu wurde in der Arbeit von Hummel [119] ein Konzept zur Evaluierung von Sensorsimulationen vorgestellt, welches anhand von Interest-Operatoren umgesetzt wurde. In der Arbeit wurde festgestellt, dass es möglich ist über eine Sensorsimulation Sensordaten zu erzeugen, die sich aus Sicht eines Interest-Operators wenig von einem realen Bild abweichen. Erreicht wurde die geringe Abweichung maßgeblich durch passende Texturierung und Modellqualität der Objekte. Des Weiteren beschäftigen sich einige Arbeiten [120–123] mit der Erzeugung und Evaluierung von Objektdetektoren auf Basis von synthetischen Bilddaten. Hierbei werden Objektdetektoren auf Basis von synthetischen und realen Bilddaten trainiert und anhand eines realen Testdatensatzes evaluiert und gegenübergestellt. Auch dabei zeigt sich, dass die Algorithmenleistung der Objektdetektoren vergleichbar ist. Aufgrund dieser Vorarbeiten wurde schließlich festgelegt, dass auch in dieser Arbeit synthetische Sensordaten zur Erzeugung der Leistungsmodelle und Objektdetektoren verwendet werden sollen. Allerdings sollten neben synthetischen Sensordaten auch reale, durch Flugexperimente erhobene, Sensordaten herangezogen werden, um das vorgestellte Konzept hinsichtlich seiner Generalisierungseigenschaften zu evaluieren (siehe Abschnitt 3.5).

Eine geeignete Simulationsumgebung muss für das in dieser Arbeit betrachtete Anwendungsbeispiel der Fahrzeugerkennung Sensordaten im visuellen und langwelligen Infrarotspektrum erzeugen können. Außerdem ist es von Vorteil, wenn ein breites Spektrum von Fahrzeugtypen und modellierte Landschaften zur Verfügung stehen. Moderne Computerspiele können zwar fotorealistische Bilddaten erzeugen, die unter anderem zum Lernen von Segmentierungsverfahren eingesetzt werden [124], besitzen aber meistens keine Möglichkeit LWIR-Sensoren zu simulieren (siehe Anforderung 1 in Abschnitt 3.1.1). Zur Anwendung soll daher schließlich die Simulationsumgebung VBS3⁴ (*"Serious Game"*) kommen, die auf Basis der Analyse von [119] den Anforderungen der VIS- und LWIR-Sensorsimulation, Modelldatenbank und modellierten Landschaften gerecht wird.

3.3.2 Algorithmen zur luftgestützten Fahrzeugerkennung

In Anbetracht der Verwendung einer Simulationsumgebung zur Erzeugung von synthetischen Bilddaten, sollen an dieser Stelle die vorausgewählten Algorithmen zur Fahrzeugerkennung (TM, BLOB, CC, DPM und CNN) aus Abschnitt 2.2.3 hinsichtlich möglicher Einschränkungen überprüft werden. Durch die Nutzung einer Sensorsimulation lassen sich für die trainierbaren Objektmodelle (CC, DPM und CNN) Trainingsdaten in großem Umfang erzeugen. Allerdings wird damit das Problem der Überanpassung (engl. *overfitting*) und der Generalisierung (engl. *generalization*) gegenüber realen Sensordaten prägnanter. Hierzu zeigt Bild 3.6 einen typischen Verlauf des Trainings- und Generalisierungsfehlers eines Objektmodells zur Fahrzeugklassifizierung über die Kapazität.



Bild 3.6: Typische Beziehungen zwischen Trainings- und Generalisierungsfehler [125].

Hierbei können die Trainingsdaten als synthetische und die Generalisierungsdaten als reale Bilddaten aufgefasst werden. Je mehr geeignete Trainingsdaten aus der Sensorsimulation generiert werden und je höher die *Kapazität* des Objektmodells ist, umso kleiner wird der Trainingsfehler. Die Kapazität eines Modells ist dabei die Fähigkeit sich an eine Vielzahl von Funktionen anpassen zu können. Modelle mit geringer Kapazität können daher

⁴ Informationen über die Simulationsumgebung VBS3 (Virtual Battlespace 3) sind auf der Herstellerseite <u>https://bisimulations.com/</u> [Letzter Zugriff: 04.10.2018] zu finden.

Schwierigkeiten haben, genügend Trainingsdaten im Modell zu berücksichtigen, um damit einen möglichst geringen Trainingsfehler zu erreichen. Beispielweise kann ein lineares Modell eine Gauß-Funktion nicht geeignet abbilden. Modelle mit hoher Kapazität unterläuft oft eine Überanpassung, indem sie alle Trainingsdaten berücksichtigen, welche zu größeren Fehlern an den Generalisierungsdaten führen. Ziel ist es die optimale Kapazität (engl. *optimal capacity*) eines Modelles zu finden.

Gerade tiefe neuronale Netze, welche eine höhere Kapazität gegenüber anderen Verfahren besitzen, neigen zur Überanpassung. Deshalb wurde in [126] und [127] festgestellt, dass die Verwendung von tiefen neuronalen Netzen zur Objektklassifikation auf Basis von synthetischen Sensordaten problematisch hinsichtlich der Generalisierung sein können. Der Grund für die Überanpassung liegt in der Merkmalsextraktion. Bei traditionellen Verfahren wird die Merkmalsextraktion im Voraus entworfen und lediglich der Klassifikator trainiert. Bei DNN wird zusätzlich die Merkmalsextration aus den Trainingsdaten bestimmt. Deshalb können diese die Artefakte in synthetischen Bildern, welche aus der Bildgenerierung stammen, erfassen und dadurch eine Generalisierung hinsichtlich realer Sensordaten schmälern. Grund hierfür sind z. B. wiederkehrende Texturen oder Fehler, die durch Antialiasing entstehen. Um den Problemen der Überanpassung und Generalisierung durch synthetische Daten entgegenzuwirken, sollen in der vorliegenden Arbeit daher folgende Maßnahmen berücksichtig werden:

- Die Datenmenge für das Training der Objektmodelle wird limitiert bzw. es wird nur eine Untermenge des zur Verfügung stehenden Datensatzes verwendet.
- Auf die Verwendung tiefer neuronaler Netze wird verzichtet, da keine etablierten Verfahren zur Umgehung von Artefakten der synthetischen Bildgenerierung existieren. Alle anderen Ansätze nutzen für die Merkmalsextraktion vorgefertigte Methoden und laufen damit nicht Gefahr die Artefakte der Bildgenerierung als dominante Merkmale zu verwenden.

Für die folgenden Untersuchungen verbleiben damit vier Methoden zur Fahrzeugdetektion:

- Template Matching mittels 3D-Modell
- BLOB-Detektion auf LWIR-Sensordaten
- Klassifizierungskaskade für unterschiedliche Szenenperspektiven
- Deformable Part Model

Neben diesen vier Objektdetektoren werden weitere Vorverarbeitungsroutinen als Perzeptionsmodulen inkludiert, um den Suchraum je nach Umgebungszustand einzuschränken und damit die Algorithmenleistung optimieren zu können. Vielversprechend ist dabei eine Straßensegmentierung (ST), wie in einigen genannten Veröffentlichungen (siehe Abschnitt 2.2.2) verwendet. Außerdem bietet sich eine Vorverarbeitung durch Farbraumanalyse (FA) und Vordergrundsegmentierung (VG), wie in [57] vorgestellt, an. Aus diesen einzelnen Perzeptionsmodulen (TM, BLOB, CC, DPM, ST, FA und VG) kann so der Perzeptionsgraph in Bild 3.7 aufgestellt werden.

Da die Straßen- und Vordergrundsegmentierung größere Bereiche im Sensorbild segmentieren können, wie z.B. ganze Straßenzüge oder Bodenbereiche, enthält der verwendete Perzeptionsgraph ein zusätzliches Perzeptionsmodul "Hypothesengenerierung mit Eckpunktdetektion". Dieser implementiert einen Kantendetektor zur Einschränkung des Suchraumes verwendet wird, da Fahrzeuge ausgeprägte Eckpunkte aufweisen.



Bild 3.7: Übersicht des verwendeten Perzeptionsgraphen zur Fahrzeugerkennung.

Der gezeigte Perzeptionsgraph besitzt 10 Perzeptionsketten, die zum Referenzieren mit Kürzeln versehen sind:

- DPM-ST: DPM-Detektor mit vorheriger Straßensegmentierung
- **DPM-VG**: DPM-Detektor mit vorheriger Vordergrundsegmentierung
- **DPM-FA**: DPM-Detektor mit vorheriger Farbraumanalyse
- CC-ST: Klassifikationskaskade mit vorheriger Straßensegmentierung
- CC-VG: Klassifikationskaskade mit vorheriger Vordergrundsegmentierung
- CC-FA: Klassifikationskaskade mit vorheriger Farbraumanalyse
- TM-ST: Template Matching mit vorheriger Straßensegmentierung
- TM-VG: Template Matching mit vorheriger Vordergrundsegmentierung
- **TM-FA**: Template Matching mit vorheriger Farbraumanalyse
- **BLOB**: BLOB-Detektor

Für die späteren Analysen werden die Klassifikatoren auch ohne Vorverarbeitung untersucht, woraus drei weitere "Perzeptionsketten" resultieren:

- **DPM**: DPM-Detektor ohne Vorverarbeitung
- CC: Klassifizierungskaskade ohne Vorverarbeitung
- TM: Template Matching ohne Vorverarbeitung

3.3.3 Umgebungszustände und deren Auswirkungen auf die Sensordaten

Zur Erfassung des Umgebungszustandes f_x wurden in Abschnitt 2.3.4 verschiedene Einflussfaktoren der Umgebung auf die Sensordatenverarbeitung diskutiert. Diese wurden in drei Kategorien gegliedert:

- Photogrammetrische Umgebungszustände
- Atmosphärische Umgebungszustände
- Topographische Umgebungszustände

Wie in Abschnitt 3.3.1 beschrieben, wird in dieser Arbeit eine Simulationsumgebung genutzt. Bild 3.8 zeigt die einstellbaren Parameter der gewählten Simulationsumgebung bzw. die

daraus extrahierbaren Umgebungszustände und ordnet diese den genannten Kategorien zu. Bei den berücksichtigten photogrammetrischen Umgebungszuständen lässt sich über den Zoomfaktor und das Sichtfeld (engl. field of view, FOV) die Bodenpixelauflösung bestimmen, welche Informationen über die Größe der Objekte in der Szene liefert. Der Sensorelevationswinkel gibt dabei die Verzerrung Die an. atmosphärischen Umgebungszustände enthalten Informationen über die Streuung in der Atmosphäre (Bewölkungsgrad, Nebeldichte, Niederschlag, Schneefall), spiegelnde Reflexionen (Niederschlag) und die Windstärke. Weiterhin kann der Sonnenstand (Tageszeit und Monat) eingestellt werden. Dieser zählt zwar nicht direkt zu den atmosphärischen Umgebungszuständen wird aber im Folgenden in diese Kategorie aufgenommen.

Dabei hat in der Simulation der Schneefall keine Auswirkung auf das Gelände, wie z. B. schneebedeckte Bäume, und die Windstärke beeinflusst hauptsächlich den Zustand von Wellen auf Meeresoberflächen. Daher werden Schneefall und Windstärke nicht zur Umgebungsbeschreibung herangezogen, da keine relevanten Effekte für die erdgerichtete Fahrzeugerkennung abgebildet werden. Die erfassten topografischen Kontextinformationen umfassen die Bodenbedeckung (Gebäude, Straßen, Wasser, Wiesen und Vegetation) sowie Informationen über die Geländehöhe.



Bild 3.8: Zuordnung der einstellbaren Parameter und Umgebungszustände der Simulationsumgebung. Alle in Blau markierten Parameter und Informationen werden zur Beschreibung der Umgebung verwendet und sind damit Teil des Umgebungszustandes.

Im Folgenden werden die einzelnen Umgebungszustände hinsichtlich ihrer Auswirkungen auf die Sensordaten diskutiert. Außerdem werden geeignete Erfassungsmöglichkeiten bei der Generierung der Metadaten und bei Einsatz der Leistungsmodelle im operationellen Betrieb diskutiert und ausgewählt.

3.3.3.1 Einflüsse der atmosphärischen Umgebungszustände

VIS- und LWIR-Sensoren werden unterschiedlich von atmosphärischen Effekten beeinflusst. Im vorherigen Abschnitt wurden als veränderbare Parameter der Sensorsimulation die Niederschlagsmenge, die Nebeldichte, der Bewölkungsgrad, die Tageszeit und der Monat identifiziert. Bild 3.9 zeigt zunächst den Einfluss der atmosphärischen Umgebungszustände auf die visuellen Bildmerkmale sowie die Erfassungsmöglichkeit. Dabei kann die Tageszeit und der Monat vom technischen System im operationellen Betrieb bzw. über die Simulationsumgebung bei der Erzeugung der Leistungsmodelle erfasst werden. Niederschlagsmenge, Nebeldichte und Bewölkungsgrad können während des Einsatzes der Leistungsmodelle entweder durch einen UAV-Operateur oder durch Wetterdaten ermittelt werden.



Bild 3.9: Einflussdiagramm der atmosphärischen Umgebungszustände auf die visuellen Bildmerkmale. Die erste Ebene, in hellblau gekennzeichnet, zeigt die Erfassungsmöglichkeiten der Umgebungszustände, dargestellt in der zweiten Ebene. Die Verbindungen zwischen den Umgebungszuständen und den Bildmerkmalen (letzte Ebene) zeigt die Beeinflussung dieser.



Bild 3.10: Einflussdiagramm der atmosphärischen Umgebungszustände auf die Bildmerkmale eines LWIR-Sensorbildes.

Bei der Metadatengenerierung können diese direkt über die verwendete Simulationsumgebung abgefragt werden. Da Wetterdaten lokal ungenau sind bzw. sich die atmosphärischen Bedingungen lokal ändern können, ohne dass die Wetterdaten dies widerspiegeln, wird die Eingabe der aktuellen Witterungsbedingungen im operationellen Betrieb durch den Operateur bevorzugt. Die Beeinflussungen der atmosphärischen Umgebungszustände auf die Bildmerkmale basieren auf den Erörterungen aus Abschnitt 2.3.2, wobei die versteckten Umgebungszustände *Jahreszeit & Sonnenstand*, abgebildet durch Tageszeit und Monat, als Verknüpfungspunkt zur geometrischen Betrachtung aus Bild 2.3 dient. Das Bildmerkmal *Dynamik* beschreibt die Ausprägung eines Bild-Histogramms. Wenn der Kontrast, die Sättigung, die Belichtung und der Schattenwurf gering sind, besitzt das Bild einen niedrigen Dynamikumfang. Dies kann für die Bildverarbeitung von Vorteil sein, da das Bild homogener und damit weniger Gradienten aufweist. Allerdings können dadurch auch Bildinformationen verloren gehen. Im Gegensatz dazu steht der hohe Dynamikumfang, der überbelichtete und unterbelichtete Bildregionen enthält. Dabei gehen ebenfalls wichtige Bildinformationen verloren und es bilden sich starke Gradienten aus, die zu Fehlinterpretationen der Bildverarbeitung führen können. Analog zu Bild 3.9 zeigt Bild 3.10 das Einflussdiagramm für den LWIR-Sensor.

3.3.3.2 Einflüsse der photogrammetrischen Umgebungszustände

Aus den Betrachtungen der geometrischen Eigenschaften von Luftbildaufnahmen in Abschnitt 2.3.1 konnten zwei elementare Abbildungsmerkmale, die Verzerrung und die Skalierung, identifiziert werden. Dabei bietet die verwendete Sensorsimulation die photogrammetrischen Umgebungszustände *Elevationswinkel, FOV* und *Zoomfaktor*. Bild 3.11 zeigt ein Einflussdiagramm der photogrammetrischen Umgebungszustände auf die Bildmerkmale. Dabei ist die Bodenpixelauflösung ein kombiniertes Maß und abhängig von dem FOV, dem Elevationswinkel und der Geländeform. In der Simulationsumgebung kann der Elevationswinkel des Sensors und die Bodenpixelauflösung direkt ausgelesen werden. Im Operationellen Betrieb kann der Elevationswinkel über die verbaute Sensorik zur Lageerfassung im Gimbal erfasst werden. Die Bodenpixelauflösung muss über die geometrischen Zusammenhänge (Elevationswinkel, Geländeform und Sensoroptik) berechnet werden.



Bild 3.11: Einflussdiagramm der photogrammetrischen Umgebungszustände auf die Bildmerkmale.

3.3.3.3 Einflüsse der topographischen Umgebungszustände

Die topographischen Umgebungszustände kodieren Informationen über die Geländeform und geographische Strukturen in den Sensordaten, wie z. B. Infrastruktur oder Vegetation, welche die Leistungscharakteristik verwendeter Sensordatenverarbeitung beeinflussen [89]. In Abschnitt 2.3.3 wurden dazu die beiden Umgebungszustände Bodenbedeckung und Geländeform diskutiert. Des Weiteren konnten die gängigen Kategorien Wasser, Ebene, Straßen, Gebäude und Vegetation der Bodenbedeckung festgestellt werden, welche ebenfalls in der Sen-

sorsimulation abgefragt werden können. Diese Umgebungszustände können sowohl direkt aus Sensordaten mittels *Scene Segmentation* gewonnen und räumlich zugeordnet als auch über Geoinformationssystem erhoben werden. In Bild 3.12 sind die topographischen Umgebungszustände in einem Diagramm dargestellt. Zum einen werden Erfassungsmöglichkeiten der Merkmale und zum anderen deren Einflüsse auf die entsprechenden Bildmerkmale aufgezeigt.



Bild 3.12: Einflussdiagramm der topographischen Umgebungszustände auf die Bildmerkmale. Nicht eingezeichnet ist hier der Einfluss auf die Dynamik des Bildes, welches von allen Bodenbedeckungsmerkmalen beeinflusst wird.

Tabelle 3.3 zeigt die Erfassungsmöglichkeiten der beiden Merkmale Bodenbedeckung und Geländeform. Dabei sind jeweils die Limitierungen bzw. Randbedingungen angegeben. Die Scene Segmentation [128] ist ein Verfahren, um aus VIS-Sensordaten eine Segmentierung des Bildes vorzunehmen und diese in Kategorien einzuteilen. Das Verfahren setzt allerdings Bilddaten im visuellen Spektrum voraus und kann damit nur bei Tageslicht eingesetzt werden. Eine weitere Möglichkeit, um die topographischen Umgebungszustände zu extrahieren, stellt die Verwendung einer vorher erstellten Datenbasis dar. Für das Geoinformationssystem existieren flächendeckende Datensätze für die Bodenbedeckung in Form von Vektordateien und für die Geländeform in Form von Rasterbanddaten. Weiterhin könnte der Operateur Eingaben zu den topographischen Umgebungszuständen liefern, wobei diese sich unter Umständen schnell ändern können oder nur schwer zu erfassen sind. Die Geländeform kann über Entfernungsmessung realisiert werden, z. B. mit LIDAR-Sensoren oder elektronischen Entfernungsmessungen (EMD), wobei ein zusätzlicher Sensor notwendig wäre. Da die Simulationsumgebung direkt Vektordateien und Rasterbanddaten exportieren kann und für das Gebiet, in dem die Flugversuche durchgeführt werden, ebenfalls entsprechende Daten vorhanden sind, wird als Erfassung der topographischen Umgebungszustände ein GIS herangezogen, sowohl für die Erzeugung als auch für die Anwendung der Leistungsmodelle während UAV-Missionen.

	Scene Segmentation	GIS	Operateur	LIDAR / EMD
Boden-	Nur bei Tageslicht	Datenbasis ist	Merkmal kann sich	
bedeckung	einsetzbar	notwendig	sehr schnell ändern	
Colöndoform		Datenbasis ist	Schwer zu erfassen	Zusätzlicher
Gelandelorm	—	notwendig	bzw. zu beschreiben	Sensor nötig

Tabelle 3.3: Gegenüberstellung der Erfassungsmöglichkeiten der topographischen Umgebungszustände.

3.3.3.4 Zusammenfassung der Erfassung der Umgebungszustände

Die vorgestellten Umgebungszustände sind in Tabelle 3.4 zusammengefasst. Dabei ist jeweils angegeben, wie die Umgebungszustände in der Sensorsimulation und im operationellen Betrieb erfasst werden. Bei der Erfassung der Umgebungszustände bei der Modellerzeugung meint die Angabe "Simulation", dass diese Parameter direkt über die Schnittstellen der verwendeten Simulationsumgebung ausgelesen werden können. Die Erfassung über das "technischem System" im operationellen Betrieb meint das direkte Auslesen von Parametern aus externen Systemen. Zum Beispiel kann der Elevationswinkel direkt über die Schnittstelle zum Gimbal ermittelt werden.

		Erfa	ssung	
	Umgebungszustände	bei der Modellerzeugung	im operationellen Betrieb	
	Niederschlagsmenge	Simulation	Operateur	
ch	Nebeldichte	Simulation	Operateur	
iris	Bewölkungsgrad	Simulation	Operateur	
sphä	Tageszeit X	Simulation	Technisches System	
mos	Tageszeit Y	Simulation	Technisches System	
atı	Monat X	Simulation	Technisches System	
	DemonstratingSimulationOperatedTageszeit XSimulationTechnischesTageszeit YSimulationTechnischesMonat XSimulationTechnischesMonat YSimulationTechnischesElevationswinkelSimulationTechnischesBodenpixelauflösungSimulationTechnischesWasserGIS-DatenGIS-Daten	Technisches System		
h.	Elevationswinkel	Simulation	Technisches System	
þ	Bodenpixelauflösung	Simulation	Technisches System	
	Wasser	GIS-Daten	GIS-Daten	
isch	Ebene	GIS-Daten	GIS-Daten	
ographi	Gebäude	GIS-Daten	GIS-Daten	
	Straße	GIS-Daten	GIS-Daten	
topc	Vegetation	GIS-Daten	GIS-Daten	
-	Reliefwert	GIS-Daten	GIS-Daten	

Tabelle 3.4: Zusammenfassung aller verwendeten Umgebungszustände und deren Erfassung. Die Abkürzung *ph.* steht hier für photogrammetrisch.

3.3.4 Methoden zur Leistungsmodellerzeugung durch maschinelles Lernen

Wie in Abschnitt 3.2.1 hergeleitet, sollen die Leistungsmodelle verwendet werden, um aus dem Umgebungszustand f_x einen Leistungsindex p für die Bildverarbeitungsalgorithmen individuell bestimmen zu können. Des Weiteren geht aus Anforderung 7 (siehe Abschnitt 3.1.2) hervor, dass die Leistungsmodelle durch wissensbasierte Verfahren und Methoden des maschinellen Lernens erzeugt werden sollen. Wie bereits in Abschnitt 2.5.3 ausgewählt, werden als wissensbasiertes Verfahren Fuzzy-Inferenzsysteme eingesetzt. Dieser Abschnitt behandelt nun die Methodenauswahl für die Erzeugung der Leistungsmodelle durch ML-Verfahren.

In Bild 3.13 ist beispielhaft ein Leistungsmodell mit den Ein- und Ausgängen dargestellt. Der Kern des Leistungsmodells stellt eine Funktion g dar, die f_x auf p abbildet, und die durch maschinelles Lernen bestimmt werden soll.



Bild 3.13: Beispielhafte Darstellung eines Leistungsmodells mit Ein- und Ausgängen.

Die Funktion $g(f_x)$ ist analytisch unbekannt, aber numerisch durch die Metadaten $\{F, P\}$ (siehe Abschnitt 3.3.4) stückweit abgetastet. Dabei kann die Beziehung zwischen dem abhängigen Leistungsindex und den unabhängigen Umgebungszustand durch *Funktionsapproximation* angenähert werden. In Abschnitt 2.5.3 wurden bereits einige Verfahren aus der Literatur diskutiert, die dazu verwendet wurden. Dabei konnten Methoden auf Basis von Ansätzen aus dem Bereich des maschinellen Lernens identifiziert werden, wobei hier weiterhin zwischen Methoden für lineare und nichtlineare Probleme bzw. anzunähernden Funktionen unterschieden wird. In Bezug auf den Anwendungsfall der Fahrzeugerkennung können für die Abbildung $g: F \to P$, $f_x \mapsto p$ folgende Eigenschaften angenommen werden:

- In Abhängigkeit von der Bodenpixelauflösung wird eine Fahrzeugerkennung in einem gewissen Entfernungsbereich einsetzbar sein. Dies bedeutet, dass eine gewisse minimale bzw. maximale Objektgröße benötigt wird, was sich z. B. mit nichtlinearen Trapez- oder Gauß-Funktion abbilden lässt.
- Da sich die Erscheinung eines Fahrzeuges mit dem Sensorelevationswinkel verändert, z. B. Seitenansicht versus Draufsicht, wird die Abhängigkeit nichtlinear sein und nur für bestimmte Winkelbereiche eine Fahrzeugerkennung möglich sein.
- Je nach Sensorart (VIS oder LWIR) beeinflusst der Sonnenstand bzw. die Tageszeit die Qualität der Sensordaten, was ebenfalls auf eine Nichtlinearität schließen lässt.

Aus diesen Eigenschaften heraus kann angenommen werden, dass die Funktion *g* ein nichtlineares Verhalten aufweisen wird. Gemäß Tabelle 2.6 aus Abschnitt 2.5.3 empfehlen sich dafür als mögliche Methoden zur Funktionsapproximation *Entscheidungsbäume* und *künstliche neuronale Netze*. Weitere bekannte Verfahren sind *polynomiale Regression, Support Vector Regression, Neuro-Fuzzy* Methoden und *Random Forests*. Alle genannten Methoden sind in Tabelle 3.5 anhand der folgenden Kriterien gegenübergestellt:

- **Große Datenmengen**: Durch den Einsatz der Simulationsumgebung werden größere Datenmengen (> 2 Mio. Bilddaten) erhoben, um so den Situationsraum genügend dicht abbilden zu können.
- Hohe Nichtlinearität: Da die Funktion *g*, je nach verwendetem Algorithmus, hohe Nichtlinearitäten aufweisen kann (s. oben), muss die gewählte Methode zur Funktionsapproximation damit umgehen können.
- **Hohe Genauigkeit**: Die Methode soll einen möglichst geringen Generalisierungsfehler erreichen können, und damit eine hohe Genauigkeit bei der Bestimmung der Leistungsindizes besitzen, um eine treffsichere Algorithmenauswahl zu realisieren.
- **Geringe Trainingszeit**: Damit die gewählte Methode eine hohe Genauigkeit erreichen kann, ist es empfehlenswert, die Modellerzeugung möglichst oft zu wiederholen. Den bei jedem neuen Training können die Startwerte (Hyperparameter) neu gewählt und damit eine Verbesserung des Generalisierungsfehlers erreicht werden. Daher sollte die Trainingszeit möglichst minimiert werden.

Aufgrund der Bewertung sollen für die vorliegende Arbeit *künstliche neuronale Netze* und *Neuro-Fuzzy Methoden* verwendet und in den Experimenten vergleichend untersucht werden. Zu bemerken ist, dass sich zwei gängige Neuro-Fuzzy Methoden, jeweils für Mamdani- und Sugeno-Typen, finden lassen. Nauck und Kruse [129] präsentieren den *NEuro Fuzzy function apPROXimator* (NEFPROX) Algorithmus, welcher einen Lernalgorithmus für FIS-Mamdani-Typen zur Funktionsapproximation darstellt. In [130] wird der *adaptive-network-based fuzzy inference system* (ANFIS) Algorithmus vorgestellt, um FIS-Sugeno-Typen zu trainieren. Auf die

x

		808011001 11011 01		
Verfahren	Große Datenmenge	Hohe Nichtlinearität	Hohe Genauigkeit	Geringe Trainingszeit
Polynomiale Regression	_	_	_	Х
Support Vector Regression	-	_	Х	_
Entscheidungsbäume	_	_	_	х
Neuro-Fuzzy	Х	Х	_	Х

Sugeno-Typen wird in dieser Arbeit verzichtet, da sie weder die Möglichkeit bieten, Expertenwissen zu kodieren, noch Vorteile gegenüber neuronalen Netzen besitzen.

Tabelle 3.5: Gegenüberstellung der Verfahren zur Funktionsapproximation aus dem Bereich maschinellem Lernen.

x

3.4 Konzeptdetaillierung der Leistungsmodellierung durch Simulationseinsatz

x

Random Forests

Neuronale Netze

Um die Verwendung einer simulationsbasierten Datengenerierung zu reflektieren, wird in diesem Abschnitt das Konzept zur Erzeugung der Leistungsmodelle (siehe Bild 3.2) um einige Details ergänzt. Außerdem wird das Konzept hinsichtlich der gewählten Methoden zur Funktionsapproximation aus vorherigem Abschnitt verfeinert. Die synthetischen Sensordaten finden in diesem Kontext ihre Anwendung zur Generierung folgender Datensätze:

- Der *Objektdatensatz*, der zum Training der Objektdetektoren (CC und DPM) mit trainierbaren Modellen verwendet wird.
- Der *Modelldatensatz*, der zur Erzeugung der Leistungsmodelle durch ML eingesetzt wird, bestehend aus Sensordaten *X* und Umgebungszustandsraum *F*.

Damit die Leistungsmodelle der Perzeptionsmodule bestimmt werden können, müssen zuerst die Klassifikatoren der Klassifizierungskaskade und des DPM trainiert werden. Dazu wird über die Simulationsumgebung der *Objektdatensatz* generiert, welcher zum Training der Klassifikatoren verwendet wird. Im Anschluss daran können die Leistungsmodelle erzeugt werden. Zur Übersicht zeigt Bild 3.14 das detaillierte Konzept zur Leistungsmodellerzeugung.



Bild 3.14: Detailliertes Konzept zur Erzeugung der Leistungsmodelle durch synthetische Datengenerierung.

Für die Leistungsmodellierung wird zunächst der *Modelldatensatz*, bestehend aus Sensordaten und korrespondierendem Umgebungszustand, mittels synthetischer Datengenerierung erzeugt. Anschließend findet die Berechnung der Leistungsindizes für jedes Perzeptionsmodul und jedes Beispiel aus den Sensordaten statt. Die ermittelten Leistungsindizes werden zusammen mit den korrespondierenden Umgebungszuständen als *Metadaten* $\{F, P\}$ bezeichnet. Die Verfahren zur Generierung der Metadaten zeigt das folgende Kapitel 4.

Mittels Funktionsapproximation (KNN und NEFPROX) werden aus den *Metadaten* die Leistungsmodelle M_i^{KNN} und M_i^{NEF} pro Perzeptionsmodul a_i abgeleitet. Dabei erzeugt der NEFPROX-Algorithmus ein FIS anhand der *Metadaten*. Weiterhin wird die Modellierung der Leistungsmodelle M_i^{EXP} mittels Fuzzy-Inferenzsysteme betrachtet, um das Leistungsverhalten von Perzeptionsmodulen durch Erfahrungswerte bzw. Expertenwissen zu ermöglichen (siehe Anforderung 7 und 8 in Abschnitt 3.1.2). Bild 3.15 zeigt eine Veranschaulichung der verschiedenen Typen von Leistungsmodellen und mit welcher Basis sie erzeugt werden. Dabei ist anzumerken, dass neben der Verwendung von FIS für die Expertenmodellierung auch der NEFPROX-Algorithmus ein FIS generiert.



Bild 3.15: Veranschaulichung der verwendeten Typen von Leistungsmodellen. In dunkelblau sind die Ausgangsdaten zur Leistungsmodellierung angegeben und in hellblau die Typen er Leistungsmodelle. Die blauen Blöcke zeigen die ML-Verfahren zur Leistungsmodellerzeugung.

Die drei verwendeten Typen von Leistungsmodellen sollen bei der Evaluierung vergleichend untersucht werden. Diese verfolgen unterschiedliche Ansätze zur Leistungsmodellierung:

- Die Experten-Leistungsmodelle M_i^{EXP} können verwendet werden, wenn keine aufwendige Datenbasis verfügbar ist oder nicht erzeugt werden kann, dafür aber Experten- oder Erfahrungswissen zur Verfügung steht, um die Leistungscharakteristik der Perzeptionsmodule zu modellieren.
- Die NEFPROX-Leistungsmodelle M_i^{NEF} bieten die Möglichkeit interpretierbare Modelle aus den Metadaten zu erzeugen.
- Die KNN-Leistungsmodelle M_i^{KNN} zählen zu den *Black-Box-Modellen*, die zwar nicht mehr interpretierbar aber eine höhere *Kapazität* als interpretierbare Modelle besitzen, und damit potenziell eine genauere Funktionsapproximation liefern können.

Tabelle 3.6 zeigt welche Typen von Leistungsmodellen für welche Perzeptionsmodule verwendet werden. Die Verfahren zur Leistungsmodellierung werden in Kapitel 5 beschrieben. Das Kapitel 6 zeigt die Verfahren zur Anwendung der Leistungsmodelle, um damit eine Algorithmenauswahl zu realisieren.

	ST	VG	FA	BLOB	ТМ	CC	DPM
M_i^{EXP}	Х	Х	Х	Х	Х	Х	х
M_i^{NEF}				х	х	х	х
M_i^{KNN}				Х	Х	Х	х

Tabelle 3.6: Zuordnung der Leistungsmodelltypen zu den Perzeptionsmodulen.

3.5 Evaluierungsstrategie

Abschließend wird in diesem Abschnitt die Evaluierung des adaptiven luftgestützten Sensoreinsatzes bzw. der Algorithmenauswahl vorgestellt. Das Vorgehenskonzept zur Evaluierungsstrategie der Algorithmenauswahl illustriert Bild 3.16. Hierbei sollen Flugmissionen durchgeführt werden, um damit typische Sensordaten X_e^i , Umgebungszustände F_e^i und Leistungsindizes P_e^i eines Missionsfluges *i* aufzunehmen. Diese aufgezeichneten Daten werden im folgenden *Evaluierungsdatensatz* genannt. Dabei werden die Missionen zum einen mittels Simulation und zum anderen in Flugversuchen durchgeführt. Während einer Mission werden die Leistungsindizes P_e^i aller Perzeptionsketten im Perzeptionsgraphen durch die Leistungsmodelle ermittelt und die leistungsstärkste Perzeptionskette eingesetzt.

Um die Korrektheit der Algorithmenauswahl zu analysieren, werden im Anschluss an eine Mission die Leistungsindizes P_g^i (*Ground Truth*) aller Algorithmen anhand der aufgezeichneten Sensordaten gemessen. Die Messung erfolgt für jede Perzeptionskette, indem die aufgezeichneten Sensordaten verarbeitet werden und die resultierenden Fahrzeugdetektionen mit den bekannten Fahrzeugpositionen abgeglichen werden. Daraus wird die Algorithmenleistung ermittelt wird. Schließlich resultieren daraus die Ergebnisdaten $\{F_e, P_e, P_g\}_i$. Diese können dann zur Analyse der Leistungsmodelle und quantitativen Evaluierung der Perzeptionskettenauswahl herangezogen werden.



Bild 3.16: Vorgehenskonzept der Evaluierungsstrategie der Algorithmenauswahl.

Bevor die Evaluierung durchgeführt werden kann, behandeln die folgenden Kapitel die Verfahren zur Erzeugung der Leistungsmodelle und deren Anwendung zur Algorithmenauswahl. Kapitel 4 und Kapitel 5 beschreiben die Verfahren zur Leistungsmodellierung, wobei Kapitel 4 die Vorbereitungen für die Leistungsmodellierung mittels ML-Verfahren, die Generierung der *Metadaten*, behandelt. Kapitel 5 zeigt dann die Verfahren zur Erzeugung der Experten-, NEFPROX- und KNN-Leistungsmodelle. Die Verfahren zur Algorithmenauswahl mittels der erzeugten Leistungsmodelle und die Integration der Algorithmenauswahl in das SPMS wird in Kapitel 6 vorgestellt.

KAPITEL 4 VERFAHREN ZUR VORBEREITUNG DER LEISTUNGSMODELLIERUNG

Damit in Kapitel 5 das Verfahren zur Erzeugung der Leistungsmodelle beschrieben werden kann, müssen zunächst, unter Verwendung des Modelldatensatzes, die Leistungsindizes der unterschiedlichen Algorithmen mit den variierenden Umgebungszuständen in Zusammenhang gebracht werden. Diese Zuordnung wird als *Metadaten* bezeichnet (siehe Abschnitt 3.2.1). Bild 4.1 zeigt anhand des detaillierten Konzeptes aus Abschnitt 3.4 die in diesem Kapitel behandelten Verfahrensschritte:

- (1) In Abschnitt 4.1 wird beschrieben, welche Informationen in den Sensordaten $x \in X$ vorhanden sind.
- (2) In Abschnitt 4.2 wird erläutert, wie die Umgebungszustände f_x erfasst und berechnet werden.
- (3) Die Generierung des *Modelldatensatzes* {*X*, *F*} aus der Simulationsumgebung wird in Abschnitt 4.4 vorgestellt.
- (4) Abschnitt 4.5.2.5 zeigt die Generierung des *Objektdatensatzes,* welcher für das Anlernen der trainierbaren Klassifikatoren (CC und DPM) eingesetzt wird.
- (5) In Abschnitt 4.5 werden die verwendeten Perzeptionsmodule $\alpha \in A$ beschrieben und das Training der Klassifikatoren präsentiert.
- (6) Die Definition des Leistungsindex $p \in P$ erfolgt in Abschnitt 4.6.
- (7) Die Generierung der Metadaten $\{F, P\}$ wird schließlich in Abschnitt 4.7 vorgestellt.



Bild 4.1: Überblick der Verfahrensschritte zur Erzeugung der Metadaten.

4.1 Simulation der Sensordaten

Wie in Abschnitt 3.3.1 erläutert, sind vorgefertigte Datensätze mit VIS- und LWIR-Sensordaten für die gegebene Anwendung nur limitiert verfügbar und die Erhebung solcher Daten durch

Flugexperimente zeitaufwendig oder unter den geforderten Bedingungen nicht durchführbar. Daher wird nun eine Sensorsimulation verwendet (siehe Abschnitt 3.4), um entsprechende Sensordaten zu erzeugen. Um bei der in dieser Arbeit betrachteten Anwendung der Fahrzeugerkennung die Leistungsfähigkeit der Perzeptionsmodule messen zu können, müssen Informationen über das wahre Auftreten von Fahrzeugen in den Sensordaten vorhanden sein. Dadurch lässt sich entscheiden, ob ein Perzeptionsmodul ein Fahrzeug erkennen konnte. Damit beinhalten die Sensordaten \tilde{I} Objektannotationen \tilde{A} , welche Objektposition, beschrieben als Rechteck, die Objektorientierung und den Objekttyp umfassen. Diese Daten werden automatisiert aus der Simulationsumgebung extrahiert. Das in den VIS- und LWIR-Sensordaten vorhandene Bildmaterial besitzen hierbei eine Bildgröße von 1920 $px \times 1080px$. Bild 4.2 zeigt ein Beispielbild mit entsprechenden Objektannotationen.



Bild 4.2: Synthetisches Beispielsensorbild im visuellen Spektrum mit Objektannotationen.

4.2 Bestimmung der Umgebungszustände

Im Folgenden wird die Bestimmung der ausgewählten Umgebungszustände aus Abschnitt 3.3.3 vorgestellt. Dabei wurden unterschiedliche Erfassungsmöglichkeiten für die Leistungsmodellierung und den operationellen Einsatz ausgewählt (siehe Abschnitt 3.3.3.4). Dazu werden in diesem Abschnitt die entsprechenden Berechnungsvorschriften angegeben.

Um die einzelnen Größen der Umgebungszustände einer Funktionsapproximation zuführen zu können, erfolgt eine Normierung der einzelnen Umgebungszustände, wie in den folgenden Abschnitten 4.2.1 bis 4.2.3 angeben. Diese Normierung soll die Erzeugung der Leistungsmodelle durch ML-Methoden schneller konvergieren lassen. Zum andern sorgt eine geeignet gewählte Normierung für vergleichbare Werteskalen der Umgebungszustände, wodurch gerade beim Training von künstlichen neuronalen Netzen die Chance auf das Erreichen eines globalen Optimums erhöht wird.

4.2.1 Atmosphärische Umgebungszustände

Dieser Abschnitt behandelt die atmosphärischen Umgebungszustände Niederschlag, Nebeldichte und Bewölkungsgrad sowie Tageszeit und Monat.

4.2.1.1 Niederschlag

In der verwendeten Sensorsimulation lässt sich der Niederschlag mit einem Wert $Q_{r,s}$ zwischen 0 und 1 vorgeben. Dabei bedeutet der Wert 0 keinen Niederschlag und der Wert 1 einen maximalen Niederschlag. Als Referenz wird die physikalische Größe der Niederschlagsmenge Q_r herangezogen, welche in Litern pro Quadratmeter und Stunde bzw. mit der Einheit mm/h angegeben wird. Dabei ist die maximale Niederschlagsmenge in der Simulationsumgebung einstellbar und wurde im gegebenen Fall zu 60 mm/h gewählt. Theoretisch ist die Niederschlagsmenge nach oben nicht limitiert, daher wird die normierte Niederschlagsmenge Q'_r definiert, die einen Wertebereich zwischen 0 und 1 besitzt. Als Normierungsgröße wird der Standardwert für starken Regen herangezogen, der bei einer Niederschlagsmenge von etwa 10 mm/h liegt [69]. Die Normierungsvorschrift lautet damit

$$Q_r' = \tanh\left(\frac{Q_r}{10\frac{mm}{h}}\right),\tag{4.1}$$

wobei die Überführung der Niederschlagswerte aus der Simulation mit $Q_r = Q_{r,s} \cdot 60 \text{ mm/s}$ erfolgt. Durch die Verwendung des Tangens hyperbolicus wird der Wertebereich zwischen 0 und 10 mm/h gleichmäßig aufgeteilt und erfasst den relevanten Bereich zwischen keinem und starkem Niederschlag. Dabei wird trotzdem noch Spielraum für höhere Niederschlagsmengen eingeräumt. In Bild 4.3 sind Aufnahmen aus der Simulation dargestellt, die den Effekt von Niederschlag für VIS- und LWIR-Sensoren zeigt bei leichtem ($Q'_r = 0,3$), mittlerem ($Q'_r = 0,5$), starkem ($Q'_r = 0,7$) und sehr starkem Regen ($Q'_r = 1$). Dabei ist zu beobachten, dass die verwendete Sensorsimulation im sichtbaren Spektrum keine nassen Oberflächen darstellen kann und damit auch keine Spiegelungen erzeugt werden können. Im thermischen Spektrum zeigt sich ein verringerter Kontrast mit zunehmender Niederschlagsmenge. Außerdem ist in Bild 4.4 die normierte Niederschlagsmenge Q'_r über Niederschlagsmenge Q_r aufgetragen.



Bild 4.3: Darstellung der Niederschlagsmenge in der Sensorsimulation. Die abgebildeten Aufnahmen zeigen von links nach rechts leichten, mittleren, starken und sehr starken Niederschlag. In der oberen Zeile (a-d) befinden sich Bilder vom VIS Sensor und in der unteren (e-h) vom LWIR Sensor.

Damit der Operateur während des operationellen Einsatzes die Niederschlagsmenge erfassen kann, empfiehlt sich eine einheitenlose Größe, da die physikalische Größe Niederschlagsmenge Q_r aus den Sensordaten schwer abzuleiten ist. Daher soll der Operateur direkt die normierte Niederschlagsmenge Q'_r angeben, wobei zur besseren Interpretation eine semantische Beschreibung eingeführt wird. Diese dient gleichzeitig als Modellierungsgrundlage für die durch Expertenwissen erzeugten Leistungsmodelle:

- Keiner bzw. kaum Niederschlag: $0 \le Q'_r < 0,1$
- Leichter Niederschlag: $0,1 \le Q'_r < 0,3$
- Mittlerer Niederschlag: $0.3 \le Q'_r < 0.5$
- Starker Niederschlag: $0,5 \le Q'_r < 0,7$
- Sehr starker Niederschlag: $0,7 \le Q'_r \le 1$



Bild 4.4: Normierte Niederschlagsmenge in Abhängigkeit der Niederschlagsmenge.

4.2.1.2 Nebeldichte

Die Nebeldichte wird in der verwendeten Sensorsimulation durch einen einheitenlosen Wert dargestellt, der einen Wertebereich zwischen 0 und 1 besitzt. Da dieser aber keine Interpretation erlaubt, wird die Sichtweite in Metern als physikalische Größe eingeführt. Außerdem kann die Sichtweite durch einen Operateur besser abgeschätzt werden. Der Zusammenhang zwischen Nebeldichte und Sichtweite wurde in der Sensorsimulation empirisch ermittelt. In Bild 4.5 sind dazu Abbildungen von verschiedenen Sichtweiten bei nebliger Szenerie aus der Simulation zu sehen. Um den Effekt von Nebel zu berücksichtigen, wird die normierte Nebeldichte ρ'_f definiert, welche von der Sichtweite abhängt. Die normierte Nebeldichte wird bei theoretischer unendlicher Sichtweite 0, was klarem Wetter entspricht, und 1 bei 0 *m* Sichtweite, und ist damit gleichzusetzen mit dem einheitenlosen Wert der Sensorsimulation. Folgende Formel definiert die normierte Nebeldichte:

$$\rho_f' = -\tanh\left(\frac{s}{5000\,m}\right) + 1\tag{4.2}$$

Zu beachten ist, dass der Normierungsfaktor von 5000 *m* durch die Sensorsimulation empirisch ermittelt wurde. Das Diagramm in Bild 4.6 zeigt die normierte Nebeldichte ρ'_f aufgetragen über die Sichtweite.


Bild 4.5: Effekt der Nebeldichte in der Sensorsimulation. Die vier dargestellten Bilder zeigen von links nach rechts den Bildeffekt durch steigende Nebeldichte. Dabei ist in (a) kein Nebel vorhanden sowie in (b) 6000 m Sichtweite, in (c) 2000 m Sichtweite und in (d) 500 m Sichtweite eingestellt.



Bild 4.6: Normierte Nebeldichte in Abhängigkeit der Sichtweite.

4.2.1.3 Bewölkungsgrad

Die Sensorsimulation bietet einen einheitenlosen Wert, der zwischen 0 und 1 gewählt werden kann. Dabei entspricht ein Bewölkungsgrad von 0 einer ungetrübten Sonneneinstrahlung auf die Szene und 1 einer nicht mehr sichtbaren Sonne, wodurch die Szene lediglich durch diffuses Licht beleuchtet wird. Bild 4.7 zeigt den simulierten Bewölkungsgrad, sowie die Effekte Schattenwurf und Beleuchtungsintensität im visuellen Spektrum. Außerdem sind die Temperaturunterschiede im LWIR-Sensorbild dargestellt. Der Flugwetterdienst nutzt zur Beschreibung des Bewölkungsgrads die *Okta*-Skala. Diese teilt den Bewölkungsgrad in die folgenden Stufen ein:

- CAVOK (engl. *clouds and visibility okay*), wolkenloser Himmel, 0/8
- NSC (engl. no significant clouds), keine wesentlichen Wolken, 1/8
- FEW (engl. *few*), wenig bewölkt, 2/8
- SCT (engl. *scattered*), aufgelockert bewölkt, 3/8 4/8
- BKN (engl. broken), durchbrochene Wolkendecke, 5/8 7/8
- OVC (engl. overcast), geschlossene Wolkendecke, 8/8
- OBSC (engl. *obscured*), Himmel nicht sichtbar, 9/8

Damit die Werte aus der Simulation mit den Angaben eines Operateurs in Beziehung gesetzt werden können, wird daher der Bewölkungsgrad W_o definiert. Dieser besitzt einen Wertebereich identisch zum einheitenlosen Wert der Sensorsimulation. Zur Interpretierbarkeit und

Modellierung der Leistungsmodelle werden zusätzlich folgende semantische Begrifflichkeiten verwendet:

- Wolkenlos: $0 \le W_o < 1/8$
- Wenig bewölkt: $1/8 \le W_o < 3/8$
- Aufgelockert bewölkt: $3/8 \le W_o < 5/8$
- Durchbrochene Wolkendecke: $5/8 \le W_o < 7/8$
- Geschlossene Wolkendecke: $7/8 \le W_o \le 1$





(f)

(e)







Bild 4.7: Darstellung der Effekte durch den Bewölkungsgrad auf einen VIS- und LWIR-Sensor. Der Bewölkungsgrad steigt in jeder Zeile von links nach rechts an, mit 0/8, 3/8, 5/8 und 8/8 Bewölkungsgrad. In der ersten Zeile (a-d) ist die Bewölkung dargestellt, in der mittleren Zeile (e-h) entsprechende Bilder des VIS-Sensors und in der letzten Zeile (i-l) die Aufnahmen des LWIR-Sensors.

4.2.1.4 Tageszeit

Die Tageszeit ist eine periodische Variable, die einen Sprung bei 0 bzw. 24 Uhr besitzt. Dies kann beim Trainieren der Leistungsmodelle zu Fehlern führen, da 0 und 24 Uhr den identischen Sonnenstand beschreiben, aber aus Sicht einer Funktionsapproximation unterschiedliche Zustände definieren. Um diesen Sprung zu vermeiden, wird die Tageszeit als Polarkoordinate interpretiert und in normierte kartesische Koordinaten gewandelt, wodurch sich zwei Variablen für die Tageszeit ergeben:

$$x_t = \sin\left(t\frac{2\pi}{24}\right) \quad y_t = \cos\left(t\frac{2\pi}{24}\right) \tag{4.3}$$

Dabei können die Variablen x_t und y_t durch die Normierung Werte zwischen -1 und 1 annehmen. Bild 4.8 zeigt die Abbildung der Tageszeit auf die beiden Variablen. Dabei ist anzumerken, dass die Tageszeit als Dezimalzeit aufzufassen ist, damit würde z. B. 7:21 Uhr dem Wert 7,35 entsprechen.



Bild 4.8: Darstellung des Umgebungszustandes Tageszeit. Die Grafik zeigt die periodische Variable Tageszeit, abgebildet auf die beiden normierten kartesischen Koordinaten bei einer Tageszeit von 3:00 Uhr.

Erfasst wird die Tageszeit in während der Modellerzeugung (in der Simulation) sowie bei der Modellanwendung (operationeller Einsatz) direkt über die Systemzeit der verwendeten Rechnerkomponenten. Dabei muss sichergestellt werden, dass diese auf die gleiche Zeitbasis synchronisiert sind. Verwendet wurde die mitteleuropäische Sommerzeit (engl. *Central European Summer Time*, CEST).

4.2.1.5 Monat

Die Tageszeiten von Sonnenaufgang und -untergang sowie der Sonnenstand variieren je nach Jahreszeit und beeinflussen damit die Ausprägung der Sensordaten des VIS- und LWIR-Sensors. Als Jahreszeit werden die Monate nach dem gregorianischen Kalender herangezogen. Da die Jahreszeiten je nach Position auf der Erde verschieden sind, wird diese Definition auf der nördlichen Hemisphäre verwendet. Da es sich beim Monat ebenfalls um eine periodische Variable handelt, wird diese als Polarkoordinaten interpretiert und nach normierten kartesischen Koordinaten umgewandelt:

$$x_M = \sin\left(M\frac{2\pi}{12}\right) \quad y_M = \cos\left(M\frac{2\pi}{12}\right) \tag{4.4}$$

Dabei ist M der numerische Monat und x_M und y_M sind die beiden resultierenden Variablen zur Kodierung der Jahreszeit.

4.2.2 Photogrammetrische Umgebungszustände

In diesem Abschnitt werden die photogrammetrischen Umgebungszustände Elevationswinkel und Bodenpixelauflösung behandelt.

4.2.2.1 Elevationswinkel

Der Elevationswinkel eines Sensors gibt z. B. Aufschluss darüber ob ein Fahrzeug durch die Draufsicht oder in Seitenansicht wahrgenommen wird. Typischerweise wird der Elevationswinkel θ in einem Wertebereich von 90° bis –90° angegeben. Dieser wird für die weiteren Verarbeitungen normiert mit:

$$\theta' = \frac{\theta}{90^{\circ}} \tag{4.5}$$

Dabei ist θ' der normierte Elevationswinkel mit dem Wertebereich von -1 bis 1, wobei bei der hier betrachteten Anwendung der Fahrzeugdetektion der interessante Wertebereich zwischen -1 und 0 liegt.

Der Sensor kann durch die Fluglage des UAVs um die optische Achse des Sensors verdreht sein. Zur Korrektur bietet sich dabei eine digitale Bildstabilisierung an, weshalb die Verdrehung um die optische Achse nicht Teil der Umgebungszustände ist. Eine Kompensation durch eine kardanische Aufhängung ist ebenfalls möglich, allerdings sind die meisten kleineren Gimbal-Systeme nur mit Höhen- und Vertikalachsen ausgestattet. Im realen Betrieb der Perzeptionskettenauswahl kann der Elevationswinkel über die Statusinformationen eines Gimbal-Systems erhoben werden, und die verwendete Simulationsumgebung bietet entsprechende Schnittstellen, um den Elevationswinkel zu setzen oder abzufragen.

4.2.2.2 Bodenpixelauflösung

Neben dem Blinkwinkel auf die Szene ist auch die Entfernung des Sensors zur Szene relevant, da diese die Objektgröße in Abhängigkeit der Sensorauflösung beeinflusst. Für die Objekterkennung spielt die Größe der Objekte eine Rolle, wobei einige Objektdetektoren begrenzt skalierungsinvariant sind, z. B. durch den Einsatz einer Skalierungspyramide. In der Fernerkundung mit bildgebenden Sensoren hat sich die Bodenpixelauflösung (engl. *ground sample distance*) [131] als Messwert etabliert, die die abgebildete Größe eines Pixels auf der Bodenoberfläche in Metern angibt. Für diesen Umgebungszustand wird die mittlere Bodenauflösung mittels der abgetasteten Fläche des Sensor-Footprints berechnet. Diese Fläche wird mit

$$A = \frac{\left|\overline{p_{TL}^{u} p_{TR}^{u}} \times \overline{p_{TL}^{u} p_{BL}^{u}}\right| + \left|\overline{p_{BL}^{u} p_{TR}^{u}} \times \overline{p_{BL}^{u} p_{BR}^{u}}\right|}{2}$$
(4.6)

bestimmt, wobei p_{TL}^u die obere linke, p_{TR}^u die obere rechte, p_{BL}^u die untere linke und p_{BR}^u die untere rechte Ecke des projizierten Bildes auf die Bodenoberfläche ist. Die Berechnung des Sensor-Footprints im operationellen Einsatz ist in Anhang B hergeleitet. Da die mittlere Bodenpixelauflösung *g* theoretisch von Null bis unendlich reichen kann, wird sie normiert mit

$$g = \sqrt{\frac{A}{H \cdot W}} \quad g' = \tanh(f_g \cdot g) \tag{4.7}$$

wobei *H* und *W* die Höhe und Breite des Bildes in Pixeln ist. Der Faktor f_g dient zur Skalierung der Werte für den Tangens hyperbolicus. Bestimmt wird er durch den Kehrwert der maximalen auftretenden Bodenpixelauflösungswert g_{max} des *Modelldatensatzes*. In Bild 4.9 ist die normierte Bodenauflösung g' über die skalierte mittlere Bodenauflösung dargestellt. Dabei ist zu sehen, dass durch den Skalierungsfaktor der Wertebereich außerhalb des

Sättigungsbereiches des Tanges hyperbolicus gelegt wird. Für Fälle von größeren skalierten mittleren Bodenpixelauflösungswerten bleibt dabei noch Spielraum im verbleibenden Wertebereich.



Bild 4.9: Faktorbestimmung der Bodenpixelauflösung. Darstellung der skalierten mittleren Bodenpixelauflösung durch den Faktor f_g , um den Wertebereich vor den Sättigungsbereich des Tanges hyperbolicus zu verschieben.

Im Gegensatz zum realen Betrieb der Algorithmenauswahl, muss in der Simulationsumgebung der Sensor-Footprint nicht berechnet werden, da diese Schnittstellen bietet, um die Punkte p_{TL}^{u} , p_{TR}^{u} , p_{BL}^{u} und p_{BR}^{u} abzufragen.

4.2.3 Topographische Umgebungszustände

Der Abschnitt stellt die Bestimmung der topographischen Umgebungszustände Bodenbedeckung und Geländeform vor.

4.2.3.1 Bodenbedeckung

Im Folgenden wird der topographische Umgebungszustand Bodenbedeckung und dessen 5 Kategorien *Wasser, Ebene, Straße, Gebäude* und *Vegetation* behandelt. Die Kategorie Vegetation umfasst dabei Bäume, Wälder, Büsche und sonstigen Bewuchs, wobei Wiesen und Grünflächen davon ausgenommen sind. Seen, Flüsse, Meere und sonstige Wasserflächen gehören der Kategorie Wasser an. In der Kategorie Ebene wird alles Sonstige, außer Gebäuden und Straßen, zusammengefasst, wie z. B. Grünflächen, Ackerland oder Strand.

Die fünf Kategorien zur Beschreibung der Bodenbedeckung werden mittels des Sensor-Footprints bestimmt. Dabei werden aus einem Geoinformationssystem die jeweiligen Flächenanteile von Wasser, Vegetation, Gebäuden und Straßen innerhalb des Sensor-Footprints ermittelt und durch die Gesamtfläche des Sensor-Footprints geteilt. Damit lässt sich ausdrücken, wie viele Anteile in der Szene welcher Kategorie angehören. Der verbleibende Flächenanteil fällt immer der Kategorie Ebene zu, wobei die Summe der fünf Kategorien immer 1 ergibt.

In Bild 4.10 sind beispielhaft die Vektordaten eines Kartenausschnittes zu sehen, mit Überlagerung des Sensor-Footprintes, und dem entsprechenden Bild eines bildgebenden Sensors im visuellen Spektrum.



Bild 4.10: Beispielhafte Darstellung der Bodenbedeckung. Die Darstellung zeigt in (a) in Gelb den Sensor-Footprint, der auf die Vektordaten im GIS gelegt ist und in (b) das entsprechende Bild des VIS Sensors aus der Simulation. Die ermittelten Anteile der Szenenkategorien in diesem Beispiel sind: 0,19 Vegetation; 0,11 Gebäude; 0,00 Wasser; 0,21 Straße und 0,49 Ebene.

4.2.3.2 Geländeform

Neben der Bodenbedeckung trägt die Beschaffenheit der Bodenoberfläche (Geländeform) zur Topographie bei. Eine schroffe Berglandschaft ist im Gegensatz zu einer ebenen Grünfläche schwieriger zu analysieren, da es zu Verdeckungen kommen kann und Objekte verkippt bzw. verdreht dargestellt werden können. Daher wird der Reliefwert r zu den Umgebungszuständen hinzugefügt, der die Rauheit der Bodenoberfläche beschreibt. Die Berechnung des Reliefwertes wird mittels Höhendaten bzw. Rasterbanddaten durchgeführt, wobei diese durch ein GIS bereitgestellt werden. Dazu wird der Sensor-Footprint zunächst durch seine minimale und maximale geographische Breite sowie Länge zu einem Rechteck aufgezogen. Der Grund hierfür liegt in der Anordnung der Höhendaten als orthogonales Raster mit einer bestimmten Gradauflösung. Im Anschluss daran werden die Höhendaten im Bereich des orthogonalen Sensor-Footprints gleichmäßig abgetastet, so dass sich eine Höhenmatrix H ergibt mit $k \times k$ Elemente. Bild 4.11 zeigt die Abtastung der Höhendaten beispielhaft mit k = 7. Dabei wird die minimale und maximale geographische Breite des Sensor-Footprints mit φ_{min} und φ_{max} bezeichnet und die minimale und maximale geographische Länge mit λ_{min} und λ_{max} angegeben. Die Schrittweite der Abtastung bestimmt sich entsprechend durch die Anzahl der Abtastpunkte k. Dabei ist φ_{step} die Schrittweite der geographischen Breite und λ_{step} die Schrittweite der geographischen Länge.



Bild 4.11: Abtastung der Höhendaten zur Bestimmung des Reliefwertes. In der Abbildung ist in blau beispielhaft ein Sensor-Footprint eingezeichnet und überlagert mit dem Höhendatenraster. Die weißen Kreise zeigen die Abtastungspunkte an, um die Höhenmatrix zu bestimmen.

Neben der Höhenmatrix *H* wird auch die Abtastungsmatrix *C* erzeugt. Sie beinhaltet die Information, ob der Abtastungspunkt innerhalb oder außerhalb des Sensor-Footprints liegt. Für das Beispiel aus Bild 4.11 ergibt sich die folgende Höhenmatrix und Abtastungsmatrix:

Für die Interpretation der Oberflächenbeschaffenheit spielt nicht der absolute Höhenwert eine Rolle, sondern der Gradient, also die räumliche Änderung bzw. erste Ableitung des Höhenwertes. Die Magnitude der Gradienten wird durch die Faltung der Höhenmatrix mit dem Sobel-Operator [132] approximiert. Dabei wird separat in x- und y-Richtung abgeleitet, wodurch die beiden gefalteten Resultate G_x und G_y entstehen. Die richtungsunabhängige Magnitude der Höhengradienten berechnet sich zu

$$M = \sqrt{G_x^2 + G_y^2}.$$
 (4.9)

Im Anschluss kann der normierte Reliefwert r' berechnet werden, welcher aus der Summe der Magnituden besteht. Dabei werden nur jene Elemente betrachtet, die auch innerhalb des Sensor-Footprints liegen, bestimmt durch die Abtastungsmatrix *C*. Damit die Reliefwerte vergleichbar sind, werden diese durch die Fläche des Sensor-Footprints geteilt. Folgende Formel gibt die Berechnungsvorschrift des normierten Reliefwertes an:

$$r = \frac{\sum_{i,j} M(i,j)C(i,j)}{A} \qquad r' = \tanh(f_r \cdot r) \tag{4.10}$$

Hierbei ist *A* die Fläche des Sensor-Footprints und f_r ein Skalierungsfaktor für den Tangens hyperbolicus. Bestimmt wird f_r , analog zum Skalierungsfaktor für die Bodenauflösung, durch den Kehrwert des maximal auftretenden Reliefwertes des *Modelldatensatzes*.

Da der Reliefwert für die Expertenmodellierung der Leistungsmodelle nicht intuitiv ist, wird für diesen der Begriff der *Geländeform* verwendet. Dabei sollen 5 Beschreibungen verwendet werden, die den entsprechenden Reliefwertebereichen zugeordnet sind:

- Flach: $0 \le r' < 0,1$
- Geneigt: $0,1 \le r' < 0,3$
- Hügelig: $0,3 \le r' < 0,5$
- Bergig: $0.5 \le r' < 0.7$
- Schroff: $0,7 \le r' \le 1$

4.3 Zusammenfassung der Umgebungszustände

Tabelle 4.1 zeigt eine Zusammenfassung der vorgestellten Umgebungszustände mit ihren Symbolen und Wertebereichen. Hierbei ist anzumerken, dass die Summe der Bodenbedeckungskategorien Wasser, Ebene, Gebäude, Straßen und Vegetation immer 1 ergibt. Bei dem Elevationswinkel ist ein theoretischer Wertebereich zwischen -1 und 1 angegeben, welcher aber für die hier betrachtete Anwendung der erdgerichteten Fahrzeugerkennung nur zwischen -1 und 0 relevant ist.

	Umgebungszustände	Symbol	Wertebereich
phärisch	Niederschlagsmenge	Q'_r	[0; 1]
	Nebeldichte	$ ho_f'$	[0; 1]
	Bewölkungsgrad	Wo	[0; 1]
	Tageszeit X	x_t	[-1;1]
som	Tageszeit Y	y_t	[-1;1]
atı	Monat X	x_M	[-1;1]
	Monat Y	\mathcal{Y}_M	[-1;1]
ph.	Elevationswinkel	θ'	[-1;1]
	Bodenpixelauflösung	g'	[0; 1]
topographisch	Wasser	t_w	[0; 1]
	Ebene	t_{os}	[0; 1]
	Gebäude	t_b	[0; 1]
	Straße	t_s	[0; 1]
	Vegetation	t_v	[0; 1]
	Reliefwert	r'	[0; 1]

Tabelle 4.1: Zusammenfassung der Umgebungszustände mit Symbol und Wertebereich.

4.4 Generierung des Modelldatensatzes

Für die Erzeugung der Metadaten und letztlich der Leistungsmodelle durch ML-Verfahren ist ein Modelldatensatz notwendig, der in seiner Variabilität den Parameterraum der Umgebungszustände soweit abdeckt, dass den Verfahren zur Funktionsapproximation (NEFPROX und KNN) genügend Beispiele zur Verfügung stehen.

Der Erzeugung des Modelldatensatzes, bestehend aus Bild-Annotationen-Paaren \tilde{I}, \tilde{A} und Umgebungszustand f_x , liegen sieben geotypische 3D-Landschaften ("Karten") zu Grunde. In ihrer Gesamtheit bieten sie die notwendige Variabilität hinsichtlich der topographischen Umgebungszustände. Die Karten sind quadratisch in ihren lateralen Abmessungen und beinhalten verschiedene Landschaftstypen. Im Folgenden werden sie kurz vorgestellt:

- *Samawah* ist eine Nachbildung der realen Stadt As-Samawah im Irak und geprägt durch orientalische Gebäude, Steppen- und Wüstenregionen.
- In der Karte *Bagdhdad Greenzone* sind hauptsächlich Wüstenregionen und eine Nachbildung eines Teiles der Stadt Bagdhdad vorzufinden.
- In *Prison* finden sich einige Gebäude, Straße und eine Gefängnisanlage. Die Landschaft zeigt gemäßigtes Klima.
- Die Karte *Rahmadi* beinhaltet eine kleine Insel mit einer Landebahn, einem Dorf und einem kleinen Hafen. Das Klima ist mediterran.
- *Sahrani* ist eine geotypische Karte, welche eine Hauptinsel und mehrere kleinere Inselgruppen zeigt. Insgesamt weist die Karte im Süden eine mediterrane und im Norden eine europäische Landschaft auf. Außerdem befindet sich eine Bergkette im Norden der Hauptinsel.
- *Takistan* besteht hauptsächlich aus schroffen Bergketten und Wüsten. Außerdem zeigt die Karte einige Täler mit geotypischen Strukturen.
- Die Karte *Warminster* ist eine Nachbildung der Region Warminster in England. Dort findet sich eine Stadt mit gemäßigtem Klima.

Damit die Leistungsmodelle den Situationsraum (vgl. Abschnitt 3.3.3.4) ausreichend abdecken können, muss der Modelldatensatz eine ausreichende Varianz besitzen. Dazu werden

verschiedene Datensätze unter Verwendung eines Generierungsparametersatzes erzeugt. Dieser sieht folgende Parametervariationen vor:

- Der Monat, in dem die Daten aufgenommen werden, im Wertebereich 1 12.
- Die Tageszeit im Wertebereich 0 23 Uhr.
- Der Elevationswinkel θ der Kamera im Wertebereich von -30 bis -90 Grad.
- Die Distanz d_s definiert den Abstand zwischen Kamera und dem Schnittpunkt der optischen Achse mit der Bodenoberfläche. Der Wertebereich liegt zwischen 20 und 100 Metern.
- Der normierte Bewölkungsgrad von 0 1.
- Die normierte Niederschlagsmenge von 0 1.
- Die normierte Nebeldichte von 0 1.
- Der Sensortyp, der wahlweise VIS- oder LWIR-Sensor sein kann.
- Die Zellenbreite d_c , die verwendet wird, um auf einer Karte Abtastpunkte zu definieren. Damit wird die Anzahl der Bild- und Annotationsdaten limitiert.

Die Generierung eines Datensatzes erfolgt in drei Schritten. Zunächst werden an zufälligen Positionen auf der Karte N Objekte des Typs Pkw platziert, welche zufällig aus einer Objektdatenbank mit 100 Pkw-Modellen ausgewählt werden. Im zweiten Schritt wird die Karte durch ein Gitternetz in Segmente unterteilt, wie in Bild 4.12 schematisch dargestellt. Dabei deutet das äußere Rechteck den Kartenrand an, die blauen Linien bilden das Gitternetz und die weißen Rauten symbolisieren die platzierten Fahrzeuge. Das Gitternetz wird variabel erzeugt, basierend auf der Distanz d_s und der Kartenbreite W und -länge L. Damit es zu keinen Randeffekten durch die Kartenbegrenzung kommt (z. B. endende Bodentexturen oder abgeschnittene Objekte), wird ein Abstand d_b vom Kartenrand zum Gitternetz eingehalten, der zu $2 \cdot d_s$ definiert wird. Die Zellenbreite d_c wird berechnet mit

$$d_c = \max\left(d_b + U(0, d_b), \frac{\max(L, W)}{100 \, m}\right). \tag{4.11}$$

Dabei ist U(a, b) eine gleichverteilte Zufallsvariable im Intervall [a, b]. An jeder Kreuzung der Gitternetzlinien werden folgende Daten aufgenommen:

- Das Sensorbild und die Objektannotationen (siehe Abschnitt 4.1)
- Die Umgebungszustände, erfasst nach den Berechnungsvorschriften aus Abschnitt 4.2

In Bild 4.12 deuten die schwarzen Pfeile bereits abgetastete Position an. Die grauen Pfeile zeigen die nächsten Abtastpunkte. Hierbei entspricht der Kreuzungspunkt dem Bildmittelpunkt. Außerdem wird der Azimutwinkel ψ der Kamera auf die Szene an jedem Kreuzungspunkt zufällig neu ermittelt, um verschiedene Perspektiven aufzunehmen. Durch dieses Verfahren werden hauptsächlich Bilddaten ohne Objekte aufgenommen. Im dritten und letzten Schritt werden alle Objekte nacheinander als Bildmittelpunkt ausgewählt. Dabei wird die Kamera bei jedem Objekt um dieses bewegt, durch Veränderung des Azimutwinkels in 10-Grad-Schritten, und es werden jeweils die zuvor angegebenen Daten erhoben. Dadurch werden Bilddaten mit verschiedenen Objektperspektiven erzeugt. Dargestellt ist der dritte Schritt in Bild 4.13.



Bild 4.12: Schematische Darstellung der Generierung eines Datensatzes durch das Gitternetz.



Bild 4.13: Schematische Darstellung der Generierung eines Datensatzes durch die Objektpositionen.

Für die Generierung des Modelldatensatzes werden für jede Karte 10 festgelegte und 50 randomisierte Generierungsparametersätze verwendet. Die 10 festen Generierungsparametersätze pro Karte dienen zum einen der Erzeugung des Objektdatensatzes (siehe Abschnitt 4.5.2.5) für das Training der Objektdetektoren und zum anderen als zusätzliche Daten für den Modelldatensatz. In Tabelle 4.2 sind die festgelegten Parameterwerte zu sehen.

Karte	Elevationswinkel	<i>d_c</i> in Meter	<i>d_s</i> in Meter	<i>d_b</i> in Meter
Prison	[30, 45, 60, 75, 90]	100	[25, 50]	100
Rahmadi, Warminster	[30, 45, 60, 75, 90]	200	[25, 50]	200
Sahrani, Takistan	[30, 45, 60, 75, 90]	300	[25, 50]	300
Bagdhdad Greenzone	[30, 45, 60, 75, 90]	400	[25, 50]	400
Samawah	[30, 45, 60, 75, 90]	700	[25, 50]	700

Tabelle 4.2: Festgelegte Generierungsparametersätze. Durch alle Kombinationen der mehrfachen Parameterwerte entstehen die 10 festgelegten Parametersätze pro Karte.

Dabei gelten für diese Generierungsparametersätze, dass:

- der Monat Juni,
- die Tageszeit 15:00 Uhr,
- der normierte Bewölkungsgrad 0,
- die normierte Nebeldichte 0,

- die normierte Niederschlagsmenge 0,
- und der Sensortyp ein visueller Sensor ist.

Die randomisierten Generierungsparametersätze sollen die nötige Varianz der Daten garantieren, damit die Leistungsmodelle im gesamten Parameterraum anwendbar sind. Dafür wird jeder Parameter im Parametersatz durch eine gleichverteilte Zufallsvariable bestimmt. Die Intervalle der entsprechenden Zufallsvariable sind in der Liste der Generierungsparameter oben angegeben. Anschließend erfolgt der oben beschriebene Generierungsprozess. Damit entstehen insgesamt 420 Datensätze, 70 durch festgelegte und 350 durch randomisierte Generierungsparametersätze. Insgesamt enthält der Modelldatensatz damit etwa 2 Millionen Bild-Annotationen-Paare mit entsprechenden Umgebungszuständen.

4.5 Perzeptionsmodule zur Fahrzeugerkennung

Dieser Abschnitt beschreibt die einzelnen Perzeptionsmodule aus dem verwendeten Perzeptionsgraphen (siehe Abschnitt 3.3.2) zur Fahrzeugerkennung. Zur Erzeugung der Leistungsmodelle prozessieren diese den Modelldatensatz, wobei ihr *Leistungsindex* (siehe Abschnitt 4.6) gemessen wird. Leistungsindex und Umgebungszustände werden anschließend als *Metadaten* zusammengeführt.

4.5.1 Segmentierung und Hypothesengenerierung

Bei der Segmentierung und Hypothesengenerierung werden Bereiche von Interesse aus dem Bild segmentiert und anschließend markante Punkte als potenzielle Kandidaten (Hypothesen) extrahiert, die dann durch einen Klassifikator überprüft und ggf. einer Objektklasse zugeordnet werden. In dieser Arbeit ist dies die Objektklasse Fahrzeuge bzw. Nicht-Fahrzeuge. In dieser Arbeit wurden vier Algorithmen zur Segmentierung und Hypothesengenerierung implementiert, wie aus dem in Bild 3.7 gezeigten Perzeptionsgraphen ersichtlich sind, und im Folgenden vorgestellt.

4.5.1.1 Straßensegmentierung

Die Segmentierung von Straßen spielt vor allem eine Rolle bei der Erkennung von zivilen Fahrzeugen. Diese sind in der Regel meist auf Straßen bzw. Asphaltflächen zu finden. Daher lässt sich der Suchraum auf solche beschränken, um so die Zahl an Fehldetektionen zu reduzieren. Über die topographischen Umgebungszustände lässt sich ermitteln, ob Straßen im Bild zu sehen sind und damit die Straßensegmentierung sinnvoll eingesetzt werden kann.

Die Methode zur Straßensegmentierung basiert auf der Analyse der Farbausprägungen von Asphaltflächen und der linienartigen Struktur bzw. ausgeprägten Abgrenzung zur Umgebung von Straßen. Dabei können Rauschen und schwache Gradienten im Bild zu Problemen bei der Konturfindung der Straßensegmente führen, weshalb zunächst das Eingangsbild mit einem Mittelwertfilter gefaltet wird. Dabei sieht der Filter für eine Größe von 3×3 wie folgt aus:

$$f_{mean} = \frac{1}{3 \cdot 3} \begin{bmatrix} 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1 \end{bmatrix}$$
(4.12)

Verwendet wird eine Filtergröße von 9 × 9. Anschließend wird das geglättete Bild vom RGB-Farbraum in den HSV-Farbraum [132] transformiert. Dadurch lassen sich Helligkeit H, Sättigung S und Farbwert V trennen und eine Segmentierung anhand eines Farbbereiches durchführen. In Bild 4.14a zeigt ein Beispielbild, welches durch Glättung und Farbsegmentierung in das Binärbild in Bild 4.14b überführt wurde. Dabei wurden empirisch die minimalen und maximalen Schwellwerte zur Segmentierung von Asphaltflächen zu

$$\begin{aligned} H_{min} &= 0\% & S_{min} = 0\% & V_{min} = 60^{\circ} \\ H_{max} &= 100\% & S_{max} = 27\% & V_{max} = 300^{\circ} \end{aligned}$$
 (4.13)

ermittelt. Der Hintergrund der Limitierung der Sättigung ist, dass gräuliche Farben eine sehr geringe Sättigung aufweisen. Die Einschränkung des Farbwertes verhindert, dass weiße und schwarze Elemente mitsegmentiert werden.







Bild 4.14: Verarbeitungsschritte der Straßensegmentierung an einem Beispiel. In (a) ist das Eingangsbild zu sehen, welches geglättet wird, (b) zeigt das Binärbild nach der Anwendung der Schwellwerte, in (c) ist das Binärbild nach den morphologischen Operationen abgebildet und in (d) ist die resultierende Segmentierungsmaske nach der Konturlinienanalyse dargestellt.

Durch Fahrzeuge oder andere Objekte auf den Straßen können Löcher im Binärbild entstehen. Diese werden durch die morphologischen Operationen Dilatation und Erosion geschlossen. Dabei vergrößert die Dilatation zunächst alle Elemente im Binärbild, um die Löcher zu schließen. Anschließend werden die vergrößerten Elemente durch die Erosion wieder geschlossene Löcher weiterhin geschlossen bleiben. verkleinert. wobei Dieser Verarbeitungsschritt wird auch als Schließen bezeichnet. Die Dilatation bildet das Maximum und die Erosion das Minimum aller Nachbarelemente:

$$I_d(x,y) = \max_{\substack{(x',y'):S(x',y')\neq 0}} I(x+x',y+y')$$

$$I_e(x,y) = \min_{\substack{(x',y'):S(x',y')\neq 0}} I(x+x',y+y')$$
(4.14)

Dabei ist I(x, y) das Eingangsbild an der Elementposition (x, y) und S(x', y') das Strukturelement, welches die umliegenden auszuwertenden Elementpositionen definiert. Dies bedeutet, dass für die Dilatation alle durch das Strukturelement definierten Elementpositionen der maximale Wert ermittelt wird und in das Ausgangsbild geschrieben wird. Das Strukturelement ist typischerweise eine Matrix mit $k \times k$ Elementen, die mit dem Wert Eins besetzt sind und in der der Anker mittig sitzt. Das Ergebnis der morphologischen Operationen ist in Bild 4.14c abgebildet, erzeugt mit einem 5 × 5 Strukturelement. Zuletzt wird das Verfahren von Suzuki [133] angewendet, um die äußeren Konturlinien zu finden. Dabei werden Konturlinien mit einer Länge in Pixeln (Umfang des Objekts) kleiner als das Maximum aus Bildbreite und -höhe entfernt. In Bild 4.14d ist die Segmentierungsmaske der Straßensegmenterkennung dargestellt, welche durch Füllen der verbleibenden Konturen entsteht.

4.5.1.2 Vordergrundsegmentierung

Ein weiteres verwendetes Verfahren zur Segmentierung ist das Schwellwertverfahren nach Otsu [134]. Generell ist diese Methode geeignet für eine Szene, in der zwei unterschiedlich illuminierte Bereiche dominieren. Gerade bei sehr flachen Elevationswinkeln des Sensors bietet das Otsu-Verfahren einen einfachen Weg, zwischen der Himmelregion und der Bodenoberfläche zu unterscheiden. Außerdem führt Schattenwurf dazu, dass jene Regionen, in denen sich Objekte befinden, tendenziell auch dunkler sind als die übrige Szene. Anzumerken ist, dass es Probleme gibt, sobald mehr als zwei dominierende Helligkeitsbereiche in der Szene vorkommen.

Der Schwellwert zur Segmentierung wird beim Otsu-Verfahren durch die Ermittlung des Minimums der Varianzen der beiden Klassen bestimmt. Dazu wird zunächst das Histogramm des Eingangsbildes berechnet, wobei bei mehrkanaligen Bildern jeder Kanal separat behandelt wird. Für das Beispielbild in Bild 4.15a ergibt sich das Histogramm, welches in Bild 4.16 dargestellt ist. Im Histogramm sind die Schwellwerte der einzelnen Farbkanäle durch senkrechte gestrichelte Linien dargestellt. Dabei wird deutlich, dass jeder Farbkanal zwei Maxima besitzt mit entsprechender Varianz, die durch den Schwellwert minimiert wird. In diesem Fall sind alle Intensitätswerte links des Schwellwertes als Vordergrund zu deuten und alle rechts davon als Hintergrund. Durch Anwendung der Schwellwerte auf die einzelnen Kanäle und mit einer Vereinigung der resultierenden Binärbilder mit anschließender Bearbeitung durch die morphologische Operation Schließen entsteht die Segmentierungsmaske, wie in Bild 4.15b dargestellt. Das Strukturelement der Schließen-Operation hat die Größe 6×6 .



Bild 4.15: Segmentierung mit dem Otsu-Verfahren anhand eines Beispiels. In (a) ist das Eingangsbild und in (b) die resultierende Segmentierungsmaske zu sehen.



Bild 4.16: Darstellung der Otsu Schwellwertbestimmung. Das gezeigte Farbhistogramm resultiert aus dem Beispielbild in Bild 4.15a.

4.5.1.3 Hypothesengenerierung mit Eckpunktdetektion

An die beiden Segmentierungsverfahren schließt sich der Hypothesengenerator mittels Eckpunktdetektion an. Bei der Eckpunktdetektion werden Punkte im Bild gesucht, die im Wesentlichen in ihrer lokalen Nachbarschaft zwei ausgeprägte Kantenrichtungen aufweisen. Gerade technische Objekte, wie z. B. Fahrzeuge, besitzen ausgeprägte Eckpunkte, die als Objekthypothese genutzt werden können. Allerdings besitzen auch Gebäude und Teile der Vegetation Eckpunkte, die zu Falschhypothesen führen können. Durch die beiden vorgeschalteten Segmentierungsschritte wird die Anzahl der Falschhypothesen bereits reduziert.

In dieser Arbeit wird der Harris-Corner-Detektor [135] verwendet. Dieser berechnet einen Wert für jede Pixelposition im Bild, wobei immer eine lokale Nachbarschaft betrachtet wird. Dabei gibt der Wert Auskunft darüber, wie ausgeprägt der Eckpunkt ist. Im Algorithmus lässt sich zum einen der Bereich der lokalen Nachbarschaft dimensionieren sowie ein freier Parameter λ vorgeben, der die Sensitivität reguliert. Typischerweise wird λ empirisch ermittelt und liegt zwischen 0,04 und 0,15. Dabei führen kleinere Werte zu einer höheren Sensitivität, was bedeutet, dass auch Eckpunkte mit kleineren Gradienten einen höheren Wert erhalten und damit mehr Eckpunkte gefunden werden. Verwendet wurde eine Nachbarschaft von 9×9 und λ wurde zu 0,1 bestimmt. Die resultierende Wertematrix wird durch einen Schwellwert t_h in ein Binärbild konvertiert, mit

$$t_h = \frac{10}{W \cdot H} \tag{4.15}$$

Dabei entsprechen *W* und *H* der Höhe und Breite des Bildes in Pixeln. Anschließend wird die Schnittmenge aus der vorherigen Segmentierungsmaske (Straßen- oder Vordergrundsegmentierung) und dem hier erzeugten Binärbild gebildet. Um Bereiche mit vielen Eckpunkten zu verbinden, wird das vorherige erhaltene Binärbild mit einem Mittelwertfilter weichgezeichnet, damit Überlappungen entstehen. Dabei wird die Größe des Mittelwertfilters ermittelt indem die mittlere Bodenpixelauflösung durch die mittlere Objektgröße geteilt wird. Das bedeutet, dass das Binärbild umso weniger weichgezeichnet wird, je kleiner das Objekt im Bild erscheint. Dies resultiert daraus, dass die Eckpunkte umso näher zusammenliegen, je weiter die Szene vom Sensor entfernt ist. Anschließend wird durch den Schwellwert t_c wieder ein Binärbild erzeugt, wobei der Schwellwert wie folgt berechnet wird:

$$t_c = \left[\frac{g \cdot 255 \, px}{o_z} + \frac{1}{2}\right] \tag{4.16}$$

Die Variable o_z ist die mittlere Objektgröße des Objekts von Interesse und g die mittlere Bodenpixelauflösung. Durch die abrundende Gaußklammer und Addition mit 1/2 wird der Schwellwert auf ganze Zahlen gerundet. Damit passt er sich dem vorher verwendeten Mittelwertfilter an, was bedeutet, dass für kleiner werdende Mittelwertfiltergrößen ein immer größerer Schwellwert berechnet wird. Dies dient dazu, dass lediglich überlappende Bereiche auch Teil der neuen Binärmaske werden. In dieser Arbeit wurde die mittlere Objektgröße von Fahrzeugen mit 4,5 m festgelegt.

Im letzten Schritt werden die Konturen mithilfe des Algorithmus von Suzuki [133] aus dem Binärbild ermittelt. Aus den Konturen werden rechteckige Bereiche von Interesse (engl. *Region of Interest*, ROI) gebildet, die als Hypothesen für die Objektklassifikation dienen. Ein Beispiel ist in Bild 4.17 zu sehen. Dabei ist in Bild 4.17a das Eingangsbild und in Bild 4.17b die Segmentierungsmaske, gebildet durch die Methode von Otsu, zu sehen. Die Binärmaske nach der Mittelwertfilterung ist in Bild 4.17c abgebildet und die resultierenden ROIs in Bild 4.17d.



Bild 4.17: Beispiel der Hypothesengenerierung mittels Eckendetektor. Das Eingangsbild (a) und die Segmentierungsmaske (b) sind die Eingangsdaten der Hypothesengenerierung, die das Binärbild (c) erzeugt und daraus die Hypothesen (d) bildet.

4.5.1.4 Farbraumanalyse

Neben den beiden Segmentierungsalgorithmen und nachfolgender Hypothesengenerierung steht auch ein Algorithmus auf Basis einer Farbraumanalyse zur Hypothesengenerierung zur Verfügung. Dabei wird die Verteilung der Fahrzeugfarben herangezogen, um diese in einem Bild zu segmentieren und auf ihre geometrischen Eigenschaften hin zu analysieren. Aus den Ergebnissen einiger Studien [136–138] setzt sich die Verteilung der Fahrzeugfarben weltweit

im Mittel aus 24% Weiß, 19% Schwarz, 16% Silber, 14% Grau, 8% Rot, 8% Blau, 6% Braun und zu 5% aus anderen Farben zusammen. Damit sind über 70 % der Fahrzeugfarben Abstufungen von Grautönen (Weiß, Schwarz, Silber und Grau) mit entsprechenden Varianzen vorzufinden. Daher werden in der folgenden Farbraumanalyse Filter für Weiß, Schwarz, Silber/Grau, Rot, Blau, Braun und zusätzlich Grün eingesetzt.

Damit die zu segmentierenden Farbflächen möglichst homogen sind, wird das Eingangsbild mit einem Medianfilter [139] gefaltet. Dieser behält die Farbübergänge bzw. Kanten, glättet aber zugleich kleine Farbvarianzen. Der Medianfilter sortiert in der lokalen Nachbarschaft, definiert durch die Filtergröße, alle Intensitätswerte und wählt den Mittleren aus. Die Filtergröße ist hierbei 3 × 3. Anschließend findet eine Farbraumkonvertierung von RGB zu HSV statt, um im HSV-Farbraum die Farbfilter anzuwenden. Die Farbfilter wurden durch empirisches Ermitteln und Analyse der erzeugten Bilddaten aus der Sensorsimulation (siehe Abschnitt Kapitel 4) definiert und ihre Parameter sind in Tabelle 4.3 zu finden. Jeder Farbfilter erzeugt eine Binärmaske, wobei die Intensitätswerte, die innerhalb des jeweiligen Minimums bzw. Maximums liegen, markiert werden. Alle Binärmasken werden durch Vereinigung zusammengeführt. Im Anschluss daran wird eine BLOB-Analyse durchgeführt, welche in Abschnitt 4.5.2.2 erläutert wurde. Dabei werden die folgenden Parameter verwendet:

$$\sqrt{\frac{A_o^{min}}{g^2}} < A_B < \sqrt{\frac{A_o^{max}}{g^2}} \quad 0.5 < Co_B < 1$$

$$0.2 < I_B < 0.8 \quad 0.1 < Ci_B < 0.5$$
(4.17)

 A_B ist die Fläche des BLOBs mit $A_o^{min} = 6 m^2$ und $A_o^{max} = 17 m^2$. Die mittlere Bodenpixelauflösung ist mit *g* gekennzeichnet. Co_B ist die Konvexität, Ci_B die Kreisförmigkeit und I_B das Verhältnis zwischen Haupt- zu Nebenachse. In Bild 4.18 ist ein Beispiel der Hypothesengenerierung mittels der Farbraumanalyse zu sehen.

		Weiß	Schwarz	Silber/Grau	Rot	Blau	Braun	Grün
Farbwert	Min	0	0	115	0	85	19	60
	Max	250	172	209	27	190	31	138
Sättigung	Min	0	0	13	50	12	10	17
	Max	8	20	80	100	45	33	51
Hellwert	Min	90	0	0	23	63	24	25
	Max	100	20	60	100	100	53	38

Tabelle 4.3: Parameter der Farbfilter zur Segmentierung von Fahrzeugen. Dabei ist der Farbwert (engl. *hue*) von 0° bis 360°, die Sättigung (engl. *saturation*) vom 0% bis 100% und der Hellwert (engl. *value*) von 0% bis 100% angegeben.



Bild 4.18: Beispiel der Hypothesengenerierung mittels Farbraumanalyse. Das Bild in (a) ist das Beispieleingangsbild und (b) zeigt die extrahierten Hypothesen durch die Farbraumanalyse.

4.5.2 Objektdetektoren

Bei der Fahrzeug- bzw. Objektdetektion handelt es sich um ein Klassifizierungsproblem, bei dem Instanzen, bzw. Hypothesen, einer Objektklasse zugeordnet und lokalisiert werden. Die verwendeten Klassifikatoren sind binäre Klassifikatoren, die zwischen zwei Klassen entscheiden können: Der Objektklasse "Fahrzeuge" L_1 und der Objektklasse "Nichtfahrzeuge" L_0 . Im Folgenden werden die vier Objekterkennungsverfahren (siehe Übersicht in Bild 3.7), welcher zur Fahrzeugdetektion eingesetzt werden, nach ihren Grundsätzen vorgestellt und Implementierungsaspekte vorgestellt. Dabei liefern alle Verfahren einen Vertrauenswert *e*, der beschreibt, wie *sicher* sich der Klassifikator ist, dass es sich um die erkannte Objektklasse "Fahrzeuge" handelt. Hierbei lässt sich mittels Evaluierungsdaten der optimale Bias *b* finden, der als Schwellwert zwischen den beiden Klassen entscheidet:

$$e - b > 0 \stackrel{\circ}{=} L_1 \tag{4.18}$$

In Abschnitt 8.2.4 werden die Ergebnisse der optimalen Bias Werte der vier Objektdetektoren beschrieben, welche mittels Receiver-Operating-Characteristic-Kurve (ROC-Kurve) ermittelt werden. Die Beschreibung und Erstellung einer ROC-Kurve befindet sich in Abschnitt 5.2.1.

4.5.2.1 Kantenbasiertes Template Matching

Beim Template Matching wird ein Template mit einem Eingangsbild verglichen, um die Ähnlichkeit zwischen diesen zu berechnen. Auf Basis der Ähnlichkeit wird schließlich eine Objektidentifizierung durchgeführt. Einfache Verfahren vergleichen die Pixelintensitäten und nutzen deren Abweichung, um die Ähnlichkeit zu berechnen. Allerdings sind diese Verfahren anfällig gegenüber Helligkeitsunterschieden, Farbvarianzen, Skalierung und Rotation. Daher wird in dieser Arbeit ein Template Matching auf Basis von Kantenstärken und -richtungen verwendet [140], das invariant gegenüber Farbvarianzen und Helligkeitsunterschiede zu ist. durch Zusätzlich werden mehrere Templates verwendet, um Fehldetektionen Objektrotationen zu kompensieren. Des Weiteren kann die Skalierung des Bildes durch die mittlere Bodenpixelauflösung an die des Template angepasst werden. Der Vorteil des Template Matchings ist die einfache Austauschbarkeit der Templates, um andere Objekte erkennen zu können. Allerdings ist das kantenbasierte Template Matching im Vergleich mit den folgenden Verfahren zur Objektdetektion (siehe Abschnitte 4.5.2.2 bis 4.5.2.4) am rechenaufwendigsten.

Für jedes Template *T*, welches ein Bild des zu erkennenden Objektes darstellt, wird zunächst ein Templatemodell T_M erzeugt. Dabei ist es das Ziel, die signifikantesten Kanten des Objekts zu extrahieren und diese später beim eigentlichen Template Matching zu verwenden, um so das Ähnlichkeitsmaß zu berechnen. Mit dem Sobel-Operator [132] wird die Ableitung in x- und y-Richtung von *T* approximiert, wobei die Filter des Sobel-Operators die Größe 3 × 3 besitzen. Anschließend kann die Stärke der Gradienten G^T und die Orientierung der Gradienten Θ^T bestimmt werden mit

$$G^T = \sqrt{G_x^2 + G_y^2}$$
 und $\Theta^T = \tan^{-1}\left(\frac{G_y}{G_x}\right)$. (4.19)

Dabei ist G_x die Ableitung in x-Richtung und G_y die Ableitung in y-Richtung des Template. Außerdem ist G^T normiert auf den Wertebereich [0, 1]. Um die signifikantesten Kanten im Modell zu beizubehalten, wird eine *Non-Maximum-Suppression* (NMS) durchgeführt. Dies bedeutet, dass in lokaler Nachbarschaft nur die stärkste Kante entlang der Orientierung behalten wird. Betrachtet wird eine lokale Nachbarschaft von einem Element, um den zu bewertenden Gradienten herum. Damit lassen sich die Kantenorientierungen in vier Richtungen eingliedern, und es kann Θ^T diskretisiert werden. In Bild 4.19a ist die graphische Visualisierung der Diskretisierung zu sehen. Dabei verlaufen Kanten in der Region I horizontal, in der Region II entlang der Hauptdiagonalen, in der Region III entlang der Nebendiagonalen und in der Region IV vertikal. Hierbei muss lediglich der obere Halbkreis betrachtet werden, da der Arkustangens zur Bestimmung der Orientierung verwendet wird und damit die Winkel des unteren Halbkreises auf die des oberen projiziert werden. Damit lassen sich die Orientierungen in vier Abstufungen einteilen, mit der Formel in Bild 4.19b.



Bild 4.19: Diskretisierung der Kantenorientierung zur Bestimmung des Template Matching Modells.

Anschließend wird für jedes Element in G^T ausgewertet, welche diskrete Orientierung θ der Gradient besitzt. Je nach Orientierung werden die entsprechenden lokalen Nachbarelemente betrachtet, um eine NMS durchzuführen. Im Folgenden sind die vier Filter für die lokale Nachbarschaftsanalyse aufgeführt:

$$F_{H} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad F_{V} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad F_{D1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad F_{D2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$
(4.20)

Dabei wird der Filter F_H bei horizontaler, F_V bei vertikaler, F_{D1} in hauptdiagonaler und F_{D2} in nebendiagonaler Orientierung verwendet. Alle Filter haben dabei ihren Anker in der Mitte. Wenn alle Nachbarelemente, definiert durch den entsprechenden Filter, einen kleineren Gradienten als den zu betrachtenden Wert aufweisen, wird der Gradient behalten. Ansonsten wird der zu betrachtende Wert auf 0 gesetzt und damit vom Modell ausgeschlossen. Die damit erzeugte dünn besetzte Matrix G_{NMS}^T wird im Anschluss durch *Minimum-Maximum-Schwellwertfilterung* in eine binäre Matrix $M^T(x, y)$ transformiert. Diese Matrix hat das Ziel, nur Gebiete mit starken Gradienten zu markieren, um letztendlich das Modell für ein Template zu erzeugen. Bei der Minimum-Maximum-Schwellwertfilterung wird jedes Element in $G_{NMS}^T(x, y)$ überprüft,

• ob der Wert des Elements kleiner als ein Schwellwert t_{min} ist, dann wird in $M^T(x, y)$ das entsprechende Element auf 0 gesetzt,

- ob der Wert des Elements größer als t_{max} ist, dann wird in $M^T(x, y)$ das entsprechende Element auf 1 gesetzt,
- ansonsten wird überprüft ob es in einer Ein-Element-Nachbarschaft einen Wert größer als t_{max} gibt, dann ist $M^T(x, y) = 1$ sonst $M^T(x, y) = 0$.

Der minimale und maximale Schwellwert ist hier mit 0,05 und 0,3 gewählt. Schließlich kann das Kantenmodell durch Multiplikation der generierten Binärmaske mit den originalen Stärken und Orientierungen der Gradienten des Template gebildet werden:

$$G_{x}^{M} = G_{x} \cdot M^{T}$$

$$G_{y}^{M} = G_{y} \cdot M^{T}$$

$$G^{M} = G^{T} \cdot M^{T}$$
(4.21)

Die verwendeten Templates sind in Bild 4.20 abgebildet und zeigen zusätzlich die Gradientenstärken, welche in das entsprechende Modell übernommen wurden. An diesen Templates ist bereits ersichtlich, dass Schattenwurf zu Fehlern in den Modellen führen kann und schließlich beim eigentlich Template Matching Prozess zu Fehlklassifikationen führt.



Bild 4.20: Verwendete Templates und daraus generierte Gradientenmodelle für das kantenbasierte Template Matching. In der oberen Zeile sind die Templates zu sehen und darunter, in der zweiten Zeile, die Gradientenmodelle G^M mit Schwerpunkt, markiert durch einen weißen Kreis.

Der Template Matching Prozess startet mit der Berechnung der Richtung der Gradienten mit dem Sobel-Operator und der Bestimmung der Stärken der Gradienten des Eingangsbildes. Dabei sei die Ableitung in x-Richtung G_x^I , die Ableitung in y-Richtung G_y^I und die Stärke der Gradienten G^I . Für jedes Template-Modell und jede Position im Bild lässt sich ein Bewertungsmaß *S* berechnen, der die Ähnlichkeit zwischen Modell und entsprechendem Bildausschnitt widerspiegelt. Das Bewertungsmaß wird berechnet zu

$$S(x,y) = \frac{1}{W \cdot H} \sum_{u=1}^{W} \sum_{v=1}^{H} \frac{G_x^I(x+u,y+v) \cdot G_x^M(u,v) + G_y^I(x+u,y+v) \cdot G_y^M(u,v)}{G^I(x+u,y+v) \cdot G^M(u,v)}, \quad (4.22)$$

dabei ist W und H die Breite und Höhe des Template, (x, y) die Position im Eingangsbild und (u, v) die Position im Template. Durch Normierung der Bewertungsmatrix auf den Wertebereich [0, 1] und Binarisierung durch den Schwellwert $t_s = 0,7$ erhält man das Binärbild S_B , welches für das Eingangsbild in Bild 4.21a beispielhaft in Bild 4.21c dargestellt ist.

Durch Konturenanalyse lassen sich die äußeren Konturen im Binärbild finden. In jeder gefundenen Kontur werden alle Werte in der Bewertungsmatrix *S*, die sich innerhalb der Kontur befinden, aufsummiert. In Bild 4.21b ist eine Bewertungsmatrix zu sehen, welche mit dem Eingangsbild aus Bild 4.21a und dem Template in Bild 4.20e erzeugt wurde. Dieser aufsummierte Wert dient als Evidenz *e* des kantenbasierten Template Matchings zur Fahrzeugerkennung. Je nach Anzahl der Templates kann es zu Mehrfacherkennungen kommen, weshalb anschließend eine NMS durchgeführt wird. Dabei werden lediglich Erkennungen behalten, wenn sich mindestens zwei solcher überlappen, wobei die Evidenzwerte gemittelt werden.

In Bild 4.21d ist die Objekterkennung auf dem Beispielbild Bild 4.21a nach der finalen NMS abgebildet.



Bild 4.21: Objekterkennung mittels kantenbasierten Template Matching an einem Beispielbild. In (a) ist das Eingangsbild zu sehen, welches mittels der Templates in Bild 4.20e in die Bewertungsmatrix (b) überführt wird. Das zugehörige Binärbild aus (b) ist in (c) zu sehen. Die Objekterkennung nach NMS ist in (d) dargestellt.

(d)

(c)

4.5.2.2 BLOB-Detektor für LWIR-Sensordaten

Ein solcher Detektor extrahiert durch ein Schwellwertverfahren sogenannte BLOBs (siehe Abschnitt 4.5.1.4) aus dem Eingangsbild. Diese werden auf Basis ihrer geometrischen Eigenschaften gefiltert und klassifiziert. Dabei ist ein BLOB im Allgemeinen eine kompakte Region im Bild, die heller oder dunkler als ihre Umgebung ist. Da Fahrzeuge durch Sonneneinstrahlung und durch ihren Motor eine ausgeprägte thermische Signatur gegenüber der Umgebungstemperatur besitzen, sind sie in LWIR-Bildern gut identifizierbar. Allerdings gibt es je nach Umgebung ähnliche thermische Signaturen, die durch thermische Abstrahlung an Vegetation oder Gebäuden entstehen. Tendenziell können sich bewegende Fahrzeuge durch ihre Wärmestrahlung gut identifiziert werden. Dabei ist die Fehlerrate besonders bei Nacht gering, da die Umgebung eine gedämpfte thermische Abstrahlung aufweist. Zu beachten ist, dass nachts abgekühlte Fahrzeuge nur einen geringen Kontrast zur Umgebung aufweisen. Um Fehldetektionen durch andere thermische Signaturen auszuschließen, werden verschiedene Eigenschaften der extrahierten thermischen Signaturen überprüft.

Damit aus dem Eingangsbild im LWIR-Spektrum die Signaturen bzw. BLOBs extrahiert werden können, werden zunächst mehrere Binärbilder durch Schwellwertbereiche generiert. Dabei werden k = 1, ..., N - 1 Schwellwertbereiche gebildet die sich ergeben aus

$$t_R^k = [\max(I) + k \cdot t_s, \min(I) + (k+1) \cdot t_s] \text{ mit } t_s = \frac{\max(I) - \min(I)}{N},$$
(4.23)

wobei *I* das Eingangsbild ist und N - 1 die Anzahl der Schwellwertbereiche angibt. Für den verwendeten BLOB-Detektor ist N = 20. Durch *Connected Component Labeling* (CCL) [139] werden aus jedem Binärbild BLOBs extrahiert, die auf Basis ihrer Mittelpunktabstände fusioniert werden. Dies bedeutet, dass BLOBs gruppiert werden, wenn der Abstand ihrer Mittelpunkte kleiner als eine Distanz *d* ist. Dabei bestimmt sich *d* durch die mittlere Objektgröße $\overline{o_z}$, geteilt durch die Bodenpixelauflösung *g*. Anschließend können die BLOBs auf vier verschiedene Kriterien hin überprüft werden:

- Die Fläche A
- Die Kreisförmigkeit Ci_B
- Die Konvexität Co_B
- Das Inertia Ratio I_B

Dabei ist die Fläche *A* die Anzahl der Pixel, die zum BLOB gehören. Die *Kreisförmigkeit* ist definiert durch folgende Formel:

$$Ci_B = \frac{4\pi A}{p^2} \tag{4.24}$$

wobei p der Umfang des BLOBs in Pixeln ist. Wenn $Ci_B = 1$, dann ist der BLOB nahezu kreisförmig und je kleiner Ci_B wird, umso weniger ähnelt der BLOB einem Kreis. Die Konvexität ist das Verhältnis der BLOB-Fläche zu der Fläche einer konvexen einhüllenden Kontur, wie in Bild 4.22a dargestellt. Dabei ist $Co_B = 1$, wenn der BLOB konvex ist und wird kleiner, sobald der BLOB konkave Eigenschaften annimmt. Außerdem kann zusätzlich das Inertia Ratio untersucht werden, welches die *Länglichkeit* eines BLOBs beschreibt. Definiert ist es durch das Verhältnis zwischen Neben- zu Hauptachse, wie in Bild 4.22b veranschaulicht. Dabei ist $I_B = 1$, wenn es sich um eine Kreisform, und $I_B = 0$ wenn es sich um eine Linienform handelt.



Bild 4.22: Konvexität und Inertia eines BLOBs. Die Abbildung (a) veranschaulicht die Berechnung der Konvexität und (b) zeigt das Inertia Ratio.

Für die Filterung der BLOBs zur Fahrzeugerkennung auf LWIR-Bildern wird die Fläche und das Inertia Ration mit folgenden Parametern verwendet:

$$\left(\frac{\overline{\sigma_z} - \sigma_o}{g}\right)^2 < A < \left(\frac{\overline{\sigma_z} + \sigma_o}{g}\right)^2 \qquad 0, 1 < I_B < 0,9 \tag{4.25}$$

Dabei ist σ_o die Standardabweichung der mittleren Objektgröße $\bar{\sigma_z}$. Für alle BLOBs, die diese Kriterien erfüllen, wird anschließend ein Vertrauenswert bestimmt. Dieser Vertrauenswert wird auf Basis der Ähnlichkeit zur gegebenen mittleren Objektgröße ermittelt. Er wird berechnet mit folgender Formel:

$$e = 2 - \left(\frac{|\bar{\sigma}_z - b_z| + \sigma_o}{\sigma_o}\right) \tag{4.26}$$

Der Wertebereich vom Vertrauenswert *e* ist $[1, -\infty]$, wobei der Wert Eins dem höchsten Vertrauen entspricht, dass es sich bei einem BLOB um ein Fahrzeug handelt. In Bild 4.23 ist das Ergebnis der BLOB-Detektion beispielhaft dargestellt.





Bild 4.23: Erkennung von Fahrzeugen mit dem BLOB-Detektor auf einem Beispiel LWIR-Sensorbild. Zu sehen ist das Eingangsbild (a) und als Referenz das visuelle Sensorbild in (b). Außerdem zeigt (c) eines der Binärbilder, das zur Erkennung benutzt wird. In (d) ist das Ergebnis der BLOB-Detektion zu sehen.

Dabei ist im Ergebnisbild Bild 4.23d bereits zu sehen, dass es durch Vegetation zu Fehlklassifikationen kommen kann. Das in Bild 4.23c dargestellte Binärbild entspricht dem Schwellwertebereich mit k = 5. Außerdem ist $\overline{\sigma_z} = 2 m$ und $\sigma_o = 1 m$.

4.5.2.3 Klassifizierungskaskade

Ein weiteres Verfahren, um Fahrzeuge in Bildern von VIS-Sensoren zu detektieren, ist die *Klassifizierungskaskade*. Vorgestellt wurde die Methode zunächst von Viola und Jones [50] mit einer Applikation zur Gesichtserkennung, und später wurde sie zur Fahrzeugdetektion auf Luftbildern verwendet, wie in [51] und [52] beschrieben. Das grundlegende Verfahren einer Klassifizierungskaskade ist die Verwendung mehrerer hintereinandergeschalteter Klassifizierungsstufen. Ziel jeder Stufe ist es, möglichst viele Negative (Nichtobjekte) auszusortieren, aber alle Positiven (Objekte) beizubehalten. Der Vorteil dieses Verfahrens ist, dass die Ausfallrate, d. h. der Anteil der fälschlich als positiv klassifiziert wird, für eine Stufe relativ hoch sein kann, z. B. 50 %, denn über die Summe aller Stufen lässt sich schließlich ein robuster Klassifikator bilden.

Ein weiterer Vorteil der Klassifizierungskaskade gegenüber einem einzelnen *monolithischen* Klassifikator ergibt sich aus dem geringen Rechenaufwand bei gleichzeitiger hoher Detektionsgenauigkeit. Außerdem eignet sich das Verfahren für kleine Objektgrößen, z. B. $24 px \times 24 px$ für die Gesichtserkennung, durch die Verwendung von *Haar*-ähnlichen oder *Local Binary Pattern* (LBP) Merkmalen. Im Folgenden werden zunächst die beiden Merkmale und anschließend das Lernverfahren zur Konstruktion der Klassifizierungskaskade vorgestellt. Der Hauptgrund, warum Merkmale gegenüber rohen Pixelwerten verwendet werden, ist die Erhöhung bzw. Reduzierung der Varianz der Objekt- bzw. Nichtobjektklasse. Dadurch wird die Separierung der beiden Klassen vereinfacht.

Die sogenannten *Haar*-ähnlichen Merkmale vergleichen zwei Regionen im Bild hinsichtlich der Differenz ihrer aufsummierten Pixelwerte. Lienhart et al. [141] stellte vier verschiedene Kategorien von Prototypen vor, die unabhängig in jede Dimension skaliert werden können und deren Position im zu betrachtenden Bildausschnitt frei gewählt werden kann. Diese sind in Bild 4.24 schematisch dargestellt.



Bild 4.24: Prototypen von einfachen *Haar*-ähnlichen und zentrischen Merkmalen. In (a) sind Kantenmerkmale, in (b) sind zentrische Merkmale, in (c) ist ein spezielles diagonales Merkmal und in (d) sind Linienmerkmale abgebildet. Dabei besitzen blaue Flächen negative und weiße Flächen positive Gewichte.

Dabei gibt es jeweils die rechteckige und rotierte Variante der Prototypen. Durch die Positionierungs- und Skalierungsmöglichkeiten der Merkmalsprototypen entsteht eine Überrepräsentierung der Basiselemente. Dies bedeutet, dass es mehr Merkmale als Pixel im Bild gibt. Der Vorteil der Merkmale ist, dass sie Domänenwissen enkodieren können (z. B. Kantenübergänge an bestimmten Positionen im Bild), und es damit möglich ist, mit wenigen prägnanten Merkmalen einen Klassifikator zu lernen. Außerdem, wie später beschrieben wird, ist eine relativ kleine Anzahl von Merkmalen ausreichend, um eine genügend hohe Detektionsgenauigkeit zu erreichen.

Die *Haar*-ähnlichen Merkmale, die alle Rechtecke bzw. um 45° rotierte Rechtecke sind, können durch die Verwendung des *Integralbildes* sehr effizient berechnet werden. Dabei ist das Integralbild eine intermediäre Darstellung eines Bildes. Das Integralbild *II* an der Stelle (*x*, *y*) hat den Wert der Summe aller Pixelwerte, die sich links und über der Position befinden. Veranschaulicht ist dies in Bild 4.25a. Ähnlich folgt die Berechnung des rotierten Integralbildes *RII*, wie in Bild 4.25c dargestellt.



Bild 4.25: Schemata der Berechnung des Integralbildes und rotierten Integralbildes. Dabei zeigt (a) das Integralbild, (b) das Schema einer Rechtecksummenberechnung im Integralbild, (c) das rotierte Integralbild und (d) das Schema einer Rechtecksummenrechnung im rotierten Integralbild.

Die Formeln für die Berechnung des Integralbildes und des rotierten Integralbildes lauten wie folgt:

$$II(x,y) = \sum_{x' \le x, y' \le y} I(x',y') \qquad RII(x,y) = \sum_{y' \le y, y' \le y - |x-x'|} I(x',y')$$
(4.27)

Dabei ist (x, y) die Position, für die im Integralbild der Integralwert berechnet werden soll, und (x', y') die Position im Originalbild. Für ein Rechteck lässt sich nun durch vier Werte im Integralbild die Summe der Pixelwerte in diesem berechnen mit

$$s(r) = II(x - 1, y - 1) + II(x + w - 1, y + h - 1) - II(x - 1, y + h - 1) - II(x + w - 1, y - 1),$$
(4.28)

wobei w die Breite und h die Höhe sowie (x, y) die obere linke Ecke des Rechtecks ist. Die Berechnung ist in Bild 4.25b dargestellt. Für die rotierten Rechtecke lautet die Berechnung wie folgt:

$$s(r) = RII(x - h + w, y + w + h - 1) + RII(x, y - 1) - RII(x - h, y + h - 1) - RII(x + w, y + w - 1)$$
(4.29)

Bild 4.25d zeigt beispielhaft die Berechnung für die rotierten Rechtecke.

Neben den *Haar*-ähnlichen Merkmalen werden auch LBP-Merkmale verwendet. Diese bilden eine bestimmte Region im Bild auf einen Binärcode ab, der ebenfalls Domänenwissen enkodieren kann. In Bild 4.26a ist das Verfahren zur Ermittlung des Binärcodes aus einer 3×3 Pixelregion dargestellt. Dabei wird jeder umliegende Pixelwert mit dem Wert des mittleren Pixels verglichen. Ist er kleiner, ist das Ergebnis eine 0, andernfalls eine 1. Durch diese Schwellwertbildung kann der Binärcode durch Aneinanderreihen der Binärwerte im Uhrzeigersinn, startend von oben links, erzeugt werden. Die Erweiterung des LBP-Merkmals, das *Multi-scale Local Binary Pattern* (MB-LBP) Merkmal [142], berechnet einen Binärcode aus größeren Pixelregionen, wie z. B. 9×9 oder 15×15 . Dabei werden in der gewählten Pixelregion 3×3 gleichgroße Zellen gebildet, wie in Bild 4.26b dargestellt. In diesen Pixelregionen wird zunächst der Mittelwert der Intensitätswerte gebildet. Anschließend wird das Berechnungsverfahren der LBP-Merkmale verwendet, um den Binärcode zu bestimmen. Durch Verwendung des Integralbildes lässt sich das MB-LBP-Merkmal ebenfalls effizient berechnen.



Bild 4.26: Darstellung der Berechnung des LBP Merkmals. In (a) ist die Berechnung des originalen LBP Merkmals zu sehen und (b) deutet die Erweiterung zum MB-LBP Merkmal an.

Wie die *Haar*-ähnlichen Merkmale, entsteht durch Positionierung und Skalierung des MB-LBP-Merkmals eine Überrepräsentierung der Basiselemente. All diese Merkmale zu berechnen, obwohl sie effizient bestimmt werden können, würde die Berechnungszeit erhöhen und den Vorteil des Verfahrens schmälern. Deswegen wird eine Klassifizierungskaskade durch sogenanntes *Boosting* erzeugt, welches für jede Stufe in der Klassifizierungskaskade einen kleinen Satz von Merkmalen auswählt und damit einen Klassifikator trainiert. Dieser muss lediglich etwas besser sein als der Zufall. Erreicht wird dies durch eine möglichst hohe Trefferquote wobei nur ein relativ kleiner Teil der Negativen aussortiert wird. Weiterhin werden die ersten Stufen mit möglichst wenigen Merkmalen trainiert, um mit wenig Rechenaufwand so viele Negative wie möglich zu eliminieren. Zur Minimierung der Ausfallrate wird von Stufe zu Stufe die Anzahl der Merkmale gesteigert. In Bild 4.27 ist ein Beispiel einer *N*-Stufigen Klassifizierungskaskade zu sehen.



Bild 4.27: Klassifizierungskaskade mit *N* Stufen. Jede Stufe wird dabei so trainiert, dass eine Trefferquote *h* und eine Ausfallrate *f* erreicht wird.

Es gibt verschiedene *Boosting*-Verfahren, die in ihrer Rechenzeitkomplexität identisch sind, sich aber in ihrem Lernalgorithmus unterscheiden. Verwendet wird der *Gentle AdaBoost* [141] Algorithmus, weil dieses Verfahren gegenüber anderen gängigen *Boosting*-Methoden eine Klassifizierungskaskade mit weniger Merkmalen erzeugt und damit eine schnellere Objektklassifikation erlaubt. Im Allgemeinen wird beim *Boosting* eine Menge von extrahierten Merkmalen und deren Zuordnung zur Klasse (Objekt oder Nichtobjekt) verwendet, um einen Klassifikator zu trainieren. Jedes Merkmal erhält am Anfang zu gleichen Teilen einen Gewichtsfaktor. Anschließend wird für jedes Merkmal ein Klassifikator trainiert, der mit dem

Gewichtsfaktor gewichtet wird, und dessen Klassifikationsfehler berechnet. Die Gewichtsfaktoren werden für Klassifikatoren mit einem relativ niedrigen Klassifikationsfehler vergrößert und für relativ hohe Klassifikationsfehler verkleinert. Der Klassifikator mit dem kleinsten Fehler wird beibehalten. Zuletzt werden die Gewichte normiert und der Prozess startet in einer Schleife wieder beim Trainieren der Klassifikatoren, bis bestimmte Kriterien erfüllt sind, wie zum Beispiel die Anzahl der maximalen Iterationen (bzw. maximale Anzahl an Merkmalen), eine minimale Trefferquote oder eine maximale Ausfallrate.

Durch die Veränderung der Perspektive auf das Objekt während eines Überfluges wird sich auch die Signifikanz der Merkmale ändern. Diese sind nämlich nicht rotationsinvariant bzw. deformierbar. Deshalb wird für bestimmte Objektorientierungen und Perspektiven jeweils eine Klassifikationskaskade erzeugt. Dieses Verfahren wurde bereits in [54] vorgeschlagen und zeigte, dass für Elevationswinkel von -45° und -65° , mit einer Flughöhe von 140 *m* bei einer Objektgröße von 40 *px* × 40 *px*, die höchsten Genauigkeiten erreicht werden konnten. In Bild 4.28a sind die geometrischen Verhältnisse mit den entsprechenden Parametern dargestellt. Je nach Objektorientierung gegenüber dem Kamerablickwinkel wird das Fahrzeug einer Kategorie (A, B oder C) zugeordnet, wie in Bild 4.28b veranschaulicht ist. Die Zuordnung erfolgt dabei durch den Winkel zwischen optischer Achse der Kamera und der Hauptachse des Fahrzeuges. In Verbindung mit den beiden vorgestellten Merkmalen, *Haar*-ähnliche und LBP, entstehen so 12 Klassifikationskaskaden, die parallel eingesetzt werden. Dabei werden die Ergebnisse durch eine NMS [50] fusioniert, welche überlappende Detektionsregionen vereinigt und mittelt. In Bild 4.29 ist ein Beispiel einer Klassifikation mittels der Klassifikationskaskaden zu sehen.



Bild 4.28: Kategorisierung der Erscheinung eines Fahrzeuges in einer Szene. Je nach Fahrzeugausrichtung gegenüber der Blickrichtung wird die Kategorisierung vorgenommen, wie in (b) dargestellt. Außerdem zeigt (a) die Parameter der Kamerablickrichtung: Den Elevationswinkel α und die Höhe über Grund h.



Bild 4.29: Detektionen der Klassifizierungskaskade anhand von Beispielen. In (a) sind Detektionen auf einem synthetischen Bild zu sehen und in (b) auf einem realen Sensorbild.

In Abschnitt 4.5.2.5 ist die Generierung der Trainingsbeispiele für die einzelnen Klassifikationskaskaden beschrieben und in Abschnitt 4.5.2.6 sind die Ergebnisdetails der einzelnen Klassifikationskaskaden dargestellt.

4.5.2.4 DPM basierter Objektdetektor

Beim DPM (siehe Abschnitt 2.2.2) handelt es sich um eine relativ neue Methode zur Objekterkennung, die von Felzenszwalb et al. [58] vorgestellt wurde. Generell basiert das Verfahren nicht allein auf der Erkennung des gesamten Objekts, sondern zusätzlich auf der Detektion von Objektteilen. Dabei werden die Objektteile durch ein Lernverfahren bestimmt, um den Aufwand des händischen Annotierens zu reduzieren. Zusätzlich werden die Positionen der Objektteile relativ zur Objektmitte, auch als latente Variablen bezeichnet, bestimmt. Im Gegensatz zu der Klassifizierungskaskade zeigt das DPM eine bessere Detektionsgenauigkeit und ist robuster gegenüber perspektivischen Verzerrungen und Objektorientierungen. Nachteilig ist je nach Modellgröße der relativ hohe Rechenaufwand und die benötigte Mindestauflösung des Objektes. Im Folgenden wird die Objekterkennung des Deformable Part Models erläutert und kurz auf den Lernprozess des Modells eingegangen. Das Modell des DPMs besteht aus $K \cdot 2$ Komponenten, wobei K die Anzahl der gewünschten Objektansichten ist, mit jeweils einem Wurzelfilter F_0 und n Teilefiltern P_n . Dabei besteht jeder Teilefilter aus dem eigentlichen Filter F_i und einer optimalen Position v_i des Filters bezüglich des Wurzelfilters. Außerdem besitzt jeder Teilefilter den Parameter d_i , der zu Berechnung der Verschiebungskosten gegenüber der optimalen Position v_i herangezogen wird. In Bild 4.30 ist ein Beispiel eines Zweikomponenten-Fahrzeugmodells zu sehen.





Bild 4.30: Beispielerkennungen eines Zweikomponenten-Fahrzeugmodells. In (a) ist die Detektion in Rot und die zugehörigen Teilemodelle in Blau eingezeichnet. Die Wurzelfilter sind in (b), die verschiedenen hochauflösenden Teilefilter in (c) und das räumliche Modell für die Positionen der Teilefilter in (d) abgebildet.

Eine entsprechende Beispieldetektion ist in Bild 4.30a dargestellt. Dabei zeigt Bild 4.30b die beiden Wurzelfilter, Bild 4.30c die Teilefilter der jeweiligen Komponente und Bild 4.30d die Visualisierung der Verschiebungskosten, wobei die Kosten von Schwarz nach Weiß steigen. Die Filter zeigen jene Gradienten auf die sie reagieren, wobei zur Extraktion dieser das HOG-Merkmal [55] verwendet wird. Außerdem besitzen die Teilefilter die doppelte Auflösung gegenüber dem Wurzelfilter und können somit auch auf Details des Objektes reagieren. Wie in Bild 4.30 zu erkennen, ist jede zweite Komponente im Modell immer eine gespiegelte Variante der Komponente davor.

Der Detektionsprozess startet mit der Berechnung einer Merkmalspyramide, die zum einen das Verfahren skalierungsinvariant macht und zum anderen für die unterschiedlichen Auflösungen der Wurzel- und Teilefilter benötigt wird. Dabei spezifiziert der Parameter λ die Anzahl der Ebenen, die in der Pyramide nach unten gegangen werden muss, um eine Ebene mit der doppelten Auflösung zu erreichen. Zunächst wird mittels des Standardverfahrens zur Pyramidenberechnung durch wiederholte Glättung und Unterabtastung eine Bildpyramide erzeugt, wie sie in der linken Hälfte in Bild 4.31 zu sehen ist. Hierbei ist $\lambda = 3$. Anschließend wird durch Merkmalsextraktion auf jeder Ebene der Bildpyramide die Merkmalspyramide H erzeugt. Dabei wird das HOG-Merkmal zur Merkmalsextraktion eingesetzt. Ein Merkmalsvektorbereich in der Merkmalspyramide wird ausgedrückt durch $\Phi(H, p, w, h)$, wobei p die Position (x, y) des Markmalvektors in der Pyramideneben l definiert und w und hdie Breite und Höhe des Bereiches bestimmt. Die Antwort eines Filters des DPM-Modells wird mit dem Skalarprodukt $F_i \cdot \Phi(H, p, w, h)$ berechnet, indem Filterwerte und Merkmalsvektorbereich in zeilenweise Anordnung (engl. row-major) gebracht werden.



Bild 4.31: Bild- und HOG-Merkmalspyramide mit Objekthypothese. Die Objekthypothese ist durch die Platzierung des Wurzelfilters nahe der Spitze der Pyramide und der Teilefilter nahe dem Boden der Pyramide definiert.

Für eine Objekthypothese und eine Modellkomponente wird das Bewertungsmaß wie folgt berechnet:

$$\sum_{i=0}^{n} F_{i} \cdot \Phi(H, p_{i}, w_{i}, h_{i}) - \sum_{i=1}^{n} d_{i} \cdot \Phi_{d}(dx_{i}, dy_{i}) + b$$
(4.30)

Der erste Term bestimmt die Summe der Antworten des Wurzelfilters und der zugehörigen Teilefilter. Der zweite Term zieht davon die Verschiebungskosten ab, wobei

$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i)$$
(4.31)

die Verschiebung des entsprechenden Teilefilters bzgl. seines Ankers ist. Das Verschiebungsmerkmal ist wie folgt definiert:

$$\Phi_d(dx, dy) = (dx, dy, dx^2, dy^2)$$
(4.32)

Durch d_i , den Deformationskoeffizienten, kann die Deformationskostenberechnung bestimmt werden, wobei die Verschiebungskosten auf Basis der quadrierten Distanz durch die Deformationskoeffizienten $d_i = (0,0,1,1)$ erreicht werden. Der letzte Term addiert den Bias *b* hinzu, der einen Ausgleich zwischen den Komponenten eines DPM-Modells vornimmt. Dadurch sind die Werte der einzelnen Bewertungsmaße vergleichbar.

Der Detektionsprozess beginnt zunächst durch Bewegung der Wurzelfilter über das Bild und sucht die besten Positionen anhand der stärksten Reaktionen des Filters. Für jede Wurzelfilterposition werden die entsprechenden Teilefilter um ihre Ankerposition bewegt, um die maximale Antwort und damit auch die beste Position zu finden. Anschließend kann mit Formel (4.30) das Bewertungsmaß berechnet werden. Durch einen Schwellwert wird bestimmt, ob es sich bei der gefundenen Position um die gesuchte Objektklasse handelt oder nicht.

Im Folgenden soll kurz der Algorithmus zum Trainieren der DPM-Modellparameter vorgestellt werden. Die Details sind in [58] nachzulesen und implementiert in [143]. Generell besteht der Lernalgorithmus aus zwei Schritten, die für eine bestimmte Anzahl wiederholt werden:

- Aktualisierung der latenten Variablen, also das Finden der besten Positionen bzw. Anker für die Wurzel- und Teilefilter, wobei die restlichen Modellparameter nicht verändert werden
- Aktualisierung der Modellparameter (Filter und Deformationskoeffizienten), wobei die Positionen der Teilefilter nicht verändert werden

Bevor die beiden Schritte iterativ ausgeführt werden, muss das Modell initialisiert werden. Zunächst wird anhand der auftretenden Pixelseitenverhältnisse aller positiven Beispiele eine Sortierung vorgenommen, so dass *K*-Gruppen entstehen. Weiterhin werden die Wurzelfilter mittels einer Standard SVM [55] gelernt. Anschließend werden die beiden vorher genannten Schritte einmal ausgeführt, wobei die einzelnen Komponenten noch keine Teilefilter besitzen. Ziel ist es hier, die optimale Position der Wurzelfilter und ihrer Filtergewichte zu bestimmen. Am Ende der Initialisierung werden die Teilefilter erzeugt. Dabei werden *n* Teilefilter generiert, die auf Basis der Filtergewichte positioniert werden. Die Positionierung erfolgt dabei entweder ausgerichtet an der vertikalen Mittellinie oder, bei einer Positionierung, die nicht zentriert ist, durch Platzieren eines weiteren gespiegelten Filters an der vertikalen Mittellinie. Die Filtergewichte der Teilefilter werden durch Interpolation des Wurzelfilters auf die doppelte Auflösung initialisiert und die Deformationskoeffizienten werde zu $d_i = (0,0,1,1)$ gesetzt. Nachdem die Initialisierung abgeschlossen ist erfolgt der iterative Lernprozess, bestehend aus den beiden genannten Schritten.



Bild 4.32: Beispieldetektionen des DPMs. In (a) sind Erkennungen in einem synthetischen Bild zu sehen, wobei hier zwei Fahrzeuge auf unterschiedlichen Pyramidenstufen detektiert wurden und noch keine NMS durchgeführt wurde. In (b) ist eine Detektion auf einem realen Bild abgebildet.

In Bild 4.32 sind Beispieldetektionen des verwendeten DPM-Modells zu sehen. In Abschnitt 4.5.2.5 ist die Generierung der Trainingsdaten beschrieben, und Abschnitt 4.5.2.6 zeigt die erzeugten DPM-Modelle und deren Ergebnisse.

4.5.2.5 Erzeugung des Objektdatensatzes für CC und DPM-Detektor

Zwei der vier verwendeten Objektdetektoren benötigen Trainingsdaten, um ein entsprechendes Objektmodell zu lernen. Für die DPM basierte Objekterkennung (siehe Abschnitt 4.5.2.4) werden aus den vorher beschriebenen 70 festgelegten Datasätzen (siehe Abschnitt 4.4) zwei neue Datensätze extrahiert. Für diese Datensätze wurden alle Bild-Annotationspaare extrahiert, welche Objekte enthalten und mit einem Elevationswinkel von -60° bzw. -75° erzeugt worden sind. Das Framework [143] zum Trainieren der DPM-Modellparameter extrahiert die Positivbeispiele und erzeugt zufällig Negativbeispiele aus den Trainingsdaten. In Bild 4.33 sind einige positive Beispieltrainingsdaten der beiden Datensätze abgebildet.



Bild 4.33: Beispielbilder aus den Datensätzen zum Trainieren der DPM Modelle. In (a) sind Beispiele aus dem -60° Datensatz dargestellt und (b) zeigt Beispiele aus dem -75° Datensatz.

Die Klassifikatoren der Klassifizierungskaskade (siehe Abschnitt 4.5.2.3) werden mit dem Datensatz von Hummel et al. [54] trainiert. Der Datensatz wurde ebenfalls mit einer synthetischen Simulationsumgebung erzeugt und enthält pro Objektkategorie 4.000 Bilder und insgesamt 7.200 negativ Beispiele. Dabei ist eine Objektkategorie eine Kombination aus Objektorientierung, Höhe des Sensors und Elevationswinkel, wie in Bild 4.28 in Abschnitt 4.5.2.3 dargestellt. Verwendet wurden die 12 Objektkategorien, die in der Evaluierung in [54] die besten Resultate zeigten. Das sind jeweils die Objektorientierungen A, B und C sowie die Elevationswinkel -45° und -65° bei einer Höhe über Grund von 140 *m*. In Bild 4.34 sind einige Beispieldaten des Datensatzes zu sehen.



Bild 4.34: Trainingsbeispiele für die Klassifizierungskaskade. In (a) sind Positivbeispiele mit -45° und in (b) mit -65° Elevationswinkel zu sehe. Zeile für Zeile sind jeweils Fahrzeuge in der Objektorientierung A, B und C abgebildet. Die Negativbeispiele sind in (c) dargestellt.

4.5.2.6 Training des CC- und DPM-Objektdetektors

Dieser Abschnitt zeigt Details zum Training der Klassifikationskaskade und des DPM-Detektors. Außerdem werden die Resultate des Trainings vorgestellt.

<u>Klassifikationskaskade</u>

Die Klassifikationskaskade (siehe Abschnitt 4.5.2.3) zur Detektion von Fahrzeugen besteht selbst aus 12 Klassifikationskaskaden, wobei jeweils drei Klassifikationskaskaden zusammen betrachtet werden. Diese decken jeweils die drei Objektorientierungen A, B und C ab. Tabelle 4.4 zeigt die Konfiguration und Ergebnisse des Trainings der Klassifikationskaskaden. Verwendet wurde die Implementierung von OpenCV [144].

Konfiguration	Anzahl Stufen			Trefferquote	Ausfallrate	
Konngulation	Α	В	С	in %	in %	
HAAR-45	19	20	20	84,3	15,7	
HAAR-65	18	20	19	80,5	19,5	
LBP-45	18	20	20	75,7	24,3	
LBP-65	19	20	20	61,8	38,2	

Tabelle 4.4: Konfiguration und Ergebnisse des Trainings der Klassifikationskaskaden

Dabei wurden folgende Lernparameter konfiguriert:

- Minimale Trefferquote: 0,9999
- Maximale Ausfallrate: 0,5
- Maximale Anzahl an Stufen pro Klassifikationskaskade: 25
- Maximale Anzahl an Merkmalen pro Stufe: 30
- Boosting Methode: GentleAda Boost

Der Konfigurationsname setzt sich aus dem verwendeten Merkmal (Haar oder LBP) und dem Elevationswinkel des VIS-Sensors im verwendeten Teildatensatz des Objektdatensatzes zusammen. Die Ergebnisse zeigen, dass die Haar-ähnlichen Merkmale zu besseren Detektionsleistungen führen, als die LBP-Merkmale. Außerdem zeigt sich, dass flachere Elevationswinkel zu höheren Trefferquoten führen. Die beiden Datensätze (-45 und -65 Grad

Elevationswinkel) wurden jeweils in Training- und Testdaten im Verhältnis 80 % zu 20 % geteilt.

DPM-Detektor

Für das Training der DPM-Klassifikatoren wurde das Framework von Girshick et al. [143] herangezogen. Die Konfiguration und Ergebnisse des Trainings sind in Tabelle 4.5 dargestellt. Beim Training wurden die Lernparameter *Komponentenanzahl* und *Lernrate* variiert. Es zeigte sich, dass die Verwendung von mehr als 3 Komponenten zu keiner weiteren Verbesserung der Ergebnisse führte. Die besten Modelle pro Datensatz sind in der Tabelle 4.5 grün hinterlegt.

Datensatz	Elevationswinke	l 60-Grad	Datensatz Elevationswinkel 75-Grad			
Komponenten	Lernrate	AP	Komponenten	Lernrate	AP	
2	0,01	0,728	2	0,01	0,735	
2	0,001	0,718	2	0,001	0,743	
2	0,0001	0,717	2	0,0001	0,741	
3	0,01	0,733	3	0,01	0,779	
3	0,001	0,729	3	0,001	0,768	
3	0,0001	0,701	3	0,0001	0,737	

Tabelle 4.5: Konfiguration und Ergebnisse des Trainings des DPM basierten Objektdetektors. Die grün hinterlegten Zeilen, zeigen jeweils den besten Klassifikator pro Datensatz.

Für die beiden besten Modelle finden sich in Bild 4.35 die Recall-Precision Kurven mit zugehöriger durchschnittlicher Genauigkeit (engl. *average precision*, AP). Hier zeigen die Ergebnisse, im Gegensatz zur Klassifikationskaskade, dass steilere Elevationswinkel zu einer besseren Erkennungsleistung führen. Die Visualisierungen der beiden DPM-Modelle sind in Anhang C im Bild 9.4 und Bild 9.5 dargestellt.



Bild 4.35: Recall-Precision Kurve der beiden besten DPM-Klassifikatoren. Dabei ist in Grün der Klassifikator abgebildet, der auf dem 60-Grad-Datensatz und in Blau der Klassifikator zu sehen, der auf dem 75-Grad-Datensatz gelernt wurde.

4.6 Definition des Leistungsindex

Zur Bewertung der Leistungscharakteristika der verwendeten Perzeptionsmodule muss eine eindeutige Definition des Leistungsindex p vorliegen. Er reflektiert die Algorithmenleistung eines Perzeptionsmoduls und belegt den Wertebereich [0,1]. Dabei bedeutet ein Wert gegen 0 eine sehr geringe und ein Wert gegen 1 eine sehr hohe Performanz des Algorithmus. Die spezifische Berechnung des Leistungsindex ist abhängig von den verwendeten Methoden.

Für Bildverarbeitungsalgorithmen der Kategorie *Objektklassifikation* bietet sich ein Standardverfahren zum Ermitteln des Leistungsindex an. Typischerweise werden binäre Objektdetektoren mittels der Wahrheitsmatrix (engl. *confusion matrix*) evaluiert. Dabei wird ausgehend von einer Menge P an positiven Beispielen, welche die zu erkennende Objekte repräsentieren, und einer Menge N an negativen Beispielen bestimmt, wie viele positive und negative Beispiele richtig bzw. falsch klassifiziert werden. Bild 4.36a zeigt die grafische Darstellung einer Wahrheitsmatrix. Dabei ist t_p die Anzahl der richtig klassifizierten positiven Beispiele und f_p die Anzahl der falsch klassifizierten negativen Beispiele. Dies sind diejenigen negativen Beispiele, welche fälschlicherweise als positive klassifiziert wurden. Das bedeutet, dass das Oval in Bild 4.36a alle richtig klassifizierten Beispiele des Objektdetektors beinhaltet. Die Anzahl der fälschlicherweise als negative klassifizierten Positiven f_n bildet zusammen mit t_p die Menge aller positiven Beispiele. Schließlich bildet t_n die Anzahl aller richtig klassifizierten Negativen, und f_p die Menge aller negativen Beispiele. Zwei elementare Messungen zur Güte eines Klassifikators sind die Trefferquote (engl. *recall*) und die Genauigkeit (engl. *precision*), dessen Berechnung in Bild 4.36b zu sehen ist.



Bild 4.36: Beispielhafte Darstellung der Wahrheitsmatrix mit Berechnung von *recall* und *precision*. In (a) bildet die graue und hellblaue Fläche die Menge an positiven Beispielen und die blaue und dunkelblaue Fläche die Menge an negativen Beispielen. Die Berechnung von recall und precision ist in (b) angegeben.

Da sich die Messungen gegenseitig beeinflussen, wurden verschiedene kombinierte Maße eingeführt. Ein etabliertes Maß ist F_{β} , welches Trefferquote und Genauigkeit kombiniert. Definiert ist das Maß für alle $\beta \ge 0$:

$$F_{\beta} = (1 + \beta^2) \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$
(4.33)

Wenn $\beta = 1$, dann wird das harmonische Mittel zwischen Trefferquote und Genauigkeit berechnet und auch als F_1 -Maß bezeichnet. Zu beachten ist, dass dieses Maß t_n nicht berücksichtigt. Andere Maße, wie der *Matthews correlation coefficient*, der *Youden-Index* oder *Cohens Kappa* [145] betrachten diesen, sind aber aufwändiger zu berechnen und oder nicht immer berechenbar, wenn eines der Grundmaße t_p , f_n , f_p oder t_n gleich Null ist. Daher wird als Leistungsindex für die Objektdetektoren das etablierte F_1 -Maß eingesetzt.

4.7 Generierung der Metadaten

Mit Hilfe des *Modelldatensatzes*, welcher die generierten Bilddaten \tilde{I} und zugehörigen Annotationen \tilde{A} inklusive Umgebungszustand f_x umfasst, lässt sich die Leistungsbestimmung

der Perzeptionsmodule zur Objektdetektion (BLOB, TM, CC und DPM) durchführen. Dabei wird die Leistung eines Algorithmus aus dem Vertrauenswert bzw. *Score s* und dem Label *l*, welches die *Ground Truth* beinhaltet, ermittelt. Das Label *l* trägt die Information, ob es sich bei dem Bildausschnitt um ein Positivbeispiel l = 1 (Objekt ist ein Fahrzeug) oder Negativbeispiel l = 0 handelt. Aus dem Vertrauenswert wird über den Bias des jeweiligen Objektdetektors das prädizierte Label l_p gebildet, wie in Abschnitt 4.5.2 erläutert. Daraus können sich folgende vier Konstellationen bilden:

- $l_p = 0$ und l = 0: Ein Negativbeispiel wurde als Negativ prädiziert (t_n) .
- $l_p = 0$ und l = 1: Ein Positivbeispiel wurde als Negativ prädiziert (f_n).
- $l_p = 1$ und l = 0: Ein Negativbeispiel wurde als Positiv prädiziert (f_p).
- $l_p = 1$ und l = 1: Ein Positivbeispiel wurde als Positiv prädiziert (t_p).

Über die Berechnungsvorschrift des F_1 -Maßes, siehe Abschnitt 4.6, wird der Leistungsindex berechnet. Um die Rechenzeit der Leistungsbestimmung zu reduzieren, wird nicht der gesamte Bildinhalt ausgewertet, sondern pro Bild die annotierten Objekte (Fahrzeuge) und 10 zufällig generierte Negativbeispiele. Bild 4.37 zeigt schematisch die Leistungsbestimmung mittels Funktionsblöcken. Jeder Objekterkennungsalgorithmus a_i bewertet auf jedem Bild die gegebenen Bildausschnitte mit dem Vertrauenswert bzw. *Score s* zur späteren Berechnung des Leistungsindex. Die Ergebnisse, Vertrauenswerte, Labels und Umgebungszustand, werden pro Algorithmus und Sensorbild in $\mathcal{D}_r^{(i)}$ abgelegt. Damit sind die *Metadaten* {F, P} pro Algorithmus in $\mathcal{D}_r^{(i)}$ zusammengefasst und der Zusammenhang zwischen Umgebungszustand und Algorithmenleistung hergestellt.



Bild 4.37: Leistungsbestimmung der Objekterkennungsalgorithmen zur Metadatengenerierung.

Die Erzeugung der Bildausschnitte für die Negativbeispiele erfolgt über die Uniformverteilung U, welche gleichverteilte Zufallszahlen in einem definierten Wertebereich erzeugt. Da hier die Fahrzeugerkennung als Anwendung betrachtet wird, werden die Bildausschnitte (x, y, w, h) so erzeugt, dass sie zwischen 1 m und 6 m groß sind. Das Rechteck des Bildausschnittes mit der linken oberen Position (x, y) und der Breite und Höhe (w, h) bestimmt sich mit:

$$x = U\left(0, W - \left\lfloor\frac{6}{g}\right\rfloor\right) \qquad y = U\left(0, H - \left\lfloor\frac{6}{g}\right\rfloor\right)$$
$$w = U\left(\frac{1}{g}, \frac{6}{g}\right) \qquad h = U\left(\frac{1}{g}, \frac{6}{g}\right) \qquad (4.34)$$

Dabei ist g die mittlere Bodenpixelauflösung, W die Breite und H die Höhe des Bildes in Pixel.

KAPITEL 5 VERFAHREN ZUR LEISTUNGSMODELLIERUNG

Das vorherige Kapitel beschäftigte sich mit den Verfahren zur Generierung der *Metadaten*. Dieses Kapitel behandelt nun die Verfahren zur Leistungsmodellierung. Zum einen wird die Leistungsmodellierung mittels Expertenwissens behandelt und zum anderen wird die Leistungsmodellierung durch die beiden ML-Verfahren NEFPROX und KNN vorgestellt. Bild 5.1 zeigt einen Überblick der Erzeugung der Leistungsmodelle.



Bild 5.1: Überblick der Verfahren zur Erzeugung der Leistungsmodelle.

Zur Umsetzung der Experten-Leistungsmodelle (1) werden in Abschnitt 5.1 zunächst die grundlegenden Informationen eines Fuzzy-Inferenzsystems wiedergegeben. Anschließend findet die Modellierung dieser mittels des Expertenwissens für alle Perzeptionsmodule statt. Abschnitt 5.1.3 behandelt die beiden ML-Verfahren, NEFPROX (2) und KNN (3), zu Funktionsapproximation mittels der erhobenen *Metadaten* (vgl. Abschnitt 4.7) für die Perzeptionsmodule BLOB, TM, CC und DPM.

5.1 Leistungsmodellierung mittels Expertenwissens

Dieser Abschnitt behandelt zunächst den grundlegenden Aufbau eines FIS, um damit eine geeignete Erfassung des Expertenwissens umzusetzen. Zur Übertragung des Expertenwissens in ein FIS wird anschließend die Funktionsweise des FIS-Inferenzprozesses dargestellt. Anschließend erfolgt die Erfassung des Expertenwissens für jedes verwendete Perzeptionsmodul. Abschließend wird das Verfahren zu Erzeugung der Experten-Leistungsmodelle präsentiert.

5.1.1 Aufbau des Fuzzy-Inferenzsystems

Ein Anwendungsbereich der Fuzzy-Logik ist die Modellierung von Expertenwissen. Dabei wird berücksichtigt, dass der Mensch vorwiegend unscharfe bzw. unpräzise Konzepte nutzt, um u. a. Sachverhalte oder Zusammenhänge zu beschreiben. Gerade dann, wenn das Erfassen eines mathematischen Modells durch analytische oder numerische Verfahren nicht möglich oder zu aufwendig ist, aber Expertenwissen vorliegt, können Fuzzy-Inferenzsysteme herangezogen werden. Für die Modellierung der Leistungsmodelle durch Expertenwissen, gilt der zugrundeliegende Gedanke, dass der Entwickler von Perzeptionsmodulen bestimmtes Wissen um den dahinterliegenden Algorithmus besitzt und dessen Eigenschaften in Bezug auf die Umgebungszustände erfassen kann. Des Weiteren ist es möglich, dass ein Operateur durch Einsatz der Sensordatenverarbeitung Erfahrungswissen besitzt und damit zusätzlich die Leistungsmodelle mit Wissen anreichert.

Der Aufbau des hier verwendeten FIS besteht aus den typischen vier Verarbeitungskomponenten und eine durch Expertenwissen modellierte Wissensbasis, welche Zugehörigkeitsfunktionen und Regeln in Form von unscharfem Wissen enthält. Um solche unscharfen Konzepte zu realisieren, wird im Allgemeinen die zweiwertige Logik auf das Einheitsintervall [0,1] erweitert. Damit ein Experte Wissen in das System übertragen kann, werden sogenannte linguistische Variablen verwendet. Diese Variablen können unterschiedlich viele Terme annehmen, z. B. könnte die Variable Höhe die Terme niedrig, halbhoch oder hoch annehmen. Damit nun ein scharfer Eingangswert (z. B. der Elevationswinkel) in eine linguistische Variable überführt werden kann, werden Zugehörigkeitsfunktionen (engl. *fuzzy sets*) verwendet. Diese geben den Grad der Zugehörigkeit, im Einheitsintervall [0,1], eines scharfen Eingangswertes zu einem linguistischen Term an. Damit wird die Überführung von präzisen bzw. scharfen Werten in die unscharfen Fuzzy-Werte realisiert, was auch als Fuzzifikation bezeichnet wird. Damit sind die Umgebungszustände f_x die scharfen Eingangswerte welche durch Zugehörigkeitsfunktionen und Regeln in den scharfen Ausgangswert Leistungsindex *p* übertragen werden.

Ein allgemeines Schaubild eines FIS ist in Bild 5.2 zu sehen. Der Kern des Systems ist die Fuzzy-Inferenz, welche ausschließlich mit Fuzzy-Werten arbeitet. Mit Hilfe der linguistischen Variablen und Terme können Experten das Wissen durch Regeln in einer Regelbasis definieren und entsprechende Zugehörigkeitsfunktionen modellieren. Die *Implikation* ist der zentrale Inferenzschritt, welcher die Bedingung der Regeln überprüft und die entsprechende Schlussfolgerung zieht. Die Schlussfolgerung jeder Regel in der Regelbasis werden durch die *Akkumulation* zusammengeführt, und schließlich überführt die *Defuzzifikation* den unscharfen Ausdruck zurück in einen präzisen Ausgangswert (z. B. eine Ventilstellung).



Bild 5.2: Allgemeiner Aufbau eines Fuzzy-Inferenzsystems.

5.1.2 Fuzzy-Inferenzprozess

Anhand eines Beispiels wird im Folgenden der Fuzzy-Inferenzprozess beschrieben. Zu beachten ist, dass für Modellierung durch Expertenwissen ein Mamdani-Typ FIS verwendet wird. Wie in Abschnitt 2.5.3 diskutiert, bietet er gegenüber anderen FIS-Typen eine ausdrucksstarke Beschreibung der Regeln, was es für Experten und Operateure einfacher macht, Wissen in das System zu bringen.
Jede Eingangsvariable $x_i = (x_1, ..., x_n)$ besitzt die Anzahl q_i an Zugehörigkeitsfunktionen $\mu_j^{(i)} = \left(\mu_1^{(i)}, \dots, \mu_{q_i}^{(i)}\right)$ und einen Wertebereich $x_r^{(i)} = x_{max}^{(i)} - x_{min}^{(i)}$. Dabei können die Zugehörigkeitsfunktionen verschiedene Formen haben, wobei Dreiecksoder Trapezfunktionen üblich sind. Generell sind alle stetigen Funktionen erlaubt, die auf das Einheitsintervall [0,1] abbilden. Im Beispiel in Bild 5.3 sind zwei Eingangsvariablen, die Nebeldichte x_1 und der Bewölkungsgrad x_2 , jeweils mit ihren drei Zugehörigkeitsfunktionen dargestellt. Verwendet wurden im Beispiel Trapezfunktionen als Zugehörigkeitsfunktionen. Die Ausgangsvariable y, der Leistungsindex, besitzt ebenfalls drei Zugehörigkeitsfunktionen, welche im Beispiel als Dreiecksfunktionen ausgeführt sind. Im Allgemeinen kann ein FIS mehrere Ausgangsvariablen $y_i = (y_1, ..., y_m)$ mit entsprechenden Zugehörigkeitsfunktionen $v_l^{(j)} = (v_1^{(j)}, \dots, v_{q_i}^{(j)})$ besitzen, wobei q_j die Anzahl der Zugehörigkeitsfunktionen angibt. Außerdem gehört zu jeder Ausgangsvariable ein Wertebereich $y_r^{(j)} = y_{max}^{(j)} - y_{min}^{(j)}$. Im Speziellen haben, die verwenden FIS lediglich eine Ausgangsvariable, den Leistungsindex, der vereinfach mit y dargestellt wird und dessen Wertebereich y_r ist. Jede Regel $R_k = (A_k, C_k)$ besitzt eine Bedingung A und eine Schlussfolgerung C, die in der Regel durch das Wort dann getrennt werden. Im Beispiel hat die Regelbasis zwei Regeln:

- *R*₁: WENN Nebeldichte ist moderat UND Bewölkungsgrad ist wolkenlos, DANN Leistungsindex ist moderat.
- *R*₂: WENN Nebeldichte ist hoch UND Bewölkungsgrad ist wolkenlos, DANN Leistungsindex ist gering.

Die erste Regel entspricht der ersten Zeile und die zweite Regel entspricht der zweiten Zeile im Bild 5.3. Für jede Regel werden die entsprechenden Zugehörigkeitsfunktionen der Eingangsvariablen ausgewertet und das Minimum (Und-Verknüpfung) der Zugehörigkeitswerte pro Regelbedingung auf die entsprechende Zugehörigkeitsfunktion der Ausgangsvariable übertragen.



Bild 5.3: Beispielhafte Visualisierung des Fuzzy-Inferenz Prozesses.

Neben der Und-Verknüpfung ist es auch möglich eine Oder-Verknüpfung (Maximum) zu verwenden. Da Und- und Oder-Verknüpfungen ineinander überführbar sind, sollen für die spätere Kodierung des Expertenwissens ausschließlich die Und-Verknüpfung verwendet werden. Nach der Regelauswertung werden die Flächen durch den ermittelten Zugehörigkeitswert der entsprechenden Zugehörigkeitsfunktionen abgeschnitten, und zusammengefasst, wie Bild 5.3 im rechten unteren Graph zeigt. Die Bestimmung des scharfen Ausgangwertes wird bei den hier verwendeten FIS durch die Berechnung des Schwerpunktes ermittelt.

5.1.3 Erfassung des Expertenwissens

Wie in Abschnitt 3.4 beschrieben, sollen vergleichend zu den ML-Verfahren zur Leistungsmodellierung die Möglichkeiten untersucht werden, Wissen, welches von Experten bzw. Entwicklungsingenieuren zur Verfügung gestellt wird, zu erfassen und ebenfalls in ein Leistungsmodell zu überführen. Damit das erfasste Expertenwissen vergleichbar ist und eindeutig beschrieben werden kann, wurde vor der Akquise des Wissens eine Taxonomie entwickelt. Diese Taxonomie nutzten die Experten bzw. den Entwicklern von den verwendeten Perzeptionsmodulen, um ihr Wissen um das Leistungsverhalten der Algorithmen darzustellen. Der folgende Abschnitt 5.1.3.1 stellt die Definition der Taxonomie vor während Abschnitt 5.1.3.2 das erfasste Wissen für jedes Perzeptionsmodul widergibt.

5.1.3.1 Definition der Taxonomie

Zur einheitlichen und interpretierbaren Erfassung des Expertenwissens, welches zur Beschreibung der Perzeptionsmodulleistung anhand der Umgebungszustände verwendet wird, wird zunächst in diesem Abschnitt eine Taxonomie eingeführt. Dazu wird zu jedem Umgebungszustand aus Abschnitt 4.2 eine Zuordnung zwischen einem Wertebereich des entsprechenden Umgebungszustandes und einer *semantischen Beschreibung* (Beschreiber) hergestellt. Der angegebene Wertebereich dient später der automatischen Erzeugung der Zugehörigkeitsfunktionen der Fuzzy-Inferenzsysteme.

Als Beschreiber werden, wie bereits in Abschnitt 4.2.1.1, verschiedene Stufen von Niederschlagsmengen, die semantisch mit *kaum, leicht, mittel, stark* und *sehr stark,* angenommen werden, verwendet. Des Weiteren wurde bereits Beschreiber für den Bewölkungsgrades, mithilfe der *Okta*-Skala, in Abschnitt 4.2.1.3 definiert. Tabelle 5.1 zeigt die entwickelte Taxonomie der Umgebungszustände. Hierbei wird jedem Beschreiber eines Umgebungszustandes eine Kennung (ID) zugeordnet, die im Folgenden zur Modellierung des Expertenwissens verwendet wird, um die entsprechenden Beschreiber zu referenzieren. Die Taxonomie des Leistungsindex in Tabelle 5.2 dargestellt.

Um achun annu aton d	Growbal	Dessbusiker	ID	Werteb	ereich
omgebungszustand	Symbol	Beschreiber	ID	Von	Bis
		sehr wenig	1	0,0	0,1
		leicht	2	0,1	0,3
Niederschlag	Q'_r	mittel	3	0,3	0,5
		stark	4	0,5	0,7
		sehr stark	5	0,7	1,0
		sehr gering	1	0,0	0,1
		gering	2	0,1	0,3
Nebeldichte	$ ho_f'$	moderat	3	0,3	0,5
		hoch	4	0,5	0,7
		sehr hoch	5	0,7	1,0
		wolkenlos	1	0,000	0,128
Bewölkungsgrad	Wo	wenig bewölkt	2	0,128	0,375
		aufgelockert bewölkt	3	0,375	0,625

Umgebungszustand		D 1 1	ID	Wertel	pereich
Umgebungszustand	Symbol	Beschreiber		Von	Bis
		durchbrochene Wolkendecke	4	0,625	0,875
		geschlossene Wolkendecke	5	0,875	1,000
		Nacht	1	22	7
		Morgen	2	7	11
Tagaggait	+	Vormittag	3	11	13
Tageszett	ι	Mittag	4	13	15
		Nachmittag	5	15	18
		Abend	6	18	22
		Frühling	1	3	5
Jahreszeit	М	Sommer	2	6	8
(Monat)	IVI	Herbst	3	9	11
		Winter	4	12	2
		orthogonal	1	-90	-80
Elevationaryinhal	0	steil	2	-80	-45
Elevationswinkei	Ø	flach	3	-45	-10
		nahe horizontal	4	-10	0
		sehr gering	1	0,0	0,1
		gering	2	0,1	0,3
(Bodonnivolauflösung)	g'	moderat	3	0,3	0,5
(bouenpixelauliosulig)		groß	4	0,5	0,7
		enorm	5	0,7	1,0
Bodenbedeckung von Wasser	+	sehr gering	1	0,0	0,2
Ebene,	$t_w, t_{os},$	gering	2	0,2	0,4
Straßen,	t_s ,	durchschnittlich	3	0,4	0,6
Vegetation,	<i>t</i> _v ,	hoch	4	0,6	0,8
Gebaude	t_b	sehr hoch	5	0,8	1,0
		flach	1	0,0	0,1
		geneigt	2	0,1	0,3
Gelandeform	r'	hügelig	3	0,3	0,5
(iteriorwert)		bergig	4	0,5	0,7
		schroff	5	0,7	1,0

Tabelle 5.1: Taxonomie der Umgebungszustände.

	Sumbol	Docabroibor	ID	Werteb	Wertebereich		
	Symbol	Beschreiber	ID	Von	Bis		
		sehr niedrig	1	0,0	0,2		
	p	niedrig	2	0,2	0,4		
Leistungsindex		moderat	3	0,4	0,6		
		hoch	4	0,6	0,8		
		sehr hoch	5	0,8	1,0		

Tabelle 5.2: Taxonomie des Leistungsindex.

5.1.3.2 Erfasstes Wissen zur Beschreibung des Leistungsverhaltens der Perzeptionsmodule

Dieser Abschnitt gibt das erfasste Wissen zur Leistungsmodellierung unter der Verwendung der vorher definierten Taxonomie wider. Hierbei wurden Experten dazu aufgefordert ihr Wissen hinsichtlich des Leistungsverhaltens ihres entwickelten Perzeptionsmoduls widerzugeben. Dabei nutzten sie zur Erläuterung der Leistungscharakteristik die Eigenschaften der Perzeptionsmodule, und ggf. Erfahrungswissen. Im Folgenden werden die einzelnen Perzeptionsmodule (ST, VG, FA, TM, BLOB, CC, DPM) separat behandelt, wobei das Perzeptionsmodul "Hypothesengenerierung mit Eckpunktdetektion" nicht mit einem Experten-Leistungsmodell hinterlegt wird und damit auch kein Expertenwissen um dieses Modul nötig ist. Es dient lediglich der finalen Hypothesengenerierung auf Basis der Straßensegmentierung oder Vordergrundsegmentierung, wie bereits in Abschnitt 3.3.2 erläutert.

Das Expertenwissen wird im Folgenden in Tabellenform dargestellt, welche in der ersten Zeile die Umgebungszustände und den Leistungsindex behandelt. In jeder weiteren Zeile befindet sich eine Regel, welche ein Teilwissen darstellt. Über die Kennungen (ID) werden die entsprechenden Bezeichner codiert. Da die Regeln zur Erzeugung der Experten-Leistungsmodelle mittels FIS herangezogen werden, nutzten diese zwischen den Regelbedingungen ausschließlich Und-Verknüpfungen, wie in vorherigem Abschnitt 5.1.2 beschrieben.

Straßensegmentierung

Tabelle 5.3 zeigt das Expertenwissen des Leistungsmodells der Straßensegmentierung. Hintergrund des Expertenwissens ist die Existenz bzw. Abwesenheit von Straßensegmenten im Sensorbild. Daher bilden die ersten 8 Zeilen das Teilwissen ab, wenn Straßensegmente im Sensorbild vorhanden sind und die Bodenoberfläche flach ist. Dabei wurde der Elevationswinkel variiert und festgestellt, dass dieser kaum Einfluss auf den Leistungsindex besitzt. Beispielsweise liest sich Regel R_5 folgendermaßen:

"WENN Elevationswinkel ist orthogonal (1) UND Ebenenanteil ist sehr gering (1) UND Vegetationsanteil ist sehr gering (1) UND Wasseranteil ist sehr gering (1) UND Straßenanteil ist sehr hoch (5) UND Gebäudeanteil ist sehr gering (1) UND Geländeform (Reliefwert) ist geneigt (2) DANN Leistungsindex ist sehr hoch (5)"

Die verbleibenden vier Regeln greifen bei Abwesenheit von Straßenanteilen in der Szene und dem damit verbundenen niedrigen Leistungsindex. Damit wird die Eigenschaft der Straßensegmentierung abgebildet, dass bei wenigen oder keinen Straßensegmenten im Bild auch keine Hypothesen für die nachfolgende Klassifikation entstehen und dadurch potentiell Fahrzeuge nicht identifiziert werden können.

R_k	Wo	θ	t _{os}	t_v	t _w	t_s	t _b	r'	p _{ST}
1	_	1	1	1	1	5	1	1	5
2	-	2	1	1	1	5	1	1	5
3	1	3	1	1	1	5	1	1	4
4	1	4	1	1	1	5	1	1	5
5	1	1	1	1	1	5	1	2	5
6	1	2	1	1	1	5	1	2	5
7	1	3	1	1	1	5	1	2	5
8	-	4	1	1	1	5	1	2	4
9	1	—	5	—	—	—	—	—	1
10	-	_	_	5	_	_	_	_	1
11		—	—	—	5	—	—	—	1
12	_	_	_	_	_	_	5	_	1

Tabelle 5.3: Regelbasis für das Leistungsmodell der Straßensegmentierung.

Vordergrundsegmentierung

Die Regelbasis des Leistungsmodells der Vordergrundsegmentierung ist in Tabelle 5.4 zu finden und kann in vier Regelgruppen eingeteilt werden:

- Die ersten fünf Regeln zeigen den Effekt der Bodenbedeckung Ebene. Bei steigendem Anteil der Kategorie Ebene wird das Vordergrundsegmentierungsverfahren zuverlässiger, da ein größerer Teil des Sensorbildes ähnliche hohe Intensitätswerte besitzt. Dies kommt dem Algorithmus zugrundeliegenden Schwellwert-Verfahren zugute, da die hellen Bereiche im Bild aussegmentiert werden.
- Einen negativen Effekt auf die Algorithmenleistung hat der zunehmende Straßenanteil (siehe Regel 6 bis 10). Je höher der Anteil der Straße im Sensorbild, umso niedriger ist der Leistungsindex. Grund dafür ist, dass Straßensegmente dunkler sind als ihre Umgebung und damit zum Vordergrund werden, sich auf der Straße befindende Fahrzeuge aber zum Hintergrund.
- Die Regeln 11 bis 14 bilden den Effekt des Elevationswinkels auf den Algorithmus ab. Bei steigendem Elevationswinkel wird immer mehr Himmel sichtbar, welcher einen Kontrast zur Bodenoberfläche im Bild erzeugt. Damit kann der Himmel zuverlässig aussegmentiert werden.
- Die letzten 5 Regeln zeigen den Einfluss des Bewölkungsgrades. Verbunden ist dieser mit dem Schattenwurf, welcher dunklere Regionen im Bild erzeugt. Da auch Fahrzeuge einen Schattenwurf besitzen, werden auch sie als Vordergrund segmentiert. Damit führt ein stärkerer Schattenwurf, also ein geringerer Bewölkungsgrad, zu einem höheren Leistungsindex.

Als Beispiel würde sich das Wissen, welches in R_{15} abgebildet ist, folgendermaßen lesen:

R _k	Wo	θ	t _{os}	t_v	t _w	t_s	t _b	r'	p_{VG}
1	-	-	1	-	_	-	-	_	3
2	-	_	2	_	_	_	-	_	3
3	_		3	—	—	—		—	4
4	—	_	4	_	_	_	-	_	4
5	_		5	—	—	—		—	5
6	_			—	—	1		—	3
7	—	_	_	_	_	2	-	_	2
8	_			—	—	3		—	1
9	-	_	_	_	_	4	-	—	1
10	-	_	_	_	_	5	-	_	1
11	_	1		—	—	—		—	3
12	—	2	_	—	_	_	-	_	5
13	_	3		—	—	—		—	5
14	_	4		—	—	—		—	5
14	1	_	_	_	_	_	-	—	5
15	2	_	_	_	_	_	-	—	5
16	3		_	_	_	_	_	_	4
17	4	_	_	_	_	_	_	—	2
18	5	_	_	_	_	_	_	_	1

"WENN Bewölkungsgrad ist *wenig bewölkt* (1) DANN Leistungsindex ist *sehr hoch* (5)"

Tabelle 5.4: Regelbasis für das Leistungsmodell der Vordergrundsegmentierung.

Farbraumanalyse

Die Regelbasis des Leistungsmodells der Farbraumanalyse zeigt Tabelle 5.5. Als Beispiel würde sich die Regel R_3 folgendermaßen lesen:

"WENN Vegetations anteil t_v ist durchschnittlich (3), DANN Leistung sindex p_{FA} ist gering (2)"

Durch empirische Versuche konnte ermittelt werden, dass ein erhöhter Anteil an Vegetation im Sensorbild die Segmentierung durch Farbraumanalyse negativ beeinflusst, da der eingesetzte Grünfilter nicht nur auf Fahrzeuge sondern auch auf Vegetation und teilweise auf Grünflächen reagiert. Damit sind diese Regionen ebenfalls Teil der Hypothesen für die Objektklassifikation und können tendenziell Fehldetektionen führen.

R _k	W_o	heta'	t _{os}	t_v	t_w	ts	t _b	r'	p_{FA}
1	-		-	1	_	_	-	_	5
2	Ι	Ι	I	2	—	—	I	_	5
3	-			3	—	—		—	2
4	-	_	-	4	—	_	-	—	1
5	_	_	_	5	_	_	_	_	1

Tabelle 5.5: Regelbasis des Leistungsmodells der Farbraumanalyse.

Kantenbasiertes Template Matching

Das kantenbasierte Template Matching nutzt zur Objektdetektion die Gradienten, die typischerweise bei Fahrzeugen im Sensorbild entstehen. Dabei können die Gradienten vor allem durch die Nebeldichte beeinflusst werden. In Tabelle 5.6 ist die Regelbasis des TM-Leistungsmodells dargestellt, wobei die ersten 5 Regeln mit dem Einfluss der Nebeldichte beschäftigten. Anzunehmen ist, dass bei steigender Nebeldichte der Leistungsindex sinkt. Um variierenden Bodenpixelauflösungen zu berücksichtigen wurden als Objekttemplates bestimmte Objektgrößen gewählt, die aber in gewissen Grenzen skaliert werden können. Daher ist anzunehmen, dass bei moderaten Bodenpixelauflösungen (Originalgröße der Objekttemplates) der höchste Leistungsindex entstehen wird. Außerhalb des moderaten Bodenpixelbereiches wird der Leistungsindex abfallen. Dieser Zusammenhang gibt die Regelbasis mit den Regeln 6 bis 10 wieder.

R_k	Wo	$ ho_f'$	heta'	g'	t	t _{os}	t_v	t_w	t _s	t _b	r'	p_{TM}
1	_	1	_	—	2	3	_	_	_	_	-	4
2	_	2	_	-	2	3	_	_	_	-	—	3
3	-	3	-	-	2	3	-	-	-	_	-	3
4	_	4	_	-	2	3	_	_	_	-	—	2
5		5		—	2	3				—	—	1
6	-	_	1	1	2	-	-	-	-	3	—	1
7	_	_	1	2	2	_	_	_	_	3	—	2
8	_	-	1	3	2	_	_	_	_	3	-	3
9	_	_	1	4	2	_	_	_	_	3	_	2
10	_	_	1	5	2	_	_	_	_	3	—	1

Tabelle 5.6: Regelbasis des Leistungsmodells des Template Matchings.

BLOB-Detektor für LWIR-Sensordaten

Tabelle 5.7 zeigt die Regelbasis des BLOB-Leistungsmodells. Da der BLOB-Detektor auf LWIR-Sensordaten arbeitet, spielt die thermische Strahlung eine zentrale Rolle. Dabei kann die thermische Strahlung der Objekte im Sensorbild durch Sonneneinstrahlung beeinflusst werden, wodurch gerade der Bewölkungsgrad als auch die Tageszeit einen Einfluss auf den Leistungsindex. In der Regelbasis ist der Einfluss des Bewölkungsgrads mit den Regeln 1 bis 5 hinterlegt. Hierbei wird angenommen, dass bei steigendem Bewölkungsgrad die thermischen Signaturen der Umgebung reduziert werden, aber Objekte mit eigener Wärmequelle (wie Fahrzeuge) weiterhin eine ausgeprägte Signatur aufweisen. Daher ist anzunehmen, dass der Leistungsindex mit steigendem Bewölkungsgrad ebenfalls steigt.

Das Verhalten bzgl. des Bewölkungsgrads lässt sich auch auf die Tageszeit übertragen. Tendenziell sollte die BLOB-Detektion über die Nacht einen höheren Leistungsindex aufweisen als während des Tages. Regel 6 bis 11 kodieren dabei das Expertenwissen über den Einfluss der Tageszeit.

R _k	W _o	θ'	t	t _{os}	t_v	t _w	t_s	t_b	r'	p_{BLOB}
1	1	_	5	-	-	_	-	-	_	2
2	2	—	5	—	—	—	—		—	2
3	3	—	5	—	—	—	—		—	3
4	4	—	5	—	—	—	—		—	4
5	5	—	5	—	—	—	—		—	5
6	_	_	1	_	_	—	_	-	—	4
7	-	—	2	—	—	—	—		—	3
8	_	—	3	—	—	—	—		—	2
9	_	_	4	_	_	—	_	-	—	2
10	_	_	5	_	_	_	_	_	—	3
11	_	_	6	_	_	_	_	_	_	4

Tabelle 5.7: Regelbasis des Leistungsmodells des BLOB-Detektors.

Klassifizierungskaskade

Für die Klassifizierungskaskade wurden Datensätze zum Training des Objektmodelles verwendet, welche Sensordaten mit -45° und -65° Elevationswinkel beinhalteten. Daher kann gefolgert werden, dass in diesem Bereich der Leistungsindex am höchsten sein wird. Tabelle 5.8 stellt die Regelbasis des CC-Leistungsmodells dar, wobei die Regeln 1 bis 4 den Einfluss des Elevationswinkels beinhalten. Daneben wird die Entfernung zum Fahrzeug einen Einfluss auf den Leistungsindex besitzen. Hierbei ist zu erwarten, dass bei geringer Entfernung zum Objekt ein hoher Leistungsindex resultiert und dieser bei steigender Distanz zum Objekt abfällt. Dieses Verhalten wurde mit den Regeln 5 bis 9 nachgebildet.

R _k	W _o	θ'	<i>g'</i>	t	t _{os}	t_v	t _w	t_s	t_b	r'	p _{CC}
1	3	1	—	—	—	_	_	-	—	-	4
2	3	2	_	_	_	_	-	-	_	-	5
3	3	3	_	—	—	—	—	—		—	5
4	3	4	—	_	_	_	_	_	-	—	2
5	1	1	1	_	_	_	-	-	_	-	5
6	1	1	2	—	_	_	_	_	-	—	4
7	1	1	3	—	—	—	—	—	1	—	3
8	1	1	4	—	—	—	—	—	1	_	1
9	1	1	5	—	—	_	_	_	-	_	1

Tabelle 5.8: Regelbasis des Leistungsmodells der Klassifikationskaskade.

DPM basierter Objektdetektor

Tabelle 5.9 zeigt die Regelbasis des DPM-Objektdetektors. Die Regeln 1 bis 5 behandeln dabei den Einfluss der Nebeldichte auf die Algorithmenleistung. Da der DPM-Detektor zur Klassifikation der Objekte das HOG-Merkmale verwendet, welches auf Basis von Gradienten erzeugt wird, wird eine Degenerierung des Leistungsindex mit zunehmender Nebeldichte erwartet. Grund dafür ist, dass Nebel den Kontrast im Sensorbild reduziert und damit die Gradienten abgeschwächt werden.

In urbanen Bereichen können die dort auftreten kantigen Strukturen zu Fehldetektionen führen. Dies ist vor allem dadurch zu begründen, dass das DPM Teilefilter verwendet, um das Objekt zu klassifizieren. Denn diese Teilefilter sprechen z. B. auf die Fahrzeugscheiben an, welche tendenziell auch bei Fenstern an Gebäuden anschlagen können. Die Regelbasis bildet dieses Verhalten mit Regel 6 bis 10 ab.

R_k	Wo	$ ho_f'$	g'	t	t _{os}	t_v	t _w	t_s	t _b	r'	р _{DPM}
1	1	1	—	2	_	_	_	_	_	—	5
2	1	2	-	2	_	_	_	_	-	—	5
3	1	3	—	2	—	—		—	1	—	4
4	1	4	—	2	_	_	-	_	-	—	2
5	1	5	—	2	—	—		—	1	—	1
6	—	—	—	4	5	—		—	1	—	5
7	_	_	_	4	4	_	1	_	2	—	5
8	_	_	_	4	3	_	-	_	3	—	4
9	—	—	—	4	2	—		—	4	—	3
10	_	_	_	4	1	_	_	_	5	_	3

Tabelle 5.9: Regelbasis des Leistungsmodells des DPM-Detektors.

5.1.4 Modellierung der Zugehörigkeitsfunktionen und Regeln

Grundlage der Modellierung der Zugehörigkeitsfunktionen ist die eingeführte Taxonomie in Abschnitt 5.1.3.1. Damit wird für jede der 13 Umgebungszustände eine linguistische Eingangsvariable erzeugt. Der Leistungsindex ist dabei die linguistische Ausgangsvariable. Die Partitionierung und Gestaltung der linguistischen Terme (Zugehörigkeitsfunktionen) erfolgt dabei auf den Angaben in Tabelle 5.1 und Tabelle 5.2. Verwendet wurden ausschließlich die Dreiecksfunktionen als Zugehörigkeitsfunktionen, die mit drei Parametern (a, b, c)beschrieben werden können:

$$f(x, a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$
(5.1)

Dabei ist *a* der linke und *c* der rechte Fuß des Dreiecks. Die Spitze ist mit *b* definiert. Außerdem muss gelten, dass $a \le b \le c$. Anhand des Umgebungszustandes *Geländeform r'* wird im Folgenden die Partitionierung und Parametrisierung der Dreiecksfunktionen beschrieben.

Für die Geländeform wurden 5 Beschreiber (linguistische Terme) definiert: *flach, geneigt, hügelig, bergig* und *schroff*. Daher besitzt die Geländeform $q_{r'} = 5$ Zugehörigkeitsfunktionen mit einem Wertebereich von [0,1]. Bei der Parametrisierung der entsprechenden Dreiecksfunktionen der linguistischen Terme wird zwischen der linken, den mittleren und der rechten Zugehörigkeitsfunktion unterschieden. In Tabelle 5.10 ist der Auszug des Umgebungszustandes Geländeform aus Tabelle 5.1 abgebildet. Hierbei ist der Bezeichner *flach* die linke Zugehörigkeitsfunktion, *schroff* die rechte Zugehörigkeitsfunktion und die übrigen sind die mittleren Zugehörigkeitsfunktionen. Der Wertebereich der einzelnen Bezeichner wird im Folgenden mit [v_l, v_r] angegeben.

	Sumbol	Dozoichnon	ID	Wertebereich			
	Symbol	Dezeichner	ID	von (v_l)	Bis (v_r)		
		flach	1	0,0	0,1		
		geneigt	2	0,1	0,3		
Geländeform (Reliefwort)	r'	hügelig	3	0,3	0,5		
(Reliefwert)		bergig	4	0,5	0,7		
		schroff	5	0,7	1,0		

Tabelle 5.10: Auszug aus der definierten Taxonomie für den Umgebungszustand Geländeform.

Die Parameter der linken Dreiecksfunktion (im Beispiel der Bezeichner *flach*) berechnen sich zu

$$b = v_l \quad c = 2v_r - b \quad a = 2b - c.$$
 (5.2)

Damit ergibt sich die vorläufige Partition für die Geländeform wie in Bild 5.4a dargestellt. Die Parameter der mittleren Dreiecksfunktionen (im Beispiel die Bezeichner *geneigt, hügelig* und *bergig*) werden wie folgt berechnet:

$$b = \frac{1}{2}(v_r + v_l) \qquad a = 2v_l - b \qquad c = 2v_r - b.$$
(5.3)

Zusammen mit der linken Dreiecksfunktion aus Bild 5.4a ergibt sich nun die Partition wie in Bild 5.4b visualisiert.



Bild 5.4: Zugehörigkeitsfunktionen des Umgebungszustandes Reliefwert.

Abschließend werden die Parameter der rechten Dreiecksfunktion mit

 $b = v_r$ $a = 2v_l - b$ c = 2b - a. (5.4)

bestimmt. Bild 5.4c zeigt die finale Partitionierung der Zugehörigkeitsfunktionen für den Umgebungszustand Geländeform. Dieses Verfahren zur Bestimmung der Partitionierung und der Parametrisierung der Zugehörigkeitsfunktionen wird für alle Umgebungszustände bis auf die Tageszeit und den Monat angewendet. Da es sich bei der Tageszeit und dem Monat um periodische Variablen handelt, müssen diese gesondert behandelt werden. Grund dafür ist eine Doppeldeutigkeit bei 0 bzw. 24 Uhr, resultierend aus der Notwendigkeit zweier unabhängigen Zugehörigkeitsfunktionen. In Abschnitt 4.2.1.4 und 4.2.1.5 wurde daher die Umwandlung in normierte kartesische Koordinaten für die ML-Methoden zur Funktionsapproximation verwendet. Dieses Verfahren wird zur Vermeidung der Doppeldeutigkeit ebenfalls für die Expertenmodellierung eingesetzt. Da die Modellierung der Tageszeit und des Monats durch zwei Variablen nicht intuitiv erscheint, wird für die Modellierung der FIS mittels Expertenwissens das folgende beschriebene Verfahren, am Beispiel der Tageszeit, verwendet.

Bild 5.5a zeigt die Darstellung der Tageszeit mit zwei kartesischen Koordinaten x_t und y_t . In dunkel Blau sind die Tageszeiten auf dem Einheitskreis angetragen. Außerdem ist der Wertebereich des Bezeichners *Nacht* [22 *Uhr*, 7 *Uhr*], aus Tabelle 5.1, mit hellblauen Punkten eingezeichnet.



Bild 5.5: Erzeugung der Zugehörigkeitsfunktion *Nacht* der Variable Tageszeit. In (a) ist am Beispiel des linguistischen Terms Nacht die Projektion auf die beiden kartesischen Koordinaten abgebildet. Die resultierenden Zugehörigkeitsfunktionen sind in (b) dargestellt.

Für jede der beiden Koordinaten x_t und y_t wird nun eine unscharfe Eingangsvariable für das FIS erzeugt und jeweils eine Zugehörigkeitsfunktion für den Term *Nacht* erstellt, wie in Bild 5.5b dargestellt. Dazu werden zunächst die minimal und maximal auftretenden Werte x_{min} , x_{max} , y_{min} und y_{max} bestimmt:

$$x_{min} = \min_{t = [v_l, v_r]} \sin\left(t\frac{2\pi}{24}\right) \qquad x_{max} = \max_{t = [v_l, v_r]} \sin\left(t\frac{2\pi}{24}\right) y_{min} = \min_{t = [v_l, v_r]} \cos\left(t\frac{2\pi}{24}\right) \qquad y_{max} = \max_{t = [v_l, v_r]} \cos\left(t\frac{2\pi}{24}\right)$$
(5.5)

Anschließend können die Parameter der Dreiecksfunktion zu

$$a_{x} = x_{min} \qquad b_{x} = \frac{1}{2}(x_{max} + x_{min}) \qquad c_{x} = x_{max}$$

$$a_{y} = y_{min} \qquad b_{y} = \frac{1}{2}(y_{max} + y_{min}) \qquad c_{y} = y_{max}$$
(5.6)

berechnet werden. Resultierend entstehen damit die in Bild 5.6 visualisierten Zugehörigkeitsfunktionen der unscharfen Eingangsvariablen x_t und y_t der Tageszeit. Bei der Modellierung der Regeln der Fuzzy-Inferenzsysteme müssen dann statt der Bedingung "Tageszeit ist Nacht" die Bedingung " x_t ist Nacht UND y_t ist Nacht" angegeben werden.



Bild 5.6: Zugehörigkeitsfunktionen der beiden Tageszeitvariablen.

In Anhang D sind für alle Umgebungszustände die erzeugten linguistischen Variablen und Terme, mit ihren Zugehörigkeitsfunktionen, zu finden. Nachdem die Zugehörigkeitsfunktionen erstellt wurden, kann das erfasste Expertenwissen aus Abschnitt 5.1.3 für jedes Perzeptionsmodul in FIS-Regeln übertragen werden. In Tabelle 5.11 ist beispielsweise die erste Regel der Regelbasis des BLOB-Detektors abgebildet. Diese wird dann übersetzt in die folgende FIS-Regel:

"WENN Nebeldichte ρ'_f ist sehr gering (1) UND Tageszeit x_t ist Morgen (2) UND Tageszeit y_t ist Morgen (2) UND Bodenbedeckung Ebene t_{os} ist durchschnittlich (3) DANN Leistungsindex p ist hoch (4)"

Die erzeugten FIS-Leistungsmodelle besitzen 15 scharfe Eingangswerte und einen Ausganswert, den Leistungsindex. Die Werte der Einzelnen Umgebungszustände müssen bei der Anwendung der Leistungsmodelle in den Wertebereichen wie in Tabelle 5.1 und Tabelle 5.2 liegen.

R _k	W _o	$ ho_f'$	$oldsymbol{ heta}'$	$oldsymbol{g}'$	t	t _{os}	t_v	t _w	t _s	t _b	r'	p
1	_	1	_	_	2	3	_	_	_	_	_	4

Tabelle 5.11: Auszug der ersten Regel aus der Regelbasis des BLOB-Detektors.

5.2 Leistungsmodellierung mittels ML-Verfahren

Bei der Leistungsmodellierung mittels der ML-Verfahren erfolgt zunächst eine Datenaufbereitung, welche im folgenden Abschnitt erläutert wird. Anschließend wird der NEFPROX-Algorithmus vorgestellt, welcher anhand der generierten *Metadaten* die Zugehörigkeitsfunktionen und Regeln eins FIS ermittelt und daraus Leistungsmodelle entwickelt. Abschließend wird das Verfahren zur Erzeugung der Leistungsmodelle mittels künstlicher neuronaler Netze präsentiert.

5.2.1 Datenaufbereitung

Die Datenaufbereitung hat das Ziel, redundante Informationen aus den Metadaten zu die Daten für eliminieren und das Training der Leistungsmodelle durch Funktionsapproximation aufzubereiten. In Bild 5.7 das Ablaufschema ist der Datenaufbereitung zu sehen.



Bild 5.7: Verfahren der Datenaufbereitung und Modellbildung.

Der erste Schritt der Datenaufbereitung ist die Vorverarbeitung der Umgebungszustände. Dabei werden die initialen Umgebungszustände in $\mathcal{D}_r^{(i)}$ nach den in Abschnitt 4.2 definierten Berechnungsvorschriften normiert oder umgewandelt, damit die Verfahren zur Funktionsapproximation eine schnellere Konvergenz beim Erzeugen der Leistungsmodelle besitzen. Außerdem wird eine Übergewichtung einzelner Umgebungszustände unterbunden, indem sich die Wertebereiche aller Variablen zwischen -1 und 1 bewegen. Die vorverarbeiteten Umgebungszustände werden zusammen mit den originalen *Scores* und *Labels* (vgl. Abschnitt 4.7) in \mathcal{D}_p abgelegt. Zusätzlich, wie in Abschnitt 4.2.2.2 und 4.2.3.2 vorgestellt, werden die beiden Faktoren f_g und f_r zur Normierung der mittleren Bodenpixelauflösung und des Reliefwerts berechnet.

Auf die Vorverarbeitung folgen zwei weitere Schritte, die *Bias-Bestimmung* und die *Clusterbildung*. Bei der Bias-Bestimmung wird die ROC-Kurve berechnet, um den optimalen Bias und die Fläche unter der Kurve (engl. *area under the curve*, AUC) zu bestimmen. Dabei ist der optimale Bias der Schwellwert, der entscheidet, ob der ermittelte *Score* eines Objektdetektors der Klasse "Fahrzeug" oder "kein Fahrzeug" zugeordnet wird. Durch die Variierung des Schwellwertes entsteht die ROC-Kurve, welche die Sensitivität über die Ausfallrate aufträgt, die mit den Formeln in Bild 5.8a bestimmt werden. Der optimale Bias ist der Kompromiss zwischen Sensitivität und Ausfallrate, welcher visuell durch den Kontaktpunkt einer 45° Tangente (optimaler Operationspunkt) an die ROC-Kurve bestimmt wird. Dies ist in der Grafik in Bild 5.8b dargestellt. In der Grafik sind zudem der Punkt mit dem

optimalen Bias, sowie die AUC eingezeichnet. Der optimale Bias wird verwendet, um im Anschluss an die Clusterbildung den Leistungsindex zu bestimmen.



Bild 5.8: Berechnung und Visualisierung einer ROC-Kurve. In (a) sind die Formeln für die Berechnung der Sensitivität und der Ausfallrate zu sehen und (b) zeigt die ROC-Kurve, dunkelblaue Linie, mit optimalen Operationspunkt und AUC (hellblaue Fläche).

Die Clusterbildung sucht in den vorverarbeiten Trainingsdaten \mathcal{D}_p nach identischen Umgebungszuständen, welche zu einem Cluster gebündelt werden können. In der darauffolgenden Berechnung des Leistungsindex werden alle Scores in einem Cluster mit dem vorher bestimmten optimalen Bias einer Klasse zugeordnet. Durch die Definition des Leistungsindex (siehe Abschnitt 4.6) wird daraus mittels der Wahrheitsmatrix der Leistungsindex berechnet. Das Ergebnis aus den Daten eines Clusters ist ein Umgebungszustand f_x mit Leistungsindex p. Damit ergeben sich aus den Cluster-Daten \mathcal{D}_c die aufbereiteten Modelldaten \widetilde{D} für die Erzeugung der Leistungsmodelle durch Funktionsapproximation. Dabei wird für die Perzeptionsmodule BLOB, TM, CC und DPM jeweils ein Leistungsmodell M_i^{NEF} mit dem NEFPROX-Algorithmus und ein Modell M_i^{KNN} durch ein künstliches neuronales Netz erzeugt. Die Beschreibungen der beiden Verfahren finden sich in den folgenden beiden Abschnitt.

5.2.2 Leistungsmodellierung mit dem NEFPROX-Algorithmus

Neben der Modellierung durch Experten lassen sich die Zugehörigkeitsfunktionen und Regelbasis eines Fuzzy-Inferenzsysteme vom Mamdani-Typ mit geeigneten Trainingsdaten auch lernen. Nauck und Kruse [129] veröffentlichten den NEFPROX (engl. *NEuro Fuzzy function apPROXimator*) Algorithmus zum Lernen der Wissensbasis eines Fuzzy-Inferenzsystems zur Funktionsapproximation durch überwachtes Lernen. Dabei gehört dieser Algorithmus den Neuro-Fuzzy Methoden an, welche an Anlehnung der Lernverfahren für künstliche neuronale Netze die Parameter eines FIS bestimmen. Generell wird dabei ein FIS durch ein spezielles neuronales Netz [146] abgebildet. Im Folgenden wird der NEFPROX-Algorithmus vorgestellt, wobei die wichtigsten Schritte erläutert und Anpassungen gegenüber der originalen Beschreibung des NEFPROX-Algorithmus in [146] beschrieben werden.

In Bild 5.9 sind die Prozessschritte des NEFPROX Algorithmus zu sehen. Mittels der *Metadaten* $\{F, P\} = \tilde{D} = \tilde{L} \cup \tilde{G} \cup \tilde{T}$ wird die Wissensbasis (Regeln und Zugehörigkeitsfunktionen) eines FIS erzeugt, wobei \tilde{L} dem Trainingsset, \tilde{G} dem Validierungsset und \tilde{T} dem Testset entspricht. Dabei wurden die Metadaten (Trainingsdaten) in den Standardverhältnissen 60%, 20% und 20% auf das Trainings-, Validierungs- und Testset aufgeteilt [147]. Ein Trainingselement (\vec{p}, t) in den Metadaten besteht aus dem Eingangsvektor \vec{p} , der dem Umgebungszustand f_x entspricht, und dem Ausgangswert t, welcher dem Leistungsindex p entspricht.



Bild 5.9: Überblick über die Prozessschritte des NEFPROX-Algorithmus. Der Algorithmus unterteilt sich in zwei Hauptkomponenten, das Lernen der Regelbasis und der Zugehörigkeitsfunktionen, jeweils umrahmt mit den blauen Rechtecken. Der Startschritt in jeder Hauptkomponente ist mit dem Asterisk Zeichen markiert.

Der nachfolgende Abschnitt 5.2.2.1 behandelt zunächst die Umsetzung des Lernens der Regelbasis aus den *Metadaten*. Anschließend behandelt Abschnitt 5.2.2.2 die Implementierung zum Lernen der Zugehörigkeitsfunktionen. Schließlich zeigt Abschnitt 5.2.2.3 welche Parameter für die Leistungsmodellierung mittels NEFPROX-Algorithmus gewählt wurden.

5.2.2.1 Lernen der Regelbasis

Der erste Prozessschritt beim Lernen der Regelbasis ist die Partitionierung der Zugehörigkeitsfunktionen für die Eingangsvariablen. Der Parameter q_i legt fest, wie viele Zugehörigkeitsfunktionen pro Eingangsvariable erzeugt werden sollen. Über die Trainingsdaten werden die Wertebereiche der Eingangsvariablen bestimmt. Dabei wird pro Eingangsvariable der minimale bzw. maximale Wert in den Trainingsdaten gesucht und die Standardabweichung verwendet, um die Randbereiche auszudehnen. Dies sorgt für eine initiale bessere Abdeckung der Trainingsdaten durch die Zugehörigkeitsfunktionen. Formal bestimmt sich der Wertebereich durch:

$$x_{min}^{(i)} = \min(\widetilde{\mathcal{D}}_i - \sigma_i) \qquad x_{max}^{(i)} = \max(\widetilde{\mathcal{D}}_i + \sigma_i)$$
(5.7)

Der Wertebereich für die Ausgangsvariable wird in identischer Weise bestimmt. Anschließend erfolgt die Erzeugung der Zugehörigkeitsfunktionen. Dabei werden in dieser Arbeit zwei Partitionierungsmethoden verwendet und damit zunächst zwei initiale Fuzzy-Inferenzsysteme erzeugt. Die beiden Partitionierungsarten sind in Bild 5.10 dargestellt und zählen zu den Grid- Partitionierungsarten. Im Folgenden werden ausschließlich Dreiecksfunktionen betrachtetet. Die erste Variante, in Bild 5.10a abgebildet, erzeugt symmetrische Dreiecksfunktionen, wobei sie gleichmäßig über den Wertebereich verteilt werden und keine Zugehörigkeitsfunktion außerhalb des Wertebereichs größer als 0 ist. In der zweite Variante (siehe Bild 5.10b) werden ebenfalls q_i symmetrische Dreiecksfunktionen erzeugt, wobei die Spitzen der äußersten Zugehörigkeitsfunktionen an den Rändern platziert und alle übrigen Funktionen dazwischen gleichmäßig verteilt werden. Die Dreiecksfunktionen schneiden sich beim Zugehörigkeitswert von 0,5. An diesen Stellen entstehen die Trennlinien und bilden das Grid, welches in Bild 5.10 durch gestrichelte Linien dargestellt ist. Mithilfe des Grids kann für jede Dimension des Trainingselements eine Zugehörigkeitsfunktion zugeordnet werden. Durch dieses Verfahren lassen sich die Regelbedingungen ermitteln.



Bild 5.10: Partitionierungsmethoden der Zugehörigkeitsfunktionen der Eingangsvariablen. In den Beispielen werden jeweils drei Zugehörigkeitsfunktionen verwendet. In (a) sind alle Zugehörigkeitsfunktionen innerhalb des Wertebereiches und in (b) werden deren Zentren gleichmäßig über den Wertebereich verteilt.

Für alle Eingangsvektoren in $\tilde{\mathcal{L}}$ werden die Regelbedingungen erzeugt, wenn diese noch nicht in der Regelbasis existieren. Dabei werden die einzelnen Bedingungen pro Dimension des Eingangsvektors durch das Maximum des Zugehörigkeitswertes selektiert:

$$\mu_{j_i}^{(i)} = \operatorname*{argmax}_{\mu_j^{(i)}, j \in \{1, \dots, q_i\}} \left(\mu_j^{(i)}(p_i) \right)$$
(5.8)

Daraus entsteht schließlich die Bedingung $A = (\mu_{j_1}^{(1)}, ..., \mu_{j_n}^{(n)})$. Dieses Verfahren ist in Bild 5.10 durch das Trainingsbeispiel p angedeutet, indem die zugehörigen Dreiecksfunktionen durch die grauen Linien markiert sind. Wenn alle Eingangsvektoren verarbeitet sind, werden die Regelschlussfolgerungen durch den Mittelwert des Grads der Zugehörigkeit der Regelbedingung ermittelt. Da nur Regeln mit der Und-Verknüpfung erzeugt werden, bestimmt sich der Grad der Zugehörigkeit einer Regel mit $R_k(\vec{p}) = A_k(\vec{p}) = \min_{i=1,...,n} \mu_{j_i}^{(i)}(p_i)$ und damit der Mittelwert pro Regel zu

$$\bar{t_k} = \frac{\sum_{(\vec{p},t)\in\tilde{\mathcal{L}}} (A_k(\vec{p})\cdot t)}{\sum_{(\vec{p},t)\in\tilde{\mathcal{L}}} A_k(\vec{p})}.$$
(5.9)

Anschließend wird iterativ für alle Regelbedingungen geprüft, ob es bereits eine Zugehörigkeitsfunktion v_l gibt, die $v_l(\bar{t_k}) > 0.5$ erfüllt. Wenn ja, dann ist die Regelschlussfolgerung die entsprechende Zugehörigkeitsfunktion v_l mit dem maximalen Grad der Zugehörigkeit. Ansonsten wird eine neue Zugehörigkeitsfunktion generiert, wobei $v_l(\bar{t_k}) = 1$ ist. Bevor die erzeugte Regelbasis optimiert wird, werden die beiden erzeugten Systeme am Trainingsset $\tilde{\mathcal{L}}$ evaluiert, indem der mittlere quadratische Fehler (engl. *mean squared error*, MSE) zwischen Ausgang der FIS $o(\vec{p})$ und dem Trainingsbeispiel t berechnet wird. Das Fuzzy-Inferenzsystem mit dem kleineren Fehler wird behalten, das andere verworfen.

Die Optimierung der Regelbasis erfolgt durch Einführung einer Leistungsmetrik P_k für eine Regel R_k . Sie wird bestimmt aus dem Mittelwert und der Varianz des Grads der Zugehörigkeit aller Trainingsbeispiele:

$$\overline{t_k} = \frac{\sum_{(\vec{p},t)\in\tilde{\mathcal{L}}}(R_k(\vec{p})\cdot t)}{\omega_k} \quad mit \ \omega_k = \sum_{(\vec{p},t)\in\tilde{\mathcal{L}}}R_k(\vec{p})$$
$$\operatorname{var}(R_k) = \frac{\sum_{(\vec{p},t)\in\tilde{\mathcal{L}}}(R_k(\vec{p})\cdot (t-\overline{t_k})^2)}{\omega_k}$$
$$P_k = \frac{y_r \cdot \omega_k}{|\tilde{\mathcal{L}}| \cdot \left(2\sqrt{\operatorname{var}(R_k)} + y_r\right)}$$
(5.10)

Dabei ist $|\tilde{L}|$ die Anzahl der Trainingsbeispiele im Trainingsset und y_r ist der Wertebereich der Ausgangsvariable *Leistungsindex*. Regeln, die mehr Trainingsbeispiele abdecken als andere Regeln, und damit eine höhere Relevanz haben, erhalten einen höheren Wert für P_k . Damit lässt sich die Regelbasis absteigend nach P_k sortieren. Durch den Parameter ρ_{ω} wird angegeben, wie viel Prozent der sortierten Regeln in die finale Regelbasis übernommen werden. Hierzu wird beginnenden mit der ersten Regel der Wert ω_k über die Folgenden aufsummiert. Mit Erreichen des Schwellwerts $\rho_{\omega} \cdot \sum_k \omega_k$, werden alle verbleibenden Regeln aus der Regelbasis entfernt. An dieser Stelle existiert nun ein fertiges FIS, wobei die Zugehörigkeitsfunktionen noch nicht angepasst wurden. Dies geschieht in der zweiten Algorithmus-Komponente, wie in Bild 5.9 dargestellt.

5.2.2.2 Lernen der Zugehörigkeitsfunktionen

Das Lernern der Zugehörigkeitsfunktionen basiert auf dem Grundprinzip der Fehlerrückführung (engl. *Backpropagation*) bei KNN. Im ersten Schritt, dem Vorwärtszweig, wird ein Trainingsbeispiel durch das Fuzzy-Inferenzsystem propagiert, und am Ausgang wird der berechnete Wert mit dem Ausgangswert des Trainingsbeispiels verglichen. Der Unterschied ist der Fehler Δy :

$$\Delta y = t - o(\vec{p}) \tag{5.11}$$

Dieser Fehler gibt zum einen die Stärke der Abweichung und zum anderen die Richtung an. Der Fehler wird verwendet, um im Rückwärtsschritt die Parameter der Dreiecksfunktionen (a,b,c)anzupassen. Zunächst erfolgt die Berechnung des Updates der Regelschlussfolgerungen. Dabei wird für jedes Trainingsbeispiel in $\tilde{\mathcal{L}}$ und jede Regel in der Regelbasis geprüft, ob der Grad der Zugehörigkeit der Regel $R_k(\vec{p}) > 0$ ist. Wenn dies der Fall ist, dann bedeutet es, dass diese Regeln, für das aktuelle Trainingsbeispiel, dazu beitragen, den Ausgangswert $o(\vec{p})$ zu bestimmen. Die Verschiebungen für die entsprechenden Zugehörigkeitsfunktionen der Schlussfolgerung bestimmen sich für $v_k(t) > 0$ zu

$$\Delta \mathbf{b} = \lambda \cdot \Delta \mathbf{y} \cdot (c - a) \cdot R_k(\vec{p}) \cdot (1 - \nu_k(t))$$

$$\Delta a = \Delta \mathbf{b} + \lambda \cdot (b - a) \cdot R_k(\vec{p})$$

$$\Delta \mathbf{c} = \Delta \mathbf{b} - \lambda \cdot (c - b) \cdot R_k(\vec{p}).$$
(5.12)

Dabei ist λ die Lernrate und die Zugehörigkeitsfunktion der Schlussfolgerung der Regel R_k ist v_k . Die Idee hinter dieser Verschiebungsberechnung ist, dass für Dreiecksfunktionen, die bereits den gewünschten Ausgangswert abdecken, die Funktionsmitte (Spitze des Dreiecks) mittels des Fehlers zum gewünschten Ausgangswert bewegt wird, um dadurch den Fehler zu minimieren. Zur zusätzlichen Fehlerminimierung werden die Seiten des Dreiecks geschmälert, damit die Beeinflussung von außerhalb liegenden Trainingsbeispielen kleiner wird. Dazu ist in Bild 5.11 ein Beispiel mit der dunkelblauen Funktion v_2 dargestellt. Anderenfalls, wenn $v_k(t) \leq 0$, bestimmen sich die Verschiebungen von (a, c) zu

$$\Delta a = \Delta b + \operatorname{sgn}(t - b) \cdot \lambda \cdot (b - a) \cdot R_k(\vec{p})$$

$$\Delta c = \Delta b + \operatorname{sgn}(t - b) \cdot \lambda \cdot (c - b) \cdot R_k(\vec{p}).$$
(5.13)

Hier ist der Ansatz, dass sich die gesamte Funktion in Richtung des gewünschten Ausgangswertes bewegt. Dabei wird die Seite der Dreiecksfunktion, die Richtung *t* zeigt, vergrößert und die andere verkleinert, um möglichst schnell den gewünschten Ausgangswert zu erreichen. Dargestellt ist diese Anpassung der Dreiecksfunktion in Bild 5.11 mit Funktion v_1 .



Bild 5.11: Update der Zugehörigkeitsfunktionen der Regelschlussfolgerung. In (a) ist der Ausgangszustand der Dreiecksfunktionen zu sehen und (b) zeigt schematisch die Verschiebung der Funktionen durch den Update-Schritt.

Die Verschiebungen bzw. Deltas der Schlussfolgerungen, resultierend aus allen Trainingsset-Beispiele, werden zunächst nicht ausgeführt, sondern erst im letzten Schritt zusammen mit den Änderungen der Zugehörigkeitsfunktionen der Regelbedingungen angewendet.

Die Adaption der Regelbedingung wird durch die Propagierung des Ausgangsfehlers zurück zur Regel mit dem Regelfehler Δr geführt. Er bestimmt sich zu:

$$\Delta r = R_k(\vec{p}) \cdot \left(1 - R_k(\vec{p})\right) \cdot (2\nu_k(\vec{p}) - 1) \cdot |\Delta y|$$
^(5.14)

Dabei entscheidet $v_k(\vec{p})$, ob die Regel einen positiven > 0,5 oder einen negativen \leq 0,5 Einfluss auf den Ausgangsfehler hat. Somit lässt sich bestimmen, ob der Grad der Zugehörigkeit der entsprechenden Zugehörigkeitsfunktionen vergrößert oder verkleinert werden muss. Die Verschiebungen der Funktionen berechnen sich mit folgenden Formeln:

$$\Delta b = \delta \cdot \Delta r \cdot (c-a) \cdot \operatorname{sgn}(p-b) \quad mit \ \delta = \begin{cases} \lambda \cdot \mu(p) & \text{wenn } \Delta r < 0, \\ \lambda \cdot (1-\mu(p)) & \text{sonst,} \end{cases}$$

$$\Delta a = \Delta b + \delta \cdot \Delta r \cdot (b-a) \\ \Delta c = \Delta b + \delta \cdot \Delta r \cdot (c-b) \end{cases}$$
(5.15)

Hierbei ist $\mu(p)$ der Grad der Zugehörigkeit der betreffenden Zugehörigkeitsfunktionen in A_k . Die Verschiebungen bzw. Deltas werden für alle Trainingsbeispiele und Regeln berechnet und aufsummiert. In Bild 5.12 ist die Verschiebung einer Zugehörigkeitsfunktion der Eingangsvariablen zu sehen.



Bild 5.12: Update der Zugehörigkeitsfunktionen der Regelbedingung. Die Ausgangsfunktion ist die mittlere dunkelblaue Funktion. Bei einer Vergrößerung des Grads der Zugehörigkeit wird die Funktion zum Trainingsbeispiel bewegt (hellblaue Funktion) und bei Verkleinerung nach rechts zur blauen Funktion.

Nachdem alle Deltas berechnet wurden, werden alle Zugehörigkeitsfunktionen aktualisiert. Dabei werden die Deltas zu den Parametern (a, b, c) addiert. Anschließend muss gewährleistet werden, dass für die Parameter der Dreiecksfunktionen weiterhin $a \le b \le c$ gilt. Außerdem dürfen die Zugehörigkeitsfunktionen der Eingangsvariablen ihre Reihenfolge nicht tauschen. Dies bedeutet, dass die Spitzen der Dreiecksfunktionen sich nicht passieren dürfen. Diese Beschränkungen garantieren eine weitere Interpretierbarkeit der Regelbasis nach dem Lernen der Zugehörigkeitsfunktionen. Nach der Aktualisierung beginnt der Lernzyklus von neuem, wobei die Lernrate bei Verbesserung des FIS gegenüber dem Validierungsset vergrößert und bei Verschlechterung verkleinert wird. Dabei führt eine größere Lernrate zu schnelleren Adaptionen der Zugehörigkeitsfunktionen, kann aber auch zum Verfehlen des Optimums führen. Eine kleine Lernrate reduziert die Adaptionsschrittweite und ermöglicht das gezielte Ermitteln eines Optimums, welches aber auch ein lokales und nicht das globale sein kann. Der Lernzyklus wird beendet, wenn

- entweder eine maximale Anzahl an Lernzyklen (Epochen) erreicht ist,
- sich über eine gewisse Anzahl an Epochen keine Verbesserung eingestellt hat,
- oder wenn der MSE gegenüber dem Validierungsset unter einer gewissen Fehlertoleranz liegt.

Final wird die Güte des gelernten Fuzzy-Inferenzsystems anhand des Testsets $\tilde{\mathcal{T}}$ über den MSE bestimmt.

5.2.2.3 Festlegung der NEFPROX-Hyperparameter

Für den NEFPROX-Algorithmus werden feste Hyperparameter verwendet, damit die Regelbasis leserlich bleibt, Wissen extrahiert werden kann und die Latenz des Inferenzprozesses reduziert wird. Tabelle 5.12 zeigt die verwendeten Hyperparameter des NEFPROX-Algorithmus. Die resultierenden NEFPROX-Leistungsmodelle mit ihren Modellparametern zeigt Tabelle 5.13.

Hyperparameter	Parameterwert
Zugehörigkeitsfunktion	Dreiecksfunktion
Anzahl Zugehörigkeitsfunk. q_i	5
Lernrate λ	0,1
Max. Anzahl Epochen e_{max}	100
Fehlerzuwachs n_e	50 Epochen
Bester Anteil der Regelbasis $ ho_\omega$	70 %

Tabelle 5.12: Verwendete NEFPROX-Hyperparameter zur Leistungsmodellierung.

Modellparameter	M_{BLOB}^{NEF}	M_{TM}^{NEF}	M_{CC}^{NEF}	M_{DPM}^{NEF}
Anzahl der Regeln	4135	4128	3472	3465
m 1 11 E 40	N 1 11 ·			1 11

Tabelle 5.13: Modellparameter der erzeugten NEFPROX-Leistungsmodelle.

Bild 5.13 zeigt eine Darstellung des Leistungsmodells, welches über den NEFPROX-Algorithmus erzeugt wurde. Mittels der festgelegten Hyperparameter und den Metadaten werden die Zugehörigkeitsfunktionen und Regeln des FIS über den NEFPROX-Algorithmus erzeugt. Bei der Anwendung des Leistungsmodells wird dem FIS der Umgebungszustand f_x präsentiert, wodurch über den regulären Fuzzy-Inferenzprozess (vgl. Abschnitt 5.1.2) der Leistungsindex berechnet wird.



Bild 5.13: Darstellung des Leistungsmodells erzeugt über den NEFPROX-Algorithmus.

5.2.3 Leistungsmodellierung mittels künstlicher neuronaler Netze

Als Alternative zum NEFPROX-Algorithmus wird in diesem Abschnitt der Erzeugung der Leistungsmodelle mittels künstlicher neuronaler Netze beschrieben (vgl. Abschnitt 3.3.3.4). Dabei behandelt der Abschnitt 5.2.3.1 zunächst im Allgemeinen die Realisierung einer Funktionsapproximation mit neuronalen Netzen, wie sie in dieser Arbeit verwendet wurde, und Abschnitt 5.2.3.2 zeigt mit welchen Parametern und Optimierungen die Leistungsmodelle erzeugt wurden.

5.2.3.1 Einsatz zur Funktionsapproximation

Ein weiteres Verfahren zur Funktionsapproximation sind künstliche neuronale Netze (KNN). Sie haben zwar die Nachteile, dass sie nicht durch Expertenwissen beschrieben und die gelernten Informationen nicht direkt ausgelesen werden können, aber dafür sind sie geeignet, unbekannte kontinuierliche Funktion zu approximieren. Im Folgenden wird kurz vorgestellt, wie ein neuronales Netz aufgebaut ist und welche Methoden und Funktionen verwendet werden, um die Modellierung der Leistungscharakteristik der Perzeptionsmodule zu realisieren.

Ein künstliches neuronales Netz besteht aus Neuronen, die einen oder mehrere Eingänge und einen Ausgang besitzen. Dabei wird ein Neuron aktiviert, wenn die Summe der Eingänge einen bestimmten Schwellwert überschreitet bzw. unterschreitet. Mit Aktivierung ist dabei gemeint, dass der Ausgangswert gegen 1 bzw. -1 läuft. Die allgemeine Formel, um den Ausgangswert y eines Neurons zu berechnen, lautet wie folgt:

$$y = \sigma(w \cdot x + b) \quad mit \ w \cdot x = \sum_{j=1}^{n} w_j x_j \tag{5.16}$$

Hierbei ist x_j ein Eingangswert, w_j das zugehörige Gewicht, b der Schwellwert bzw. Bias und σ die Aktivierungsfunktion. Das Gewicht gibt dabei die Relevanz eines Eingangs auf den Ausgang an. Die Aktivierungsfunktion überführt die Aktivierungen der Eingänge auf den Ausgang des Neurons. Sie ist vor allem wichtig, um neuronale Netze zu trainieren. Denn um den Fehler des Netzes zu korrigieren, wird die Aktivierungsfunktion abgeleitet, um die Korrekturrichtung zu bestimmen, in der die Gewichts- und Bias-Werte verändert werden müssen. Außerdem beeinflusst sie, wie schnell ein Netz konvergiert und saturiert, und beeinflusst damit die Trainingszeit.



In Bild 5.14 ist ein Neuron und die in dieser Arbeit verwendete Aktivierungsfunktion, der Tangens hyperbolicus, zu sehen. Das abgebildete Neuron verbildlicht die Formel (5.16).

Bild 5.14: Graphische Darstellung eines Neurons und Tangens hyperbolicus Aktivierungsfunktion. In (a) ist das Neuron zu sehen und (b) zeigt die Aktivierungsfunktion.

Ein künstliches neuronales Netz besteht mindestens aus einer Eingangsschicht, einer Ausgangsschicht und meistens auch aus einer oder mehreren versteckten Schichten *l*. Dabei ist jedes Neuron mit allen Neuronen der Vorgängerschicht verbunden. Ein solches künstliches neuronales Netz ist in Bild 5.15 mit zwei versteckten Schichten illustriert.



Bild 5.15: Beispiel eines künstlichen neuronalen Netzes mit zwei versteckten Schichten.

Für die Eingangsschicht gilt der Spezialfall, dass die Neuronen lediglich den Wert der entsprechenden Eingangsvariable ausgeben. Die Parameter der Neuronen werden üblicherweise mit der Methode der Fehlerrückführung (engl. *backpropagation*) gelernt. Für den Fall der Regression besitzt ein Netz beliebig viele Eingänge, hier also die Anzahl der Umgebungszustände, und einen Ausgang, welcher in dieser Arbeit der Leistungsindex ist. So wie bei der NEFPROX-Methode werden die *Metadaten* als Trainingsdaten $\tilde{D} = \tilde{L} \cup \tilde{G} \cup \tilde{T}$ verwendet, um die Parameter des Netzes zu lernen. Dabei gibt es verschiedene Methoden, wie die Fehlerbestimmung des Backpropagation-Algorithmus ausgeführt ist. Die einfachste Methode ist das Gradientenverfahren, welches über die Ableitung den steilsten Gradienten der Fehlerfunktion findet, um den Netzfehler zu minimieren. Typisch für dieses Verfahren ist, dass es sich mit einem Zickzack-Kurs dem Optimum nähert und im Vergleich zu anderen Methoden langsam ist. Eine verbesserte Variante ist die der konjugierten Gradienten, die den Gradienten

des vorherigen Schrittes berücksichtigen. Bild 5.16 zeigt die beiden Methoden im Vergleich, wobei das Optimum in der Mitte der Grafik ist.



Bild 5.16: Schematischer Vergleich zwischen dem Gradienten- und konjugierten Gradientenverfahren. Der hellblaue Kurs repräsentiert die Annäherung an das Optimum des Gradientenverfahrens und der dunkelblaue Kurs zeigt im Vergleich dazu das konjugierte Gradientenverfahren.

Eine weitere Beschleunigung des konjugierten Gradientenverfahrens wird durch die Skalierung der Schrittweite (Lenrate λ) erreicht, indem nicht entlang des Gradienten durch ein Liniensuchverfahren das Minimum gesucht wird, sondern dieses durch einen veränderlichen Skalierungsfaktor pro Lerniteration verändert wird, um sich dem Optimum anzunähern. Diese Methode wird auch als SCG (engl. *scaled conjugate gradient*) [148] bezeichnet und wird in dieser Arbeit zur Fehlerbestimmung herangezogen.

5.2.3.2 Parametrisierung und Optimierung

Zur Optimierung der KNN-Leistungsmodelle werden diese mehrfach erzeugt. Implementiert ist die Stoppbedingung durch eine feste Anzahl (N = 100) zu trainierenden Netzen. Jedes Netz wird mit einer unterschiedlichen Anzahl versteckter Schichten l und Neuronen pro versteckte Schicht n_l erzeugt, welche durch Zufall generiert werden, mit l = U(1,3) und $n_l = U(10,200)$. Aus diesen wird das Netz mit der kleinsten MSE ausgewählt. Eine Zusammenstellung der Hyperparameter ist in Tabelle 5.14 zu finden.

Hyperparameter	Parameterwert
Max. Anzahl Epochen e _{max}	10000
Fehlerzuwachs n_e	50 Epochen
Lernrate λ	0,01
Anzahl versteckter Schichten <i>l</i>	1 - 3
Anzahl der Neuronen n_l	10 - 200
Aktivierungsfunktion	Sigmoid
Fehlerrückführung	SCG

Tabelle 5.14: Verwendete Hyperparameter zur Leistungsmodellierung mittels KNN.

Dabei gibt e_{max} die maximale Anzahl an Epochen an, die das Training der Netze absolvieren darf, bevor das Training beendet wird. Der Fehlerzuwachs n_e gibt an, wie viele Epochen vergehen dürfen, in denen keine Verbesserung des Netzes erreicht wird, bevor das Training abgeschlossen wird und die Lernrate λ legt die initiale Schrittweite für das SCG fest.

Ein Beispiel eines KNN-Leistungsmodells zeigt Bild 5.17. Die Eingangsschicht besteht aus 15 Neuronen, welche der Anzahl der Umgebungszustände f_x entspricht. Die Ausgangsschicht enthält lediglich ein Neuron, welches den Leistungsindex p ausgibt. Zwischen Ein- und Ausgangsschicht befinden sich 1 – 3 versteckte Schichten mit jeweils 10 – 200 Neuronen. Bei der Verwendung der KNN-Leistungsmodell muss darauf geachtet werden, dass die Eingangsdaten (Umgebungszustand) die bereits normierten Werte (siehe Abschnitt 4.2) enthalten. Modellparameter

···· F····	BLUB		DFM	1 M
Verstecke Schichten <i>l</i>	2	3	3	3
Anzahl der Neuronen pro Schicht	165-102	173-122-111	140-177-66	182-154-185
Tabelle 5.15: Moo	dellparameter der er	zeugten KNN-Lei	stungsmodelle.	
Eingangsschicht	Versteckte Sc	hichten [1 – 3]	Ausgangsso	chicht
Niederschlag				
Nebel dichte				
Bewölkungsgrad ——				
Tageszeit $x_t \longrightarrow$				
Tageszeit $y_t \longrightarrow$				
Monat x_M \longrightarrow				
Monat $y_M \longrightarrow$) - 2(0 - 2(
Elevations winkel	n [1(n [10		• Leistungsindex
Bodenpixelauflösung	Irone	rone		
Wasser —	Neu	Net		
Ebene				
Straßen ——				
Vegetation				
Gebäude				
Reliefwert				

Tabelle 5.15 zeigt die Modellparameter, die beim Erzeugen der KNN-Leistungsmodelle entstanden sind.

M^{KNN}

M^{KNN} DDM M_{TM}^{KNN}

MRIOR

Bild 5.17: Darstellung des Leistungsmodells erzeugt mittels künstlicher neuronaler Netze. Aus Darstellungsgründen sind die Verbindungen zwischen den Neuronen nicht eingezeichnet. Im den erzeugten KNNs sind immer alle Neuronen zwischen zwei Schichten miteinander verbunden.

KAPITEL 6 VERFAHREN ZUR ALGORITHMENAUSWAHL

Dieses Kapitel beschreibt das verwendete Verfahren zur Algorithmenauswahl. Dazu stellt Abschnitt 6.1 die prozeduralen Details der Auswahl der Perzeptionsketten vor. Anschließend wird das in Abschnitt 2.4.3 vorgestellte Konzept des SPMS erweitert, um die Algorithmenauswahl in dieses zu integrieren (siehe Abschnitt 6.2). Abschließend erfolgt die Beschreibung der Implementierung des um die Algorithmenauswahl erweiterte SPMS in Abschnitt 6.3.

6.1 Auswahl der Perzeptionsketten

Im vorherigen Kapitel wurde beschrieben, wie die Leistungsmodelle erzeugt wurden. Sie werden nun verwendet, um die eigentliche Algorithmenauswahl (Auswahl der Perzeptionsketten) zur Laufzeit der Perzeptionssystems zu realisieren. Dazu stellte bereits Abschnitt 3.2.2 das übergeordnete Konzept vor, welches im Wesentlichen zwei Komponenten enthält:

- Die situationsaktuelle *Leistungsbestimmung*, welche mittels des erfassten Umgebungszustands die Leistungsindizes der im System mitgeführten Perzeptionsmodule über die Leistungsmodelle berechnet.
- Die *Selektion*, welche anhand des verwendeten Perzeptionsgraphs die Leistungen der gesamten Perzeptionsketten anhand der Einzelmodulleistungen ermittelt und die leistungsstärkste Perzeptionskette auswählt.

Bild 6.1 zeigt ein Flussdiagramm, welches den implementierten Prozessablauf der Algorithmenauswahl veranschaulicht. Im Initialfall, wenn noch keine Perzeptionskette ausgewählt ist bzw. der Perzeptionsauftrag gerade an das Perzeptionsmanagement übermittelt wurde, werden folgende Schritte ausgeführt:

- Ausrichtung der Sensoren auf den Startpunkt des Suchpfades
- Bestimmung des Umgebungszustands f_x
- Berechnung der Leistungsindizes aller Perzeptionsmodule im Perzeptionsgraphen
- Auswahl der geeignetsten Perzeptionskette und Aktivierung dieser in der Sensordatenverarbeitung

Der *Suchpfad* beschreibt den Weg, den der Sensor auf der Bodenoberfläche überstreicht. Nach dieser Initialisierungsphase wird der Umgebungszustand kontinuierlich weiter erfasst, um die Leistungsindizes der einzelnen Module ständig zu aktualisieren. Im Anschluss daran wird festgestellt, ob sich Leistungsindizes im Vergleich zur letzten Aktualisierung verändert haben. Wenn keine Veränderung eingetroffen ist, beginnt die Leitungsbestimmung wieder mit dem nächsten neu erfassten Umgebungszustand. Falls ein oder mehrere Modulleistungen einen anderen Wert annehmen, werden diejenigen Perzeptionsketten neu bewertet, die die entsprechenden Perzeptionsmodule beinhalten. Abschließend wird geprüft, ob die nun leistungsstärkste Perzeptionskette jene ist, die von der Sensordatenverarbeitung gerade ausgeführt wird. Wenn dies der Fall ist, dann startet der Auswahlprozess wieder mit der Aktualisierung der Modulleistungen. Anderenfalls wird ein Wechsel der Perzeptionskette durchgeführt bevor der Auswahlprozess von vorne beginnt.



Bild 6.1: Flussdiagramm des implementierten Prozessablaufs der Algorithmenauswahl.

6.2 Erweiterung des Sensor- und Perzeptionsmanagementsystems

Um die Algorithmenauswahl in das bestehende SPMS zu integrieren und allen gestellten Anforderungen aus Abschnitt 3.1 zu genügen, wurde dieses angepasst. Seine grundlegende Struktur wurde in Abschnitt 2.4.3 beschrieben, welche bei der hier beschrieben Erweiterung beibehalten wird.

Bild 6.2 zeigt ein detaillierteres Systemschaubild [100] des SPMS. Ausgehend von einem Perzeptionsauftrag (engl. *Perception Task*), welcher eine Aufgabenbeschreibung enthält (z. B. Suche nach Personen oder Fahrzeugen in einem definierten Gebiet), erzeugt der *Perception Solver* zwei Elemente. Zum einen wird ein Perzeptionsgraph erzeugt (siehe Abschnitt 3.3.2) welcher die Perzeptionsmodule enthält, die potenziell den Perzeptionsauftrag bearbeiten

können, und an den Processing Coordinator weitergeben. Zum anderen werden die Anforderungen an die Missionssensorik zusammengestellt und dem Sensor Planner zur weiteren Verarbeitung übertragen. Dieser zerlegt die Anforderungen in zwei Parametergruppen, eine für die Gaze Control und eine für den Sensor Configurator. Dabei hat die Gaze Control die Aufgabe, die Sensoren auszurichten, um z. B. entlang einer Straße nach Fahrzeugen zu suchen. Der Sensor Configurator selektiert und parametrisiert die Missionssensorik entsprechend den nötigen Anforderungen, welche aus der nachfolgenden Sensordatenverarbeitung abgeleitet wurden. Basierend auf dem Perzeptionsgraphen wählt der Processing Coordinator geeignete Verarbeitungsalgorithmen aus und verwaltet diese hinsichtlich ihres Ressourcenverbrauchs (Rechenzeit und Speicherplatz) sowie deren Algorithmenleistung. Abschließend werden die Ergebnisse der Sensordatenverarbeitung im Result Validator fusioniert und überprüft. Dabei ist konzeptionell vorgesehen, dass ein Quality Feedback zurück an den Processing Coordinator gegeben wird, um die Sensordatenverarbeitung zu optimieren.



Bild 6.2: Systemschaubild des Sensor und Perzeptionsmanagementsystems [100].

Gemäß den in Abschnitt 3.1 formulierten Anforderungen (vgl. Anforderung 5) wird zur Leistungsbestimmung der Umgebungszustand herangezogen. Deshalb wird eine Komponente benötigt, die diesen erfasst und dem SPMS hinzugefügt werden muss, um die vorgestellte Algorithmenauswahl integrieren zu können. Bild 6.3 zeigt das Systemschaubild des erweiterten SPMS, mit der Komponente der Umgebungserfassung, welche die Umgebungszustände ermittelt. Im ursprünglichen SPMS ist nicht vorgesehen, dass bei einer Änderung der Algorithmenleistung, und damit einer potentiellen Adaption der Sensordatenverarbeitung, auch das Sensormanagement informiert werden muss. Dies kann jedoch potentiell der Fall sein, da z. B. unterschiedliche Sensorarten oder Sensorparameter für die optimale Operation eines Perzeptionsmoduls nötig sind. Daher wird eine neue Komponente, die Algorithmenauswahl, dem Konzept hinzugefügt, welche den Perception Solver und die Auswahl und Adaption der Sensordatenverarbeitung enthält. Ausgangspunkt der Algorithmenauswahl ist der Umgebungszustand und die daraus ermittelten Leistungsindizes der einzelnen Perzeptionsmodule. Auf Basis der festgestellten Leistung wird eine Auswahl getroffen und diese anschließend der Sensordatenverarbeitung mitgeteilt (vgl. Abschnitt 3.2.2). Die Sensoreinsatzplanung ist mit der *Optimization* aus dem Sensormanagement, siehe Abschnitt 2.4.1, vergleichbar. Auf Basis des aktuellen Systemzustandes wird zum einen die Sensorauswahl getroffen und zum anderen die Positionierung bzw. Führung des Sensors optimiert. Für die in dieser Arbeit verwendeten Perzeptionsketten steht für jede Kette fest, welcher Sensor eingesetzt werden muss. Je nach Anwendung können die Kriterien für die Sensorausrichtung unterschiedlich sein. Für eine Objektverfolgung könnte ein wesentliches Kriterium sein, dass das Objekt immer im Sensorerfassungsbereich sein muss. Im Falle der hier betrachteten Anwendung der erdgerichteten Fahrzeugerkennung wird der Suchpfad vorgegeben.



Bild 6.3: Systemschaubild des erweiterten SPMS. Hervorgehoben sind die Blöcke, die zentraler Bestandteil des erweiterten Konzeptes sind.

6.3 Umsetzung des erweiterten SPMS

Damit die Algorithmenauswahl in einem operationellen Betrieb evaluiert werden kann, wurde das so erweiterte SPMS als Software-Prototyp umgesetzt. Um das System anhand der in Abschnitt 1.3 festgelegten Beispielapplikation (erdgerichtete Fahrzeugerkennung) untersuchen zu können, wurden des Weiteren folgende Detailfunktionen ergänzt:

- Die *Sensorausrichtung* führt den VIS- und LWIR-Sensor mittels eines Gimbals entlang eines definierten Bodenpfades (Suchpfad).
- Das Gimbal-System nutzt die Funktion eines *Geo-Locks*⁵, um den Bodenpfad entlang von Stützpunkten abzutasten.
- Die *Sensorauswahl* berechnet den Sensor-Footprint (siehe Anhang B) und stellt diesen der Umgebungserfassung bereit, um die Umgebungszustände Bodenbedeckung und Bodenpixelauflösung zu berechnen.
- Für den Operateur werden sowohl die Sensordaten als auch die Ergebnisse der *Sensordatenverarbeitung* zur Verfügung gestellt, um gerade für die Flugexperimente eine Überwachungsmöglichkeit zu schaffen.
- Die *Umgebungserfassung* bietet eine Schnittstelle für den Operateur, über die dieser die aktuelle lokale Wetterlage (Bewölkungsgrad, Niederschlagsmenge und Nebeldichte) an das SPMS übertragen kann.

Das implementierte SPMS ist mit Funktionsblöcken in Bild 6.4 dargestellt. Die *Sensordatenauswahl* empfängt die Sensordaten (SD) des VIS- und-LWIR Sensors und wählt auf Basis der Entscheidungen der *Algorithmenauswahl & Sensoreinsatzplanung* die benötigten Sensordaten für die *Sensordatenverarbeitung* aus. Außerdem wird mithilfe der hinterlegten Elevationsdaten⁶ im GIS, der Telemetrie (Tel) und der Gimbal-Lage (GL), der Sensor-Footprint

⁵ Mit dem Begriff *Geo-Lock* wird die Ausrichtung eines Gimbal-Systems auf eine fixe Position an der Bodenoberfläche verstanden.

⁶ Verwendet wurden die NASA SRTM 3 Elevationsdaten [165]

berechnet, wie in Anhang B beschrieben. Die *Umgebungserfassung* nutzt den Sensor-Footprint und die GIS-Daten (Rasterband- und Vektordaten), um die topographischen Umgebungszustände zu bestimmen, wie im Abschnitt 4.2.3 erläutert. Zusätzlich werden Wetterinformationen dem System von Extern (durch Operateureingabe oder direkt aus der Simulationsumgebung) übermittelt sowie die Gimbal-Lage erfasst, um schließlich den Umgebungszustand f_x zu berechnen. Dieser wird von der *Algorithmenauswahl* verwendet, um die Leistungsindizes der implementierten Perzeptionsmodule zu bestimmen und anschließend die geeignetste Perzeptionskette auszuwählen.

Ein Perzeptionsauftrag an das SPMS beinhaltet die Perzeptionsaufgabe, z.B. eine Fahrzeugerkennung und das Suchgebiet, in diesem Fall den Suchpfad für die *Sensorausrichtung*. In der in dieser Arbeit betrachteten Anwendung die Perzeptionsaufgabe immer eine Fahrzeugerkennung.

Zunächst prüft die Algorithmenauswahl mittels des aktuellen Umgebungszustands, welche Perzeptionskette *c_i* verwendet werden soll und überträgt anschließend die entsprechenden Ausführungsbefehle an die Sensordatenverarbeitung. Die Implementierungsdetails zur Auswahl der Perzeptionskette wurden im vorherigen Abschnitt 6.1 erläutert. Daneben wird der gewünschte Suchpfad an die Sensorausrichtung übertragen. Diese nutzt die Telemetrie für die Bestimmung der Position des Gimbals und zusätzlich die bereitgestellten Elevationsdaten des GIS, um die Bodenhöhe entlang des Suchpfades zu definieren. Anschließend wird die Geo-Lock Funktionalität des Gimbals verwendet, um den Suchpfad in diskreten Abständen abzutasten. Des Weiteren werden die komprimierten Sensordaten und die Ergebnisse der ausgeführten Perzeptionskette an eine Bodenkontrollstation übertragen. Der Missionsoperateur bzw. Experimentleiter hat damit die Möglichkeit, die Ergebnisse zu überprüfen und das System zu überwachen.



Bild 6.4: Schaubild des implementierten SPMS für die Flugexperimente. Die Abkürzungen Tel, SD und STRG stehen für die Flugtelemetrie, die Sensordaten und die Steuerungsbefehle. Alle Komponenten, die mit dem hellblauen Container "GIS" markiert sind, verwenden GIS-Daten.

KAPITEL 7 EXPERIMENTE

Anhand der Forschungsziele dieser Arbeit werden in diesem Kapitel die Untersuchungsgegenstände für die Experimentdurchführung definiert. Dabei werden die Untersuchungsgegenstände zur Entwicklung der einzelnen Experimente herangezogen. Daneben werden die Umgebungen, in denen die Experimente durchgeführt wurden, beschrieben. Die Experimente umfassen die Untersuchungen der erzeugten Leistungsmodelle und deren Einsatz in UAV-Missionen.

7.1 Untersuchungsgegenstände

Zur Evaluierung der Leistungsmodelle und des umgesetzten Systems zur Algorithmenauswahl in der Simulation und im Flugexperiment wurden Untersuchungsgegenstände zur Erreichung der festgelegten Forschungsziele aus Abschnitt 1.3 definiert. Die relevanten Forschungsziele für die Experimente waren die Folgenden:

- Ziel 3: Modellierung und Analyse der Einflüsse ausgewählter Umgebungszustände auf die sensordatenverarbeitenden Algorithmen hinsichtlich deren Leistungsverhaltens
- Ziel 4: Umsetzung der Algorithmenauswahl in ein prototypisches System
- Ziel 5: Evaluierung der Algorithmenauswahl anhand von repräsentativen Missionen

Die aus diesen Zielen abgeleiteten Untersuchungsgegenstände sind in nachfolgender Listung angegeben:

- 1 Analyse der erzeugten Leistungsmodelle (Ziel 3)
 - 1.1 Analyse der durch Expertenwissen erzeugten Leistungsmodelle
 - 1.2 Analyse der durch ML-Verfahren erzeugten Leistungsmodelle
- 2 Evaluierung der Algorithmenauswahl (Ziel 5)
 - 2.1 Evaluierung der Algorithmenauswahl hinsichtlich wechselnder Tageszeiten
 - 2.2 Evaluierung der Algorithmenauswahl unter atmosphärischen Einflüssen
 - 2.3 Evaluierung der Algorithmenauswahl in urbanen Szenerien
 - 2.4 Evaluierung der Algorithmenauswahl in ländlichen Szenerien
- 3 Überprüfung des angepassten SPMS-Konzeptes im Flugexperiment (Ziel 4)

Um das **Ziel 3** zu erreichen wurden bereits zur Modellierung der Einflüsse auf die sensordatenverarbeitenden Algorithmen Leistungsmodelle entwickelt (siehe Kapitel 5). Der erste Untersuchungsgegenstand (1) zielt auf die Analyse der Leistungsmodelle. Dabei wurde zwischen der Analyse der Experten-Leistungsmodelle (1.1) und der durch ML-Verfahren erzeugten Leistungsmodelle (1.2) unterschieden. Die Evaluierung der Algorithmenauswahl, anhand von repräsentativer Mission (**Ziel 5**), durch Einsatz der Leistungsmodelle wurde durch den zweiten Untersuchungsgegenstand (2) behandelt. Zur Strukturierung der Mission wurden vier weitere Untersuchungsgegenstände eingefügt, die die Algorithmenauswahl hinsichtlich wechselnder Tageszeiten (2.1), unterschiedlicher atmosphärischer Einflüsse (2.2), urbane Szenerien (2.3) und ländliche Szenerien (2.4) evaluiert. Der letzte Untersuchungsgegenständ

(3) wurde von **Ziel 4** abgeleitet und behandelte die Überprüfung des angepassten SPMS-Konzepts im Flugexperiment, und zielt damit auf die Realflugexperimente ab.

7.2 Experimentbeschreibung zur Analyse der Experten-Leistungsmodelle

In diesen Untersuchungen zum *Untersuchungsgegenstand 1.1* wurde analysiert, ob das erfasste Expertenwissen das tatsächliche Leistungsverhalten der modellierten Perzeptionsmodule nachbilden kann. Als Referenz dazu wurden die generierten *Metadaten* herangezogen. Da diese die Leistungsmessung für die Perzeptionsmodule TM, BLOB, CC und DPM beinhaltet, wurden die entsprechenden Experten-Leistungsmodelle diesen gegenübergestellt. Exemplarisch wurden einzelne Umgebungszustände variiert, während alle anderen festgelegt werden. Dadurch kann der Leistungsindex über einen Umgebungszustand aufgetragen werden. Es wurden dabei jene variable Umgebungszustände ausgewählt, die anhand der Eigenschaften der entsprechenden Perzeptionsmodule potentiell einen Einfluss besitzen.

Aus dem Vergleich zwischen den Resultaten der Experten-Leistungsmodells und der Messung (aus den *Metadaten*) wurde überprüft, ob die Regeln, welche durch die Experten verfasst wurden, korrekt sind. Wenn Leistungsdifferenzen entstanden sind, wurden diese durch manuelle Anpassung der Regelschlussfolgerungen diese Differenz korrigiert, um so eine Verbesserung der Experten-Leistungsmodelle durchzuführen. Um die Verbesserung zu quantifizieren wurden die NEFPROX- und KNN-Leistungsmodelle dem verbesserten Experten-Leistungsmodell gegenübergestellt. Damit soll gezeigt werden, ob ein Experten-Leistungsmodell vergleichbare Ergebnisse liefern kann.

Die übrigen Experten-Leistungsmodelle (ST, FA und VG) wurden zusammen mit den bereits analysierten Experten-Leistungsmodellen durch die Evaluierung der Algorithmenauswahl anhand von weiteren Realflug-Mission (siehe Abschnitt 7.4.2) untersucht. Damit wurden nicht nur die Experten-Leistungsmodelle weiter analysiert, sondern auch der *Untersuchungsgegenstand 3* (Überprüfung des angepassten SPMS-Konzeptes im Flugexperiment) behandelt.

7.3 Experimentbeschreibung zur Analyse der ML-Leistungsmodelle

Bei der Analyse der KNN- und NEFPROX-Leistungsmodelle (*Untersuchungsgegenstand 1.2*) wurden zunächst die die erzeugten Metadaten untersucht. Dabei wurde untersucht, ob die erfassten Umgebungszustände den Situationsraum genügend gut abdecken, indem die Gleichverteilung der Daten analysiert wurden. Außerdem wurde evaluiert ob tatsächlich eine nichtlineare Abhängigkeit zwischen Umgebungszuständen und dem Leistungsindex vorliegt (siehe Diskussion in Abschnitt 3.3.3.4). Die Analysemethoden, welche für diese beiden Untersuchungen (Gleichverteilung und Nichtlinearität) verwendet wurden sind in Abschnitt 8.2.1 zu finden.

Neben der Analyse der Metadaten zur Erzeugung der Leistungsmodelle wurden die NEFPROX-(siehe Abschnitt 5.2.2) und KNN-Leistungsmodelle (siehe Abschnitt 5.2.3) gegenübergestellt, um festzustellen, welche der beiden Verfahren das geeignetere ist, um für die folgenden Untersuchungen der Algorithmenauswahl ein Modell auszuwählen. Dazu wurde zum einen evaluiert, welches Modell die geringeren Abweichungen gegenüber einem Testdatensatz besitzt. Der Testdatensatz war dabei Teil der *Metadaten*, welcher nicht zum Training der NEFPROX- und KNN-Leistungsmodellen verwendet wurden. Außerdem werden die Verläufe der Leistungsindizes exemplarisch anhand der Bodenpixelauflösung und des Elevationswinkels untersucht, um Auffälligkeiten (z. B. Sprünge im Verlauf der Leistungsindex) zu identifizieren.

7.4 Missionsplanung und -beschreibung zur Evaluierung der Algorithmenauswahl

Zur Evaluierung der Algorithmenauswahl (*Untersuchungsgegenstand 2*) wurden 5 Aufklärungs- bzw. Überwachungsmission in einer Simulationsumgebung bzw. im Realflug durchgeführt. Dabei fanden 3 dieser Missionen in der Simulation und 2 im Flugexperiment statt. Die Flugexperimente dienten der zusätzlichen Überprüfung des implementierten SPMS (*Untersuchungsgegenstand 3*) und zur weiteren Analyse der Experten-Leistungsmodelle (*Untersuchungsgegenstand 1.1*). In den folgenden Abschnitten wird jeweils der verwendete Experimentalaufbau für Simulation und Flugexperimente vorgestellt. Anschließend werden die einzelnen Missionen im Detail präsentiert.

Unmittelbares Resultat der Missionen war ein Evaluierungsdatensatz $\tilde{\mathcal{E}}$, welcher Sensordaten, Objektannotationen und Umgebungszustände enthält. Um diesen zu erzeugen, wurden die Missionen auf unter Bezug auf die genannten Untersuchungsgenstände gestaltet. In Tabelle 7.1 sind alle Missionen mit den jeweiligen Freiheitsgraden und den zugeordneten Untersuchungsgegenständen aufgelistet.

	Mission	Variable Umgebungszustände	Untersuchungsgegenstand	Nr.
uo	Observierung	Wetter, Tageszeit, Elevationswinkel, Bodenbedeckung Wasser, Reliefwert	Analyse der durch ML-Verfahren erzeugten Leistungsmodelle. Evaluierung der Algorithmenauswahl hinsichtlich wechselnder Tageszeiten und atmosphärischer Bedingungen	1.2 2.1 2.2
Simulati	Unfallszenario	Alle geographischen Szenen- inhalte, Elevationswinkel	Analyse der durch ML-Verfahren erzeugten Leistungsmodelle. Evaluierung der Algorithmenauswahl in urbanen Szenerien mit flachen Elevationswinkeln	1.2 2.3
	Gebietsaufklärung I	Bodenbedeckung Ebene und Vegetation, Elevationswinkel	Evaluierung der Algorithmenauswahl in ländlichen Szenerien und Top-Down Ansicht	1.2 2.4
ersuch	Gebietsaufklärung II	Bodenbedeckung Ebene und Straße, GSD, Elevationswinkel	Überprüfung des angepassten SPMS Konzeptes im Flugexperiment. Evaluierung der Algorithmenauswahl in ländlichen Szenerien. Analyse der Leistungsmodelle	1.1 1.2 2.4 3
Flugve	Gebietsaufklärung III	Alle Bodenbedeckungen ohne Wasser, GSD, Elevationswinkel	Überprüfung des angepassten SPMS Konzeptes im Flugexperiment. Evaluierung der Algorithmenauswahl in ländlichen Szenerien. Analyse der Leistungsmodelle	1.1 1.2 2.4 3

Tabelle 7.1: Untersuchungsgegenstand und variable Umgebungszustände der Missionen.

In der Mission *Observierung* und *Unfallszenario* wurden die ausgewählten Leistungsmodelle (KNN oder NEFPROX) aus vorheriger Analyse anhand dieser untersucht, indem keine Perzeptionsketten, sondern nur die Perzeptionsmodule BLOB, TM, CC und DPM zur Algorithmenauswahl eingesetzt wurden. In den weiteren drei Missionen (*Gebietsaufklärung* I bis III) wurden komplette Perzeptionsketten untersucht, wobei ebenfalls die ausgewählten Leistungsmodelle aus der Analyse der ML-Leistungsmodelle sowie die Experten-Leistungsmodelle der Vorverarbeitungsstufen (ST, FA und VG) verwendet wurden. Zusätzlich dazu untersuchten die Missionen *Gebietsaufklärung* II und III die Algorithmenauswahl von Perzeptionsketten, die rein durch Experten-Leistungsmodelle bewertet wurden. Weiterhin, in Anlehnung an den Untersuchungsgegenstand 1.1, wurden anhand des Leistungsverhaltens der

Perzeptionsmodule eine weiter Analyse der Experten-Leistungsmodelle durchgeführt. Dabei wurden die Differenzen zwischen Modell und Messung erfasst und durch manuelle Anpassung der Regelbasis eine Verbesserung der Experten-Leistungsmodelle angestrebt.

7.4.1 Missionen in der virtuellen Umgebung

Im Folgenden wird zunächst der Experimentalaufbau zur Durchführung der Experimente in der Simulation vorgestellt. Anschließend werden die drei simulierten Missionen, mit ihren Generierungsparametern, Umgebungszuständen und situativen Elemente, beschrieben.

7.4.1.1 Durchführung der Experimente in der Simulation

Für die Untersuchungen in der Simulation wurde eine entsprechender Experimentalaufbau entworfen und aufgebaut. Dieser enthält die die folgenden essentiellen Komponenten:

- Simulation der physikalischen Umgebung
- Simulation des sensortragenden Luftfahrzeugs
- Simulation des Szenarios
- Skript zur Steuerung des Experiments
- Bedienschnittstelle für den Experimentleiter zur Überwachung des Experiments

Bild Übersicht der 7.1 eine Experimentalumgebung. Die eingesetzte zeigt Simulationsumgebung stellt Flugmodelle mit entsprechender Flugdynamik und der Möglichkeit, einen Gimbal mit mehreren Sensoren zu simulieren bereit. Dies wurde genutzt, um die Simulation des sensortragenden Luftfahrzeuges zu realisieren. Die Gestaltung der physikalischen Umgebung wurde über die in der Simulationsumgebung vorhandene 3D-Karten Sahrani, siehe Abschnitt 4.4, realisiert. Des Weiteren wurden die Szenarien für die einzelnen Missionen über einen in der Simulationsumgebung integrierten Editor modelliert. Dabei wurden Fahrzeuge platziert und deren Verhalten modelliert sowie einzelne Umgebungszustände (Witterung, Tageszeit und Monat) parametrisiert.



Bild 7.1: Aufbau der Experimentalumgebung für die Untersuchungen in der Simulation.

Für jede durchgeführte Mission wurde ein Missionsskript angefertigt, welches den Ablauf einer Mission steuert. Dieses wurde über eine Bedienoberfläche erzeugt, indem auf einer topographischen Karte der Mission die Flugroute, der Suchpfad und die Aufklärungsaufgabe eingegeben wurde. In der umgesetzten Mission besteht die Flugroute immer aus Wegpunkten und die Aufklärungsaufgabe ist immer eine erdgerichtete Fahrzeugerkennung. Schematisch ist diese in Bild 7.2 dargestellt.

Während des Ablaufes des Missionsskripts werden der Flugsteuerung des unbemannten Luftfahrzeugs die Flugroute bzw. die Wegpunkte übertragen und vom UAV abgeflogen. Dazu wird die Flugroute mit den folgenden beiden Wegpunktearten beschrieben:

- Die *Overfly*-Wegpunkte müssen überflogen werden, bevor der Weiterflug zum nächsten Wegpunkt erfolgen kann.
- An *Loiter*-Wegpunkten kreist das Flächenflugzeug für eine bestimmte Zeitspanne in einem bestimmten Radius um den Wegpunkt, bevor der nächste Wegpunkt aktiv wird.

Die Flughöhe wurde für die Experimente auf 400 Meter über Boden festgelegt.



Bild 7.2: Schematische Darstellung der Definition eines Auftrages. Hierbei stellen die Kreise die Geo-Lock Positionen für das Gimbal-System dar, die Dreiecke symbolisieren platzierte Fahrzeuge und die Rauten definieren die Wegpunkte für das UAV.

Der Perzeptionsauftrag, welcher Suchpfad und Aufklärungsaufgabe enthält, wurde anschließend an das SPMS übermittelt. Die *Sensorausrichtung* übernahm dabei die Führung des Gimbal-Systems entlang des Suchpfades. Die Simulation der VIS- und LWIR-Sensordaten ist an die Gimbal-Lage gekoppelt. Die Sensordaten werden mittels GigE Vision Protokoll⁷ an das SPMS übertragen. Außerdem wurde der Zoomfaktor der simulierten Sensoren, je nach Mission, entweder dynamisch durch das SPMS verändert, um eine bestimmte Bodenpixelauflösung konstant zu halten, oder ein Wert für den Zoomfaktor vorgegeben.

Während der Experimente wurden die Sensordaten, Umgebungszustände (siehe Abschnitt 4.2) und die Fahrzeugpositionen durch das SPMS aufgezeichnet. Die tatsächlichen Fahrzeugpositionen pro Frame werden verwendet, um damit die reale Leistung der Perzeptionsmodule messen und diese dann mit der Prognose der Leistungsmodelle vergleichen zu können. Um die Datenmengen der Aufzeichnungen während der Missionen zu reduzieren, wurde für jede Mission bzw. jeden Missionsabschnitt die Aufzeichnungsraten (Bilder pro Sekunde, *BpS*) für die VIS- und LWIR-Sensordaten individuell anhand der situativen Szenenvielfalt angepasst. So wurde zum Beispiel bei einer kontinuierlichen Abtastung entlang einer Straße die BpS erhöht, während bei einer mehrstündigen Beobachtung eines stehenden Fahrzeuges diese reduziert wurde.

⁷ Details zum GigE Vision Standard: https://www.visiononline.org/ [Letzter Zugriff: 20.08.2018]

7.4.1.2 Mission: Observierung

Die Rahmenhandlung dieser Mission stellte eine 24 Stunden Observierung eines verdächtigen Fahrzeuges dar. Damit konnte das Systemverhalten sowohl bzgl. wechselnder Tageszeiten als auch hinsichtlich sich ändernder der atmosphärischen Bedingungen evaluiert werden. Da die Objektdetektoren lediglich in gewissen Bereichen der Bodenpixelauflösung skalierungsinvariant sind, wurde für diese Mission eine konstante Bodenpixelauflösung gewählt, indem der Zoomfaktor entsprechend durch das SPMS kontrolliert wurde. In Tabelle 7.2 sind die Eckdaten der Missionsparameter gelistet.

Parameter	Wert	
Missionsstart	01.11.2017 um 13:00 Uhr	
Missionsende	02.11.2017 um 13:00 Uhr	
Bodenpixelauflösung	0,0694 <i>m/px</i>	
Taballa 7.2. Mission an anomaton dan Mission Obsernionun a		

Tabelle 7.2: Missionsparameter der Mission Observierung.

Diese Mission war zeitlich in fünf Abschnitte unterteilt und ist im Folgenden chronologisch aufgeführt:

- Beginnend am 01.11.2017 um 13:00 Uhr stand das Fahrzeug an einer Hafenanlage. Das Fluggerät kreiste in der Nähe um einen Wegpunkt mit 12 *m/s*. Die Sensoren waren mittels Geo-Lock auf das Fahrzeug gerichtet. Die Aufzeichnungsrate beträgt 0,1 *BpS*.
- Um 16:00 Uhr stieg ein Fahrer in das Fahrzeug und fuhr über eine Straße durch drei Siedlungen zu einer Einrichtung, welche um 16:30 Uhr erreicht wurde. Das UAV verfolgte die Fahrt mit einer Fluggeschwindigkeit von 30 *m/s*. Während des Fluges wurde das Fahrzeug im Mittelpunkt des Kamerabildes gehalten. Die Aufzeichnungsrate betrug hierbei 1 *BpS*.
- Das Fahrzeug stand anschließend bis 2:00 Uhr nachts am Folgetag an dieser Einrichtung. Das Fluggerät kreiste in der Nähe mit 12 m/s, während die Sensoren auf das Fahrzeug gerichtet waren. Die Aufzeichnungsrate betrug 0,05 *BpS*.
- Um 2:00 Uhr Fuhr das Fahrzeug in eine nahegelegene Stadt und bleibt um 2:30 Uhr vor einem Gebäude stehen. Wie zuvor wurde das Fahrzeug während der Fahrt vom UAV verfolgt. Die Fluggeschwindigkeit entlang der Wegpunkte betrug 30 *m/s*. Die Aufzeichnungsrate lag bei 1 *BpS*.
- Im letzten Missionsabschnitt kreiste das UAV in der Nähe des Fahrzeuges mit 12 *m/s* bis die Mission schließlich am 02.11.2017 um 13:00 Uhr endete. Die Sensoren waren während dieses Abschnittes weiterhin auf das Fahrzeug gerichtet und zeichneten mit 0,05 *BpS* auf.

Eine Lagekarte mit den fünf Missionsabschnitten, der eingezeichneter Flugroute und der Bewegung des Fahrzeuges ist in Bild 7.3 dargestellt. Die hinterlegten verwendeten GIS-Daten sind optisch hervorgehoben. Grüne Elemente entsprechen Vegetation, blaue Elemente zeigen Wasserflächen, rötliche Linien sind Straßen und rötliche Punkte bzw. Rechtecke stellen Gebäude dar. Die übrigen freien Flächen gehören der Bodenbedeckung *Ebene* an. Für die einzelnen Missionsabschnitte finden sich in Anhang E detailliertere Lagekarten.



Bild 7.3: Lagekarte der Mission Observierung. Die weiße Linie zeigt die Flugroute und die gelbe Linie die Bewegung des Fahrzeugs. Die Karte zeigt zusätzlich eine Visualisierung der GIS-Daten.

Die Observierungsmission diente zunächst dem Zweck die Leistungsmodelle, die durch ML-Verfahren erzeugt wurden, zu untersuchen und deckt damit den *Untersuchungsgegenstand 1.2* ab. Daneben wurden die Auswirkungen verschiedener Tageszeiten und Wetterbedingungen evaluiert, welche die Untersuchungsgegenstände 2.1 und 2.2 adressieren. In Bild 7.4 sind die Entwicklungen der Wettereffekte Bewölkungsgrad, Nebeldichte und Niederschlagsmenge über die Missionszeit dargestellt.



Bild 7.4: Wetterentwicklung der Mission Observierung.

Die sich aus dem Flugweg und der Sensorausrichtung ergebenden Reliefwerte und Elevationswinkel der Sensoren während der Missionszeit illustriert Bild 7.5.



Bild 7.5: Reliefwert- und Elevationswinkeländerung der Mission Observierung.

In Bild 7.6 sind die einhergehenden Bodenbedeckungen abgebildet. Mit Ausnahme des ersten Missionsabschnittes veränderte sich diese über große Zeitabschnitte kaum. Im ersten Missionsabschnitt dominiert die Bodenbedeckung *Wasser*, da sich das Fahrzeug im Bereich des Hafens aufhielt. Insgesamt wurden bei dieser Mission ca. 15000 Bild-Annotatierungs-Paaren aufgezeichnet.



Bild 7.6: Variation der Bodenbedeckung in der Mission Observierung.

In Anhang E sind die Veränderungen der Witterung, des Reliefwertes, des Elevationswinkels und der Bodenbedeckung für jeden Missionsabschnitt gesondert aufgeschlüsselt.

7.4.1.3 Mission: Unfallszenario

Die Aufgabe dieser Mission beinhaltete die Erstaufklärung eines Unfalls, welcher sich in einer Stadt an einer Kreuzung ereignet hatte. Durch den Unfall bildete sich ein Stau aus, mit der Folge, dass sich 26 Fahrzeuge entlang des Sensor-Suchpfades befanden. Bild 7.7 zeigt die Lagekarte der Mission, mit den entsprechenden definierten Markierungselementen aus vorheriger Mission. Tabelle 7.3 listet die Missionsparameter der Mission auf.

Parameter	Wert	
Missionsstart	01.5.2017 um 12:00 Uhr	
Missionsdauer	56 Sekunden	
Reliefwert	0	
Bodenpixelauflösung	0,0833 <i>m/px</i>	
Normierte Nebeldichte	0	
Normierte Niederschlagsmenge	0	
Normierter Bewölkungsgrad	0,5	
Parameter	Wert	
-------------------------------------	--------	
Fluggeschwindigkeit	25 m/s	
Bewegungsgeschwindigkeit Sensorpfad	12 m/s	

Tabelle 7.3: Missionsparameter in der Mission Unfallszenario.



Bild 7.7: Lagekarte der Mission Unfallszenario.

Diese Mission wurde für den *Untersuchungsgegenstand 2.3*, Evaluierung der Algorithmenauswahl in urbanen Szenerien, konzipiert. Zusätzlich diente sie als Leistungstest für die Objektdetektoren, da sich unterschiedliche Fahrzeuge auf der Straße befanden und diese zum Teil partiell verdeckt waren. Des Weiteren wurde in dieser Mission das Verhalten der Leistungsmodelle unter sehr flachen Elevationswinkeln sowie unter sich verändernden Bodenbedeckungen überprüft. In Bild 7.8 ist der Elevationswinkel über die Missionszeit aufgetragen.



Bild 7.8: Veränderung des Elevationswinkels in der Mission Unfallszenario.

Bild 7.9 zeigt die Bodenbedeckungen über die Missionszeit. Anzumerken ist, dass durch den flachen Elevationswinkel mehr Bodenfläche im Bild dargestellt wird, so dass am Anfang der Mission Teile der Meeresoberfläche zu sehen sind und sich deshalb in den Daten auch die Kategorie *Wasser* wiederfindet.



Bild 7.9: Variation der Bodenbedeckung in der Mission Unfallszenario.

Die Daten dieser Mission umfassen 850 Bild-Annotationspaare, bei einer Aufnahmerate von 15 *BpS*.

7.4.1.4 Mission: Gebietsaufklärung I

In dieser Mission wurde ein vordefiniertes Gebiet aufgeklärt, welches neben einem Dorf und zwischen zwei Straßen lag. In diesem Gebiet befanden sich kleine Wälder, eine Farm mit Feldern sowie vier Fahrzeuge, welche detektiert werden sollen. Die Lagekarte der Mission ist in Bild 7.10 abgebildet, wobei der Buchstabe "S" den Beginn eines Sensorpfades markiert und "E" das Ende.



Bild 7.10: Lagekarte der Mission Gebietsaufklärung I. Der Kartenausschnitt ist wie in vorheriger Mission zusätzlich mit einer Visualisierung der GIS-Daten dargestellt. Die Flugroute ist in Weiß und der Sensorpfad am Boden in Gelb eingezeichnet. Weiße Kreise entsprechen den Positionen der Fahrzeuge.

Tabelle 7.4 zeigt die Missionsparameter dieser Mission.

Parameter	Wert
Missionsstart	1.7.2017 um 18:00 Uhr
Missionsdauer	160 Sekunden
Reliefwert	0
Bodenpixelauflösung	0,0556 <i>m/px</i>
Normierte Nebeldichte	0
Normierte Niederschlagsmenge	0
Normierter Bewölkungsgrad	0,25
Fluggeschwindigkeit	13 m/s
Bewegungsgeschwindigkeit Sensorpfad	2,5 m/s

Tabelle 7.4: Missionsparameter in der Mission Gebietsaufklärung I.

Gegenstand der Evaluierung in dieser Mission ist die Untersuchung der Algorithmenauswahl in ländlichem Gebiet (*Untersuchungsgegenstand 2.4*). Daneben wurde ein steiler Elevationswinkel forciert, da so eine Top-Down-Ansicht auf das Gelände erreicht wurde. Damit wurde überprüfen, wie sich die eingesetzten Perzeptionsmodule bei steilen Elevationswinkeln verhalten, und ob dies von den Leistungsmodellen abgebildet wird. Deshalb befand sich die Flugroute direkt über dem Aufklärungsgebiet. In Bild 7.11 ist der Elevationswinkel über die Missionszeit aufgezeichnet.



Bild 7.11: Änderung des Elevationswinkels in der Mission Gebietsaufklärung I.

Weiterhin gilt in dieser Mission die Aufmerksamkeit der Veränderung der Bodenbedeckung *Vegetation* bzw. *Ebene*. In Bild 7.12 sind die Bodenbedeckungen über die Missionszeit aufgetragen. Zu sehen ist die Veränderung der Vegetation mit vier dominierenden Bereichen. Diese Mission besteht aus 3100 Bild-Annotations-Paaren, wobei die Aufnahmerate 20 *BpS* entsprach.



Bild 7.12: Veränderung der Bodenbedeckung in der Mission Gebietsaufklärung I.

7.4.2 Flugexperimente

Die Flugexperimente, welche sich aus zwei Missionen zusammensetzen, wurden für die Untersuchungsgegenstände 1.1, 1.2, 2.4 und 3 ausgelegt. Für 2.4 und 3 bedeutet dies, dass beide Missionen in ländlichen Gebieten durchgeführt wurden und damit auch ein Nachweis geführt werden kann, dass das SPMS mit der implementierten Algorithmenauswahl in realen Situationen einsetzbar ist. Außerdem wurde der *Untersuchungsgegenstand 1.1*, Evaluierung der durch Expertenwissen erzeugten Leistungsmodelle, betrachtet. Hierbei wurde überprüft ob die erzeugten Experten-Leistungsmodelle auch unter realen Bedingungen einsetzbar sind, und ob diese rein durch die Analyse der Missionsdaten verbessert werden können. Außerdem wurden neben den Experten-Leistungsmodellen auch die Leistungsmodelle der ML-Verfahren verwendet (*Untersuchungsgegenstand 1.2*).

7.4.2.1 Durchführung der Flugexperimente

Zur Durchführung der Flugexperimente wurde ein Modellflugzeug verwendet, welches dem Flugzeugtyp "Wilga" nachempfunden war (Bild 7.14a). Ausgerüstet war das Fluggerät mit einem elektrischen Propellerantrieb. Zur Flugsteuerung und -navigation wurde ein Autopilot vom Typ *Pixhawk⁸* eingesetzt. Bild 7.13 zeigt den schematischen Aufbau des Flugsetups, inklusive der Bedienmannschaft. Das UAV wurde durch einen Piloten mittels Funkfernsteuerung manuell mit Sichtkontakt geflogen. Ein Missionsoperateur erteilte während der Flugmissionen in der BKS mittels Touchscreens dem SPMS Perzeptionsaufträge, wie bereits in Abschnitt 7.4.1.1 dargestellt. Zur Überwachung der Experimente wurden auf den Displays der Bedienschnittstelle Sensordaten und Statusinformationen angezeigt. Der Experimentleiter koordinierte die Flugexperimente, indem er Start, Missionszeitraum und Landung abstimmte. Außerdem überwachte er zusätzlich zum Piloten die UAV-Telemetrie und die Flugroute. Die Datenverbindung zur Luftfahrtzeugsteuerung erfolgt über einen RC-Datenlink (engl. *Radio Control*). Die Datenverbindung zwischen BKS und SPMS wurde via LTE (engl. *Long Term Evolution*) hergestellt.



Bild 7.13: Schematische Darstellung des experimentalen Flugsetups mit Bedienmannschaft.

Die Komponenten der BKS wurden in einem Fahrzeug (siehe Bild 7.14b) integriert und besteht aus zwei Touchscreens mit entsprechendem Rechner, wie in Bild 7.14c dargestellt. Dabei zeigte der obere Touchscreen ein Interface, um die Sensordaten einzusehen und der untere

⁸ Mehr Informationen zum Pixhawk FMS finden sich auf der Herstellerseite https://pixhawk.org [Letzter Zugriff: 23.03.2018].

Touchscreen ein Interface, um die Lagekarte zu betrachten und Perzeptionsaufträge an das UAV zu übertragen. Außerdem konnte der Missionsoperateur dem SPMS die aktuellen Witterungsverhältnisse mitteilen, wie in Abschnitt 3.3.3 beschrieben. Bei dem verwendeten Gimbal handelte es sich um einen OTUS 170 des Herstellers DST Control (vgl. Bild 7.14d). Dieser besaß eine integrierte Geo-Lock Funktionalität und war mit einem VIS-Sensor vom Typ Sony FCB-EX480EP mit einer Auflösung von $744px \times 576px$ bei 25 *BpS* ausgerüstet. Außerdem war der Sensor mit einen 18 × optischen Zoom ausgestattet. Die Prozesse des implementierten SPMS wurden auf dem Luftfahrzeug auf einem Single Board Computer vom Typ LE-37C des Herstellers Commell ausgeführt.



Bild 7.14: Schaubilder des Setups für die Flugexperimente. In (a) ist das verwendete Fluggerät zu sehen und (b) zeigt die mobile BKS und den Monitor, links im Bild, für den Flugtestleiter. Die Arbeitsumgebung des Missionsoperateurs ist in (c) dargestellt und (d) zeigt das verwendete Gimbalsystem.

Während bei den Experimenten in der Simulation die Fahrzeugpositionen direkt für jedes Sensorframe aus dem Simulationssystem ausgelesen wurden, erfolgte bei den aufgezeichneten realen Sensordaten eine nachträgliche manuelle Annotation der Fahrzeugpositionen.

7.4.2.2 Mission: Gebietsaufklärung II

In dieser Mission wurde ein Gebiet mit drei stehenden Fahrzeugen aufgeklärt. Das Flugexperiment zur Durchführung der Mission fand auf dem Gelände der Universität der Bundeswehr München auf einem abgesperrten und nicht-öffentlichen Bereich statt. In Bild 7.15 ist die Lagekarte für diese Mission zu sehen, mit den Beschriftungselementen aus den vorherigen simulierten Missionen. Die Missionsparameter sind in Tabelle 7.5 abgebildet.

Parameter	Wert
Missionsstart	01.8.2017 um 10:11:53 Uhr
Missionsdauer	33 Sekunden
Reliefwert	0
Normierte Nebeldichte	0
Normierte Niederschlagsmenge	0
Normierter Bewölkungsgrad	0
Durchschnittliche Fluggeschwindigkeit	28 <i>m/s</i>
Durchschnittliche Flughöhe	100 m (AGL)

Tabelle 7.5: Missionsparameter der Mission Gebietsaufklärung II.



Bild 7.15: Lagekarte des Flugexperiments der Mission Gebietsaufklärung II.

Im Rahmen dieser Mission wurde ein konstanter Zoomfaktor für den Sensor verwendet, um eine Veränderung der Bodenpixelauflösung zu erreichen. Diese und der Elevationswinkel sind in Bild 7.16 über die Missionszeit aufgetragen.



Bild 7.16: Veränderung des Elevationswinkels und der Bodenpixelauflösung der Mission Gebietsaufklärung II.

Neben der Untersuchung der Veränderung der Bodenpixelauflösung stand bei dieser Mission der Einfluss von Straßenelementen im Vordergrund. Dabei ist in der ersten Missionshälfte eine Veränderung der Kategorie *Straße* und *Ebene* zu sehen, während die zweite Missionshälfte hauptsächlich die Kategorie Ebene zeigt. Dargestellt ist die Variation der Bodenbedeckung in Bild 7.17. Die Aufzeichnung der Missionsdaten ergaben 820 Bild-Annotierungs-Paare.



Bild 7.17: Variation der Bodenbedeckung der Mission Gebietsaufklärung II.

7.4.2.3 Mission: Gebietsaufklärung III

Wie in der vorherigen Mission wurde auch hier ein Gebiet aufgeklärt, in dem zwei Fahrzeuge platziert wurden. Der Unterschied bestand darin, dass mehr Veränderungen der Bodenbedeckung entlang des Sensorpfades am Boden auftraten als in vorheriger Mission, wie in der Lagekarte in Bild 7.18 dargestellt. Durchgeführt wurde die Mission auf dem Fliegerhorst Lager Lechfeld. Die Missionsparameter und konstanten Parameter während der Missionszeit sind in Tabelle 7.6 gelistet.



Bild 7.18: Lagekarte der Mission Gebietsaufklärung III.

Parameter	Wert
Missionsstart	21.09.2017 um 11:55:40 Uhr
Missionsdauer	68 Sekunden
Reliefwert	0
Normierte Nebeldichte	0
Normierte Niederschlagsmenge	0
Normierter Bewölkungsgrad	0,2
Durchschnittliche Fluggeschwindigkeit	25 m/s
Durchschnittliche Flughöhe	80 m (AGL)

Tabelle 7.6: Missionsparameter der Mission Gebietsaufklärung III.

Die variablen Parameter in dieser Mission waren der Elevationswinkel und die Bodenpixelauflösung, welche in Bild 7.19 für die Dauer der Missionszeit dargestellt sind. Außerdem veränderten sich alle Bodenbedeckungen, außer der Kategorie *Wasser*. Dabei dominierte in der ersten Missionshälfte die Kategorie *Ebene* und in der letzten Missionshälfte die Kategorien *Straße* und *Gebäude*, wie in Bild 7.20 visualisiert. Insgesamt wurden 1730 Bild-Annotierungs-Paare in dieser Mission aufgezeichnet.



Bild 7.19: Veränderung des Elevationswinkels und der GSD in der Mission Gebietsaufklärung III.



Bild 7.20: Variation der Bodenbedeckung in der Mission Gebietsaufklärung III.

KAPITEL 8 ANALYSE UND ERGEBNISSE

Die Analyse der Leistungsmodelle und die Ergebnisse der eingesetzten Algorithmenauswahl zum adaptiven luftgestützten Sensoreinsatz werden in diesem Kapitel vorgestellt. Zunächst erfolgt eine Analyse der Experten-Leistungsmodelle. Außerdem werden dies durch exemplarische manuelle Anpassung verbessert. Anschließend wird eine Analyse der NEFPROX- und KNN-Leistungsmodelle besprochen anhand der aufgenommenen Metadaten. Die Ergebnisse der Experimente zur Validierung der Algorithmenauswahl werden anhand der 5 Missionen abschließend präsentiert.

8.1 Analyse und Verbesserung der Experten-Leistungsmodelle

In den folgenden Untersuchungen wurden die Experten-Leistungsmodelle (M_{BLOB}^{EXP} , M_{TM}^{EXP} , M_{CC}^{EXP} und M_{DPM}^{EXP}) an einzelnen Fallbeispielen überprüft. Dazu wurden bestimmte Umgebungszustände fest angenommen und jeweils ein weiterer Umgebungszustand frei variiert. Die resultierenden Leistungsindizes wurden anschließend denen der NEFPROX- und KNN-Leistungsmodelle gegenübergestellt, um die Qualität der Modelle vergleichen zu können. Außerdem wird ein Vergleich mit den gemessenen Leistungsindizes, welche aus den Metadaten {F, P} extrahiert wurden, durchgeführt, um ggf. eine Verbesserung der Experten-Leistungsmodelle zu erreichen, indem die Regelschlussfolgerungen angepasst wurden.

8.1.1 Experten-Leistungsmodell des BLOB-Detektors

Der BLOB-Detektor wurde hauptsächlich zur Fahrzeugerkennung während der Nachtzeit entworfen. Deshalb wurde hier dessen Algorithmenleistung hinsichtlich der Variation der Tageszeit untersucht. Tabelle 8.1 zeigt die festgelegten Werte der übrigen Umgebungszustände.

Q'_r	$oldsymbol{ ho}_f'$	W _o	t	М	θ	g	t_w	t _{os}	t _b	ts	t_v	<i>r</i> ′
0	0	0	-	8	-45°	0,0693 m/px	0	0,4	0	0	0,6	0

Tabelle 8.1: Feste Umgebungszustände zur Einflussuntersuchung der Tageszeit auf den BLOB-Detektor.

Eine Gegenüberstellung der Verläufe der Leistungsindizes des Experten-, NEFPROX und KNN-Leistungsmodells ist in Bild 8.1a dargestellt. Zusätzlich dazu sind zum Vergleich die *Ground Truth* Daten (Messung) aus den *Metadaten* eingezeichnet. An der Messung ist zu erkennen, dass der BLOB-Detektor während den Nachtzeiten einen höheren Leistungsindex liefert als während den Tagzeiten.

Das Experten-Leistungsmodell kommt zu einem vergleichbaren Leistungsverlauf, wobei zwischen 10:00 und 22:00 Uhr ein geringerer Leistungsindex vorausgesagt wird, als dies die Messung zeigt. Um die Abweichung zu verringern, müssen bei den entsprechenden Regelschlussfolgerungen Bezeichner gewählt werden, die zu einem höheren Leistungsindex führen. Die manuellen angepassten Regeln in der Regelbasis des BLOB-Detektors sind in Tabelle 8.2 angegeben.

R_k	W _o	$oldsymbol{ heta}'$	t	t _{os}	t _v	t _w	t _s	t _b	<i>r</i> ′	p_{BLOB}
6	_	_	1	_	_	_	_	—	—	5 (4)
7	—	—	2	—	—	—	—	—	—	4 (3)
8	_	_	3	_	_	_	_	_	—	3 (2)
9	—	—	4	—	—	—	—	—	—	3 (2)
10	_	_	5	_	_	_	_	_	—	4 (3)
11	_	_	6	_	_	_	_	_	-	5 (4)

Tabelle 8.2: Angepasste Regeln der Regelbasis des BLOB-Detektors (Tageszeit). Die grün markierten Werte sind diejenigen, die von der originalen abweichen. Der Wert, welcher in Klammern angegeben ist, zeigt den Wert vor Anpassung der Regelbasis an.

Aus der Regelanpassung entstand schließlich der Leistungsverlauf des Experten-Leistungsmodells wie in Bild 8.1b dargestellt. Die Fehler, berechnet mit der L2-Norm, gegenüber der Messung der einzelnen Leistungsmodelle zeigt Tabelle 8.3. Hierbei war zu erkennen, dass das angepasste Experten-Leistungsmodell den zweit geringsten Fehler aufweist. Für den untersuchten Umgebungszustand besitzen dieses damit eine bessere Prädiktion als das ursprüngliche Experten-Leistungsmodell und das NEFPROX-Leistungsmodell.



Bild 8.1: Verläufe der Leistungsindizes der BLOB-Leistungsmodelle über die Tageszeit.

	FIS	NEFPROX	KNN	FIS (angepasst)		
Fehler	1,0862	0,8122	0,4506	0,5629		

Tabelle 8.3: Gegenüberstellung der Leistungsmodellfehler des BLOB-Detektors gegenüber der Messung.

Da der Bewölkungsgrad einen Einfluss auf den Kontrast und die Ausprägung thermischer Signaturen im LWIR-Sensorbild besitzt, wurden dieser ebenfalls untersucht. Hierbei wurden die übrigen Umgebungszustände zu den Werten in Tabelle 8.4 gewählt.

Q'_r	$oldsymbol{ ho}_f'$	W _o	t	М	θ	g	t _w	t _{os}	t _b	ts	t_v	<i>r</i> ′
0	0	_	15:00	1	-65°	0,0694 <i>m/px</i>	0	0,4	0	0	0,6	0

Tabelle 8.4: Gewählte Umgebungszustände zur Einflussuntersuchung des Bewölkungsgrads auf den BLOB-Detektor.

Bild 8.2a zeigt den Leistungsverlauf der einzelnen Modelle über den Bewölkungsgrad. Hierbei war eine deutliche Abweichung des Experten-Leistungsmodells gegenüber der Messung festzustellen. Das Experten-Leistungsmodell lieferte über den gesamten Wertebereich des Bewölkungsgrades niedriger Leistungsindizes als dies die Messung zeigte. Der prinzipielle Verlauf war allerdings korrekt, wodurch zur Anpassung der Regeln die Regelschlussfolgerungen so gewählt wurden, das höhere Leistungsindizes vom Modell ausgegeben werden. Tabelle 8.5 zeigt die angepasste Regelbasis.

R _k	W _o	$oldsymbol{ heta}'$	t	t _{os}	t_v	t _w	t _s	t _b	<i>r</i> ′	<i>p</i> _{BLOB}
1	1	_	5	_	_	_	_	_	-	3 (2)
2	2	—	5	—	_	Ι	—		—	4 (2)
3	3	_	5	_	_	_	_	-	—	4 (3)

Tabelle 8.5: Angepasste Regeln der Regelbasis des BLOB-Detektors (Bewölkungsgrad). Die grün markierten Werte sind diejenigen, die von der originalen abweichen. Der Wert, welcher in Klammern angegeben ist, zeigt den Wert vor Anpassung der Regelbasis an.

Aus der Anpassung der Regeln entstand der Leistungsverlauf des Experten-Leistungsmodells wie in Bild 8.2b dargestellt. Die Fehler (L2-Norm) der einzelnen Leistungsmodelle gegenüber der Messung sind in Tabelle 8.6 zusammengefasst. Durch die Anpassung konnte das Experten-Leistungsmodell den zweit geringsten Fehler erreichen. Lediglich das NEFPROX-Leistungsmodell besitzt einen kleineren Fehler.



Bild 8.2: Verläufe der Leistungsindizes der BLOB-Leistungsmodelle über den Bewölkungsgrad.

	FIS	NEFPROX	KNN	FIS (angepasst)	
Fehler	1,2672	0,3315	0,7440	0,6474	

Tabelle 8.6: Gegenüberstellung der Leistungsmodellfehler des BLOB-Detektors gegenüber der Messung.

8.1.2 Experten-Leistungsmodell des Template Matchings

Das *Template Matching* nutz die Gradienten im Bild, um Fahrzeuge zu klassifizieren. Dabei kann die Nebeldichte die Stärke der Gradienten reduzieren, wodurch potenziell eine geringere Detektionsgenauigkeit entsteht. Deshalb wurde dieser Umgebungszustand untersucht, wobei für die anderen Umgebungszustände die festen Werten aus Tabelle 8.7 gewählt wurden.

Q'_r	$ ho_f'$	W _o	t	М	θ	g	t _w	t _{os}	t _b	t _s	t_v	r'
0	_	0	09:00	10	-90°	0,0694 <i>m/px</i>	0	0,4	0	0	0,6	0

Tabelle 8.7: Feste Umgebungszustände zur Einflussuntersuchung der Nebeldichte auf das TM.

Bild 8.3a zeigt die Verläufe der Leistungsindizes des Experten-, KNN- und NEFPROX-Leistungsmodells. Zum Vergleich sind die Ground Truth Daten (Messung) aus den *Metadaten* angegeben.

Das Experten-Leistungsmodell zeigte bei höheren Nebeldichten eine deutliche Abweichung zur Messung. In diesem Bereich war der Leistungsindex geringer als die Messung dies aufzeigte. Zur Regelanpassung wurde für die diesen Bereich Regelschlussfolgerungen gewählt, die zu einem höheren Leistungsindex führen. Die Regelanpassungen sind in Tabelle 8.8 dargestellt.

R_k	W _o	$ ho_f'$	$oldsymbol{ heta}'$	g ′	t	t _{os}	t_v	t _w	t _s	t _b	<i>r</i> ′	р _{ТМ}
4	_	4	_	—	2	3	_	_	_	_	—	3 (2)
5	_	5	_	_	2	3	_	-	_	_	_	3 (1)

Tabelle 8.8: Angepasste Regeln der Regelbasis des Template Matchings (Nebeldichte). Die grün markierten Werte sind diejenigen, die von den originalen abweichen. Der Wert, welcher in Klammern angegeben ist, zeigt den Wert vor Anpassung der Regelbasis an.

Aus der Anpassung der Regelbasis des Template Matchings entstand der Leistungsverlauf des Experten-Leistungsmodells wie in Bild 8.3 dargestellt. Die Fehler der Leistungsmodelle gegenüber der Messung sind in Tabelle 8.9 aufgeführt. Durch die Anpassung konnte das Experten-Leistungsmodell den geringsten Fehler im Gegensatz zu allen anderen Leistungsmodellen erreichen.



Bild 8.3: Verläufe der Leistungsindizes der TM-Leistungsmodelle über die Nebeldichte.

		FIS	NEFPROX	KNN	FIS+	
	Fehler	0,7511	0,9265	1,2865	0,1768	
Та	belle 8.9: Gegenüberst	ellung der Leistung	smodellfehler des	TM-Detektors geg	enüber der Messun	ıg.

Weiterhin wurde die Veränderung der Bodenpixelauflösung untersucht. Dabei wurden die übrigen Umgebungszustände zu den Werten, welch in Tabelle 8.10 gezeigt sind, festgelegt.

Q'_r	$ ho_f'$	W _o	t	М	θ	g	t _w	t _{os}	t _b	ts	t_v	<i>r</i> ′
C	0	0	08:00	10	-90°	_	0	0	0,6	0,1	0,3	0
Та	Tabelle 8.10: Gewählte Umgebungszustände zur Einflussuntersuchung der GSD auf das TM.											

Bild 8.4a zeigt die Leistungsindizes der einzelnen Modelle gegenüber der Messung in Abhängigkeit der Bodenpixelauflösung. Dabei zeigte sich, dass das TM nur in einem gewissen Bereich, zwischen 0,3 und 0,6, einsetzbar ist. Das Experten-Leistungsmodell zeigt dabei eine Abweichung in den Randbereichen, die zu höheren Leistungsindizes führte als dies die Messung anzeigte. Diese Randbereiche wurden durch entsprechende Regelanpassung (siehe Tabelle 8.11) zu niedrigeren resultierenden Leistungsindizes korrigiert.

R_k	W _o	$ ho_f'$	$oldsymbol{ heta}'$	g '	t	tos	t_v	t _w	ts	t _b	<i>r</i> ′	р _{ТМ}
9	_	—	1	4	2	_	_	_	_	3	-	3 (2)
10	_	-	1	5	2	_	_	-	-	3	—	3 (1)

Tabelle 8.11: Angepasste Regeln der Regelbasis des Template Matchings (Bodenpixelauflösung). Die grün markierten Werte sind diejenigen, die von den originalen abweichen. Der Wert, welcher in Klammern angegeben ist, zeigt den Wert vor Anpassung der Regelbasis an.

Bild 8.4b stellt den Leistungsindex des angepassten Experten-Leistungsmodells gegenüber der Messung dar. Anhand von Tabelle 8.12 konnte festgestellt werden, dass auch hier die Anpassung den Fehler gegenüber der Messung reduzieren konnte. Damit besaß das ExpertenLeistungsmodell den zweit geringsten Fehler. Lediglich das KNN-Leistungsmodell zeigte einen geringeren Fehler.



Bild 8.4: Verläufe der Leistungsindizes der TM-Leistungsmodelle über die Bodenpixelauflösung.

	FIS	NEFPROX	KNN	FIS+
Fehler	0,6055	0,8391	0,2769	0,3972

Tabelle 8.12: Gegenüberstellung der Leistungsmodellfehler des TM-Detektors gegenüber der Messung.

8.1.3 Experten-Leistungsmodell der Klassifizierungskaskade

Die Klassifizierungskaskade wurde mit unterschiedlichen Datensätzen trainiert, in welchen die Fahrzeuge in unterschiedlichen Elevationswinkel vorlagen. Dazu soll zunächst der Einfluss des Elevationswinkels betrachtet werden. In Tabelle 8.13 sind die übrigen Umgebungszustände und deren gewählten Werte angegeben.

Q'_r	$ ho_f'$	W _o	t	М	θ	g	t_w	tos	t _b	ts	t_v	<i>r</i> ′
0	0	0,5	10:00	8		0,0694	0	0,6	0	0,2	0,2	0

Tabelle 8.13: Feste Umgebungszustände zur Einflussuntersuchung des Elevationswinkels auf die Klassifizierungskaskade.

Bild 8.5 zeigt den Verlauf der Leistungsindizes der Leistungsmodelle über den Elevationswinkel aufgetragen. Das Experten-Leistungsmodell zeigte im Allgemeinen die größte Abweichung. Während die Messung zeigte, dass der Leistungsindex bei steigendem Elevationswinkel höher wurde, zeigte das Experten-Leistungsmodell ein gleichbleibendes Niveau des Leistungsindex über den Elevationswinkel.



Bild 8.5: Verläufe der Leistungsindizes der CC-Leistungsmodelle über den Elevationswinkel.

Durch eine reine Regeländerung konnte die Abweichung nicht reduziert werden. Zur Verbesserung müssten die Zugehörigkeitsfunktionen verändert werden. Dies wurde aber nicht ausgeführt, da dadurch alle Regeln der Regelbasis eine Änderung erfahren würden. Tabelle

8.14 fast die Fehler (L2-Norm) der einzelnen Leistungsmodelle gegenüber der Messung zusammen.

	FIS	NEFPROX	KNN
Fehler	0,1865	0,8865	0,6526

Tabelle 8.14: Gegenüberstellung der Leistungsmodellfehler der Klassifizierungskaskade gegenüber der Messung.

Weiterhin wurde der Einfluss der Bodenpixelauflösung auf das Experten-Leistungsmodell der Klassifizierungskaskade untersucht. Für diese Untersuchung wurden die übrigen Umgebungszustände zu den in Tabelle 8.15 abgebildeten Werten gewählt.

Q'_r	$ ho_f'$	W _o	t	М	θ	g	t_w	t _{os}	t _b	t _s	t_v	<i>r</i> ′
0	0	0	08:00	4	-90°	_	0	0,4	0	0	0,6	0

Tabelle 8.15: Gewählte Umgebungszustände zur Einflussuntersuchung der GSD auf die Klassifizierungskaskade.

Der Verlauf der Leistungsindizes des Experten-, KNN- und NEFPROX-Leistungsmodells gegenüber der Messung zeigt Bild 8.6a. Hierbei war eine Abweichung des Experten-Leistungsmodells bei steigender Bodenpixelauflösung deutlich erkennbar. Das Experten-Leistungsmodell lieferte zu geringe Leistungsindizes, gerade bei sehr hohen Bodenpixelauflösungen. Um diese Abweichung zu reduzieren, wurde wiederum die Regelbasis der Klassifikationskaskade, wie in Tabelle 8.16 angegeben, angepasst.

R _k	W _o	θ'	g ′	t	t _{os}	t_v	t _w	t _s	t _b	<i>r'</i>	p _{CC}
6	1	1	2	_	_	_	_	_	_	-	5 (4)
7	1	1	3	_	_	_	_	_	_	-	5 (3)
8	1	1	4	—	—	—	—	—	—	-	4 (1)
9	1	1	5	_	_	_	_	_	_	-	4 (1)

Tabelle 8.16: Angepasste Regeln der Regelbasis der Klassifizierungskaskade (Bodenpixelauflösung). Die grün markierten Werte sind diejenigen, die von den originalen abweichen. Der Wert, welcher in Klammern angegeben ist, zeigt den Wert vor Anpassung der Regelbasis an.

Aus der Anpassung der Regeln resultiert ein verbesserter Verlauf des Leistungsindex über die Bodenpixelauflösung wie in Bild 8.6b dargestellt. Tabelle 8.17 zeigt die Fehler der einzelnen Leistungsmodelle gegenüber der Messung. Den geringsten Fehler besitzt nach wie vor das KNN-Leistungsmodell.



Bild 8.6: Verläufe der Leistungsindizes der CC-Leistungsmodelle über die Bodenpixelauflösung.

	FIS	NEFPROX	KNN	FIS (angepasst)
Fehler	1,2893	1,1263	0,1940	0,2093

Tabelle 8.17: Gegenüberstellung der Leistungsmodellfehler der Klassifizierungskaskade gegenüber der Messung.

8.1.4 Experten-Leistungsmodell des DPM-Detektors

Der DPM-Detektor verwendet zur Merkmalsextraktion das HOG-Merkmal, welches auf Basis der Gradienten im Bild erzeugt wird. Daher wird der Einfluss der Nebeldichte untersucht, da diese den Kontrast reduziert und damit auch die Gradienten abschwächt. Zu dieser experimentellen Untersuchung wurden die restlichen Umgebungszustände zu den Werten in Tabelle 8.18 festgelegt.

Q'_r	$ ho_f'$	Wo	t	М	θ	g	tw	tos	t _b	ts	t_v	r'
0	_	0	07:00	1	-80°	0,0694	0	0,2	0,4	0,2	0,2	0
11 0 4	0 1 .	TT					1	1 17	1 1 1 1		DDL	L D . 1

Tabelle 8.18: Feste Umgebungszustände zur Einflussuntersuchung der Nebeldichte auf den DPM-Detektor.

Die Leistungsverläufe der einzelnen Leistungsmodelle (Experte, KNN und NEFPROX) sind gegenüber den Ground Truth Daten (Messung), gewonnen aus den *Metadaten*, in Bild 8.7 dargestellt. Hierbei zeigte das Experten-Leistungsmodell einen ähnlichen Leistungsverlauf wie das der Messung. Bei sehr hohen Nebeldichten kommt es allerdings zu einer Abweichung. Hier lieferte das Experten-Leistungsmodell einen zu geringen Leistungsindex. Um diesen bei sehr hohen Nebeldichten zu erhöhen wurde die entsprechend Regelschlussfolgerung angepasst, wie in Tabelle 8.19 dargestellten.

R _k	W _o	$ ho_f'$	g ′	t	t _{os}	t_v	t _w	ts	t _b	<i>r</i> ′	p_{DPM}
5	1	5	-	2	_	_	_	_	—	-	3 (1)

Tabelle 8.19: Angepasste Regeln der Regelbasis des DPM-Detektors (Nebeldichte). Die grün markierten Werte sind diejenigen, die von den originalen abweichen. Der Wert, welcher in Klammern angegeben ist, zeigt den Wert vor Anpassung der Regelbasis an.

Aus der Anpassung der Regelbasis des DPM-Detektors resultierte der Leistungsindex des Experten-Leistungsmodells wie in Bild 8.7b abgebildet. Die Fehler der einzelnen Leistungsmodelle zeigt Tabelle 8.20. Die Fehler zeigten, dass durch die Anpassung des Experten-Leistungsmodells der Ursprungsfehler halbiert werden konnte und damit das Experten-Leistungsmodell deutlich besser abschnitt, als die anderen Leistungsmodelle.



Bild 8.7: Verläufe der Leistungsindizes der DPM-Leistungsmodelle über die Nebeldichte.

	FIS	NEFPROX	KNN	FIS+
Fehler	0,9448	1,2597	1,3286	0,4822

Tabelle 8.20: Gegenüberstellung der Leistungsmodellfehler des DPM-Detektors gegenüber der Messung.

Weiterhin wurde das Experten-Leistungsmodell des DPM-Detektors hinsichtlich des Einflusses der Bodenbedeckung *Gebäude* untersucht. Die übrigen Umgebungszustände wurden zu den Werten wie in Tabelle 8.21 gewählt. Hierbei ist anzumerken, dass bei steigendem Anteil der *Gebäude*, entsprechend der Anteil der *Ebene* reduziert wurde.

Q'_r	$oldsymbol{ ho}_f'$	W _o	t	М	θ	g	t_w	tos	t _b	ts	t_v	<i>r</i> ′
0	_	0	14:00	5	-65°	0,0694	0	1.0	-	0	0	0

Tabelle 8.21: Feste Umgebungszustände zur Einflussuntersuchung der Bodenbedeckung *Gebäude* auf den DPM-Detektor.

Bild 8.8a zeigt die Verläufe der Leistungsindizes der einzelnen Leistungsmodelle über die Bodenbedeckung *Gebäude*. An der Messung konnte festgestellt werden, dass ab 65% Gebäudeanteil die Leistung des DPM-Detektors auf 0 sinkt. Alle Leistungsmodelle zeigten eine Abweichung bei steigender Bodenbedeckung (Gebäude), und lieferten damit zu hohe Leistungsindizes. Zu Verbesserung des Experten-Leistungsmodells wurde die Regelbasis des DPM-Detektors angepasst, wie in Tabelle 8.22 angegeben.

R_k	Wo	$ ho_f'$	g ′	t	tos	t_v	t _w	ts	t _b	<i>r</i> ′	р _{DPM}
8	_	_	—	4	3	_	—	_	3	—	3 (4)
9	_	_	_	4	2	-	_	_	4	—	2 (3)
10	_	_	_	4	1	_	_	_	5	_	1 (3)

Tabelle 8.22: Angepasste Regeln der Regelbasis des DPM-Detektors (Bodenbedeckung Gebäude). Die grün markierten Werte sind diejenigen, die von den originalen abweichen. Der Wert, welcher in Klammern angegeben ist, zeigt den Wert vor Anpassung der Regelbasis an.

Durch die Anpassung der Regeln resultierte der Leistungsverlauf des Experten-Leistungsmodells über die Bodenbedeckung (Gebäude) wie in Bild 8.8b dargestellt. Anhand der Fehler (siehe Tabelle 8.23) zeigte sich, dass durch die Regelanpassung das Experten-Leistungsmodell den geringsten Fehler gegenüber allen anderen Leistungsmodellen besaß.





	FIS	NEFPROX	KNN	FIS+
Fehler	1,1512	1,1613	0,8944	0,8227

Tabelle 8.23: Gegenüberstellung der Leistungsmodellfehler des DPM-Detektors gegenüber der Messung.

8.2 Analyse der Leistungsmodellierung durch ML-Verfahren

Die Analyse der Leistungsmodelle umfasste zunächst die Untersuchung der *Metadaten* hinsichtlich ihrer Gleichverteilung der einzelnen Umgebungszustände. Anschließend erfolgte die Überprüfung auf eine nichtlineare Abhängigkeit zwischen den Umgebungszuständen und dem Leistungsindex. Weiterhin wurden ein Vergleich zwischen den NEFPROX-Leistungsmodellen und den KNN-Leistungsmodellen durchgeführt, um das geeignetere Modell für die Algorithmenauswahl einzusetzen. Dabei umfasste der Vergleich zum einen eine Evaluierung an einem Testdatensatz und zum anderen eine Untersuchung, um bei kontinuierlichen Umgebungszustandsänderungen, Aufschluss über die Leistungsverläufe zu erhalten.

8.2.1 Verwendete Methoden zur Analyse der Leistungsmodelle

Damit die Leistungsmodelle nicht nur anhand der durchgeführten Missionen bewertet werden, sondern auch der Prozess der Modellerzeugung analysiert werden kann, wurden in die Datenaufbereitung aus Abschnitt 5.2.1 drei Analysemethoden integriert. Diese dienen dem Zweck, zum einen die Datenaufbereitung der *Metadaten* zu evaluieren und zum anderen den Zusammenhang zwischen Umgebungszustände und dem Leistungsindex aufzuzeigen. Bild 8.9 verdeutlicht an welcher Stelle die Analysemethoden in die Datenaufbereitung aus Bild 5.7 eingefügt wurden.



Bild 8.9: Integration der Analysemethoden in die Datenaufbereitung und Modellbildung.

Die Analysemethoden sind eine Hauptkomponentenanalyse (engl. Principal Component Analysis, PCA) [149], die Ermittlung einer diskrete Gleichverteilung (DG) und eine Bestimmung des linearen Korrelationskoeffizienten (engl. Linear Correlation Coefficient, LCC). Die Hauptkomponentenanalyse wurde dazu verwendet, um die Metadaten {F, P} hinsichtlich ihrer Varianz zu untersuchen. Da die PCA durch Linearkombination die Dimension des Umgebungszustands reduzieren kann, ohne dabei nennenswerte Information zu verlieren, ist es möglich, die Informationsverteilung in den Metadaten zu ermitteln. Eine schematische Veranschaulichung der PCA bietet Bild 8.10. In diesem Beispiel wird ein zweidimensionaler Raum mit Beispieldaten durch neue Achsen (Hauptkomponenten) beschrieben. Dabei stehen alle Hauptachsen immer senkrecht zueinander und die erste Hauptachse wird so bestimmt, dass die Varianz entlang dieser maximiert wird. Zu sehen ist, dass die erste Hauptkomponente die Datenpunkte bereits besser beschreibt als einer der Achsen des originalen Koordinatensystems. Ideal sind die Datenpunkte in den Metadaten verteilt, wenn alle Hauptkomponenten die gleiche Varianz besitzen und damit die Daten über den gesamten Merkmalsraum gleichmäßig verteilt sind. Die Ergebnisse der Analyse mittels der PCA stellt der Abschnitt 8.2.2 dar.



Bild 8.10: Beispiel einer PCA im zweidimensionalen Fall. HK1 und HK2 stehen für die erste und zweite Hauptkomponente.

Zur Analyse der Metadaten hinsichtlich deren Gleichverteilung wurde die DG eingesetzt. Dabei wurde der Mittelwert jedes Umgebungszustandes \overline{F} und dessen Standardabweichung S_F berechnet mit

$$\bar{F} = \frac{1}{N} \sum_{i=1}^{N} f_i \qquad \sigma_F = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} |f_i - \bar{F}|^2}.$$
(8.1)

Ziel für jeden Umgebungszustand war eine optimale Gleichverteilung der Datenpunkte. Die optimale Gleichverteilung für einen Umgebungszustand berechnete sich mit den folgenden Formeln:

$$\overline{F_{opt}} = \frac{a+b}{2} \qquad \sigma_{F,opt} = \sqrt{\frac{(b-a)^2}{12}} = \frac{b-a}{2\sqrt{3}} \approx 0,289(b-a)$$
(8.2)

Der minimale Wert eines Umgebungszustandes wird mit *a* und der maximale Wert wird mit *b* bezeichnet. Beispielweise wäre für den Bewölkungsgrad a = 0 und b = 1. Daraus resultiert der optimale Mittelwert zu 0,5 und die optimale Standardabweichung zu \approx 0,289 für eine optimale Gleichverteilung.

Um den Einfluss und die Signifikanz der Umgebungszustände auf und für den Leistungsindex aufzuzeigen, wurde der Korrelationskoeffizient r nach Pearson [150] verwendet. Dieser misst den linearen Zusammenhang zwischen zwei Variablen in einem normierten einheitenlosen Maß. Dadurch konnte überprüft werden, ob tatsächlich ein nichtlinearer Zusammenhang, wie in Abschnitt 3.3.3.4 vermutet, besteht. Ist der Korrelationskoeffizient Null, dann besteht kein linearer Zusammenhang zwischen den beiden Variablen, wie im Beispiel in Bild 8.11a. Ist r =1, dann besteht ein perfekter Linearitätszusammenhang. Je größer die Streuung der Daten ist, umso kleiner wird r (siehe Bild 8.11b und Bild 8.11c). Das Vorzeichen des Korrelationskoeffizienten bestimmt dabei die Richtung der Steigung. Wenn beide Variablen gemeinsam steigen oder sinken, ist das Vorzeichen positiv, ansonsten negativ (siehe Bild 8.11d). Neben dem Korrelationskoeffizienten wurde auch der P-Wert als Maß für die Signifikanz bestimmt. Ist diese nahe Null, ist die Korrelation der beiden untersuchten Variablen signifikant. Typischerweise wird geprüft, ob der *P*-*Wert* unter einer Schwelle $\alpha = 0.05$ liegt. Dies bedeutet ein Risiko von 5%, dass eine Korrelation als signifikant eingestuft wird, obwohl gar keine Korrelation vorliegt. Für die Evaluierung der Leistungsmodelle anhand der Trainingsdaten ließ sich mit dem *P-Wert* bestimmen, ob eine Korrelation vorlag. Wenn dies der Fall war, konnte eine Aussage dahingehend getroffen werden, in welche Richtung sich der Leistungsindex bei Änderung einer der Umgebungszustände bewegt hat.



Bild 8.11: Visualisierung des Pearson Korrelationskoeffizienten.

8.2.2 Analyse der Metadaten

Zur Analyse des Modelldatensatzes wurde zunächst der Prozess der Datenaufbereitung aus Abschnitt 5.2.1 evaluiert. Ziel war es, die Datenpunkte möglichst gleichmäßig im Parameterraum zu verteilen. Dies bedeutet, die Wertebereiche zu harmonisieren und redundante Datenpunkte zu fusionieren. Wie in Abschnitt 8.2.1 erläutert, wurde die Hauptkomponentenanalyse verwendet, um die Verteilung der Information im Modelldatensatz zu bestimmen.

	$\widetilde{\mathcal{D}_r}$	$\widetilde{\mathcal{D}_p}$	$\widetilde{\mathcal{D}}$
HK1	99,957	24,186	24,569
HK2	0,037	18,254	15,478
HK3	0,004	13,232	15,309
HK4	0,001	12,340	12,887
HK5	0,000	10,626	11,805
HK6	0,000	8,773	6,045
HK7	0,000	4,340	4,235
HK8	0,000	2,336	2,827
HK9	0,000	1,956	2,389
HK10	0,000	1,758	1,807
HK11	0,000	1,219	1,185
HK12	0,000	0,692	0,820
HK13	0,000	0,182	0,444
HK14	0,000	0,065	0,145
HK15	0.000	0.040	0.055

Tabelle 8.24: Ergebnisse der PCA anhand der Metadaten. Die mit Grün markierten Hauptkomponenten repräsentieren hierbei 99 % der enthaltenen Informationen.

Tabelle 8.24 zeigt die Ergebnisse der Hauptkomponentenanalyse. Die Zahlenwerte in den einzelnen Spalten geben den prozentualen Informationsgehalt pro Hauptkomponente an, wobei die Summe immer 100 % ergibt. Hierbei kann festgestellt werden, dass die Verteilung der Information auf mehrere Hauptkomponenten (Dimensionen) durch die Vorverarbeitung der Rohdaten $\widetilde{\mathcal{D}_r}$ verbessert wurde. Anzumerken ist, dass die Reduzierung auf eine Hauptkomponente in $\widetilde{\mathcal{D}_r}$ durch den im Verhältnis großen Wertebereich des Elevationswinkels [-90°, 90°] zurückzuführen ist. Mit der Clusterbildung konnten die Metadaten so verteilt werden, dass mit 12 Dimensionen mindestens 99 % der Informationen repräsentiert werden konnten. Dies bedeutet, dass die 15 Umgebungszustände auf 12 Dimensionen reduziert werden könnten, ohne nennenswerte Informationen zu verlieren. Damit sind mindestens 3 Variablen nicht linear unabhängig. Zurückzuführen ist die scheinbar ungleichmäßige Verteilung der Daten auf die gekoppelten Umgebungszustände Tageszeit, Monat und die Bodenbedeckung. Diese können im Parameterraum nur begrenzte Werte annehmen, da die Tageszeit und der Monat periodische Variablen sind, die durch jeweils 2 abhängige Variablen repräsentiert werden, und die Bodenbedeckung durch 5 Variablen abgebildet wird, welche in der Summe immer 1 ergeben müssen. Damit ergeben sich 9 Variablen, die nicht linear unabhängig sind. Deshalb ist die gezeigte Informationsverteilung der Datenpunkte für die Erzeugung der Leistungsmodelle zunächst plausibel.

Um die Gleichverteilung der Datenpunkte in den Metadaten zu untersuchen, wurde Mittelwert, Standardabweichung, Minimum und Maximum jedes Umgebungszustandes bestimmt. Diese sind in Tabelle 8.25 abgebildet. Erkennbar war, dass für die mit Grün markierten Umgebungszustände eine bessere Gleichverteilung vorlag, als bei den anderen. Dies ist daran zu erkennen, dass der Mittelwert und die Standardabweichung nahe den optimalen Werten für eine Gleichverteilung lag. Dabei kann geschlussfolgert werden, dass für die grün markierten Umgebungszustände eine bessere Approximation des Leistungsindex erreicht werden kann. Zurückzuführen war die nicht optimale Gleichverteilung auf die verwendeten Generierungsparameter bei der Datenerzeugung (siehe Abschnitt 4.4). Da eine gleichverteilte Abtastung des Situationsraumes zu aufwendig war, wurde ein stochastischer Prozess bei der Generierung der Metadaten verwendet.

Zusätzlich zu den Analysen der Umgebungszustände wurden auch die Mittelwerte, Standardabweichungen, Minimum und Maximum der Leistungsindizes des BLOB-, CC-, DPMund TM-Perzeptionsmoduls berechnet, wie der Tabelle 8.25 in den letzten vier Zeilen zu entnehmen ist. Hierbei konnte geschlussfolgert werden, dass der *DPM-Detektor* die höchste und das kantenbasierte *Template Matching* die niedrigste Detektionsgenauigkeit besaß.

Variablenname	Symbol	Mittelwert	$\overline{F_{opt}}$	σ	σ_{opt}	Min.	Max.
Bewölkungsgrad	Wo	0,4461	0,5	0,322	0,289	0	1
Nebeldichte	$ ho_f'$	0,4019	0,5	0,319	0,289	0	1
Niederschlagsmenge	Q'_r	0,3928	0,5	0,321	0,286	0	0,99
Elevationswinkel	heta'	-0,6166	-0,66	0,194	0,191	-0,99	-0,33
Bodenpixelauflösung	g'	0,4873	0,48	0,762	0,162	0,20	0,76
Tageszeit (X-Wert)	x_t	-0,1734	0	0,673	0,577	-1	1
Tageszeit (Y-Wert)	y_t	-0,1631	0	0,701	0,577	-1	1
Monat (X-Wert)	<i>x</i> _{<i>M</i>}	-0,0976	0	0,645	0,577	-1	1
Monat (Y-Wert)	y_M	-0,2314	0	0,722	0,577	-1	1
Vegetation	t_v	0,4228	0,5	0,317	0,289	0	1
Ebene	t _{os}	0,3979	0,5	0,333	0,289	0	1
Wasser	t _w	0,0731	0,5	0,201	0,289	0	1
Gebäude	t_b	0,0412	0,39	0,088	0,222	0	0,77
Straße	t_s	0,0571	0,28	0,068	0,162	0	0,56
Reliefwert	t_t	0,0110	0,38	0,027	0,219	0	0,76
Leistungsindex (BLOB)	p_{BLOB}	0,7960		0,308		0	1
Leistungsindex (CC)	p_{CC}	0,8289		0,270		0	1
Leistungsindex (DPM)	p_{DPM}	0,8559		0,252		0	1
Leistungsindex (TM)	p_{TM}	0,1445		0,264		0	1

Tabelle 8.25: Mittelwert, Standardabweichung, Minimum und Maximum der Metadaten.

Bei der Vorverarbeitung des Modelldatensatzes nach Abschnitt 5.2.1, wurde der Faktor für die Skalierung der Bodenpixelauflösung zu $f_g = 7,2$ und der Faktor für die Skalierung des Reliefwertes zu $f_r = 0,0365$ bestimmt.

8.2.3 Überprüfung auf Nichtlinearität

Der Einfluss der Umgebungszustände auf den modellierten Leistungsindex wurde mit dem linearen Korrelationskoeffizienten anhand der Metadaten analysiert, um diese auf Nichtlinearität zu überprüfen. Die Ergebnisse sind in Tabelle 8.26 dargestellt. Dabei beschreibt der Korrelationskoeffizient r die Einflussstärke, und der P-Wert gibt die Signifikanz an. Werte, die mit Rot gekennzeichnet sind, haben keine ausreichende Signifikanz (> 0,05), und damit besteht keine Korrelation zwischen Umgebungszustand und Leistungsindex. Die mit Grün markierten Korrelationskoeffizienten waren signifikant und lieferten eine Aussage über die Beziehung zwischen Umgebungszustand und Leistungsindex. Alle anderen unmarkierten deuteten auf eine nichtlineare Beziehung Werte hin. Positive Werte der Korrelationskoeffizienten zeigten an, dass bei steigendem Wert des betreffenden Umgebungszustands der Leistungsindex ebenfalls stieg. Ein umgekehrtes Verhalten lag bei negativen Korrelationskoeffizienten vor.

	BL	OB	С	С	DF	M	ТМ	
	r	P-Wert	r	P-Wert	r	P-Wert	r	P-Wert
Bewölkungsgrad	0,1765	0,0000	0,0167	0,0000	-0,0041	0,0138	0,0096	0,0000
Nebeldichte	0,2508	0,0000	0,0022	0,1815	-0,0013	0,4304	-0,0519	0,0000
Niederschlag	0,2658	0,0000	-0,0292	0,0000	-0,0164	0,0000	-0,0718	0,0000
Elevation	-0,1228	0,0000	-0,0430	0,0000	-0,0012	0,4918	0,0467	0,0000
GSD	0,0828	0,0000	0,1517	0,0000	-0,1172	0,0000	0,0293	0,0000
Tageszeit X	0,1385	0,0000	-0,0187	0,0000	-0,0445	0,0000	0,0007	0,6619
Tageszeit Y	0,2177	0,0000	0,0095	0,0000	-0,0404	0,0000	0,0177	0,0000
Monat X	0,0080	0,0000	0,0654	0,0000	0,0018	0,2797	0,1314	0,0000
Monat Y	0,1884	0,0000	0,0298	0,0000	0,0128	0,0000	0,1196	0,0000
Vegetation	-0,0762	0,0000	-0,0316	0,0000	-0,0467	0,0000	0,0309	0,0000
Ebene	0,0682	0,0000	-0,0069	0,0000	0,0598	0,0000	-0,0416	0,0000
Wasser	0,0041	0,0144	-0,0085	0,0000	0,0067	0,0001	0,0101	0,0000
Gebäude	-0,0162	0,0000	0,1073	0,0000	-0,0376	0,0000	-0,0085	0,0000
Straße	0,0338	0,0000	0,0580	0,0000	-0,0436	0,0000	0,0382	0,0000
Reliefwert	-0,0916	0,0000	-0,0678	0,0000	0,0359	0,0000	-0,0290	0,0000

Tabelle 8.26: Ergebnisse der LCC-Analyse.

Die Mehrheit der Umgebungszustände zeigte eine nichtlineare Beziehung gegenüber dem Leistungsindex. Daher können die gewählten Verfahren zur Funktionsapproximation (KNN und NEFPROX) zur Erzeugung der Leistungsmodelle als geeignet betrachtet werden, da gerade diese für nichtlineare Problemstellungen verwendet werden.

8.2.4 Vergleich der ML-Verfahren zur Leistungsmodellierung

Dieser Abschnitt vergleicht die Leistungsmodelle, welche durch die beiden verwendeten ML-Verfahren erzeugt wurden. Ziel ist es, die geeignetsten Leistungsmodelle zu identifizieren und diese bei der Evaluierung der Algorithmenauswahl einzusetzen.

8.2.4.1 Analyse der Modellgüte

Zur Bestimmung der Modellgüte mussten im Vorfeld der optimale Bias der Objektdetektoren durch die Analyse der ROC-Kurve anhand der Trainingsdaten ermittelt werden (vlg. Abschnitt 5.2.1). Tabelle 8.27 zeigt die ermittelten Werte.

	BLOB	CC	DPM	ТМ
Bias	0,6571	1,9968	-0,3498	183380
AUC	0,6437	0,5912	0,7395	0,3961

Tabelle 8.27: Optimaler Bias und AUC der Objektdetektoren.

Die Ergebnisse der Modellgüte für die beiden Verfahren zur Funktionsapproximation, *künstliche neuronale Netze* und der *NEFPROX*-Algorithmus, sind in Tabelle 8.28 aufgeführt. Für die einzelnen Leistungsmodelle sind jeweils die durchlaufenen Trainingsepochen (Epochen) angegeben. Außerdem ist jene Epoche (Beste Epoche) angezeigt, bei der das entsprechende Modell den geringsten MSE anhand des Validierungsdatensatzes aufwies. Zusätzlich dazu ist der MSE, welcher am Trainingsdatensatz erreicht werden konnte, angegeben.

Im Vergleich besaßen die KNN-Leistungsmodelle einen geringeren MSE (in der Tabelle grün hinterlegt) und können damit die Funktionsapproximation besser als die durch NEFPROX erzeugten Leistungsmodelle lösen. Weiterhin konnte festgestellt werden, dass das BLOB- Leistungsmodell die größte Abweichung und das TM-Leistungsmodell die kleinste Abweichung zum Validierungsdatensatz aufweist. Die CC- und DPM-Leistungsmodelle liegen dazwischen. Die KNN-Leistungsmodelle besitzen zwar die bessere Modellgüte anhand des Validierungsdatensatzes, aber für einen praktischen Einsatz dieser sollen im folgenden Abschnitt typische Leistungsverläufe analysiert werden.

	Parameter	BLOB	CC	DPM	ТМ
	Epochen	1493	2040	2035	1567
N	Beste Epoche	1443	1990	1985	1517
Kľ	MSE (Training)	0,0672	0,0537	0,0487	0,0383
	MSE (Validierung)	0,0690	0,0560	0,0523	0,0388
×	Epochen	52	54	59	59
RO	Beste Epoche	1	3	8	8
IEFF	MSE (Training)	0,0932	0,0799	0,0778	0,0675
Z	MSE (Validierung)	0,0950	0,0808	0,0784	0,0694

Tabelle 8.28: Ergebnisse des Trainings der Leistungsmodelle mittels KNN und NEFPROX. Die mit Grün markierten MSE-Werte sind diejenigen Leistungsmodelle, die den geringsten erreicht hatten.

8.2.4.2 Analyse der Leistungsverläufe

In den folgenden Untersuchungen wurde exemplarisch die Modellaussage (der Leistungsindex) der NEFPROX- und KNN-Leistungsmodelle in Abhängigkeit des Elevationswinkels und der Bodenpixelauflösung ermittelt, um sie vergleichend gegenüberzustellen. Hierbei wurden die übrigen Umgebungszustände zu den Werten aus Tabelle 8.29 festgelegt.

Q'_r	$oldsymbol{ ho}_f'$	W _o	t	М	θ	g	t_w	t _{os}	t _b	ts	t_v	<i>r</i> ′
0	0	0,5	14:00	7	-	_	0	0,3	0	0	0,7	0
	Tabelle 8.29: Gewählte Umgebungszustände für die exemplarische Untersuchung.											

Bild 8.12 zeigt den Leistungsindex des BLOB-Detektors ermittelt aus dem KNN-Leistungsmodell. Bild 8.13 präsentiert die Leistungsindizes des NEFPROX-Leistungsmodells.



Bild 8.12: Leistungsindizes des KNN-Leistungsmodells des BLOB-Detektors in Abhängigkeit des Elevationswinkels und der Bodenpixelauflösung.

Im Allgemeinen zeigten beide Modelle ein ähnliches Verhalten. Die höheren Leistungsindizes wurden jeweils bei geringen Bodenpixelauflösungen und flacherem Elevationswinkel erreicht. Die niedrigsten Leistungsindizes wurden bei beiden Modellen bei hohen Bodenpixelauflösungen und flachem Elevationswinkel angezeigt. Bei genauerer Betrachtung der beiden Leistungsverläufe stellte sich heraus, dass das KNN-Leistungsmodell deutlichere Ausprägungen der Leistungsniveaus zeigte als das NEFPROX-Leistungsmodell. Das NEFPROX-Leistungsmodell lieferte homogenere Leistungsverläufe und konnte damit Details im Leistungsverlauf nicht komplett erfassen. Zurückzuführen sind die Gründe auf die verwendete maximale Anzahl von 5 Zugehörigkeitsfunktionen pro Umgebungszustand.



Bild 8.13: Leistungsindizes des NEFPROX-Leistungsmodells des BLOB-Detektors in Abhängigkeit des Elevationswinkels und der Bodenpixelauflösung.

Die Leistungsindizes des *Template Matching*, ermittelt durch das KNN- und NEFPROX-Leistungsmodell, sind in Bild 8.14 und Bild 8.15 dargelegt.

Hierbei zeigten die beiden Leistungsmodelle vergleichbare Leistungsverläufe. Bei dem NEFPROX-Leistungsmodell konnte ein Artefakt bei höheren Bodenpixelauflösungen und flacherem Elevationswinkel identifiziert werden (Rechteck). Zurückzuführen war dieses Artefakt auf die Reduzierung der Regelbasis. Dieses wurde durchgeführt, um einer Überanpassung des NEFPROX-Leistungsmodells entgegenzuwirken. Damit lagen in dem Bereich des Artefakts nicht genügend Regeln zur Verfügung, um einen kontinuierlichen Leistungsverlauf darzustellen.



Bild 8.14: Leistungsindizes des KNN-Leistungsmodells des Template Matchings in Abhängigkeit des Elevationswinkels und der Bodenpixelauflösung.



Bild 8.15: Leistungsindizes des NEFPROX-Leistungsmodells des Template Matchings in Abhängigkeit des Elevationswinkels und der Bodenpixelauflösung.

In Bild 8.16a sind die Leistungsindizes der Klassifizierungskaskade anhand des KNN-Leistungsmodells dargestellt. Im Vergleich zu dem Leistungsverlauf des NEFPROX-Leistungsmodell (siehe Bild 8.16b) zeigte sich, dass wie beim NEFPROX-Leistungsmodell des *Template Matchings*, ein Artefakt in Form eines Rechteckes zu sehen war. Dieses war wieder auf die Reduzierung der Regelbasis zurückzuführen. Im Allgemeinen lieferte das NEFPROX-Leistungsmodell niedrigere Leistungsindizes als das KNN-Leistungsmodell.



Bild 8.16: Leistungsindex der Klassifizierungskaskade in Abhängigkeit des Elevationswinkels und der Bodenpixelauflösung. In (a) sind die Leistungsindizes des KNN-Leistungsmodells und in (b) die des NEFPROX-Leistungsmodells dargestellt.

Die Leistungsindizes des DPM-Detektors ermittelt durch das KNN-Leistungsmodell sind in Bild 8.17a einzusehen. Hierbei war erkennbar, dass ein sehr hoher Leistungsindex in großen Bereichen auftrat. Lediglich bei hohen Bodenpixelauflösungen und sehr flachen Elevationswinkel reduzierte sich der Leistungsindex. In Bild 8.17b sind die Leistungsindizes der NEFPROX-Leistungsmodells dargestellt. Hierbei zeigte sich zwar ein ähnlicher Leistungsverlauf, aber das rechteckige Artefakt war wieder deutlich zu erkennen.



Bild 8.17: Leistungsindex des DPM-Detektors in Abhängigkeit des Elevationswinkels und der Bodenpixelauflösung. In (a) sind die Leistungsindizes des KNN-Leistungsmodells und in (b) die des NEFPROX-Leistungsmodells dargestellt.

8.2.4.3 Auswahl der durch ML-Verfahren erzeugten Leistungsmodelle

Die Analyse der Modellgüte (vgl. Abschnitt 8.2.4.1) kam zu dem Schluss, dass die KNN-Leistungsmodelle eine höhere Modellgüte besaßen als die NEFPROX-Leistungsmodelle. In den weiteren Analysen bzgl. der Leistungsverläufe (siehe Abschnitt 8.2.4.2) konnte festgestellt werden, dass die NEFPROX-Leistungsmodelle homogenere Leistungsverläufe aufzeigten, und dadurch weniger Detailverläufe als das KNN-Leistungsmodell berücksichtigten. Außerdem besaß das NEFPROX-Leistungsmodell Artefakte in den Leistungsverläufen, welche auf einen Optimierungsschritt in der Erzeugung der NEFPROX-Leistungsmodelle zurückzuführen waren. Aus diesen Gründen wurden für die folgende Evaluierung der Algorithmenauswahl die KNN-Leistungsmodelle anstatt der NEFPROX-Leistungsmodelle eingesetzt.

8.3 Evaluierung der Algorithmenauswahl

In diesem Abschnitt werden die Ergebnisse der Evaluierung der Algorithmenauswahl anhand der in den 5 Experimentalmissionen gewonnen Evaluierungsdaten (siehe Abschnitt 7.4) dargestellt. Für die Evaluierung wurde ein *Gütemaß q_p* eingeführt, um die Verbesserung der Leistungsfähigkeit durch eine Algorithmenauswahl quantifizieren zu können. Das *Gütemaß* berechnet sich mit

$$q_p = 1 - \sum_{i=1}^{N} (p_i - 1)^2, \tag{8.3}$$

wobei N die Anzahl der Datenpunkte in den Evaluierungsdaten in einer Mission angibt und p_i den Leistungsindex am Zeitenpunkt bzw. Datenpunkt i enthält. Das *Gütemaß* p_t berechnet damit den gemittelten Wert der Fehlerquadrate. Außerdem wurde ein Maß, die *Genauigkeit*, eingeführt, um festzustellen, welchen Grad der Übereinstimmung die Algorithmenauswahl mittels den Modellen gegenüber den Messungen besitzt.

Die Darstellung der Ergebnisse war für jede Mission gleich und wird vorab, an dieser Stelle beschrieben. Es werden zwei Diagramme verwendet, um die folgenden Sachverhalte darzustellen:

- **Diagramm 1**: Dieses Diagramm stellt den Leistungsindex einer jeden betrachteten Perzeptionskette über die Missionszeit dar (siehe z. B. Bild 8.20). Dabei zeigt der obere Graph die Messung (*Ground Truth*) und der untere Graph die Leistungsindizes der Leistungsmodelle.
- **Diagramm 2**: In diesem Diagramm werden zwei Graphen verwendet, um die Algorithmenauswahl zu visualisieren (siehe z. B. Bild 8.21). Dabei zeigt der obere Graph die resultierenden Leistungsindizes durch Algorithmenauswahl mittels Messung und Leistungsmodellen. Außerdem ist die Differenz zwischen diesen in Rot gekennzeichnet. Des Weiteren zeigt der untere Graph die zu jedem Zeitschritt ausgewählte Perzeptionskette an.

Abschließend wird in drei Tabellen dargestellt, welche *Gütemaße* die jeweiligen Perzeptionsketten besitzen und welche Verbesserung mittels der Algorithmenauswahl (AA) erreicht wurde. Die drei Tabellen enthalten die folgenden Informationen:

- **Tabelle 1:** Diese Tabelle zeigt die *Gütemaße* der einzelnen Perzeptionsketten anhand der Messung und der Leistungsmodelle.
- **Tabelle 2:** Die *Gütemaße* der Algorithmenauswahl (Messung und Modell) sowie die *Genauigkeit* der Algorithmenauswahl resultierend aus den Leistungsmodellen gibt diese Tabelle an. Visuell ist die *Genauigkeit* die Fläche unter der Leistungsindexkurve der Messung geteilt durch die Fläche unter der Modell-Leistungsindexkurve.
- **Tabelle 3:** Diese Tabelle gibt die prozentualen Verbesserungen, welche durch die eingesetzte Algorithmenauswahl erreicht wurden, an. Außerdem zeigt sie das mögliche Potential, dass theoretisch durch eine Algorithmenauswahl erreichbar gewesen wäre.

Im Folgenden werden für die einzelnen Perzeptionsketten aus Darstellungsgründen die Kürzel aus Abschnitt 3.3.2 verwendet.

8.3.1 Verfahren zur Evaluierung der Algorithmenauswahl

Dieser Abschnitt behandelt das Verfahren zur Evaluierung der Algorithmenauswahl. Dabei werden die Aussagen der erzeugten Leistungsmodelle mit einer direkten Messung der Algorithmenleistung (Ground Truth) verglichen. Bild 8.18 zeigt das verwendete Verfahren, um für jede Mission die Leistungsindizes jeder Perzeptionskette zu messen und über die Leistungsmodelle zu bestimmen.



Bild 8.18: Verfahren zum Vergleich zwischen Messung und den Leistungsmodellen.

Um die Messung durchzuführen, wurde das Verfahren zur Leistungsbestimmung wie bei der Generierung der Metadaten herangezogen (vgl. Abschnitt 4.7) und auf alle Perzeptionsketten c_j angewendet, welche den des Evaluierungsdatensatzes $\tilde{\mathcal{E}}$ (Missionen) verarbeiten. Da hier jede Kette mit einem Objekterkennungsalgorithmus endet, lässt sich der Vertrauenswert, der *Score s*, und durch die Annotationen der Evaluierungsdaten das *Label l* bestimmen. Dabei werden immer mehrere Bild-Annotierungs-Paare zusammengefasst, um genügend *Labels* und *Scores* für eine Ermittlung des Leistungsindex zu erhalten. Würde jedes Bild-Annotierungs-Paar einzeln ausgewertet werden, käme es zu Ausreißern und der Leistungsindex würde über die Zeit stark fluktuieren. Deshalb wird bei der Messung des Leistungsindex pro Zeitschritt ein Zeitfenster gewählt, indem alle Bild-Annotierungs-Paare analysiert werden. Vergleichbar ist dieses Verfahren mit dem *Sliding Window* Ansatz, bei dem ein Betrachtungsfenster zeitlich jeweils ein Bild weitergeschoben wird, wie in Bild 8.19 dargestellt. Gewählt wurde die Größe des Betrachtungsfensters auf Basis der Aufnahmerate der einzelnen Missionsabschnitte, indem mindestens 20 Bild-Annotationspaare akkumuliert werden.



Bild 8.19: Darstellung der Messung des Leistungsindexes. Das Sliding Window ist durch einen schwarzen Rahmen dargestellt und die vorherige Position mit einer grauen Fläche angedeutet.

Schließlich wird für die Berechnung des Leistungsindex das F_1 -Maß (siehe Abschnitt 4.6) bestimmt, unter Zuhilfenahme des optimalen Bias b_i (siehe Abschnitt 5.2.1). Das Resultat der Messung ist der Leistungsindex pro Perzeptionskette und Zeitschritt P_g . Für die Ermittlung der Leistungsindizes mit Hilfe der Leistungsmodelle wurden die Umgebungszustände des Evaluierungsdatensatzes $\tilde{\mathcal{E}}$ über das gewählte *Sliding Window* gemittelt und über die Vorverarbeitung (siehe Abschnitt 5.2.1) normiert, wobei die Bodenpixelauflösung und der Reliefwert mittels der Faktoren f_g und f_r (vgl. Abschnitt 8.2.2) skaliert wurden. Anschließend wurde der aufbereitete Umgebungszustand allen Leistungsmodellen präsentiert, welche aus diesem dann die jeweiligen Leistungsindizes berechneten. Über den Perzeptionsgraphen (siehe Abschnitt 3.3.2) können die Leistungsindizes pro Perzeptionskette und Missionszeitschritt akkumuliert werden. Daraus resultiert eine Matrix P_M mit allen Leistungsindizes über die Missionszeit einer jeden Perzeptionskette.

8.3.2 Mission: Observierung

Die Beschreibung dieser Mission sowie die aufgezeichneten Umgebungsmerkmale über die Missionszeit sind in Abschnitt 7.4.1.1 zu finden. Die Mission *Observierung* behandelt die folgenden Untersuchungsgegenstände:

- 1 Analyse der erzeugten Leistungsmodelle
 - 1.2 Analyse der durch ML-Verfahren erzeugten Leistungsmodelle
- 2 Evaluierung der Algorithmenauswahl
 - 2.2 Evaluierung der Algorithmenauswahl hinsichtlich wechselnder Tageszeiten
 - 2.3 Evaluierung der Algorithmenauswahl unter atmosphärischen Einflüssen

Im speziellen wurden deshalb die Einflüsse der Wettergrößen, der Tageszeit und des Elevationswinkels auf die Leistungsmodelle des Template Matchings, des BLOB-Detektors, der Klassifizierungskaskade und des DPM-Detektors untersucht. Gesondert wurde der Einfluss der Bodenbedeckung *Wasser* evaluiert. Wurde diese in den Leistungsmodellen berücksichtigt, führt das beim DPM-Leistungsmodell zu Abweichungen gegenüber der Messung zu Beginn der Mission. Dies konnte auf die *Metadaten* zurückgeführt werden, da dort zu wenige Beispiele mit dieser Bodenbedeckung aufzufinden waren. Daher wurde die Kategorie hier und in den folgenden Missionen nicht weiter berücksichtigt, indem dieser Umgebungszustand auf 0 gesetzt wurde. Zunächst findet eine Gesamtbetrachtung der Mission in Abschnitt 8.3.2.1 statt. In den anschließenden Abschnitten 8.3.2.2 bis 8.3.2.6 werden für die fünf Missionsabschnitte (siehe Anhang E) die Ergebnisse gesondert präsentiert.

8.3.2.1 Gesamtbetrachtung

Bild 8.20 zeigt den Verlauf der Leistungsindizes während der Mission *Observierung*. Für alle Algorithmen erzielten die Leistungsmodelle einen vergleichbaren Kurvenverlauf der Leistungsindizes über die Tageszeit gegenüber der Messung. Daraus konnte geschlussfolgert werden, dass der Einfluss der Tageszeit passend abgebildet wurde. Daneben war ein Anstieg des CC- und DPM-Leistungsmodells bei steigendem Bewölkungsgrad ab 16:00 Uhr Missionszeit zu erkennen. Gerade am Ende der Mission konnte festgestellt werden, dass ein normierter Bewölkungsgrad und normierte Nebeldichte größer als 0,5 einen negativen Effekt auf den Leistungsindex besaß. Ein aussagekräftiger Effekt des Niederschlages konnte in dieser Mission nicht festgehalten werden. Außerdem konnte festgestellt werden, dass das *Template Matching* in dieser Mission die niedrigste Detektionsgenauigkeit aufwies. Des Weiteren konnten folgende Feststellungen getroffen werden:

• Das KNN-Leistungsmodell für die Klassifikationskaskade lieferte generell höhere Leistungsindizes als die Messung, und das DPM-Leistungsmodell zeigte ab 17:30 Uhr Missionszeit einen flacheren Abfall, als dies die Messung aufzeigte. Zurückzuführen waren diese Abweichungen auf die Generalisierungseigenschaften der KNN.

- Alle Modelle sind sensitiv auf den schwankenden Elevationswinkel, welcher aus dem *Loiter-Pattern* des Flächenluftfahrzeuges resultierte. Dies stellte sich optisch als Schwingung des Leistungsindex über die Missionszeit dar.
- Das DPM-Leistungsmodell liefert in einigen Abschnitten höhere Leistungsindizes als 1. Grund dafür sind keine oder zu wenige Stützstellen, die diesen Randbereich beschreiben. Dadurch extrapolierten die künstlichen neuronalen Netze in diesen Bereichen, wodurch kleinere Werte als 0 und größere Werte als 1 auftreten konnten. Diese wurden deshalb limitiert zu $0 \le p \le 1$.

Im Allgemeinen zeigte sich, dass der DPM-Detektor in Tageslichtsituationen leistungsfähiger als die anderen Verfahren war. Nachts ist der BLOB-Detektor nur dann einsetzbar, wenn das Fahrzeug eine entsprechende thermische Signatur aufwies (z. B. bei laufendem Fahrzeugmotor). Dies war in der Mission ab 02:00 Uhr der Fall.



Leistungsindizes der Messung

Bild 8.20: Leistungsindizes der Messung und der KNN-Leistungsmodelle der Mission Observierung.

Die resultierende Algorithmenauswahl der Mission ist in Bild 8.21 dargestellt. Die Differenzen zwischen den Leistungsindizes der Messung und des Leistungsmodells konnten vor allem in der Morgen- und Abenddämmerung beobachtet werden. In Tabelle 8.30 sind die *Gütemaße* (siehe Definition in Abschnitt 8.3) der Perzeptionsketten gegenübergestellt.

	BLOB	CC	DPM	ТМ
Messung	0,089	0,085	0,495	0,020
Modell	0,091	0,234	0,536	0,021
Differenz	0,001	0,150	0,042	0,001

Tabelle 8.30: Leistungsdifferenzen zwischen Messung und Modellen der Mission *Observierung*. Die Werte in der Tabelle wurden über das Gütemaß q_p bestimmt.

Zu bemerken ist, dass die BLOB- und TM-Leistungsmodelle für diese Mission weniger Differenz gegenüber der Messung zeigten als die anderen. Als generell leistungsfähigster Algorithmus hat sich für diese Mission der DPM-Detektor erwiesen.



Bild 8.21: Resultierende Algorithmenauswahl in der Mission *Observierung*. Auf der Ordinate sind die Nummern der einzelnen Algorithmen aufgetragen. Diese sind in der Legende in Bild 8.20 kenntlich gemacht.

Die Auswertung der Genauigkeit (siehe Definition in Abschnitt 8.3) der Algorithmenauswahl ist in Tabelle 8.31 zusammengefasst. Sie lässt sich anhand der Differenz der Gütemaße der 96,14 %. Algorithmenauswahl feststellen und betrug Tabelle 8.32 zeigt die Leistungssteigerung durch die verwendete Algorithmenauswahl anhand der einzelnen Perzeptionsketten. Im Verhältnis zu einer durchgängigen Nutzung des leistungsstärksten Algorithmus (DPM) konnte durch zusätzliche Algorithmenauswahl ein Plus von 11,17 % Leistung erreicht werden. Maximal wären 15,63 % möglich gewesen, was zu einer Abweichung von 4,47 % zwischen Messung und Modell führte. Gegenüber der schwächten Perzeptionskette war eine Verbesserung von 2696,41 % erreichbar.

	Algorithmenauswahl
Messung	0,572
Modell	0,550
Abweichung	-3,86 %
Genauigkeit	96,14 %

Tabelle 8.31: Auswertung der Genauigkeit der Algorithmenauswahl der Mission Observierung.

	ТМ	CC	BLOB	DPM
Messung	0,020	0,085	0,089	0,495
Verbesserung AA mittels Messung in [%]	2808,76	576,51	539,61	15,63
Verbesserung AA mittels Modells in [%]	2696,41	550,38	514,91	11,17
Abweichung in [%]	-112,34	-26,13	-24,70	-4,47

Tabelle 8.32: Leistungssteigerung durch Algorithmenauswahl der Mission *Observierung*. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.2.2 Ergebnisse aus Missionsabschnitt 1

In diesem Missionsabschnitt wurde an einer Hafenanlage stehendes Fahrzeug beobachtet. Der Verlauf der Leistungsindizes der Messung und der Modelle über diesen Missionsabschnitt ist in Bild 8.22 dargestellt. Der Leistungsindex des DPM stieg im Verlauf des Missionsabschnittes an, während sich die Leistung der anderen Algorithmen kaum veränderte. Die Leistungssteigerung im Laufe des Abschnittes war durch den Schattenwurf eines dem Fahrzeug nahestehendem Silo zurückzuführen. Zu Beginn der Mission bildete der Schattenwurf zusätzliche Kanten auf dem Fahrzeug (siehe Bild 8.23a), welche die Detektionsgenauigkeit reduzierte. Im Verlaufe des Missionsabschnittes wurde der Schattenwurf diffuser (siehe Bild 8.23b) und damit die Gradienten der Schatten schwächer. Dies resultierte in einen gesteigertem Leistungsverhalten gegenüber der Situation am Anfang des Missionsabschnittes.



Bild 8.22: Leistungsindizes der Messung und Leistungsmodelle der Mission Observierung Abschnitt 1.



Bild 8.23: Verbesserung der Erkennbarkeit des Fahrzeugs bei diffuserem Schattenwurf. Die Abbildung in (a) zeigt den Schattenwurf zu Beginn (13:00 Uhr) und (b) visualisiert den diffuseren Schattenwurf am Ende (16:00 Uhr) des Missionsabschnittes.

Der Vergleich der Algorithmenauswahl zwischen Messung und Modellen präsentiert Bild 8.24. In diesem Missionsabschnitt war keine Differenz bei der Algorithmenauswahl festzustellen. Über den gesamten Missionsabschnitt zeigte das DPM den höchsten Leistungsindex und wurde demnach exklusiv ausgewählt.



Bild 8.24: Resultierende Algorithmenauswahl in der Mission *Observierung* Abschnitt 1. Die Algorithmennummern geben die folgenden Perzeptionsmodule an: #1 BLOB, #2 CC, #3 DPM und #4 TM.

Tabelle 8.33 zeigt die Differenzen der einzelnen Algorithmen anhand des Gütemaßes q_t . Das BLOB- und TM-Leistungsmodell besaßen die geringsten Abweichungen und das CC-Modell die größte Differenz gegenüber der Messung. Als leistungsstärkster Algorithmus hatte sich in diesem Missionsabschnitt der DPM-Detektor herausgestellt.

	BLOB	CC	DPM	ТМ
Messung	0,130	0,054	0,795	0,025
Modell	0,138	0,274	0,834	0,014
Differenz	0,008	0,219	0,039	-0,011

Tabelle 8.33: Leistungsdifferenzen zwischen Messung und Modellen der Mission Observierung Abschnitt 1.

Da in diesem Missionsabschnitt ein Algorithmenwechsel keine Verbesserung der Leistung herbeiführte, konnte der geeignetste Algorithmus (DPM) über den gesamten Missionsabschnitt eingesetzt werden. Damit war die Genauigkeit der Algorithmenauswahl durch die Leistungsmodelle 100 %, wie in Tabelle 8.34 dargestellt. Die Auswertung der Leistungssteigerungen durch die Algorithmenauswahl zeigt Tabelle 8.35.

	Algorithmenauswahl
Messung	0,795
Modell	0,795
Abweichung	0 %
Genauigkeit	100 %

Tabelle 8.34: Auswertung der Genauigkeit anhand der Algorithmenauswahl der Mission Observierung Abschnitt 1.

	ТМ	CC	BLOB	DPM
Messung	0,025	0,054	0,130	0,795
Verbesserung AA mittels Messung in [%]	3052,59	1362,21	511,63	0,00
Verbesserung AA mittels Modells in [%]	3052,59	1362,21	511,63	0,00
Abweichung in [%]	0,00	0,00	0,00	0,00

Tabelle 8.35: Leistungssteigerung durch Algorithmenauswahl der Mission *Observierung* Abschnitt 1. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.2.3 Ergebnisse aus Missionsabschnitt 2

Während des zweiten Missionsabschnittes fuhr ein Fahrer das Fahrzeug von der Hafenanlage zu einer Einrichtung. Dabei passierte das Fahrzeug gegen Anfang des Missionsabschnittes eine Stadt. Dabei kam es zu partiellen oder vollständigen Verdeckungen (siehe Bild 8.26) des Fahrzeuges. Daher zeigte der DPM-Leistungsindex der Messung in Bild 8.25 zwischen 16:00 und 16:05 Uhr eine geringere Detektionsgenauigkeit als im weiteren Verlauf des Missionsabschnittes. Das DPM-Leistungsmodell konnte diesen Effekt über die Bodenbeschaffung *Gebäude* erfassen.



Bild 8.25: Leistungsindizes der Messung und Leistungsmodelle der Mission *Observierung* Abschnitt 2.

Die resultierende Algorithmenauswahl aus den Leistungsindizes in Bild 8.25 präsentiert Bild 8.27. Dabei stimmten Messung und Leistungsmodelle bis auf eine geringe Differenz überein. Diese Abweichung zeigt sich gegen 16:04 Uhr Missionszeit, welche durch die erwähnte Verdeckungssituation zustande kam.

Die einzelnen Leistungsdifferenzen stellt Tabelle 8.36 dar. Bis auf das CC-Leistungsmodelle konnten die Leistungsmodelle einen vergleichbaren Leistungsindex bestimmen, was sich in der Differenz der Leistungsindizes niederschlug. Tabelle 8.37 beinhaltet die Auswertung der

Genauigkeit der Algorithmenauswahl. Hierbei konnte eine Genauigkeit von 99,89 % erreicht werden. Tabelle 8.38 zeigt die Leistungssteigerungen durch die verwendete Algorithmenauswahl. Maximal wäre gegenüber der leistungsstärksten Perzeptionskette (DPM) eine Verbesserung durch Algorithmenauswahl um noch 0,12 % möglich gewesen.



Bild 8.26: Verdeckungssituationen während des zweiten Missionsabschnittes der Mission *Observierung*. In (a) ist eine partielle und in (b) eine vollständige Verdeckung des zu beobachtenden Fahrzeuges zu sehen.



Bild 8.27: Resultierende Algorithmenauswahl in der Mission *Observierung* Abschnitt 2. Die Algorithmennummern geben die folgenden Perzeptionsmodule an: #1 BLOB, #2 CC, #3 DPM und #4 TM.

	BLOB	CC	DPM	ТМ
Messung	0,057	0,053	0,923	0,060
Modell	0,063	0,361	0,954	0,055
Differenz	0,006	0,308	0,030	-0,005

Tabelle 8.36: Leistungsdifferenzen zwischen Messung und Modellen der Mission Observierung Abschnitt 2.

	Algorithmenauswahl
Messung	0,924
Modell	0,923
Abweichung	-0,12 %
Genauigkeit	99,89 %

Tabelle 8.37: Auswertung der Genauigkeit anhand der Algorithmenauswahl der Mission Observierung Abschnitt 2.

	CC	BLOB	ТМ	DPM
Messung	0,053	0,057	0,060	0,923
Verbesserung AA mittels Messung in [%]	1629,73	1534,80	1436,85	0,12
Verbesserung AA mittels Modells in [%]	1627,73	1532,91	1435,08	0,00
Abweichung in [%]	-2,00	-1,89	-1,78	-0,12

Tabelle 8.38: Leistungssteigerung durch Algorithmenauswahl der Mission Observierung Abschnitt 2. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

Ergebnisse aus Missionsabschnitt 3 8.3.2.4

In diesem Missionsabschnitt stand das Fahrzeug vor einer Einrichtung während die Abenddämmerung einsetzte und es schließlich Nacht wurde. Dabei stieg die Niederschlagsmenge und der Bewölkungsgrad an. Anhand des DPM-Leistungsindex über den Missionsabschnitt, abgebildet in Bild 8.28, zeigte sich der Beginn des Sonnenunterganges durch die Verminderung dessen. Durch fehlendes Tageslicht konnte ab ca. 17:30 Uhr im visuellen Spektrum das Fahrzeug nicht mehr ausgemacht werden.



Leistungsindizes der Messung

Bild 8.28: Leistungsindizes der Messung und Leistungsmodelle der Mission Observierung Abschnitt 3.

Durch den Vergleich der Leistungsindizes der Messung und der Modelle (siehe Bild 8.28) konnte festgestellt werden, dass die Phase der Leistungsreduzierung bei den Modellen gestreckter war. Außerdem bestimmte das CC-Leistungsmodell einen höheren Leistungsindex als dies die Messung aufdeckte. Im Vergleich der Algorithmenauswahl (siehe Bild 8.29) zeigte sich die Differenz hauptsächlich im Bereich der angesprochenen Dämmerungsphase.



Bild 8.29: Resultierende Algorithmenauswahl in der Mission *Observierung* Abschnitt 3. Die Algorithmennummern geben die folgenden Perzeptionsmodule an: #1 BLOB, #2 CC, #3 DPM und #4 TM.

Die Leistungsdifferenzen zwischen der Messung und den Leistungsmodellen präsentiert Tabelle 8.39. Das BLOB- und TM-Leistungsmodell besaßen gegenüber dem CC- und DPM-Leistungsmodell geringere Differenzen. Über den gesamten Missionsabschnitt war die Klassifikationskaskade der leistungsstärkste Algorithmus.

Bei der Auswertung der Algorithmenauswahl (vgl. Tabelle 8.41) konnte festgestellt werden, dass eine Verbesserung von 37,40 % erreicht wurde. Maximal wäre eine Leistungssteigerung von bis zu 49,41 % möglich gewesen. Die Genauigkeit der Algorithmenauswahl betrug in diesem Missionsabschnitt 91,75 % (siehe Tabelle 8.40).

	BLOB	CC	DPM	ТМ
Messung	0,025	0,138	0,068	0,003
Modell	0,034	0,498	0,188	0,006
Differenz	0,009	0,361	0,120	0,003

Tabelle 8.39: Leistungsdifferenzen zwischen Messung und Modellen der Mission Observierung Abschnitt 3.

	Algorithmenauswahl
Messung	0,206
Modell	0,189
Abweichung	-8,04 %
Genauigkeit	91,75 %

Tabelle 8.40: Auswertung der Genauigkeit anhand der Algorithmenauswahl der Mission Observierung Abschnitt 3.
	ТМ	BLOB	DPM	CC
Messung	0,003	0,025	0,068	0,138
Verbesserung AA mittels Messung in [%]	6907,84	707,07	200,75	49,41
Verbesserung AA mittels Modells in [%]	6344,48	642,19	176,57	37,40
Abweichung in [%]	-563,36	-64,88	-24,18	-12,01

Tabelle 8.41: Leistungssteigerung durch Algorithmenauswahl der Mission *Observierung* Abschnitt 3. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.2.5 Ergebnisse aus Missionsabschnitt 4

Im 4. Missionsabschnitt bewegte sich das Fahrzeug von der Einrichtung in eine Stadt über eine Landstraße. Dabei startete die Fahrt gegen 02:00 Uhr und endet gegen 02:04 Uhr. Die Leistungsindizes der Messung und der Modelle finden sich in Bild 8.30. Der BLOB-Detektor konnte das Fahrzeug während der Fahrt detektieren, da der Motor des Fahrzeuges eine Wärmequelle bildete und somit eine entsprechende thermische Signatur im LWIR-Sensorbild dargestellt wurde (siehe Bild 8.31a).

Nach der Abkühlungsphase des Motors verblasste die Wärmesignatur (siehe Bild 8.31b) und der BLOB-Detektor konnte das Fahrzeug nicht weiter erkennen. Da dem BLOB-Leistungsmodell keine Information über den Zustand des Fahrzeugs bekannt sein konnten, weichen die absoluten Leistungsindizes zwischen Messung und Modell ab. Allerdings zeigte das BLOB-Leistungsmodell einen erhöhten Leistungsindex zu Beginn des Missionsabschnittes, da der Bewölkungsgrad $W_o = 1$ war und ein flacher Elevationswinkel ($\theta < -45^\circ$) zu Beginn der Mission herrschte.



Bild 8.30: Leistungsindizes der Messung und Leistungsmodelle der Mission Observierung Abschnitt 4.



Bild 8.31: LWIR-Signatur des Fahrzeuges im 4. Abschnitt der Mission Observierung. In (a) ist die Signatur während der Fahrt und in (b) nach der Abkühlungsphase des Fahrzeugmotors dargestellt.

Bild 8.32 stellt die resultierende Algorithmenauswahl dar. Die Differenzen der einzelnen Leistungsmodelle gegenüber der Messung sind dabei in Tabelle 8.42 abgebildet. Die leistungsstärksten Algorithmen waren hierbei der BLOB- und DPM-Detektor. Die Auswertung der Algorithmenauswahl in Tabelle 8.42 zeigt, dass eine Verbesserung von 47,64 % erreicht wurde, wobei maximal 63,34 % möglich gewesen wären. Die Algorithmenauswahl durch die Leistungsmodelle konnten insgesamt eine Genauigkeit von 90,60 % erreichen, wie in Tabelle 8.43 dargestellt.



Bild 8.32: Resultierende Algorithmenauswahl in der Mission *Observierung* Abschnitt 4. Die Algorithmennummern geben die folgenden Perzeptionsmodule an: #1 BLOB, #2 CC, #3 DPM und #4 TM.

	BLOB	СС	DPM	ТМ
Messung	0,117	0,026	0,183	0,057
Modell	0,159	0,012	0,142	0,028
Differenz	0,042	-0,013	-0,040	-0,030

Tabelle 8.42: Leistungsdifferenzen zwischen Messung und Modellen der Mission Observierung Abschnitt 4.

	Algorithmenauswahl
Messung	0,298
Modell	0,270
Abweichung	-9,61 %
Genauigkeit	90,60 %

Tabelle 8.43: Auswertung der Genauigkeit anhand der Algorithmenauswahl der Mission Observierung Abschnitt 4.

	CC	ТМ	BLOB	DPM
Messung	0,026	0,057	0,117	0,183
Verbesserung AA mittels Messung in [%]	1064,69	421,89	154,84	63,34
Verbesserung AA mittels Modells in [%]	952,75	371,73	130,34	47,64
Abweichung in [%]	-111,93	-50,16	-24,49	-15,70

Tabelle 8.44: Leistungssteigerung durch Algorithmenauswahl der Mission Observierung Abschnitt 4. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

Ergebnisse aus Missionsabschnitt 5 8.3.2.6

Im letzten Missionsabschnitt fand die Observierung des Fahrzeuges vor einem Gebäude in einer Stadt statt. Dabei kreiste das Luftfahrzeug in der Nähe der Stadt, wodurch sich der Elevationswinkel zyklisch änderte. Außerdem fiel zu Beginn der Mission der Bewölkungsgrad auf 0,5 ab und um 05:30 Uhr zog Nebel auf, welcher um 8:30 Uhr wieder komplett verschwunden war. Bild 8.33 zeigt die Leistungsindizes der einzelnen Algorithmen durch Messung und durch die Leistungsmodelle.



Leistungsindizes der Messung

Bild 8.33: Leistungsindizes der Messung und Leistungsmodelle der Mission Observierung Abschnitt 5.

Die Leistungskurven zeigten dabei vergleichbare Formen. Da zu Beginn des Missionsabschnittes die Wolkendecke geschlossen war ($W_0 > 0.9$) zeigte der BLOB-Detektor höhere Leistungsindizes als die anderen Algorithmen. Ab der Sonnenaufgangsphase stieg der Leistungsindex des DPM-Detektors an. Durch den sich ändernden Elevationswinkel, und der daraus resultierenden Verdeckungssituation des Fahrzeuges durch ein Gebäude, kam bei der Messung eine zyklische Veränderung des Leistungsindex zustande. Diese konnte durch das DPM-Leistungsmodell nicht abgebildet werden. Daneben zeigte das DPM-Leistungsmodell einen leicht ansteigenden Leistungsindex, welcher durch den Sonnenstand begründet werden kann. Denn umso höher die Sonne steht, umso weniger Schattenwurf fällt von den Gebäuden auf das Fahrzeug.

Der Vergleich der Algorithmenauswahl in Bild 8.34 zeigt Differenzen zu Beginn des Missionsabschnittes, bis schließlich die Sonnenaufgangsphase einsetzte. Des Weiteren zeigte der DPM-Detektor die höchste Detektionsgenauigkeit gegenüber den anderen Algorithmen. Die einzelnen Gütemaße der eingesetzten Algorithmen präsentiert Tabelle 8.45. Dabei zeigen die Gütewerte, dass der DPM-Detektor der leistungsstärkste in diesem Missionsabschnitt war. Das Resultat der Algorithmenauswahl (siehe Tabelle 8.47) bilanzierte, dass eine Leistungsverbesserung um 2,11 % erreicht wurde und maximal eine Verbesserung um 6,52 % möglich gewesen wären. Dabei konnte eine *Genauigkeit* der Algorithmenauswahl durch die Leistungsmodelle von 95,84 % (vgl. Tabelle 8.46) ermittelt werden.



Bild 8.34: Resultierende Algorithmenauswahl in der Mission *Observierung* Abschnitt 5. Die Algorithmennummern geben die folgenden Perzeptionsmodule an: #1 BLOB, #2 CC, #3 DPM und #4 TM.

	BLOB	CC	DPM	ТМ
Messung	0,133	0,052	0,768	0,029
Modell	0,124	0,002	0,748	0,033
Differenz	-0,009	-0,050	-0,020	0,004

Tabelle 8.45: Leistungsdifferenzen zwischen Messung und Modellen der Mission Observierung Abschnitt 5.

	Algorithmenauswahl
Messung	0,818
Modell	0,784
Abweichung	-4,15 %
Genauigkeit	95,84 %

Tabelle 8.46: Auswertung der Genauigkeit anhand der Algorithmenauswahl der Mission Observierung Abschnitt 5.

	ТМ	CC	BLOB	DPM
Messung	0,029	0,052	0,133	0,768
Verbesserung AA mittels Messung in [%]	2763,17	1466,75	515,82	6,52
Verbesserung AA mittels Modells in [%]	2644,48	1401,81	490,30	2,11
Abweichung in [%]	-118,69	-64,95	-25,53	-4,42

Tabelle 8.47: Leistungssteigerung durch Algorithmenauswahl der Mission *Observierung* Abschnitt 5. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.3 Mission: Unfallszenario

Diese Mission fokussiert sich auf die folgenden Untersuchungsgegenstände:

- 1 Analyse der erzeugten Leistungsmodelle
 - 1.2 Analyse der durch ML-Verfahren erzeugten Leistungsmodelle
- 2 Evaluierung der Algorithmenauswahl
 - 2.3 Evaluierung der Algorithmenauswahl in urbanen Szenerien
- 3 Überprüfung des angepassten SPMS-Konzeptes im Flugexperiment

Daher soll der Einfluss von urbanen Szenerien auf die Fahrzeugdetektoren bestimmt werden. Dabei wurden alle Bodenbedeckungen, bis auf die Kategorie *Wasser*, einbezogen. Außerdem wurde der Einfluss eines flachen Elevationswinkels überprüft.

Im Allgemeinen zeigte das TM- und CC-Leistungsmodell einen gleichbleibenden niedrigen Leistungsindex gegenüber den anderen beiden KNN-Leistungsmodellen DPM und BLOB, wie in Bild 8.35 dargestellt. Die Messung zeigte ein ähnliches Verhalten.



Leistungsindizes der Messung

Bild 8.35: Leistungsindizes der Messung und Leistungsmodelle der Mission Unfallszenario.

Daneben waren für die DPM- und BLOB-Leistungsindizes in dieser Mission Abweichungen zwischen Messung und Modell zu beobachten. Grund war der flache Elevationswinkel, welcher in urbanen Situationen zu vermehrten partiellen oder kompletten Verdeckungen der Fahrzeuge durch Gebäude führte. Da diese Information den Leistungsmodellen nicht vorliegen kann, kam es zu Differenzen zwischen Messung und Modell.

Der DPM-Detektor zeigte am Anfang der Mission den höchsten Leistungsindex und war damit auch robust gegenüber urbanen Situationen. Gegen Ende (ca. 52. Sekunde) wirkte sich die Bodenbedeckung *Vegetation* negativ auf den DPM-Leistungsindex aus. Der Unterschied zwischen Messung und Leistungsmodell ab der 20. Sekunde resultierte aus einer Verdeckungssituation. Die Kategorien *Vegetation* und *Gebäude* beeinflussen den BLOB-Detektor negativ, während *Ebene* und *Straße* zu einer Erhöhung des Leistungsindex führten. Die Abweichung des Modells gegenüber der Messung zwischen der 21. und 28. Sekunde lag in der Sensitivität des Leistungsmodells auf die Szenenkategorie *Vegetation*. Erst ab der 28. Sekunde war diese geringer als 0,5. Ab diesem Zeitpunkt war weniger als die Hälfte des Sensorbildes durch Vegetationselemente geprägt, was zu einer Erhöhung des BLOB-Leistungsindex führte.

Resultierend aus den Leistungsindizes in Bild 8.35 ergibt sich das Ergebnis der Algorithmenauswahl in Bild 8.36. Dort war vor allem eine Differenz während der Verdeckungssituation (20. bis 28. Sekunde) zu sehen und gegen Ende der Mission (um die 49. und 56. Sekunde), da sich hier die Fahrzeuge am Bildrand befanden und damit relativ klein erscheinen.



Bild 8.36: Ergebnisse der Algorithmenauswahl in der Mission *Unfallszenario*. Die Algorithmennummern geben die folgenden Perzeptionsmodule an: #1 BLOB, #2 CC, #3 DPM und #4 TM.

Das *Gütemaß* zeigte für das DPM-Leistungsmodell die geringste Abweichung gegenüber den anderen und ist gleichzeitig der leistungsstärkste Algorithmus für diese Mission. Die größte Abweichung besaß das Leistungsmodell des BLOB-Detektors. Alle *Gütemaße* sind in der Tabelle 8.48 zu finden.

	BLOB	CC	DPM	ТМ
Messung	0,408	0,214	0,510	0,273
Modell	0,218	0,328	0,540	0,303
Differenz	-0,189	0,114	0,029	0,030
Tabelle 8 48: Leistungsdifferenzen zwischen Messung und Modellen der Mission <i>Unfallszenario</i>				

l'abelle 8.48: Leistungsdifferenzen zwischen Messung und Modellen der Mission *Unj*

In der Mission Unfallszenario konnte eine Verbesserung der Algorithmenleistung (siehe Tabelle 8.50) gegenüber dem geeignetsten Algorithmus (DPM) um 16,69 % erreicht werden. Die Messung zeigte, dass eine maximale Verbesserung von 58,65 % möglich gewesen wäre. Die Genauigkeit der Algorithmenauswahl mittels Leistungsmodellen lag bei 73,55 %, wie in Tabelle 8.49 angegeben.

	Algorithmenauswahl
Messung	0,809
Modell	0,595
Abweichung	-26,45 %
Genauigkeit	73,55 %

Tabelle 8.49: Auswertung der Genauigkeit anhand der Algorithmenauswahl der Mission Unfallszenario.

	СС	ТМ	BLOB	DPM
Messung	0,214	0,273	0,408	0,510
Verbesserung AA mittels Messung in [%]	277,68	196,78	98,54	58,65
Verbesserung AA mittels Modells in [%]	177,79	118,29	46,03	16,69
Abweichung in [%]	-99,89	-78,50	-52,51	-41,96

Tabelle 8.50: Leistungssteigerung durch Algorithmenauswahl der Mission Unfallszenario. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.4 Mission: Gebietsaufklärung I

Das Hauptaugenmerk bei dieser Mission lag auf den folgenden Untersuchungsgegenständen:

- Analyse der erzeugten Leistungsmodelle 1
 - Analyse der durch ML-Verfahren erzeugten Leistungsmodelle 1.2
- 2 Evaluierung der Algorithmenauswahl
 - 2.4 Evaluierung der Algorithmenauswahl in ländlichen Szenerien

Damit liegt das Hauptaugenmerk auf der Bodenbedeckung Ebene und Vegetation sowie dem Elevationswinkel. Untersucht wurden alle Perzeptionsketten, die entweder die Klassifizierungskaskade und den DPM-Detektor verwenden. Unberücksichtigt blieb der BLOB-Detektor, da die Mission bei Tageszeit stattfand, sowie das kantenbasierte Template Matching, da dieses während der Mission nicht durch die Algorithmenauswahl ausgewählt wurde, weder durch Messung noch durch die KNN-Leistungsmodelle.

Die Leistungsindizes dieser Mission sind in Bild 8.37 dargestellt, wobei für die einzelnen Perzeptionsketten die Kürzel aus Abschnitt 3.3.2 verwendet werden. Im Allgemeinen führte die Kategorie Vegetation zu geringeren Leistungsindizes, was bei 5, 40, 90 und 120 Sekunden nach Missionsstart zu beobachten war. Im Gegenteil dazu hatte die Kategorie Ebene einen positiven Effekt auf die Leistungsindizes, da sich gleichmäßigere Strukturen und damit auch weniger Fehlerquellen im Sensorbild befanden. Hierbei war die Perzeptionskette DPM-VG am robustesten gegenüber den beiden Kategorien der Bodenbedeckung. Ein Einfluss des steilen Elevationswinkels (< -50°) konnte nicht beobachtet werden. Generell lieferte die Perzeptionsketten mit DPM-Detektor in dieser Mission höhere Leistungsindizes als die mit Klassifizierungskaskade, da die Fahrzeuge hoch aufgelöst in den Sensordaten vorlagen und damit der DPM-Detektor seinen Vorteil ausspielen konnte.



Bild 8.37: Leistungsindizes der Messung und Leistungsmodelle der Mission Gebietsaufklärung I.

In dieser Mission waren ebenfalls Abweichungen zwischen Messung und Modelle zu sehen, wobei die im vorigen Absatz beschriebenen Effekte ebenfalls durch die Leistungsmodelle abgebildet werden konnten. Allerdings traf dies beim Abfall des Leistungsindex nach 120 Sekunden Missionszeit nicht zu. Zu begründen ist diese Abweichung damit, dass ein Fahrzeug nur partiell im Bild (siehe Bild 8.38a) zu sehen war und diese versteckte Information den Modellen nicht vorliegen konnte.



Bild 8.38: Partielle Sichtbarkeit eines Fahrzeuges und Gesteinsformation. Die Abbildung in (a) zeigt ein partiell sichtbares Fahrzeug und (b) beinhaltet die Gesteinsformation. Beide Situationen führen reduzierten Detektionsgenauigkeiten.

In Bild 8.39 ist die resultierende Algorithmenauswahl dieser Mission zu sehen. Die Abweichungen des Leistungsindex waren zunächst bei steigendem Vegetationsanteil in der Szenerie zu beobachten. Außerdem war eine Abweichung gegen Ende der Mission im Graphen erkennbar. Zurückzuführen ist diese auf eine Gesteinsformation (siehe Bild 8.38b), die zu Fehlklassifikationen bei DPM-FA und DPM-VG führten und nicht durch die KNN-Leistungsmodelle erfasst wurden.



Bild 8.39: Resultierende Algorithmenauswahl in der Mission *Gebietsaufklärung I*. Die Algorithmennummern geben die folgenden Perzeptionsketten an: #1 CC, #2 CC-FA, #3 CC-ST, #4 CC-VG, #5 DPM, #6 DPM-FA, #7 DPM-ST und #8 DPM-VG.

Die *Gütemaße* in Tabelle 8.51 zeigen für die Perzeptionsketten mit CC-Leistungsmodelle größere Differenzen als bei den Ketten mit dem DPM-Leistungsmodell. Daraus kann gefolgert werden, dass für diese Mission das DPM-Leistungsmodell genauer modelliert werden konnte. Als leistungsstärkste Perzeptionskette hat sich dabei die DPM-Perzeptionskette erwiesen. Die größte Abweichung ist bei der Perzeptionskette CC-ST festzustellen.

	СС	CC-FA	CC-ST	CC-VG	DPM	DPM-FA	DPM-ST	DPM-VG
Messung	0,508	0,759	0,817	0,522	0,942	0,928	0,891	0,941
Model	0,390	0,663	0,538	0,656	0,997	0,947	0,909	0,957
Differenz	-0,119	-0,096	-0,279	0,134	0,055	0,020	0,018	0,016

Tabelle 8.51: Leistungsdifferenzen zwischen Messung und Modellen der Mission Gebietsaufklärung I.

Tabelle 8.53 zeigt die Auswertung der Algorithmenauswahl. Hierbei zeigte sich, dass die Leistung der im Mittel geeigneten Perzeptionskette (DPM) erreicht werden konnte. Die Messung deckte auf, dass ein Plus von maximal 2,75 % für diese Mission möglich war. Die Genauigkeit der Algorithmenauswahl mittels Leistungsmodelle lag insgesamt bei 97,41 % (vgl. Tabelle 8.52).

	Algorithmenauswahl
Messung	0,967
Modell	0,942
Abweichung	-2,59 %
Genauigkeit	97,41 %

Tabelle 8.52: Auswertung der Genauigkeit anhand der Algorithmenauswahl der Mission Gebietsaufklärung I.

	CC	CC-VG	CC-FA	CC-ST	DPM-ST	DPM-FA	DPM-VG	DPM
Messung	0,508	0,522	0,759	0,817	0,891	0,928	0,941	0,942
Verbesserung AA mittels Messung in [%]	90,42	85,44	27,44	18,40	8,57	4,31	2,81	2,75
Verbesserung AA mittels Modells in [%]	85,43	80,59	24,11	15,30	5,72	1,58	0,12	0,06
Abweichung in [%]	-4,98	-4,85	-3,34	-3,10	-2,84	-2,73	-2,69	-2,69

Tabelle 8.53: Leistungssteigerung durch Algorithmenauswahl mittels KNN-Modelle der Mission *Gebietsaufklärung I*. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.5 Mission: Gebietsaufklärung II

Im Vordergrund der Evaluierung in dieser Mission standen die folgenden Untersuchungsgegenstände:

- 1 Analyse der erzeugten Leistungsmodelle
 - 1.1 Analyse der durch Expertenwissen erzeugten Leistungsmodelle
 - 1.2 Analyse der durch ML-Verfahren erzeugten Leistungsmodelle
- 2 Evaluierung der Algorithmenauswahl
 - 2.4 Evaluierung der Algorithmenauswahl in ländlichen Szenerien
- 3 Überprüfung des angepassten SPMS-Konzeptes im Flugexperiment

Untersucht wurden dabei die Effekte von Bodenpixelauflösung, Elevationswinkel, Bodenbedeckung *Ebene* und *Straße* auf alle Perzeptionsketten, die entweder einen DPM-Detektor oder eine Klassifizierungskaskade enthalten. Alle anderen Perzeptionsketten lieferten während der gesamten Mission niedrigere Leistungsindizes und wurden deshalb von der Algorithmenauswahl nicht selektiert. Daher wurden diese in dieser Evaluierung nicht weiter berücksichtig.

Zunächst wird die Mission im folgenden Abschnitt 8.3.5.1 unter Verwendung der KNN-Leistungsmodelle ausgewertet. Dieser wird anschließend einer Auswertung mittels der Experten-Leistungsmodelle gegenübergestellt.

8.3.5.1 Auswertung der Mission mit KNN-Leistungsmodellen

Die Leistungsindizes der einzelnen Perzeptionsketten sind in Bild 8.40 abgebildet, wobei zur übersichtlicheren Darstellung die Kürzel aus Abschnitt 3.3.2 verwendet wurden. Grundsätzlich hatten die Perzeptionsketten mit Farbanalyse die höchsten Leistungsindizes (CC-FA und DPM-FA). Dabei führte eine geringe Bodenpixelauflösung und ein steiler Elevationswinkel zu höheren Leistungsindizes bei allen Perzeptionsketten, wie um die 11. Sekunde zu beobachten war. Die CC-ST Perzeptionskette zeigte am Anfang der Mission den höchsten Leistungsindex. Anschließend sank dieser im Laufe der Missionszeit ab, da keine Straßensegmente mehr im



Sensorbild sichtbar waren. Die Veränderung der Kategorie *Straße* über die Missionszeit ist in Bild 7.17 dargestellt.

Bild 8.40: Leistungsindizes der Messung und KNN-Leistungsmodelle der Mission Gebietsaufklärung II.

Anfänglich zeigte der Vergleich des Leistungsindex zwischen Messung und Leistungsmodell noch einen ähnlichen Verlauf, allerdings sind ab ca. der 17. Sekunde Abweichungen zu beobachten. Zum einen waren die verwendeten Algorithmen in realen Szenen, wie zu erwarten, fehleranfälliger und produzierten damit mehr Fehl- und Falschdetektionen. Außerdem wies das Real-Bildmaterial Bewegungsunschärfen in der 22. und 26. Sekunde auf, die sich negativ auf die Erkennungsleistung der Algorithmen auswirkten.

In Bild 8.41 ist der Leistungsindex durch die Algorithmenauswahl mittels Messung und Modell abgebildet. Die stärksten Abweichungen fanden sich zu Beginn der Mission und um die 26. Sekunde herum. Zurückzuführen sind sie auf das DPM-Leistungsmodell, welches durchschnittlich höhere Leistungsindizes als das CC-Leistungsmodell lieferte. Damit wurden Perzeptionsketten mit DPM-Algorithmus fälschlicherweise öfter ausgewählt, was in Bild 8.41 im unteren Graphen erkennbar ist (z. B. Algorithmus #6, DPM-FA). Die Messung deckte auf, dass die Klassifizierungskaskade auf den realen Daten eine höhere Detektionsgenauigkeit besaß, da wahrscheinlich die Auflösungen der Fahrzeuge zu klein für den DPM-Detektor waren.

Insgesamt zeigt der Vergleich der *Gütemaße* in Tabelle 8.54, dass die Leistungsmodelle im Durchschnitt höhere mittlere Leistungsindizes lieferten als die der Messung. Den geringsten Fehler besaß dabei das CC-Leistungsmodell, während der größte Fehler mit der Perzeptionskette CC-FA einherging. Als leistungsstärkster Algorithmus für diese Mission stellte sich die DPM-FA Perzeptionskette heraus.

	СС	CC-FA	CC-ST	CC-VG	DPM	DPM-FA	DPM-ST	DPM-VG
Messung	0,548	0,574	0,598	0,549	0,695	0,709	0,592	0,690
Modell	0,584	0,812	0,731	0,733	0,611	0,821	0,748	0,745
Abweichung	0,037	0,237	0,133	0,184	-0,085	0,112	0,157	0,055
abelle 8.54: Leis	tungsdiffer	enzen zwis	chen Messi	ing und KN	N-Modelle	n der Missi	on <i>Gebietsa</i>	ufklärung I



Bild 8.41: Vergleich der Algorithmenauswahl durch KNN-Modelle in der Mission *Gebietsaufklärung II*. Die Algorithmennummern geben die folgenden Perzeptionsketten an: #1 CC, #2 CC-FA, #3 CC-ST, #4 CC-VG, #5 DPM, #6 DPM-FA, #7 DPM-ST und #8 DPM-VG.

In Tabelle 8.55 ist die Auswertung der Algorithmenauswahl dargestellt. In dieser realen Mission konnte eine Genauigkeit der Algorithmenauswahl mittels Leistungsmodelle von 88,88 % erreicht werden (siehe Tabelle 8.55). Die Bewegungsunschärfe in den Sensordaten trägt maßgeblich zu den Differenzen zwischen Messung und Modellen bei und schlägt sich in der prozentualen Verbesserung von 3,63 % gegenüber dem geeignetsten Algorithmus (DPM-FA) nieder. Maximal wäre eine Verbesserung von 16,58 % möglich gewesen, wie Tabelle 8.56 zeigt. Gegenüber der Perzeptionskette mit dem niedrigsten Leistungsindex (CC) konnte eine Verbesserung mittels Algorithmenauswahl von 34,15 % erreicht werden.

	Algorithmenauswahl
Messung	0,827
Modell	0,735
Abweichung	-11,11 %
Genauigkeit	88,88 %

Tabelle 8.55: Genauigkeit der Algorithmenauswahl mittels KNN-Modelle Mission Gebietsaufklärung II.
--

	CC	CC-VG	CC-FA	DPM-ST	CC-ST	DPM-VG	DPM	DPM-FA
Messung	0,548	0,549	0,574	0,592	0,598	0,690	0,695	0,709
Verbesserung AA mittels Messung in [%]	50,92	50,69	43,88	39,74	38,22	19,81	18,85	16,58
Verbesserung AA mittels Modells in [%]	34,15	33,95	27,89	24,21	22,86	6,50	5,65	3,62

	СС	CC-VG	CC-FA	DPM-ST	CC-ST	DPM-VG	DPM	DPM-FA
Abweichung in [%]	-16,77	-16,74	-15,99	-15,53	-15,36	-13,31	-13,21	-12,95

Tabelle 8.56: Leistungssteigerung durch Algorithmenauswahl mittels KNN-Modelle der Mission *Gebietsaufklärung II.* Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.5.2 Auswertung der Mission mit Experten-Leistungsmodellen

dieser Mission sollten alternativ zu den KNN-Leistungsmodellen für In die Klassifizierungskaskade und den DPM-Detektor auch die entsprechenden Experten-Leistungsmodelle untersucht werden. Bild 8.42 zeigt dafür die Gegenüberstellung der Leistungsindizes der Messung und der Modelle. Gegenüber der KNN-Leistungsmodelle (vgl. Bild 8.40) ist zunächst ersichtlich, dass diese den Leistungsindex in einer gröberen Auflösung liefern. Dies liegt an der verwendeten Anzahl der Zugehörigkeitsfunktionen zur Expertenmodellierung der entsprechenden FIS. Zu erkennen war weiterhin, dass wie im Abschnitt zuvor, eine geringe Bodenpixelauflösung und ein steiler Elevationswinkel zu höheren Leistungsindizes aller Perzeptionsketten führte, wie um die 11. Sekunde erkenntlich ist.



Bild 8.42: Leistungsindizes der Messung und Experten-Leistungsmodelle der Mission Gebietsaufklärung II.

Bild 8.43 zeigt den Vergleich der Algorithmenauswahl auf Basis der Messung und der Experten-Modelle. Die deutlichsten Abweichungen sind am Anfang der Mission, um die 16. Sekunde und um die 26. Sekunde aufzufinden. Die Ursachen für die Fehleinschätzung der Modelle am Anfang der Mission lagen zum einen an dem unterschiedlichen Verhalten der Bildverarbeitungsalgorithmen auf synthetischen und realen Daten und zum anderen an der unzureichenden Modellierung der Experten-Leistungsmodellen. Die Abweichungen um die 16. und 26. Sekunde waren auf die Bewegungsunschärfe in den realen Sensordaten zurückzuführen.



Bild 8.43: Vergleich der Algorithmenauswahl durch Experten-Modelle in der Mission *Gebietsaufklärung II*. Die Algorithmennummern geben die folgenden Perzeptionsketten an: #1 CC, #2 CC-FA, #3 CC-ST, #4 CC-VG, #5 DPM, #6 DPM-FA, #7 DPM-ST und #8 DPM-VG.

Um die unzureichende Modellierung der Experten-Leistungsmodellen zu beheben, wurde die Mission nachträglich analysiert und den Regelbasen des CC- und DPM-Leistungsmodell zusätzliche Regeln hinzugefügt. Die zusätzlichen Regeln des CC-Leistungsmodells finden sich in Tabelle 8.57 und wurden aus den folgenden Analysen formuliert:

- *R*₁: Bei ca. der 5,5. Sekunde Missionszeit zeigte die Messung einen geringen Leistungsindex während hauptsächlich die Bodenbedeckung *Ebene* vorherrschte und wenige Anteile *Straße* im Bild ersichtlich war.
- R_2 : Bei ca. der 12. Sekunde Missionszeit konnte ein sehr hoher Leistungsindex festgestellt werden, wobei ausschließlich die Bodenbedeckung *Ebene* präsent war.
- *R*₃: Bei ca. der 13. Sekunde Missionszeit erreichte der CC-Detektor ein sehr hoher Leistungsindex, wobei der Elevationswinkel steil und die Bodenbedeckung *Ebene* sehr hoch war.
- *R*₄: Bei ca. der 29. Sekunde Missionszeit fiel der Leistungsindex auf einen geringen Wert ab. Dabei zeigte das Sensorbild neben der Bodenbedeckung *Ebene* etwas *Vegetation*.

R_k	W _o	$oldsymbol{ heta}'$	t	tos	t_v	t_w	t_s	t _b	<i>r</i> ′	pcc
1	-			4			2	_	—	2
2	-			-			5	—	—	5
3	-	2		5			—	—	—	5
4	_	_	_	4	2	_	_	_	_	2

Tabelle 8.57: Eingebrachte Regeln für das CC-Leistungsmodell durch Analyse der Mission Gebietsaufklärung II.

Die Tabelle 8.58 zeigt die eingebrachten Regeln für das DPM-Leistungsmodell. Die Regeln wurden aus den folgenden Zusammenhängen abgeleitet:

- *R*₁: Bei ca. 5,5 Sekunden Missionszeit zeigte die Messung einen hohen Leistungsindex während hauptsächlich die Bodenbedeckung *Ebene* vorherrschte und wenige Anteile *Straße* im Bild ersichtlich war.
- *R*₂: Bei ca. 12 Sekunden Missionszeit konnte ein sehr hoher Leistungsindex festgestellt werden, wobei ausschließlich die Bodenbedeckung *Ebene* präsent war.
- *R*₃: Bei ca. 16,5 Sekunden Missionszeit erreichte der DPM-Detektor einen sehr hohen Leistungsindex. Der Anteil der Bodenbedeckung *Ebene* war gering und der Anteil der *Straße* war hoch.
- R_4 : Bei ca. 16,5 Sekunden Missionszeit erreichte der DPM-Detektor einen hohen Leistungsindex. Der Anteil der Bodenbedeckung *Ebene* und *Straße* war durchschnittlich.
- R_5 : Bei ca. 13 Sekunden Missionszeit erreichte der DPM-Detektor ein sehr hoher Leistungsindex, wobei der Elevationswinkel steil und die Bodenbedeckung *Ebene* sehr hoch war.
- R_6 : Bei ca. 14 Sekunden Missionszeit konnte der DPM-Detektor einen mittleren Leistungsindex erreichen. Dabei war der Elevationswinkel flach und die Bodenbedeckung *Ebene* sehr hoch.
- *R*₇: Bei ca. 29. Sekunden Missionszeit fiel der Leistungsindex auf einen mittleren Wert ab. Dabei zeigte das Sensorbild neben der Bodenbedeckung *Ebene* etwas *Vegetation*.

Aus diesen Regeländerungen ergaben sich nun die Leistungsindizes der verbesserten Modelle, wie in Bild 8.44 dargestellt. Der daraus resultierende Vergleich der Algorithmenauswahl mit verbesserten Modellen in Bild 8.45 zeigt gegenüber der Algorithmenauswahl mit originalen Experten-Leistungsmodellen eine leichte Verbesserung der Differenzen um die 11. und 30. Sekunde.



Bild 8.44: Leistungsindizes der Messung und verbesserten Experten-Leistungsmodelle der Mission Gebietsaufklärung II.

R _k	W _o	$oldsymbol{ heta}'$	$oldsymbol{g}'$	t _{os}	t_v	t_w	ts	t _b	<i>r</i> ′	p _{DPM}
1	_	_	_	4	_	_	2	-	_	4
2	-			—	—		5			5
3	Ι	Ι	I	2	—	Ι	4		1	5
4	-	Ι		3	—		3		1	4
5	-	2	_	5	_	_	_	-	_	5
6	_	3	_	5	_	_	_	_		3
7				4	2					3

Tabelle 8.58: Eingebrachte Regeln für das DPM-Leistungsmodell durch Analyse der Mission Gebietsaufklärung II.



Bild 8.45: Vergleich der Algorithmenauswahl durch verbesserte Experten-Modelle in der Mission *Gebietsaufklärung II.* Die Algorithmennummern geben die folgenden Perzeptionsketten an: #1 CC, #2 CC-FA, #3 CC-ST, #4 CC-VG, #5 DPM, #6 DPM-FA, #7 DPM-ST und #8 DPM-VG.

Die *Gütemaße* der Messung und der verbesserten Experten-Leistungsmodelle sind in der Tabelle 8.61 dargestellt. Hierbei zeigte sich, dass die Leistungsmodelle im Allgemeinen höhere Leistungsindizes liefern, als dies die Messung aufzeigt.

	СС	CC-FA	CC-ST	CC-VG	DPM	DPM-FA	DPM-ST	DPM-VG
Messung	0,548	0,574	0,598	0,549	0,695	0,709	0,592	0,690
Modell	0,567	0,808	0,716	0,726	0,883	0,923	0,850	0,865
Abweichung	0,020	0,234	0,118	0,177	0,187	0,214	0,259	0,176

Tabelle 8.59: Leistungsdifferenzen zwischen Messung und Experten-Modellen der Mission Gebietsaufklärung II.

Die Algorithmenauswahl mittels des verbesserten Experten-Modells konnte im Vergleich zur Messung eine Genauigkeit von 86,22 % erreichen, wie in Tabelle 8.54 ersichtlich ist. Gegenüber der leistungsstärksten Perzeptionskette (DPM-FA) konnte eine leichte Verbesserung von 0,62 % durch die Algorithmenauswahl erreicht werden. Maximal wäre eine Leistungssteigerung von 16,58 % möglich gewesen. Gegenüber der Perzeptionskette mit dem niedrigsten Gütemaß (CC) erreichte die Algorithmenauswahl eine Verbesserung von 30,26 %. Im Vergleich zu den KNN-Leistungsmodellen schnitten die Experten-Leistungsmodelle etwas schlechter ab.

	Algorithmenauswahl
Messung	0,827
Modell	0,713
Abweichung	-13,78 %
Genauigkeit	86,22 %

Tabelle 8.60: Auswertung der Genauigkeit anhand der Algorithmenauswahl mittels Experten-Modelle der Mission Gebietsaufklärung II.

	СС	CC-VG	CC-FA	CC-ST	DPM-ST	DPM-VG	DPM	DPM-FA
Messung	0,548	0,549	0,574	0,598	0,592	0,690	0,695	0,709
Verbesserung AA mittels Messung in [%]	50,92	50,69	43,88	38,22	39,74	19,81	18,85	16,58
Verbesserung AA mittels Modell in [%]	30,26	30,07	24,19	19,30	20,61	3,42	2,59	0,62
Abweichung in [%]	-20,66	-20,62	-19,69	-18,92	-19,13	-16,40	-16,27	-15,96

 Tabelle
 8.61:
 Leistungssteigerung
 durch
 Algorithmenauswahl
 mittels
 Experten-Modelle
 der
 Mission

 Gebietsaufklärung
 II.
 Die
 beiden in
 Grün markierten
 Werte
 stellen jeweils
 die
 maximale
 (links)
 und
 minimale

 (rechts)
 Verbesserung
 durch
 AA
 mittels
 Leistungsmodellen
 dar.

8.3.6 Mission: Gebietsaufklärung III

Wie in der vorherigen Mission zielt diese Mission auf die folgenden Untersuchungsgegenstände ab:

- 1 Analyse der erzeugten Leistungsmodelle
 - 1.1 Analyse der durch Expertenwissen erzeugten Leistungsmodelle
 - 1.2 Analyse der durch ML-Verfahren erzeugten Leistungsmodelle
- 2 Evaluierung der Algorithmenauswahl
 - 2.4 Evaluierung der Algorithmenauswahl in ländlichen Szenerien
- 3 Überprüfung des angepassten SPMS-Konzeptes im Flugexperiment

Evaluiert wurden der Einfluss der Bodenpixelauflösung, des Elevationswinkels und alle Bodenbedeckungen ohne die Kategorie *Wasser*. Ebenfalls wurden hier, wie bereits in der vorherigen Mission, die Perzeptionsketten mit Klassifizierungskaskade und DPM-Detektor evaluiert. Zunächst erfolgt eine Auswertung der Mission mit KNN-Leistungsmodellen und anschließend wird zum Vergleich diese mit Experten-Leistungsmodellen ausgewertet.

8.3.6.1 Auswertung der Mission mit KNN-Leistungsmodellen

Die Leistungsindizes durch Messung und KNN-Leistungsmodelle über die Missionszeit sind in Bild 8.46 dargestellt. In der Messung war zu erkennen, dass die CC und CC-VG Perzeptionsketten ein ähnliches Verhalten aufwiesen und alle übrigen Perzeptionsketten einen vergleichbaren Kurvenverlauf besaßen. Diese beiden Gruppen zeigten in der Veränderung des Leistungsindizes teilweise dieselben Effekte auf:

- Mit der 16. Sekunde stieg der Leistungsindex der Perzeptionsketten CC und CC-VG durch den steilen Elevationswinkel und der geringeren Bodenpixelauflösung.
- Gegen Ende der Mission fielen die Leistungsindizes durch den relativ hohen Anteil an der Kategorie *Straße* und *Gebäude*.
- Am Ende der Mission (68. Sekunde) stiegen die Leistungsindizes wieder durch den ansteigenden Anteil der Kategorie *Ebene*.



Zum Vergleich ist der Verlauf des Elevationswinkels und der Bodenpixelauflösung in Bild 7.19 abgebildet und die Variation der Bodenbedeckung in Bild 7.20 dargestellt.

Bild 8.46: Leistungsindizes der Messung und KNN-Leistungsmodelle der Mission Gebietsaufklärung III.

Die größte Abweichung zwischen Messung und Leistungsmodellen zeigte sich um die 34. Sekunde. Dort war ein Fahrzeug in Seitenansicht zu sehen, welches aber vom DPM-Detektor nicht erkannt wurde. Gleichzeitig generierten die Straßensegmentierung und die Farbanalyse keine Hypothese für dieses Fahrzeug.

Die Differenz der resultierenden Algorithmenauswahl in Bild 8.47 zeigt durch vorherigen Sachverhalt die höchste Abweichung um die 34. Sekunde. In den übrigen Missionsabschnitten kamen Messung und Modell, bis auf kleinere Abweichungen, zu identischen Aussagen.

Der Vergleich der Abweichungen, siehe Tabelle 8.62, durch das *Gütemaß* zeigt gegenüber dem vorherigen Flugexperiment eine geringere Einzeldifferenz der Algorithmen. Die geringste Abweichung zeigte die DPM-VG Perzeptionskette, wobei über die gesamte Missionszeit der DPM-ST Algorithmus den höchsten mittleren Leistungsindex besaß. Die größte Differenz ließ sich beim Leistungsmodell der CC-ST Perzeptionskette feststellen.

	CC	CC-FA	CC-ST	CC-VG	DPM	DPM-FA	DPM-ST	DPM-VG
Messung	0,643	0,841	0,857	0,673	0,868	0,869	0,873	0,872
Modell	0,667	0,844	0,717	0,764	0,921	0,946	0,872	0,901
Abweichung	0,023	0,003	-0,140	0,091	0,053	0,077	-0,001	0,029
Taballa 9 62. L	histungsdiff	forongon gu	vicebon Mo	coung und l	Modellon d	or Miccion	Cobiotsquff	lärung III

Tabelle 8.62: Leistungsdifferenzen zwischen Messung und Modellen der Mission Gebietsaufklärung III.



Bild 8.47: Vergleich der Algorithmenauswahl der Mission *Gebietsaufklärung III*. Die Algorithmennummern geben die folgenden Perzeptionsketten an: #1 CC, #2 CC-FA, #3 CC-ST, #4 CC-VG, #5 DPM, #6 DPM-FA, #7 DPM-ST und #8 DPM-VG.

In dieser Mission konnte durch die Algorithmenauswahl mittels Modells die Algorithmenleistung der leistungsstärksten Perzeptionskette (DPM-ST) leicht übertroffen werden, mit 1,22 % (vgl. Tabelle 8.68). Maximal wäre eine Verbesserung von 7,79 % möglich gewesen. Gegenüber der Perzeptionskette mit dem geringsten Gütemaß konnte die Algorithmenauswahl eine Verbesserung von 37,30 % erreichen. Die Genauigkeit der Algorithmenauswahl mittels KNN-Modell ist mit 92,38 % (siehe Tabelle 8.63) höher als die Genauigkeit im erste Flugexperiment (Abschnitt 8.3.5) und vergleichbar mit den Genauigkeiten der ersten beiden Mission (Abschnitt 8.3.1 und 8.3.3), welche in der synthetischen Umgebung stattfanden.

	Algorithmenauswahl
Messung	0,941
Modell	0,883
Abweichung	-6,09 %
Genauigkeit	93,84 %

Tabelle 8.63: Auswertung der Genauigkeit anhand der Algorithmenauswahl mittels KNN-Modelle der Mission Gebietsaufklärung III.

	CC	CC-VG	CC-FA	CC-ST	DPM	DPM-FA	DPM-VG	DPM-ST
Messung	0,643	0,673	0,841	0,857	0,868	0,869	0,872	0,873
Verbesserung AA mittels Messung in [%]	46,20	39,68	11,87	9,72	8,39	8,23	7,88	7,79
Verbesserung AA mittels Modells in [%]	37,30	31,18	5,06	3,04	1,79	1,64	1,31	1,22
Abweichung in [%]	-8,90	-8,51	-6,81	-6,68	-6,60	-6,59	-6,57	-6,56

Tabelle 8.64: Leistungssteigerung durch Algorithmenauswahl mittels KNN-Modelle der Mission *Gebietsaufklärung III*. Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

8.3.6.2 Auswertung der Mission mit Experten-Leistungsmodellen

Ebenso wurden in dieser Mission abermals die KNN-Leistungsmodelle für die Klassifizierungskaskade und den DPM-Detektor durch die entsprechenden Experten-Leistungsmodelle ersetzt. Daraus resultieren die Leistungsindizes der Modelle wie im unteren Graphen in Bild 8.48 dargestellt. Der obere Graph zeigt die Leistungsindizes der Perzeptionsketten ermittelt durch Messung. Wie bereits in vorheriger Mission zeigen die Experten-Leistungsmodelle eine *gröbere* Auflösung. Dadurch gehen Feinheiten der Leistungsvariation verloren. Tendenziell konnten die Experten-Leistungsmodelle die Perzeptionsketten so bewerten, dass die Reihenfolge hinsichtlich des Leistungsvermögens erhalten blieb. Die absoluten Werte der Leistungsindizes sind aber verschieden.



Bild 8.48: Leistungsindizes der Messung und Experten-Leistungsmodelle der Mission Gebietsaufklärung III.

In Bild 8.51 ist der Vergleich der Algorithmenauswahl mittels Messung (Ground Truth) und Experten-Leistungsmodellen einzusehen. Hierbei zeigen sich vor allem die Abweichungen am Beginn der Mission, um die 34. Sekunde und am Ende der Mission.

Um die Differenzen zu reduzieren, wurde die Mission nachträglich analysiert und die Regelbasen des CC- und DPM-Leistungsmodell mit zusätzlichen Regeln angereichert. Die zusätzlichen Regeln des CC-Leistungsmodells finden sich in Tabelle 8.65 und wurden aus den folgenden Analysen formuliert:

- *R*₁: Bei ca. der 3. Sekunde Missionszeit lieferte der CC-Detektor einen niedrigen Leistungsindex, wobei der Elevationswinkel flach und die Bodenbedeckung *Ebene* sehr hoch war.
- *R*₂: Bei ca. der 13. Sekunde Missionszeit konnte ein durchschnittlicher Leistungsindex ausgewertet werden. Die Bodenpixelauflösung war dabei gering.
- *R*₃: Bei ca. der 67. Sekunde Missionszeit wurde ein sehr geringer Leistungsindex festgestellt. Dabei war die Bodenbedeckung *Gebäude* sehr gering.

• *R*₄: Bei ca. der 60. Sekunde Missionszeit zeigte sich ein sehr geringer Leistungsindex, während die Bodenbedeckung *Gebäude* gering war.

R _k	W _o	θ'	<i>g</i> ′	t _{os}	t_v	t _w	t _s	t _b	<i>r</i> ′	p _{cc}
1	-	3	-	5	-	-	-	-	-	2
2	_	_	1	_	_	_	_	_	—	3
3		—	—	—	—	_	Ι	1	—	1
4	_	_	_	_	_	_	_	2	_	1

Tabelle 8.65: Eingebrachte Regeln für das CC-Leistungsmodell durch Analyse der Mission Gebietsaufklärung III.



Bild 8.49: Vergleich der Algorithmenauswahl durch Experten-Modelle in der Mission *Gebietsaufklärung III*. Die Algorithmennummern geben die folgenden Perzeptionsketten an: #1 CC, #2 CC-FA, #3 CC-ST, #4 CC-VG, #5 DPM, #6 DPM-FA, #7 DPM-ST und #8 DPM-VG.

Die Tabelle 8.66 zeigt die eingebrachten Regeln für das DPM-Leistungsmodell. Die Regeln wurden aus den folgenden Zusammenhängen abgeleitet:

- R_1 : Bei ca. 8 Sekunden Missionszeit konnte ein sehr hoher Leistungsindex ermittelt werden, wobei der Anteil der Kategorie *Ebene* sehr hoch war.
- *R*₂: Bei ca. 67 Sekunden Missionszeit erreichte der DPM-Detektor einen geringen Leistungsindex. Dabei war die Bodenbedeckung *Straße* hoch und die Kategorie *Gebäude* sehr gering.
- *R*₃: Bei ca. 60 Sekunden Missionszeit zeigte der DPM-Detektor ebenfalls einen geringen Leistungsindex, während die Bodenbedeckung *Gebäude* gering und die Anteile von *Straßen* hoch war.

R _k	W _o	$oldsymbol{ heta}'$	<i>g</i> ′	t _{os}	t_v	t _w	t _s	t _b	<i>r</i> ′	p _{DPM}
1	_		—	5		—			—	5
2	-	_	_	-	-	_	4	1	-	2
3	_	_	_	_	_	_	4	2	—	2

Tabelle 8.66: Eingebrachte Regeln für das DPM-Leistungsmodell durch Analyse der Mission Gebietsaufklärung III.

Bild 8.50 zeigt die Leistungsindizes der verbesserten Experten-Leistungsmodelle, welche sich aus den angegebenen zusätzlichen Regeln ergeben. Der daraus resultierende Vergleich der Algorithmenauswahl mit verbesserten Modellen in Bild 8.51 zeigt gegenüber der Algorithmenauswahl mit originalen Experten-Leistungsmodellen vor allem eine Verbesserung der Differenzen am Ende der Mission.



Bild 8.51: Vergleich der Algorithmenauswahl durch verbesserte Experten-Modelle in der Mission *Gebietsaufklärung III*. Die Algorithmennummern geben die folgenden Perzeptionsketten an: #1 CC, #2 CC-FA, #3 CC-ST, #4 CC-VG, #5 DPM, #6 DPM-FA, #7 DPM-ST und #8 DPM-VG.

013												
		CC	CC-FA	CC-ST	CC-VG	DPM	DPM-FA	DPM-ST	DPM-VG			
	Messung	0,643	0,841	0,857	0,673	0,868	0,869	0,873	0,872			

0,620

-0,238

Modell

Abweichung

0,475

-0,169

0,767

-0,074

Tabelle 8.67 zeigt die Gütemaße der einzelnen Perzeptionsketten. Die geringste Differenz zwischen verbesserten Experten-Modell und Messung zeigt die CC-VG Perzeptionskette. Die größte Abweichung war bei der Perzeptionskette CC-ST zu finden.

Tabelle 8.67: Leistungsdifferenzen zwischen Messung und Experten-Modellen der Mission Gebietsaufklärung III.

0,681

0,008

0,882

0,014

0,912

0,043

0,817

-0,056

0,860

-0,012

Tabelle 8.69 zeigt die Auswertung der Leistungssteigerung durch die Algorithmenauswahl. Gegenüber der leistungsstärksten Perzeptionskette (DPM) konnte eine prozentuale Verbesserung von 1,41 % erlangt werden. Maximal wäre eine Verbesserung von 7,79 % möglich gewesen. Im Vergleich zu der Perzeptionskette mit dem geringsten Leistungsindex (CC) zeigt die Algorithmenauswahl mittels Experten-Modell eine Leistungssteigerung um 37,54 %. Die Genauigkeit der Algorithmenauswahl mittels Leistungsmodellen lag insgesamt bei 94,05 % (siehe Tabelle 8.68). Im Vergleich zu den Auswertungen dieser Mission mit KNN-Leistungsmodellen, konnte durch die Verbesserung der Experten-Leistungsmodelle sogar eine Steigerung der Leistungsfähigkeit der Algorithmenauswahl erreicht werden.

	Algorithmenauswahl
Messung	0,941
Modell	0,885
Abweichung	-5,92 %
Genauigkeit	94,05 %

Tabelle 8.68: Auswertung der Genauigkeit anhand der Algorithmenauswahl mittels Experten-Modelle der Mission Gebietsaufklärung III.

	CC	CC-VG	CC-FA	CC-ST	DPM	DPM-FA	DPM-VG	DPM-ST
Messung	0,643	0,673	0,841	0,857	0,868	0,869	0,872	0,873
Verbesserung AA mittels Messung in [%]	46,20	39,68	11,87	9,72	8,39	8,23	7,88	7,79
Verbesserung AA mittels Modells in [%]	37,54	31,41	5,25	3,23	1,97	1,82	1,50	1,41
Abweichung in [%]	-8,66	-8,27	-6,62	-6,50	-6,42	-6,41	-6,39	-6,38

Tabelle 8.69: Leistungssteigerung durch Algorithmenauswahl mittels Experten-Modelle der Mission *Gebietsaufklärung III.* Die beiden in Grün markierten Werte stellen jeweils die maximale (links) und minimale (rechts) Verbesserung durch AA mittels Leistungsmodellen dar.

KAPITEL 9 DISKUSSION UND AUSBLICK

Zunächst werden in diesem Kapitel die Untersuchungsergebnisse im Kontext der Aufgabenstellung zusammengefasst und diskutiert. Dabei werden die definierten Untersuchungsgegenstände (siehe Abschnitt 7.1), welche aus den Forschungszielen (vgl. Abschnitt 1.3) abgeleitet wurden, anhand der Analysen und Ergebnissen aus dem vorherigen Kapitel diskutiert. Anschließend findet die Diskussion der erreichten Forschungsziele dieser Arbeit statt. In Abschnitt 9.2 werden schließlich dann einzelne Kritikpunkte näher beleuchtet. Hierauf wird ein Ausblick zur weiteren Verwendbarkeit des vorgestellten Ansatzes gegeben. Abschließend erfolgt eine Zusammenfassung der technisch-wissenschaftlichen Beiträge dieser Arbeit.

9.1 Zusammenfassende Diskussion der Untersuchungsergebnisse

Dieser Abschnitt diskutiert zunächst die Untersuchungsgegenstände, die aus den Forschungszielen 3 bis 5 (siehe Abschnitt 1.3) abgeleitet wurden. Anschließend werden die erreichten Forschungsziele (1 bis 5) in zusammengefasster Form besprochen.

9.1.1 Diskussion der Untersuchungsgegenstände

Die Diskussion der Untersuchungsgegenstände wird anhand der Analysen und Ergebnisse aus dem vorherigen Kapitel 8 geführt. Dabei werden diese im Folgenden einzeln betrachtet. Zunächst behandelt Abschnitt 9.1.1.1 die Analyse der erzeugten Leistungsmodelle. Anschließend wird die Evaluierung der Algorithmenauswahl in Abschnitt 9.1.1.2 diskutiert. Abschließend bespricht Abschnitt 9.1.1.3 die Überprüfung des angepassten SPMS-Konzeptes im Flugexperiment.

9.1.1.1 Analyse der erzeugten Leistungsmodelle

Der Untersuchungsgegenstand "Analyse der erzeugten Leistungsmodelle" wurde aus dem Forschungsziel 3 (vgl. Abschnitt 1.3) abgeleitet. Dieser besteht aus zwei Unterpunkten, die im Folgenden diskutiert werden, bevor schließlich ein Fazit gezogen wird.

Analyse der durch Expertenwissen erzeugten Leistungsmodelle

In Abschnitt 8.1 wurde nachgewiesen, dass die Experten-Leistungsmodelle den tatsächlichen Leistungsverlauf eines Bildverarbeitungsalgorithmus nachbilden konnten. Dabei wurde auch festgestellt, dass unter Verwendung von Vergleichsdaten die Experten-Leistungsmodelle manuell durchaus in einem Maße verbessert werden können, dass sie einen geringeren Modellfehler besitzen, als die durch ML-Verfahren erzeugten Leistungsmodelle. Beispielsweise wies ein so verbessertes Experten-Leistungsmodell einen Fehler von 0,18 (siehe Tabelle 8.9 in Abschnitt 8.1.2) auf, während für ein entsprechendes ML-Leistungsmodell ein Fehler von 1,29 (siehe Tabelle 8.9 in Abschnitt 8.1.2) festgestellt wurde. Weiterhin konnte in den Realflug-Missionen (siehe Abschnitt 8.3.5 und 8.3.6) gezeigt werden, dass der Einsatz von Expertenwissen zur Leistungsmodellierung verwendbar ist, vor allem dann, wenn keine Datenbasis für die Leistungsmodellierung in den ML-Verfahren vorhanden ist, oder diese nur schwer erzeugt werden kann. Auch hier konnte durch Anpassung der Experten-Leistungsmodelle, anhand der Analyse der Missionsdaten, eine Verbesserung der Algorithmenauswahl in den Realflug-Missionen erreicht werden. Damit konnte sogar das Potential zur Leistungssteigerung mehr ausschöpft werden, als das durch die ML-Verfahren erzeugten Leistungsmodelle möglich war. Dazu zeigt Tabelle 8.64 eine Verbesserung von 1,2 % durch Algorithmenauswahl mittels ML-Modellen bei einem theoretisch erreichbaren Potential von 7,8 % auf. Die Algorithmenauswahl durch Experten-Modelle erreichte dahingegen eine Verbesserung von 1,4 % (vgl. Tabelle 8.69).

Analyse der durch ML-Verfahren erzeugten Leistungsmodelle

Bei der Analyse der durch ML-Verfahren erzeugten Leistungsmodelle wurde zunächst die generierte Datenbasis zur Leistungsmodellierung untersucht. Hierbei wurde festgestellt, dass die Daten insgesamt eine akzeptable Gleichverteilung aufwiesen (vgl. Abschnitt 8.2.2). Somit hatten die Leistungsmodelle genügend Stützpunkte zur Funktionsapproximation zur Verfügung, um den Situationsraum geeignet zu erfassen. Allerdings hat sich bei einzelnen Umgebungszuständen gezeigt, dass diese in den Daten unterrepräsentiert waren, und dadurch eine Leistungsbestimmung ungenauer war. Dies zeigte sich z. B. anhand der Auswertung der Mission in Abschnitt 8.3.2, bei der festgestellt wurde, dass der Einfluss der Bodenbedeckung *Wasser* durch die Modelle unberücksichtigt blieb. Des Weiteren konnte in Abschnitt 8.2.3 gezeigt werden, dass die verwendeten Umgebungszustände zur Ermittlung der Algorithmenleistung überwiegend einen nichtlinearen Zusammenhang aufwiesen. Deshalb kann die Auswahl von NEFPROX und KNN als ML-Verfahren zur Leistungsmodellierung als gerechtfertigt angesehen werden, da diese hauptsächlich für nichtlineare Probleme eingesetzt werden.

In Abschnitt 8.2.4 wurden die beiden ML-Verfahren verglichen. Dabei wurde ermittelt, dass die künstlichen neuronalen Netze geeignetere Leistungsmodelle liefern als die NEFPROX-Leistungsmodelle. Deutlich wird dies an dem berechneten Modellfehler. Dort wies z. B. ein KNN-Leistungsmodell nur den halben Modellfehler (0,039, vgl. Tabelle 8.28) im Gegensatz zu einem NEFPROX-Leistungsmodell (0,069, vgl. Tabelle 8.28) auf. Des Weiteren zeigten die KNN-Leistungsmodelle, im Gegensatz zu den NEFPROX-Leistungsmodellen, bei der Analyse der Leistungsverläufe (siehe Abschnitt 8.2.4.2) stetigere Verläufe und eine detailliertere Abbildung der Algorithmenleistungen über der Variation zweier Umgebungsparameter. Aus diesen Gründen wurden die KNN-Leistungsmodelle den NEFPROX-Leistungsmodellen für die Evaluierung der Algorithmenauswahl vorgezogen.

Fazit zur Analyse der erzeugten Leistungsmodelle

Grundsätzlich konnte in den Analysen in Abschnitt 8.1 und 8.2 gezeigt werden, dass sowohl die Experten-Leistungsmodelle als auch die Leistungsmodelle, welche durch ML-Verfahren erzeugt wurden, für die Algorithmenauswahl einsetzbar waren. Die Experten-Leistungsmodelle besitzen dabei Vorteile in der einfacheren und schnelleren Modellierung. Außerdem kann aus den Erfahrungen jeder weiteren analysierten UAV-Aufklärungsmission oder durch zusätzliches Expertenwissen eine Modellverbesserung erreicht werden. Vor allem für sensordatenverarbeitende Algorithmen, die relativ einfache Methodiken verwenden,

eignet sich die Leistungsmodellierung mittels Expertenwissens. Im Gegensatz zu den ML-Verfahren benötigt sie keine aufwendige Datenerhebung. Sobald eine Leistungsbestimmung eines komplexeren Bildverarbeitungsalgorithmus (z. B. Klassifikatoren) durchgeführt werden muss, und entsprechende Trainingsdaten zur Verfügung stehen, eignen sich ML-Verfahren. Diese können außerdem detailliertere Prognosen des Leistungsverhaltens eines betrachteten Algorithmus liefern und damit auch eine potenziell bessere Algorithmenauswahl realisieren, als dies durch Experten-Modelle möglich ist.

Grundsätzlich ist das KNN-Leistungsmodell dem NEFPROX-Leistungsmodell hinsichtlich des Modellfehlers vorzuziehen. Der einzige Nachteil des KNN gegenüber den durch den NEFPROX-Algorithmus erzeugten FIS ist die problematische Interpretierbarkeit. Falls Wissen aus den Leistungsmodellen abgeleitet werden muss, spielen die NEFPROX-Leistungsmodelle ihre Stärke aus, denn dieses verwenden eine interpretierbare und nachvollziehbare Regelbasis.

Aus den vorangegangenen Analysen und Diskussionen konnte damit das **Forschungsziel 3** erreicht werden.

9.1.1.2 Evaluierung der Algorithmenauswahl

Grundsätzlich konnte in den Untersuchungen in Abschnitt 8.3 aufgezeigt werden, dass eine situationsgerechte Algorithmenauswahl immer das Potential hat, die Aufklärungsleistung zu verbessern. Selbst im Vergleich mit dem Fall, dass zufällig die leistungsstärkste Sensordatenverarbeitung ausgewählt wurde, war durch Einsatz der Algorithmenauswahl eine theoretische Verbesserung von bis zu 64 % möglich, wie die Vergleiche in Tabelle 8.44 nachweisen. In den Experimenten der durchgeführten Mission wurde anschließend gezeigt, dass die eingesetzten technischen Verfahren zur Algorithmenauswahl dieses Potential bis zu 47,64 % ausschöpfen konnten (siehe z. B. Tabelle 8.44). Dabei wurden die zugrundeliegenden Leistungsmodelle zur Algorithmenauswahl auf Basis von synthetischen Daten mittels einer Sensorsimulation erzeugt. Hierbei konnte grundsätzlich demonstriert werden, dass die Modelle sowohl in Simulationsexperimenten als auch in realen Flugversuchen einsetzbar sind, wobei die absolute Verbesserung der Systemleistung in den realen Flugexperimenten geringer ausfielen, was auch zu erwarten war. Um diesen Unterschied zwischen den realen und synthetischen Missionen zu quantifizieren, wurden jeweils die Genauigkeiten bei der Leistungsbestimmung angegeben. Erreicht wurden vergleichbare Genauigkeit zwischen simulierten, 99 % (siehe Tabelle 8.37), und realen Experimenten, 94 % (siehe Tabelle 8.69). Allerdings zeigte sich, dass die Algorithmenauswahl in den realen Flugmissionen das Potential nur wenig ausschöpfen konnten.

Evaluierung der Algorithmenauswahl hinsichtlich wechselnder Tageszeiten

Anhand der Mission *Observierung* (siehe Abschnitt 8.3.2) wurde die Algorithmenauswahl spezifisch hinsichtlich wechselnder Tageszeiten untersucht. Diese Mission hatte eine Dauer von 24 Stunden. Dabei approximierten die verwenden Modelle die tatsächliche Algorithmenleistung sehr genau über die gesamte Missionszeit. Beispielsweise wurde für einen Algorithmus die Gesamtleistung zu 0,495 (vgl. Tabelle 8.30) bestimmt, und das entsprechende Leistungsmodell besaß dazu eine Differenz von 0,042 (siehe Tabelle 8.30). Dies entspricht einer prozentualen Differenz von lediglich 8 %. Damit konnte gezeigt werden, dass die Algorithmenauswahl den Einfluss der Tageszeit korrekt berücksichtigt.

Evaluierung der Algorithmenauswahl unter atmosphärischen Einflüssen

In Abschnitt 8.3.2 konnte die Annahme aus Abschnitt 2.3.2 bestätigt werden, dass bei steigender Nebeldichte und steigendem Bewölkungsgrad die Algorithmenleistungen der betrachteten Sensordatenverarbeitungsroutinen sinken und dies die Algorithmenauswahl abbilden konnte. Eine pauschale Aussage bzgl. des Einflusses der Niederschlagsmenge auf die Algorithmenleistung konnte aus den Ergebnissen nicht abgeleitet werden. In der Mission *Observierung* (siehe Abschnitt 8.3.2) wurden die atmosphärischen Bedingungen über die Missionszeit verändert. Theoretisch wäre durch Algorithmenauswahl eine Verbesserung von 15,63 % (vgl. Tabelle 8.32) möglich gewesen. Erreicht wurden 11,17 % (vgl. Tabelle 8.32). Dies deutet an, dass die Leistungsmodellierung auch die atmosphärischen Einflüsse erfasste und damit eine Verbesserung der Gesamtleistung erreicht werden konnte. Zu sehen ist dies z. B. anhand der Leistungsverläufe in Bild 8.30. Dort steigt die Algorithmenleistung, während der Bewölkungsgrad sinkt.

Evaluierung der Algorithmenauswahl in urbanen Szenerien

Die Mission Unfallszenario (vgl. Abschnitt 8.3.3) evaluierte die Algorithmenauswahl anhand eines urbanen Szenarios. Grundsätzlich wurde festgestellt, dass die sensordatenverarbeitenden Algorithmen eine reduzierte Leistung aufwiesen, im Gegensatz zur Mission mit ländlichen Szenerien. Ursache dafür waren vor allem Verdeckungssituationen durch Gebäude, wodurch die Szene nicht vollständig eingesehen werden konnte.

Des Weiteren zeigte sich, dass in urbanen Szenerien das Potential zur Leistungssteigerung durch die Algorithmenauswahl am größten war. Das theoretische Potential betrug bis zu 58,65 % (vgl. Tabelle 8.50). Durch die eingesetzte Algorithmenauswahl konnte das Potential bis zu 16,69 % genutzt werden.

Evaluierung der Algorithmenauswahl in ländlichen Szenerien

Die Missionen in Abschnitt 8.3.4, 8.3.5 und 8.3.6 beinhalteten Aufklärungsaufgaben in ländlichen Szenerien. Hierbei konnte zunächst festgestellt werden, dass der Einfluss der Vegetation, wie z. B. Wälder, Bäume oder Büsche, zu geringeren Algorithmenleistungen führte. Dies konnte durch die Analyse der Leistungsverläufe in Bild 8.37 belegt werden. Grundsätzlich zeigte sich dabei auch, dass eine gleichmäßig strukturierte Bodenbedeckung (z. B. Wiesen oder Felder) zu höheren Algorithmenleistungen führte.

Die Ergebnisse aus Tabelle 8.53 zeigten, dass bei ländlichen Szenerien die eingesetzte Sensordatenverarbeitung im Einzelnen relativ gute Leistungswerte ohne Algorithmenauswahl aufweist. Deshalb war das Potential, welches durch Algorithmenauswahl erreicht werden konnte, relativ gering (2,75 %). Daraus kann geschlussfolgert werden, dass eine Algorithmenauswahl für ländliche Szenen zwar eine Verbesserung bringen kann, aber für urbane Szenarien das Potential deutlich höher ist.

Fazit zur Evaluierung der Algorithmenauswahl

Bei der Evaluierung der Algorithmenauswahl konnte festgestellt werden, dass diese grundsätzlich eine Verbesserung, im Gegensatz zu dem jeweilig leistungsstärksten Algorithmus, erreichen konnte. Zusätzlich konnte nachgewiesen werden, dass die Einflüsse der Tageszeit, der atmosphärischen Effekte und der topographischen Veränderungen erfasst werden konnten und damit eine entsprechende Prognose der Algorithmenleistung erzielt wurde. Aus diesen Argumentationen heraus kann auch das **Forschungsziel 5**, Evaluierung der Algorithmenauswahl anhand von repräsentativen Missionen (vgl. Abschnitt 1.3), als erreicht betrachtet werden.

9.1.1.3 Überprüfung des angepassten SPMS-Konzeptes im Flugexperiment

Dieser Abschnitt behandelt den Untersuchungsgegenstand "Überprüfung des angepassten SPMS-Konzepts im Flugversuch, welcher aus dem Forschungsziel 4 "Umsetzung der Algorithmenauswahl in ein prototypisches System" (vgl. Abschnitt 1.3) abgeleitet wurde.

Das vorgestellte angepasste SPMS-Konzept (siehe Abschnitt 6.2) wurde für die Untersuchungen prototypisch umgesetzt (vgl. Abschnitt 6.3) und in simulierten und realen UAV-Missionen erprobt. Dabei sind die Ergebnisse und deren Bewertung in Abschnitt 8.3 beschrieben. Hierbei konnte die Anwendbarkeit in der Luftfahrtdomäne für reale Anwendungen demonstriert werden, wobei das Konzept selbst nicht auf diese Domäne limitiert ist. Mittels Flugexperimenten (siehe Abschnitt 8.3.5 und 8.3.6) konnte demonstriert werden, dass auch in realen Umgebungen unter Verwendung von realen, UAV-getragenen Sensorsystemen eine Leistungsverbesserung durch die adaptive Algorithmenauswahl erreicht wird. Damit wurde gezeigt, dass das angepasste SPMS-Konzept auch im Flugexperiment erfolgreich einzusetzen ist und damit das **Forschungsziel 4** erfüllt wird.

9.1.2 Diskussion der erreichten Forschungsziele

In diesem Abschnitt werden alle Forschungsziele (vgl. Abschnitt 1.3) zusammenfassend bewertet. Dazu werden diese zunächst an dieser Stelle nochmals widergegeben:

- **Ziel 1**: Entwicklung eines Konzepts zum adaptiven luftgestützten Sensoreinsatz unter Berücksichtigung wechselnden Umgebungszustände
- Ziel 2: Implementierung verschiedener sensordatenverarbeitender Algorithmen zur luftgestützten Fahrzeugerkennung mit situativ unterschiedlichen Leistungseigenschaften
- Ziel 3: Modellierung und Analyse der Einflüsse ausgewählter Umgebungszustände auf die sensordatenverarbeitenden Algorithmen hinsichtlich deren Leistungsverhaltens
- Ziel 4: Umsetzung der Algorithmenauswahl in ein prototypisches System
- Ziel 5: Evaluierung der Algorithmenauswahl anhand von repräsentativen Missionen

Das Erreichen der **Forschungsziele 3 bis 5** wurden bereits im vorherigen Abschnitt 9.1.1 anhand der Untersuchungsgegenstände aus Abschnitt 7.1 bewertet, und es konnte gezeigt werden, dass diese durch die durchgeführten Experimente und der daraus resultierenden Ergebnisse erfüllt wurden.

Das **Forschungsziel 1** befasst sich mit der Entwicklung des Konzepts zum adaptiven luftgestützten Sensoreinsatz. In Abschnitt 3.2 wurde dazu ein verallgemeinertes Konzept aufgestellt, welches in Abschnitt 3.4 detailliert wurde. Dabei wurden in Abschnitt 3.1 Anforderungen aufgestellt, die eine Modularisierung der Bildverarbeitungsalgorithmen forderten, um eine Erweiterbarkeit durch weitere Algorithmen zu gewährlisten. Daher ermöglicht das vorgestellte Konzept eine Erweiterung der Algorithmenauswahl mit neuen Bildverarbeitungsalgorithmen ohne zusätzliche Änderung der bereits existierenden Leistungsmodelle. Dies birgt den Vorteil, dass die Leistungsmodellierung für neue Algorithmen unabhängig von den bereits bestehenden erfolgen kann. Problematisch kann dabei allerdings eine Wechselwirkung des Leistungsverhaltens zwischen zwei Algorithmen sein. Zusätzlich,

wie durch Anforderung 8 aufgestellt, wurden die verwendeten Umgebungszustände zur Leistungsbestimmung so gewählt, dass sie unabhängig von der eingesetzten Sensorik erfasst werden können. Damit kann das Konzept auch für weitere Sensortypen verwendet werden. In diesem Sinne kann das **Forschungsziel 1** als erfüllt betrachtet werden.

Die Implementierung verschiedener sensordatenverarbeitender Algorithmen zur luftgestützten Fahrzeugerkennung wurde im **Forschungsziel 2** festgelegt. Dazu wurden zunächst in Abschnitt 3.3.2 geeignete Bildverarbeitungsverfahren identifiziert und deren konkrete Umsetzung in Abschnitt 4.5 beschrieben. In den Experimenten (vgl. Abschnitt 8.3) wurde gezeigt, dass durch die Bestimmung der Algorithmenleistung, anhand des Umgebungszustandes, die spezifischen Schwachstellen der einzelnen eingesetzten Bildverarbeitungsalgorithmen identifiziert werden konnten. Damit ist theoretisch eine Verbesserung der zugrundeliegenden Algorithmen möglich. Auch hier wurde das Forschungsziel erreicht.

9.2 Diskussion von Kritikpunkten

Obwohl die Forschungsziele alle erreicht wurden, bergen die verwendeten Verfahren und Methoden Kritikpunkte, die an dieser Stelle dargelegt und in ihren Auswirkungen auf die Ergebnisse diskutiert und mögliche Lösungen in zukünftigen Arbeiten aufgezeigt werden sollen.

9.2.1 Unvollständige Erfassung der Umgebungszustände

Durch die verwendeten Umgebungszustände in dieser Arbeit bleiben sensorspezifische Effekte unbeobachtet. Als Beispiel hierfür wäre der Effekt der Bewegungsunschärfe (siehe Abschnitt 8.3.5) zu nennen, der während den Realflug-Experimenten auftrat. Er beeinflusste die Algorithmenleistung negativ, was jedoch in den eingesetzten Leistungsmodellen nicht abgebildet werden konnte. Mögliche Lösungsansätze, um spezifische Sensoreffekte zu berücksichtigen, ohne dabei Bildbeschreiber zu verwenden, wären die Erfassung dieser über physikalische Gegebenheiten. Bei einem VIS-Sensor könnte die Bewegungsunschärfe über die Bodengeschwindigkeit des Sensorpfades und der Sensoreigenschaften (z. B. FOV) berechnet werden. Generell ist zu empfehlen, dass die Qualität der Sensordaten ermittelt und bei der Berechnung der Systemleistung berücksichtigt wird.

Im Gegensatz zu der verwendeten Umgebungserfassung nutzen andere Verfahren *Bildmerkmale* bzw. Merkmale, die aus den Sensordaten extrahiert werden. In [27] werden Fuzzy-Inferenzsysteme durch maschinelles Lernen erzeugt, die auf Basis der *Bildmerkmale* eine Algorithmenauswahl vornehmen. Des Weiteren wird in [28] eine Methode präsentiert, die ebenfalls *Bildmerkmale* verwendet. Dabei werden für einen Algorithmus jene *Bildmerkmale* in einer Datenbank gespeichert, wo diese sich am leistungsstärksten, relativ zu den anderen Algorithmen, zeigen. Mittels einer Vergleichsmetrik wird auf Basis der *Bildmerkmale* des aktuellen Sensorbildes ein Algorithmus ausgewählt. Im Allgemeinen haben *Bildmerkmale* den Vorteil, dass sie spezifische Sensoreffekte erfassen können und damit auch auf diese in geeigneter Weise reagiert werden kann.

9.2.2 Defizite in der Leistungsmodellierung durch ML-Verfahren

Bei der Analyse der Leistungsmodelle in Abschnitt 8.3 wurde festgestellt, dass die generierten Daten zur Leistungsmodellierung durch ML-Verfahren nicht vollständig gleichverteilt vorlagen (siehe Abschnitt 8.2.2) und es dadurch zu Abweichungen bei der Leistungsbestimmung kam. Dies lässt sich darauf zurückführen, dass eine gleichverteilte Abbildung des Parameterraums, selbst mittels synthetischer Datengenerierung, extrem aufwendig bzw. nicht möglich war. Daher wurde im vorliegenden Fall zur Generierung des Modelldatensatzes ein stochastischer Prozess angewendet. Dieser tastete den betrachteten Parameterraum zufällig ab. Für die weitere Verbesserung der Generierung des Modelldatensatzes könnten Methoden aus dem Forschungsbereich *Metalearning* [108] eingesetzt werden, um die Varianz des Modelldatensatzes zu erhöhen. Typischerweise werden hierbei vor allem statistische Merkmale der Datensätze erhoben, um ein holistisches Modell zu erzeugen, welches ein Ranking der Algorithmen vornimmt. Es werden dabei zwar keine umgebungsbeschreibenden Merkmale eingesetzt, aber das Verfahren könnte um diese erweitert werden. Damit weist dieses Verfahren einen alternativen Weg auf, um die generierten Stützstellen im Parameterraum zu extrapolieren.

9.2.3 Einschränkungen in der Erweiterbarkeit

In dieser Arbeit wurden die einzelnen Bildverarbeitungsalgorithmen unabhängig voneinander betrachtet, indem separate Leistungsmodelle erzeugt wurden (siehe Kapitel 5). Dadurch ist eine einfache Funktionserweiterung möglich. Allerdings führt dies dazu, dass die Beeinflussung zwischen den einzelnen sensordatenverarbeitenden Algorithmen unberücksichtigt bleibt. Ein Beispiel dafür ist ein Algorithmus zur Fahrzeugdetektion, welcher auf bewaldetem Gebiet bekanntermaßen Fehldetektionen erzeugt. Um diese auszuschließen, wurde eine vorgeschaltete Segmentierung eingebracht, die bewaldete Gebiet vom Suchraum des Fahrzeugdetektors exkludierte. Damit konnte indirekt die Leistung der Fahrzeugdetektion durch Minimierung der Fehldetektionen erhöht werden. Würde eine solche Abhängigkeit berücksichtigt werden, müsste bei der Leistungsbestimmung der Fahrzeugdetektion diese spezielle Segmentierungsleistung miteinbezogen werden. In diesem Fall würde dies dazu führen, dass bei einer Funktionserweiterung einige Modelle zur Leistungsbestimmung neu erzeugt werden müssten. Lösen ließe sich dieses Problem durch eine Gesamtbetrachtung aller verwendeten sensordatenverarbeitenden Algorithmen, allerdings mit dem Nachteil der aufwendigeren Erweiterbarkeit. Ein Kompromiss könnte die Leistungsmodellierung auf Perzeptionskettenebene sein.

9.2.4 Nachteile durch Verwendung synthetischer Daten

Implizit wurde durch die Experimente die Algorithmenauswahl anhand von synthetischen und realen Flugexperimenten evaluiert (vgl. Abschnitt 7.4). Dabei kam die Diskussion in Abschnitt 9.1.1.2 zu dem Ergebnis, dass die Genauigkeit der Leistungsprädiktion durch die verwendeten Modelle, welche auf Basis von synthetischen Daten erzeugt wurden, vergleichbar war. Es wurden dabei Genauigkeiten von bis zu 99 % (siehe Tabelle 8.37) in den simulierten und bis zu 94 % (siehe Tabelle 8.69) in den realen Missionen erreicht. Dieses Ergebnis unterstützt zunächst die Aussagen aus [121], dass durch synthetische Bilddaten erzeugte Modelle durchaus sinnvoll auf reale Sensordaten anwendbar sind.

Grundsätzlich konnte festgestellt werden, dass durch die Verwendung synthetischer Daten die Algorithmenauswahl in realen Missionen einsetzbar war. Bei kritischer Betrachtung zeigt sich aber, dass das Verbesserungspotential in den realen Missionen nur wenig ausgenutzt werden konnte, da merkliche Unterschiede bei der Modellgüte zu verzeichnen waren. Beispielweise zeigt die Auswertung in Tabelle 8.56, dass ein theoretisches Potential von 16,48 % möglich war, aber lediglich 3,62 % ausgeschöpft wurden. Weiterhin konnte in einer zweiten Realflug-Mission eine ähnliche Situation erfasst werden. Dazu zeigt Tabelle 8.64, dass ein Potential von 7,79 % möglich war und nur 1,22% erreicht wurden. Vermutlich sind diese Ergebnisse auf die Verwendung von synthetischen Daten zurückzuführen, was aber noch überprüft werden müsste.

Kann die Ursache auf die synthetischen Daten zurückgeführt werden, wäre eine weitere Erhöhung des genutzten Potentials durch die Methoden aus dem Bereich *Domänenadaption* (engl. *domain adaptation*) [151] denkbar. Ziel dabei ist es, mit Fokus auf Objektdetektoren, ein existierendes Modell zur Objektdetektion auf eine neue Domäne zu übertragen bzw. für diese zu adaptieren. Damit könnten die auf synthetischen Daten trainierten Objektdetektoren für die realen Sensordatenspezifika angepasst werden.

9.3 Ausblick

Mit dem in dieser Arbeit vorgestellten Konzept zum adaptiven Sensoreinsatz können sensorbasierte Systeme realisiert werden, die bei Änderung der Umgebung ihre internen Prozessabläufe anpassen, um die Systemleistung aufrechtzuerhalten oder zu verbessern. Außerdem steht durch die Ermittlung der Algorithmenleistung ein Kriterium zur Verfügung, welches von anderen Systemen verwendet kann.

Im Umfeld der luftgestützten Aufklärungssysteme könnte die Beurteilung von Algorithmenleistungen herangezogen werden, um zu entscheiden, ob eine perzeptive Aufgabe automatisiert von einem Perzeptionssystem oder nach wie vor besser durch einen menschlichen Operateur durchgeführt werden sollte. Dabei sind auch intermediäre Zustände denkbar, wobei ein Teil der Aufgabe durch den Operateur und ein anderer durch die Automation durchgeführt wird. Diese Aufgabentransition zwischen Maschine und Menschen lässt sich durch Änderung des *Automationsgrads* (engl. *levels of automation*) [152,153] beschreiben. Untersucht wird diese z. B. im Bereich MUM-T (engl. *manned-unmanned teaming*) von Ruf et al. in [154] und [155].

Auch für Objekttrackinganwendungen lässt sich die Leistung der Objektdetektoren als Vertrauenswert für die Schätzung der Trajektorie des verfolgten Objekts berücksichtigen. So wird in [156] eine mathematische Methode zum Tracking von mehreren Objekten am Boden mittels UAVs vorgestellt. Das UAV ist dabei mit einem Gimbal ausgerüstet, welcher einen Sensor trägt. Dabei wird die Evidenz für die Detektion eines Objektes lediglich durch gaußsches Rauschen modelliert. Hier könnte die Algorithmenleistung herangezogen werden, um diese Methode für reale Systeme realistischer umsetzen zu können.

Des Weiteren ist vorstellbar, dass in Multi-UAV Anwendungen mit unterschiedlich konfigurierten UAVs eine Aufteilung der perzeptiven Aufgaben auf die verschiedenen Plattformen anhand der einzelnen Algorithmenleistungen realisiert werden kann. Diese UAVs können sich dabei hinsichtlich der Sensorik und der Sensordatenverarbeitung unterscheiden. Bei einem kooperativen Einsatz von mehreren UAVs für eine gemeinsame Aufgabe können die Subaufgaben durch Bestimmung der Systemleistung entsprechend auf die UAVs aufgeteilt werden. Ein Beispiel dafür gibt die Multi-UAV Ladezonenaufklärung für einen bemannten Helikopter [157], wobei die Algorithmenleistung zusätzlich in die Bewertung der Landepunkte einfließen soll.

Weiterhin können aus einem solchen Leistungsmodell die Randbedingungen für einen flugtechnischen Einsatz eines UAVs bei der Durchführung einer Perzeptionsaufgabe ermittelt werden. Die zugrundeliegenden Modelle bestimmen u. a. durch systemisch veränderliche Parameter (Sensorelevationswinkel oder Bodenpixelauflösung) die Einzelleistungen der verwendeten Algorithmen. Durch numerische Optimierung lassen sich die systemischen Parameter bei fixen Umgebungszuständen so bestimmen, dass die Algorithmenleistung maximiert wird. Zum einen lässt sich bei der Missionsplanung die Flugtrajektorie optimal für die sensordatenverarbeitenden Algorithmen anpassen. Zum anderen, wenn sich Umgebungszustände ändern, kann die Fluglage und -position während eines Auftrages an die Anforderungen des Sensorsystems angepasst werden. Damit kann eine Verbindung zwischen Algorithmenauswahl und Sensorauswahl realisiert werden, was zu einer Verknüpfung des angepassten SPMS in Abschnitt 6.1 mit dem Sensormanagementkonzept aus [23] führt.

9.4 Technisch-wissenschaftliche Beiträge

Die technisch-wissenschaftlichen Beiträge dieser Arbeiten werden in diesem Abschnitt abschließend zusammengefasst. Ziel der Arbeit war die Entwicklung eines Konzepts zum adaptiven Sensoreinsatz an Board von UAVs und dessen Umsetzung mittels geeigneter Methoden. Folgende innovativen Beiträge wurden dabei geleistet und jeweils in Veröffentlichungen dem Fachpublikum vorgestellt:

- Eine Modellierung des Leistungsverhaltens von sensordatenverarbeitenden Algorithmen unter wechselnden atmosphärischen Bedingungen wurde vorgestellt [158]. Hierbei wurde im Gegensatz zu anderen Verfahren Expertenwissen herangezogen und dieses in Fuzzy-Inferenzsysteme übertragen.
- Die Leistungsmodellierung von Fahrzeugdetektoren mittels Methoden des maschinellen Lernens zur Algorithmenauswahl wurde realisiert. Während der Stand der Technik einzelne Einflussgrößen untersuchte, wurde hier die Leistung anhand von atmosphärischen, photogrammetrischen und topographischen Einflussgrößen gemessen. Eingesetzt wurden Fuzzy-Inferenzsysteme [112] und künstliche neuronale Netze.
- Eine erstmalige Umsetzung einer Algorithmenauswahl zur Leistungssteigerung der Sensordatenverarbeitung an Bord von UAVs und deren Evaluierung in simulierten und realen Flugversuchen wurde gezeigt [159].
- Die erstmalige Untersuchung des Einflusses von gleichzeitig mehreren Umgebungszuständen auf Bildverarbeitungsalgorithmen zur luftgestützten Fahrzeugerkennung wurde präsentiert [159].

Des Weiteren liefert diese Arbeit einen technischen Beitrag zur weiteren Ausgestaltung des SPM-Paradigmas [30,160], indem sie die Auswahl der sensordatenverarbeitenden Algorithmen auf Basis der Umgebungszustände in dieses integrierte. Schließlich konnte ein wissenschaftlicher Beitrag zum Einsatz von Simulationsumgebungen geleistet werden. Dabei wurde gezeigt, dass synthetisch erzeugte Objektklassifikatoren und Leistungsmodelle auch in UAV-Anwendungen auf Missionsebene unter realen Bedingungen eingesetzt werden können.

Literaturverzeichnis

- I. Colomina, P. Molina, Unmanned aerial systems for photogrammetry and remote sensing: A review, ISPRS J. Photogramm. Remote Sens. 92 (2014) 79–97. doi:10.1016/j.isprsjprs.2014.02.013.
- [2] K. Kanistras, G. Martins, M.J. Rutherford, K.P. Valavanis, A survey of unmanned aerial vehicles (UAVs) for traffic monitoring, in: 2013 Int. Conf. Unmanned Aircr. Syst., IEEE, Atlanta, GA, 2013: pp. 221–234. doi:10.1109/ICUAS.2013.6564694.
- [3] K. Whitehead, C.H. Hugenholtz, S. Myshak, O. Brown, A. LeClair, A. Tamminga, T.E. Barchyn, B. Moorman, B. Eaton, Remote sensing of the environment with small unmanned aircraft systems (UASs), part 2: scientific and commercial applications, J. Unmanned Veh. Syst. 02 (2014) 86–102. doi:10.1139/juvs-2014-0007.
- [4] M. Erdelj, E. Natalizio, K.R. Chowdhury, I.F. Akyildiz, Help from the Sky: Leveraging UAVs for Disaster Management, IEEE Pervasive Comput. 16 (2017) 24–32. doi:10.1109/MPRV.2017.11.
- [5] M.B. Bejiga, A. Zeggada, A. Nouffidj, F. Melgani, A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery, Remote Sens. 9 (2017). doi:10.3390/rs9020100.
- [6] S. Candiago, F. Remondino, M. De Giglio, M. Dubbini, M. Gattelli, Evaluating multispectral images and vegetation indices for precision farming applications from UAV images, Remote Sens. 7 (2015) 4026–4047. doi:10.3390/rs70404026.
- K. Máthé, L. Buşoniu, Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection, Sensors (Switzerland). 15 (2015) 14887–14916. doi:10.3390/s150714887.
- [8] M.L. Cummings, S. Bruni, S. Mercier, P.J. Mitchell, Automation architecture for single operator, multiple UAV command and control, Int. C2 J. 1 (2007) 1–24. doi:10.1017/CB09781107415324.004.
- [9] X. Luo, Y. Wu, Y. Huang, J. Zhang, Vehicle flow detection in real-time airborne traffic surveillance system, Trans. Inst. Meas. Control. 33 (2011) 880–897. doi:10.1177/0142331209103042.
- [10] J.R. Martínez-de-Dios, L. Merino, A. Ollero, L.M. Ribeiro, X. Viegas, Multi-UAV Experiments: Applications to Forest Fires, Mult. Heterog. Unmanned Aer. Veh. 37 (2007) 207–228. doi:10.1007/978-3-540-73958-6.
- [11] P. Rudol, P. Doherty, Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery, 2008 IEEE Aerosp. Conf. (2008) 1– 8. doi:10.1109/AERO.2008.4526559.
- [12] J.C. Hodgson, S.M. Baylis, R. Mott, A. Herrod, R.H. Clarke, Precision wildlife monitoring using unmanned aerial vehicles, Sci. Rep. 6 (2016) 1–7. doi:10.1038/srep22574.
- S. Leva, M. Aghaei, F. Grimaccia, PV power plant inspection by UAS: Correlation between altitude and detection of defects on PV modules, 2015 IEEE 15th Int. Conf. Environ. Electr. Eng. EEEIC 2015 Conf. Proc. (2015) 1921–1926. doi:10.1109/EEEIC.2015.7165466.
- [14] A. Schulte, Kognitive und kooperative Automation zur Führung unbemannter Luftfahrzeuge, in: 2. Interdiszip. Work. Kognitive Syst., Duisburg, 2012.

- [15] G. Cai, J. Dias, L. Seneviratne, A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends, Unmanned Syst. 02 (2014) 175–199. doi:10.1142/S2301385014300017.
- [16] J.M. Peschel, R.R. Murphy, On the human-machine interaction of unmanned aerial system mission specialists, IEEE Trans. Human-Machine Syst. 43 (2013) 53–62. doi:10.1109/TSMCC.2012.2220133.
- [17] A.P. Tvaryanas, Human systems integration in remotely piloted aircraft operations, Aviat. Sp. Environ. Med. 77 (2006) 1278–1282.
- [18] N. Theissing, A. Schulte, Designing a Support System to Mitigate Pilot Error While Minimizing Out-of-the-Loop-Effects, in: D. Harris (Ed.), Eng. Psychol. Cogn. Ergon. - 13th Int. Conf. EPCE 2016, Held as Part HCI Int. 2016, Springer International Publishing, Toronto, ON, Canada, 2016: pp. 439–451. doi:10.1007/978-3-319-40030-3_43.
- [19] R. Strenzke, J. Uhrmann, A. Benzler, F. Maiwald, A. Rauschert, A. Schulte, Managing Cockpit Crew Excess Task Load in Military Manned-Unmanned Teaming Missions by Dual-Mode Cognitive Automation Approaches, in: AIAA Guid. Navig. Control Conf., 2011. doi:10.2514/6.2011-6237.
- [20] J. Uhrmann, A. Schulte, Concept, Design and Evaluation of Cognitive Task-based UAV Guidance, Int. J. Adv. Intell. Syst. 5 (2012) 145–158.
- [21] J. Gleason, A. Nefian, X. Bouyssounousse, T. Fong, G. Bebis, Vehicle detection from aerial imagery, IEEE Int. Conf. Robot. Autom. (2011) 2065–2070.
- [22] D. Hoiem, A.A. Efros, M. Hebert, D. Hoiem, A.A. Efros, · M Hebert, M. Hebert, Putting Objects in Perspective, Int J Comput Vis. 80 (2008) 3–15. doi:10.1007/s11263-008-0137-5.
- [23] A.O. Hero, D. Cochran, Sensor Management: Past, Present, and Future, IEEE Sens. J. 11 (2011) 3064–3075. doi:10.1109/JSEN.2011.2167964.
- [24] G. Cheng, J. Han, A survey on object detection in optical remote sensing images., ISPRS J. Photogramm. Remote Sens. 117 (2016) 11–28. doi:10.1016/j.isprsjprs.2016.03.014.
- [25] T. van Valkenburg- van Haarst, W. van Norden, F. Bolderheij, Automatic Sensor Management: challenges and solutions, Proc. SPIE. 6945 (2008). doi:10.1117/12.777607.
- [26] J.R. Rice, The Algorithm Selection Problem, Adv. Comput. 15 (1976) 65–118. doi:10.1016/S0065-2458(08)60520-3.
- [27] J. Cepeda-Negrete, R.E. Sanchez-Yanez, Automatic selection of color constancy algorithms for dark image enhancement by fuzzy rule-based reasoning, Appl. Soft Comput. J. 28 (2015) 1–10. doi:10.1016/j.asoc.2014.11.034.
- [28] N. Hochgeschwender, M.A. Olivares-Mendez, H. Voos, G.K. Kraetzschmar, Context-based selection and execution of robot perception graphs, IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA. (2015) 1–4. doi:10.1109/ETFA.2015.7301631.
- [29] R. Feris, R. Bobbitt, S. Pankanti, I.B.M.T.J. Watson, N. York, Efficient 24 / 7 Object Detection in Surveillance Videos, 12th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS), 2015. (2015) 1–6.
- [30] M. Russ, P. Stuetz, Application of a probabilistic market-based approach in UAV sensor & perception management, in: Inf. Fusion (FUSION 2013), 2013 16th Int. Conf., Istanbul,
2013: pp. 676-683.

- [31] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, B. Schiele, Vision based victim detection from unmanned aerial vehicles, 2010 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2010) 1740–1747. doi:10.1109/IROS.2010.5649223.
- [32] K. Kanistras, G. Martins, M.J. Rutherford, K.P. Valvanis, A Survey of Unmanned Aerial Vehicles for Traffic Surveillance, in: K.P. Valavanis, G.J. Vachtsevanos (Eds.), Handb. Unmanned Aer. Veh., Springer Netherlands, 2015: pp. 2643–2666. doi:10.1007/978-90-481-9707-1_122.
- [33] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, B. Rinner, Networked UAVs as aerial sensor network for disaster management applications, Elektrotechnik Und Informationstechnik. 127 (2010) 56–63. doi:10.1007/s00502-010-0717-2.
- [34] L. Merino, F. Caballero, J.R. Martinez-de-Dios, I. Maza, A. Ollero, An Unmanned Aerial System for Automatic Forest Fire Monitoring and Measurement, J. Intell. Robot. Syst. 65 (2011) 533–548.
- [35] A. Gaszczak, T.P. Breckon, J. Han, Real-time people and vehicle detection from UAV imagery, Proc. Int. Soc. Opt. Eng. 7878 (2011). doi:10.1117/12.876663.
- [36] R. Gade, T.B. Moeslund, Thermal cameras and applications: A survey, Mach. Vis. Appl. 25 (2014) 245–262. doi:10.1007/s00138-013-0570-5.
- [37] J.A.J. Berni, P.J. Zarco-Tejada, L. Suárez, E. Fereres, L. Suarez, E. Fereres, Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle, IEEE Trans. Geosci. Remote Sens. 47 (2009) 722–738. doi:10.1109/TGRS.2008.2010457.
- [38] H. Aasen, A. Burkart, A. Bolten, G. Bareth, Generating 3D hyperspectral information with lightweight UAV snapshot cameras for vegetation monitoring: From camera calibration to quality assurance, ISPRS J. Photogramm. Remote Sens. 108 (2015) 245–259. doi:10.1016/j.isprsjprs.2015.08.002.
- [39] S. Scherer, S. Singh, L. Chamberlain, S. Saripalli, Flying fast and low among obstacles, Proc. - IEEE Int. Conf. Robot. Autom. (2007) 2023–2029. doi:10.1109/ROBOT.2007.363619.
- [40] A. Moreira, P. Prats, M. Younis, G. Krieger, I. Hajnsek, K. Papathanassiou, A Tutorial on Synthetic Aperture Radar, IEEE Geosci. Remote Sens. Mag. (2013) 1–43. doi:10.1109/MGRS.2013.2248301.
- [41] K. Ouchi, Recent trend and advance of synthetic aperture radar with selected topics, Remote Sens. 5 (2013) 716–807. doi:10.3390/rs5020716.
- [42] D. Andrea, S. Lupashin, A. Sch, M. Sherback, R.D. Andrea, UAV photogrammetrie, ETH Zürich, 2009. doi:10.3929/ethz-a-010782581.
- [43] X. Niu, A semi-automatic framework for highway extraction and vehicle detection based on a geometric deformable model, ISPRS J. Photogramm. Remote Sens. 61 (2006) 170– 186. doi:10.1016/j.isprsjprs.2006.08.004.
- [44] B. L. Saux, M. Sanfourche, Robust vehicle categorization from aerial images by 3Dtemplate matching and multiple classifier system, Image Signal Process. Anal. (ISPA), 2011 7th Int. Symp. (2011) 466–470.

- [45] C. Xu, H. Duan, Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft, Pattern Recognit. Lett. 31 (2010) 1759–1772. doi:10.1016/j.patrec.2009.11.018.
- [46] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, in: Found. Fuzzy Log. Soft Comput., Springer Berlin Heidelberg, 2007: pp. 789--798. doi:10.1007/978-3-540-72950-1.
- [47] J. Choi, Y. Yang, Vehicle detection from aerial images using local shape information, Adv. Image Video Technol. 5414 (2009) 227–236.
- [48] S. Hinz, U. Stilla, Car Detection in Aerial Thermal Imagery by Local and Global Evidence Accumulation, Pattern Recognit. Lett. 27 (2006) 308--315.
- [49] H. Moon, R. Chellappa, A. Rosenfeld, Performance analysis of a simple vehicle detection algorithm, Image Vis. Comput. 20 (2002) 1–13. doi:10.1016/S0262-8856(01)00059-2.
- [50] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, CVPR. 1 (2001) 511–518.
- [51] Y. Xu, G. Yu, Y. Wang, X. Wu, Y. Ma, A hybrid vehicle detection method based on violajones and HOG + SVM from UAV images, Sensors (Switzerland). 16 (2016). doi:10.3390/s16081325.
- [52] Y. Xu, G. Yu, X. Wu, Y. Wang, Y. Ma, An Enhanced Viola–Jones Vehicle Detection Method From Unmanned Aerial Vehicles Imagery, IEEE Trans. Intell. Transp. Syst. 18 (2017) 1– 12. doi:10.1109/TITS.2016.2617202.
- [53] H. Grabner, T.T. Nguyen, B. Gruber, H. Bischof, On-line boosting-based car detection from aerial images, ISPRS J. Photogramm. Remote Sens. 63 (2008) 382–396. doi:10.1016/j.isprsjprs.2007.10.005.
- [54] G. Hummel, D. Smirnov, A. Kronenberg, P. Stuetz, Prototyping and training of computer vision algorithms in a synthetic UAV mission test bed, in: AIAA SciTech 2014, American Institute of Aeronautics and Astronautics, National Harbour, 2014: pp. 1–10. doi:10.2514/6.2014-1322.
- [55] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 1 (2005) 886–893. doi:10.1109/CVPR.2005.177.
- [56] S. Tuermer, F. Kurz, P. Reinartz, U. Stilla, Airborne vehicle detection in dense urban areas using HoG features and disparity maps, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 6 (2013) 2327–2337. doi:10.1109/JSTARS.2013.2242846.
- [57] T. Moranduzzo, F. Melgani, Automatic car counting method for unmanned aerial vehicle images, IEEE Trans. Geosci. Remote Sens. 52 (2014) 1635–1647. doi:10.1109/TGRS.2013.2253108.
- [58] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, Computer (Long. Beach. Calif). 32 (2010) 1627–1645. doi:10.1109/TPAMI.2009.167.
- [59] G. Cheng, J. Han, L. Guo, X. Qian, P. Zhou, X. Yao, X. Hu, Object detection in remote sensing imagery using a discriminatively trained mixture model, ISPRS J. Photogramm. Remote Sens. 85 (2013) 32–43. doi:10.1016/j.isprsjprs.2013.08.001.
- [60] H.T. Niknejad, S. Mita, D. McAllester, T. Naito, Vision-based vehicle detection for

nighttime with discriminately trained mixture of weighted deformable part models, IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC. (2011) 1560–1565. doi:10.1109/ITSC.2011.6082826.

- [61] J.X. Tang, C.W. Deng, G.B. Huang, B.J. Zhao, Compressed-Domain Ship Detection on Spaceborne Optical Image Using Deep Neural Network and Extreme Learning Machine, Ieee Trans. Geosci. Remote Sens. 53 (2015) 1174–1185. doi:10.1109/tgrs.2014.2335751.
- [62] J. Han, D. Zhang, G. Cheng, L. Guo, J. Ren, Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning, IEEE Trans. Geosci. Remote Sens. 53 (2015) 3325–3337. doi:10.1109/TGRS.2014.2374218.
- [63] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, M. Zuair, Deep Learning Approach for Car Detection in UAV Imagery, Remote Sens. 9 (2017) 312. doi:10.3390/rs9040312.
- [64] J. Albertz, Einführung in die Fernerkundung: Grundlagen der Interpretation von Luftund Satellitenbildern, 4th ed., WBG (Wissenschaftliche Buchgesellschaft), Darmstadt, 2009.
- [65] V. Reilly, B. Solmaz, M. Shah, Geometric constraints for human detection in aerial imagery, Eur. Conf. Comput. Vis. (2010) 252--265. doi:10.1007/978-3-642-15567-3.
- [66] Hsu-Yung Cheng, Chih-Chia Weng, Yi-Ying Chen, Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks, IEEE Trans. Image Process. 21 (2012) 2152–2159. doi:10.1109/TIP.2011.2172798.
- [67] T. Lillesand, R.W. Kiefer, J. Chipman, Remote sensing and image interpretation, 7th ed., Wiley, 2015.
- [68] P.C. Barnum, S. Narasimhan, T. Kanade, Analysis of rain and snow in frequency space, Int. J. Comput. Vis. 86 (2010) 256–274. doi:10.1007/s11263-008-0200-2.
- [69] K. Garg, S.K. Nayar, When Does a Camera See Rain?, IEEE Int. Conf. Comput. Vis. 2 (2005) 1067–1074.
- [70] C. Dai, Y. Zheng, X. Li, Layered Representation for Pedestrian Detection and Tracking in Infrared Imagery, Comput. Vis. Pattern Recognit. - Work. 2005. CVPR Work. IEEE Comput. Soc. Conf. (2005) 13. doi:10.1109/CVPR.2005.483.
- [71] M. Bertozzi, A. Broggi, P. Grisleri, T. Graf, M. Meinecke, Pedestrian detection in infrared images, IEEE IV2003 Intell. Veh. Symp. Proc. (Cat. No.03TH8683). (2003) 662–667. doi:10.1109/IVS.2003.1212991.
- [72] D. Gerónimo, A.M. López, A.D. Sappa, T. Graf, Survey of pedestrian detection for advanced driver assistance systems, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1239–1258. doi:10.1109/TPAMI.2009.122.
- [73] S.G. Narasimhan, S.K. Nayar, Vision and the Atmosphere, Int. J. Comput. Vis. 48 (2002) 233–254. doi:10.1023/A:1016328200723.
- [74] S. Kahraman, R. De Charette, Influence of Fog on Computer Vision Algorithms, 2017.
- [75] L.L. Grewe, Detecting and counteracting atmospheric effects, in: S.S. Iyengar, R.R. Brooks (Eds.), Distrib. Sens. Networks, Chapman and Hall/CRC, 2004: pp. 213–224.
- [76] J.-P. Tarel, N. Hautiere, Fast visibility restoration from a single color or gray level image, Comput. Vis. 2009 IEEE 12th Int. Conf. (2009) 2201–2208.

doi:10.1109/ICCV.2009.5459251.

- [77] V.J. Norris, R.S. Evans, D.G. Currie, Performance comparison of visual, infrared, and ultraviolet sensors for landing aircraft in fog, Enhanc. Synth. Vis. 1999. SPIE-3691 (1999) 2–20. doi:10.1109/DASC.1999.863751.
- [78] L. Lorenzi, F. Melgani, G. Mercier, A complete processing chain for shadow detection and reconstruction in VHR images, IEEE Trans. Geosci. Remote Sens. 50 (2012) 3440–3452. doi:10.1109/TGRS.2012.2183876.
- [79] A. Bansal, H. Badino, D. Huber, Understanding how camera configuration and environmental conditions affect appearance-based localization, IEEE Intell. Veh. Symp. Proc. (2014) 800–807. doi:10.1109/IVS.2014.6856605.
- [80] H.Z. Yuan, X.Q. Zhang, Z.L. Feng, Horizon detection in foggy aerial image, IASP 10 2010 Int. Conf. Image Anal. Signal Process. (2010) 191–194. doi:10.1109/IASP.2010.5476135.
- [81] P. Radecki, M. Campbell, K. Matzen, All Weather Perception: Joint Data Association, Tracking, and Classification for Autonomous Ground Vehicles, ArXiv. 1605.02196 (2016) 35.
- [82] P. Molina, M.P. Eulalia, I. Colomina, T. Vitoria, P. Silva, J. Skaloud, W. Kornus, R. Prades, C. Aguilera, Drones to the Rescue! Unmanned Aerial Search Missions Based on Thermal Imaging and Reliable Navigation, InsideGNSS. 7 (2012) 38–47.
- [83] S.G. Narasimhan, C. Wang, S.K. Nayar, All the Images of an Outdoor Scene, in: A. Heyden,
 G. Sparr, M. Nielsen, P. Johansen (Eds.), Comput. Vis. ECCV 2002, Springer Berlin
 Heidelberg, Berlin, Heidelberg, 2006: pp. 148–162. doi:10.1007/3-540-47977-5.
- [84] T. Moranduzzo, F. Melgani, Detecting Cars in UAV Images With a Catalog-Based Approach, IEEE Trans. Geosci. Remote Sens. 52 (2014) 6356–6367. doi:10.1109/TGRS.2013.2296351.
- [85] T.P. Breckon, S.E. Barnes, M.L. Eichner, K. Wahren, Autonomous Real-time Vehicle Detection from a Medium Level UAV, Proc. 24th Int. Conf. Unmanned Air Veh. Syst. (2009) 1–9.
- [86] T. Blaschke, G.J. Hay, M. Kelly, S. Lang, P. Hofmann, E. Addink, R. Queiroz Feitosa, F. van der Meer, H. van der Werff, F. van Coillie, D. Tiede, Geographic Object-Based Image Analysis - Towards a new paradigm, ISPRS J. Photogramm. Remote Sens. 87 (2014) 180– 191. doi:10.1016/j.isprsjprs.2013.09.014.
- [87] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, S. Belongie, Objects in context, in: Proc. IEEE Int. Conf. Comput. Vis., 2007: pp. 1–8. doi:10.1109/ICCV.2007.4408986.
- [88] C. Galleguillos, S. Belongie, Context based object categorization: A critical survey, Comput. Vis. Image Underst. 114 (2010) 712–722. doi:10.1016/j.cviu.2010.02.004.
- [89] W.S. Zheng, S. Gong, T. Xiang, Quantifying and transferring contextual information in object detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 762–777. doi:10.1109/TPAMI.2011.164.
- [90] A.O. Hero, D.A. Castañón, D. Cochran, K. Kastella, Foundations and Applications of Sensor Management, Springer, New York, 2008. doi:10.1007/978-0-387-49819-5.
- [91] E.K.P. Chong, C.M. Kreucher, A.O. Hero, Partially Observable Markov Decision Process Approximations for Adaptive Sensing, Discret. Event Dyn. Syst. 19 (2009) 377–422.

doi:10.1007/s10626-009-0071-x.

- [92] R.B. Washburn, M.K. Schneider, J.J. Fox, Stochastic dynamic programming based approaches to sensor resource management, Proc. Fifth Int. Conf. Inf. Fusion. FUSION 2002. (IEEE Cat.No.02EX5997). 1 (2002) 608–615. doi:10.1109/ICIF.2002.1021210.
- [93] J.L. Williams, J.W. Fisher III, A.S. Willsky, Performance Guarantees for Information Theoretic Active Inference, Proc. Elev. Int. Conf. Artif. Intell. Stat. 2 (2007) 616–623.
- [94] G.A. McIntyre, K.J. Hintz, Comprehensive Approach to Sensor Management, Part I: A Survey of Modern Sensor Management Systems, IEEE Trans. SMC. (1999).
- [95] G. Ng, K. Ng, Sensor management what, why and how, Inf. Fusion. 1 (2000) 67–75. doi:10.1016/S1566-2535(00)00009-9.
- [96] J.L. Crowley, D. Hall, R. Emonet, Autonomic Computer Vision Systems, 2007 Int. Conf. Comput. Vis. Syst. ICVS. (2007).
- [97] O. Borzenko, Y. Lespérance, M. Jenkin, INVICON: A toolkit for knowledge-based control of vision systems, Fourth Can. Conf. Comput. Robot Vision, CRV 2007. (2007) 387–394. doi:10.1109/CRV.2007.41.
- [98] S. Moisan, J.-P. Rigault, M. Acher, P. Collet, P. Lahire, Run Time Adaptation of Video-Surveillance Systems: A Software Modeling Approach, Int. Conf. Comput. Vis. Syst. 6962 (2011) 203–212. doi:10.1007/978-3-642-23968-7_21.
- [99] N. Hochgeschwender, S. Schneider, H. Voos, G.K. Kraetzschmar, Declarative Specification of Robot Perception Architectures, in: Int. Conf. Simulation, Model. Program. Auton. Robot., Springer, 2014: pp. 291–302.
- [100] M. Russ, P. Stuetz, Airborne sensor and perception management: A conceptual approach for surveillance UAS, in: Proc. 15th Int. Conf. Inf. Fusion (FUSION 2012), IEEE, Singapore, 2012: pp. 2444–2451.
- [101] M. Russ, M. Schmitt, C. Hellert, P. Stuetz, Airborne sensor and perception management: Experiments and Results for surveillance UAS, in: AIAA Infotech@aerosp. Conf. Guid. Navig. Control Co-Located Conf. (AIAA 2013-5144), AIAA, Boston, MA, USA, 2013: pp. 1– 16. doi:10.2514/6.2013-5144.
- [102] D. Smirnov, P. Stuetz, Use case driven approach for ontology-based modeling of reconnaissance resources on-board UAVs using OWL, in: 2017 IEEE Aerosp. Conf., IEEE, Big Sky, Montana, USA, 2017.
- [103] K.A. Smith-Miles, Cross-disciplinary perspectives on meta-learning for algorithm selection, ACM Comput. Surv. 41 (2008) 1–25. doi:10.1145/1456650.1456656.
- [104] M. Tenorth, M. Beetz, KnowRob: A knowledge processing infrastructure for cognitionenabled robots, Int. J. Rob. Res. 32 (2013) 566–590. doi:10.1177/0278364913481635.
- [105] P. Robertson, J.M. Brady, Adaptive image analysis for aerial surveillance, IEEE Intell. Syst. Their Appl. 14 (1999).
- [106] D.P. Chau, K. Subramanian, F. Bremond, Adaptive Neuro-Fuzzy Controller for Multi-Object Tracker, Int. Conf. Comput. Vis. Syst. (2015) 466--476.
- [107] H. Zhang, S. Cholleti, S.A. Goldman, J.E. Fritts, Meta-Evaluation of Image Segmentation Using Machine Learning, 2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 1 (2014) 1138–1145. doi:10.1109/CVPR.2006.185.

- [108] Q. Sun, B. Pfahringer, Pairwise meta-rules for better meta-learning-based algorithm ranking, Mach. Learn. 93 (2013) 141–161. doi:10.1007/s10994-013-5387-y.
- [109] M.A. Muñoz, M. Kirley, S.K. Halgamuge, A Meta-Learning Prediction Model of Algorithm Performance for Continuous Optimization Problems, Int. Conf. Parallel Probl. Solving from Nat. 3242 (2012) 226--235. doi:10.1007/b100601.
- [110] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man. Mach. Stud. 7 (1975) 1–13. doi:10.1016/S0020-7373(75)80002-2.
- [111] A. Hamam, N.D. Georganas, A comparison of mamdani and sugeno fuzzy inference systems for evaluating the quality of experience of hapto-audio-visual applications, HAVE 2008 - IEEE Int. Work. Haptic Audio Vis. Environ. Games Proc. (2008) 87–92. doi:10.1109/HAVE.2008.4685304.
- [112] C. Hellert, P. Stuetz, Performance Prediction and Selection of Aerial Perception Functions during UAV Missions, in: 2017 IEEE Aerosp. Conf., IEEE, Big Sky, Montana, USA, 2017. doi:10.1109/AERO.2017.7943734.
- [113] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J.T. Lee, S. Mukherjee, J.K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, M. Desai, A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video, Proc. IEEE Comptuer Vis. Pattern Recognit. (2011) 3153–3160. doi:10.1109/CVPR.2011.5995586.
- [114] F. Tanner, B. Colder, C. Pullen, D. Heagy, M. Eppolito, V. Carlan, C. Oertel, P. Sallee, Overhead imagery research data set - An annotated data library & tools to aid in the development of computer vision algorithms, Appl. Imag. Pattern Recognit. Work. (AIPRW), IEEE. (2009) 1–8. doi:10.1109/AIPR.2009.5466304.
- [115] R. Collins, X. Zhou, S.K. Teh, An open source tracking testbed and evaluation web site, IEEE Int. Work. Perform. Eval. Track. Surveill. (2005) 17–24.
- [116] G. Heitz, D. Koller, Learning Spatial Context: Using Stuff to Find Things, in: Comput. Vis. – ECCV 2008, 2008: pp. 30–43. doi:10.1007/978-3-540-88682-2_4.
- [117] S. Razakarivony, F. Jurie, Vehicle detection in aerial imagery : A small target detection benchmark, J. Vis. Commun. Image Represent. 34 (2016) 187–203. doi:10.1016/j.jvcir.2015.11.002.
- [118] M. Bonetto, P. Korshunov, G. Ramponi, T. Ebrahimi, Privacy in mini-drone based video surveillance, 2015 11th IEEE Int. Conf. Work. Autom. Face Gesture Recognit. 04 (2015) 1–6. doi:10.1109/FG.2015.7285023.
- [119] G. Hummel, On synthetic datasets for development of computer vision algorithms in airborne reconnaissance applications, Universität der Bundeswehr München, 2017.
- [120] M. Nentwig, M. Miegler, M. Stamminger, Concerning the applicability of computer graphics for the evaluation of image processing algorithms, 2012 IEEE Int. Conf. Veh. Electron. Safety, ICVES 2012. (2012) 205–210. doi:10.1109/ICVES.2012.6294288.
- [121] A. Rozantsev, V. Lepetit, P. Fua, On Rendering Synthetic Images for Training an Object Detector, Comput. Vis. Image Underst. 137 (2015) 24–37.
- [122] A. Shafaei, J.J. Little, M. Schmidt, Play and Learn: Using Video Games to Train Computer Vision Models, in: Br. Mach. Vis. Conf. 2016, 2016.

- [123] M. Johnson-Roberson, C. Barto, R. Mehta, S.N. Sridhar, R. Vasudevan, Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks?, 2017 IEEE Int. Conf. Robot. Autom. (2017) 746–753. doi:10.1109/ICRA.2017.7989092.
- [124] S.R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for Data: Ground Truth from Computer Games, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Comput. Vis. – ECCV 2016, Springer International Publishing, Amsterdam, The Netherlands, 2016: pp. 102–118. doi:10.1007/978-3-319-46475-6_7.
- [125] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. doi:10.1561/2000000039.
- [126] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and Harnessing Adversarial Examples, (2014) 1–11. doi:10.1109/CVPR.2015.7298594.
- [127] A. Nguyen, J. Yosinski, J. Clune, Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, 2015 IEEE Conf. Comput. Vis. Pattern Recognit. (2015) 427–436. doi:10.1109/CVPR.2015.7298640.
- [128] D. Marmanis, J.D. Wegner, S. Galliani, K. Schindler, M. Datcu, U. Stilla, Semantic Segmentation of Aerial Images with an Ensemble of CNNs, ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 3 (2016) 473–480. doi:10.5194/isprs-annals-III-3-473-2016.
- [129] D. Nauck, R. Kruse, Neuro-fuzzy systems for function approximation, Fuzzy Sets Syst. 101 (1999) 261–271. doi:10.1016/S0165-0114(98)00169-9.
- [130] J.S.R. Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference System, IEEE Trans. Syst. Man Cybern. 23 (1993) 665–685. doi:10.1109/21.256541.
- [131] J. Fernández-Hernandez, D. González-Aguilera, P. Rodríguez-Gonzálvez, J. Mancera-Taboada, Image-Based Modelling from Unmanned Aerial Vehicle (UAV) Photogrammetry: An Effective, Low-Cost Tool for Archaeological Applications, Archaeometry. 57 (2015) 128–145. doi:10.1111/arcm.12078.
- [132] P. Azad, T. Gockel, R. Dillmann, Computer Vision: Das Praxisbuch, Elektor Verlag, 2007.
- [133] S. Suzuki, K. Abe, Topological structural analysis of digital binary image by border following, Comput. Vision, Graph. Image Process. 30 (1985) 32–46.
- [134] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, J. Electron. Imaging. 13 (2004) 146–165. doi:10.1117/1.1631315.
- [135] T. Tuytelaars, K. Mikolajczyk, Local Invariant Feature Detectors: A Survey, Found. Trends[®] Comput. Graph. Vis. 3 (2007) 177–280. doi:10.1561/0600000017.
- [136] D. Li, Z. Lu, T. Bansal, E. Schilling, P. Sinha, ForeSight: Mapping vehicles in visual domain and electronic domain, in: Proc. - IEEE INFOCOM, IEEE, 2014: pp. 1995–2003. doi:10.1109/INFOCOM.2014.6848140.
- [137] M. Silvey, PPG data shows white continues to dominate as most popular global vehicle
color, PPG Corp. Commun. (2014) 1.
http://corporate.ppg.com/Media/Newsroom/2014/PPG-data-shows-white-
continues-to-dominate-as-most (accessed January 1, 2017).
- [138] BASF, BASF European Color Report For Automotive OEM Coatings 2016, Münster, 2017.

- [139] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.
- [140] P. Shiju, Edge Based Template Matching, Code Proj. (2010). http://www.codeproject.com/Articles/99457/Edge-Based-Template-Matching (accessed October 6, 2016).
- [141] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, Proc. 25th DAGM Pattern Recognit. Symp. (2003) 297–304. doi:10.1007/978-3-540-45243-0_39.
- [142] S. Liao, X. Zhu, Z. Lei, L. Zhang, S.Z. Li, Learning Multi-scale Block Local Binary Patterns for Face Recognition, in: Int. Conf. Biometrics, 2007: pp. 828–837.
- [143] R.B. Girshick, P.F. Felzenszwalb, D. McAllester, Discriminatively Trained Deformable Part Models, Release 5, (2012). http://people.cs.uchicago.edu/~rbg/latent-release5/ (accessed February 24, 2016).
- [144] G. Bradski, The OpenCV Library, Dr. Dobb's J. Softw. Tools. (2000).
- [145] D. Powers, Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation, J. Mach. Learn. Technol. 2 (2011) 37–63.
- [146] D.D. Nauck, Data Analysis with Neuro-Fuzzy Methods, Otto-von-Guericke-Universitaet Magdeburg, 2000.
- [147] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed., Elsevier Science & Technology, 2016.
- [148] M.F. Møller, A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning Supervised Learning, Neural Networks. 6 (1993) 525–533. doi:10.1016/S0893-6080(05)80056-5.
- [149] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Second Edi, Springer, 2017.
- [150] J. Lee Rodgers, W. Alan Nice Wander, Thirteen ways to look at the correlation coefficient, Am. Stat. 42 (1988) 59–66. doi:10.1080/00031305.1988.10475524.
- [151] V.M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual Domain Adaptation: A survey of recent advances, IEEE Signal Process. Mag. 32 (2015) 53–69. doi:10.1109/MSP.2014.2347059.
- [152] T.B. Sheridan, Adaptive Automation, Level of Automation, Allocation Authority, Supervisory Control, and Adaptive Control: Distinctions and Modes of Adaptation, IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans. 41 (2011) 662–667. doi:10.1109/TSMCA.2010.2093888.
- [153] M.R. Endsley, D.B. Kaber, Level of automation effects on performance, situation awareness and workload in a dynamic control task, 1999. doi:10.1080/001401399185595.
- [154] C. Ruf, P. Stuetz, Establishing a Variable Automation Paradigm for UAV-Based Reconnaissance in Manned-Unmanned Teaming Missions: Experimental Evaluation and Results, in: J. Chen (Ed.), Adv. Hum. Factors Robot. Unmanned Syst. Proc. AHFE 2017 Int. Conf. Hum. Factors Robot. Unmanned Syst., Springer, Cham, 2019: pp. 24–35. doi:10.1007/978-3-319-94346-6_3.
- [155] C. Ruf, P. Stuetz, Model-Driven Sensor Operation Assistance for a Transport Helicopter Crew in Manned-Unmanned Teaming Missions: Selecting the Automation Level by

Machine Decision-Making, in: P. Savage-Knepshield, J. Chen (Eds.), Adv. Hum. Factors Robot. Unmanned Syst. Proc. AHFE 2016 Int. Conf. Hum. Factors Robot. Unmanned Syst., Springer International Publishing, Cham, 2017: pp. 253–265. doi:10.1007/978-3-319-41959-6_21.

- [156] N. Farmani, L. Sun, D. Pack, Optimal UAV Sensor Management and Path Planning for Tracking Multiple Mobile Targets, ASME 2014 Dyn. Syst. Control Conf. 2 (2014) V002T25A003. doi:10.1115/DSCC2014-6232.
- [157] M. Schmitt, P. Stuetz, Multi-UAV based Helicopter Landing Zone Reconnaissance: Information Level Fusion and Decision Support, in: Eng. Psychol. Cogn. Ergon. - 14th Int. Conf. EPCE 2017, Held as Part HCI Int. 2017, Springer International Publishing, Vancouver, BC, Canada, 2017. doi:10.1007/978-3-319-58475-1_20.
- [158] C. Hellert, P. Stuetz, Selecting Adequate Aerial Perceptual Functions with Fuzzy Logic, in: First Int. Conf. Appl. Syst. Vis. Paradig., 2016.
- [159] C. Hellert, P. Stütz, Adaptive Algorithmenauswahl für die Fahrzeugdetektion auf Luftbildaufnahmen, in: Forum Bild., 2018: p. 12.
- [160] C. Hellert, D. Smirnov, M. Russ, P. Stuetz, A High Level Active Perception Concept For UAV Mission Scenarios, in: Dtsch. Luft- Und Raumfahrtkongress 2012, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., Berlin, Germany, 2012.
- [161] NIMA National Imagery and Mapping Agency, Department of Defense World Geodetic System 1984, Tech. Rep. 8350.2, Third Ed. (2000) 175. doi:NGA.STND.0036_1.0.0_WGS84 NATIONAL.
- [162] L. Papula, Vektorrechnung, in: E. Schmitt (Ed.), Math. Formelsammlung, 9th ed., Vieweg & Sohn Verlag, Wiesbaden, 2006: pp. 45–66.
- [163] G.A. Korn, T.M. Korn, Appendix B: B9. Plane and Spherical Trigonometry: Formulas Expressed in Terms of the Haversine Function, in: Math. Handb. Sci. Eng. Defin. Theorems, Formulas Ref. Rev., 3rd ed., Dover Publications, Mineola, 2000: pp. 892–893.
- [164] J. Herzog, Aerial Photography UAV Icon, Creat. Commons BY Namensnennung 4.0.
 (2017). https://de.wikipedia.org/wiki/Datei:Aerial_Photography_UAV_Icon.svg (accessed May 22, 2018).
- [165] NASA JPL, NASA Shuttle Radar Topography Mission Global 3 arc second, NASA EOSDIS L. Process. DAAC. (2013). doi:10.5067/MEaSUREs/SRTM/SRTMGL3.003.

Anhang

A Koordinatensysteme

Im Folgenden werden die in dieser Arbeit verwendeten Koordinatensysteme erläutert. Als Weltkoordinatensystem wird das geodätische Referenzsystem WGS84 [161] verwendet. Dabei ist der Erdradius das arithmetische Mittel von Äquatorradius R_A und Polarradius R_P und kann mit

$$R = \frac{2R_A + R_P}{3} = 6.371.008,767 \text{ m} \quad R_A = 6.378.137 \text{ m} \quad R_P = 6.356.752,3 \text{ m}$$
(9.1)

berechnet werden. Eine dreidimensionale Position wird mit einem Breiten- und Längengrad, sowie einer Höhe, entweder über Meeresspiegel (AMSL, engl. *above mean sea level*) oder über Grund (AGL, engl. *above ground level*), angegeben. Der Breitengrad φ und der Längengrad λ werden in Grad angegeben und die Höhe in Metern.

Des Weiteren gibt es das Flugzeugkoordinatensystem, welches seinen Ursprung im Schwerpunkt des Flugzeuges hat und die Rollachse x^a entlang des Rumpfes zum Cockpit zeigt und ein Rechtssystem ist. Die Nickachse zeigt entlang des rechten Flügels und die positive Richtung der Gierachse zeigt nach unten. Für die am Flugzeug angebrachte Sensorik zur Umweltwahrnehmung kann jeweils ein Koordinatensystem definiert werden. In dieser Arbeit werden bildgebende Sensoren (Kameras) verwendet und es lässt sich ein Kamerakoordinatensystem definieren, welches seinen Ursprung in der Mitte des Bildsensors hat und bei dem die Rollachse x^c identisch mit der optischen Achse des Sensors ist. Das aufgenommene Bild durch den Bildsensor besitzt ein Bildkoordinatensystem, wobei der Ursprung oben links im Bild und Einheit in Pixeln ist. In Bild 9.1 sind die vorgestellten Koordinatensysteme nochmals illustriert.



Bild 9.1: Bezeichnungen der verwendeten Koordinatensysteme.

B Sensor-Footprint

In der Fernerkundung ist der Sensor-Footprint die Fläche am Boden, die ein bildgebender Sensor zu einem Zeitpunkt gleichzeitig wahrnehmen kann. In dieser Arbeit wird der Sensor-Footprint verwendet, um geographische Informationen über das zu beobachtende Gebiet und die Bodenauflösung zu bestimmen.

Um den Sensor-Footprint zu berechnen, wird ein relatives kartesisches Weltkoordinatensystem erzeugt, das seinen Ursprung unter dem Flugzeugkoordinatenursprung hat und lediglich um die Höhe über Meeresspiegel verschoben ist. Außerdem sei die Ebene S^r definiert, die einer Approximation der Bodenoberfläche entspricht. Diese liegt in der xy-Ebene und ist um die Höhe h^r , welche der Höhe über MSL der Bodenoberfläche unter der Position des Flugzeuges entspricht, verschoben. In Bild 9.2 ist die approximierte Bodenoberfläche dargestellt, sowie die Position p_a^r des Flugzeugs.





Ziel ist es, die Ebene S^r in das Kamerakoordinatensystem zu transformieren und dort mit der optischen Achse des Sensors zu schneiden, um den Schnittpunkt auf der Bodenoberfläche zu erhalten. Das in dieser Arbeit betrachtete System sieht Sensoren vor, die durch eine kardanische Aufhängung frei in zwei Achsen bewegt werden können. Dabei handelt es sich um den Elevations- und Azimutwinkel bzw. Höhen- und Vertikalwinkel. Allgemein lässt sich die Ebene S^r durch homogene Transformation zunächst in das Flugzeugkoordinatensystem und anschließend in das Kamerakoordinatensystem transformieren:

$$S^c = T_{ac} T_{ra} S^r \tag{9.2}$$

Dabei ist T_{ac} die Transformationsmatrix um von Flugzeugkoordinaten in Kamerakoordinaten und T_{ra} die Transformationsmatrix, um von relativen Weltkoordinaten in Flugzeugkoordinaten zu transformieren. Das Flugzeugkoordinatensystem ist gegenüber dem relativen Weltkoordinatensystem lediglich um die Flughöhe h_a^r verschoben und um 180° gerollt. Außerdem lässt sich die Flugzeuglage ignorieren, da die Lage des Kamerakoordinatensystems durch eingebaute Sensorik in der kardanischen Aufhängung gegenüber dem Weltkoordinatensystem angegeben ist, wodurch

$$T_{ra} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 180 & \sin 180 & 0 \\ 0 & -\sin 180 & \cos 180 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h_a^r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -h_a^r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(9.3)

ist. Für die weitere Transformation in das Kamerakoordinatensystem wird die Lage benötigt, die durch den Roll-, Nick- und Gierwinkel, die mit den Formelzeichen ϕ , θ und ψ bezeichnet werden, gegeben ist. Gedreht wird zunächst um die *z*-Achse mit dem Gierwinkel, anschließend

um die *y*-Achse mit dem Nickwinkel und zum Schluss um die *x*-Achse mit dem Rollwinkel. Dabei sehen die Drehmatrizen wie folgt aus für ein Rechtssystem:

$$R_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad R_{\theta} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad R_{\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9.4)$$

Neben der Rotation kann das Kamerakoordinatensystem gegenüber dem Flugzeugkoordinatensystem zusätzlich verschoben sein um einen Translationsvektor t^c , der die Verschiebung des Sensors bzgl. des Schwerpunktes des Flugzeugs beschreibt. Daraus ergibt sich die Transformationsmatrix

$$T_{ac} = \begin{bmatrix} R_{\phi} R_{\theta} R_{\psi} & 0\\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_{x}^{c} \\ 0 & 1 & 0 & t_{y}^{c} \\ 0 & 0 & 1 & t_{z}^{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$
(9.5)

Die transformierte Ebene S^c kann jetzt mit der optischen Achse, welche der *x*-Achse entspricht, der Kamera geschnitten werden [162], um die zentralen Bodenpunkt p_c^c zu erhalten, sofern sich Ebene und Strahl schneiden. Der zentrale Bodenpunkt in Kamerakoordinaten entspricht dem Punkt in der Mitte des aufgenommenen Bildes einer Kamera. Durch Rücktransformation des Punktes in das relative kartesische Koordinatensystem erhält man den zentralen Bodenpunkt p_c^r in relativen kartesischen Koordinaten:

$$p_c^r = T_{ra}^{-1} T_{ac}^{-1} p_c^c \tag{9.6}$$

Anschließend lassen sich die relative Flugzeugposition und der relative zentrale Bodenpunkt auf die Bodenoberfläche projizieren durch Entfernen der Höheninformation bzw. *z*-Achse, um die Umrechnung in geodätische Koordinaten vorzunehmen. Damit lässt sich die Distanz *d* zwischen zentralem Bodenpunkt und Flugzeugposition und der Winkel α zwischen *x*-Achse und $\overline{p_a^r p_c^r}$ bestimmen. Der Winkel α entspricht dabei dem Winkel nach magnetisch Nord. Außerdem ist die Flugzeugposition in geodätischen Koordinaten (φ_a , λ_a) bekannt. Dadurch lässt sich über die umgestellte Haversine Formel [163] die geodätischen Koordinaten des zentralen Bodenpunktes berechnen:

$$d_{a} = \frac{d}{R}$$

$$\varphi_{c} = \sin^{-1}(\sin\varphi_{a}\cos d_{a} + \cos\varphi_{a}\sin d_{a}\cos\alpha)$$

$$\lambda_{c} = \lambda_{a} + \operatorname{atan2}(\sin\alpha\sin d_{a}\cos\varphi_{a}, \cos d_{a} - \sin\varphi_{a}\sin\varphi_{c})$$
(9.7)

Die Distanz d_a wird Winkelabstand genannt und R ist der Erdradius (siehe Anhang A). Für den Sensor-Footprint F_c benötigt man wird die vier Eckpunkte, aufgespannt durch den vertikalen α_v und horizontalen Öffnungswinkel α_h der Kamera, wie in Bild 9.3 dargestellt. Die Berechnung der geodätischen Koordinaten der vier Eckpunkte ist identisch mit dem des zentralen Bodenpunktes, mit der Ausnahme, dass die optische Achse entsprechend rotiert werden muss, bevor der Schnittpunkt mit der Bodenoberfläche berechnet wird:

$$\overrightarrow{v_{0}} = (1,0,0)^{T}$$

$$\overrightarrow{v_{TL}} = R_{\theta}(\alpha_{v})R_{\psi}(-\alpha_{h})\overrightarrow{v_{0}} \qquad \overrightarrow{v_{TR}} = R_{\theta}(\alpha_{v})R_{\psi}(\alpha_{h})\overrightarrow{v_{0}}$$

$$\overrightarrow{v_{BL}} = R_{\theta}(-\alpha_{v})R_{\psi}(-\alpha_{h})\overrightarrow{v_{0}} \qquad \overrightarrow{v_{BR}} = R_{\theta}(-\alpha_{v})R_{\psi}(\alpha_{h})\overrightarrow{v_{0}}$$
(9.8)

Die x-Achse ist mit $\overline{v_0}$ bezeichnet und die vier rotierten Vektoren sind aus Sicht des aufgenommenen Bildes durch die Kamera benannt: oben-links $\overrightarrow{v_{TL}^c}$, oben-rechts $\overrightarrow{v_{TR}^c}$, unten-links $\overrightarrow{v_{BL}^c}$ und unten-rechts $\overrightarrow{v_{BR}^c}$.



Bild 9.3: Darstellung des Sensor-Footprints auf der Bodenoberfläche.

Durch die Annahme, dass die Bodenoberfläche eine Ebene ist, ist der Fußabdruck ungenau, wenn das Gelände nicht eben ist. Der Grad der Genauigkeit lässt sich verbessern, wenn an den berechneten geodätischen Koordinaten die Höhe des Grundes, sofern eine entsprechende Datenbank verfügbar ist, neu ermittelt und jeweils die Bodenoberfläche mit dieser Höhe verwendet wird um den jeweiligen Punkt neu zu bestimmen.

C DPM-Modelle

In Bild 9.4 und Bild 9.5 sind die beiden erzeugten DPM Modellmengen zu sehen. Dabei zeigt die erste Spalte die Wurzelfilter der verschiedenen Modellkomponenten, die zweite Spalte die zugehörigen Teilefilter und die letzte Spalte das räumliche Modell, welches die Verschiebungskosten visualisiert, wobei hellere Regionen höhere Kosten bedeuten.



Bild 9.4: DPM-Modellmenge erzeugt auf dem 60° Elevationsdatensatz.



Bild 9.5: DPM-Modellmenge erzeugt auf dem 75° Elevationsdatensatz.

D Zugehörigkeitsfunktionen der modellierten FIS

Für die modellierten Fuzzy-Inferenzesysteme wurden die Zugehörigkeitsfunktionen für die Umgebungszustände Bewölkungsgrad, Elevationswinkel, die Bodenbedeckung und der normierte Reliefwert wie in Bild 9.6 verwendet. Außerdem ist dort die Modellierung der Zugehörigkeitsfunktionen für den Leistungsindex zu finden.



Bild 9.6: Zugehörigkeitsfunktionen der modellierten FIS.

Bei den Zugehörigkeitsfunktionen des normierten Elevationswinkels sind die Beschriftungen der Funktionen so zu verstehen, dass erst die Steigung und dann die Richtung angegeben ist. Damit meint z. B. "flach-Boden" einen flachen Elevationswinkel, wobei der Sensor Richtung Bodenoberfläche gerichtet ist.

E Missionsabschnitte der Observierungsmission

Die Mission Observierung (siehe Abschnitt 7.4.1.1) besteht aus insgesamt fünf Abschnitten. Die detaillierten Lagekarten der einzelnen Abschnitte sind in Bild 9.7 dargestellt. Hierbei zeigt die weiße Linie die Flugroute und die gelbe Linie die Bewegung des Fahrzeugs bzw. die Abtastungsroute der Sensorik. Bild 9.7a und in Bild 9.7c beinhalten anstatt der Fahrzeugbewegung die Fahrzeugposition, markiert mit einem weißen Kreis, da hier das Fahrzeug nicht fährt. Weiterhin markiert ein der Buchstabe "S" den Start einer Route und "E" das Ende.



Bild 9.7: Lagekarten der einzelnen Missionsabschnitte der Mission *Observierung*. In (a) ist der Missionsabschnitt 1, in (b) der Missionsabschnitt 2, in (c) der Missionsabschnitt 3 und in (d) der Missionsabschnitt 4 und 5 dargestellt.

Im Folgenden werden die Missionsparameter und die Veränderungen der Umgebungszustände für jeden Missionsabschnitt präsentiert.

Missionsabschnitt 1

In diesem Missionsabschnitt steht das zu beobachtende Fahrzeug an einer Hafenanlage. Die Eckdaten der Missionsparameter sind in Tabelle 9.1 zusammengefasst.

Parameter	Wert
Start Missionsabschnitt	01.11.2017 um 13:00 Uhr
Ende Missionsabschnitt	01.11.2017 um 16:00 Uhr
Bodenpixelauflösung	0,00694 <i>m/px</i>
Normierte Nebeldichte	0
Normierte Niederschlagsmenge	0
Normierter Bewölkungsgrad	0
Fluggeschwindigkeit	12 m/s
Aufzeichnungsrate	0.1 <i>Bps</i>

Tabelle 9.1: Missionsparameter des 1. Abschnittes der Mission Observierung.

Bild 9.8 zeigt den Verlauf des Reliefwertes und des Elevationswinkels in diesem Missionsabschnitt. In Bild 9.9 ist die Veränderung der Bodenbedeckung über die Tageszeit aufgetragen.



Bild 9.8: Reliefwert- und Elevationswinkeländerung des 1. Abschnittes der Mission Observierung.



Bild 9.9: Variation der Bodenbedeckungen des 1. Abschnittes der Mission Observierung.

Missionsabschnitt 2

Im zweiten Missionsabschnitt fährt ein Fahrer das zu beobachtende Fahrzeug entlang einer Straße von der Hafenanlage zu einer Einrichtung und passiert dabei drei Dörfer. In Tabelle 9.2 sind die Missionsparameter dargestellt.

Parameter	Wert
Start Missionsabschnitt	01.11.2017 um 16:00 Uhr
Ende Missionsabschnitt	01.11.2017 um 16:30 Uhr
Bodenpixelauflösung	0,00694 <i>m/px</i>
Fluggeschwindigkeit	30 m/s
Aufzeichnungsrate	1 Bps

Tabelle 9.2: Missionsparameter des 2. Abschnittes der Mission Observierung.

Während diesem Missionsabschnitt steigt der Bewölkungsgrad auf 0,5, wie in Bild 9.10 abgebildet. Desweitern zeigt Bild 9.11 die Veränderung des Reliefwerts und des Elevationswinkels über die Zeit. In Bild 9.12 sind die Bodenbedeckungen über die Tageszeit aufgetragen.



Bild 9.10: Wetterentwicklung im Abschnitt 2 der Mission Observierung.



Bild 9.11: Reliefwert- und Elevationswinkeländerung des 1. Abschnittes der Mission Observierung.



Bild 9.12: Variation der Bodenbedeckungen des 2. Abschnittes der Mission Observierung.

Missionsabschnitt 3

Der Missionsabschnitt 3 beinhaltet die Observierung des Fahrzeuges während der Nachtphase. Dabei steigt neben dem Bewölkungsgrad auch der Niederschlag an. In Tabelle 9.3 sind die Missionsparameter des 3. Abschnitts gelistet.

Parameter	Wert
Start Missionsabschnitt	01.11.2017 um 16:30 Uhr
Ende Missionsabschnitt	02.11.2017 um 02:00 Uhr
Bodenpixelauflösung	0,00694 <i>m/px</i>
Reliefwert	0
Fluggeschwindigkeit	12 m/s
Aufzeichnungsrate	0.05 Bps

Tabelle 9.3: Missionsparameter des 3. Abschnittes der Mission Observierung.

Bild 9.13 zeigt die Veränderung der Witterung, Bild 9.14 stellt den Elevationswinkel über die Missionszeit dar und Bild 9.15 präsentiert die Variation der Bodenbedeckung.



Bild 9.13: Wetterentwicklung im Abschnitt 3 der Mission Observierung.





Bild 9.14: Elevationswinkeländerung des 3. Abschnittes der Mission Observierung.

Bild 9.15: Variation der Bodenbedeckungen des 3. Abschnittes der Mission Observierung.

Missionsabschnitt 4

Während diesem Missionsabschnitt fährt der Fahrer das Fahrzeug von der Einrichtung in eine Stadt und parkt dieses von einem Gebäude. Die Eckdaten der Missionsparameter sind in Tabelle 9.4 zusammengefasst.

Parameter	Wert
Start Missionsabschnitt	02.11.2017 um 02:00 Uhr
Ende Missionsabschnitt	02.11.2017 um 02:30 Uhr
Bodenpixelauflösung	0,00694 <i>m/px</i>
Normierte Nebeldichte	0
Normierte Niederschlagsmenge	0
Normierter Bewölkungsgrad	1
Fluggeschwindigkeit	30 m/s
Aufzeichnungsrate	1 <i>Bps</i>

Tabelle 9.4: Missionsparameter des 4. Abschnittes der Mission Observierung.

In Bild 9.16 ist der Reliefwert und Elevationswinkel des 4. Missionsabschnittes über die Missionszeit aufgetragen. Des Weiteren zeigt Bild 9.17 die Bodenbedeckung.



Bild 9.16: Reliefwert- und Elevationswinkeländerung des 4. Abschnittes der Mission Observierung.



Bild 9.17: Variation der Bodenbedeckungen des 4. Abschnittes der Mission Observierung.

Missionsabschnitt 5

Im letzte Missionsabschnitt wird das in der Stadt geparkte Fahrzeug weiterhin beobachtet. Dabei verringert sich der Bewölkungsgrad und in der Morgendämmerung zieht Nebel auf. Tabelle 9.5 stellt die Missionsparameter dar.

Parameter	Wert
Start Missionsabschnitt	01.11.2017 um 02:30 Uhr
Ende Missionsabschnitt	02.11.2017 um 13:00 Uhr
Bodenpixelauflösung	0,00694 <i>m/px</i>
Reliefwert	0
Fluggeschwindigkeit	12 m/s
Aufzeichnungsrate	0,05 <i>Bps</i>

Tabelle 9.5: Missionsparameter des 5. Abschnittes der Mission Observierung.

Die Witterungsverhältnisse sind in Bild 9.18 dargestellt. Bild 9.19 zeigt den Verlauf des Elevationswinkels über die Missionszeit und Bild 9.20 präsentiert die Variationen der Bodenbedeckung.



0 -45 -90 -90 02:30 04:00 05:30 07:00 08:30 10:00 11:30 13:00 Tageszeit

Bild 9.19: Elevationswinkeländerung des 5. Abschnittes der Mission Observierung.



Bild 9.20: Variation der Bodenbedeckungen des 5. Abschnittes der Mission Observierung.