

Kontextsensitive Erkennung und Interpretation fahrrelevanter statischer Verkehrselemente

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Dennis Nienhüser

aus Osnabrück

Tag der mündlichen Prüfung: 14. Januar 2014

Erster Gutachter: Prof. Dr.-Ing. J. Marius Zöllner

Zweiter Gutachter: Prof. Dr.-Ing. Christoph Stiller

Kurzfassung

Kognitive Assistenzsysteme unterstützen Verkehrsteilnehmer in Form von Fahrerassistenzsystemen aktiv oder passiv im Straßenverkehr. Als Komponenten (teil-)autonomer Fahrzeuge handeln sie sogar selbständig. In jedem Fall sollen sie Verkehrsteilnehmern eine sichere, komfortable und energieeffiziente Fahrt zum Ziel ermöglichen. Um dies zu leisten, müssen sie Verkehrssituationen erkennen und verstehen. Ein wesentlicher Aspekt davon sind statische Verkehrselemente (Verkehrszeichen und Ampeln). Herkömmliche Fahrerassistenzsysteme basieren auf Kartendiensten und Eigenlokalisierung und hängen dadurch von der Vollständigkeit und Aktualität der zugrunde liegenden Kartendaten sowie der Qualität der Positionsbestimmung ab. Dynamische Informationen wie durch Verkehrsbeeinflussungsanlagen gesteuerte Wechselverkehrszeichen oder Ampeln können sie nicht auswerten. Am Markt erhältliche kognitive Assistenzsysteme erkennen eine stark begrenzte Menge von Verkehrszeichen (Überholverbote, Tempolimits und verwandte Schilder), ihre Leistung ist aber weit vom menschlichen Situationsverständnis entfernt.

In dieser Arbeit werden Methoden und Verfahren zur Umwelterkennung und Situationsinterpretation entwickelt, mit denen statische Verkehrselemente erkannt und im Kontext der Verkehrssituation interpretiert werden. Es werden robuste Verfahren zur Erkennung charakteristischer Merkmale von Verkehrselementen vorgestellt und komplette Prozessketten entwickelt, mit denen alle Arten von Verkehrselementen in Echtzeit detektiert, klassifiziert, verfolgt und zeitlich fusioniert werden können. Darauf aufbauend werden Methoden entwickelt, um Eigenschaften von Verkehrselementen und Beziehungen zu anderen konkreten oder abstrakten Objekten zu analysieren und auszuwerten. Ein wichtiger Anwendungsfall ist die Bestimmung der Relevanz von Verkehrselementen in Bezug auf eine sichere und zielgerichtete Aufgabenausführung. Die Praxistauglichkeit der entwickelten Methoden und Verfahren wird durch umfangreiche Experimente demonstriert. Dabei wird auf die Verwendung realer Daten, kostengünstiger Sensorik und Echtzeitverarbeitung Wert gelegt.

Abstract

Cognitive assistance systems support traffic participants in the form of driver assistance system in an active or passive manner. Part of (semi-) autonomous vehicles they even act on their own. Each of their variants shall ensure a safe, comfortable and energy efficient ride. A premise to accomplish said goals is the ability to recognize and understand driving situations. This includes static traffic elements (traffic signs and traffic lights). Conventional driver assistance systems are based on map services and sensors for localization. Hence their performance is inherently constricted by the integrity and the actuality of map data as well as the precision of localization. They are agnostic to dynamic information like variable message signs controlled by traffic control systems as well as traffic lights. Off-the-shelf cognitive assistance systems recognize but a limited subset of traffic signs (speed limits, no overtaking signs and related ones). Their performance is nowhere near human situation understanding.

This work presents methods and techniques for environment perception and situation interpretation. They allow for recognizing static traffic elements and their interpretation in the context of the traffic situation at hand. Robust methods to recognize characteristic traffic element features are developed and a set of complete processing chains is presented which detect, classify, track and fuse the most important types of traffic elements. Subsequently methods to analyze and evaluate properties of traffic elements and their relation to other precise or abstract objects are presented. An important use case is relevance estimation of traffic elements subject to a safe and goal-oriented task execution. The practical feasibility of the proposed methods and techniques is demonstrated in an extensive set of experiments. Great importance is attached to usage of real data, cheap sensors and real-time performance.

Inhaltsverzeichnis

Kurzfassung	i
Symbolverzeichnis	vii
1 Einführung	1
1.1 Problemstellung und Zielsetzung	2
1.2 Rahmenbedingungen und Schwerpunkte der Arbeit	3
1.3 Einordnung und Wissenschaftlicher Beitrag	5
1.4 Aufbau der Arbeit	6
2 Stand der Technik	9
2.1 Themenfelder	9
2.2 Überblick über Forschungsarbeiten	11
2.3 Forschungsgruppen	21
2.3.1 LaRA – La Route Automatisée	21
2.3.2 Signalverarbeitungs-Gruppe der Universität Alcalá	23
2.3.3 VisLab	24
2.4 Ausgewählte Einzelarbeiten	27
2.4.1 Viola/Jones-Ansatz zur Erkennung von Tempolimits	27
2.4.2 Erkennung von Ampeln mittels Hough-Transformation	28
2.4.3 Erkennung von Ampeln mittels Viola/Jones-Objekterkennung	29
2.4.4 Detektion von Text auf Vorwegweisern	29
2.4.5 Bayessche Fusion zur Schätzung des gültigen Tempolimits	31
2.5 Zusammenfassung und Forschungsbedarf	31
3 Umfelderkennung	33
3.1 Sensorik	34
3.2 Segmentierung und Detektion	37
3.2.1 Bildpunktbasierte Segmentierung	39
3.2.2 Hough-Transformation für Rechtecke	41
3.2.3 Radiale Symmetrie für gleichseitige n-Ecke	44
3.2.4 Radiale Symmetrie für Kreise	47
3.2.5 Morphologische Bildverarbeitung	47
3.2.6 Viola/Jones-Verfahren	50
3.2.7 Laufzeitoptimierungen	52
3.3 Merkmalsextraktion und Klassifikation	54
3.3.1 Grauwert-Transformation	56

3.3.2	Blockweise Diskrete Kosinustransformation	56
3.3.3	Blockweise Histogramm-Transformation	57
3.3.4	Histogram of Oriented Gradients (HOG)	59
3.3.5	Hybride Merkmalstransformationen	59
3.3.6	Hierarchische Support-Vektor-Maschinen (SVMs)	59
3.3.7	Aktives Lernen	66
3.4	Tracking und temporale Fusion	70
3.4.1	Tracking	71
3.4.2	Konfidenzberechnung für statische Elemente	74
3.4.3	HMMs zur Zustandsschätzung semi-statischer Elemente	76
3.5	Anwendungen	78
3.5.1	Erkennung von Ronden	78
3.5.2	Erkennung von Warnhinweisen	80
3.5.3	Erkennung von Vorwegweisern	81
3.5.4	Erkennung von Zusatzzeichen	85
3.5.5	Erkennung von Ampeln	85
3.6	Zusammenfassung	89
4	Situationsinterpretation	91
4.1	Modellbasierte Bestimmung räumlicher Relationen	91
4.2	Kontextsensitive Dempster/Shafer-Fusion	95
4.2.1	Dempster/Shafer-Fusion	96
4.2.2	Bestimmung des gültigen Tempolimits	97
4.2.3	Adaption an den Situationskontext	100
4.3	Markov-Logik-Netze (MLNs)	104
4.3.1	Inferenz	107
4.3.2	Lernen	108
4.3.3	Wissensrepräsentation	109
4.3.4	Erkennung von Baustellen	110
4.3.5	Relevanzbestimmung für eingeschränkte Tempolimits	113
4.3.6	Relevanzbestimmung für Ampeln	114
4.4	Zusammenfassung	117
5	Experimente und Evaluation	119
5.1	Erkennung von Ronden	120
5.2	Erkennung von Warnhinweisen	135
5.3	Erkennung von Vorwegweisern	137
5.4	Erkennung von Zusatzzeichen	140
5.5	Erkennung von Ampeln	143
5.6	Semi-automatische Kartierung mit einem Segway-Personentransporter	148
5.7	Verkehrszeichenerkennung auf Mobilgeräten	151
5.8	Probabilistische Zuordnung von Verkehrszeichen zu Fahrspuren	156
5.9	Adaptive Dempster/Shafer-Fusion zur Bestimmung des Tempolimits	158

5.10	Markov-Logik-Netze	162
5.11	Zusammenfassung	165
6	Schlussbetrachtung	169
6.1	Zusammenfassung	169
6.2	Diskussion	170
6.3	Ausblick	170
A	Quellennachweise	173
A.1	Veröffentlichungen	173
A.2	Studentische Arbeiten	175
A.3	Literatur	176
A.4	Patente	187
A.5	Abbildungen	188

Symbolverzeichnis

Abkürzungen

ADAS	advanced driver assistance system
ADR	angular diversity ranking
CB-RS	class-biased random sampling
D/S	Dempster/Shafer
DoG	difference of Gaussians
DCT	diskrete Kosinus-Transformation
DST	Evidenztheorie von Dempster und Shafer
FFT	schnelle Fourier-Transformation
FP	false positive
FN	false negative
GIS	Geoinformationssystem
GPS	Global Positioning System
GPU	Grafikprozessor (englisch graphics processing unit)
HOG	histogram of oriented gradients
HMM	hidden Markov model
IMU	inertial measurement unit
JNI	java native interface
LED	Leuchtdiode
LCP	local contour pattern
LBP	local binary pattern
LDA	linear discriminant analysis
MAP	maximum a posteriori
MCMC	Markov chain Monte Carlo
MDA	multiple discriminant analysis
MMS	Mensch-Maschine-Schnittstelle
MLP	multilayer perceptron

MPE	most probable explanation
MN	Markov-Netz
MLN	Markov-Logik-Netz
MTT	Multi-Target-Tracker
OCR	optical character recognition
PCA	principal components analysis
V2I	vehicle to infrastructure
RANSAC	random sample consensus
RBF	radial basis function
ROI	region of interest (englisch für interessante Region)
RS	random sampling
SAT	satisfiability
SLAM	simultaneous localization and mapping
StVO	Straßenverkehrs-Ordnung
TP	true positive
SVM	Support-Vektor-Maschine
TPCB-RS	track-pruned class-biased random sampling
TPCB-US	track-pruned class-biased uncertainty sampling
TPCB-ADR	track-pruned class-biased angular diversity ranking
Tof	time of flight
US	uncertainty sampling
VwV-StVO	Verwaltungsvorschrift zur Straßenverkehrs-Ordnung
Wvz	Wechselverkehrszeichen

Konventionen

Skalare	klein, nicht fett, kursiv: x, y, \dots
Vektoren	klein, fett, nicht kursiv: $\mathbf{x}, \mathbf{y}, \dots$
Mengen	groß, nicht fett, kursiv: X, Y, \dots
Matrizen	groß, fett, nicht kursiv: $\mathbf{X}, \mathbf{Y}, \dots$

Zur Verbesserung der Lesbarkeit werden Mengenklammern nur für Mengen einer Kardinalität von zwei oder mehr geschrieben; für Mengen geringerer Kardinalität werden sie ausgespart, wenn aus dem Kontext ersichtlich ist, dass es sich um eine Menge handelt.

In Konfusionsmatrizen wird die (absolute oder relative) Häufigkeit von Klassifikationsergebnissen eingetragen. Pro Klasse wird jeweils eine Spalte und eine Zeile verwendet. Ein Eintrag m_{ij} in der Konfusionsmatrix \mathbf{M} gibt die Häufigkeit an, mit der Objekte der Klasse in Spalte j als Objekte der Klasse in Reihe i klassifiziert werden. Freie Einträge in der Matrix stehen für die Häufigkeit 0. Beispielsweise kann der Konfusionsmatrix

	ω_0	ω_1	ω_2
ω_0	23		3
ω_1		42	
ω_2			7

entnommen werden, dass 42 Objekte der Klasse ω_1 korrekt als ω_1 , 3 Objekte der Klasse ω_2 als ω_0 und keine Objekte der Klasse ω_0 als ω_1 klassifiziert wurden.

Symbole und Operatoren

\mathbf{x}	Vektor \mathbf{x} mit Komponenten $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$
$\mathbf{M}^{m,n}$	Matrix mit m Zeilen und n Spalten und Komponenten (m_{ij})
$\mathbf{M}(a : b ; c : d)$	Blockmatrix zu \mathbf{M}
2^X	Potenzmenge zu X
\emptyset	leere Menge
κ	Allgemeiner Bezeichner für Parameter
μ	Allgemeiner Bezeichner für Gewichte
δ_{ij}	Kronecker-Delta mit $\delta_{ij} := \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$
$\delta(A)$	Indikator-Funktion mit $\delta(A) := \begin{cases} 1, & A \text{ ist wahr} \\ 0, & \text{sonst} \end{cases}$
$P(\mathbf{x})$	Wahrscheinlichkeitsmaß
$P(\mathbf{x} \mathbf{y})$	Bedingte Wahrscheinlichkeit
$\mathcal{N}_p(\mathbf{x}, \mathbf{Y})$	Dichtefunktion der p -dimensionalen Normalverteilung mit Erwartungswert \mathbf{x} und Kovarianzmatrix \mathbf{Y}
$\mathbf{x}^{(t)}$	\mathbf{x} zum Zeitpunkt t
\mathbf{x}_W	\mathbf{x} im Weltkoordinatensystem

\mathbf{x}_B	\mathbf{x} im Bildkoordinatensystem
\mathbf{x}_S	\mathbf{x} im Sensorkoordinatensystem
\mathbf{x}_F	\mathbf{x} im Fahrzeugkoordinatensystem
\hat{x}	Schätzung von x
\dot{x}	Ableitung von x
$\text{sgn}(x)$	Vorzeichenfunktion mit $\text{sgn}(x) := \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases}$
$\exp(x)$	Exponentialfunktion mit $\exp(x) := e^x$
$ x $	Betrag von x
$ M $	Anzahl der Elemente der Menge M
$ \mathbf{M} $	Determinante der Matrix M
$\ \mathbf{x}\ $	Euklidische Norm von \mathbf{x}
$\langle \mathbf{x}, \mathbf{y} \rangle$	Skalarprodukt von \mathbf{x} und \mathbf{y}
$\binom{n}{k}$	Binomialkoeffizient
$\mathcal{O}(f)$	Funktionenklasse asymptotischer oberer Schranken (O-Kalkül, Landau-Symbol)
\overline{AB}	Strecke von A nach B
$\lfloor x \rfloor$	größte ganze Zahl $\leq x$
$\lceil x \rceil$	kleinste ganze Zahl $\geq x$
$x \gg y$	x ist deutlich (mehrere Größenordnungen) größer als y
$x \bmod y$	Rest der Division von x und y mit $x \bmod y = x - y \lfloor \frac{x}{y} \rfloor$
Θ^B	region of interest (ROI): Untermenge von Punkten im Bild B
B_{Θ}	kleinster rechteckiger Bildausschnitt von B , der Θ^B komplett enthält
$A * B$	Faltung von A und B
\ominus	Morphologische Erosion
\oplus	Morphologische Dilatation
\circ	Morphologisches Öffnen
\bullet	Morphologisches Schließen
\odot	Morphologischer Top-Hat-Operator
$\Omega = \{\omega_0, \dots, \omega_{n-1}\}$	Menge nominaler Klassen

1. Einführung

Zwei Drittel der deutschen Autofahrer empfinden die Beschilderung in Europa als schlecht verständlich; drei Viertel verfahren sich aufgrund einer zu großen Anzahl an Wegweisern. Viele holen sich Hilfe in Form von Navigationsgeräten an Bord, 80 % vertrauen bei widersprüchlichen Angaben dennoch den Verkehrszeichen [VA08]. Die Skepsis gegenüber den Angaben von Navigationsgeräten ist begründet, schließlich werden derzeit beispielsweise nur etwa 70 % der Tempolimits korrekt von ihnen erfasst [Jam+11]. Dynamische Informationen dargestellt durch Wechselverkehrszeichen werden nicht berücksichtigt. Autofahrer müssen einen nicht unerheblichen Teil ihrer Aufmerksamkeit der Erkennung der 650 unterschiedlichen Arten von Verkehrszeichen widmen. Durchschnittlich alle 25 Meter begegnet ihnen davon eines – 25 Millionen gibt es alleine in Deutschland [VA12, Seite 66].

Wie können intelligente Systeme einer zukünftigen Generation von Fahrerassistenzsystemen in diesen Situationen helfen? Sie müssen Verkehrszeichen und Lichtsignalanlagen erkennen, verstehen und in den Kontext der Fahrsituation richtig einordnen. Die vorliegende Arbeit widmet sich dieser Problemstellung und untersucht Methoden und Verfahren zur Erkennung und Interpretation solcher Verkehrselemente mit dem Ziel, relevante von nicht relevanten unterscheiden zu können. Diese Art der automatischen Erkennung kommt nicht nur Autofahrern zugute. Zukünftige Fahrzeuge werden für kurze oder längere Zeit das Fahren übernehmen oder in Kooperation mit dem Fahrer ausführen. Damit das nicht nur in aufwendig umgebauten Teststrecken funktioniert, muss das Fahrzeug in der Lage sein, mit der bestehenden Verkehrsinfrastruktur umzugehen. Für sicheres und den Verkehrsregeln entsprechendes Fahren muss beispielsweise die Vorfahrtregelung an Ampelkreuzungen erkannt und richtig interpretiert werden, was die Erkennung von Ampelzuständen einschließt. Durch Kartendienste bereitgestelltes Hintergrundmaterial muss ständig auf seine Korrektheit überprüft und um dynamische Informationen ergänzt werden, um Situationen wie Baustellen meistern zu können. Neben verbesserter Sicherheit und erhöhtem Fahrkomfort hilft die Automatisierung des Fahrens aktiv dabei, Energie zu sparen. Dies kann unsichtbar für den Fahrer geschehen, indem die rekuperative Bremse eines Elektrofahrzeugs vor einer Ampelkreuzung optimal eingestellt wird, oder teilautonom durch kurzzeitige Übernahme der Geschwindigkeitsregelung in der gleichen Situation. Durch konsequente Anwendung derartiger Fahr- und Betriebsstrategien kann ein signifikanter Beitrag zur Erhöhung der Reichweite geleistet werden.

In allen diesen Szenarien spielt die Erkennung und Interpretation von Verkehrselementen eine wichtige Rolle. Im Folgenden wird eine Forschungsfrage mit klar definierten Begrifflichkeiten vorgestellt, um aus den oben umgangssprachlich formulierten Problemstellungen zu lösende Teilprobleme abzuleiten und Rahmenbedingungen und Schwerpunkte der Arbeit zu definieren.

1.1. Problemstellung und Zielsetzung

Zur Förderung eines gemeinsamen Verständnis der im Titel der Arbeit – *Kontextsensitive Erkennung und Interpretation fahrrelevanter statischer Verkehrselemente* – genannten Begriffe werden diese zunächst näher erläutert.

Begriffsbildung “statisches Verkehrselement” In der Verkehrsdomäne treten statische und dynamische Objekte auf: Elemente der Verkehrsinfrastruktur und Verkehrsteilnehmer. Die statischen Elemente der Verkehrsinfrastruktur werden neben den Verkehrsflächen weiter unterteilt in die Verkehrsbeschilderung (inklusive Wegweisung), Fahrbahnmarkierungen, Leiteinrichtungen, Signalisierung (Lichtsignalanlagen beziehungsweise Ampeln) und Beleuchtung.

Die Unterscheidung zwischen statischen und dynamischen Elementen bezieht sich auf ihr inhärentes räumliches Verhalten. Statische Elemente sind ortsfest, können aber ein dynamisches Verhalten in Form unterschiedlicher Zustände an den Tag legen. Beispiele dafür sind Lichtsignalanlagen und Wechselverkehrszeichen. Dynamische Elemente sind beweglich; sie können innerhalb eines Beobachtungszeitraums aber auch ein statisches Verhalten an den Tag legen – wartende Fußgänger etwa oder parkende Fahrzeuge.

Begriffsbildung “Fahrrelevanz” Eine Definition des Begriffs *Fahrrelevanz* findet sich bei Rösler [Rö09]: Fahrrelevanz wird verstanden als “die Bedeutung, die Elemente beziehungsweise Ereignisse der Verkehrssituation für die sichere Ausführung der Fahraufgabe ohne Schädigung der eigenen Person oder anderer Verkehrsteilnehmer haben.” Diese auf Sicherheitsaspekte fokussierte Sicht wird um die Berücksichtigung eines Ziels erweitert übernommen:



Definition 1.1. Als fahrrelevant werden die Elemente bzw. Ereignisse der Verkehrssituation bezeichnet, die für die sichere und zielgerichtete Ausführung der Fahraufgabe ohne Schädigung der eigenen Person oder anderer Verkehrsteilnehmer von Bedeutung sind.

In der ursprünglichen Definition von Rösler ist beispielsweise ein Vorwegweiser auf der Autobahn nicht fahrrelevant. Die erweiterte Definition 1.1 betrachtet ihn dagegen als fahrrelevant, wenn er einen Hinweis auf ein Zwischenziel enthält.

Begriffsbildung “Verkehrssituation”



Definition 1.2. Unter dem Begriff *Verkehrssituation* und seinem Synonym *Fahrsituation* wird die Menge aller Entitäten (konkrete und abstrakte Objekte), ihrer Eigenschaften und der Relationen zwischen ihnen innerhalb eines räumlich und zeitlich abgeschlossenen Bereichs verstanden.

Um zu entscheiden, ob zwei Entitäten gemäß Definition 1.2 zur gleichen Verkehrssituation gehören, muss eine räumliche und zeitliche Distanz festgelegt werden. An dieser Stelle wird dies bewusst für konkrete Anwendungen offen gelassen.

Begriffsbildung “Kontextsensitivität” Die Fahrrelevanz eines Vorwegweisers hängt wie oben beschrieben vom Fahrtziel ab. Verallgemeinert lässt sich feststellen, dass die Bedeutung eines Objekts nicht alleinig vom Objekt selbst abhängt, sondern im Zusammenhang mit anderen (konkreten oder abstrakten) Objekten zu interpretieren ist.



Definition 1.3. Unter dem Begriff *Kontext* wird die Menge aller Entitäten einer Verkehrssituation (siehe Definition 1.2) verstanden, die im Zusammenhang mit der Lösung einer Aufgabe beziehungsweise der Erfüllung eines Ziels stehen. Ein Verfahren beziehungsweise eine Anwendung verhält sich *kontextsensitiv*, wenn sie Kontextinformationen in ihrem Verhalten berücksichtigt oder wenn berechnete Ergebnisse von Kontextinformationen abhängen.

Ein kontextsensitives Fahrerassistenzsystem würde auf einen Vorwegweiser nur aufmerksam machen, wenn er im Zusammenhang mit der zu fahrenden Route steht. Ähnlich muss ein durch ein Zusatzzeichen *Bei Nässe* eingeschränktes Tempolimit nur berücksichtigt werden, wenn die Fahrbahn nass ist. Intern kann Kontextsensitivität genutzt werden, um etwa die Erkennung von Halteverboten nur im Zusammenhang mit Parkvorgängen auszuführen.

Begriffsbildung “Situationsinterpretation”



Definition 1.4. Eine *Situationsinterpretation* oder *Interpretation einer Verkehrssituation* leitet neue Informationen aus der Analyse von zwei oder mehr Entitäten einer Verkehrssituation ab.

Im Unterschied zur Umfelderkennung, bei der basierend auf Sensordaten eine Menge von Objekten und ihre Eigenschaften erkannt werden, betrachtet die Situationsinterpretation mehrere Entitäten und insbesondere ihre Eigenschaften und Beziehungen gleichzeitig.

Mit den eingeführten Begriffen kann die oben umgangssprachlich formulierte Problembeschreibung als Forschungsfrage mit fest definierten Begrifflichkeiten formuliert werden:

**Welche statischen Verkehrselemente sind in der
aktuellen Verkehrssituation fahrrelevant?**

Zur Beantwortung dieser Frage sollen im Rahmen dieser Arbeit geeignete Methoden und Verfahren zur Umfelderkennung und Situationsinterpretation entwickelt werden. Dafür werden zunächst zu lösende Teilprobleme identifiziert und Rahmenbedingungen festgelegt.

1.2. Rahmenbedingungen und Schwerpunkte der Arbeit

Um eine Entscheidung darüber treffen zu können, ob ein statisches Verkehrselement fahrrelevant ist, muss zunächst seine Existenz festgestellt werden. Dafür können verschiedene Informationsquellen

eingesetzt werden. Die beiden wichtigsten sind die Umfelderkennung und Kartendienste. Die Umfelderkennung verwendet Videokameras als Hauptsensor, um Informationen zu akquirieren, und ermittelt mit Hilfe von Methoden aus Bereichen wie Maschinensehen und Maschinellem Lernen sichtbare statische Verkehrselemente. Wichtige Teilprobleme sind die Segmentierung (Bildung von Objekthypothesen), die Klassifikation (Verifikation von Objekthypothesen, Typbestimmung sowie Lernen), das Tracking (Verfolgung von Objekthypothesen, Verifikation raum-zeitlicher Konsistenz) und die temporale Fusion (Verifikation zeitlicher Konsistenz von Zuständen und Zustandsänderungen). Kartendaten werden in Kombination mit Sensorik zur Lokalisation als Kartendienste eingesetzt, um Informationen über statische Verkehrselemente aus Hintergrundwissen (der sogenannten digitalen Karte) zu extrahieren. Ein wichtiges Teilproblem hierbei ist die Karteneinpassung (auch Kartenabgleich, Map Matching).

Potentiell fahrrelevante Verkehrselemente müssen nicht nur erkannt, sondern auch in den Kontext der Verkehrssituation eingeordnet werden. Zu dieser Situationsinterpretation zählt die Modellierung von Verkehrsregeln, die zur Interpretation von Relationen zwischen Objekten genutzt werden. Neben statischen Verkehrselementen müssen auch andere Objekte (konkret oder abstrakt, zum Beispiel Fahrspuren und Wetter) in der Interpretation berücksichtigt werden. Daraus ergeben sich die Teilprobleme der Erkennung dieser Informationen, der Informationsfusion und der Analyse von Eigenschaften und Relationen.

Die praktische Umsetzbarkeit der entwickelten Verfahren und Methoden ist ein wichtiger Teilaspekt der Arbeit. Sie werden als Funktionen von kognitiven Assistenzsystemen erprobt, die in Versuchsträgern zum Einsatz kommen. Solche Versuchsträger können seriennahe Automobile sein, die mit zusätzlicher Sensorik ausgestattet sind oder durch Umbaumaßnahmen für spezielle Einsatzmöglichkeiten wie das autonome oder teilautonome Fahren umgerüstet wurden. Da die entwickelten Verfahren und Methoden nicht auf Automobile beschränkt sein sollen, können auch andere Versuchsträger (etwa Fahrräder oder Segway Personentransporter) und sogar mit Mobilgeräten ausgestattete Menschen zum Einsatz kommen. Alle Einsatzszenarien spielen sich im Straßenverkehr ab. Die Verfügbarkeit von Sensorik zur Umgebungserfassung wird stets vorausgesetzt.

Als Forschungsarbeit ist die Umsetzung in Serienfahrzeugen nicht von primärem Interesse. Im Sinne der angewandten Forschung wird aber darauf geachtet, dass eine derartige Umsetzung mit vertretbarem Aufwand durchgeführt werden könnte. Insbesondere wird in den durchgeführten Experimenten Hardware verwendet, die in äquivalenter Form in Serienfahrzeugen verwendet wird oder mittelfristig dort zum Einsatz kommen könnte. Es werden online lauffähige Verfahren entwickelt, die mit realen Daten umgehen können.

Alle Fahrversuche werden im Großraum Karlsruhe in Baden-Württemberg durchgeführt. Entsprechend wird die deutsche Straßenverkehrs-Ordnung [Sta13] als Grundlage der zu berücksichtigenden Straßenverkehrsregeln sowie der Klassifikation von Verkehrselementen zugrunde gelegt.

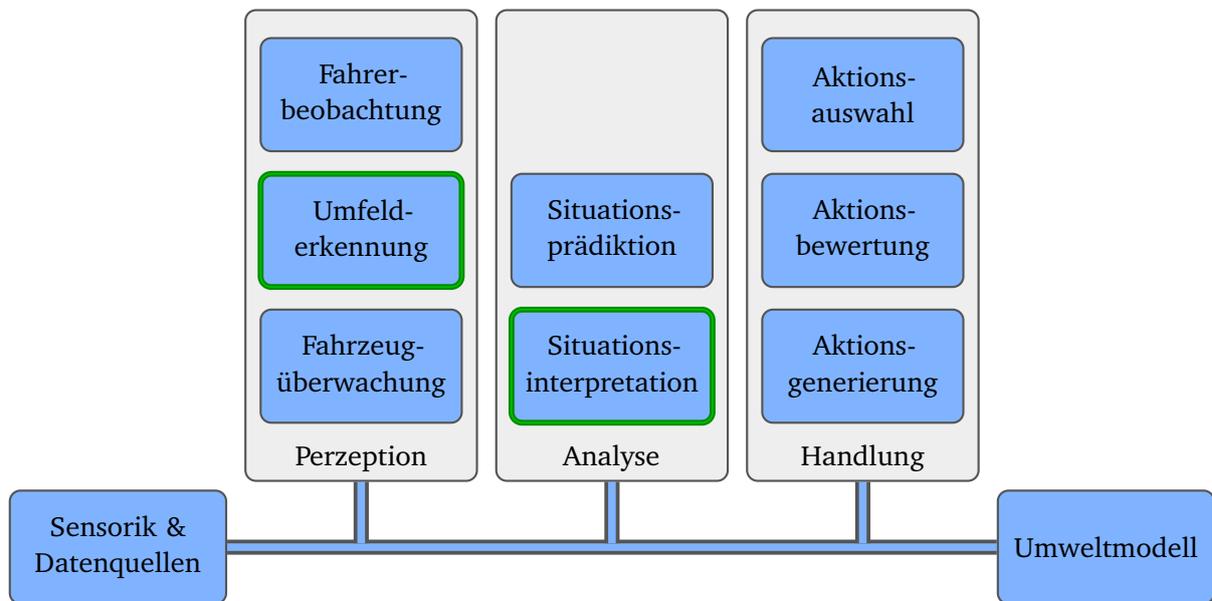


Abb. 1.1.: Komponenten kognitiver Assistenzsysteme. Verschiedene Sensoren dienen zur Wahrnehmung des Umfelds; Perception und Analyse der Daten führen zum Aufbau eines Umweltmodells gemäß der Wissensmodellierung. Die grün hervorgehobenen Komponenten Umfelderkennung und Situationsinterpretation werden im Rahmen der Arbeit schwerpunktmäßig behandelt.

1.3. Einordnung und Wissenschaftlicher Beitrag

Kognitive technische Systeme erfassen mit Hilfe verschiedener Sensoren ihre Umwelt. In Form von Assistenzsystemen für den Straßenverkehr interagieren sie durch passive und aktive Mensch-Maschine-Schnittstellen mit dem Fahrer oder steuern direkt die Aktorik des Fahrzeugs an. Mittels Perception, Lernen, Inferenz und Planen bilden sie kognitive Fähigkeiten und Kontrollmechanismen, um Verkehrssituationen wahrzunehmen, zu analysieren und angemessen zu handeln. Die in der vorliegenden Arbeit behandelten Schwerpunkte betten sich wie in Abbildung 1.1 dargestellt in kognitive Assistenzsysteme ein. Objekte im Umfeld, insbesondere Verkehrszeichen und Lichtsignalanlagen, werden erkannt und Relationen zwischen ihnen und anderen Entitäten im Rahmen der Situationsinterpretation analysiert, um zusätzliches Wissen abzuleiten. Die gewonnenen Informationen werden im Umweltmodell abgelegt und stehen Assistenzsystemen zur Verfügung, die beispielsweise als passive Fahrerassistenzsysteme auf relevante Elemente hinweisen können oder sie zur autonomen Fahrzeugsteuerung bei der Generierung von Fahraktionen berücksichtigen können.

Die Umfelderkennung stellt den sensornahen Teil dar, verantwortlich für die Erkennung statischer Verkehrselemente. Aufgrund ihrer charakteristischen Merkmale wird primär eine Farbvideokamera als Sensor verwendet. Die für die Erkennung zu lösenden Teilprobleme stammen aus den Bereichen Signalverarbeitung, Bildverarbeitung und Maschinelles Lernen. Wie in vergleichbaren Problemstellungen üblich, wird die Erkennungsaufgabe in die Teilaufgaben Segmentierung, Detektion, Merkmalstransformation, Klassifikation, Tracking und temporale Fusion aufgeteilt. Aufgrund der vielfältigen Merkmale statischer Verkehrselemente eignet sich nicht ein einzelnes Verfahren zur Segmentierung und Transformation aller Merkmale. Daher werden für einzelne Merkmale geeignete Verfahren entwickelt sowie

ihre Anwendbarkeit und Kombinationsmöglichkeiten betrachtet. Zur Klassifikation wird mit der hierarchischen Support-Vektor-Maschine (SVM) demgegenüber ein für alle Merkmale geeignetes Verfahren entwickelt. Seine Skalierbarkeit für eine seriennahe Umsetzung wird durch eine detaillierte Untersuchung im Themenfeld aktives Lernen aufgezeigt. Durch Kombination mit dem ebenfalls für alle Verkehrselemente einsetzbaren Tracking mittels Kalman-Filter und der temporalen Fusion für statische und semi-statische Elemente (ortsstatische Elemente mit sich zeitlich ändernden Zuständen) wird eine Verarbeitungskette aufgebaut, die sich durch die merkmalsabhängig flexibel auswechselbaren Komponenten zur universellen Erkennung statischer Verkehrselemente eignet.

Zur Analyse von Verkehrssituationen im Rahmen der Situationsinterpretation wird die Frage nach der Fahrrelevanz statischer Verkehrselemente in den Vordergrund gestellt. Sie baut auf den in der Umfelderkennung extrahierten Informationen auf und untersucht anhand typischer Problemstellungen, welche weiteren Kontextinformationen eingebracht werden müssen. Es werden geeignete Verfahren zur kontextsensitiven Informationsfusion und Relevanzbestimmung wie etwa Markov-Logik-Netze erarbeitet, welche die Kombination unsicherer Informationen aus der Perzeption mit sicherem Regelwissen (beispielsweise Verkehrsregeln) ermöglichen.

Damit wird erstmalig ein Ansatz zur kontextsensitiven Situationsinterpretation für Verkehrselemente entwickelt, der eine geschlossene Kette von der Aufnahme von Sensordaten bis hin zur Entscheidung über die Fahrrelevanz umsetzt. Wesentliche Beiträge im Bereich der Umfelderkennung sind die Entwicklung einer Auswahlstrategie (TPCB-ADR) im Bereich des aktiven Lernens und die Entwicklung der hierarchischen SVM zur Beschleunigung der Klassifikation. Die Entwicklung der kontextsensitiven Dempster/Shafer-Fusion zeigt die Nützlichkeit von Kontextinformationen zur Verbesserung der Informationsfusion auf. Mit Markov-Logik-Netzen wird ein Ansatz des statistisch relationalen Lernens auf die Situationsinterpretation übertragen und seine Anwendung in mehreren Problemstellungen unter anderem zur Bestimmung der Fahrrelevanz von Ampeln gezeigt. Alle Konzepte und Methoden werden durch umfangreiche Experimente validiert.

1.4. Aufbau der Arbeit

Die vorliegende Arbeit besteht aus drei Blöcken in vier Kapiteln. Eine Beschreibung von für die Arbeit relevanten verwandten wissenschaftlichen Arbeiten erfolgt im Stand der Technik in Kapitel 2. Die entwickelten Konzepte und Methoden werden in den Kapiteln zur Umfelderkennung (Kapitel 3) und Situationsinterpretation (Kapitel 4) vorgestellt. Kapitel 5 beschreibt die Validierung, für die verschiedene Experimente durchgeführt wurden.

Stand der Technik In Kapitel 2 werden relevante Themenfelder identifiziert und eine Kategorisierung verwandter Arbeiten entwickelt, anhand der eine Kurzzvorstellung von etwa 80 wissenschaftlichen Veröffentlichungen durchgeführt wird. Ausgewählte Einzelarbeiten sowie Forschungsgruppen werden anschließend im Detail vorgestellt und offene Fragestellungen diskutiert.

Umfeldererkennung Es werden Verfahren zur Segmentierung, zur Detektion, zum Tracking, zur Klassifikation und zur temporalen Fusion beschrieben. Aufgrund der intensiven Forschungsarbeit im vergangenen Jahrzehnt bilden aus der Literatur bekannte Verfahren hierfür eine sehr gute Ausgangsbasis. Geeignete Verfahren werden ausgewählt und gemäß der Anforderungen der Arbeit an eine echtzeitnahe Verarbeitung auf Standardrechnern unter Verwendung von kostengünstiger Sensorik angepasst und erweitert. Damit entstehen robuste Verfahren, die für die Situationsinterpretation geeignete Ergebnisse liefern. Nennenswerte Beiträge sind die Einführung der hierarchischen SVM (eine Mischung aus einer SVM und einem Entscheidungsbaum zum effizienten Umgang mit Mehrklassenproblemen) und die Entwicklung einer neuartigen Auswahlstrategie im Rahmen des aktiven Lernens. Ein Teil der entwickelten Methoden und Verfahren vor allem im Bereich der Segmentierung und Merkmalstransformation ist speziell auf charakteristische Eigenschaften bestimmter Verkehrselemente zugeschnitten, andere wie etwa die Klassifikation und das Tracking können generisch eingesetzt werden. Eine modulare Prozesskette wird eingeführt, um die generischen Komponenten gemeinsam zu nutzen und durch Austausch merkmalspezifischer Komponenten verschiedene Anwendungen zu realisieren:

- Die Erkennung von Ronden muss kreisförmige Verkehrszeichen detektieren und mit einer großen Anzahl unterschiedlicher Klassen umgehen können. Zur Detektion eignet sich die von Barnes und Zelinsky [BZ04] vorgestellte Fast Radial Symmetry; der effiziente Umgang mit einer Vielzahl von Klassen wird durch die Einführung der hierarchischen SVM erreicht. Teile dieser Anwendung wurden in den Publikationen [NZG07; NGZ08; NZ13] vorgestellt.
- Vorwegweiser zeichnen sich durch ihre Größe und einen einfarbigen Hintergrund aus, so dass ein Detektionsverfahren basierend auf Farbmerkmalen eingesetzt werden kann. Nach einer Unterscheidung von Text- und Sinnbildregionen kann die Erkennung von Texten über Standard-OCR-Verfahren und die Erkennung von Sinnbildern über eine SVM durchgeführt werden. Die vorgestellte Anwendung zur Erkennung von Vorwegweisern basiert auf der Arbeit [XNZ09].
- Zur Erkennung von Zusatzzeichen müssen kleine Objekte detektiert werden, die in einer Vielzahl verschiedener Typen vorkommen. Hier wird ausgenutzt, dass Zusatzzeichen immer in der Nähe von anderen Verkehrselementen angebracht werden, so dass die Suchregion entsprechend eingeschränkt werden kann. Eine angepasste Hough-Transformation und eine hierarchische SVM werden zur Detektion und Klassifikation eingesetzt. Diese Anwendung wurde in [Nie+10] publiziert.
- Durch ihre Aufstellung in Innenstädten und vergleichsweise kleine Größe stellen Ampeln eine Herausforderung für die Detektion dar. In der temporalen Fusion müssen Zustandsübergänge zwischen den verschiedenen Ampelfarben behandelt werden. Das vorgestellte Verfahren zur Erkennung von Ampeln basiert auf den Publikationen [Dre+10; NDZ10].

Situationsinterpretation Die Verfahren zur Situationsinterpretation werden vor dem Hintergrund der Forschungsfrage zur Ermittlung der Fahrrelevanz von statischen Verkehrselementen entwickelt. Um mit der durch den Einsatz von Sensorik und Verfahren zur Umfeldererkennung inhärenten Unsicherheit umzugehen, werden probabilistische Ansätze zu Grunde gelegt.

- Die Fusion mit einem Fahrspurerekenner (Abschnitt 4.1) wertet räumliche Relationen zwischen

Verkehrszeichen und Fahrspuren aus, um die Gültigkeit von Verkehrszeichen für Fahrspuren zu bestimmen. Im Sinne des Verzichts auf teure Sensorik wird das Verfahren rein bildbasiert umgesetzt und die wesentliche Herausforderung ist der Umgang mit fehlenden Distanzinformationen. Sie werden geschätzt und Konfidenzen für die Position von Verkehrszeichen und Fahrspuren im Fahrzeugkoordinatensystem werden aufgestellt, die wiederum zur Berechnung von Zuordnungen maximaler Konfidenz genutzt werden. Diese Anwendung entstand in Zusammenarbeit mit Thomas Gumpp [NGZ08].

- Die in Abschnitt 4.2 präsentierte kontextsensitive Fusion zur Bestimmung des gültigen Tempolimits basiert auf den Veröffentlichungen [WNZ08; GNL09; NGZ09]. Sie nutzt einen bekannten Ansatz zur Informationsfusion, die Dempster/Shaffer-Fusion, um Umfelderkennung und Kartendienste zu fusionieren. Neuartig ist die Nutzung des sogenannten Situationskontexts, mit dessen Hilfe die Zuverlässigkeit der Informationsquellen geschätzt wird und ihre Massefunktionen adaptiert werden. Dadurch werden Fehler in typischen Konfliktsituationen wie etwa Baustellen verhindert.
- Mit Markov-Logik-Netzen wird in Abschnitt 4.3 die Übertragung eines Verfahrens des statistischen relationalen Lernens gezeigt. Durch die Verbindung von Unsicherheit und komplexen relationalen Strukturen eignet es sich als allgemeines Framework zur Situationsinterpretation. Die Anwendung von MLNs auf konkrete Problemstellungen zur Erkennung von Baustellen, zur Bestimmung der Gültigkeit von Zusatzzeichen und zur Bestimmung der Fahrrelevanz von Ampeln (letztere in [NGZ11] publiziert) wird vorgestellt.

Experimente und Evaluation Zur Evaluation der in den Bereichen Umfelderkennung und Situationsinterpretation entwickelten Konzepte und Methoden wird eine Vielzahl von Experimenten durchgeführt. Sie validieren den Einsatz der Verfahren zur Erkennung deutscher Verkehrselemente, die Berücksichtigung der Rahmenbedingungen in Bezug auf kostengünstige Sensorik und Online-Verarbeitung auf Standard-Hardware, die Nützlichkeit der gewonnenen Informationen zur Entwicklung von Assistenzsystemen und die Eignung der Verfahren zur Bestimmung der Fahrrelevanz von Verkehrselementen.

- Die Evaluation der Erkennung von Ronden (kreisförmige Verkehrszeichen), Warnhinweisen (dreieckige Verkehrszeichen), Vorwegweisern, Zusatzzeichen und Ampeln belegt die Einsetzbarkeit der modularen Verarbeitungskette und der in ihr verwendeten Komponenten für die wichtigsten Arten deutscher Verkehrselemente.
- Ein sehr großer Datensatz wird verwendet, um die Leistung der zur Skalierung der Klassifikation entwickelten Methoden zu evaluieren. Im Bereich des aktiven Lernens wird eine neu entwickelte Auswahlstrategie mit anderen Auswahlstrategien verglichen. Die Leistung der hierarchischen SVM wird der Leistung einer Standard-SVM gegenübergestellt.
- Die Leistung der Methoden der Situationsinterpretation zur Bestimmung der Fahrrelevanz von Verkehrselementen wird durch Experimente zur Zuordnung von Verkehrszeichen zu Fahrspuren, zur Bestimmung des gültigen Tempolimits mittels kontextsensitiver Dempster/Shaffer-Fusion und zur Bestimmung der fahrrelevanten Ampeln durch Inferenz auf einem MLN gezeigt.

2. Stand der Technik

Erste Forschungsarbeiten zur Erkennung von Verkehrszeichen wurden Anfang der 90er Jahre vor allem im Rahmen des EUREKA-PROMETHEUS-Projekts (PROgramme for a European Traffic of Highest Efficiency and Unprecedented Safety) veröffentlicht [Rit92; Pri+93; PR93; Jan+93; Est+94; Pic+94; ZRJ94; AKA96; AA96; Fra+97]. Die aus heutiger Sicht (2013) stark eingeschränkte Rechenleistung limitierte die Auswahl echtzeitfähiger Algorithmen sehr. Durch den hohen Aufwand war ein Serieneinsatz nicht realisierbar und die Forschungsaktivitäten gingen in den folgenden zehn Jahren deutlich zurück. Der technische Fortschritt vor allem durch Verbesserungen der Sensorik, der Miniaturisierung von Computern und ständige Erhöhungen der Rechenleistung aber verbesserten die Rahmenbedingungen stetig. Dies führte schließlich zu steigendem industriellen Interesse an kognitiven Fahrerassistenzsystemen. Parallel dazu stiegen die Forschungsaktivitäten seit 2005 merklich an, wie an der Anzahl der Veröffentlichungen zu den verschiedenen Themenbereichen abgelesen werden kann. Da sich durch die deutlichen Verbesserungen der Hardware auch die Auswahl möglicher Algorithmen stark gewandelt hat, liegt der Fokus im Folgenden auf der Betrachtung von Forschungsarbeiten seit 2005. Weiterhin relevante ältere Arbeiten werden ebenfalls vorgestellt. Um den Überblick über das breite Forschungsfeld zu erleichtern, werden zunächst Themenfelder identifiziert und verschiedene Kategorisierungen verwandter Arbeiten eingeführt. Danach wird ein Überblick über etwa 80 aktuelle Forschungsarbeiten gegeben. Die Arbeiten größerer Forschungsgruppen sowie wichtige Arbeiten einzelner Autoren werden anschließend im Detail vorgestellt und diskutiert.

2.1. Themenfelder

Einige Veröffentlichungen zur Erkennung von Verkehrselementen widmen sich nur speziellen Teilaspekten des Erkennungsvorgangs, etwa der Segmentierung [LB04; LA+05; GJ+05; GJ+07; KBA11], der Klassifikation [SHW05; PND06; Ove11; HW12] oder dem Tracking [LA+07]. Viele Arbeiten jedoch stellen in sich abgeschlossene Systeme dar, die eine konkrete Erkennungsaufgabe lösen. Die Zeichen-erkennung wird zur Inventarisierung bzw. Kartierung vorhandener Zeichen genutzt [SPJ04; LA+07; McL+08; Seg+10]. Zur Unterstützung von Menschen mit Mobilitätseinschränkungen kommt die Erkennung von Ampeln zum Einsatz [Shi+02; SHW05; KKY07]. Touristen werden durch die automatische Übersetzung von Texten auf Verkehrszeichen unterstützt [Che+04; Bou+11]. Passive Fahrerassistenzsysteme präsentieren dem Fahrer Verkehrszeichen auf verschiedene Arten. Um dabei die Überlastung des Fahrers durch zu viele Informationen zu vermeiden, schlagen einige Autoren eine Filterung anhand der Sichtbarkeit vor [STB09; DDT11].

Der mit Abstand verbreitetste Sensor ist die monokulare Farbvideokamera. Einige Autoren verwenden un-

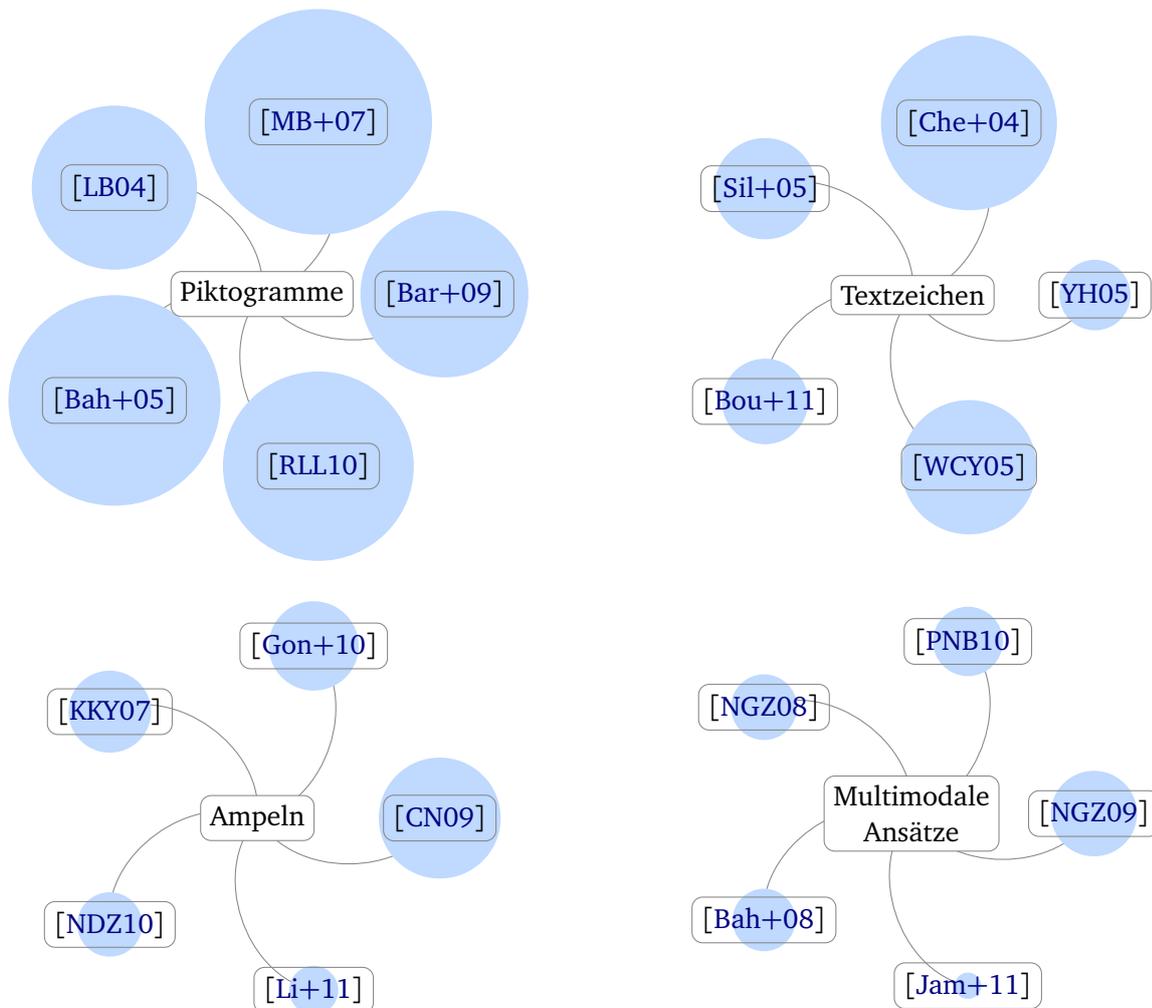


Abb. 2.1.: Veröffentlichungen zur Erkennung von statischen Verkehrselementen und ihre altersgewichtete Anzahl an Zitaten (AAZ) mit $AAZ := c/a$ dargestellt durch die Kreisgröße. Dabei ist c die Anzahl der Zitate am 07. Februar 2013 laut Google Scholar [Goo13] und a das Alter der Publikation in Jahren zu diesem Zeitpunkt.

terstützend Entfernungsmessungen durch eine zweite Kamera als Stereosystem [LK04; Eck07; McL+08; Bar+09]. Daneben gibt es Experimente mit aktiven Kameras, die Verkehrszeichen bei der Vorbeifahrt verfolgen [MKS00; Gu+10; GYT11] oder Infrarot-Kameras zur Ausnutzung der Retroreflexivität von Verkehrszeichen [McL+08]. Schließlich werden GPS-Sensoren eingesetzt, um über die Kenntnis der eigenen Position eine Fusion mit Kartendiensten durchzuführen [GWF03; Bah+08; LBB08; McL+08; Bar+10].

Die Forschungsaktivität ist insbesondere im Bereich der Erkennung piktogrammbasierter Verkehrszeichen hoch, wie der Analyse von Querverweisen in Forschungsarbeiten (siehe Abbildung 2.1) entnommen werden kann. Dies liegt unter anderem an der industriellen Nachfrage an Systemen zur Erkennung von Tempolimits, die 2008 im ersten Serieneinsatz eines solchen Systems in der BMW 7er Serie mündete. Die Entwicklung des Systems wurde von der israelischen Firma Mobileye [Mob] durchgeführt, eigenen Angaben zufolge das weltweit größte Entwicklungszentrum für Maschinensehen. Die Funktionalität wurde sukzessive um die Erkennung von Überholverbots und bestimmten Zusatzzeichen ergänzt und

auf eine größere Anzahl unterstützter Länder erweitert. Details zur verwendeten Methodik sind nicht in Form von wissenschaftlichen Arbeiten veröffentlicht. Die Angaben in Patenten sind unspezifisch [Ste+08; Ste+11]. Andere Fahrzeughersteller wie Mercedes, Audi, Opel, Saab und Volkswagen zogen in den kommenden Jahren mit dem gleichen oder ähnlichen Funktionsumfang nach. Neben Eigenentwicklungen und dem von Mobileye angebotenen System arbeiten Automobilzulieferer an vergleichbaren Systemen, etwa Bosch [Jan03; Jan04], Siemens [Bah+06; Bah+05; Bah+08; Kel+08], Hella [Bec09], Continental [BG10] und Harman [NM11; WCH13; YCW10]. Die Vermarktung der Systeme führte dazu, dass der Begriff *Verkehrszeichenerkennung* im Sprachverständnis häufig mit der Erkennung von Tempolimits gleichgesetzt wird, obwohl der Bereich der Verkehrszeichenerkennung ein viel größeres Spektrum abdeckt.

Ein weiteres Themenfeld ist die Erkennung von Textzeichen insbesondere auf Vorwegweisern, die durch die Verwendung von hochauflösenden Kameras ermöglicht wird [MKS00; WCY05; AJ06]. Seit einigen Jahren ist zudem eine verstärkte Aktivität auf dem Gebiet der Ampelerkennung festzustellen [LK04; Eck07; KKY07; CN09; SO09; Li+11; SSD12; CGL13]. Anwendung findet sie unter anderem im Komfortbereich und zur Verbesserung der Energieeffizienz. Zum autonomen Fahren ist diese Funktionalität sogar eine Voraussetzung.

Erste Arbeiten zum Einsatz von Mobilgeräten [SPJ04; LY10; Bou+11], die Kamera und Verarbeitungseinheit kombinieren und nicht fest verbaut werden, deuten einen möglichen Trend zukünftiger Forschungsarbeiten an. Aufgrund der Einsetzbarkeit in beliebigen Fahrzeugen und der oft bereits vorhandenen kostengünstigen Hardware ist diese Lösung für Anwender besonders attraktiv. Für Erkennung und Situationsinterpretation ergeben sich neue Herausforderungen durch die in Mobilgeräten verbauten stark unterschiedlichen Kamerasensoren sowie die ständig wechselnde Ausrichtung des Geräts im Fahrzeug.

Neben der bislang diskutierten rein bildbasierten Erkennung von Verkehrselementen entstanden einige Arbeiten zur Informationsfusion mit Kartendiensten [LK04; Bah+08; LBB08; Bar+10; PNB10; DL11; DLB11]. Ziel dieser Ansätze ist die Kombination der Vorteile der jeweiligen Informationsquellen zu einem insgesamt robusteren System.

2.2. Überblick über Forschungsarbeiten

Trotz einer hohen Gesamtanzahl an Veröffentlichungen im Bereich der Umfelderkennung gibt es nur wenige Überblicksstudien über das Themenfeld: Die Studie von Fleyeh und Dougherty [FD05] wurde bereits 2005 veröffentlicht und enthält daher keine aktuellen Arbeiten. Fu und Huang [FH10] präsentierten im Jahr 2010 einen aktuelleren, aber nicht sehr umfangreichen Überblick. Schließlich gibt es eine Studie von Mogelmoose, Trivedi und Moeslund [MTM12] aus dem Jahr 2012, die zwar ein breites Spektrum an Arbeiten erfasst, sich allerdings nur mit Detektion und Tracking auseinandersetzt.

Um dennoch einen Überblick über das breite Themenspektrum zu erhalten, wird in diesem Abschnitt eine Übersicht über etwa 80 verwandte Arbeiten gegeben. Um sie schnell einordnen zu können, wird eine Kategorisierung verwendet, die das Jahr der Veröffentlichung, eine Typisierung sowie die Länder (Länderkürzel nach ISO 3166) angibt, in denen die in Experimenten verwendeten Daten aufgenommen

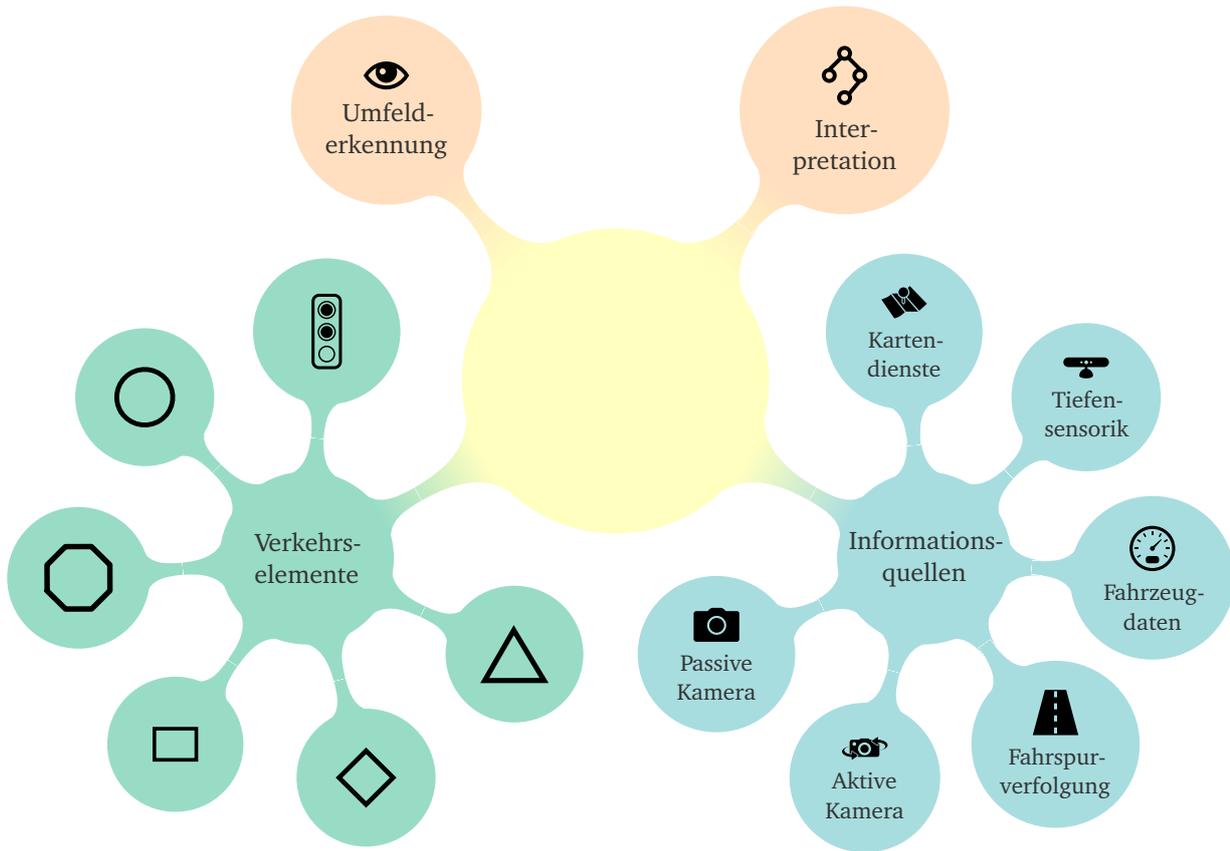
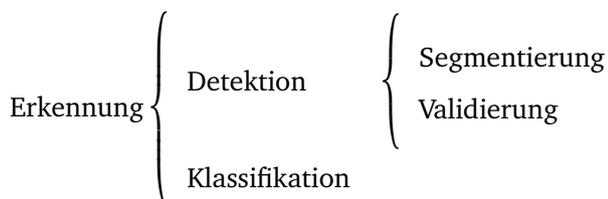


Abb. 2.2.: Schema zur Typisierung verwandter Forschungsarbeiten. Die Darstellung verwendet die Symbole [Ré13; Ste13; Vol13; Mic13a; Chr13; Mic13b; Jas13; Art13].

wurden. Die Typisierung basiert auf Abbildung 2.2 und unterscheidet zwischen Arbeiten zur Umfelderkennung und zur Interpretation. Die verwendeten Informationsquellen sowie die betrachteten Arten von Verkehrselementen werden angegeben. Arbeiten der gleichen Forschungsgruppe (siehe Abschnitt 2.3) sind mit identischen Hintergrundfarben gekennzeichnet. Neben der Kategorisierung jeder Arbeit wird das Ziel angegeben und das methodische Vorgehen kurz beschrieben. Auf Angaben zur Leistung (in Bezug auf Laufzeit oder Erkennungsraten) der Ansätze wird bewusst verzichtet, da durch unterschiedliche Länder, Sensorik, Verarbeitungseinheiten und Datensätze keine Vergleichbarkeit gewährleistet ist. Die Begriffe Segmentierung (englisch *segmentation*), Detektion (englisch *detection*) und Erkennung (englisch *recognition*) werden in verwandten Arbeiten nicht immer unterschieden. In dieser Arbeit wird folgende Bedeutung benutzt:



Wie dem Überblick über verwandte Forschungsarbeiten in Tabelle 2.1 entnommen werden kann, sind fast alle Publikationen im Bereich der Umfelderkennung angesiedelt; nur wenige widmen sich der Situationsinterpretation. Einer der Hauptgründe dafür ist vermutlich die Abhängigkeit der Situationsinterpretation von der Umfelderkennung: Erst wenn für alle zu betrachtenden Objekte der Szene robust funktionie-

rende Perzeptionskomponenten entwickelt wurden, können Algorithmen zur Situationsinterpretation getestet werden. Da die Veröffentlichung von Datensätzen oder Implementierungen von Algorithmen noch eine große Ausnahme darstellt, beginnt jedes Forschungsteam von vorne mit der Entwicklung.

Im Bereich der Segmentierung werden Farb-, Kanten- und Texturmerkmale eingesetzt. Einigkeit herrscht darüber, dass Farbmerkmale stark vom Umgebungslicht beeinflusst werden, das im Verkehrsbereich beeinflusst von Sonnenlicht, Schweinwerfern, Blinkern, Tunneln und Straßenlaternen besonders großen Schwankungen unterliegt. Da kein robustes Verfahren bekannt ist, Lichtquellen und ihre Farben in Kamerabildern automatisch zu ermitteln, versuchen einige Autoren über einen Weißabgleich die farbverfälschenden Effekte von Lichtquellen zu eliminieren [Bro+07; Zhe+12]. Dafür müssen allerdings sehr einschränkende Annahmen getroffen werden, etwa dass die mittlere untere Bildregion graue Straße zeigt. Andere Ansätze versuchen, durch eine Transformation in einen anderen Farbraum robustere Ergebnisse zu erzielen. Für den Fall von durch Umgebungslicht verfälschten Farben hilft dieses Vorgehen jedoch nicht. Diese Erkenntnis spiegelt sich über den betrachteten Zeitraum durch einen Trend weg von Farbmerkmalen hin zu Kantenmerkmalen wider: Wurden Farbmerkmale zunächst noch häufig als primäres Merkmal verwendet, so werden sie in aktuellen Arbeiten nicht benutzt oder nur unterstützend eingesetzt. Kantenmerkmale werden oft in Verbindung mit Varianten der Hough-Transformation verwendet. Eine Reihe wichtiger Veröffentlichungen dazu beschreibt die Fast Radial Symmetry [BZ04; LB04; BL06] als robuste und echtzeitfähige Methode zur Detektion von Kreisen bzw. regelmäßigen Polygonen, die in vielen Arbeiten eingesetzt wird [SD06; PC07; Kel+08; NKMS08; Bar+09; HZ09; Gu+10; RLL10; Liu+11; SSD12; Zhe+12]. Auch das von Dalal und Triggs [DT05] ursprünglich zur Fußgängererkennung vorgeschlagene HOG-Merkmal basiert auf Kanten. Es findet im Bereich der Erkennung von Verkehrselementen sowohl zur Segmentierung [Ale+07; TZG09; Liu+11; ZS11a] als auch zur Klassifikation [Ove11; PMN13] Anwendung. In beiden Bereichen verbreitet eingesetzt werden auch Varianten des Viola/Jones-Verfahrens [VJ01] zur Objekterkennung [LK04; Bah+05; Kel+08; Bar+09; TZG09; Gu+10; LVPLAC10; Seg+10; YCW10; Che+11; Deg+11]. Darüber hinaus werden zahlreiche weitere Lernverfahren zur Klassifikation eingesetzt, etwa SVMs [LA+05; GJ+07; Kou07; MB+07; TZG09; YCW10; Liu+11; GG+11; PMN13], neuronale Netze [TB04; LK04; Bro+07; Mou+07; HZ09; Jia+11], Boosting [Ove11], Bayes-Klassifikatoren [Kou07] oder Random Forests [ZS11b].

Arbeiten im Bereich der Situationsinterpretation gibt es bislang nur wenige. Sie beschäftigen sich mit der Sichtbarkeit von Verkehrszeichen [STB09; Dom+10; DDT11] und der Gültigkeit von Tempolimits [Bar+10; Meu+10; PNB11].

Kategorisierung	Ziele und methodisches Vorgehen
2000 [MKS00] ○ □ 👁 📷 JP	<i>Erkennung von Verkehrszeichen.</i> YUV-Schwellwertverfahren zur Farbsegmentierung. Validierung von ROIs per Hough-Transformation für Kreise bzw. einen Histogramm-Ansatz für Linien und Analyse der Ergebnisse für Rechtecke. Ausrichtung der Telekamera auf die prädiizierte Position eines gefundenen Verkehrszeichens. Template Matching zur Klassifikation kreisförmiger Zeichen. Grauwerttransformation, adaptives Schwellwertverfahren und Histogramm-Ansatz zur Segmentierung von Schriftzeichen und Symbolen auf Vorwegweisern. Template Matching zur Klassifikation von Schriftzeichen und Abgleich mit Wörterbuch.

Kategorisierung	Ziele und methodisches Vorgehen
2002 [LLX02] ○   CN	<i>Detektion von Verkehrszeichen.</i> Filterung des Kantenbildes durch HSI-Schwellwertregeln. Genetischer Algorithmus zur Kreisdetektion. Einfaches Bewegungsmodell zur Einschränkung des Suchbereichs.
2002 2004 [ST02; TB04] ○   NO	<i>Erkennung von Tempolimits.</i> RGB-Schwellwertverfahren zur Farbklassifikation. Nachbarschaftsanalyse und Template Matching zur Detektion von Tempolimits. Regelbasierte Segmentierung der ersten Ziffer, Binarisierung, Klassifikation mit neuronalem Netz.
2004 2008 [BZ04; BZF08] ○   AU	<i>Erkennung von Tempolimits.</i> Fast Radial Symmetry zur Segmentierung, Größennormalisierung und Kreuzkorrelation mit Referenzmodellen zur Klassifikation.
2004 [LK04]      DE	<i>Erkennung von Ampeln.</i> Vergleich von Farbsegmentierung, Hough-Transformation für Kreise und Matched Filter zur Segmentierung von Ampellichtern. Klassifikation mit neuronalem Netz. Tracking und temporale Fusion durch Mehrheitsentscheidung. Alternative Implementierung über Viola/Jones-Verfahren.
2004 [LB04; Bar05; BL06] △ ◇ ○ ○   AU	<i>Erkennung von Verkehrszeichen.</i> Generalisierung der Fast Radial Symmetry für Kreise zur Extended Fast Radial Symmetry für n-Ecke. Berücksichtigung von Einschränkungen des Rotationswinkels zur Verbesserung der Performanz (Laufzeit und Robustheit).
2005 [Bah+05] ○   DE	<i>Erkennung von Tempolimits.</i> Viola/Jones-Verfahren mit Haar-Wavelets erweitert um Farbinformationen. Merkmalsreduktion durch LDA. Typbestimmung durch Nächster-Nachbar-Klassifikation zu durch Gauß-Mixturen repräsentierten Referenzmodellen. Tracking anhand eines einfachen Bewegungsmodells. Temporale Fusion durch altersgewichtete Konfidenzbildung.
2005 2007 2008 [GJ+05; LA+05; VR05; GJ+07; GJ+08] △ ◇ □ ○   ES	<i>Detektion von Verkehrszeichen.</i> HSI-Schwellwerte, Zusammenhangskomponenten und Randbestimmung. Mittelpunktbestimmung von ROIs und Extraktion der Abstände zum Rand als Merkmale. Nutzung der FFT, um rotationsinvariante Merkmale zu erhalten. Projektive Abbildungen zu Basisformen zur Normalisierung. Klassifikation der Form über minimale Differenz zu Referenzmodellen bzw. lineare SVM.
2005 [Shn05] ◇ □ ○   US	<i>Erkennung von Verkehrszeichen.</i> Schwellwertregeln im RGB-Farbraum, Zusammenhangskomponenten. Regelbasierte Filterung der ROIs. Template Matching zur Klassifikation.
2005 [SHW05] ◇ □   US	<i>Klassifikation von Verkehrszeichen.</i> Manuell segmentierte Verkehrszeichenbilder als Eingabe. Farbklassifikation mittels SVM basierend auf HS-Farbhistogrammen als Merkmal. Formklassifikation durch Extraktion von Eckpunkten und Matching auf Referenzmodelle.

Kategorisierung	Ziele und methodisches Vorgehen
2005 [WCY05]   US	<i>Detektion von Text auf Vorwegweisern.</i> Clustering extrahierter Farbmerkmalen zu Eckpunkten. Filterung nicht senkrechter Ebenen basierend auf Punktkorrespondenzen in aufeinander folgenden Bildern. Region Growing auf Farbmerkmalen in der Umgebung von Kantenpixeln innerhalb von Richtungswegweisern zur Segmentierung von Textregionen.
2006 [AJ06]   KR	<i>Segmentierung von Pfeil- und Textregionen auf Vorwegweisern.</i> RGB-Schwellwerte, Zusammenhangskomponenten und Filterregeln zur Segmentierung von Richtungswegweisern. Segmentierung von Pfeilen über genetische Algorithmen und geometrische Analyse von Eckpunkten.
2006 [SD06]       GR	<i>Erkennung von Verkehrszeichen.</i> Schwellwertregeln im $L^*a^*b^*$ -Farbraum zur Bildung von vier Binärbildern als Eingabe für die Extended Fast Radial Symmetry zur Segmentierung. Tracking und Prädiktion zur Reduktion der Laufzeit. Kreuzkorrelation zu Referenzmodellen zur Klassifikation.
2007 [Ale+07]       AT	<i>Detektion von Verkehrszeichen.</i> histogram of oriented gradients (HOG)-ähnliche Merkmalsextraktion mit angepassten Zellgrößen. Klassifikation über Sliding Window Verfahren und Vergleich mit Referenzmodellen mit trainierten Gewichten.
2007 [Bro+07]      IT	<i>Erkennung von Verkehrszeichen.</i> RGB-Schwellwerte und Zusammenhangskomponenten zur Bildung von ROIs. Template Matching zur Klassifikation der Form. Neuronale Netze zur Typbestimmung.
2007 [Eck07]     DE	<i>Erkennung von Ampeln.</i> Hough-Transformation für Kreise zur Segmentierung von Ampeln. Verifikation durch Ermittlung des Ampelkastens und Schwellwertregeln für extrahierte Merkmale. Tracking mittels Kalman-Filter unterstützt durch Abstandsschätzung basierend auf Stereo-Kamerasystem.
2007 [KKY07]    US	<i>Erkennung von Ampeln.</i> Schwellwertregeln im RGB-Farbraum zur Segmentierung von Ampelfarben. Keine Angaben zur Generierung von ROIs. Regelbasierte Selektion, keine Validierung.
2007 [Kou07]     SG	<i>Klassifikation von Verkehrszeichen.</i> Nutzung manuell segmentierter Daten zum Vergleich von Random Forest mit anderen Klassifikatoren: SVMs, Bagging SVM und AdaBoost Naive Bayes. Experimente mit Farb- und Grauwertmerkmalen und PCA zur Merkmalsreduktion.
2007 2008 [LA+07 ; LA+08]     ES	<i>Tracking von Verkehrszeichen.</i> Tracking von Schildern mit Hilfe von Kalman-Filtern unter Berücksichtigung von Mehrfacherkennungen im gleichen Bild. Zuordnung über minimale Ähnlichkeit. Prädiktion in Weltkoordinaten unterstützt durch Odometrie und GPS. Temporale Fusion über altersgewichtete Konfidenzberechnung.
2007 [MB+07]       ES	<i>Erkennung von Verkehrszeichen.</i> Schwellwertregeln im HSI-Farbraum für farbige und im RGB-Farbraum für achromatische Schilder. Bildung von Zusammenhangskomponenten. Klassifikation der Form nach [GJ+07]. Klassifikation der Schildtypen mittels SVMs mit RBF-Kernel.

Kategorisierung	Ziele und methodisches Vorgehen
2007 [Mou+07; Bar+08] □ ○ 👁 📷 FR DE	<i>Erkennung von Tempolimits.</i> Hough-Transformation für Kreise zur Segmentierung von runden Tempolimits. Kantenbasierte Rechtecksegmentierung ohne nähere Angaben zum Verfahren für rechteckige (amerikanische) Tempolimits. Regelbasierte Segmentierung der Zahl im Binärbild und anschließende Ziffernsegmentierung über Zusammenhangskomponenten und Histogramm-Verfahren. Neuronales Netz zur Klassifikation von Ziffern.
2007 [PC07] △ ◇ ○ ○ 👁 📷 PT	<i>Erkennung von Verkehrszeichen.</i> Schwellwertregeln im HSV-Farbraum zur Segmentierung von blau und rot. Zusammenhangskomponenten und Filterregeln zur Bildung von ROIs. Klassifikation der Form über Fast Radial Symmetry (Kreis) und Harris Corner Detection und Regeln (Dreieck und Quadrat). Nicht erkannte Formen werden anhand der Farbklassifikation als Stopp-Schild klassifiziert oder abgelehnt.
2008 [Bah+08] ○ 👁 📷 🗺 DE	<i>Bestimmung des gültigen Tempolimits.</i> Fusion nach Bayes zur Ermittlung des gültigen Tempolimits. Berücksichtigung des Straßentypes bei der Generierung von Wahrscheinlichkeiten anhand des Kartendienstes.
2008 [Car+08] ○ 👁 📷 IT	<i>Detektion von Aufhebungszeichen.</i> Scanline-Verfahren auf Gradientenbild und Zustandsautomat zur Segmentierung des schwarzen Bandes von Aufhebungszeichen. Hough-Transformation für Kreise zur Verifikation von Kandidaten. Tracking in Bildkoordinaten über minimale Distanz.
2008 [Ham+08] ○ □ 👁 📷 FR DE	<i>Erkennung von Aufhebungs- und Zusatzzeichen.</i> Hough-Transformation für Kreise zur Segmentierung von Aufhebungszeichen und Verifikation über neuronales Netz. Segmentierung von Rechtecken unterhalb von Tempolimits (keine Angaben zum Verfahren) und Validation mittels neuronalem Netz zur Detektion von Zusatzzeichen.
2008 [Kel+08] □ 👁 📷 🕒 US	<i>Erkennung von Tempolimits.</i> Kantenbasierte Segmentierung aufrechter Rechtecke ähnlich zum Fast Radial Symmetry Verfahren. Anschließende ROI-Bildung durch Viola/Jones-Verfahren mit nachfolgender Rotationskorrektur. Klassifikation von Zahlen nach LDA-Merkmalreduktion über minimale Mahalanobis-Distanz zu Modellvektoren (Gauß-Mixturen). Tracking in Weltkoordinaten und temporale Fusion über Mehrheitsentscheidung.
2008 [LBB08] ○ 👁 📷 🕒 🗺 FR	<i>Verbesserung der Selbstlokalisierung.</i> Dempster/Shافر (D/S)-Fusion zur Verbesserung der Selbstlokalisierung.

Kategorisierung	Ziele und methodisches Vorgehen
2008 [McL+08]     IE	<i>Inventarisierung von Verkehrszeichen.</i> Stereo-Kameras mit einseitiger Infrarot-Beleuchtung zur Detektion von Verkehrszeichen unter Ausnutzung der Retroreflexivität. Schwellwertfilterung des linken Kamerabildes, Filterung des Himmels und Extraktion von Zusammenhangskomponenten. Regelbasierte Filterung und Korrespondenzsuche im rechten Bild durch Reflektionsmodell und Template Matching. Berechnung von 3D-Position und Transformation in Weltkoordinaten anhand der zeitlich geglätteten GPS-Position.
2008 [NKMS08]     DE	<i>Detektion von Verkehrszeichen.</i> Beschleunigung von Hough-Transformation oder Extended Fast Radial Symmetry durch Einschränkungen im Suchraum mittels Auswertung eines Symmetriemerkmals im Bild.
2008 [XK08]     UK	<i>Segmentierung von Verkehrszeichen.</i> Schwellwertregeln im LCH-Farbraum (CIE-CAM97) und Quadtree-Histogramm-Verfahren zur Segmentierung. Unterschiedliche Parameter abhängig vom Wetter (manuelle Einteilung der Bilder).
2009 [Bar+09]       ES	<i>Erkennung von Verkehrszeichen.</i> Viola/Jones-artiges Verfahren zur Detektion. Fast Radial Symmetry für Kreise bzw. Hough-Transformation für Dreiecke zur Verifikation und Bestimmung der Modellparameter. Klassifikation per Forest-ECOC (error correcting output codes). Bestimmung der Weltposition durch Stereoinformationen und Global Positioning System (GPS)/inertial measurement unit (IMU).
2009 [Kir+09]     ES	<i>Erkennung von Verkehrszeichen.</i> Segmentierung mittels Schwellwertregeln im HSI-Farbraum und Form- und Typklassifikation durch SVMs. Das Vorgehen entspricht im Wesentlichen dem von [LA+07].
2009 [Car+09]     IT	<i>Detektion von Verkehrszeichen.</i> Hough-Transformation für Linien und geometrische Analyse zur Segmentierung von Dreiecken und Rechtecken.
2009 [CN09]    FR	<i>Erkennung von Ampeln.</i> Morphologischer TopHat-Operator, Zusammenhangskomponenten und Verifikationsregeln zur ROI-Bildung. Adaptive Template-Matching zur Validierung.
2009 [HZ09]    DE	<i>Erkennung von Verkehrszeichen.</i> Kombination von Fast Radial Symmetry und SIFT-Merkmalen mit Nächster-Nachbar-Suche zu Referenzmodellen zur Kreissegmentierung. Aktive Konturen zur Verbesserung der SIFT-Ergebnisse. Neuronale Netze (multilayer perceptron (MLP)) auf SIFT- und SURF-Merkmalen zur Klassifikation von Tempolimits.
2009 [SO09]    US	<i>Erkennung von Ampeln.</i> Training von HS-Gaußverteilungen zur Modellierung von Ampelfarben. Binarisierung über Schwellwerte, Zusammenhangskomponenten und regelbasierte Filterung. Temporale Konsistenzprüfung.

Kategorisierung	Ziele und methodisches Vorgehen
2009 [STB09] ○   FR	<i>Sichtbarkeitsanalyse von Verkehrszeichen.</i> Betrachtung von Saliency als Klassifikationsproblem. Farbhistogramme und SVM zur Klassifikation der Saliency von Verkehrszeichen. Anwendung auf das gesamte Bild zur Bildung einer Saliency Map.
2009 [TZG09]    BE	<i>Erkennung und Positionsbestimmung von Verkehrszeichen.</i> Fahrzeug ausgestattet mit acht Kameras (zwei pro Seite). RGB-Schwellwertverfahren als Vorfilter gefolgt von Viola/Jones-Verfahren zur Segmentierung. Verifikation durch SVM basierend auf Farb-, HOG- und Haar-ähnlichen Merkmalen. Weitere SVMs zur Klassifikation der Schildart (Form und Farbe). Korrespondenzsuche auf Objektebene in benachbarten Kameras, Bestimmung der 3D-Position. Minimum Description Length zur Assoziation von 2D-Ansichten und 3D-Positionen. Abschließende SVM zur Typbestimmung.
2010 [Bar+10] ○     FR	<i>Bestimmung des gültigen Tempolimits.</i> Logik-Regeln für den Spezialfall von unterschiedlichen Tempolimits in der Nähe von Ausfahrten. Zustandsautomat und Kopplung mit Fahrspurverfolgung zur Auswahl des gültigen Tempolimits.
2010 2011 [Dom+10; DDT11]   JP	<i>Sichtbarkeitsanalyse von Verkehrszeichen.</i> Region Growing mit zufälligen Startpunkten im $L^*a^*b^*$ -Farbraum in der Umgebung detektierter Verkehrszeichen. Bestimmung der Sichtbarkeit von Verkehrszeichen als gewichtete Summe von Farb-, Kanten- und Texturmerkmalen zwischen Verkehrszeichen und den per Region Growing ermittelten umliegenden Blöcken. Temporale Fusion als Durchschnitt der Einzelmessungen.
2010 [Gu+10] ○   JP	<i>Erkennung von Verkehrszeichen mit aktivem Kamerasystem.</i> Weitwinkelkamera zur Detektion und Telekamera zur aktiven Verfolgung eines Verkehrszeichens. Fast Radial Symmetry gefolgt von Viola/Jones-Verfahren zur Detektion. Typbestimmung per Template Matching.
2010 [Kas+10] △ ○   DE	<i>Erkennung von Verkehrszeichen.</i> Bestimmung einer Attention Map über Gabor Filter, difference of Gaussians (DoG) und weitere Merkmale. Schwellwertverfahren zur Bestimmung von ROIs. Schwellwertverfahren zur Farbsegmentierung im RGBY-Farbraum. Multiplikative Kombination verschiedener schwacher Klassifikatoren auf Basis von Farb-, Form- und Kantenmerkmalen zur Klassifikation.
2009 2011 [KAB09; KBA11] △ □ ○ ○   US ES	<i>Erkennung von Verkehrszeichen.</i> Extraktion von Gabor-Merkmalen im $L^*a^*b^*$ -Farbraum und k-Means Clustering. Regelbasierte Filterung. Klassifikation der Form anhand des Verhältnisses von Umfang und Flächeninhalt und der Anzahl an Seiten durch Analyse des Abstands zum Rand. Klassifikation über Spektralmerkmale und Kreuzkorrelation mit Referenzmodellen.
2010 [LY10] △ ○   TW	<i>Erkennung von Verkehrszeichen.</i> Schwellwertregeln im HSI-Farbraum zur Segmentierung. Template Matching zur Formklassifikation und Typbestimmung. Verwendung eines Smartphones als Videokamera, WLAN-Streaming an datenverarbeitenden Computer.

Kategorisierung	Ziele und methodisches Vorgehen
2010 [LVPLAC10; PLAC10] △ ○ 👁 📷 ES	<i>Detektion von Verkehrszeichen.</i> Vorfilterung des Bildes mittels local contour patterns (LCPs) und Bildung von ROIs. Viola/Jones-Verfahren mit Texturmerkmalen (Quantum Features) angewandt auf ROIs zur Detektion.
2010 [Meu+10] ○ 🔄 📷 DE	<i>Bestimmung des gültigen expliziten Tempolimits.</i> Temporale Fusion nach Bayes. Zusammenfassung unterschiedlicher Tempolimit-Schilder mit gleichem Inhalt auf gleicher Höhe. Regeln zur Validierung aufeinander folgender Tempolimits.
2010 [PNB10] ○ 👁 📷 🗺 🕒 FR	<i>Bestimmung des gültigen Tempolimits.</i> D/S-Fusion zur Ermittlung des gültigen Tempolimits. Als Informationsquellen gehen die kamerabasierte Erkennung von Tempolimits und aus Kartendaten ermittelte Informationen zum Tempolimit ein. Die Konfidenz der Selbstlokalisierung geht in die Berechnung der Massefunktion des Kartendienstes ein.
2010 [RLL10] ○ ◇ △ ○ 👁 📷 PL JP	<i>Erkennung von Verkehrszeichen.</i> HSV-Segmentierung und dichte-basierte Regionenklassifikation. Region Growing und Zusammenhangskomponenten. Extended Fast Radial Symmetry zur Segmentierung von Formen. Tracking per Kalman-Filter und Nutzung der Prädiktion zur Adaption der Segmentierung. Merkmalsextraktion mittels Größennormalisierung, Farbdiskretisierung und Distanztransformation. Selektion diskriminativer Regionen zur Merkmalsreduktion. Nächster-Nachbar-Klassifikation zu Referenzmodellen und temporale Fusion.
2010 [Seg+10] △ 👁 📷 🗺 HR	<i>Inventarisierung von Verkehrszeichen.</i> Viola/Jones-Verfahren zur Detektion dreieckiger Verkehrszeichen. Größennormalisierung von Kandidaten, multiple discriminant analysis (MDA) und Nächster-Nachbar-Klassifikation zur Typbestimmung. Abgleich mit Geoinformationssystem (GIS)-Datenbank.
2010 [YCW10] □ 👁 📷 US	<i>Erkennung von Tempolimits.</i> Viola/Jones-Verfahren mit local binary pattern (LBP)-Merkmalen zur Segmentierung amerikanischer Tempolimits. Lineares Tracking zur Begrenzung von Suchregionen bei der Verfolgung von Kandidaten. Rotationsnormalisierung, Binarisierung mit adaptivem Thresholding und Extraktion von Zahlen durch Zusammenhangskomponenten. Dimensionsreduktion mittels principal components analysis (PCA), Klassifikation mit linearer SVM. Temporale Fusion durch Mehrheitsentscheidung.
2011 [Che+11] △ □ ○ ○ 👁 📷 CN	<i>Erkennung von Verkehrszeichen.</i> RGB-Schwellwerte und Zusammenhangskomponenten zur Bildung von ROIs. Viola/Jones-Verfahren zur Detektion. Extraktion von SURF-Merkmalen und Nächster-Nachbar-Klassifikator mit Referenzmodellen.
2011 [DL11; DLB11] ○ 👁 📷 🗺 🕒 FR	<i>Bestimmung des gültigen Tempolimits.</i> D/S-Fusion mit der Erkennung von Tempolimits und aus Kartendaten ermittelten Informationen zum Tempolimit als Informationsquellen.

Kategorisierung	Ziele und methodisches Vorgehen
2011 [Deg+11] ○   JP	<i>Unüberwachtes Lernen zur Verbesserung der Klassifikation von Verkehrszeichen.</i> Viola/Jones-artiges Verfahren zur Detektion von runden Verkehrszeichen. Bei Detektion Umgebungssuche in vorhergehenden Bildern ohne Detektionsergebnis mit farbbasierter Segmentierung. Bilder ohne Detektionsergebnis in der zeitlichen Nachbarschaft werden als Hintergrund angesehen. Unüberwachtes Lernen auf diesen Ergebnissen.
2011 [Jia+11] △ ○ ○   CN	<i>Detektion von Verkehrszeichen.</i> HSV-Vorfilterung und per Ridge Regression gelernte RGB-Schwelwertregeln zur Segmentierung. Auf RANSAC basierende Ellipsenerkennung zur Lokalisation perspektivisch verzerrter Kreise. Template Matching zur Formbestimmung. Binarisierung mit Schwellwertbestimmung nach dem Verfahren von Otsu. Extraktion von radialen Histogramm-Merkmalen und Klassifikation mit neuronalem Netz (MLP).
2011 [Li+11]    CN	<i>Erkennung von Ampeln.</i> TopHat-Operator und regelbasierte Filterung. Schätzung des Ampelkastens anhand des detektierten Ampellichts und Verifikation der Farbe des Ampelkastens. Histogrammanalyse zur Klassifikation der Ampelfarbe.
2011 [Liu+11] ○ □   CN	<i>Erkennung von Tempolimits.</i> Nutzung von HOG-Merkmalen zur Filterung des Kantenbildes, anschließend Fast Radial Symmetry. Größennormalisierung und Histogrammausgleich von ROIs und Extraktion von Fourier-Wavelets zur Klassifikation mit SVM. Symmetrie-Ansatz zur Detektion von Zusatzzeichen. Rotationsnormalisierung und Klassifikation mit SVM. Temporale Fusion.
2011 [Mar+11] △ ○   ES	<i>Detektion von Verkehrszeichen.</i> Bayes-Klassifikator mit Gauß-Mixturen im HS-Farbraum und Zusammenhangskomponenten zur Segmentierung. Schwellwertregeln für Größe und Ansichtsverhältnis. Tracking mittels Kalman-Filter. Temporale Konsistenzprüfung. Beeinflussung der a-priori-Wahrscheinlichkeiten anhand der Verkehrszeichen zugeordneten Regionen.
2011 [GG+11] △ ○   ES	<i>Erkennung von Verkehrszeichen.</i> Hough-Transformation für Kreise und Hough-Transformation für Linien sowie geometrische Einschränkungen für Dreiecke zur Segmentierung. SVMs mit RBF-Kernel zur Klassifikation. Tracking mittels Kalman-Filter. Bestimmung der 3D-Position mit Hilfe eines Stereosystems. V2I-Kommunikation mit derartig ausgestatteten Verkehrsschildern zur Ermittlung ihrer Position und ihres Typs.
2011 [Ove11] ○   NZ	<i>Klassifikation von Tempolimits.</i> Vergleich unterschiedlicher HOG-Merkmale in einem Kaskadenklassifikator zur Erkennung von Tempolimits. Evaluation auf einem umfangreichen Datensatz.
2011 [PNB11] ○ □    FR	<i>Bestimmung des gültigen Tempolimits.</i> Erweiterung des Systems aus [PNB10] um die Erkennung von Zusatzzeichen durch [Ham+08] sowie Fahrspurmarkierungen zur Spezialisierung auf Tempolimits an Ausfahrten. Entscheidung anhand von Zusatzzeichen, Art der Fahrspurmarkierung und Position des Fahrzeugs relativ zur Fahrspurmarkierung.

Kategorisierung	Ziele und methodisches Vorgehen
2011 [ZS11b; ZS11a]     FR DE	<i>Erkennung von Verkehrszeichen.</i> Vergleich von morphologischen Operatoren (TopHat und BlackHat), Farbverstärkung und Farbfilterung zur Vorverarbeitung. SVM mit HOG-Merkmalen zur Detektion. Vergleich von Nächster-Nachbar-Suche und Random Forests zur Klassifikation.
2012 2013 [PMN12; PMN13]    FR DE	<i>Erkennung von Zusatzzeichen.</i> Bestimmung von Randpunkten von Piktogrammen, Pfeilen und Texten auf Zusatzzeichen als Saatpunkte für ein Region Growing Verfahren. Vergleich mit kanten-, farb- und graphbasierten Detektionsverfahren. Klassifikation in vier Unterkategorien mittels HOG- und Intensitätsmerkmalen und SVM.
2012 [SSD12]   GR FR	<i>Erkennung von Ampeln.</i> Transformation in L*a*b*-Farbraum. Farbfilterung und Reduktion von Überstrahlung. Erkennung von Ampellichtern durch Fast Radial Symmetry und Verifikation über Tracking.
2012 [Zhe+12]      CN	<i>Erkennung von Verkehrszeichen.</i> RGB-Farbsegmentierung nach vorhergehendem Weißabgleich. Fast Radial Symmetry zur Detektion von Kreisen. Zusammenhangskomponenten und Douglas-Peucker-Verfahren zur Vereinfachung von Formen. Detektion von Dreiecken, Rechtecken und Oktagonen anhand der Anzahl von Polygonseiten. Extraktion von BRISK-Merkmalen (Binary Robust Invariant Scalable Keypoints) und Klassifikation durch Vergleich mit Referenzmodellen. Verfolgung mittels Kanade–Lucas–Tomasi-Verfahren.
2013 [CGL13]   CN	<i>Erkennung von Ampeln.</i> Schwellwertregeln im RGB-Farbraum für schwarze Pixel und Zusammenhangskomponenten zur Segmentierung von Ampelkästen. Ampellichtsegmentierung in ROIs im YcbCr-Farbraum. Regelbasierte Verifikation. Extraktion von Gabor-Wavelet-Merkmalen und Independent Component Analysis (ICA). Nächster-Nachbar-Klassifikator zu Referenzmodellen.

Tab. 2.1.: Kategorisierung und Beschreibung der Ziele und des methodischen Vorgehens verwandter Forschungsarbeiten (siehe Abschnitt 2.2).

2.3. Forschungsgruppen

Weltweit gibt es mehrere Teams, die in Zusammenarbeit wichtige Publikationen zu verwandten Themen veröffentlichten. Sie sind in Tabelle 2.1 durch identische Hintergrundfarben hervorgehoben und werden im Folgenden im Detail vorgestellt.

2.3.1. LaRA – La Route Automatisée

Die drei Forschungsteams *IMARA* (Informatique, Mathématiques et Automatique pour la Route Automatisée) von *INRIA* (French National Institute for Research in Computer Science and Control), *CAOR*



Abb. 2.3.: Komponenten eines adaptiven Templates (a) und Erkennungsergebnisse (b), (c) für französische Ampeln nach Charette und Nashashibi [CN09].

(Centre de Robotique) von Mines ParisTech und *LIVIC* (Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs) arbeiten unter dem Namen *LaRA* (La Route Automatisée) an Informations- und Telekommunikationslösungen im Transportwesen [Lar]. In diesem Rahmen entstanden einige Arbeiten zur Bestimmung des gültigen Tempolimits in europäischen Ländern (vornehmlich Frankreich und Deutschland). Die bildbasierte Detektion erkennt runde Schilder mittels Hough-Transformation für Kreise und rechteckige Schilder durch ein nicht näher beschriebenes Verfahren [Mou+07]. Geschwindigkeitsangaben werden über ein zweistufiges Verfahren erkannt, das zunächst eine Segmentierung innerhalb von Schildern durchführt, um Ziffern zu extrahieren, und diese anschließend per neuronalem Netz klassifiziert. Zur Verbesserung der Erkennungsgenauigkeit wurde später die Segmentierung der Ziffern verbessert [Ham+08]. Die Erkennung von Aufhebungsschildern wurde durch ein neuronales Netz realisiert, welches auf den erkannten Kreisen arbeitet [Bar+08]. Zusatzzeichen unterhalb von Geschwindigkeitsbeschränkungen werden durch einen ähnlichen Ansatz erkannt, der eine Segmentierung von Rechtecken mit einer Erkennung durch ein neuronales Netz kombiniert. Die Segmentierung wurde später durch ein Region-Growing-Verfahren ersetzt [PMN12] und die Klassifikation um einen HOG-Ansatz ergänzt, der vier Unterkategorien unterscheidet [PMN13]. Das bildbasierte System wurde des Weiteren mit einem Kartendienst kombiniert [Bar+10; PNB10; PNB11].

Eine weitere Arbeit widmet sich der Erkennung von Ampeln. Charette und Nashashibi [CN09] präsentieren ein generisches System zur Erkennung von Ampeln und ihrer Zustände. Es verbindet eine Erkennung von Lichtpunkten mit einer Validierung über adaptives Template Matching. Auch wenn eine Farbkamera mit einer Auflösung von 640×480 Bildpunkten zum Einsatz kommt, wird ihre Farbinformation nicht genutzt. Zur Erkennung von Lichtpunkten kommt der weiße Top-Hat-Operator zum Einsatz, gefolgt von einer Extraktion von Zusammenhangskomponenten. Anschließend werden die extrahierten Regionen anhand verschiedener Kriterien gefiltert. Die Validierung der gefundenen Regionen übernimmt ein Template-Matching-Verfahren. Es besteht hierarchisch aus Komponenten, deren Parameter an die gefundenen Regionen adaptiert werden (vergleiche Abbildung 2.3a). Die von jeder Komponente berechnete Konfidenz wird zu einer Gesamtkonfidenz kombiniert und zur Validierung bzw. Ablehnung der Regionen benutzt. Bei der Evaluation des Systems auf zwei Videosequenzen (Gesamtlänge circa 16 Minuten) in einer Innenstadt werden 98.41 % aller Ampeln mindestens einmal korrekt erkannt. Eine Klassifikationsrate bezogen auf Bilder wird nicht angegeben. Die Laufzeit wird mit durchschnittlich

26.7 Hz bei Verwendung eines 2.9 GHz Prozessors angegeben.

Diskussion In den verschiedenen Veröffentlichungen rund um die Erkennung des gültigen Tempolimits werden robuste Verfahren eingesetzt. Hierzu zählen insbesondere die formbasierte Segmentierung, die Klassifikation durch ein überwachtes Lernverfahren und die Informationsfusion nach Dempster/Shafer. Durch die Segmentierung einzelner Ziffern von Tempolimits wird das Klassifikationsproblem von (in Deutschland) 14 Klassen von Geschwindigkeitsbeschränkungen reduziert auf 10 verschiedene Klassen von Ziffern. Interessant wären Untersuchungen dazu, ob diese Problemreduktion den durch die zusätzlich notwendige Segmentierung eingebrachten Fehler aufwiegt. Andernfalls könnte auf die globale Segmentierung [Bar+08] und die spezielle Handhabung von Aufhebungszeichen [Ham+08] verzichtet werden. Die Erkennungsrate auf den Datensätzen der Forschungsgruppe ist mit 94 % (bezogen auf gültige Tempolimits) hoch bei echtzeitnaher Verarbeitung von 20 Bildern pro Sekunde [Bar+08].

Der von Charette und Nashashibi gewählte Einsatz des Top-Hat-Operators zur Segmentierung von Ampeln überzeugt. Leider wird auf die Nutzung von Farbinformation – mit der Argumentation der alternativen Einsatzmöglichkeit einer Grauwertkamera – verzichtet, so dass keine Erfahrungswerte zur Nutzung dieses bedeutungstragenden Merkmals vorliegen. Ein Vergleich des adaptiven Template-Matching-Verfahrens mit alternativen Erkennungsverfahren wäre interessant, um die Robustheit des Template-Matchings in diesem Anwendungsfall abzuschätzen. Die Evaluation wird in Bezug auf reale Ampeln durchgeführt und eine Ampel als erkannt gewertet, wenn sie in mindestens einem Bild über den gesamten Beobachtungszeitraum erkannt wird. In diesem Sinne werden 98.41 % der Ampeln der Testmenge korrekt erkannt.

2.3.2. Signalverarbeitungs-Gruppe der Universität Alcalá

Innerhalb der Signalverarbeitungs-Gruppe (Departamento Teoría de la Señal y Comunicaciones) der Universität Alcalá entstanden eine Vielzahl von Publikationen zur Erkennung spanischer Verkehrszeichen vor allem zu Zwecken der Inventarisierung [MB+07] (Abbildung 2.4b, 2.4c). Der Großteil der Veröffentlichungen behandelt verschiedene Ansätze zur Klassifikation der Form von Verkehrszeichen [GJ+05; LA+05; VR05; GJ+07; GJ+08].

Zur Segmentierung farbiger Verkehrszeichen wird eine Farbklassifikation im HSI-Farbraum mittels Schwellwerten für Farb- und Sättigungswerte durchgeführt. Achromatische Verkehrszeichen werden über Schwellwerte im RGB-Farbraum erkannt [VR05]. Zusammenhängende Regionen positiv klassifizierter Bildpunkte werden als rechteckige *Blob of Interest (BoI)* extrahiert. Ausgehend von den vier Seiten dieser Rechtecke wird die horizontale bzw. vertikale Distanz von der jeweiligen Seite bis zum ersten positiv klassifizierten Bildpunkt einer Reihe bzw. Spalte als Merkmalsvektor extrahiert (siehe Abbildung 2.4a) [MB+07]. Basierend darauf wurden verschiedene Arten der Klassifikation der Form entwickelt: Mittels Korrelation zu Referenzmodellen [VR05], linearer SVM [LA+05], Anwendung von FFT und Vergleich mit Referenzmodellen [GJ+05] sowie der Anwendung von FFT und SVM [GJ+07]. Basierend auf der Bestimmung von 2D-Homographien wird eine Normalisierung der Form vorgenommen [GJ+08]. Abhängig von der klassifizierten Form werden die inneren Bildpunkte des Verkehrszeichens

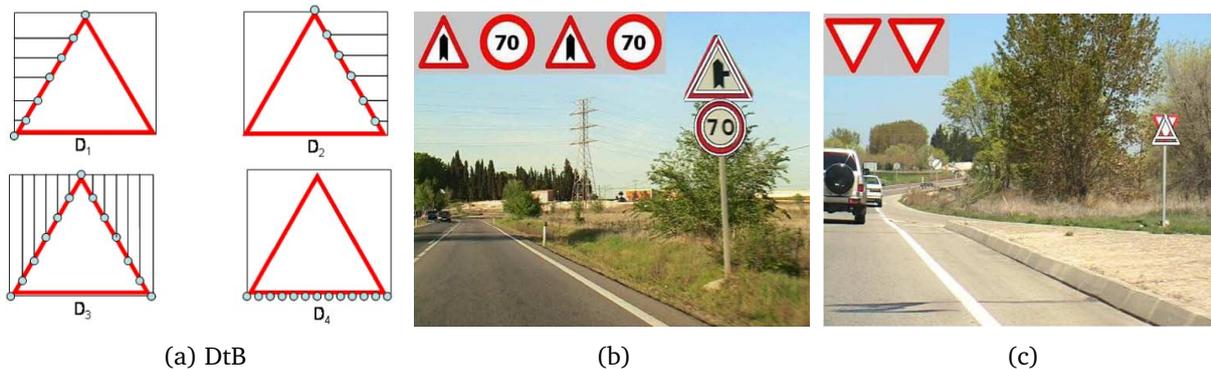


Abb. 2.4.: *Distance to Border (DtB)* Transformation (a) und Erkennungsergebnisse (b), (c) für verschiedene Verkehrszeichentypen nach Maldonado-Bascón u. a. [MB+07].

nach vorhergehender Grauwerttransformation und Größennormalisierung extrahiert und durch eine lineare SVM klassifiziert, um Piktogramme zu unterscheiden. Verkehrszeichen werden mittels Kalman-Filter verfolgt, um kurze Störungen zu unterdrücken [LA+07] und gemäß Mehrheitsentscheidung eine temporale Fusion durchzuführen [LA+08].

Maldonado-Bascón u. a. evaluieren das System mit insgesamt 5176 Bildern einer Größe von 720×576 Bildpunkten in 5 Videosequenzen (circa 4 km Fahrt) [MB+07]. Dabei kommt es zu einer Falschpositivrate von 0.4% pro Bild; im Durchschnitt wird also etwa alle 250 Bilder ein Hintergrundobjekt als Verkehrszeichen erkannt. Die Autoren geben an, dass jedes der 104 enthaltenen Verkehrszeichen mindestens zwei Mal erkannt wird, nennen aber keine detaillierten Erkennungsraten. Die Verarbeitung eines Bildes benötigt durchschnittlich 1.77 Sekunden (0.56 fps) auf einen Pentium 4-M mit 2.2 GHz.

Diskussion Durch den Anwendungsfall der Inventarisierung ist keine frühzeitige Erkennung von Verkehrszeichen notwendig. Entsprechend ist die Evaluation nicht auf einzelne Bilder, sondern reale Verkehrszeichen bezogen. Um das Verfahren auf andere Anwendungsfälle zu übertragen, bei denen im Bild kleine Verkehrszeichen (Verkehrszeichen in großer Distanz) erkannt werden sollen, ist die Übertragbarkeit der Farbsegmentierung zu evaluieren. Hierbei könnte es zu Problemen kommen, wenn die farbigen Ränder der untersuchten roten Verkehrszeichen im Bild klein erscheinen. Die hohe Laufzeit des Algorithmus ist nicht auf einzelne Komponenten aufgeschlüsselt. Vermutlich entstehen durch die Segmentierung sehr viele Regionen, die anschließend von einer SVM zur Formbestimmung bzw. zwei SVMs zur Formbestimmung und Typklassifikation untersucht werden. Eine Auswechslung der Segmentierung (Farbsegmentierung und Formklassifikation) durch modellbasierte Verfahren, die Kantenmerkmale nutzen, könnte sowohl die Laufzeit reduzieren als auch die Robustheit erhöhen. Die Klassifikation von Verkehrszeichen mittels Grauwertextraktion und SVM erscheint robust.

2.3.3. VisLab

Das *VisLab* (Artificial Vision and Intelligent Systems Laboratory) der Universität Parma führt Grundlagenforschung und angewandte Forschung im Automobilbereich schwerpunktmäßig im Bereich Ma-

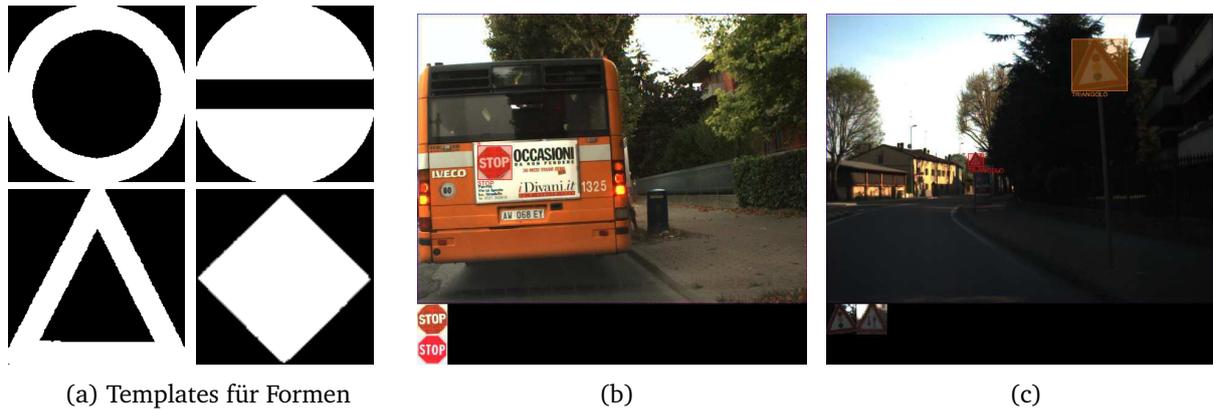


Abb. 2.5.: Broggi u. a. [Bro+07] nutzen Farbklassifikation und einfache Formschablonen zur Segmentierung verschiedener italienischer Verkehrszeichen. Sie werden mittels neuronaler Netze klassifiziert.

schinensehen durch [Vis]. Zur Erkennung verschiedener italienischer Verkehrszeichen stellen Broggi u. a. [Bro+07] einen auf Farbklassifikation und neuronalen Netzen basierenden Ansatz vor. Die Farbklassifikation benutzt Schwellwertentscheidungen im RGB-Farbraum, um Regionen roter, blauer und gelber Farbe zu extrahieren. Um die Abhängigkeit von der Farbe des Umgebungslichts zu reduzieren, wird ein Weißabgleich durchgeführt. Zur Bestimmung der Farbe des Umgebungslichts wird eine Region im unteren Bildbereich extrahiert und als grau (Straße) angenommen. Über die Abweichung vom Grauwert werden Parameter für den Weißabgleich bestimmt. Extrahierte Farbregionen werden binarisiert und mittels Template Matching in fünf Formen klassifiziert. Für alle Formen werden neuronale Netze trainiert; die Regionen werden auf Grauwertbilder der Größe 50×50 skaliert, ihr Kontrast normalisiert und als Eingabe für das formabhängige neuronale Netz benutzt. Eine quantitative Evaluation wurde nicht durchgeführt.

Um auch Aufhebungsschilder zu erkennen – die mangels Farbinformationen nicht durch die in Vorarbeiten vorgestellten Ansätze verarbeitet werden können – präsentieren Caraffi u. a. [Car+08] ein weiteres System. Es nutzt den schwarzen diagonalen Streifen (vergleiche Abbildung 2.6) auf Aufhebungszeichen als primäres Segmentierungsmerkmal. Um seine Position zu detektieren, werden vertikale Kanten mittels eines einfachen Filters hervorgehoben. Ein Zustandsautomat wird verwendet, um die charakteristischen Kantenübergänge in Bildreihen zu detektieren. Auf diese Weise ergibt sich ein Binärbild, in dem der schwarze Streifen von Kandidaten hervorgehoben ist; mittels eines Region-Growing-Verfahrens und vorgegebenen Regeln werden daraus ROIs extrahiert. Anschließend wird mittels Hough-Transformation für Kreise eine Verifikation durchgeführt. Ein einfaches bildbasiertes Tracking wird zur Verfolgung von Kandidaten eingesetzt. Das System erkennt auf einem Testdatensatz von circa 10000 Bildern 35 von 37 Aufhebungszeichen in mindestens zwei aufeinander folgenden Bildern.

Da sich der von Broggi u. a. [Bro+07] vorgestellte Ansatz der Formklassifikation über Template Matching als wenig robust erwies, wurde er von Cardarelli u. a. [Car+09] durch einen Ansatz ersetzt, der mittels Hough-Transformation Linien erkennt und deren Lage analysiert. Damit können Rechtecke und Dreiecke, aber keine Kreise erkannt werden. In einer weiteren Veröffentlichung [MC08] wurde eine Erweiterung des Systems beschrieben, bei der die Schwellwertentscheidung im RGB-Farbraum ersetzt wurde durch eine Farbklassifikation im HSV-Farbraum. Da das Training des neuronalen Netzes



Abb. 2.6.: Die Segmentierung von Caraffi u. a. [Car+08] detektiert schwarze diagonale Streifen und führt eine Hough-Transformation für Kreise zur Verifikation durch.

mittels künstlich erzeugter Trainingsbeispiele keine zuverlässigen Resultate erzielen, wurde es durch reale Trainingsbeispiele ergänzt. Diese wurden durch manuelle Auswahl von Experten erweitert. Bei Verwendung einer Farbkamera mit 752×480 Bildpunkten und einem Pentium 4 Prozessor verarbeitet das System 10 Bilder pro Sekunde. Auf einem 100 km langen Testsatz werden 74 % der Verkehrszeichen korrekt erkannt.

Diskussion Durch die Integration verschiedener robuster Verfahren in [Car+09; MC08] konnte die Erkennungsleistung des ursprünglichen Systems [Bro+07] verbessert werden. Mit einer Erkennungsrate von 74 % ist es aber nicht alltagstauglich. Da von den Autoren keine detaillierte Fehleranalyse durchgeführt wurde, können über die Fehlerursachen nur Mutmaßungen angestellt werden. Eine Fehlerquelle könnte die Farbklassifikation im ersten Schritt darstellen. Da Verkehrszeichen im Bild eher klein sind und die Aufnahmebedingungen nicht kontrollierbar sind, sind hier Fehler zu erwarten. Das eingesetzte Verfahren zum Weißabgleich soll dem entgegenwirken. Es beruht auf der Annahme, dass vor dem Fahrzeug grauer Straßenbelag zu sehen ist, der von einer einzigen Lichtquelle gleichmäßig angestrahlt wird. Hier wären Untersuchungen interessant, wie sich Verletzungen dieser Annahme durch Verdeckungen durch andere Fahrzeuge, Farbvariationen des Straßenbelags und die Beleuchtung durch Scheinwerfer auswirken. Bei der Extraktion von ROIs ist mit einer hohen Varianz der ermittelten Position zu rechnen, da Rotationen auftreten und die Randbereiche nicht stabil bestimmt werden können. Dies muss vom Klassifikationsverfahren kompensiert werden, indem beispielsweise eine sehr große Menge an Trainingsdaten verwendet wird.



(a) Haar-Wavelets mit Farbinformationen

(b) Erkennung von Tempolimits

(c) Erkennung von Wechselverkehrszeichen

Abb. 2.7.: Die sechs wichtigsten Haar-Wavelets (a) und Erkennungsergebnisse (b), (c) für verschiedene Verkehrszeichen nach Bahlmann u. a. [Bah+05].

2.4. Ausgewählte Einzelarbeiten

Im Folgenden werden weitere ausgewählte Arbeiten zu den oben aufgeführten Problemstellungen im Detail vorgestellt, diskutiert und bewertet.

2.4.1. Viola/Jones-Ansatz zur Erkennung von Tempolimits

Die von Viola und Jones 2001 veröffentlichte Arbeit „Robust real-time object detection“ fand in vielen späteren Veröffentlichungen Anwendung als Grundlage der Objekterkennung. Der vorgestellte Ansatz kombiniert Haar-Wavelets als Merkmale mit einem Kaskaden-Klassifikator zu einem Objektdetektor. Das Training erfolgt durch ein Boosting-Verfahren. Bahlmann u. a. [Bah+05] nutzen diesen Ansatz und erweitern die Haar-Wavelets um die Berücksichtigung von Farbinformationen im RGB-Farbraum (vergleiche Abbildung 2.7a). Da die Merkmale nicht skalierungsinvariant sind, werden für verschiedene Größen (Durchmesser von 14, 20, 28, 40 und 54 Bildpunkte) Detektoren trainiert. Objekte werden mit Hilfe eines einfachen Bewegungsmodells verfolgt.

Die Klassifikation basiert auf einer LDA-Transformation. Die ersten 25 Basisvektoren der LDA werden für jede Klasse durch Gauß-Mixturen repräsentiert. Zur Klassifikation werden die inneren Bildpunkte der detektierten Kreise extrahiert, transformiert und mit den gespeicherten Basisvektoren gemäß Maximum Likelihood verglichen. Darüber hinaus wird eine temporale Fusion durchgeführt.

Zur Evaluation des Systems wird eine 30-minütige Videosequenz in einer Auflösung von 384×288 Bildpunkten verwendet. Dabei werden 85 % aller zu erkennenden Verkehrszeichen im letzten Bild einer Tracking-Sequenz korrekt erkannt. Eine der Hauptfehlerursachen sind Verkehrszeichen, die eine Größe aufweisen, die nicht den trainierten Größen entsprechen. Die Verarbeitungsgeschwindigkeit beträgt etwa 10 Hz auf einem System mit einem 2.8 GHz Intel Xeon Prozessor.

Diskussion Der Hauptbeitrag der Arbeit ist die Erweiterung des Viola/Jones-Verfahrens um die Berücksichtigung von Farbinformationen bei der Merkmalsextraktion. Sie wird vom AdaBoost-Training ebenso wie die Haar-Merkmale beim Aufbau des Klassifikators berücksichtigt. Die Evaluation bestätigt die Nützlichkeit der Farbinformationen: Von den sechs wichtigsten Merkmalen benutzen fünf Farbinformationen. Die Falschpositivrate kann durch die Nutzung von Farbinformationen um eine Größenordnung gesenkt werden. Ein Nachteil des Verfahrens ist die Abhängigkeit von der trainierten Größe der Verkehrszeichen. Die Evaluation bestätigt, dass Verkehrszeichen in Zwischengrößen den Hauptanteil der Segmentierungsfehler ausmachen. Bei der Repräsentation von Klassen durch Gaußmixturen stellt sich die Frage, ob hierdurch die notwendige Ausdrucksstärke gewährleistet wird. Da die Klassifikation unabhängig von den anderen Komponenten des Systems ist, könnte sie aber einfach durch ein besser geeignetes Verfahren ersetzt werden.

2.4.2. Erkennung von Ampeln mittels Hough-Transformation

In seiner Dissertation „*Erkennung von Ampeln aus Onboard-Videosequenzen für Fahrerassistenzsysteme*“ entwickelte Eckert [Eck07] ein System zur Detektion und Verfolgung von deutschen Ampeln. Als Sensorik wird ein Stereo-Kamerasystem mit einer Auflösung von jeweils 1280×1024 Pixeln und einer Bildrate von 12 Hz verwendet. Die Segmentierung basiert auf einer Hough-Transformation für Kreise. Ausgehend von der Größe und Position so erkannter möglicher Ampellichter wird der Ampelkasten geschätzt und seine Parameter anschließend mittels Template Matching angepasst. Die Verifikation erfolgt nicht über ein automatisches Klassifikationsverfahren, sondern über manuell parametrisierte Schwellwertentscheidungen. Dabei werden Flächenmerkmale basierend auf Intensität, Sättigung und Kantendichte benutzt. Schließlich werden Kandidaten aussortiert, die eine geringe Distanz zueinander aufweisen. Mit Hilfe einer zweiten Kamera wird im Stereo-Verfahren die Entfernung von Punkten geschätzt, daraus die 3D-Position des Ampelkandidaten geschätzt und diese zur Filterung von Falschpositiven genutzt. Ein Tracking mittels Kalman-Filter wird zur Verfolgung von Objekten eingesetzt.

Zur Evaluation des Systems wird eine Videosequenz von etwa 12 Minuten Länge (8869 Bilder) mit insgesamt 37 Ampelanlagen eingesetzt. Eine Ampelanlage wird als korrekt erkannt bezeichnet, wenn mindestens eine der fahrrelevanten Ampeln mindestens einmal erkannt wird. Die Erkennung von nicht fahrrelevanten Ampeln (für andere Fahrspuren sowie Fußgängerampeln) wird ignoriert und nicht in die Falschpositivrate eingerechnet. In diesem Sinne werden 35 Ampelanlagen korrekt erkannt. Eine Erkennungsrate bezogen auf Ampeln und Einzelbilder wird nicht angegeben. Die Laufzeit des Systems beträgt 2–4 Sekunden pro Bild bei Verwendung eines Pentium 4 Prozessors mit 3.2 GHz.

Diskussion Die Verwendung der Hough-Transformation als Basisverfahren für die Segmentierung stellt eine starke Einschränkung für die Ampelerkennung dar. Einerseits wird dadurch die Erkennung in größerer Entfernung unmöglich (die Hough-Transformation benötigt einen minimalen Radius zur Erkennung von Kreisen, Ampeln wiederum sind vergleichsweise kleine Objekte), andererseits ist die Erkennung von Ampeln mit Richtungspfeilen nicht möglich. Diese stellen einen wesentlichen Anteil der Ampeln in Deutschland dar. In der Arbeit gibt es einige interessante Vorschläge zur Feinregistrierung

von Ampeln und zur Entwicklung eines Gütemaßes zur Verifikation. Die hohe Falschpositivrate von 1.6 Fehlerkennungen pro Minute könnte durch den Einsatz eines Lernverfahrens zur Verifikation von Ampeln reduziert werden. Die Evaluation gibt die Klassifikationsrate des Systems bei geringem Datenumfang mit 95 % an. Detaillierte Angaben zu Erkennungsraten für einzelne Ampeln werden nicht gemacht. Die hohe Falschpositivrate und der hohe Laufzeitaufwand verhindern einen praktischen Einsatz des Systems.

2.4.3. Erkennung von Ampeln mittels Viola/Jones-Objekterkennung

Eine Ampelerkennung als Kombination von HSI-Farbklassifikation, Viola/Jones-Erkennung und Camshift-Tracking wird von Gong u. a. [Gon+10] vorgestellt. Zunächst wird eine HSI-Transformation durchgeführt und über Schwellwerte eine Farbklassifikation durchgeführt. Nach der Anwendung morphologischer Operatoren zur Rauschunterdrückung (Erosion und Dilatation) werden Ampelkandidaten über Zusammenhangskomponenten extrahiert. Diese Bereiche werden mittels auf Haar-Merkmalen aufbauendem AdaBoost-Klassifikator validiert und durch den Camshift-Algorithmus verfolgt. Letzterer operiert auf dem Farbhistogramm der extrahierten bzw. prädizierten Bereiche. Als Sensor kommt eine im Fahrzeug montierte Videokamera mit einer Auflösung von 780×580 Bildpunkten und einem Teleobjektiv zum Einsatz. Details zur Laufzeit und eine quantitative Evaluation werden nicht angegeben.

Diskussion Die Kombination der drei Verfahren (HSI-Farbklassifikation, Viola/Jones-Erkennung und Camshift-Tracking) erscheint für die Problemstellung angemessen. Die erste Segmentierungsstufe filtert weniger stark als die von Charette und Nashashibi [CN09] verwendete und legt dadurch mehr Gewicht auf die Validierung durch AdaBoost. Durch die Verwendung der schnell berechenbaren Haar-Merkmale sollte dies auch in Echtzeit durchführbar sein. Als Nachteil der Validierung durch den Viola/Jones-Ansatz ist die Notwendigkeit des aufwendigen Neu-Trainings bei Änderungen an der Sensorik anzusehen. Leider fehlen in der Veröffentlichung Details zur Umsetzung. Das Zusammenspiel der einzelnen Verarbeitungsschritte und die auftretenden Probleme werden nur angedeutet. Durch das komplette Fehlen einer quantitativen Evaluation ergeben sich keine Erkenntnisse zur Robustheit der Implementierung.

2.4.4. Detektion von Text auf Vorwegweisern

Wu, Chen und Yang [WCY05] stellen ein System zur Detektion amerikanischer Vorwegweiser und zur Extraktion der Textregionen in ihnen vor. Die Möglichkeit der Texterkennung mittels OCR wird kurz diskutiert, aber nicht umgesetzt.

Als erster Segmentierungsschritt werden Eckpunkte gemäß dem Verfahren von Shi/Tomasi [ST94] extrahiert. Die 3×3 Nachbarschaft eines jeden solchen Merkmalspunkts wird durch zwei Gaußverteilungen im HSI-Farbraum modelliert. Ein k-Means Clustering mit fester Anzahl von Clustern auf den Merkmalen aller Eckpunkte gruppiert die gefundenen Punkte. Es werden Korrespondenzen gebildet zwischen den Punkten aufeinander folgender Bilder. Cluster mit drei oder mehr korrespondierenden Punkten werden daraufhin untersucht, ob sie sich auf einer Ebene senkrecht zur Fahrtrichtung befinden. Solche Cluster werden als Kandidaten für Vorwegweiser gespeichert, andere verworfen.



Abb. 2.8.: Ergebnisse der Detektion von Vorwegweisern und Textregionen nach Wu, Chen und Yang [WCY05]. Bilder in der gleichen Spalte gehören zur gleichen Szene. Die obere Zeile zeigt die erkannten Merkmalspunkte, die untere Zeile die zu einem späteren Zeitpunkt erkannten Vorwegweiser und Textregionen. Die Bilder stammen von einer Farbvideokamera mit einer Auflösung von 640×480 Bildpunkten.

Kandidatenregionen für Vorwegweiser werden weiter auf Textregionen untersucht. Dazu werden in der als umschließendes Rechteck eines jeden Clusters modellierten Region Kanten extrahiert und die Farbverteilung der Umgebung von Kantenpixeln erneut als Gaußmixturen modelliert. In einer Art Region Growing werden Pixel zu potentiellen Textregionen zusammengefasst. Über eine Hough-Transformation wird abschließend die Ausrichtung von Textregionen bestimmt.

Das Verfahren wird auf 22 Videosequenzen von jeweils etwa 30 Sekunden Länge evaluiert. Dabei werden 94.4 % der Vorwegweiser detektiert bei einer Falschpositivrate von 17.9 %. Die Detektion von Textregionen ermittelt 88.9 % der Regionen bei einer Falschpositivrate von 9.2 %. Bei der verwendeten Farbvideokamera mit 640×480 Bildpunkten als Eingabe läuft das System mit 8–16 Hz auf einem 1.8 GHz Pentium IV Prozessor.

Diskussion Die Verwendung von Eckpunkten als primärem Segmentierungsmerkmal erscheint nicht vorteilhaft. Wie in Abbildung 2.8 zu sehen ist, werden Eckpunkte auch in zahlreichen anderen Teilen der Szene gefunden. Um die Echtzeitverarbeitung zu ermöglichen, müssen Wu, Chen und Yang die Anzahl verarbeiteter Eckpunkte auf 150 begrenzen. Dabei ist nicht davon auszugehen, dass korrespondierende Eckpunkte stabil in aufeinander folgenden Bildern gefunden werden. Gerade das ist aber Voraussetzung für die nachfolgende Unterscheidung von Vorwegweisern und anderen Objekten. Beim k-Means Clustering stellt sich die Frage, inwiefern sich negative Auswirkungen ergeben, wenn die fest vorgegebene Anzahl von 10 Clustern nicht der tatsächlichen Menge von Objekten entspricht. Insgesamt bleibt es fraglich, ob das System mit der Vielzahl aufeinanderfolgender Schritte, die jeweils über mehrere Parameter optimal eingestellt werden müssen, robust und flexibel eingesetzt werden kann. Sowohl die Segmentierung als auch die Extraktion der Textregionen könnten über eine stärkere Betrachtung der

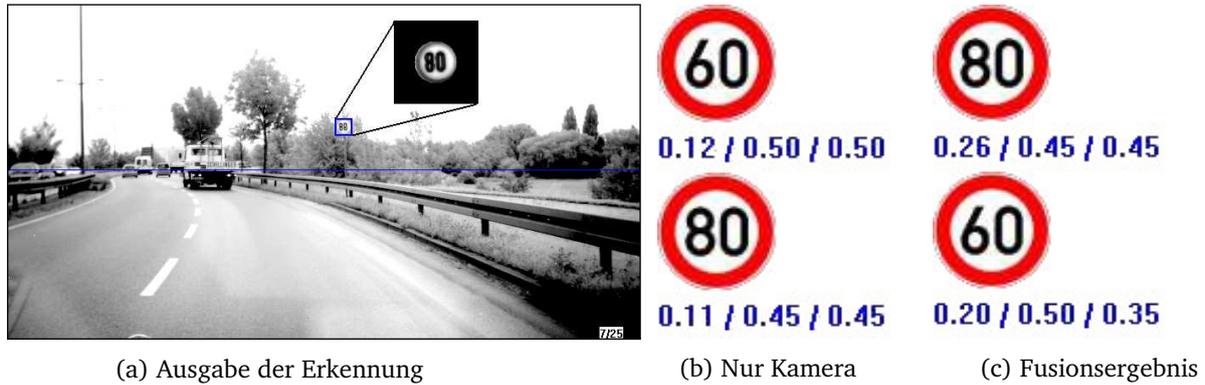


Abb. 2.9.: Qualitative Evaluation der Bayesschen Fusion zur Schätzung des gültigen Tempolimits an einer Beispielszene nach Bahlmann u. a. [Bah+08].

Homegenität der Hintergrundfarbe der gesuchten Wegweiser vereinfacht werden.

2.4.5. Bayessche Fusion zur Schätzung des gültigen Tempolimits

Bei Nacht, schlechtem Wetter und Gegenlicht ist mit Perzeptionsfehlern zu rechnen. Kartendienste versagen in Baustellen, bei Wechselverkehrszeichen und in nicht kartierten Gebieten. Bahlmann u. a. [Bah+08] schlagen ein Bayessches Fusionsverfahren vor, das beide Informationsquellen kombiniert, um eine insgesamt robustere Schätzung des gültigen Tempolimits zu erhalten. Die Ergebnisse der Perzeption gehen über deren Wahrscheinlichkeitsschätzung für sichtbare Tempolimits ein. Vom Kartendienst werden Tempolimits und Straßentyp auf von Hand modellierte Wahrscheinlichkeiten abgebildet. Aus den beiden Wahrscheinlichkeiten wird eine fusionierte Wahrscheinlichkeit berechnet. Die Laufzeit des System ist nicht angegeben. Es ist jedoch davon auszugehen, dass die Laufzeit der Fusion im Vergleich zu den anderen Komponenten zu vernachlässigen ist. Die Evaluation wird qualitativ anhand der Beispielszene in Abbildung 2.9 durchgeführt.

Diskussion Mangels einer quantitativen Evaluation lassen sich die Vor- und Nachteile der Fusion in Bezug auf die Einzelkomponenten kaum abschätzen. Wie von den Autoren angemerkt hängt die Qualität der Fusion stark von der Parametrierung der Wahrscheinlichkeiten ab. Die für die beiden Informationsquellen nachteiligen Situationen gehen nicht explizit in die Fusion ein. Es ist fraglich, ob die statische Parametrierung in allen Situationen geeignet ist. Interessant wäre die Erkennung nachteiliger Situationen und die Adaption der Fusion (bzw. die Wahrscheinlichkeiten der Informationsquellen) daran.

2.5. Zusammenfassung und Forschungsbedarf

Dank starker Forschungsaktivitäten im Bereich der Erkennung von Verkehrselementen konnten im vergangenen Jahrzehnt große Leistungssteigerungen erzielt werden. Dazu tragen unter anderem die Fast Radial Symmetry, das Viola/Jones-Verfahren und das HOG-Merkmal bei, welche auch in anderen Forschungsbereichen Anwendung finden. Dennoch wird das Problem nicht als gelöst betrachtet. Systeme

zur Erkennung von Tempolimits, die in Serienfahrzeugen verbaut werden, bleiben aktuell noch weit hinter der Erkennungsleistung des Menschen zurück. Assistenzsysteme im alltäglichen Einsatz müssen mit einer großen Situationsvielfalt – Baustellen, Dunkelheit, Gegenlicht, Regen, Schnee, Verdeckungen und mehr – umgehen können. Ein großes Potential haben in diesem Bereich lernende und sich selbst adaptierende Systeme. Eine ständige Herausforderung stellen nationale Besonderheiten dar. Sie erschweren auch die Austauschbarkeit von Datensätzen zu Zwecken der Evaluation, was wiederum die Entstehung von Benchmarks behindert. Dementsprechend gibt es bislang noch kaum relevante Benchmarks und Evaluationen werden auf eigenen Datensätzen durchgeführt, so dass keine vergleichende Evaluation verschiedener Ansätze möglich ist [Sta+11; MTM12].

Durch die immer stärkere Verbreitung von durch Wechselverkehrszeichen in LED-Technik dargestellten Informationen ist die Erkennung dynamischer Informationen wichtiger denn je. Aus ähnlichem Grund ist die Erkennung von Lichtsignalen ein Forschungsfeld mit großem Potential; dieser Trend zeigt sich bereits an der steigenden Anzahl an Veröffentlichungen zur Erkennung von Ampeln. Zustandswechsel, nationale Besonderheiten und der Einsatz vor allem in Innenstädten sind hier besondere Herausforderungen. Für die eng verwandten Anwendungsfelder des (teil-)autonomen Fahrens und der Energieeffizienz ist die Erkennung dynamischer Informationen eine Notwendigkeit [Bar+11; Nie+12; Koh+13].

Um die Akzeptanz des Fahrers zu fördern und eine echte Hilfe als Assistenzsystem darzustellen, muss ein besseres Szenenverständnis aufgebaut werden, um die Relevanz von Verkehrselementen zu beurteilen. Diese Kontextsensitivität schließt die Nutzung von Informationen über andere konkrete und abstrakte Objekte ein, um ein Verständnis der Bedeutung von Verkehrszeichen in der aktuellen Verkehrssituation aufzubauen. Der Mangel von öffentlich verfügbaren Datensätzen und Algorithmen mag ein Grund dafür sein, dass es bislang noch kaum Arbeiten zur Situationsinterpretation gibt. Dementsprechend groß ist hier der Forschungsbedarf, wie auch Mogelmoose, Trivedi und Moeslund in ihrer Studie „Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey“ [MTM12] erkennen und die Integration von Kontextwissen (“integration of context”) und die Berücksichtigung der Relevanz von Verkehrszeichen (“ensuring that the detected signs are relevant for the ego car”) als offene Problemstellungen nennen.

3. Umfelderkennung

Als Grundlage der anschließenden Interpretation einer Verkehrsszene ist die ständige Erfassung der vorliegenden Verkehrssituation notwendig. Dabei wird das Ziel verfolgt, möglichst viele und genaue Informationen über Objekte in der Umgebung und ihre Eigenschaften zu sammeln. Zur automatisierten Extraktion der gewünschten Informationen wird die in Abbildung 3.1 dargestellte Prozesskette genutzt, die ausgehend von Sensordaten in mehreren Schritten Daten vorverarbeitet, filtert und auf Objektmodelle abbildet. Die Verarbeitungsreihenfolge entspricht typischerweise der Leserichtung von links nach rechts: Sensordaten werden vorverarbeitet, interessante Regionen ermittelt und klassifiziert, Objekthypothesen werden verfolgt und im Rahmen der Interpretation werden unter Berücksichtigung des Situationskontexts weitere Informationen hinzugefügt. Die busartige Zusammenschaltung der Komponenten erlaubt darüber hinaus aber auch die Einflussnahme von Komponenten aufeinander. Beispielsweise kann Wissen über die Historie einer Objekthypothese innerhalb des Trackings genutzt werden, um Segmentierungsparameter anzupassen, oder die Parameter der Belichtungsregelung der Kamera können abhängig von der Fahrzeuggeschwindigkeit modifiziert werden.

In den folgenden Abschnitten 3.1 bis 3.4 werden die einzelnen Komponenten der Prozesskette zur Umfelderkennung vorgestellt. Sie basieren auf aus der Literatur bekannten Verfahren, die auf die Problemstellung der Erkennung von Verkehrselementen übertragen und gegebenenfalls erweitert werden. Eigene Beiträge sind insbesondere die Laufzeitoptimierung der Hough-Transformation für Rechtecke, die Entwicklung der blockweisen diskreten Kosinustransformation als Merkmalstransformation, die Entwicklung der hierarchischen SVM zur beschleunigten Lösung von Mehrklassenproblemen basierend auf Standard-SVMs und die Entwicklung der TPCB-ADR Auswahlstrategie im Rahmen des aktiven Lernens. Basierend auf konkreten Parametrierungen der modularen Prozesskette durch die vorgestellten Verfahren werden in Abschnitt 3.5 Anwendungen zur Erkennung verschiedener Arten von Verkehrselementen präsentiert.

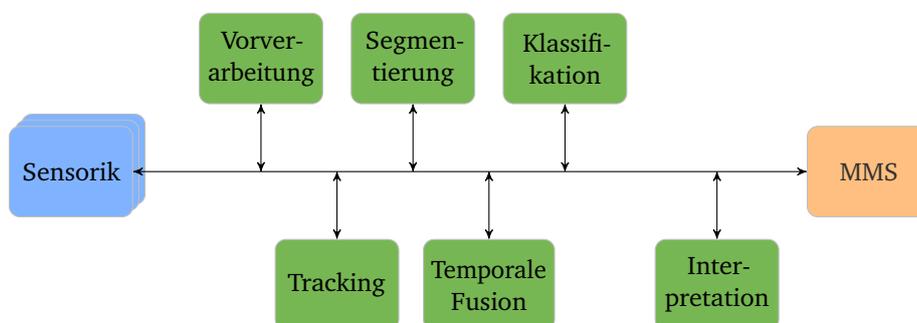


Abb. 3.1.: Prozesskette zur Situationserfassung und -interpretation.

3.1. Sensorik

Eine Vielzahl unterschiedlicher Sensoren wird bereits in Serienfahrzeugen eingesetzt. Weitere Sensoren werden im Labor und in Versuchsfahrzeugen erprobt; abhängig von ihrem Preis und ihren Eigenschaften ist ein zukünftiger Einsatz in Serienfahrzeugen zu erwarten. Ausgehend von den in Abschnitt 1.2 definierten Anforderungen werden fünf Kriterien zur Beurteilung der Eignung verschiedener Sensorsysteme benutzt: Der Preis des Sensorsystems in Hinblick auf einen möglichen Einsatz in Serienfahrzeugen sowie die Eignung des Sensorsystems zur Bestimmung von Textur, Form, Relativ- und Absolutposition von Verkehrselementen. Die Bewertung erfolgt in fünf Abstufungen, für die folgende Symbole verwendet werden:

Symbol	Eignung	Preis
★★★★	sehr gut	im Serieneinsatz
★★★☆☆	gut	niedriger Preis
★★☆☆☆	eingeschränkt	derzeit noch recht hoher Preis, bei dem aber durch Massenproduktion oder technologische Fortschritte mittelfristig mit einem Einsatz in Serienfahrzeugen zu rechnen ist
★☆☆☆☆	mangelhaft	hoch
☆☆☆☆☆	nicht geeignet	sehr hoch

Stellvertretend für die einzelnen Sensorsysteme wurden aktuell am Markt erhältliche Sensoren ausgewählt, um die Eignung zu bestimmen. Videokameras und Laserscanner wurden jeweils in zwei Unterklassen (Monochrom- und Farbvideo bzw. Laserscanner mit wenigen und vielen Zeilen) aufgeteilt, da sich deren Eignung in einzelnen Kriterien unterscheidet. Die ausgewählten Sensoren sind in Tabelle 3.1 dargestellt.

Die Eignung dieser Sensoren in den einzelnen Kriterien ist in Tabelle 3.2 zusammengefasst. Wie der Tabelle zu entnehmen ist, können nur Kamerasysteme für sich alleine genommen alle gewünschten Informationen abdecken. Trägheitsnavigationssystem und Laserscanner erfüllen aufgrund ihres Preises die Anforderungen derzeit noch nicht. Für die Bestimmung der absoluten Position mittels Kameras können Verfahren wie Visual SLAM eingesetzt werden, die aber ein Forschungsfeld für sich darstellen und im Rahmen der Arbeit nicht untersucht werden sollen. Die Kombination von Farbvideokamera und GPS-Sensor dagegen kann alle gewünschten Kriterien zufriedenstellend abdecken. Der zusätzliche Einsatz einer Tof-Kamera (insbesondere in Form einer 2D/3D-Fusion) oder eines Stereo-Kamerasystems ist denkbar, lässt aber keinen deutlichen Informationsgewinn erwarten. Für Anwendungen, die keine Absolutposition von Elementen benötigen, ist der Einsatz einer Farbvideokamera ausreichend.

Typ	Hersteller Modell	Abbildung
Monochromvideo	Point Grey Research Dragonfly 2 DR2-HIBW-CS	
Farbvideo	Point Grey Research Dragonfly 2 DR2-HICOL-CS	
GPS	Navilock NL-302U	
Trägheitsnavigationssystem	Oxford Technical Solutions RT3003	
Time of flight	PMD Technologies PMD[vision]® CamCube 3.0	
Laser (wenige Zeilen)	Ibeo Automotive ibeo LUX	
Laser (viele Zeilen)	Velodyne HDL-32E	

Tab. 3.1.: Ausgewählte Vertreter einzelner Sensorsysteme.

Bildsignale

Ein Bildsignal kann als Funktion $g(\mathbf{x})$ mit

$$g : \mathbb{R}^2 \rightarrow \mathbb{R}^Q \quad [3.1]$$

beschrieben werden mit $Q \in \mathbb{N}$ der Anzahl Bildkanäle. Durch den Aufnahmeprozess findet eine Digitalisierung sowohl im Orts- als auch im Wertebereich statt. Der Ortsbereich wird auf einen festen Ausschnitt beschränkt und der Wertebereich eingeschränkt. Sei $\mathbb{N}_k := \mathbb{N}/k\mathbb{N}$ eine abkürzende Schreibweise für den Modul k auf den natürlichen Zahlen $\mathbb{N} := \{0, 1, 2, 3, \dots\}$. Ein Farbraum $\mathfrak{F} \in \mathbb{N}_k^n$ ist ein n -dimensionaler Modul k . Er repräsentiert den Wertebereich des Bildsignals. Neben dem Wertebereich ist die Interpretation von Werten im Farbraum von Bedeutung. Für gängige Farbräume werden die folgenden Abkürzungen

	Textur	Form	Position relativ	Position absolut	Preis
Monochromvideo	★★★★☆	★★★★★	★★☆☆☆	★★☆☆☆	★★★★★
Farbvideo	★★★★★	★★★★★	★★☆☆☆	★★☆☆☆	★★★★☆
GPS	☆☆☆☆☆	☆☆☆☆☆	★★☆☆☆	★★★★☆	★★★★★
Trägheitsnavigationssystem	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	★★★★★	★★☆☆☆
Ultraschall	☆☆☆☆☆	★★☆☆☆	☆☆☆☆☆	☆☆☆☆☆	★★★★★
Radar	☆☆☆☆☆	★★☆☆☆	★★☆☆☆	☆☆☆☆☆	★★★★★
Time of flight	★★☆☆☆	★★☆☆☆	★★★★☆	☆☆☆☆☆	★★☆☆☆
Laser (wenige Zeilen)	☆☆☆☆☆	★★☆☆☆	★★★★☆	★★☆☆☆	★★☆☆☆
Laser (viele Zeilen)	☆☆☆☆☆	★★★★☆	★★★★★	★★☆☆☆	★★☆☆☆

Tab. 3.2.: Eignung von Sensorsystemen gemäß der Anforderungen an die Extraktion von Informationen und über den Preis.

verwendet:

$\mathfrak{M} := \mathbb{N}_1^1$	Binär-Raum
$\mathfrak{B} := \mathbb{N}_{255}^1$	Bayer-Raum
$\mathfrak{G} := \mathbb{N}_{255}^1$	Grauwert-Raum
$\mathfrak{R} := \mathbb{N}_{255}^3$	RGB-Raum
$\mathfrak{H} := \mathbb{N}_{255}^3$	HSV-Raum

Ein Bild $\mathbf{B} \in \mathfrak{F}^{h,w}$ mit Breite w und Höhe h ist eine $h \times w$ Matrix über \mathfrak{F} . Seine Komponenten $b_{ij} \in \mathfrak{F}$ werden Bildpunkte oder Pixel genannt. Zusätzlich wird die Notation

$$b(\mathbf{x}) := b_{h-x_2, x_1} \quad \mathbf{x} \in \mathbb{N}_w \times \mathbb{N}_h \quad [3.2]$$

eingeführt, um der in der Bildverarbeitung verbreiteten Konvention Rechnung zu tragen, den Bildursprung in die linke untere Ecke zu legen.

Eine Blockmatrix $\mathbf{R}(a : b ; c : d)$ zu \mathbf{B} stellt einen rechteckigen Bildausschnitt dar. Sie wird aus \mathbf{B} gebildet, indem eine neue Matrix aus den Reihen a bis b und den Spalten c bis d gebildet wird.

Farbvideokameras liefern Bilder \mathbf{B} der Größe $w \times h$. Tabelle 3.3 stellt die Eigenschaften verbreiteter Bildkonventionen zusammen, von denen die in Tabelle 3.1 dargestellte Dragonfly die Farbräume \mathfrak{G} , \mathfrak{B} und \mathfrak{R} als Ausgabe unterstützt. Für alle Farbräume berechnet sich die Speichermenge eines Pixels d als $d = nk$ mit Ausnahme des Bayer-Farbraums: Hier kann der Farbwert für einen Bildpunkt nicht aus den einzelnen Kanälen berechnet werden, sondern muss mittels Interpolation aus den umliegenden Bildpunkten ermittelt werden – die einzelnen Bildpunkte sind abwechselnd für unterschiedliche Farben (rot, grün und blau) empfindlich.

Preisgünstige GPS-Sensoren liefern georeferenzierte Positionen (geographische Breite, Länge und Höhe über dem Meeresspiegel) mit einer Frequenz von etwa 1 Hz. Die Genauigkeit weicht um mehrere Grö-

Farbraum	Kurzform	n	k	d
Binär	\mathfrak{B}	1	1	1 bit
Monochrom	\mathfrak{G}	1	8	8 bit
Bayer	\mathfrak{B}	1	8	24 bit
RGB	\mathfrak{R}	3	8	24 bit
HSV	\mathfrak{H}	3	8	24 bit

Tab. 3.3.: Eigenschaften verbreiteter Bildkonventionen.

ßenordnungen von der Genauigkeit von Trägheitsnavigationssystemen ab. Insbesondere in Innenstädten weicht die Positionsschätzung typischerweise im Dekameterbereich von der tatsächlichen Position ab. Neben Positionsdaten sind abgeleitete Daten wie Kompassrichtung und Geschwindigkeit verfügbar. Bei niedriger Geschwindigkeit ist die Schätzung der Kompassrichtung oft ungenau. Aufgrund der niedrigen Aktualisierungsrate ist sie bei starker Beschleunigung ungenau bzw. verzögert.

3.2. Segmentierung und Detektion

Statische Verkehrselemente bestehen aus zusammenhängenden Teilen und werden bei der Abbildung durch eine Videokamera aufgrund der Projektionseigenschaften auf einen zusammenhängenden Bildbereich abgebildet. Die Bestimmung solcher Bereiche ist das Ziel der Segmentierung. Sie werden als ROIs bezeichnet.



Definition 3.1. Eine region of interest (ROI) $\Theta^{\mathbf{B}}$ zum Bild \mathbf{B} ist eine nichtleere Teilmenge von $\Theta_{\cup}^{\mathbf{B}} := \{\mathbf{x} \mid \mathbf{x} \in \mathbf{B}\}$. Das i -te Element von $\Theta^{\mathbf{B}}$ ist $\Theta^{\mathbf{B}}(i)$ mit $1 \leq i \leq |\Theta^{\mathbf{B}}|$. Abkürzend wird statt $\Theta^{\mathbf{B}}$ auch Θ verwendet, wenn das zugehörige Bild \mathbf{B} aus dem Kontext ersichtlich ist.

Neben der expliziten Aufzählung der Punkte einer ROI Θ ist es möglich, sie parametrisch über geometrische Primitive anzugeben. Beschreibt etwa $K = (\mathbf{c}, r)$ einen Kreis mit Mittelpunkt \mathbf{c} und Radius r , dann ist $\Theta(K) := \{\mathbf{x} \mid \mathbf{x} \in \mathbf{B} \wedge \|\mathbf{c} - \mathbf{x}\| \leq r\}$ eine parametrische ROI.

Zum Umgang mit ROIs werden häufig Operationen wie Farbraumtransformation, Umwandlung in ein Bild und Größenskalierung benötigt. Zur Farbraumtransformation reicht es aus, das Bezugsbild zu ändern: Wenn $\mathbf{B}_2 \in \mathfrak{F}_2^{h,w}$ eine Farbraumtransformation von $\mathbf{B}_1 \in \mathfrak{F}_1^{h,w}$ ist, dann entspricht $\Theta^{\mathbf{B}_2}$ einer Farbraumtransformation der ROI $\Theta^{\mathbf{B}_1}$. Zur Umwandlung in ein Bild wird zu einer ROI $\Theta^{\mathbf{B}}$ der kleinste Bildausschnitt von \mathbf{B} gewählt, der Θ vollständig enthält.



Definition 3.2. Sei $\Theta^{\mathbf{B}}$ eine ROI zum Bild $\mathbf{B} \in \mathfrak{F}^{h,w}$. Der kleinste Bildausschnitt von \mathbf{B} , der Θ vollständig enthält, wird als \mathbf{B}_{Θ} bezeichnet mit

$$\mathbf{B}_{\Theta} := \mathbf{B}(\min_{\mathbf{x} \in \Theta^{\mathbf{B}}} (h - x_2) : \max_{\mathbf{x} \in \Theta^{\mathbf{B}}} (h - x_2); \min_{\mathbf{x} \in \Theta^{\mathbf{B}}} (x_1) : \max_{\mathbf{x} \in \Theta^{\mathbf{B}}} (x_1)) \quad [3.3]$$

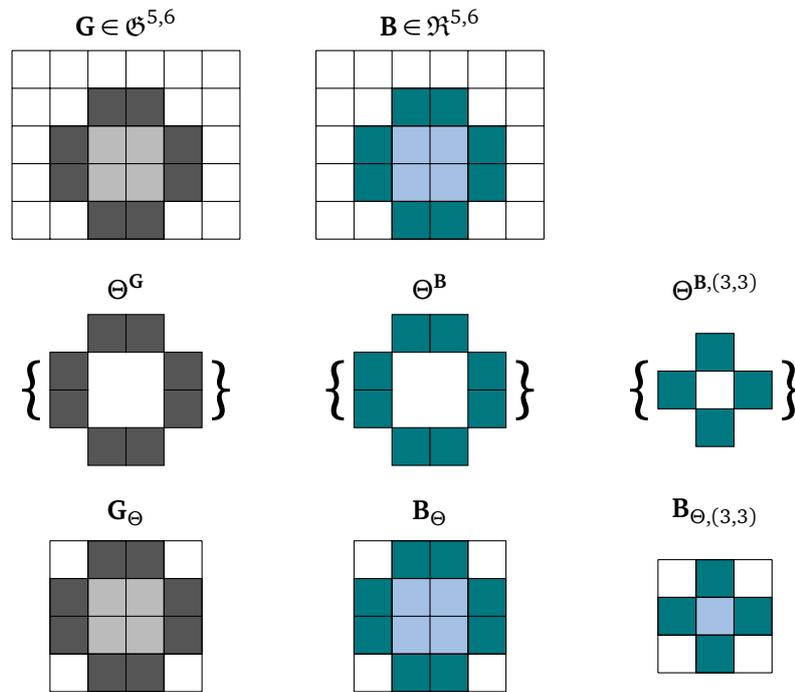


Abb. 3.2.: Veranschaulichung von ROI-Operationen. Das Bild \mathbf{G} geht durch Grauwerttransformation aus \mathbf{B} hervor. Die ROI $\Theta^{\mathbf{B}}$ stellt eine mögliche Segmentierung von \mathbf{B} dar. Alle anderen ROIs und Bilder sind Ergebnisse von Operationen auf $\Theta^{\mathbf{B}}$.

 und $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.

Die Größenskalierung einer ROI sei wie folgt definiert:

Definition 3.3. Sei $\Theta^{\mathbf{B}}$ eine ROI zum Bild \mathbf{B} und $\mathbf{C} \in \mathfrak{F}^{v,u}$ das auf die Größe $u \times v$ skalierte Bild \mathbf{B}_Θ . Sei $r(\mathbf{x}, u, v)$ die Menge der Punkte, auf die $\mathbf{x} \in \mathbf{B}_\Theta$ in \mathbf{C} abgebildet wird. Dann ist

$$\Theta^{\mathbf{B},(u,v)} := \bigcup_{\mathbf{x} \in \Theta^{\mathbf{B}}} r(\mathbf{x}, u, v) \quad [3.4]$$

 die auf die Größe $u \times v$ skalierte ROI zu $\Theta^{\mathbf{B}}$. Analog sei $\mathbf{B}_{\Theta,(u,v)}$ das auf die Größe $u \times v$ skalierte Bild \mathbf{B}_Θ .

Abbildung 3.2 veranschaulicht die verschiedenen Operationen auf ROIs. Im Folgenden werden Verfahren zur Segmentierung und Detektion vorgestellt. Terminologie und Notation sind an Beyerer, Puente León und Frese angelehnt [BPF12]. Die Segmentierung bestimmt eine Menge von ROIs. Dazu muss die Anzahl und die Position aller Objekte sowie die Abbildung der Objekte auf Bildpixel unter den gegebenen Aufnahmebedingungen bestimmt werden.



Abb. 3.3.: Segmentierung von rot gemäß der Schwellwertentscheidung [3.5].

3.2.1. Bildpunktbasierte Segmentierung

Viele Algorithmen identifizieren die Anzahl und Position von Objekten implizit, indem sie Bildpixel bestimmen, die festgelegte Kriterien erfüllen und daraus ROIs bilden. Bei diesem Vorgehen müssen die gebildeten ROIs anschließend auf mögliche Objektverdeckungen (mehrere ROIs gehören zu einem Objekt) und benachbarte Objekte (eine ROI bildet mehrere Objekte ab) untersucht werden.



Definition 3.4. Eine vollständige Segmentierung teilt Θ_U^B in n ROIs $\Theta_1, \dots, \Theta_n$ auf, so dass gilt $\bigcup_{i=1}^n \Theta_i = \Theta_U^B$.

Um zuverlässige Ergebnisse zu erhalten, muss das im Abbildungsprozess immer vorhandene Rauschen berücksichtigt werden. Verbreitete Verfahren stützen sich auf charakteristische Eigenschaften der Objekte. Dies kann ihre Farbe, ihre Textur, ihre Form oder eine Kombination dieser Eigenschaften sein. Bildpunkt-basierte Verfahren betrachten jeden Bildpunkt isoliert und bestimmen aufgrund seiner Eigenschaften – Farbe oder Grauwert – die Objektzugehörigkeit.

Ein einfaches Verfahren zur Segmentierung stellt eine Schwellwertfunktion dar. Gegebenenfalls kombiniert mit einer Farbraumtransformation findet es verbreiteten Einsatz als Segmentierungsverfahren für Verkehrszeichen [MKS00; LLX02; ST02; TB04; VR05; Shn05; AJ06; SD06; Bro+07; KKY07; MB+07; PC07; XK08; Kir+09; TZG09; LY10; RLL10; Che+11; Jia+11; SSD12; Zhe+12; CGL13]. Wissen über den Aufbau des zugrundeliegenden Farbraums oder Trainingsbeispiele können genutzt werden, um manuell eine Schwellwertfunktion zu definieren. Beispielsweise gelingt mit dem Kriterium

$$\begin{aligned} h(\mathbf{x}) \text{ ist falsch} &\Rightarrow \mathbf{x} \in \Theta_1 \\ h(\mathbf{x}) \text{ ist wahr} &\Rightarrow \mathbf{x} \in \Theta_2 \end{aligned} \tag{3.5}$$

mit $h(\mathbf{x}) := (h_h(\mathbf{x}) < 10 \vee h_h(\mathbf{x}) > 320) \wedge h_s(\mathbf{x}) > 120 \wedge h_v(\mathbf{x}) > 40$ die Segmentierung der roten Farbe des Verkehrszeichens in Abbildung 3.3. Aufgrund wechselnder Beleuchtung und Alterung bzw. Beschädigung von Verkehrselementen ist das manuelle Vorgehen meist nicht optimal. Eine Verbesserung erhält man, indem man die Segmentierung als Klassifikationsproblem $s : \mathfrak{F} \rightarrow \Omega = \{\omega_0, \dots, \omega_{n-1}\}$ auffasst, das

Bildpunkte als Merkmale betrachtet und auf eine Menge von Klassen Ω abbildet. Die Segmentierung wird dann durch das Kriterium

$$s(\mathbf{x}) = \omega_i \Rightarrow \hat{\Theta}(\mathbf{x}) := \Theta_i \quad [3.6]$$

gelöst. Die Entscheidungsfunktion s kann mit einem beliebigen überwachten Lernverfahren wie etwa der in Abschnitt 3.3.6 beschriebenen SVM gelernt werden. Als Lernbeispiele werden zu Hintergrundobjekten gehörige Bildpunkte für ω_0 und zu Vordergrundobjekten gehörige Bildpunkte für die positiven Klassen $\omega_1, \dots, \omega_{n-1}$ benutzt. Im Hinblick auf die Echtzeitverarbeitung muss beachtet werden, dass jeder Bildpunkt klassifiziert werden muss, so dass s entsprechend schnell berechenbar sein muss.

Die gemäß einer vollständigen Segmentierung gebildeten ROIs Θ_i können zu mehreren Objekten gehören (vergleiche Abbildung 3.3). Für die weitere Verarbeitung ist es aber wünschenswert, dass jedes Θ_i genau einem Objekt entspricht. Da Verkehrselemente (wenn sie nicht teilweise verdeckt sind) auf zusammenhängende Bildbereiche abgebildet werden, eignet sich dafür eine zusammenhängende Segmentierung.



Definition 3.5. Eine *vollständige, zusammenhängende Segmentierung* ist eine vollständige Segmentierung, bei der alle ROIs zusammenhängend sind.

Eine vollständige Segmentierung kann durch die Bildung von Zusammenhangskomponenten [SB85] zu einer vollständigen, zusammenhängenden Segmentierung erweitert werden.

Eine Farbsegmentierung eignet sich vor allem für großflächige Verkehrselemente wie Vorwegweiser (vergleiche Abbildung 3.4). Bei Objekten geringer Größe im Bild treten durch die Aufnahme Diskretisierungseffekte auf, die keine robuste Farbsegmentierung erlauben. Falls kleine Verkehrselemente bzw. Verkehrselemente in größerer Entfernung segmentiert werden sollen, muss daher eine Kombination mit anderen Segmentierungsverfahren eingesetzt werden und die merkmalsbasierte Klassifikation so parametrisiert werden, dass nur schwach gefiltert wird: Die Falschnegativrate wird auf Kosten einer hohen Falschpositivrate niedrig gehalten.

Aufgrund der weitgehenden Unabhängigkeit von der Umgebungsbeleuchtung sind Kanten geeignete Primärmerkmale für die Segmentierung bzw. Detektion. Die Form von Verkehrselementen entspricht im Kantenbild perspektivisch verzerrten geometrischen Grundformen (Dreieck, Quadrat, Viereck, Achteck, Kreis). Ein Verfahren zur modellbasierten Segmentierung von Linien wurde bereits 1962 von Paul V. C. Hough vorgestellt [Hou62] und später nach ihm *Hough-Transformation* genannt. Duda und Hart [DH72] sowie Ballard [Bal81] erweiterten das Verfahren zur *Generalisierten Hough-Transformation* zur Erkennung beliebiger Formen. Die Generalisierte Hough-Transformation zeichnet sich durch ihre Robustheit aus, ist aufgrund des hohen Laufzeitaufwands jedoch nicht direkt im Rahmen dieser Arbeit einsetzbar. Im Folgenden werden geeignete Erweiterungen bzw. Spezialisierungen beschrieben, die zur Segmentierung von Verkehrselementen geeignet sind.

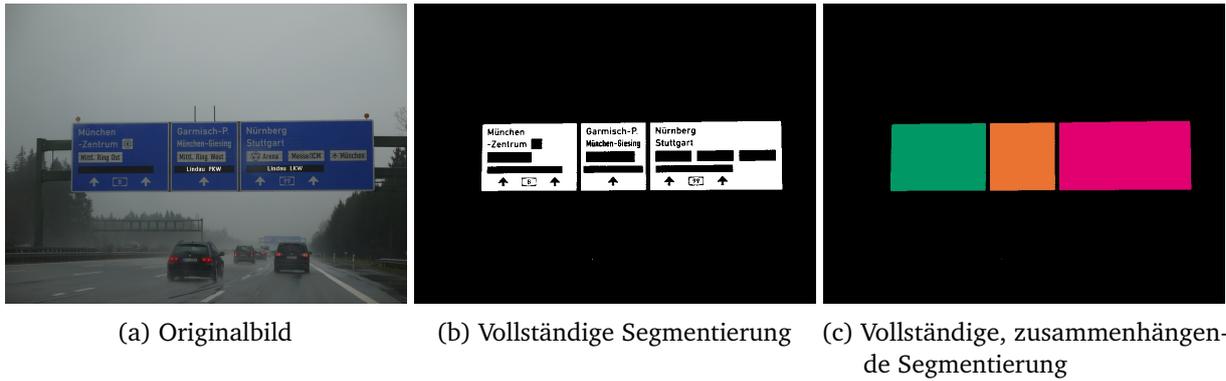


Abb. 3.4.: Vollständige Segmentierung (b) des Originalbilds (a) gemäß einer Schwellwertentscheidung analog zu [3.5] und vollständige, zusammenhängende Segmentierung (c) durch Extraktion von Zusammenhangskomponenten nach [SB85]. In (b) und (c) sind die zwei bzw. vier ROIs jeweils durch unterschiedliche Farben markiert.

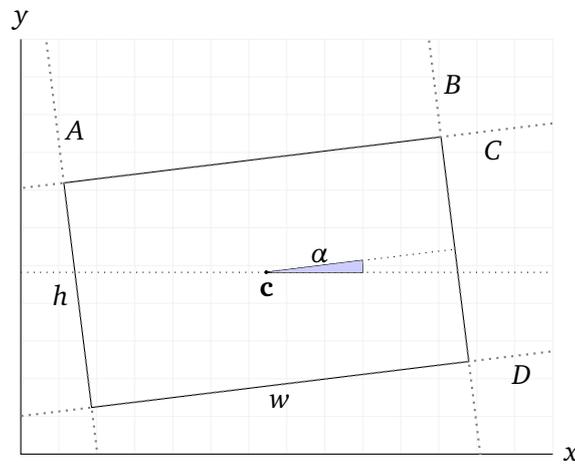


Abb. 3.5.: Rechteck-Modell

3.2.2. Hough-Transformation für Rechtecke

Das Tupel $R = (c, w, h, \alpha)$ modelliert wie in Abbildung 3.5 dargestellt ein Rechteck R mit Mittelpunkt c , Größe $w \times h$ und Rotation α . Die direkte Bestimmung möglicher Rechtecke in einem Kantenbild mittels Generalisierter Hough-Transformation ist zu aufwendig für eine Echtzeitverarbeitung. Daher wird aufbauend auf der Arbeit von Jung [JS04] eine mehrstufige Hough-Transformation mit stark reduzierter Laufzeit eingesetzt [Nie+10]. Sie führt eine Hough-Transformation für Geraden durch und bestimmt anschließend anhand geometrischer Kriterien mögliche Rechtecke.

Punkte (x, y) auf einer Geraden erfüllen für feste Werte $\alpha, r \in \mathbb{R}$ die Bedingung

$$r = x \cos(\alpha) + y \sin(\alpha).$$

Durch eine Faltung des zu analysierenden Grauwertbildes G mit dem als Scharr-Operator (nach [Sch00])

bekanntem Filterkern \mathbf{S} mit

$$\mathbf{S} := \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \quad [3.7]$$

erhält man die Gradientenfelder \mathbf{G}_x in x - und \mathbf{G}_y in y -Richtung:

$$\begin{aligned} \mathbf{G}_x &:= \mathbf{S} * \mathbf{G} \\ \mathbf{G}_y &:= \mathbf{S}^\top * \mathbf{G}. \end{aligned} \quad [3.8]$$

Die Transformation des Bildpunkts $g(\mathbf{x})$ in den Hough-Raum erfolgt mittels

$$\begin{aligned} h &: \mathbb{N}^2 \rightarrow \mathbb{R}^2 \\ h(\mathbf{x}) &:= \begin{pmatrix} \alpha \\ r \end{pmatrix} \end{aligned} \quad [3.9]$$

mit

$$\begin{aligned} \alpha &:= \tan\left(\frac{g_y(\mathbf{x})}{g_x(\mathbf{x})}\right) \\ r &:= \mathbf{x}^\top \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}. \end{aligned} \quad [3.10]$$

Zur Durchführung der Hough-Transformation wird ein Akkumulator A verwendet. Die Bildpunkte des Grauwertbildes werden traversiert und, falls es sich um ein Kantenpixel handelt, gemäß Formel [3.9] in den Hough-Raum transformiert und der Akkumulator an der entsprechenden Stelle inkrementiert:

$$A(h(\mathbf{x})) = A(h(\mathbf{x})) + \xi(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbf{G}.$$

Dabei ist $\xi > 0$ eine Gewichtungskonstante. Sie kann anhand der Kantenstärke aus \mathbf{G}_x und \mathbf{G}_y abgeleitet werden. Robustere Ergebnisse erhält man dabei, wenn man schwache Kanten ausschließt. Dazu eignet sich insbesondere das Canny-Verfahren [Can86] als robuster Algorithmus zur Kantendetektion. Nach Bestimmung des Canny-Bildes \mathbf{C} aus \mathbf{G} setzt man $\xi(\mathbf{x}) := c(\mathbf{x})$. Als weitere Optimierung kann man Kantenpixel ausschließen, deren Gradientenrichtung aus Anwendungssicht unmöglich sind.

Abschließend werden Häufungsstellen im Akkumulator bestimmt, deren Positionen $\begin{pmatrix} \alpha & r \end{pmatrix}^\top$ im Akkumulator die Parameter der von ihnen repräsentierten Geraden darstellen. Mittels einer Analyse der Maximalstellen wird die Anzahl der Geraden bestimmt.

Kombinationen von jeweils vier Geraden (A,B,C,D) sind Rechteckkandidaten R . Falls (A,B) und (C,D) parallel sind und (A,C) orthogonal zueinander stehen, handelt es sich um ein Rechteck. Für n von der Hough-Transformation bestimmte Geraden ergeben sich kombinatorisch $\binom{n}{4}$ mögliche Rechteckkandidaten. Die Untersuchung aller Kombinationen hat wegen $\binom{n}{4} = 1/24n(n-1)(n-2)(n-3) \leq n^4$ eine

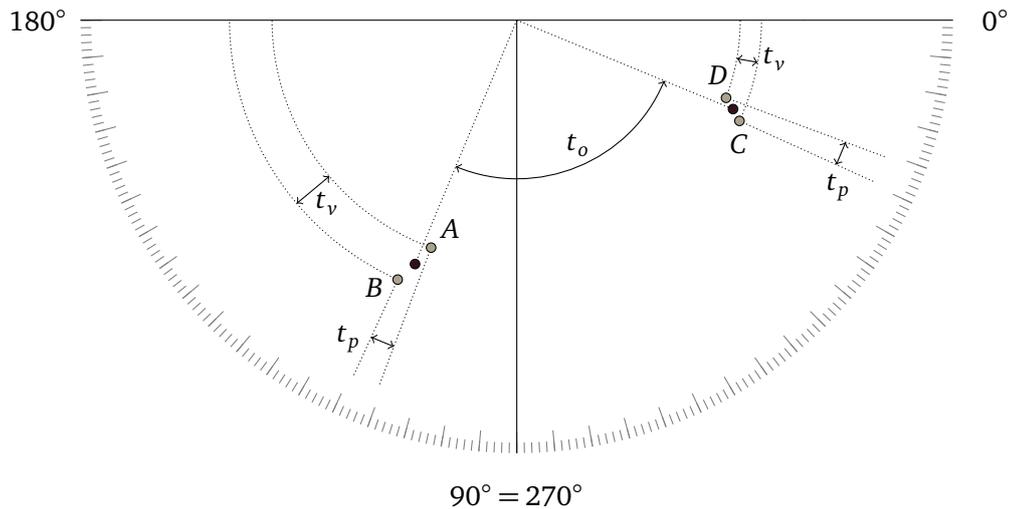


Abb. 3.6.: Illustration der Bewertung eines Rechteckkandidaten $R = (A, B, C, D)$ gemäß $s(R)$ [3.11]. Die Darstellung verwendet Polarkoordinaten (ν, α) . Die hellen Punkte markieren die Seitenlinien, die dunklen Punkte die Mittlung von Winkel α und Wert im Hough-Raum ν der parallelen Seiten.

asymptotische Laufzeit von $\mathcal{O}(n^4)$. Anstelle der naiven Untersuchung aller Kombinationsmöglichkeiten kann man Geraden nach Winkel sortieren und solche mit ähnlichen Winkeln als mögliche parallele Seiten gruppieren. Anschließend sortiert man die parallelen Seiten erneut und kombiniert solche, die einen Winkelunterschied von etwa $\pi/2$ haben. Alle anderen Kombinationsmöglichkeiten werden verworfen. Dadurch ändert sich zwar der asymptotische Laufzeitaufwand nicht, aber die zu untersuchende Anzahl von Kombinationen ist in praktisch relevanten Fällen gering genug, um den Echtzeiteinsatz des Algorithmus zu gewährleisten.

Bei der Beurteilung von Parallelität und Orthogonalität muss berücksichtigt werden, dass die ermittelten Modellparameter verrauscht sind. Dafür eignet sich die Berechnung eines Fehlermaßes $s(R)$, welches Abweichungen von perfekter Parallelität, Orthogonalität und gleicher Werte im Hough-Raum gewichtet kombiniert (vergleiche Abbildung 3.6)

$$s(R) := \mu_p t_p + \mu_o t_o + \mu_\nu t_\nu. \quad [3.11]$$

Dabei ist t_p das Fehlermaß für die Parallelität von (A, B) und (C, D) mit

$$t_p := \frac{|\alpha_A - \alpha_B|}{\alpha_A + \alpha_B} + \frac{|\alpha_C - \alpha_D|}{\alpha_C + \alpha_D}. \quad [3.12]$$

Die Orthogonalität geht ein als t_o mit

$$t_o := \left| \frac{\alpha_A + \alpha_B}{2} - \frac{\alpha_C + \alpha_D}{2} \right| - \frac{\pi}{2} \quad [3.13]$$

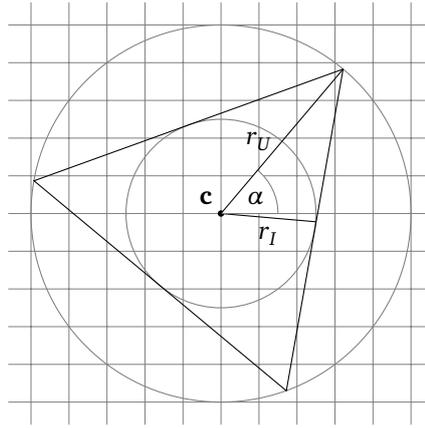


Abb. 3.7.: Geometrisches Modell eines n-Ecks für n=3 mit Mittelpunkt \mathbf{c} , Inkreisradius r_I , Umkreisradius r_U und Rotation α . Die Rotation gibt den kleinsten Winkel zwischen x -Achse und einem Seitenpunkt im Koordinatensystem mit Ursprung \mathbf{c} an. Damit gilt $0 \leq \alpha < \frac{2\pi}{n}$.

und die Ähnlichkeit der Werte im Hough-Raum von parallelen Seiten geht ein als t_v mit

$$t_v := \frac{|v_A - v_B|}{v_A + v_B} + \frac{|v_C - v_D|}{v_C + v_D}. \quad [3.14]$$

Dabei gibt α_I den Winkel der Geraden I an und v_I seinen Wert im Hough-Raum. Bei μ_p , μ_o und μ_v handelt es sich um Gewichte.

Rechteckkandidaten R mit hohem Fehlerwert $s(R) > \kappa_s$ werden entfernt. Abschließend werden Duplikate entfernt. Dazu wird ein Ähnlichkeitsmaß $c(R_1, R_2)$ benötigt, das die Abweichung der Modellparameter der Rechteckkandidaten R_1 und R_2 bewertet:

$$c(R_1, R_2) := \mu_\alpha |\alpha_{R_1} - \alpha_{R_2}| + \mu_c \|\mathbf{c}_{R_1} - \mathbf{c}_{R_2}\| + \mu_s \frac{|w_{R_1} h_{R_1} - w_{R_2} h_{R_2}|}{w_{R_1} h_{R_1} + w_{R_2} h_{R_2}} \quad [3.15]$$

mit Gewichten μ_α , μ_c und μ_s . Gilt $c(R_1, R_2) < \kappa_c$, dann wird der schlechter bewertete Kandidat $\text{argmax}(s(R_1), s(R_2))$ entfernt. Geeignete Schwellwerte κ_c und κ_s können empirisch bestimmt werden. Übrig bleibende Rechteckkandidaten stellen das Ergebnis der Segmentierung als parametrische ROIs dar.

3.2.3. Radiale Symmetrie für gleichseitige n-Ecke

Loy, Barnes und Zelinsky entwickelten ein Verfahren zur Detektion gleichseitiger n-Ecke [BZ04; LB04]. Es ähnelt in der Vorgehensweise einer Hough-Transformation, indem Punkte aus dem Kantenbild auf Strecken in Dualräumen projiziert werden [Yue+90]. Die Abbilder erhöhen die Werte in den jeweiligen Dualräumen (Akkumulatoren). In dem von Barnes und Zelinsky vorgeschlagenen Verfahren kommen drei Dualräume zum Einsatz. Zwei dieser Dualräume sind dreidimensional, wobei die ersten beiden Dimensionen denen des Kantenbildes entsprechen. Die dritte Dimension repräsentiert den Inkreisradius

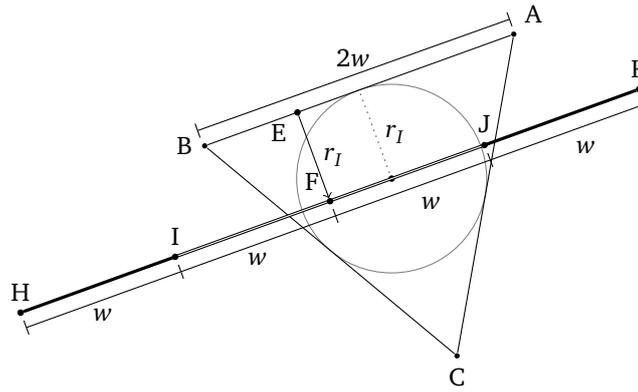


Abb. 3.8.: Transformation eines Kantenschnittpunkts E im Dreieck (A, B, C) in den Dualraum für den Radius r_I . Für jeden der Punkte $\mathbf{x} \in \overline{IJ}$ wird im Dualraum der Punkt $(x, y, r_I)^T$ um den Wert 1 erhöht, während für Punkte aus \overline{HI} und \overline{JK} der Wert um 1 verringert wird. In der Grafik nicht dargestellt ist die Gegenrichtung des Gradienten.

des n -Ecks. Der Inkreis ist der größte Kreis, der vollständig im n -Eck liegt (vergleiche Abbildung 3.7). Ein Punkt in diesen Dualräumen gibt für ein n -Eck ($n > 2$ beliebig, aber fest) den Mittelpunkt und Radius seines Inkreises an. Die beiden Dualräume unterscheiden sich in ihrer Verwendung als Akkumulator: Der erste wird für jeden projizierten Streckenpunkt fest um den Wert 1 erhöht, während der zweite um einen gradientenabhängigen Wert erhöht wird. Der dritte Dualraum schließlich ergibt sich aus einer punktwisen Multiplikation der beiden ersten Dualräume gefolgt von einer Marginalisierung entlang der Inkreisradius-Dimension. Dieser Dualraum wird für die Maximumsuche (Kandidatenextraktion) eingesetzt. Da er nur zweidimensional ist, kann die Maximumsuche sehr schnell durchgeführt werden.

Experimente im Rahmen dieser Arbeit zeigten allerdings, dass dieses Verfahren nicht die gewünschte Robustheit erzielt: Durch die Marginalisierung überlagern sich im Dualraum Informationen zu verschiedenen Radii und Kantenrichtungen. Dieses Rauschen führt zu vielen Falschpositiven. Des Weiteren ist das Verfahren nicht in der Lage, die Rotation der n -Ecke zu bestimmen. Die Rotation ist eine wichtige Information, um die nachfolgende Klassifikation von Kandidaten zu erleichtern, indem eine Rotationskorrektur vorgenommen wird, zu stark rotierte Objekte verworfen werden oder eine Vorklassifikation durchgeführt wird (z.B. Unterscheidung zwischen dem deutschen Zeichen 205 "Vorfahrt gewähren", dessen Dreiecksspitze nach unten zeigt, und Gefahrzeichen mit nach oben zeigender Dreiecksspitze). Das im Folgenden präsentierte erweiterte Verfahren löst beide Probleme, ohne signifikant mehr Rechenleistung zu benötigen. Es verwendet das in Abbildung 3.7 dargestellte Modell, das ein n -Eck S_n als Tupel $S_n = (\mathbf{c}, r_I, \alpha)$ repräsentiert.

Gegeben seien das Eingabebild $\mathbf{G} \in \mathcal{O}^{h,w}$, das nach dem Verfahren von Canny [Can86] in ein Binärbild $\mathbf{C} \in \mathcal{M}^{h,w}$ transformiert wird. Die Gradientenfelder \mathbf{G}_x und \mathbf{G}_y werden gemäß Formel [3.8] aus \mathbf{G} gebildet. Für die gesuchten n -Ecke ($n > 2$ beliebig, aber fest) sei eine Menge möglicher Inkreisradii $R = \{r_0, r_1, \dots, r_m\}$ gegeben.

Die Detektion erfolgt zweistufig. Zunächst wird im ersten Schritt eine Hough-Transformation mit einem dreidimensionalen Dualraum eingesetzt, die rotationsinvariant n -Ecke bestimmt. Der Dualraum entspricht dem ersten von Loy und Barnes vorgeschlagenen Dualraum.

Für jeden Kantenpixel \mathbf{x} mit $c(\mathbf{x}) = 1$ werden die Mittelpunkte aller möglichen n -Ecke bestimmt, von denen \mathbf{x} Teil sein könnte (vergleiche Abbildung 3.8). Diese müssen eine Entfernung von $r_i \in R$ in positiver oder negativer Gradientenrichtung von E sowie einen Abstand von d in orthogonaler Gradientenrichtung mit $0 \leq d \leq w$ und

$$w := r_i \tan(\pi/n), \quad [3.16]$$

haben, da ein n -Eck eine Seitenlänge $s_n = 2w$ hat. Für $r_t \in R$ ergibt das die Menge $s(\mathbf{x}, r_t)$ mit

$$s(\mathbf{x}, r_t) := \left\{ \mathbf{x} \pm r_t \frac{\mathbf{g}_x}{\|\mathbf{g}_x\|} + w_i \frac{\mathbf{g}_x^\perp}{\|\mathbf{g}_x^\perp\|} \mid -w \leq w_i \leq w \right\} \quad [3.17]$$

und $\mathbf{g}_x := \begin{pmatrix} g_x(\mathbf{x}) \\ g_y(\mathbf{x}) \end{pmatrix}$ und $\mathbf{g}_x^\perp := \begin{pmatrix} -g_y(\mathbf{x}) \\ g_x(\mathbf{x}) \end{pmatrix}$.

Jeder Bildpunkt \mathbf{x} mit $c(\mathbf{x}) = 1$ wird für alle Radien $r_t \in R$ gemäß $s(\mathbf{x}, r_t)$ auf eine Menge von Punkten im Akkumulator transformiert und die Werte im Akkumulator inkrementiert. Um den Einfluss von zu langen Strecken zu reduzieren, wird $s(\mathbf{x}, r_t)$ wie in Abbildung 3.8 dargestellt verlängert. Punkte im Hough-Raum, die in diese Bereiche fallen, führen zum Dekrementieren der Werte im Akkumulator und filtern so Strecken, die deutlich länger als die Seitenlängen des erwarteten n -Ecks sind. Nach Abschluss der Transformation wird der Akkumulator auf Häufungspunkte untersucht.

Jeder Häufungspunkt im Akkumulator liefert die Parameter \mathbf{c} und r_I des in Abbildung 3.7 dargestellten Modells des n -Ecks. Damit ist es bis auf die Rotation α eindeutig bestimmt, welche im letzten Schritt des Verfahrens bestimmt wird. Dazu wird für alle möglichen Rotationen α_i die Anzahl der Kantenpixel $k(\alpha_i)$ bestimmt und die Rotation mit maximaler Kantenpixelzahl ausgewählt:

$$k(\alpha_i) := \sum_{\mathbf{x} \in e(\mathbf{c}, r_I, \alpha_i)} c(\mathbf{x}) \quad [3.18]$$

$$\alpha := \arg \max_{\alpha_i} k(\alpha_i)$$

mit

$$e(\mathbf{c}, r_I, \alpha_i) := \{P_{j-1} + t(P_j - P_{j-1}) \mid t \in [0, 1], j \in \{1, \dots, n\}\}$$

und

$$P_j := \mathbf{c} + \frac{r_I}{\cos(\pi/n)} \begin{pmatrix} \cos(\alpha_i + j \frac{2\pi}{n}) \\ \sin(\alpha_i + j \frac{2\pi}{n}) \end{pmatrix}.$$

Mit Hilfe des Bresenham-Algorithmus [Bre65] kann $e(\mathbf{c}, r_I, \alpha_i)$ effizient berechnet werden, wodurch auch die notwendige Diskretisierung von t vorgegeben ist. Bei der Diskretisierung von α_i ist zu beachten, dass aufgrund der Symmetrie der n -Ecke die Betrachtung von

$$0 \leq \alpha < \frac{2\pi}{n} \quad [3.19]$$

ausreichend ist. Ferner kann man gezielt eine Menge der für statische Verkehrselemente auftretenden Rotationswinkel wählen und nur solche Bildpunkte mit passenden Gradientenrichtungen transformieren [BL06]. Damit werden beispielsweise vertikale Kanten bei dreieckigen Warnhinweisschildern ignoriert.

Eine weitere wichtige Kenngröße ist der Anteil von Kantenpixeln $d(S_n)$ des nach dem obigen Verfahren bestimmten n -Ecks S_n . Er ergibt sich mit Formel [3.16] zu $d(S_n) := \frac{k(\alpha)}{2nw}$ und eignet sich als Gütemaß für das Detektionsergebnis bzw. zur direkten Ablehnung von Kandidaten mit $d(S_n) < \kappa_d$ für einen festzulegenden Schwellwert κ_d .

3.2.4. Radiale Symmetrie für Kreise

Das im vorhergehenden Abschnitt 3.2.3 beschriebene Verfahren detektiert für $n = \infty$ Kreise. Da dieser Spezialfall in der Praxis hohe Bedeutung hat, sei an dieser Stelle auf zwei für diesen Fall wichtige Vereinfachungen hingewiesen.

Für $n = \infty$ ergibt sich nach Formel [3.16] $w = 0$, d.h. die beiden Transformationsgeraden degenerieren zu zwei Punkten. Damit wird für diesen Fall das von der Transformation eingebrachte Rauschen minimiert. Gleichzeitig ist für diesen Spezialfall die Laufzeit am geringsten.

Die Einschränkung des Rotationswinkels α in Formel [3.19] erfolgt aufgrund von Mehrdeutigkeiten: Die n -Ecke $S_{n_1} = \{\mathbf{x}, r_I, \alpha\}$ und $S_{n_2} = \{\mathbf{x}, r_I, \alpha + \frac{2k\pi}{n}\}$ sehen identisch aus. Für Kreise mit $n = \infty$ folgt daraus das geometrisch anschauliche Ergebnis, dass alle Kreise gleich aussehen, die sich nur im Rotationswinkel unterscheiden. In der Detektion kann folglich die Bestimmung eines Rotationswinkels übersprungen werden.

Beide Vereinfachungen wirken sich positiv auf Robustheit und Laufzeit des Algorithmus aus. Da es für Kreise jedoch nicht möglich ist, aus der Form einen Rotationswinkel zu bestimmen, wird eine nachfolgende Klassifikation texturierter Verkehrselemente erschwert: Die in der Praxis auftretenden Rotationen von Texten und Piktogrammen durch ungenaue Anbringung kann nicht aufgrund der während der Detektion extrahierten Informationen kompensiert werden.

3.2.5. Morphologische Bildverarbeitung

Lichtsignale erscheinen im Videobild in der Regel heller als ihre Umgebung. Dies trifft insbesondere auf Ampeln zu, die zwecks guter Sichtbarkeit mit Kontrastblenden umgeben sind. Diese Eigenschaft kann man in der Segmentierung ausnutzen. Geeignete Verfahren dafür finden sich in der morphologischen Bildverarbeitung.

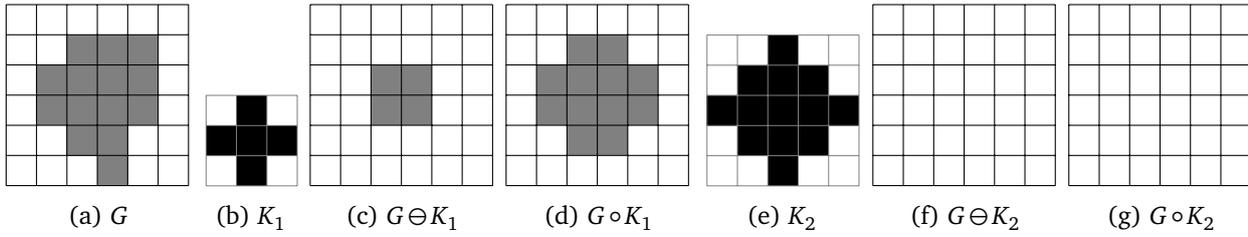


Abb. 3.9.: Beispiele für die Anwendung des morphologischen Öffnens auf ein Bild G mit einem kreisähnlichen Objekt und strukturierenden Elementen K_1 und K_2 (siehe Text).



Definition 3.6. Sei $G \in \mathcal{G}^{h,w}$ ein Grauwertbild und $K \in \mathcal{M}^{r,s}$ ein Binärbild (strukturierendes Element). Die *Erosion* \ominus ist gegeben durch

$$\begin{aligned} \ominus : \mathcal{G}^{h,w} \times \mathcal{M}^{r,s} &\rightarrow \mathcal{G}^{h,w} \\ G \ominus K &:= G_K^\ominus \\ g_K^\ominus(\mathbf{x}) &:= \min_{\substack{\mathbf{y} \in K \\ k(\mathbf{y})=1}} (g(\mathbf{x} + \mathbf{y})). \end{aligned}$$



Definition 3.7. Sei $G \in \mathcal{G}^{h,w}$ ein Grauwertbild und $K \in \mathcal{M}^{r,s}$ ein Binärbild (strukturierendes Element). Die *Dilatation* \oplus ist gegeben durch

$$\begin{aligned} \oplus : \mathcal{G}^{h,w} \times \mathcal{M}^{r,s} &\rightarrow \mathcal{G}^{h,w} \\ G \oplus K &:= G_K^\oplus \\ g_K^\oplus(\mathbf{x}) &:= \max_{\substack{\mathbf{y} \in K \\ k(\mathbf{y})=1}} (g(\mathbf{x} + \mathbf{y})). \end{aligned}$$

Basierend auf den beiden Operationen Erosion und Dilatation wird das morphologische Öffnen \circ wie folgt definiert:



Definition 3.8. Sei $G \in \mathcal{G}^{h,w}$ ein Grauwertbild und $K \in \mathcal{M}^{r,s}$ ein Binärbild (strukturierendes Element). Das *morphologische Öffnen* \circ ist gegeben durch

$$\begin{aligned} \circ : \mathcal{G}^{h,w} \times \mathcal{M}^{r,s} &\rightarrow \mathcal{G}^{h,w} \\ G \circ K &:= (G \ominus K) \oplus K. \end{aligned}$$

Ampeln erscheinen in Bildern als kreisähnliche Lichtpunkte, weisen also höhere Grauwerte als ihre Umgebung auf. Durch die Rasterung der Bildpunkte entstehen am Übergang zwischen Ampellicht und Kontrastblende Mischfarben, welche die Farbklassifikation und Extraktion der Form erschweren. Durch morphologisches Öffnen kann dieser Effekt begrenzt werden; es eignet sich darüber hinaus, Rauschen in Form von sehr kleinen Objekten mit ampelähnlichen Farben zu unterdrücken. Dazu wählt man ein strukturierendes Element geringer Größe. Abbildung 3.9 zeigt die Anwendung eines solchen

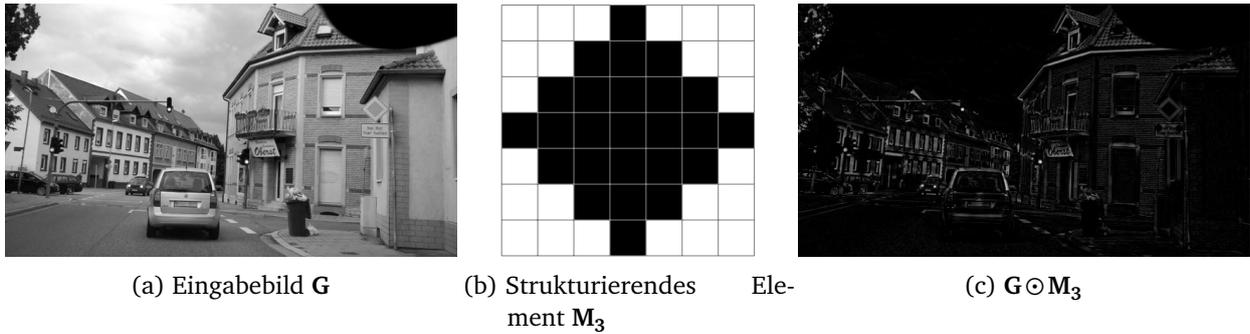


Abb. 3.10.: Anwendung des Top-Hat Operators auf das Bild (a) mit einem kreisartigen strukturierendem Element M_3 (b). Gleichmäßig texturierte großflächige Objekte werden maskiert, kleinere Objekte wie die Ampellichter bleiben unberührt.

strukturierenden Elements K_1 [3.9b] bzw. K_2 [3.9e] auf ein Bild G [3.9a] in [3.9d] und [3.9g]: Je nach Größe des strukturierenden Elements wird der Rand des Objekts an die Form des strukturierenden Elements angepasst, oder das Objekt wird vollständig maskiert, falls das strukturierende Element nicht in das Objekt hineinpasst.

Analog zum morphologischen Öffnen ist das morphologische Schließen \bullet definiert:

Definition 3.9. Sei $G \in \mathcal{G}^{h,w}$ ein Grauwertbild und $K \in \mathcal{M}^{r,s}$ ein Binärbild (strukturierendes Element). Das *morphologische Schließen* \bullet ist gegeben durch

$$\begin{aligned} \bullet : \mathcal{G}^{h,w} \times \mathcal{M}^{r,s} &\rightarrow \mathcal{G}^{h,w} \\ G \bullet K &:= (G \oplus K) \ominus K. \end{aligned}$$



Aufbauend auf dem Öffnen definiert sich der (weiße) Top-Hat-Operator (die deutsche Bezeichnung Zylinderhut ist wenig verbreitet):

Definition 3.10. Sei $G \in \mathcal{G}^{h,w}$ ein Grauwertbild und $K \in \mathcal{M}^{r,s}$ ein Binärbild (strukturierendes Element). Der (weiße) *Top-Hat-Operator* \odot ist gegeben durch

$$\begin{aligned} \odot : \mathcal{G}^{h,w} \times \mathcal{M}^{r,s} &\rightarrow \mathcal{G}^{h,w} \\ G \odot K &:= G - (G \circ K) \end{aligned}$$



wobei der Operator $-$ für die Berechnung der Differenz zweier Bilder steht. Der schwarze Top-Hat-Operator ist analog zum weißen über die Operation $(G \bullet K) - G$ definiert. Er ist hier aber nicht weiter von Bedeutung.

Die Größe von Ampellichtern im Bild kann nach oben abgeschätzt werden; großflächige Objekte in Ampelfarbe kommen als Kandidaten nicht in Frage. Der Top-Hat Operator eignet sich, solche großflächigen Hintergrundobjekte zu maskieren. Dazu wählt man ein strukturierendes Element in Form des

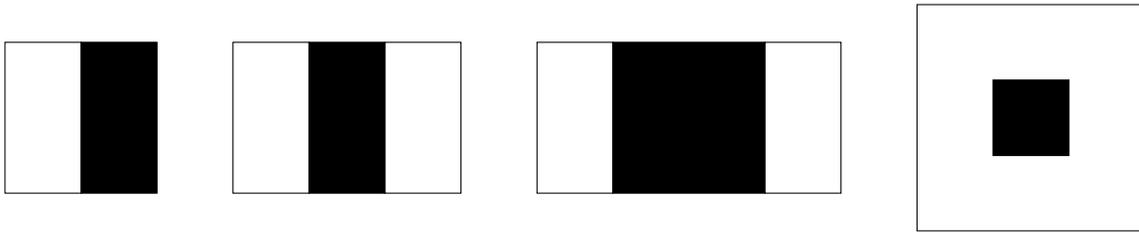


Abb. 3.11.: Haar-Basismerkmale. Durch Drehung um 45° und 90° entstehen weitere Haar-Merkmale. Angewendet auf einen Bildausschnitt berechnet ein Haar-Merkmal einen skalaren Wert als Differenz der Summe der Grauwerte im weiß markierten Bereich des Haar-Merkmals und der Summe der Grauwerte im schwarz markierten Bereich.

Ampellichts, welches größer als die maximale Größe zu erkennender Ampellichter ist. Abbildung 3.10 veranschaulicht das Vorgehen.

Für die Segmentierung von Ampellichtern lassen sich zwei morphologische Operationen einsetzen: Das morphologische Öffnen mit einem kreisförmigen strukturierenden Element geringer Größe und der weiße Top-Hat Operator mit einem kreisförmigen strukturierenden Element, das größer als die maximale Bildgröße eines Ampellichts ist.

3.2.6. Viola/Jones-Verfahren

Die bisher vorgestellten Segmentierungs- und Detektionsverfahren basieren auf explizit oder implizit vorgegebenen Objektmodellen. Eine Alternative dazu stellen maschinelle Lernverfahren dar, die anhand von Lernbeispielen und einer (potentiell sehr großen) Menge von Merkmalen eine Entscheidungsfunktion lernen. Für jede mögliche Objektansicht im Bild legt die Entscheidungsfunktion fest, ob sich an dieser Stelle ein Objekt befindet. Um mit diesem Vorgehen robuste Ergebnisse in Echtzeit zu erhalten, muss die Entscheidungsfunktion sowohl eine sehr niedrige Falschpositivrate als auch eine sehr niedrige Laufzeit aufweisen – die Anzahl möglicher Objektansichten im Bild ist extrem groß und nur ein Bruchteil davon stellen tatsächlich Objekte dar. Ein Verfahren, das dieses Vorgehen umsetzt und die Randbedingungen bezüglich Laufzeit und Robustheit erfüllen kann, wurde von Viola und Jones vorgestellt [VJ01] und nach ihnen Viola/Jones-Verfahren genannt.

Detektion

Objektansichten werden im Bild über Rechtecke mit einem vorgegebenen (beliebigen, aber festen) Ansichtsverhältnis modelliert. Im sogenannten *sliding window* Verfahren werden alle unter diesen Einschränkungen möglichen Objektansichten traversiert: Die Position und Skalierung des ansichtsfesten Rechtecks wird variiert, was für die jeweiligen Skalierungsfaktoren zum namensgebenden “Gleiten” des Rechtecks über das Bild führt. Durch zusätzliche Anwendung der in Abschnitt 3.2.7 vorgestellten Einschränkung der Suchregionen können die zu betrachtenden Skalierungsfaktoren und Positionen weiter reduziert werden.

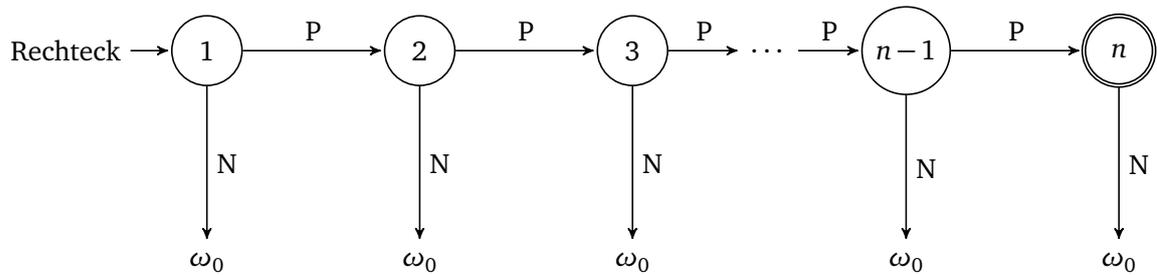


Abb. 3.12.: Schematische Darstellung eines n-stufigen Kaskaden-Klassifikators. Jede Stufe lehnt einen Bildausschnitt (Rechteck) entweder direkt ab (N) oder reicht ihn weiter zur nächsten Stufe (P). Bildausschnitte, welche die letzte Stufe erreichen und nicht abgelehnt werden, werden weiter verarbeitet.



Abb. 3.13.: Detektion eines Andreaskreuz mit dem Viola/Jones-Verfahren. Links das Ergebnis des 15-stufigen Kaskaden-Klassifikators, rechts das Ergebnis des Clustering-Verfahrens.

Innerhalb eines jeden Bildausschnitts (Rechtecks) werden Haar-Merkmale (Abbildung 3.11) ausgewertet. Sie können sehr schnell berechnet werden, wenn ein Integralbild – ein Bild, das in jedem Bildpunkt die Summe aller Grauwerte im Rechteck zwischen Bildursprung und dem Bildpunkt speichert – verwendet wird. Als weitere Beschleunigung wird eine sogenannte Kaskade verwendet, ein degenerierter Entscheidungsbaum, der in jedem Baumknoten Beispiele entweder direkt ablehnt oder zur nächsten Stufe weiterreicht (Bild 3.12). Jede Stufe stellt einen *starken Klassifikator* dar: Ein Klassifikator, der eine vorgegebene Falschpositivrate unter- und eine vorgegebene Positivrate überschreitet. Er trifft die Klassifikationsentscheidung anhand eines Schwellwertvergleichs auf der gewichteten Summe einzelner *schwacher Klassifikatoren*: Klassifikatoren, die im Schnitt nur besser als zufällige Entscheidungen sein müssen. Als schwache Klassifikatoren werden Entscheidungsbäume einer festgelegten Tiefe verwendet, die jeweils einzelne Haar-Merkmale in den Baumknoten verwenden.

Der Großteil aller möglichen Objektansichten wird durch die Kaskade abgelehnt. Die übrig bleibenden Rechtecke werden abschließend anhand ihrer Ähnlichkeit (in Bezug auf Skalierungsfaktor und Position) mittels eines Clustering-Verfahrens gruppiert. Cluster, die eine Minimalgröße nicht überschreiten, werden ignoriert. Von allen anderen Clustern wird ein repräsentatives Mitglied ausgewählt und der Ergebnismenge hinzugefügt (vergleiche Abbildung 3.13).

Training

Als überwachtes Lernverfahren benötigt die Detektion nach Viola/Jones annotierte Beispiele zum Training eines Modells. Sie werden in positive und negative Beispiele eingeteilt und sollten eine repräsentative Abbildung der Eingabedaten darstellen. Dies erreicht man durch eine möglichst große, nicht redundante Menge an Beispielen, die alle relevanten Verkehrssituationen und Umweltbedingungen widerspiegeln.

Das Training der Kaskade erfolgt sequentiell beginnend mit der ersten Stufe. Jede Stufe erhält als Eingabedaten die Trainingsbeispiele, die von der vorhergehenden Stufe nicht abgelehnt wurden. Die erste Stufe nutzt alle Trainingsbeispiele. Mittels AdaBoost [FS95] werden so lange neue Entscheidungsbäume trainiert und der Stufe hinzugefügt, bis die Entscheidungsfunktion der Stufe eine vorgegebene Falschpositivrate f unter- und eine vorgegebene Positivrate p überschreitet. Aufgrund der Filterung der Eingabedaten durch vorhergehende Stufen müssen spätere Stufen ein schwierigeres Problem lösen als frühere. Dies führt dazu, dass frühere Stufen in der Regel deutlich weniger Entscheidungsbäume ausbilden müssen und entsprechend eine geringere Laufzeit aufweisen. Für die Detektion erreicht man damit die gewünschte Eigenschaft, dass ein Großteil aller Hintergrundobjekte in einer frühen Stufe der Kaskade mit geringer Laufzeit abgelehnt wird.

Die Parameter f und p können zur Abschätzung der Falschpositivrate F und Positivrate P bezogen auf ein Bild der Trainingsmenge benutzt werden. Da jede Stufe eine Falschpositivrate von maximal f aufweist, gilt für eine Kaskade mit n Stufen

$$\begin{aligned} F &\leq f^n \\ P &\geq p^n \end{aligned} \tag{3.20}$$

Für $f = 0.25$, $p = 0.999$ und eine Kaskade mit $n = 20$ Stufen werden Objekte mit einer Wahrscheinlichkeit von mehr als $P \geq 0.999^{20} \approx 98\%$ gefunden. Wird eine Kamera verwendet, die 30 Bilder pro Sekunde aufnimmt und werden jeweils 500000 Objektansichten untersucht, ist mit $F \leq 500000 \cdot 0.25^{20}$ Falschpositiven pro Bild zu rechnen, was einem Erwartungswert von $60 \cdot 60 \cdot 30 \cdot 500000 \cdot F \approx 0,049$ Falschpositiven pro Stunde entspricht. Allerdings beziehen sich diese Abschätzungen auf die Trainingsmenge und können nur auf solche Verkehrssituationen verallgemeinert werden, die von der Trainingsmenge repräsentiert werden. Bei der Wahl der Parameter f und p ist des Weiteren zu beachten, dass sie sich stark auf die Laufzeit des Trainingsvorgangs und der späteren Detektion auswirken. Werden f zu niedrig und p zu hoch gewählt, ist unter Umständen keine Echtzeitverarbeitung möglich.

3.2.7. Laufzeitoptimierungen

Gemäß der in Abschnitt 1.2 definierten Anforderungen müssen die Segmentierungs- und Detektionsverfahren online lauffähig sein. Die oben vorgestellten Algorithmen wurden bereits unter diesem Gesichtspunkt ausgewählt und sind (eine effiziente Implementierung vorausgesetzt) prinzipiell online lauffähig. Ob die Ausführung in Echtzeit erfolgt, hängt nach der Implementierung von den zur Verfü-

gung stehenden Hardware-Ressourcen einerseits und der Menge der zu verarbeitenden Informationen andererseits ab. Falls es möglich ist, die zur Verfügung stehenden Hardware-Ressourcen zu beeinflussen, können sie für den Einsatzzweck optimiert werden [Mü+10].

Darüber hinaus ist eine Optimierung der Laufzeit möglich, wenn der zeitliche und räumliche Zusammenhang der zu verarbeitenden Informationen berücksichtigt wird. Bei einer Videokamera mit Frequenz f Hz und Auflösung $w \times h$ Bildpunkte müssen pro Sekunde $I := f \cdot w \cdot h$ Bildpunkte verarbeitet werden. Durch verschiedene im Folgenden vorgestellte Ansätze ist es möglich, die Menge der zu verarbeitenden Bildpunkte zu reduzieren und dadurch die Verarbeitung zu beschleunigen.

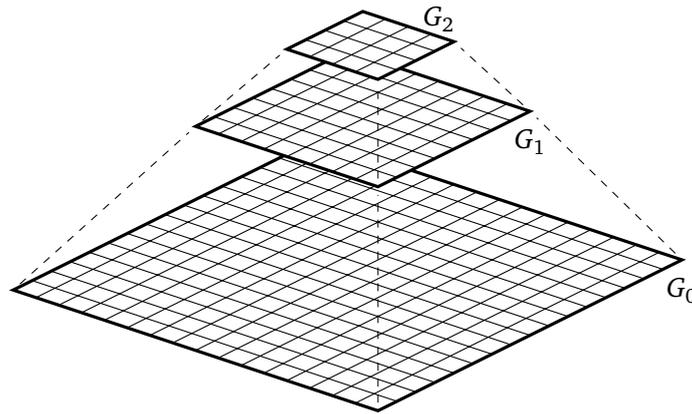
Zeitliche Filterung Einer der einfachsten Ansätze zur Reduktion der Laufzeit besteht in der Reduktion der Frequenz f : Anstelle der Frequenz f des Sensors wird für die Verarbeitung die Frequenz $f' < f$ verwendet, einige Videobilder werden also schlicht nicht verarbeitet. Dieses Vorgehen hat den offensichtlichen Nachteil, dass einige Objekte nicht oder verspätet erkannt werden.

Einschränkung von Suchregionen Abhängig von der Anbringung der Kamera kommt es vor, dass Teile des Fahrzeugs – typischerweise Teile der Motorhaube oder Teile der Kamerahalterung – im Videobild zu erkennen sind. Aufgrund ihrer Nähe zur Kamera sind keine zu erkennenden Objekte in diesen Regionen zu erwarten. Die entsprechenden Bildregionen können daher aus I entfernt werden.

Die Position und Ausrichtung der Videokamera sowie ihre Optik wird während der Fahrt nicht verändert. Das Fahrzeug selbst ändert seine Position und die Rotation um die z-Achse durch Lenken signifikant, während Änderungen des Nick- und Rollwinkels nur in eingeschränkten Bereichen vorkommen. Die Anbringung und Größe von Verkehrselementen folgt Vorschriften der VwV-StVO [Sta]. Nach der Projektion dieser Bereiche unter Beachtung der Bewegungsräume des Fahrzeugs ins Bild bleiben Bildregionen übrig, in denen bestimmte Verkehrselemente nicht zu erwarten sind. Diese Regionen können entweder komplett aus I entfernt werden oder an das Segmentierungs- bzw. Detektionsverfahren weitergegeben werden, welches seinen Suchraum einschränken kann.

Pyramidenansätze Die Laufzeit der modellbasierten Ansätze und der morphologischen Operatoren hängt von der Größe der gesuchten Objekte ab. Beispielsweise werden bei der radialen Symmetrie für Kreise für jeden betrachteten Radius ein bzw. zwei zusätzliche Akkumulatoren benötigt; beim Top-Hat Operator hängt die (laufzeitrelevante) Größe des strukturierenden Elements von der maximalen Größe der zu segmentierenden Ampellichter ab. An dieser Stelle eignen sich Gauß-Pyramiden [BPF12], um die Größe der Elemente zu begrenzen; anstatt auf dem Originalbild zu arbeiten, wird die Segmentierung auf skalierten Bildern durchgeführt. Zum Originalbild G_0 wird das um den Faktor 2^{-s} skalierte Bild G_s (Level s) berechnet durch s -malige Anwendung der Operation ∇ :

$$\begin{aligned} \nabla : \mathfrak{G}^{w,h} &\rightarrow \mathfrak{G}^{w/2,h/2} \\ \nabla(g(\mathbf{x})) &:= g'(2\mathbf{x}) \end{aligned} \quad [3.21]$$

Abb. 3.14.: Bildpyramiden G_1 und G_2 zum Originalbild G_0 .

mit

$$G'_i := G_i * \frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}. \quad [3.22]$$

Das Bild G_{i+1} enthält nur $1/4$ der Anzahl Bildpunkte von G_i (vergleiche Abbildung 3.14) und dementsprechend beschleunigt sich die Segmentierung bzw. Detektion. Die Wahl der Pyramidenlevel, auf denen gearbeitet wird, verlangt eine Abwägung zwischen Laufzeit und Erkennungsqualität: Aufgrund des Informationsverlusts durch die Skalierung ist auf höheren Leveln mit schlechteren Ergebnissen zu rechnen.

Kontextabhängige Laufzeitoptimierungen Die oben vorgestellten Optimierungsansätze können sehr gut durch Kontextinformationen dynamisch gesteuert werden. Sobald Verkehrselemente im Videobild gefunden und verfolgt werden, kann die Prädiktion genutzt werden, um ihre Position und Größe im nächsten Videobild zu schätzen. Auf diese Weise ergeben sich Suchregionen, die als starke Einschränkung der Menge zu segmentierender Bildpunkte genutzt werden können. Auch Segmentierungsparameter wie die möglichen Radien von Kreisen können hiermit eingeschränkt werden. Der Laufzeitanteil der Segmentierung wird in diesem Fall zugunsten von Tracking und Klassifikation stark gesenkt. Da diese Ansätze stark anwendungsabhängig sind, werden konkrete Umsetzungen kontextabhängiger Laufzeitoptimierungen in Abschnitt 3.5 und Kapitel 5 beschrieben.

3.3. Merkmalsextraktion und Klassifikation

Kandidaten statischer Verkehrselemente werden von Segmentierungs- und Detektionsverfahren in Videobildern lokalisiert. Um festzustellen, ob ein Kandidat (eine ROI) tatsächlich ein statisches Verkehrs-

element darstellt und wenn ja, um welchen Typ es sich handelt, wird eine automatische Klassifikation durch überwachte Lernverfahren eingesetzt.

Gegeben sei eine Menge nominaler Kategorien Ω , die sogenannten *Klassen*, mit

$$\Omega := \{\omega_0, \dots, \omega_{n-1}\} \quad [3.23]$$

und $n > 1$. Sie repräsentieren Typen statischer Verkehrselemente oder allgemeiner Kategorien von Objekten. Ferner existieren die echten Teilmengen

$$\begin{aligned} \Omega_p &\subset \Omega \\ \text{und } \Omega_n &\subset \Omega \end{aligned} \quad [3.24]$$

der positiven und negativen Klassen mit

$$\Omega_p \cap \Omega_n = \emptyset. \quad [3.25]$$

Im Allgemeinen gilt $\Omega_p \cup \Omega_n \neq \Omega$, es kann also Klassen geben, die weder positiv noch negativ sind. Dies wird später genutzt, um eine Klasse zu bilden, die Kandidaten enthält, die nicht eindeutig einer positiven oder negativen Klasse zuzuordnen sind.

Aufgabe der Klassifikation ist die Abbildung von ROIs $\Theta \in \Theta_{\cup}^{\mathbb{B}}$ auf eine Klasse $\omega \in \Omega$. Dazu wird zunächst eine *Merkmalstransformation* $f : \Theta_{\cup}^{\mathbb{B}} \rightarrow M$ durchgeführt, die Θ in den *Merkmalsraum* $M \in \mathbb{R}^q$ überführt. Gemäß einer Entscheidungsfunktion

$$\begin{aligned} d : M &\rightarrow \Omega \\ \hat{\omega} &:= d(f(\Theta)) \end{aligned} \quad [3.26]$$

wird die eigentliche Klassifikation durchgeführt.

Die Merkmalstransformation f wird so gewählt, dass Daten mit gleicher Klassenzugehörigkeit auf ähnliche Werte in M abgebildet werden. Gleichzeitig wird versucht, die Dimension q möglichst gering zu halten, um Laufzeit und Speicheraufwand zu minimieren.

Der Eingaberaum ergibt sich aus der Problemstellung bzw. der damit verbundenen Datenakquise. Im Rahmen dieser Arbeit definiert er sich über ROIs, die durch Segmentierung und Detektion der Eingabedaten bestimmt werden. Die Wahl von Merkmalsraum und -transformation muss problemspezifisch vorgenommen werden. Für die Klassifikation von statischen Verkehrselementen geeignete Merkmalstransformationen werden in den Abschnitten 3.3.1 bis 3.3.5 vorgestellt.

Um die Entscheidungsfunktion d zu parametrieren werden Daten aus einer Trainingsmenge mit bekannter Klassenzugehörigkeit eingesetzt um ein *Entscheidungsmodell* (kurz: Modell) zu trainieren, welches den Fehler auf einer von der Trainingsdatenmenge disjunkten Menge von Testdaten minimiert. Das trainierte Modell definiert d vollständig. Zur Klassifikation von Verkehrselementen sind SVMs ein geeignetes Verfahren; ihre Anwendung wird in den Abschnitten 3.3.6 und 3.3.7 präsentiert.

3.3.1. Grauwert-Transformation

Sei $\Theta^{\mathbf{B}}$ eine ROI. Eine Grauwert-Transformation f_g bildet Θ auf $\mathbf{x} \in \mathbb{R}^q$ wie folgt ab:

$$(f_g(\Theta))_i := g_{\Theta,(u,v)} \begin{pmatrix} i \bmod u \\ \lfloor i/u \rfloor \end{pmatrix} \quad [3.27]$$

mit $\mathbf{G} \in \mathcal{G}$ der Grauwert-Transformation von \mathbf{B} und $uv = q$. Es wird also eine Transformation vom RGB-Farbraum auf Grauwerte, eine Größennormalisierung und eine Transformation zum kleinsten umschließenden Rechteck vorgenommen. Letztere ist notwendig, da im Allgemeinen die Anzahl der Bildpunkte von Θ auch nach Größennormalisierung unbekannt ist. Durch die Ergänzung von Bildpunkten aus der Umgebung löst f_g dieses Problem, bringt dabei aber auch Merkmale ein, die nicht zum Objekt gehören (vergleiche Abbildung 3.3). Falls das nicht erwünscht ist, können nicht zu $\Theta^{\mathbf{B}}$ gehörige Bildpunkte in $\mathbf{G}_{\Theta,(u,v)}$ durch feste Werte ersetzt werden. Für modellbasierte Detektionsverfahren wie der radialen Symmetrie für Kreise, bei denen die Anzahl der Elemente der ROI nach Größennormalisierung bekannt ist, kann auf die Transformation auf das kleinste umschließende Rechteck verzichtet werden.

Die Grauwert-Transformation ist eine der einfachsten Transformationen. Verschiebung, Skalierung und Drehung des Bildausschnitts wirken sich stark auf das Transformationsergebnis aus. Das Θ zugrundeliegende Segmentierungs- bzw. Detektionsverfahren muss demgegenüber daher robust sein.

Für die Klassifikation von Verkehrselementen eignet sich die Grauwert-Transformation insbesondere dort, wo Details der Textur entscheidend für die Klassenzugehörigkeit sind. Das ist beispielsweise bei Tempolimits der Fall, bei Zusatzzeichen oder bei der Unterscheidung der verschiedenen Typen von Halt- und Parkverboten (die jeweils unterschiedliche Pfeile benutzen).

3.3.2. Blockweise Diskrete Kosinustransformation

Einer der Nachteile der Grauwert-Transformation ist die vergleichsweise hohe Dimensionalität des Merkmalsraums. Mit Hilfe einer diskreten Kosinustransformation (die auch z.B. Teil des JPEG-Kompressionsverfahrens ist) kann die Dimension des Merkmalsraums verringert werden. Dazu wird eine Transformation der Bilddaten in den Frequenzbereich vorgenommen und eine Tiefpassfilterung durchgeführt.

Sei $\Theta^{\mathbf{B}}$ eine ROI zum Bild $\mathbf{B} \in \mathfrak{R}^{h,w}$ und \mathbf{G} die Grauwerttransformation von \mathbf{B} . Dann ist $\mathbf{H} := \mathbf{G}_{\Theta,(n,n)}$ das auf die Größe $n \times n$ normalisierte kleinste umschließende Rechteck zur Grauwerttransformation von Θ mit $n := 2^r$ für $r \in \mathbb{N}$. Die Koeffizienten-Matrix \mathbf{D} der diskreten Kosinustransformation von \mathbf{H} ist

$$\mathbf{D} = \mathbf{C}^n \mathbf{H} (\mathbf{C}^n)^\top \quad [3.28]$$

mit

$$C_{ij}^n := \sqrt{\frac{\alpha_i}{n}} \cos\left(\frac{\pi(2j+1)i}{2n}\right) \quad [3.29]$$

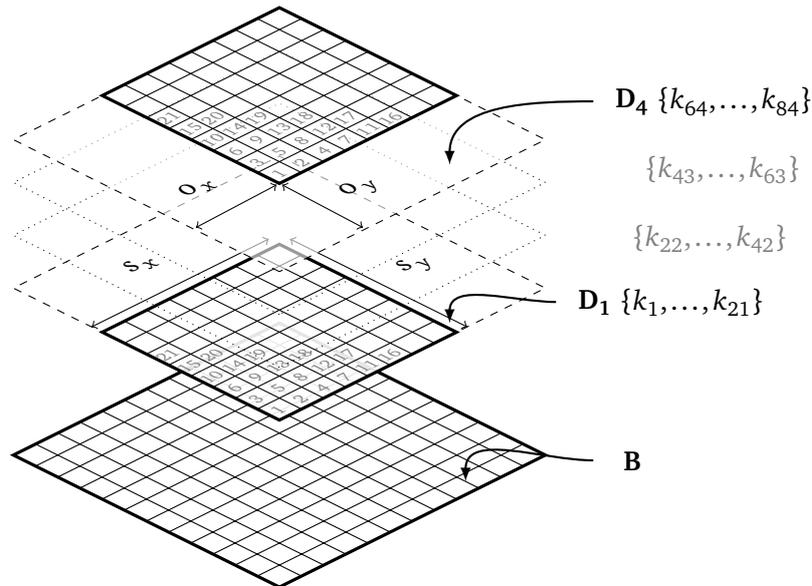


Abb. 3.15.: Schematische Darstellung der blockweisen diskreten Kosinustransformation für $\mathbf{B} \in \mathcal{G}^{12,12}$ mit $s_x = s_y = 8$ und $o_x = o_y = 4$. Die vier Blockmatrizen $D_1 \dots D_4$ decken unterschiedliche Bereiche von \mathbf{B} ab und definieren die Eingabebereiche für die Berechnung der DCT. Die jeweils ersten k Koeffizienten davon bilden zusammen den Merkmalsvektor \mathbf{k} .

und $\alpha_0 = 1$ sowie $\alpha_i = 2$ für $i > 0$.

Die Frequenzen in x und y -Richtung steigen mit den jeweiligen Indizes, so dass die tiefsten Frequenzen durch die Einträge in der oberen linken Ecke von \mathbf{D} repräsentiert werden. Zur Tiefpassfilterung werden daher die ersten (gemäß $i \cdot j$) k Koeffizienten von D extrahiert. Um die Strukturen in den Teilbereichen der ROI zu erhalten, wird die oben dargestellte diskrete Kosinustransformation nicht auf \mathbf{H} angewandt, sondern auf Blockmatrizen $\mathbf{H}_{i,j}$ fester Größe zu \mathbf{H} mit

$$\mathbf{H}_{i,j} = \mathbf{H}(i o_y : i o_y + s_y; j o_x : j o_x + s_x). \quad [3.30]$$

Dabei sind wie in Abbildung 3.15 illustriert o_y und o_x die Verschiebung in vertikaler bzw. horizontaler Richtung und s_y und s_x die vertikale bzw. horizontale Größe der Blockmatrizen.

3.3.3. Blockweise Histogramm-Transformation

Die Histogramm-Transformation verwendet analog zu Formel [3.30] überlappende Blockmatrizen $\mathbf{B}_{i,j}$. Für jede Blockmatrix (Bildausschnitt) wird pro Bild-Kanal ein Histogramm berechnet (vergleiche Abbildung 3.16). Jede Klasse (engl. *bin*) der einzelnen Histogramme entspricht im Merkmalsraum einer Dimension.

Neben der Nutzung rechteckiger Blöcke ist es auch möglich, die Histogramme auf anders gewählten Bildausschnitten zu berechnen, zum Beispiel Kreisen. Ersetzt man das zugrundeliegende Grauwert- bzw. Farbbild durch ein Kantenbild, erhält man eine Spezialform der nachfolgend beschriebenen HOG-

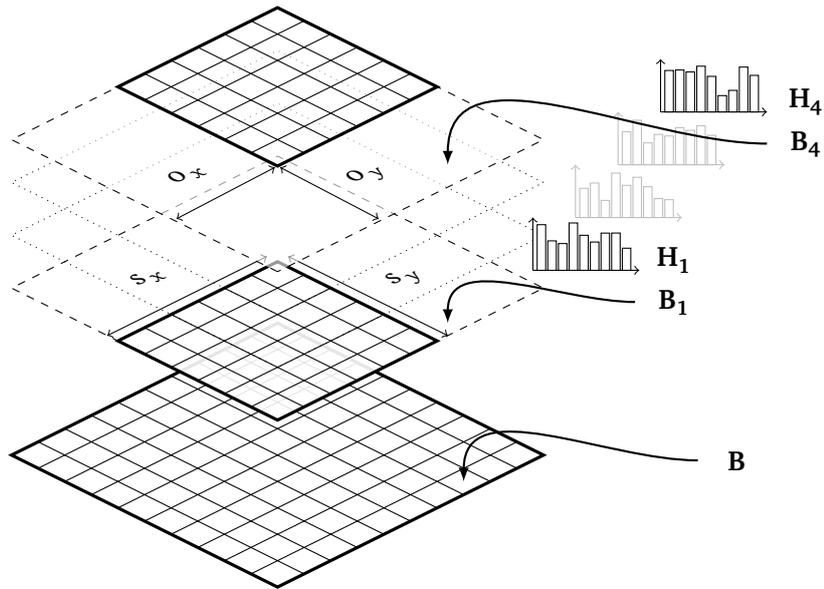


Abb. 3.16.: Schematische Darstellung der blockweisen Histogramm-Transformation für $B \in \mathbb{G}^{10,10}$ mit $s_x = s_y = 6$ und $o_x = o_y = 4$. Die vier Blockmatrizen $B_1 \dots B_4$ decken unterschiedliche Bereiche von B ab und definieren die Eingabebereiche für die Berechnung der vier Histogramme $H_1 \dots H_4$.

Transformation.

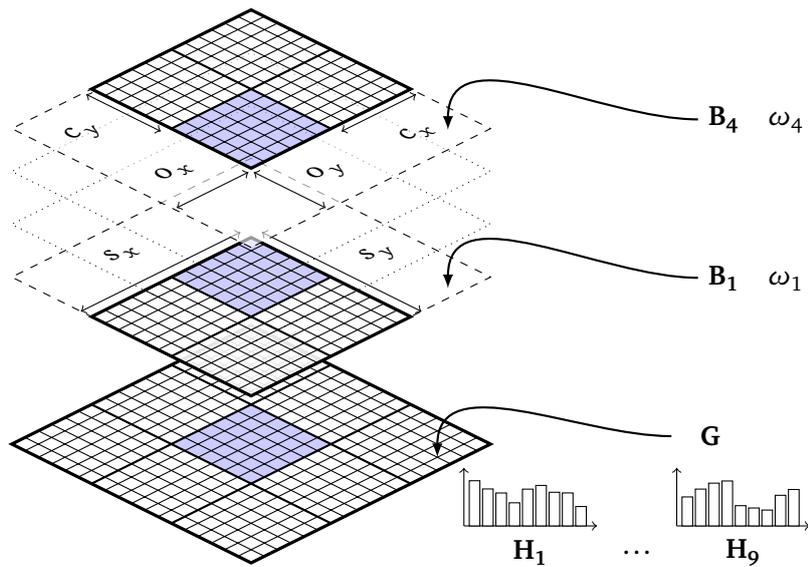


Abb. 3.17.: Schematische Darstellung der HOG-Transformation für $B \in \mathbb{G}^{18,18}$ mit Blockgröße $s_x = s_y = 12$, Blockverschiebung $o_x = o_y = 6$ und Zellengröße $c_x = c_y = 6$. Die neun Blockmatrizen $B_1 \dots B_9$ decken unterschiedliche Bereiche von B ab und definieren die Eingabebereiche für die Berechnung der neun Histogramme $H_1 \dots H_9$.

3.3.4. Histogram of Oriented Gradients (HOG)

Das von Dalal und Triggs [DT05] für die Erkennung von Personen vorgeschlagene HOG-Merkmal eignet sich auch als Merkmalstransformation für Verkehrselemente. Es ähnelt der in Abschnitt 3.3.3 vorgestellten blockweisen Histogramm-Transformation, arbeitet allerdings nicht direkt auf Bildpunkten, sondern auf Gradienten, die über Gleichung [3.8] berechnet werden können. Die Blöcke werden weiter unterteilt in Zellen der Größe $c_x \times c_y$, auf denen Histogramme berechnet werden. Pro Block wird aus diesen ein Histogramm zusammengesetzt, indem die Histogramme der zugehörigen Zellen über den Block normalisiert und aneinander gehängt werden. Abbildung 3.17 veranschaulicht das Vorgehen.

3.3.5. Hybride Merkmalstransformationen

Die in den vorhergehenden Abschnitten beschriebenen Merkmalstransformationen lassen sich beliebig miteinander und untereinander zu neuen, sogenannten *hybriden* Merkmalstransformationen kombinieren. Zwei Merkmalsvektoren $\mathbf{x} \in \mathbb{R}^n$ und $\mathbf{y} \in \mathbb{R}^m$ werden dazu zu einem Merkmalsvektor $\mathbf{z} \in \mathbb{R}^{n+m}$ kombiniert:

$$\mathbf{z} := \begin{pmatrix} x_1 & \dots & x_n & y_1 & \dots & y_m \end{pmatrix}^\top. \quad [3.31]$$

Analog werden hybride Merkmalstransformationen definiert, die mehr als zwei Merkmalstransformationen kombinieren. Aufgrund des bereits erwähnten Ziels der Dimensionsreduktion des Merkmalsraums ist beim praktischen Einsatz hybrider Merkmalstransformationen sorgfältig zwischen höherer Ausdrucksstärke und größerer Dimension – sowie der damit einhergehenden Notwendigkeit einer größeren Trainingsdatenmenge – abzuwägen. Ein sinnvoller Einsatz eines hybriden Merkmalsraums ist vor allem dann gegeben, wenn konträre Merkmale kombiniert werden. Beispielsweise kann zur Erkennung von Ampeln (Ampelkasten inklusive Ampellichter) ein hybrider Merkmalsraum verwendet werden, der eine HOG-Transformation mit einer blockweisen Histogramm-Transformation angewendet auf das in den HSV-Farbraum transformierte Bild kombiniert: Erstere bildet im Merkmalsvektor die Form des Ampelkastens und der Ampellichter ab, letztere die Farbe des aktiven Ampellichts bzw. der aktiven Ampellichter (vergleiche Abschnitt 3.5.5).

3.3.6. Hierarchische Support-Vektor-Maschinen (SVMs)

Eine Support-Vektor-Maschine ist eines der nach heutigem Stand der Forschung leistungsfähigsten Klassifikationsverfahren. Es teilt Objekte (repräsentiert durch Vektoren in einem Vektorraum) einer von zwei Klassen zu. Die Entscheidung, welcher Klasse ein Objekt zugeordnet wird, wird anhand seiner Lage im Vektorraum in Bezug auf eine Hyperebene (*Entscheidungsmodell*) getroffen. Anhand von ausgewählten Vektoren, deren Klassenzugehörigkeit vorgegeben ist (*Trainingsbeispiele*), wird eine Teilmenge von ihnen als sogenannte Stützvektoren (englisch *support vectors*) in einer einmalig durchzuführenden Trainingsphase bestimmt. Sie bestimmen die trennende Hyperebene.

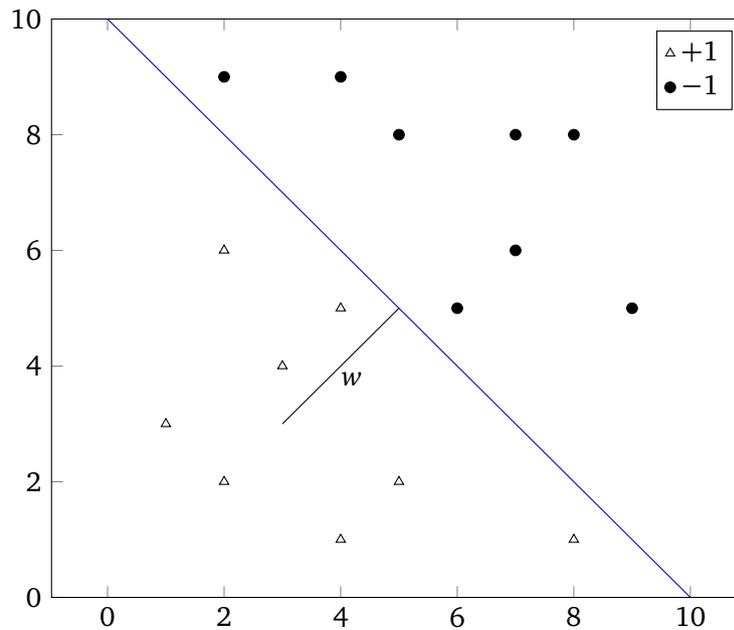


Abb. 3.18.: Eine SVM bestimmt in der Trainingsphase eine trennende Hyperebene zwischen zwei Klassen. Die Hyperebene wird zur Klassifikation von Objekten genutzt, deren Label nicht bekannt ist.

Seien $\mathbf{x}_i \in \mathbb{R}^d$ Vektoren im Merkmalsraum mit $i = \{1, \dots, \ell\}$ und Label $y_i \in \{-1, +1\}$, d.h. $\Omega = \{-1, +1\}$. Falls eine Hyperebene existiert, welche die Punkte mit Label -1 von denen mit Label $+1$ trennt, gilt für Punkte auf dieser Hyperebene

$$\mathbf{x}^\top \mathbf{w} + b = 0 \tag{3.32}$$

wobei \mathbf{w} eine Normale der Hyperebene bezeichnet. Über die Lage eines Punktes bezüglich der Hyperebene kann damit sein Label bestimmt werden: Die Entscheidungsfunktion $d(\mathbf{x})$ für einen Punkt \mathbf{x} ist

$$d(\mathbf{x}) = \text{sgn}(\mathbf{x}^\top \mathbf{w} + b). \tag{3.33}$$

Gilt außerdem für alle i

$$y_i(\mathbf{x}_i^\top \mathbf{w} + b) \geq 1, \tag{3.34}$$

dann existiert ein Rand der Breite $\frac{1}{\|\mathbf{w}\|}$ zwischen den beiden Klassen und der Hyperebene. Vektoren, die Gleichung 3.34 mit Gleichheit erfüllen, liegen auf der Hyperebene H_1 mit $\mathbf{x}_i^\top \mathbf{w} = 1$ bzw. H_2 mit $\mathbf{x}_i^\top \mathbf{w} = -1$. Sie werden Stützvektoren genannt. Der Abstand zwischen H_1 und H_2 ist $\frac{2}{\|\mathbf{w}\|}$ und seine Maximierung wird als Optimierungsproblem aufgefasst, mit dem eine optimale trennende Hyperebene gefunden wird:

$$\begin{aligned} &\text{Minimiere} \quad \frac{1}{2} \|\mathbf{w}\|^2 \\ &\text{unter der Nebenbedingung} \quad y_i(\mathbf{x}_i^\top \mathbf{w} + b) \geq 1 \quad \forall i. \end{aligned} \tag{3.35}$$

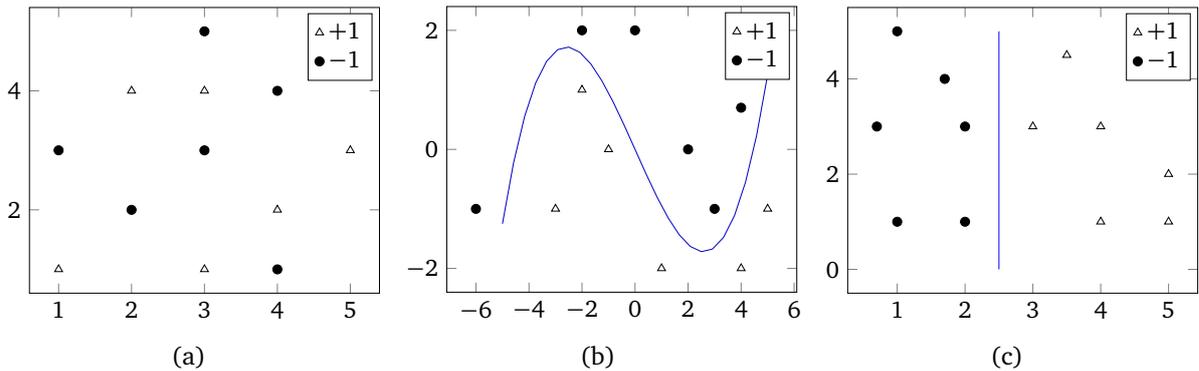


Abb. 3.19.: Als Teil der SVM-Klassifikation werden Objekte vom Eingaberaum I (a) in den Merkmalsraum F (b) transformiert. Des Weiteren wird eine implizite Transformation in den Kernelraum K (c) durchgeführt.

Separierbarkeit

Die lineare Separierbarkeit von Daten durch eine Hyperebene ist nicht in jedem Fall erreichbar. Das in Abbildung 3.19b dargestellte Zweiklassenproblem ist ein Beispiel dafür. Da dieser Fall in der Praxis häufig auftritt, wurden verschiedene Ansätze entwickelt, um dennoch eine Separierung der Daten zu ermöglichen: Die Einführung von Schlupfvariablen in den Nebenbedingungen des Optimierungsproblems und die implizite Transformation in einen höherdimensionalen Raum [Bur98]. Beide Verfahren können miteinander kombiniert werden.

Durch die Einführung von Schlupfvariablen werden gezielt Fehler bei der Klassifikation der Trainingsdaten zugelassen, diese Fehler jedoch minimiert. Dazu wird das eigentliche Optimierungsproblem – die Maximierung des Randbereichs der separierenden Hyperebene – relaxiert. Jede Nebenbedingung repräsentiert ein Trainingsdatum und die Summe der Verletzungen der Nebenbedingungen wird minimiert. Das Optimierungsproblem ist nun

$$\begin{aligned}
 &\text{Minimiere} && \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^{\ell} \xi_i \\
 &\text{unter den Nebenbedingungen} && y_i (\mathbf{x}_i^\top \mathbf{w} + b) \geq 1 - \xi_i \quad \forall i \\
 &&& \xi_i \geq 0 \quad \forall i
 \end{aligned} \tag{3.36}$$

wobei die Konstante c ein Parameter ist, der zur Abwägung zwischen der Größe des Trennbereichs und der korrekten Trennung der Trainingsdaten benutzt werden kann.

Dieses Optimierungsproblem ist konvex und wird üblicherweise in seiner dualen Form durch Einführung von Lagrange-Multiplikatoren α gelöst. Dabei wird \mathbf{w} als Linearkombination der Punkte \mathbf{x}_i dargestellt: $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i$.

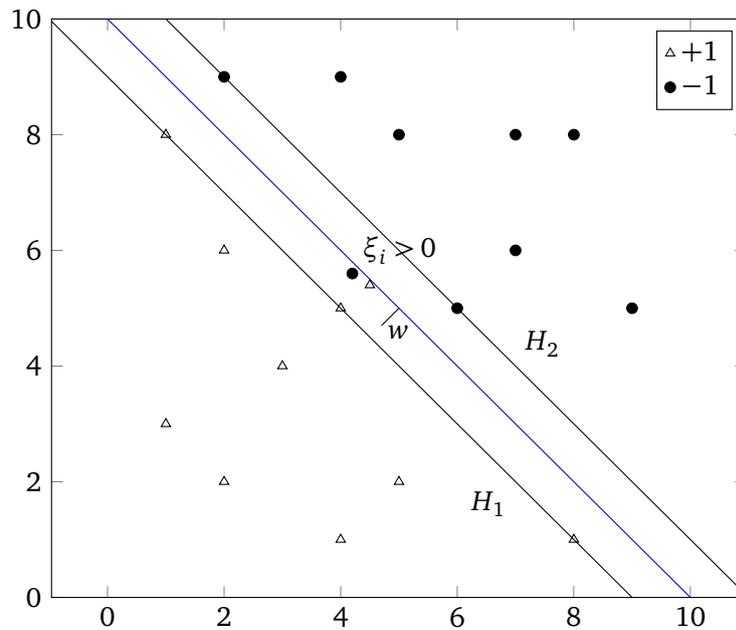


Abb. 3.20.: Durch die Einführung von Schlupfvariablen im Optimierungsproblem werden Fehler zugelassen.

$$\begin{aligned}
 &\text{Maximiere} \quad \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\
 &\text{unter den Nebenbedingungen} \quad 0 \leq \alpha_i \leq c \quad \forall i \quad [3.37] \\
 &\text{und} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0.
 \end{aligned}$$

Die Entscheidungsfunktion dazu ist

$$d(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^{\ell} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b \right). \quad [3.38]$$

Diejenigen Daten \mathbf{x}_i , für deren Lagrange-Multiplikatoren $\alpha_i = 0$ gilt, sind für die Entscheidungsfunktion irrelevant. Dies ist praktisch relevant, denn das Modell benötigt nur die sogenannten Stützvektoren \mathbf{x}_i mit $\alpha_i > 0$, um d zu berechnen. Im praktischen Einsatz werden auf diese Weise Speicherplatz und Laufzeit reduziert.

Schließlich wird durch eine Transformation in einen anderen Raum die Möglichkeit geschaffen, Hyper-ebenen höherer Dimension (aus Sicht des Merkmalsraums) zu verwenden und damit eine nichtlineare Trennung der Daten zu ermöglichen. Dazu wird $\mathbf{x} \in \mathbb{R}^q$ in den Kernelraum \mathbb{R}^o transformiert, wofür die Kernel-Funktion ϕ (kurz: der Kernel ϕ) benutzt wird mit $\phi : \mathbb{R}^q \rightarrow \mathbb{R}^o$. Diese Transformation muss nicht explizit durchgeführt werden, sondern es muss lediglich das Skalarprodukt $k(\mathbf{x}, \mathbf{y}) := \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$ in \mathbb{R}^o berechenbar sein. Dadurch ergibt sich eine sehr effiziente Berechnung, weswegen diese Transformation

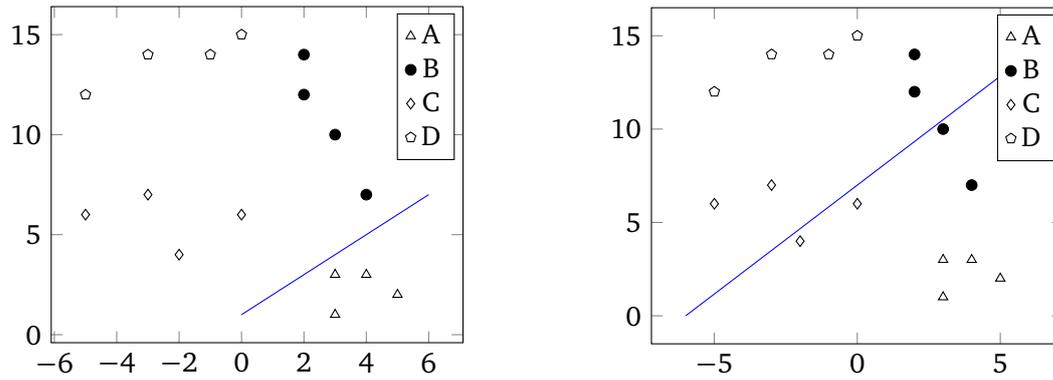


Abb. 3.21.: Links: SVM zur Klassifikation von A-{B,C,D} als Teil des *one-against-all* Verfahrens. Rechts: SVM zur Klassifikation von A-D als Teil des *one-against-one* Verfahrens.

auch unter dem Namen *Kernel-Trick* bekannt ist. Die neue Entscheidungsfunktion ist

$$d(\mathbf{x}) = \operatorname{sgn} \left(\sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \right). \quad [3.39]$$

Verbreitet genutzte Kernel sind der polynomielle Kernel k_{poly} vom Grad d (üblicherweise 1, 2 oder 3) mit

$$k_{\text{poly}}(\mathbf{x}, \mathbf{y}) := (\delta \langle \mathbf{x}, \mathbf{y} \rangle + \epsilon)^d \quad [3.40]$$

und skalaren Parametern δ und ϵ sowie der RBF-Kernel k_{rbf} mit

$$k_{\text{rbf}}(\mathbf{x}, \mathbf{y}) := \exp \left(-\gamma \|\mathbf{x} - \mathbf{y}\|^2 \right) \quad [3.41]$$

und skalarem Parameter γ . In den RBF-Kernel geht der Abstand zwischen Datenpunkten ein, der mit dem Parameter γ skaliert wird. Für hohe Werte von γ erhält man einen schmalen Kernel, so dass im Extremfall während des Lernens alle Trainingsdaten zu Stützvektoren werden. Die Entscheidungsfunktion [3.39] klassifiziert dann nur Daten in unmittelbarer Nähe zu Trainingsdaten richtig und entscheidet sich ansonsten immer für eine der beiden Klassen. Umgekehrt führt ein niedriger Wert von γ im Extremfall dazu, dass alle Stützvektoren Einfluss auf die Entscheidungsfunktion nehmen. Um gute Klassifikationsergebnisse zu erhalten, wird γ daher innerhalb des Bereichs $(\min_{\mathbf{x} \neq \mathbf{y}} \|\mathbf{x} - \mathbf{y}\|^{-2}, \max_{\mathbf{x} \neq \mathbf{y}} \|\mathbf{x} - \mathbf{y}\|^{-2})$ gewählt.

Mehrklassen SVM

Viele Klassifikationsprobleme bestehen aus $n > 2$ Klassen. Ein verbreiteter Ansatz, diese Problemstellung mit SVMs zu lösen besteht in der Reduktion des Mehrklassenproblems auf mehrere Zweiklassenprobleme. Der *one-against-all*-Ansatz führt für n zu unterscheidende Klassen n Einzelklassifikationen durch. Dabei wird jeweils ein Zweiklassenproblem bestehend aus der i -ten Klasse und der Vereinigung der übrigen Klassen betrachtet. Der *one-against-one*-Ansatz führt für alle n Klassen eine paarweise Klassifikation durch. Diejenige Klasse, welche die meisten Einzelklassifikationen gewinnt, wird als endgültiges Klassifi-

kationsergebnis ausgewählt. Die Anzahl durchzuführender Einzelklassifikation ist bei diesem Ansatz $(n-1) + (n-2) + \dots + 1 = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$.

Um beispielsweise eine Klassifikation der vier Klassen A, B, C und D durchzuführen, reduziert der *one-against-all* Ansatz die Klassifikation auf vier Zweiklassenprobleme A-{B,C,D}, B-{A,C,D}, C-{A,B,D} und D-{A,B,C}. Demgegenüber betrachtet der *one-against-one* Ansatz sechs Zweiklassenprobleme A-B, A-C, A-D, B-C, B-D und C-D (Abbildung 3.21).

Nicht nur die Ermittlung einer Klasse ist für praktische Anwendungen relevant, sondern auch eine Abschätzung, ob das Ergebnis korrekt ist. Dafür wird ein Konfidenzmaß $\hat{P}(y = \omega | \mathbf{x})$ als Schätzung der Wahrscheinlichkeit $P(y = \omega | \mathbf{x})$ bestimmt. Sei

$$P_{ij} := P(y = \omega_i | y = \omega_i \vee y = \omega_j, \mathbf{x}) \quad [3.42]$$

die Wahrscheinlichkeit, dass \mathbf{x} der Klasse ω_i angehört, wenn \mathbf{x} entweder ω_i oder ω_j angehört. Ein üblicher Ansatz, \hat{P}_{ij} festzulegen besteht in der Betrachtung des Abstands von \mathbf{x} zur trennenden Hyperebene. Ein einfaches Konfidenzmaß erhält man daraus durch Mehrheitsabstimmung gemäß

$$\hat{P}(y = \omega_i | \mathbf{x}) = \frac{2}{n(n-1)} \sum_{j, j \neq i} \delta(\hat{P}_{ij} > \hat{P}_{ji}) \quad \forall i. \quad [3.43]$$

Im Rahmen dieser Arbeit wird stattdessen das von Wu, Lin und Weng [WLW04; CL11] vorgeschlagene Konfidenzmaß benutzt, mit dem auf synthetischen und realen Daten bessere Klassifikationsergebnisse erzielt werden. Zur Festlegung von \hat{P}_{ij} benutzt es eine Sigmoid-Funktion gemäß

$$\hat{P}_{ij} := \frac{1}{1 + e^{cg(\mathbf{x})+d}} \quad [3.44]$$

mit Skalaren c, d , die anhand der Trainingsdaten aufgestellt werden und $g(x) := \sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$ (vergleiche Formel [3.39]). Durch die Lösung des Optimierungsproblems

$$\begin{aligned} & \text{Minimiere } \sum_{i=1}^{\ell} \sum_{j, j \neq i} (\hat{P}_{ji} \hat{P}_i - \hat{P}_{ij} \hat{P}_j)^2 \\ & \text{unter den Nebenbedingungen } \hat{P}_i \geq 1 \quad \forall i \quad [3.45] \\ & \sum_{i=1}^{\ell} \hat{P}_i = 1. \end{aligned}$$

wird daraus $\hat{P}_i := \hat{P}(y = \omega_i | \mathbf{x})$ berechnet, was angewendet auf alle Klassen ω_i dank der Nebenbedingungen in [3.45] direkt das Konfidenzmaß $\hat{P}(y = \omega | \mathbf{x})$ festlegt.

Hierarchische SVM zur Klassifikation von Verkehrselementen

Ausgehend von einer ROI Θ als potentielles Verkehrselement bestimmt die Klassifikation eine Klasse $\omega \in \Omega$. Durch Anwendung einer Merkmalstransformation erhält man aus der ROI einen Datenpunkt $f(\Theta) =: \mathbf{x}$, der als Eingabe für eine SVM geeignet ist:

$$\omega = d(\mathbf{x}). \quad [3.46]$$

Um eine optimale Qualität und Laufzeit zu erhalten, muss die Merkmalstransformation wie in Abschnitt 3.5 beschrieben passend zu den zu erkennenden Verkehrselementen gewählt und gegebenenfalls mit einer Merkmalsselektion kombiniert werden.

Bei der Klassifikation einer größeren Anzahl von Typen von Verkehrselementen mittels des *one-against-one* Verfahrens müssen für n Klassen $n(n-1)/2$ binäre Klassifikationsprobleme gelöst werden. Aus lauffzeittechnischen Gründen ist dies problematisch. Vor einem ähnlichen Problem steht das *one-against-all* Verfahren, das zwar weniger, aber komplexere binäre Klassifikationsprobleme lösen muss. Eine leistungsfähige Alternative bietet der Einsatz einer *hierarchischen SVM*: Es werden Gruppen gleichartiger Verkehrselemente (Klassen ähnlicher Merkmale) gebildet und in einer Baumstruktur angeordnet. Alle Blätter des Baumes sind Klassen $\omega \in \Omega$ und die Menge der Klassen Ω_k eines inneren Knotens k ist die Vereinigung der Klassen seiner Kinder. Der Wurzelknoten hat die Klassenmenge Ω (vergleiche Abbildung 3.22). Jedem inneren Knoten ist eine SVM zugeordnet, deren Klassifikationsproblem für jedes Kind eine Klasse aufweist. Die Merkmalstransformation wird spezifisch für die von Ω_k repräsentierten Verkehrselemente gewählt. Das Vorgehen entspricht einem Entscheidungsbaum, dessen innere Knoten SVMs darstellen. Das Klassifikationsergebnis entscheidet, welcher Entscheidungspfad im Baum eingeschlagen wird.

Wie die Einfärbung der SVMs bzw. Klassen in Abbildung 3.22 verdeutlicht, werden bei der Klassifikation mit einer hierarchischen SVM nicht alle Teil-SVMs durchlaufen. Aus diesem Grund wird die Konfidenz \hat{P} über der Potenzmenge 2^Ω definiert. Im Unterschied zum *one-against-one* und *one-against-all* Verfahren enthält \hat{P} nicht für alle atomaren Ereignisse (Klassen) Konfidenzen, sondern stattdessen aggregierte Konfidenzen für mehrere Klassen. Dies wird später in Abschnitt 4.2 in der D/S-Fusion ausgenutzt. Die Konfidenz $\hat{P}(\Psi|\mathbf{x})$ mit $\Psi \in 2^\Omega$ wird während der Traversierung des Baumes berechnet. Sei $\hat{P}^{[i]}(\Psi|\mathbf{x})$ die Konfidenz im i -ten Traversierungsschritt. Initial sei für alle $\Psi \in 2^\Omega$

$$\hat{P}^{[0]}(\Psi|\mathbf{x}) := \begin{cases} 1, & \Psi = \Omega \\ 0, & \text{sonst.} \end{cases} \quad [3.47]$$

Sei $\hat{P}_k(\omega_k|\mathbf{x})$ mit $\omega_k \in \Omega_k$ die Konfidenz der SVM zum aktuellen Knoten im k -ten Traversierungsschritt. Dann wird $\hat{P}^{[k]}$ für alle $\Psi \in 2^\Omega$ wie folgt bestimmt:

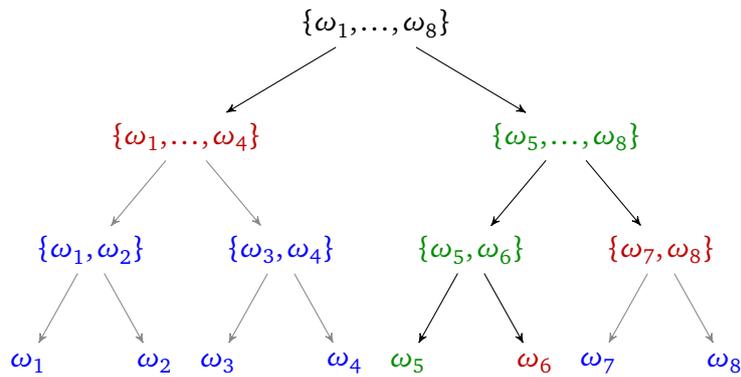


Abb. 3.22.: Traversierung der hierarchischen SVM für das Klassifikationsergebnis ω_5 .

$$\hat{P}^{[k]}(\Psi|\mathbf{x}) := \begin{cases} \hat{P}^{[k-1]}(\Omega_k|\mathbf{x})\hat{P}_k(\Psi|\mathbf{x}), & \Psi = \{\omega_k\}, \omega_k \in \Omega_k \\ 0, & \Psi = \Omega_k \\ \hat{P}^{[k-1]}(\Psi|\mathbf{x}), & \text{sonst} \end{cases} \quad [3.48]$$

Nach Durchlaufen des letzten Knotens z wird für alle $\Psi \in 2^\Omega$

$$\hat{P}(\Psi|\mathbf{x}) := \hat{P}^{[z]}(\Psi|\mathbf{x}) \quad [3.49]$$

gesetzt.

Die Nutzung einer hierarchischen SVM ist sinnvoll, wenn eine größere Anzahl von Klassen zu unterscheiden ist und sich die Merkmale einiger Klassen ähneln, so dass sie zusammengefasst werden können. Konkrete Anwendungen dafür werden in Abschnitt 3.5 beschrieben. Neben dem Umgang mit einer großen Anzahl an Klassen bergen sehr große Datensätze weitere Herausforderungen, für die im folgenden Abschnitt 3.3.7 zum aktiven Lernen Lösungsansätze präsentiert werden.

3.3.7. Aktives Lernen

Überwachte Lernverfahren benötigen annotierte Trainingsdaten, um eine Entscheidungsfunktion zu lernen. Trainingsdaten für statische Verkehrselemente können mit vertretbarem Aufwand erstellt werden, indem während der Fahrt Daten aufgezeichnet werden. Die nachfolgende Annotation der Daten durch Experten benötigt allerdings viel Zeit. Insbesondere für den Einsatz in Serienfahrzeugen ist eine sehr gute Leistung des Klassifikators essentiell, wofür umfangreiche Mengen annotierter Trainingsdaten benötigt werden. Um den Aufwand für die Annotation selbiger zu senken, ist es wünschenswert, gezielt und automatisch Trainingsdaten zur Annotation auszuwählen, welche zu einer überdurchschnittlichen Lernrate führen. Das Themenfeld *aktives Lernen* beschäftigt sich mit dieser Problemstellung: Anstelle der klassischen zufälligen Auswahl von Lerndaten (*passives Lernen*) werden gemäß einer Auswahlstrategie (englisch *query strategy*) Trainingsdaten zur Annotation ausgewählt [Set09]. Die so ausgewählten Daten werden wie gehabt von einem menschlichen Experten annotiert.

Bei der Anwendung von aktiven Lernverfahren auf die Klassifikation von statischen Verkehrselementen liegen die nicht annotierten Trainingsdaten als Menge A vor, auf die wahlfrei zugegriffen werden kann (*pool-based sampling*). Auswahlstrategien dürfen nicht auf binäre Klassifikationsprobleme beschränkt sein, sondern müssen eine beliebige Anzahl von Klassen unterstützen. Des Weiteren soll die Arbeitszeit des menschlichen Experten sinnvoll genutzt werden. Insbesondere sind kurze Annotationszeiten gefolgt von längeren Wartepausen zu vermeiden – dieses Szenario würde entstehen, wenn der aus theoretischer Sicht ideale Ansatz verwendet würde, immer genau ein neues Trainingsbeispiel auszuwählen und danach das Modell neu zu trainieren. Aktive Lernverfahren müssen im Rahmen dieser Arbeit daher in jeder Iteration eine Menge S von Trainingsbeispielen zur Annotation auswählen mit $|S| = h \gg 1$, bevor ein neues Modell trainiert wird. Dieses Vorgehen wird Batch-Modus (englisch *batch-mode*) genannt.

Eine der am häufigsten eingesetzten Auswahlstrategien ist *uncertainty sampling (US)* [Set09]. Sie kann für alle Lernverfahren eingesetzt werden, welche für eine ROI Θ die Berechnung von Konfidenzen $\hat{P}(\omega|f(\Theta))$ unterstützen. Sei

$$c(\Theta) := \max_{\omega \in \Omega} \hat{P}(\omega|f(\Theta)) \quad [3.50]$$

die maximale Konfidenz aller Klassen zur ROI Θ . Die Auswahlmenge $S_{US}(h)$ wird von US als jene h Trainingsbeispiele mit geringster Konfidenz gemäß

$$S_{US}(h) := \begin{cases} \operatorname{argmin}_{\Theta \in A} c(\Theta) & h = 1 \\ S_{US}(h-1) \cup \operatorname{argmin}_{\Theta \in A \setminus S_{US}(h-1)} c(\Theta) & h > 1 \end{cases} \quad [3.51]$$

berechnet. Zwar erfüllt US gemäß [3.51] als Auswahlstrategie die geforderten Kriterien (Mehrklassenprobleme und Batch-Modus), bringt jedoch auch einen Nachteil mit sich: Redundanz in den Trainingsdaten wird nicht berücksichtigt. Redundante Daten weisen ähnliche Konfidenzen auf; sind diese gering, werden sie bei entsprechender Batch-Größe h in S_{US} einbezogen. Dies führt dazu, dass die Lernrate geringer als möglich ausfällt. Ein möglicher Ansatz, Redundanz in den Daten zu berücksichtigen, wurde für SVMs und binäre Klassifikationsprobleme von Brinker [Bri03] vorgestellt. Vor dem Hintergrund des Versionsraums (der Raum der konsistenten Hypothesen) wird dazu eine gewichtete Kombination aus dem Abstand von Datenpunkten zur trennenden Hyperebene und dem Winkel zwischen den zu Datenpunkten korrespondierenden Hyperebenen betrachtet. Erstere ist eine lauffzeittechnisch günstige Heuristik zur Minimierung des Versionsraums, zweite ein Redundanzmaß – redundante Daten weisen ähnliche Winkel auf, während solche mit unterschiedlichen Winkeln zu einer stärkeren Reduktion des Versionsraums führen.

Durch die Einschränkung auf binäre Klassifikationsprobleme ist Brinkers Ansatz nicht direkt im Rahmen dieser Arbeit nutzbar. Demir, Persello und Bruzzone [DPB11] stellen eine Erweiterung des Ansatzes für Mehrklassenprobleme vor, der die Winkelbetrachtung als Redundanzmaß übernimmt, allerdings die lineare Kombination von Relevanz und Redundanz aufgibt. Um diese Eigenschaft zu erhalten, wird an dieser Stelle eine alternative Erweiterung von Brinkers Ansatz auf Mehrklassenprobleme mit dem Namen *angular diversity ranking (ADR)* vorgestellt. Diese Auswahlstrategie berechnet die Auswahlmenge

S_{ADR} mit

$$S_{\text{ADR}}(h) := \begin{cases} \arg \max_{\Theta \in A} (1 - c(\Theta)) & h = 1 \\ \arg \max_{\Theta \in A \setminus S_{\text{ADR}}(h-1)} (\lambda(1 - c(\Theta)) + (1 - \lambda)a(\Theta, S_{\text{ADR}}(h-1))) & h > 1 \end{cases} \quad [3.52]$$

mit

$$a(\Theta_1, S) := \min_{\Theta_2 \in S} \left(\cos^{-1} \left(\frac{|k(f(\Theta_1), f(\Theta_2))|}{\sqrt{k(f(\Theta_1), f(\Theta_1)) k(f(\Theta_2), f(\Theta_2))}} \right) \right). \quad [3.53]$$

wobei $k(\cdot, \cdot)$ die Kernelfunktion der SVM ist.

Aufgrund der in [3.52] ersichtlichen Abhängigkeit der Elemente voneinander ist eine iterative Berechnung von S_{ADR} notwendig. Algorithmus 3.1 stellt eine effiziente Berechnung dafür dar. Damit ist ein allgemein verwendbarer, effizient berechenbarer Algorithmus als Erweiterung von Brinkers Ansatz auf Mehrklassenprobleme gegeben. Im Folgenden werden Erweiterungen des Algorithmus für die Domäne der Erkennung von Verkehrselementen betrachtet.

Algorithmus 3.1 angular diversity ranking (ADR) Algorithmus für Mehrklassen-Klassifikationsprobleme und SVMs.

Eingabe: Trainingsdaten im Array i
Gewicht $\lambda \in [0..1]$

Ausgabe: Permutation des Arrays i

```

1: for  $j = 0$  to  $n$  do
2:    $p[j] \leftarrow 1 - c(\Theta_j)$ 
3:    $a[j] \leftarrow 0$ 

4: for  $k \leftarrow 0$  to  $h - 1$  do
5:    $i_{\max} \leftarrow k$ 
6:    $s_{\max} \leftarrow 0$ 
7:   for  $j \leftarrow k$  to  $h - 1$  do
8:      $s \leftarrow \lambda p[j] + (1 - \lambda) a[j]$ 
9:     if  $s > s_{\max}$  then
10:       $i_{\max} \leftarrow j$ 
11:       $s_{\max} \leftarrow s$ 

12:    $\text{swap}(i[k], i[i_{\max}])$ 
13:    $\text{swap}(p[k], p[i_{\max}])$ 
14:    $\text{swap}(a[k], a[i_{\max}])$ 

15:   for  $j \leftarrow k + 1$  to  $h - 1$  do
16:     if  $k = 0$  or  $a(\Theta_j, \Theta_k) < a[j]$  then
17:        $a[j] \leftarrow a(\Theta_j, \Theta_k)$ 

```

Für den zukünftigen Einsatz in Fahrzeugen ist es wichtig, keine Fehlalarme (Erkennung nicht vorhandener Verkehrselemente) zu produzieren. Auf den Erkennungsprozess übertragen entspricht dies einer Minimierung der Falschpositivrate. In einem iterativen Lernprozess kann man das unterstützen, indem

man gezielt solche Trainingsbeispiele Θ auswählt, welche die Bedingungen

$$y(\Theta) \in \Omega_n \quad \text{und} \quad [3.54]$$

$$\arg \max_{\omega \in \Omega} \hat{P}(\omega | f(\Theta)) \in \Omega_p \quad [3.55]$$

erfüllen, also negative Beispiele, die positiv klassifiziert werden. Im Rahmen des aktiven Lernens ist die tatsächliche Klassenzugehörigkeit $y(\Theta)$ zur Auswahlzeit nicht bekannt, so dass Beispiele gemäß Bedingung 3.54 nicht direkt bestimmt werden können. Indirekt kann ihr Anteil aber erhöht werden, indem bei der Auswahl positiv klassifizierte Beispiele bevorzugt werden: Ihr Anteil in der Auswahlmenge wird gezielt auf einen höheren Wert gesetzt als der tatsächliche Anteil positiver Beispiele in den Trainingsdaten. Dieser Anteil ist zwar ebenfalls nicht bekannt, kann jedoch auf Basis der ursprünglichen Trainingsmenge leicht abgeschätzt werden.

In den Algorithmus wird dieser Vorzug für positive (bzw. negative) Beispiele über einen Parameter $\kappa_p \geq 0$ eingebracht mit

$$\kappa_p = |\{\Theta \in S | \arg \max_{\omega \in \Omega} \hat{P}(\omega | f(\Theta)) \in \Omega_p\}| \quad [3.56]$$

$$h - \kappa_p = |\{\Theta \in S | \arg \max_{\omega \in \Omega} \hat{P}(\omega | f(\Theta)) \in \Omega_n\}|. \quad [3.57]$$

Eine der Hauptquellen für Redundanz in den Trainingsdaten ist die mehrfache Aufnahme des gleichen Verkehrszeichens während der Vorbeifahrt. Zusätzlich zur Redundanzbetrachtung mittels ADR kann dieses Wissen genutzt werden, indem die Anzahl der Trainingsbeispiele des gleichen Verkehrselements begrenzt wird. Dafür wird das Tracking von Verkehrszeichen eingesetzt und eine weitere Nebenbedingung

$$|T(\Theta)| \leq \kappa_t \quad \forall \Theta \in S \quad [3.58]$$

mit einem Parameter $\kappa_t > 0$ aufgestellt. Hierbei ist $T(\Theta)$ die Menge aller Beispiele, die zum gleichen Track gehören. Wird κ_t niedrig gewählt, werden weniger Trainingsbeispiele des gleichen Verkehrselements aufgenommen und dadurch die Redundanz verringert. Damit ergibt sich der um die Nebenbedingungen [3.56] – [3.58] erweiterte Algorithmus 3.2.

Ohne aktives Lernen besteht der Trainingsprozess aus den nacheinander folgenden Schritten der Datenaufnahme, Datenannotation und dem Training eines Modells. Um aktives Lernen zu integrieren, werden Datenannotation und Modelltraining iterativ durchgeführt und eine Auswahlstrategie für aktives Lernen einbezogen. Nach der Datenaufnahme ergibt sich damit folgender iterativ durchgeführter Ablauf:

- (I) Training eines Modells basierend auf T
- (II) Auswahlstrategie bestimmt Batch S aus A , $A := A \setminus S$
- (III) Annotation von S , $T := T \cup S$.

Die Anzahl durchzuführender Iterationen ist beliebig; anhand von Qualitätskriterien kann ein Abbruchkriterium festgelegt werden. Dafür wird wie in Abbildung 3.23 dargestellt eine Evaluation auf der von

Algorithmus 3.2 track-pruned class-biased angular diversity ranking (TPCB-ADR) Algorithmus für Mehrklassen-Klassifikationsprobleme und SVMs.

Eingabe: Trainingsdaten im Array i
Gewicht $\lambda \in [0..1]$

Ausgabe: Permutation des Arrays i

```

1: for  $j = 0$  to  $n$  do
2:    $p[j] \leftarrow 1 - c(\Theta_j)$ 
3:    $a[j] \leftarrow 0$ 

4:  $k \leftarrow 0$ 
5: for  $l \leftarrow 0$  to  $\text{length}(i) - 1$  and  $k < h$  do
6:    $i_{\max} \leftarrow k$ 
7:    $s_{\max} \leftarrow 0$ 
8:   for  $j \leftarrow l$  to  $h - 1$  do
9:      $s \leftarrow \lambda p[j] + (1 - \lambda) a[j]$ 
10:    if  $s > s_{\max}$  then
11:       $i_{\max} \leftarrow j$ 
12:       $s_{\max} \leftarrow s$ 

13:   if  $\kappa_t > T(i[i_{\max}], S)$  &  $\kappa_p > P(i[i_{\max}], S)$  &  $h - \kappa_p > N(i[i_{\max}], S)$  then
14:      $\text{swap}(i[k], i[i_{\max}])$ 
15:      $\text{swap}(p[k], p[i_{\max}])$ 
16:      $\text{swap}(a[k], a[i_{\max}])$ 

17:   for  $j \leftarrow l + 1$  to  $h - 1$  do
18:     if  $l = 0$  or  $a(\Theta_j, \Theta_l) < a[j]$  then
19:        $a[j] \leftarrow a(\Theta_j, \Theta_l)$ 
20:    $k \leftarrow k + 1$ 

```

den Trainingsdaten disjunkten Testdatenmenge durchgeführt. Als Qualitätskriterien eignen sich die Unterschreitung einer vorgegebenen Falschpositivrate und/oder die Überschreitung einer vorgegebenen Klassifikationsrate.

3.4. Tracking und temporale Fusion

Die sensorielle Erfassung des eigenen Zustands und des Umfelds erfolgt mit einer für die jeweiligen Sensoren eigenen Taktung. Neben den zu einem Zeitpunkt aktuellen Momentaufnahmen liefert auch die zeitliche Verknüpfung der Daten wertvolle Informationen. In diesem Kontext bezeichnet das Tracking die Verfolgung von Objekten über einen Zeitraum. Dabei wird eine Verknüpfung zwischen realen Objekten und ihren Abbildern in zeitlich aufeinander folgenden Daten hergestellt. Das Tracking kann genutzt werden, um zusätzliche Informationen zu einem Objekt zu erhalten (beispielsweise seine Geschwindigkeit). Es legt außerdem den Grundstein für die *temporale Fusion*, die die zeitliche Konsistenz von Beobachtungen prüft (Abschnitt 3.4.2) und Zustandswechsel behandelt (Abschnitt 3.4.3).

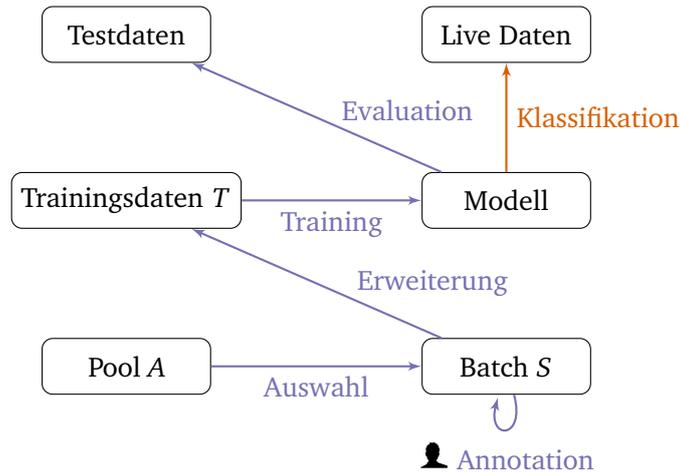


Abb. 3.23.: Trainingsprozess für aktives Lernen. Blaue Pfeile stellen Abläufe dar, die offline durchgeführt werden. Die orange dargestellte Klassifikation läuft online und ist vom eigentlich Trainingsprozess unabhängig.

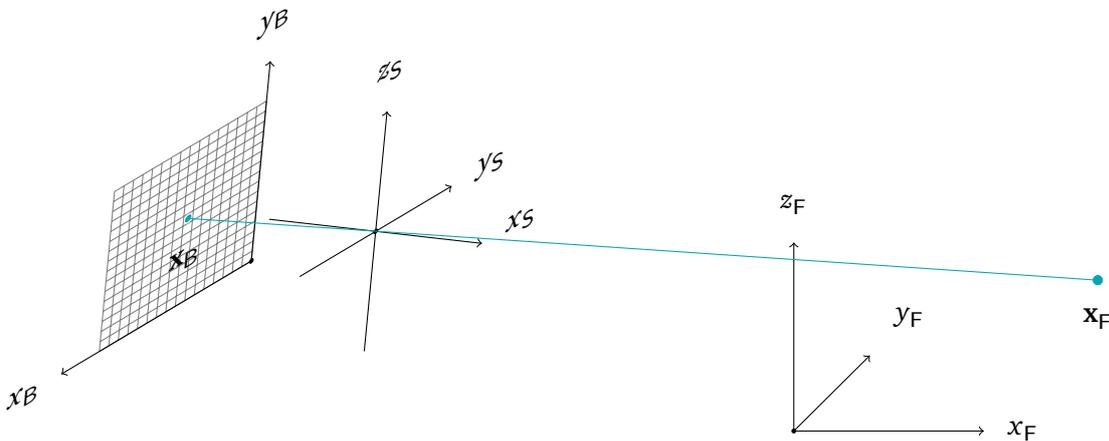


Abb. 3.24.: Fahrzeug-, Sensor- und Bildkoordinatensystem.

3.4.1. Tracking

Die im Rahmen dieser Arbeit behandelte Problemstellung gibt als Randbedingung sich bewegend (vom Fahrzeug mitbewegte) Sensoren vor. Demgegenüber sind die zu verfolgenden Objekte ortsfest. Mit einer Kamera als primärem Sensor liegt die Verfolgung von Objekten im Bildkoordinatensystem nahe. Dabei ist jedoch zu beachten, dass sich Position und Größe eines Objektbildes bei einer linearen Bewegung des Fahrzeugs aufgrund der Projektion ins Kamerabild nichtlinear verändern. Ein Standard-Schätzverfahren wie das (einfache) Kalman-Filter kann darauf nicht angewendet werden. Zweckmäßiger ist daher die Objektverfolgung im Fahrzeugkoordinatensystem. Um sie umzusetzen, müssen Transformationen zwischen Bild- und Fahrzeugkoordinatensystem bekannt sein.

Sei $\mathbf{x}_F \in \mathbb{R}^3$ ein Punkt im rechtshändigen Fahrzeugkoordinatensystem, bei dem die x -Achse in Fahrtrichtung, die y -Achse nach links und die z -Achse nach oben zeigt (vergleiche Abbildung 3.24). Sei $\mathbf{x}_B \in \mathbb{N}^2$ ein Punkt im Bildkoordinatensystem. Durch eine einmalig durchzuführende Kamerakalibrierung wer-

den extrinsische und intrinsische Kameraparameter bestimmt [BPF12]. Die extrinsischen Parameter beschreiben Lage und Orientierung der Kamera in Bezug auf das Fahrzeugkoordinatensystem über eine orthogonale Rotationsmatrix $\mathbf{R} \in \mathbb{R}^{3,3}$ und einen Translationsvektor $\mathbf{t} \in \mathbb{R}^3$. Damit kann der Punkt \mathbf{x}_F auf den Punkt $\mathbf{x}_S = \begin{pmatrix} x_S & y_S & z_S \end{pmatrix}^\top$ in das Kamerakoordinatensystem transformiert werden gemäß

$$\mathbf{x}_S = \mathbf{R}\mathbf{x}_F + \mathbf{t}. \quad [3.59]$$

Die intrinsischen Parameter beinhalten die Brennweite $\mathbf{f} = \begin{pmatrix} f_x \\ f_y \end{pmatrix}$ und den Bildhauptpunkt $\mathbf{p} = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$.

Damit ergibt sich der ins Bildkoordinatensystem transformierte Punkt \mathbf{x}_B mit

$$\mathbf{x}_B = \begin{pmatrix} p_x - \frac{y_S}{x_S} f_x \\ p_y - \frac{z_S}{x_S} f_y \end{pmatrix}. \quad [3.60]$$

Die umgekehrte Transformation von Bildkoordinaten in Fahrzeugkoordinaten benötigt zusätzliches Wissen über die Distanz von Objekten. Dieses kann in Form zusätzlicher Sensoren wie Stereo- oder ToF-Sensoren im Rahmen einer 2D/3D-Fusion hinzugezogen werden. Sind keine solchen Sensoren vorhanden, kann Wissen über die Größe des beobachteten Objekts eingebracht werden. Wie in Abschnitt 3.5 angegeben ist die Größe von Verkehrselementen festgelegt. Mit der so bekannten Breite eines Objekts b_F und der Breite seines Abbilds b_B als ROI Θ wird seine Distanz d gemäß Lochkameramodell und Strahlensatz als

$$d = \frac{b_F}{b_B} f_x \quad [3.61]$$

berechnet. Damit ergibt sich der Punkt \mathbf{x}_S im Kamerakoordinatensystem als

$$\mathbf{x}_S = \begin{pmatrix} d \\ \frac{(p_x - x)d}{f_x} \\ \frac{(p_y - y)d}{f_y} \end{pmatrix} \quad [3.62]$$

und darüber der Punkt \mathbf{x}_F im Fahrzeugkoordinatensystem als

$$\mathbf{x}_F = \mathbf{R}^\top(\mathbf{x}_S - \mathbf{t}). \quad [3.63]$$

Neben den Transformationen zwischen Fahrzeug- und Bildkoordinatensystem muss die Dynamik der Verkehrsszene berücksichtigt werden. Sei

$$\mathbf{x}_D := \left(p_x \ p_y \ p_z \ v_x \ v_y \ v_z \ a_x \ a_y \ a_z \right)^\top \quad [3.64]$$

der dynamische Zustand eines Objekts bestehend aus der Position, Geschwindigkeit und Beschleunigung entlang der einzelnen Achsen bezogen auf das Fahrzeugkoordinatensystem. Innerhalb eines Zeitraums Δt

verändert sich der Zustand $\mathbf{x}_D^{(t)}$ zu $\mathbf{x}_D^{(t+1)}$ gemäß

$$\mathbf{x}_D^{(t+1)} = \mathbf{F}^{(t)} \mathbf{x}_D^{(t)} \quad [3.65]$$

mit

$$\mathbf{F} := \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad [3.66]$$

Falls Fahrzeugsensoren zur Messung der Steuerung – Gas, Bremse, Lenkung – verfügbar sind, kann die sich daraus ergebende Dynamikänderung in den Zustandsübergang einberechnet werden. Das Modell erweitert sich zu

$$\mathbf{x}_D^{(t+1)} = \mathbf{F}^{(t)} \mathbf{x}_D^{(t)} + \mathbf{B}^{(t)} \mathbf{u}^{(t)}. \quad [3.67]$$

Dabei stellt \mathbf{u} die Steuereingangsgrößen dar und \mathbf{B} transformiert sie auf den Dynamikzustand.

Zu beachten ist, dass die Modellierung von \mathbf{F} gemäß [3.65] sowohl die Bewegung des Fahrzeugs als auch eine etwaige Bewegung des Objekts erfasst, während die Modellierung gemäß [3.67] die Bewegung des Fahrzeugs als bekannte Größe auffasst und nur die Bewegung des Objekts selbst in \mathbf{F} verfolgt. In beiden Fällen gilt jedoch, dass die Fahrzeugbewegung nicht vollständig bzw. nur ungenau bekannt ist. Diese Unsicherheit lässt sich als Prozessrauschen \mathbf{w} in das Modell einbringen. Die Position des Verkehrselements zum nächsten Zeitpunkt wird geschätzt als

$$\mathbf{x}_D^{(t+1)} = \mathbf{F}^{(t)} \mathbf{x}_D^{(t)} + \mathbf{B}^{(t)} \mathbf{u}^{(t)} + \mathbf{w}^{(t)}. \quad [3.68]$$

Die relative Position \mathbf{z}_B und die Größe von Verkehrselementen wird durch die Umfelderkennung in Bildkoordinaten gemessen. Gemäß Formel [3.63] wird \mathbf{z}_B im Fahrzeugkoordinatensystem auf den Punkt \mathbf{z}_F abgebildet. Auch hier ergeben sich aufgrund von Fehlern in der Detektion, der begrenzten Auflösung der Sensorik und Ungenauigkeiten in der Transformation Unsicherheiten, die als Messrauschen \mathbf{v} modelliert werden:

$$\mathbf{z}_F^{(t)} = \mathbf{H}^{(t)} \mathbf{x}_D^{(t)} + \mathbf{v}^{(t)} \quad [3.69]$$

mit

$$\mathbf{H} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad [3.70]$$

der Transformation des Dynamikzustands in den Messraum.

Mit der Zustandsgleichung [3.68] und der Messgleichung [3.69] sind die Voraussetzungen geschaffen, um ein Kalman-Filter [Kal60] zur Verfolgung eines Objekts einzusetzen. Es können jedoch mehrere Objekte gleichzeitig im Sichtfeld der Kamera sein. Um auch damit umgehen zu können, muss das Modell zu einem sogenannten *Multi-Target-Tracker* erweitert werden. Jedes Objekt wird dabei von einem eigenen Kalman-Filter verfolgt; ein Assoziationsschritt wird eingeführt, innerhalb dessen ROIs zu Kalman-Filtern zugeordnet werden. Anhand des prädizierten Zustands und der Kovarianzmatrix des Kalman-Filters wird ein sogenanntes *Gate* berechnet. Liegt der Zustand eines Objekts innerhalb dieses Gates, so ist eine Assoziation mit dem zugrundeliegenden Kalman-Filter möglich, andernfalls nicht. Intuitiv beschreibt das Gate die Menge der akzeptablen Nachfolgezustände eines Trackers. Für jede ROI ergibt sich darüber eine Menge von Kalman-Filtern, denen sie zugeordnet werden kann. Ist die Menge leer, wird ein neues Kalman-Filter zur Verfolgung des neuen Objekts erzeugt. Andernfalls wird anhand des Klassifikationsergebnisses der ROI und der Distanz zum prädizierten Zustand entschieden, welchem Kalman-Filter sie zugeordnet wird. Darüber hinaus entfernen sich Objekte durch die Fahrzeugbewegung aus dem Sichtfeld der Kamera. Daher werden Kalman-Filter entfernt, wenn ihnen über einen längeren Zeitraum keine ROIs zugeordnet werden.

3.4.2. Konfidenzberechnung für statische Elemente

Durch die Bewegung des Fahrzeugs ändern sich ständig die relativen Positionen von Objekten zur Kamera. Dies wird im Bild als Positions- und Größenänderung der Objektabbilder wahrgenommen und durch Segmentierung (Abschnitt 3.2) und Tracking (Abschnitt 3.4) behandelt. In Kombination mit der Klassifikation (Abschnitt 3.3) ergibt sich so pro Objekt eine Beobachtungsserie $O = (o^{(1)}, \dots, o^{(n)})$ bestehend aus Tupeln

$$o^{(t)} = (\Theta^{(t)}, \hat{P}^{(t)}(\omega)) \quad [3.71]$$

mit Informationen über die ROI $\Theta^{(t)}$ sowie Konfidenzmaßen $\hat{P}^{(t)}(\omega)$ zum Zeitpunkt t .

Das von der Beobachtungsserie O repräsentierte Objekt kann ein statisches Verkehrselement, ein anderes Objekt oder (bei Fehlern im Tracking) mehrere Objekte darstellen. Aufgabe der temporalen Fusion ist die Bestimmung des Typs und die Berechnung einer Konfidenz für die Korrektheit der ermittelten Informationen zum durch die Beobachtungsserie repräsentierten Objekt. Dies schließt die Erkennung von Fehlern im Tracking und den Umgang mit semi-statischen (zustandsändernden) Elementen wie beispielsweise Ampeln ein.

Sei

$$Q_k := \begin{cases} \operatorname{argmax}_{\omega \in \Omega} \hat{P}(\omega) & k = 1 \\ \operatorname{argmax}_{\omega \in \Omega \setminus \bigcup_{j=1}^{k-1} \{Q_j\}} \hat{P}(\omega) & k > 1 \end{cases} \quad [3.72]$$

die Klasse mit k-höchster Konfidenz und $Q_k^{(i)}$ die Klasse mit k-höchster Konfidenz des i -ten Elements der Beobachtungsserie O . Sei

$$\hat{P}_k := \hat{P}(Q_k) \quad [3.73]$$

die k -höchste Konfidenz. Sei

$$Q_0 := \begin{cases} \omega_0, & \hat{P}_1 < \kappa_m \\ \omega_0, & \hat{P}_1 - \hat{P}_2 < \kappa_d \\ Q_1, & \text{sonst} \end{cases} \quad [3.74]$$

die Klassifikation mit Ablehnung wenig stark ausgeprägter Maxima: Q_0 lehnt Klassen als ω_0 ab, wenn die maximale Konfidenz zu gering ist (definiert über den Schwellwert κ_m) oder sich nicht stark von der zweithöchsten Konfidenz abhebt (definiert über den Schwellwert κ_d).

Die *Mehrheitsabstimmung* c_m bestimmt diejenige Klasse, die am häufigsten gemäß Q_0 in der Beobachtungsserie klassifiziert wird:

$$c_m(O) := \operatorname{argmax}_{\omega \in \Omega} \sum_{i=1}^n \delta_{\omega, Q_0^{(i)}}. \quad [3.75]$$

Ist das Maximum mehrdeutig, wird entweder $c_m(O) = \omega_0$ entschieden – also sich für die Ablehnung als Hintergrundobjekt entschieden – oder sich zugunsten der Klasse innerhalb der mehrdeutigen Werte entschieden, die zuletzt positiv klassifiziert wurde.

Durch die Annäherung an Verkehrselemente bei der Vorbeifahrt erscheinen sie zum Ende einer Beobachtungsserie größer im Bild und werden zuverlässiger korrekt klassifiziert. Dieser Umstand wird von der *exponentiellen Glättung* c_e berücksichtigt. Sei

$$c_e^{(i)}(\omega) := \begin{cases} \hat{P}^{(i)}(\omega) & i = 1 \\ c_e^{(i-1)}(\omega) + \kappa_e(\hat{P}^{(i)}(\omega) - c_e^{(i-1)}(\omega)) & i > 1 \end{cases} \quad [3.76]$$

und sei $0 \leq \kappa_e \leq 1$ ein Parameter zur Abwägung zwischen stärkerer Glättung ($\kappa_e \rightarrow 0$) und Bevorzugung der Beobachtungen am Ende der Serie ($\kappa_e \rightarrow 1$). Aus $c_e^{(n)}$ berechnet die exponentielle Glättung die Klasse $c_e(O)$ als

$$c_e(O) := \operatorname{argmax}_{\omega \in \Omega} c_e^{(n)}(\omega). \quad [3.77]$$

Analog zu Gleichung [3.74] kann außerdem eine Ablehnung bei wenig stark ausgeprägtem Maximum durchgeführt werden.

Erweiterte Verfahren der temporalen Fusion können weiteres Wissen einbringen. Die temporale Fusion

kann auch als Klassifikationsproblem aufgefasst werden, bei dem die Entscheidung anhand gelernter Beobachtungssequenzen getroffen wird. Da die hier vorgestellten Verfahren bereits sehr gute praktische Ergebnisse erzielen, wird darauf an dieser Stelle verzichtet.

3.4.3. HMMs zur Zustandsschätzung semi-statischer Elemente

Semi-statische Verkehrselemente ändern über die Zeit ihren Zustand und damit ihr Aussehen. Seltene Zustandsänderungen, wie sie von Wechselverkehrszeichen vorgenommen werden, benötigen keine spezielle Berücksichtigung. Lichtzeichen (Ampeln und Warnlichter z.B. an Kreuzungsübergängen) jedoch führen häufige Zustandswechsel durch. Ohne Berücksichtigung dieser Zustandswechsel käme es leicht zu Verwechslungen von Warnlichtern und Ampeln. Ampeln würden zum Zeitpunkt des Zustandswechsels als Hintergrundobjekt abgelehnt oder als mehrere Objekte aufgefasst werden.

Aufgrund ihres Zustandswechselverhaltens können Lichtzeichen in wenige Gruppen gleichen Verhaltens unterteilt werden. Die drei wichtigsten sind Ampeln mit drei Lichtern (zyklische Zustandsfolge rot \rightarrow gelb-rot \rightarrow grün \rightarrow gelb), Ampeln mit zwei Lichtern (zyklische Zustandsfolge gelb \rightarrow rot \rightarrow aus) und Warnlichter (zyklische Zustandsfolge gelb \rightarrow aus). Die Zustandsschätzung soll kurzfristige Fehlerkennungen in der Prozesskette korrigieren. Die im vorherigen Abschnitt 3.4.2 vorgestellten Verfahren sind an dieser Stelle nicht geeignet, da es bei einem Zustandswechsel des beobachteten Objekts zu Fehlern käme: Das Objekt würde abgelehnt, in zwei Objekte aufgeteilt oder der Zustandswechsel erst nach einer Verzögerung von mehreren Bildern erkannt. Im Hinblick auf eine frühzeitige Erkennung beispielsweise von roten Ampeln ist dies nicht akzeptabel.

Abhilfe schafft die Modellierung des Ampelzustands als ein nicht direkt beobachtbarer Zustand. Die (verrauschten) Messungen der Ampelfarbe durch Detektion und Klassifikation stellen Beobachtungen dar, aufgrund derer auf den versteckten Zustand geschlossen wird. Diese Modellierung liegt hidden Markov models (HMMs) zugrunde [Rab89]. Zur Zustandsschätzung wird jede Lichtzeichengruppe durch ein eigenes HMM modelliert, dessen versteckten Zustände der aktuell angezeigten Lichtfarbe entsprechen. Die Ausgabesymbole sind die von der Perzeption beobachteten Farben; Fehler in der Erkennung werden also über die Emissionswahrscheinlichkeiten abgebildet [NDZ10]. Abbildung 3.25 zeigt die drei derart modellierten HMMs $\Lambda := \{\lambda_1, \lambda_2, \lambda_3\}$.

Sei $O = \{o^{(1)}, \dots, o^{(n)}\}$ eine beobachtete Zustandsfolge und $\Omega = \{D, R, B, G, A\}$. Die Klasse R steht hier für die Beobachtung rot, B für rot-gelb, G für grün, A für gelb und D für andere Objekte. Mittels

$$\lambda := \operatorname{argmax}_{\lambda_i \in \Lambda} P(O|\lambda_i) \quad [3.78]$$

wird das HMM berechnet, das mit maximaler Wahrscheinlichkeit die Beobachtung O erzeugt hat. Dieses Ergebnis wird als Klassifikation der Lichtzeichenart benutzt. Gleichung [3.78] kann mit Hilfe des Forward-Algorithmus effizient berechnet werden [Rab89]. Um eine Zustandsfolge abzulehnen – festzulegen, dass es sich nicht um ein Lichtzeichen handelt – kann entweder Λ um ein weiteres HMM λ_0 mit einem versteckten Zustand erweitert werden (Abbildung 3.25), oder es kann die Zusatzforderung

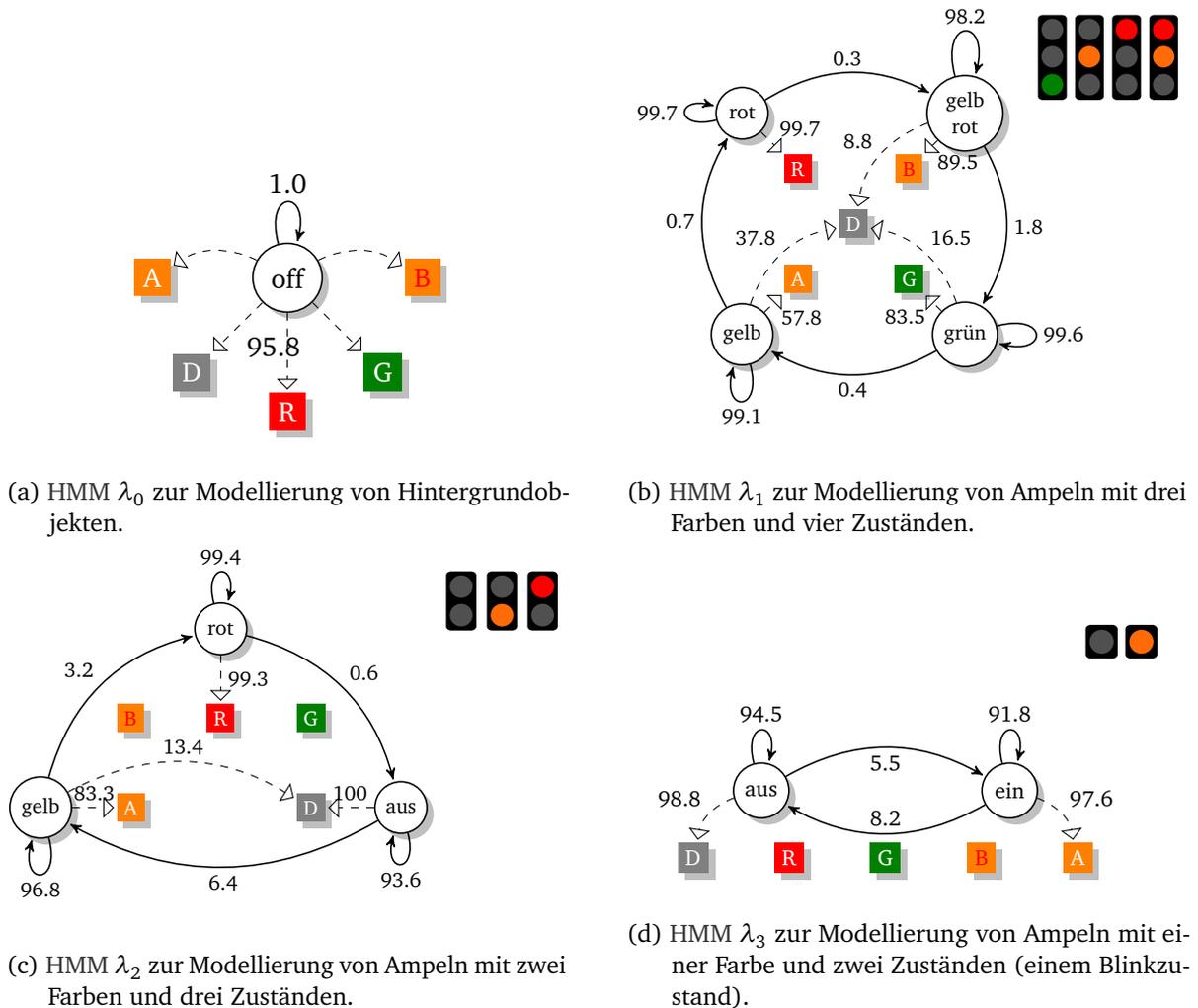


Abb. 3.25.: HMMs für verschiedene Arten von Ampeln sowie Hintergrundobjekte (siehe Text).

$P(O|\lambda) > \kappa(|O|)$ gestellt werden für einen geeigneten Schwellwert $\kappa(|O|)$.

Nach der Bestimmung der Lichtzeichenart gemäß Gleichung [3.78] wird die wahrscheinlichste Abfolge der versteckten Zustände bestimmt, um Erkennungsfehler zu eliminieren. Sei $Q = \{q_1, \dots, q_n\}$ eine Folge versteckter Zustände mit $q_i \in Q_\lambda$ und Q_λ die Menge der versteckten Zustände des HMM λ . Die wahrscheinlichste Abfolge Q_{\max} ist

$$Q_{\max} := \operatorname{argmax}_{Q_i \in Q} P(Q_i|O, \lambda) \tag{3.79}$$

und kann effizient mit Hilfe des Viterbi-Algorithmus berechnet werden [Rab89]. Insbesondere der aktuelle Zustand (letzter Zustand der wahrscheinlichsten Zustandsfolge) ist hierbei von Interesse. Wenn die komplette Zustandsabfolge nicht benötigt wird, kann er auch über den Forward-Algorithmus berechnet werden.

Die HMMs $\lambda_0, \dots, \lambda_3$ in Abbildung 3.25 enthalten bereits Übergangs- und Emissionswahrscheinlichkeiten. Initial können diese frei gewählt werden. Sie werden anschließend anhand gegebener Trainingsdaten mit Hilfe des Baum-Welch-Algorithmus [Rab89] dahingehend optimiert, dass die Wahrscheinlichkeit der

Beobachtungssequenzen in den Trainingsdaten maximiert wird.

3.5. Anwendungen

Verkehrselemente zeichnen sich durch eine Vielzahl unterschiedlicher Merkmale aus, für deren Erkennung in den vorherigen Abschnitten geeignete Methoden vorgestellt wurden. Zur Erkennung konkreter Arten von Verkehrselementen müssen passende Verfahren ausgewählt, parametrisiert und kombiniert werden. Dafür werden im Folgenden entsprechende Prozessketten (vergleiche Abbildung 3.1) vorgestellt. Einige Komponenten können generisch verwendet werden:

Komponente	Datenquelle / Verfahren
Sensorik	Farbvideokamera
Tracking	Kalman-Filter und Multi-Target-Tracker (MTT)
Klassifikation	Hierarchische SVM

Sie müssen aber für die jeweilige Anwendung parametrisiert werden, etwa durch Anpassung der Baumstruktur der hierarchischen SVM. Die übrigen Verfahren – Segmentierung, Detektion, Merkmalsextraktion und temporale Fusion – sind wie in den folgenden Abschnitten beschrieben problemspezifisch anzupassen.

3.5.1. Erkennung von Ronden

Eine Vielzahl von Verkehrszeichen besitzt eine runde Form. Aufgrund ihrer Farbgebung zeichnet sich ihre Form im Kantenbild deutlich ab. Sie werden in Fahrtrichtung so aufgestellt, dass sie im Bild der in Fahrtrichtung ausgerichteten Kamera als Kreis bzw. durch perspektivische Verzerrung annähernd als Kreis erscheinen. Für die Segmentierung eignet sich daher die radiale Symmetrie für Kreise (Abschnitt 3.2.4). Ronden werden laut Verwaltungsvorschrift zur Straßenverkehrs-Ordnung (VwV-StVO) [Sta] folgendermaßen aufgestellt:

Geschwindigkeitsbereich	Durchmesser
bis 20 km/h	42 cm
mehr als 20 bis 80 km/h	60 cm
mehr als 80 km/h	75 cm

Anhand des maximalen Durchmessers von 75 cm (oder 100 cm um die sehr selten verwendeten Übergrößen mit abzudecken), typischen Entfernungen zwischen Kamera und Verkehrszeichen sowie der verwendeten Optik der Kamera kann die maximale Größe der Abbilder von Ronden im Videobild berechnet werden. Sie geht als Parameter r_{\max} in die radiale Symmetrie ein. In Verbindung mit Bildpyramiden

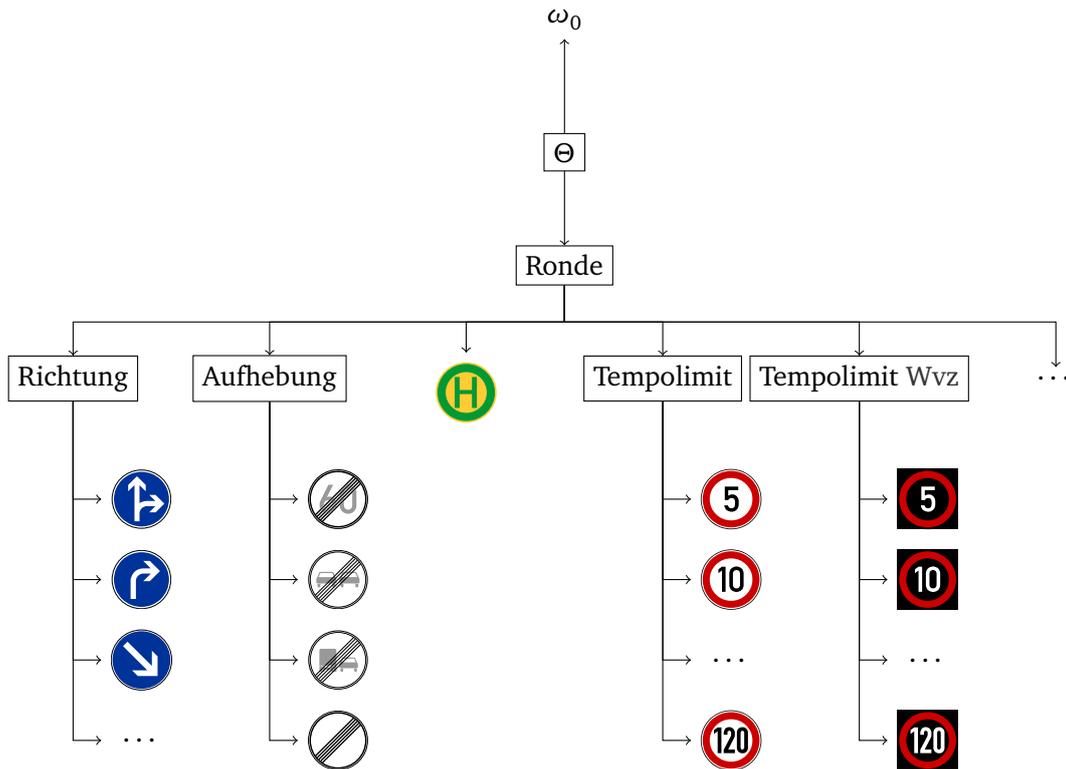


Abb. 3.26.: Gruppierung von Ronden aufgrund ähnlicher Merkmale und Anordnung in einer Baumstruktur für eine hierarchische SVM.

(Abschnitt 3.2.7) wird die Größe der Akkumulatoren begrenzt und dadurch die Laufzeit reduziert. Das maximale Pyramidenlevel ist in Abhängigkeit von der Auflösung der Kamera zu wählen.

Zur Klassifikation kommt eine hierarchische SVM (Abschnitt 3.3.6) mit der in Abbildung 3.26 skizzierten Baumstruktur zum Einsatz. Als Merkmalstransformation eignen sich sowohl die Grauwerttransformation als auch die blockweise diskrete Kosinus-Transformation (DCT). Die Segmentierung erkennt auch Wechselverkehrszeichen. Solche mit LED-Technik weisen stark unterschiedliche Merkmale auf verglichen mit Verkehrsschildern der gleichen Bedeutung. Daher werden sie durch eigene Klassen in Ω repräsentiert. Dies erlaubt auch die spätere Unterscheidung von Verkehrsschildern und Wechselverkehrszeichen mit LED-Technik (vergleiche Abschnitt 4.2).

Wechselverkehrszeichen in LED-Technik passen ihre Helligkeit an die Umgebungshelligkeit an. Das dafür notwendige Dimmen wird üblicherweise durch Pulsweitenmodulation realisiert, was abhängig von der Verschlusszeit der Videokamera in Bildflackern resultiert (Abbildung 3.27). Dies erschwert die Klassifikation, da Teile des Verkehrszeichens geringere Helligkeit aufweisen. Der Effekt kann durch einen einfachen Filter stark reduziert werden, indem das Bezugsbild \mathbf{B} der ROI $\Theta^{(t)}$ zum Zeitpunkt t durch das Bild \mathbf{B}' ersetzt wird. Dabei wird \mathbf{B}' als das komponentenweise Maximum von $\mathbf{B}^{(t)}$ und des größennormalisierten $\mathbf{B}^{(t-1)}$ gebildet [NGZ08]:

$$b'(\mathbf{x}) := \max \left(b_{\Theta^{(t)}}(\mathbf{x}), b_{\Theta^{(t-1)},(u,v)}(\mathbf{x}) \right) \quad [3.80]$$



Abb. 3.27.: Reduktion von Bildflackern durch Pulsweitenmodulation bei Wechselverkehrszeichen in LED-Technik. Jede Spalte stellt oben zwei aufeinander folgende ROIs des selben Wechselverkehrszeichens dar und darunter die verbesserte ROI.

mit $u \times v$ der Größe von $\mathbf{B}_{\Theta(t)}$. Abbildung 3.27 zeigt Beispiele dieser Operation angewendet auf die ursprüngliche ROI.

Neben der Segmentierung angewendet auf das Videobild wird das Tracking von ROIs genutzt, um für jede ROI eine weitere Segmentierung durchzuführen. Dabei wird die Suchregion eingeschränkt auf die lokale Umgebung der prädizierten Position. Die Suchradien der radialen Symmetrie für Kreise werden entsprechend der prädizierten Größe gewählt.

3.5.2. Erkennung von Warnhinweisen

Warnhinweise in Form dreieckiger Verkehrszeichen mit nach oben zeigender Spitze gibt es in einer Vielzahl von Varianten. Sie besitzen einen roten Rand auf weißem Grund mit unterschiedlichen Symbolen. Eine Ausnahme bildet lediglich das Andreaskreuz: Dieser Warnhinweis weist keine dreiecke Form auf. Das Richtzeichen *Vorfahrt gewähren* ist durch seine Form und Farbgebung sehr ähnlich. Es besitzt jedoch kein Symbol und eine andere Rotation (die Spitze zeigt nach unten). Warnhinweise sind größer als Ronden (vergleiche Abschnitt 3.5.1) und werden laut VwV-StVO [Sta] folgendermaßen aufgestellt:

Geschwindigkeitsbereich	Seitenlänge
weniger als 50 km/h	63 cm
50 bis 100 km/h	90 cm
mehr als 100 km/h	126 cm

Zur Segmentierung dreieckiger Warnhinweise eignen sich die in Abschnitt 3.2.3 vorgestellte radiale Symmetrie für gleichseitige n-Ecke (mit $n=3$) und das Viola/Jones-Verfahren (Abschnitt 3.2.6). Vorteil der radialen Symmetrie ist eine genauere Bestimmung von Skalierung und Position von Dreiecken. Außerdem kann sie im Gegensatz zum Viola/Jones-Verfahren die Rotation bestimmen. Demgegenüber weist das Viola/Jones-Verfahren eine geringere Laufzeit auf und ist weniger anfällig für Falschpositive, da neben den Kanten der Dreiecksseiten weitere Merkmale berücksichtigt werden.



Abb. 3.28.: Vorwegweiser dienen laut VwV-StVO sowohl der Unterrichtung von Ortsfremden als auch der Verteilung des Verkehrs auf das Straßennetz [Sta; Com10].

Die auf Warnhinweisen verwendeten Piktogramme sind sehr unterschiedlich und lassen sich nicht in Untergruppen mit ähnlichen Merkmalen aufteilen. Die hierarchische SVM zur Klassifikation verwendet daher zwei Stufen. Auf der ersten werden Hintergrundobjekte über die Klasse ω_0 abgelehnt. Dafür eignet sich als Merkmalstransformation die blockweise DCT. Die zweite Stufe unterscheidet die Typen von Warnhinweisen und setzt dazu eine Grauwerttransformation als Merkmalstransformation ein.

3.5.3. Erkennung von Vorwegweisern

Vorwegweiser sind komplexe, großflächige Verkehrszeichen, die auf einem einfarbigen Hintergrund Texte und Sinnbilder zeigen (Abbildung 3.28). Die auf ihnen enthaltenen Informationen können in vielen Fällen komplett durch Kartenmaterial ersetzt werden. Eine bildbasierte Erkennung ist dennoch relevant, um veraltetes bzw. unvollständiges Kartenmaterial kompensieren zu können, um auf die Kenntnis der Fahrzeugposition verzichten zu können und um dynamisch dargestellte Informationen (Abbildung 3.29) extrahieren zu können, die im Kartenmaterial nicht repräsentierbar sind.

Um die Bedeutung von Vorwegweisern zu erfassen, müssen sowohl die auf ihnen dargestellten Sinnbilder als auch die Texte erkannt werden. Dazu ist ein zweistufiges Segmentierungsverfahren geeignet, das zunächst Größe und Position eines Vorwegweisers bestimmt und danach innerhalb des Vorwegweisers eine Aufteilung in Text-, Sinnbild- und Hintergrundregionen vornimmt. Die extrahierten Text- und Sinnbildregionen werden dann getrennt voneinander per optical character recognition (OCR)-Analyse und Klassifikation weiterverarbeitet [XNZ09].

Detektion

Wegen der Einfarbigkeit des Hintergrunds des Vorwegweisers und seiner Größe eignet sich eine bildpunkt-basierte Segmentierung als erste Stufe der Detektion. Dazu wird das Eingabebild $\mathbf{B} \in \mathfrak{R}$ in den HSV-Farbraum transformiert und gemäß [3.5] die Segmentierung durchgeführt. Als h wird

$$h(\mathbf{x}) := h_h(\mathbf{x}) > \kappa_{h1} \wedge h_h(\mathbf{x}) < \kappa_{h2} \wedge h_s(\mathbf{x}) > \kappa_s \wedge h_v(\mathbf{x}) > \kappa_v \quad [3.81]$$



Abb. 3.29.: Verkehrsbeeinflussungsanlagen zeigen dynamisch Umleitungen und Routenempfehlungen an. Der abgebildete Vorwegweiser verwendet dafür sowohl die Prismen- als auch die LED-Technik.

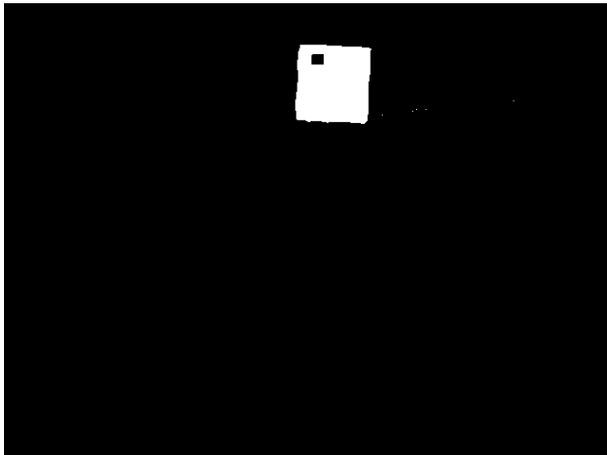
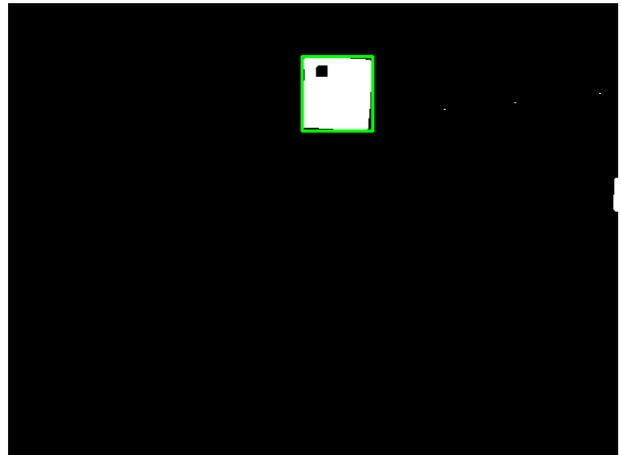
verwendet mit empirisch festgelegten Schwellwerten κ_{h1} , κ_{h2} , κ_s und κ_v . Das Ergebnis ist das Binärbild $B_0 \in \mathfrak{B}$. Durch morphologisches Schließen (Definition 3.9) mit einem rechteckigen strukturierenden Element $C \in \mathfrak{B}^{2,2}$ mit $c(x) := 1 \forall x \in C$ wird Rauschen in Form kleiner Objekte unterdrückt und man erhält das Binärbild $B_1 \in \mathfrak{B}$ mit

$$B_1 := B_0 \bullet C. \quad [3.82]$$

Die Extraktion von Zusammenhangskomponenten liefert schließlich eine Menge von ROIs, von denen diejenigen mit ausreichender Größe das Ergebnis der Segmentierung darstellen. Abbildung 3.30 zeigt die Ergebnisse der einzelnen Schritte an einem Beispiel.

Um die Klassifikation der Symbole und die Extraktion des Textes zu erleichtern, wird die ROI in der zweiten Segmentierungsstufe zunächst normalisiert. Optimal wäre eine Transformation, welche die perspektivische Verzerrung ausgleicht. Die für diese Transformation notwendigen Parameter können aus den vier Eckpunkten der ROI direkt berechnet werden [Jä05]. Problematisch bei diesem Vorgehen ist allerdings, dass im Allgemeinen keine robuste Extraktion der vier Eckpunkte gelingen wird; kleinere Abweichung würden sich stark verfälschend auf das normalisierte Bild auswirken. Aus diesem Grund wird eine vereinfachte, robustere Transformation zur Normalisierung angewandt, die lediglich die Rotation der ROI ausgleicht. Dazu werden die Gradienten der Kanten des Vorwegweisers betrachtet und der am häufigsten vorkommende Gradientenwert wird als Rotation der ROI gesetzt. Zur Umsetzung berechnet man zu G_Θ nach Formel [3.8] G_x und G_y und maskiert innere Bildpunkte (deren Gradienten nicht betrachtet werden sollen). Daraus bildet man $A \in \mathbb{R}^{h,w}$ mit

$$a(x) := \tan \left(\frac{g_y(x)}{g_x(x)} \right) \text{ mod } \pi. \quad [3.83]$$

(a) Eingangsbild B (b) Farbklassifikation B_0 (c) Morphologisches Schließen B_1 

(d) Zusammenhangskomponenten

Abb. 3.30.: Verarbeitungsschritte der ersten Segmentierungsstufe.

Die Matrix A enthält die Gradientenwerte aus den Randbereichen der ROI, wobei orthogonale Gradienten auf den gleichen Wert abgebildet werden. Alle vier Kanten des Vorwegweisers werden im Idealfall also auf identische Gradientenwerte abgebildet. Um den dadurch zu erwartenden Peak (Spitzenwert) in A zu extrahieren, wird ein Histogramm H auf A verwendet. Der maximale Wert h_{\max} des Histogramms H wird extrahiert und die normalisierte ROI $\hat{\Theta}$ wird gebildet, indem Θ um $-h_{\max}$ rotiert wird. Abbildung 3.31 veranschaulicht die Rotationsnormalisation.

Nach der vorbereitenden Rotationsnormalisation macht man sich zur Extraktion der Symbol- und Textregionen zunutze, dass das Ansichtsverhältnis von Symbolen nahezu quadratisch ist, während Textzeilen eine größere Breite als Höhe aufweisen. Über ein dichtebasiertes Region Growing auf dem Canny-Bild C von $G_{\hat{\Theta}}$ erhält man Unterregionen [YH05]. Über das horizontale Projektionsprofil werden einzelne Textzeilen voneinander getrennt. Die Aufteilung in Text- und Symbolregionen erfolgt über einen Schwellwert für das Ansichtsverhältnis der extrahierten Regionen. Abbildung 3.32 zeigt ein Beispiel für die Zwischenschritte in der zweiten Segmentierungsstufe.



(a) Eingangsbild mit R_0

(b) Rotationsnormalisation R_1

Abb. 3.31.: Rotationsnormalisation eines Bildes basierend auf der erkannten ROI. Perspektivische Verzerrungen werden durch die Rotationsnormalisation nicht ausgeglichen.



(a) ROI

(b) Canny-Bild

(c) Dichtebasiertes Region Growing

(d) Text- und Symbolregionen

Abb. 3.32.: Segmentierung in Text- und Symbolregionen.

Texterkennung und Sinnbild-Klassifikation

Die Erkennung von Texten in als Textregionen deklarierten ROIs kann durch existierende OCR-Verfahren erfolgen. Eine Verbesserung der Erkennungsleistung kann erreicht werden, indem die Erkennung auf die auf allen Verkehrszeichen verwendeten Varianten der DIN-Schrift (DIN 1451) optimiert wird. Außerdem wird ein Wörterbuch mit den auf den Vorwegweisern vorkommenden Distanzen, Ortsnamen und verwandten Worten eingesetzt. Dieses Wörterbuch kann (sofern verfügbar) mit Hilfe von Kartendaten und der Kenntnis der Fahrzeugposition weiter eingegrenzt werden. Ergebnisse der OCR-Erkennung werden gemäß der minimalen Editierdistanz (Levenshtein-Distanz) zu Wörterbucheinträgen korrigiert bzw. bei zu hoher Distanz verworfen.

Zur Erkennung der Symbole eignet sich eine SVM unter Verwendung einer Grauwerttransformation. Abbildung 3.33 zeigt Beispiele der Grauwerttransformation von extrahierten Symbolregionen.

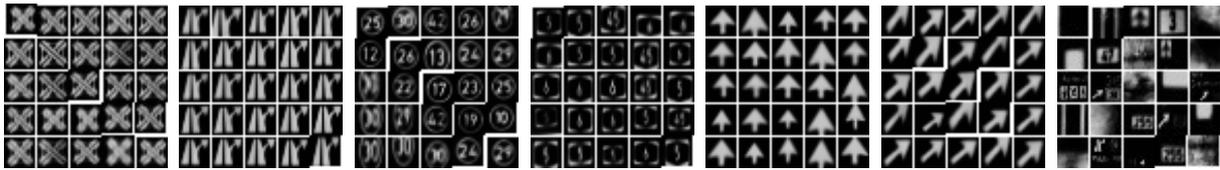


Abb. 3.33.: Grauwerttransformation extrahierter Symbolregionen, gruppiert nach Symbolart.

Geschwindigkeitsbereich	Größe 1 (mm)	Größe 2 (mm)	Größe 3 (mm)
20 bis weniger als 50 km/h	231 × 420	315 × 420	420 × 420
50 bis 100 km/h	330 × 600	450 × 600	600 × 600
mehr als 100 km/h	412 × 750	562 × 750	750 × 750

Tab. 3.4.: Größen von Zusatzzeichen für unterschiedliche Ansichtsverhältnisse und Geschwindigkeitsbereiche.

3.5.4. Erkennung von Zusatzzeichen

Zusatzzeichen ähneln Vorwegweisern insofern, als dass sie ebenfalls Texte und/oder Piktogramme auf einem einfarbigen (weißen) Hintergrund zeigen. Aufgrund ihrer geringen Größe (Tabelle 3.4) würde der für Vorwegweiser gewählte Ansatz über Farbklassifikation und OCR-Erkennung allerdings keine robusten Ergebnisse erzielen. Ihre laut Straßenverkehrs-Ordnungs (StVOs) [Sta13] geforderte Anbringung unmittelbar unterhalb anderer Verkehrszeichen kann jedoch für die Segmentierung in Form einer starken Einschränkung der Suchregionen ausgenutzt werden, indem Zusatzzeichen nur unterhalb anderer erkannter Verkehrszeichen gesucht werden. Folglich kann die in Abschnitt 3.2.2 vorgestellte Hough-Transformation für Rechtecke in Echtzeit ausgeführt und zur Segmentierung von Zusatzzeichen genutzt werden. Die auf ihnen verwendeten Texte sind im Unterschied zu Vorwegweisern auf wenige bekannte, ortsunabhängige Begriffe eingeschränkt. Dadurch ist es möglich, sie ähnlich wie Piktogramme zu behandeln und die Erkennung des gesamten Zusatzzeichens wie in Abbildung 3.34 skizziert über eine hierarchische SVM durchzuführen [Nie+10].

3.5.5. Erkennung von Ampeln

Ampeln bestehen aus bis zu drei übereinander montierten Lichtern, die fest in einem Ampelkasten verbaut sind und in der Regel mit zusätzlichen Kontrastblenden zur besseren Sichtbarkeit für Autofahrer versehen werden. Sie können bildbasiert detektiert und ihr Zustand kann festgestellt werden.

Zur Segmentierung kommen mehrere charakteristische Merkmale von Ampeln in Frage. Der Ampelkasten ist meist rechteckig und daher durch eine Hough-Transformation für Rechtecke (Abschnitt 3.2.2) extrahierbar. Der Kontrast zur Umgebung ist allerdings nicht immer stark ausgeprägt, so dass mit vielen Falschpositiven bzw. Falschnegativen zu rechnen wäre. Außerdem ist dieser Ansatz, ausgeführt auf einem kompletten Bild, laufzeittechnisch aufwendig. Besser eignet sich die Extraktion der Ampellichter als Basismerkmal. Da die Lichter meist rund sind, könnte die radiale Symmetrie für Kreise (Abschnitt 3.2.4) zum Einsatz kommen. Hierdurch wären jedoch keine robusten Ergebnisse zu erwarten, da die Ampellichter mit 20–30 cm Durchmesser deutlich kleiner als Ronden sind. Außerdem ist durch Überstrahlung,

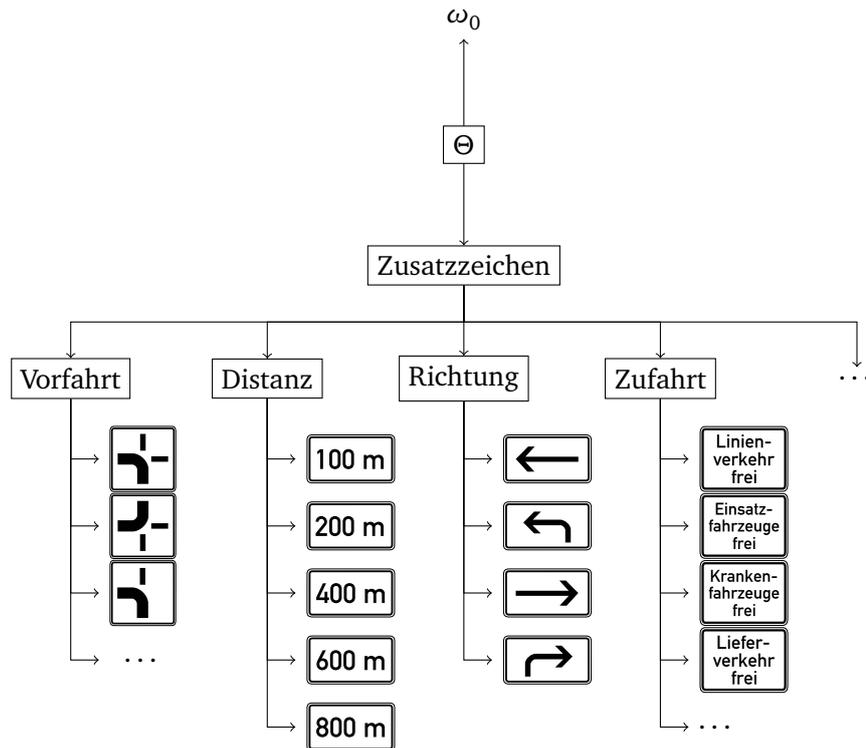


Abb. 3.34.: Auszug der Struktur einer hierarchischen SVM zur Klassifikation von Zusatzzeichen.

Reflektion, Abschirmblenden (Schuten) und pfeilförmige Lichter die Kreisform in sehr vielen Fällen im Bild nicht gegeben. Aus diesen Gründen wird auf eine primär kantenbasierte Segmentierung verzichtet und stattdessen die im Folgenden vorgestellte Kombination von Farbklassifikation, morphologischer Bildverarbeitung und regelbasierter Filterung zur Detektion von Ampellichtern benutzt. Anschließend wird ausgehend vom Ampellicht der Ampelkasten geschätzt und seine Parameter in einer Postsegmentierung angepasst. Dazu wird ein Gütekriterium bestehend aus Kanten- und Farbmerkmalen optimiert. Die Klassifikation schließlich verwendet eine SVM mit einer hybriden Merkmalstransformation.

Detektion von Ampellichtern

Wie in Abschnitt 3.2.5 beschrieben eignet sich der morphologische Top-Hat-Operator bei der Wahl eines geeigneten strukturierenden Elements dazu, großflächige Objekte zu maskieren, ohne Ampellichter zu verändern. Er wird auf das Bild $\mathbf{G} \in \mathcal{G}^{h,w}$ angewendet, das durch eine Grauwerttransformation aus dem Eingangsbild \mathbf{B} entsteht:

$$\mathbf{T} := \mathbf{G} \odot \mathbf{K} \tag{3.84}$$

mit einem kreisartigen strukturierenden Element \mathbf{K} analog zu Abbildung 3.9e.

Des Weiteren wird eine bildpunkt-basierte Segmentierung auf dem Bild $\mathbf{H} \in \mathcal{H}^{h,w}$ ausgeführt, das durch eine HSV-Farbraumtransformation aus \mathbf{B} entsteht. Die Farbsegmentierung wird gemäß [3.81] durchgeführt. Zur Vermeidung von Falschnegativen werden die Schwellwerte κ_{h1} , κ_{h2} , κ_s und κ_v schwach filternd eingestellt; hauptsächlich werden über den Farbkanal blaue Bildpunkte und über den Grauwert

dunkle Bildpunkte maskiert. Das Ergebnis sind die ROIs Θ_0^H (Hintergrund) und Θ_1^H (Ampelfarben).

Die Ergebnisse des Top-Hat-Operators und der Farbsegmentierung werden zu einem Binärbild \mathbf{B}_0 mit

$$b(\mathbf{x}) := \begin{cases} 1, & \mathbf{x} \in \Theta_1 \wedge t(\mathbf{x}) > \kappa_T \\ 0, & \text{andernfalls} \end{cases} \quad [3.85]$$

kombiniert, das kleine Regionen heller Bildpunkte in Ampelfarben enthält. Durch die Bestimmung von Zusammenhangskomponenten in diesem Bild erhält man eine Menge von ROIs.

Um die Berechnung zu beschleunigen, wird als Eingabebild nicht das ursprüngliche Kamerabild verwendet, sondern mehrere kleine Suchbilder. Sie werden abhängig von der Ausrichtung der Kamera, den möglichen Größen von Ampellichtern (20 oder 30 cm) und deren Aufhängungshöhen mit Hilfe der in Abschnitt 3.2.7 beschriebenen Bildpyramiden berechnet. Damit kann die Größe des strukturierenden Elements \mathbf{K} stark begrenzt werden, wodurch die Laufzeit deutlich reduziert wird.

Für jede ROI Θ werden anschließend vier Kriterien überprüft. Wird mindestens eines verletzt, wird Θ aus der Menge der ROIs entfernt. Sowohl die Größe von Θ als auch sein Ansichtsverhältnis muss innerhalb eines festgelegten Bereichs liegen. Basierend auf der Größe von Θ und einem angenommenen Durchmesser von 25 cm wird seine Position im Fahrzeugkoordinatensystem berechnet; die geschätzte Höhe der Ampel wird auf einen festgelegten Bereich geprüft. Als drittes Kriterium wird das Verhältnis der Anzahl Bildpunkte in Θ zur Anzahl Bildpunkte der konvexen Hülle von Θ überprüft; es darf einen festgelegten Minimalwert nicht unterschreiten. Auf diese Weise werden Regionen mit größeren Löchern und stark konvexe Regionen aussortiert. Schließlich wird die Exzentrizität [Jä05] von Θ bestimmt und Θ abgelehnt, wenn sie auf ein deutlich linienförmiges Objekt hinweist. Abbildung 3.35 veranschaulicht die Detektion an einem Beispiel.

Detektion des Ampelkastens

Das Verhältnis der Größe von Ampellichtern zur Größe des Ampelkastens ist für alle Ampeln sehr ähnlich, so dass eine Schätzung des Ampelkastens basierend auf der Größe und Position des Ampellichts möglich ist. Dies setzt voraus, dass die Ampelfarbe basierend auf der HSV-Farbklassifikation richtig geschätzt wird und dass die Größenbestimmung des Ampellichts exakt ist. Insbesondere letzteres kann nicht immer garantiert werden, da durch Überstrahlung, Reflektionen und Fehler in der Segmentierung Ungenauigkeiten entstehen. Ausgehend von der geschätzten Position des Ampelkastens \mathbf{x} , seiner Größe $w \times h$ und seiner Rotation α wird das Ampelkastenmodell als Tupel $K = (\mathbf{R}, \mathbf{x}, w, h, \alpha)$ per Gradientenabstiegsverfahren optimiert. Als Gütemaß $k(K)$ kommt eine gewichtete Kombination von Gütemaßen für Farb- und Kantenmerkmale zum Einsatz:

$$k(K) := \mu k_c(K) + k_e(K) \quad [3.86]$$

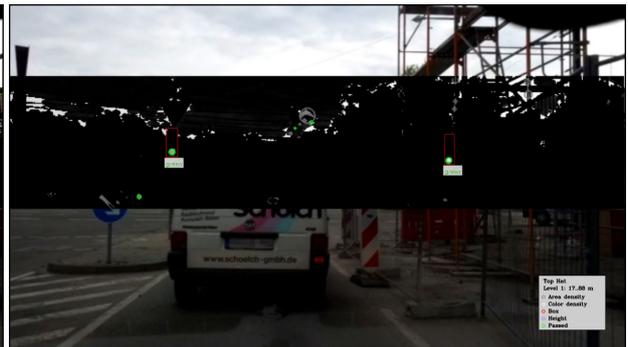
Das Gütemaß für Farbmerkmale $k_c(K)$ bewertet die Hintergrundbereiche des durch K gegebenen



(a) Originalbild mit eingezeichnetem Erkennungsergebnis.



(b) Bildpyramide Stufe 0 (Originalbild)



(c) Bildpyramide Stufe 1



(d) Bildpyramide Stufe 2



(e) Bildpyramide Stufe 3

Abb. 3.35.: Segmentierung von Ampellichtern mittels TopHat-Operator, Farbklassifikation und Validierungsregeln. Die Bilder (b)–(e) zeigen die anhand der Parameter der Kamerakalibrierung ermittelten Suchregionen für verschiedene Stufen der Bildpyramiden, die jeweils einem Distanzbereich entsprechen.

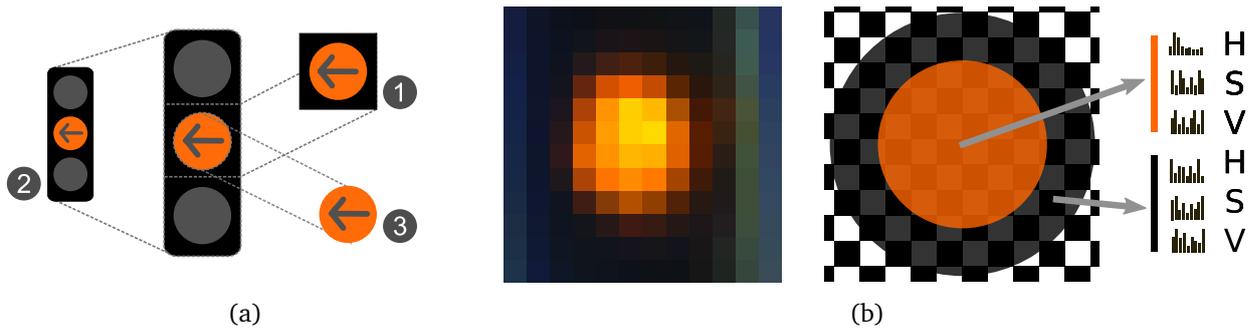


Abb. 3.36.: Ampeln enthalten wie in (a) dargestellt verschiedene nützliche Merkmale zur Klassifikation: Den Ampelkasten an sich (2) sowie das aktive Licht (3) und seinen Hintergrund (1). Farbhistogramme angewendet auf die Licht- und Hintergrundbereiche wie in (b) skizziert extrahieren Farbmerkmale.

Ampelkastens: Achromatische Bildpunkte geringer Sättigung werden positiv, andere negativ bewertet. Das Gütemaß für Kantenmerkmale $k_e(K)$ arbeitet auf dem Kantenbild zu Θ und bewertet Kantenpixel im Randbereich positiv, wenn ihr Gradient die Richtung α bzw. $\alpha + \pi/2$ aufweist. Andernfalls werden Bildpunkte im Randbereich negativ bewertet. Der Innenbereich wird von k_e ignoriert.

Klassifikation

Sowohl die Form als auch die Farben von Ampeln sind nützliche Merkmale für die Klassifikation. Aus diesem Grund wird eine hybride Merkmalstransformation (Abschnitt 3.3.5) verwendet, die eine HOG-Transformation (Abschnitt 3.3.4) zur Repräsentation von Formmerkmalen mit einem blockweisen Farbhistogramm (Abschnitt 3.3.3) im HSV-Farbraum kombiniert. Das Farbhistogramm wird wie in Abbildung 3.36 dargestellt auf unterschiedliche Regionen angewandt, um getrennte Histogramme für die einzelnen Ampellichter und den Ampelkasten zu extrahieren. Die Klassifikation der extrahierten Merkmale wird mit Hilfe einer SVM durchgeführt.

Temporale Fusion

Zur Validierung und zur Unterscheidung zwischen Warnlichtern und Ampeln wird für die in Abschnitt 3.4.3 vorgestellten HMMs die Wahrscheinlichkeit berechnet, dass sie die vorliegende Beobachtungsreihe O erzeugen. Für das gemäß Formel [3.78] berechnete Ergebnis $\lambda = \lambda_0$ wird das Objekt abgelehnt und andernfalls als die zugehörige Lichtzeichenart klassifiziert. Mit Hilfe des Viterbi-Algorithmus wird die wahrscheinlichste Zustandsfolge Q_{\max} nach Formel [3.79] berechnet und der letzte Zustand von Q_{\max} als Schätzung der aktuellen Ampelfarbe gesetzt.

3.6. Zusammenfassung

Die Interpretation einer Verkehrssituation basiert darauf, dass die in ihr enthaltenen Objekte und ihre Eigenschaften bekannt sind. Angesichts der in Abschnitt 1.2 definierten Rahmenbedingungen

sind Videokameras geeignete primäre Informationsquellen zur Objekterkennung. In diesem Kapitel wurden dafür Methoden und Verfahren vorgestellt. Sie leisten nicht nur die eigentliche Erkennung, sondern berücksichtigen darüber hinaus Laufzeiteinschränkungen, ohne auf Robustheit zu verzichten. Um dies zu erreichen, wird die Form von Verkehrselementen als Hauptmerkmal zur Detektion verwendet. Modellbasierte Segmentierungsverfahren dienen als robuste Grundlage der Detektion; verschiedene Laufzeitoptimierungen wurden vorgestellt, mittels derer ein Echtzeit-Einsatz möglich ist.

Zur Unterscheidung von Verkehrselementen wurde der Einsatz von Support-Vektor-Maschinen präsentiert. Um sie universell einsetzen zu können, werden verschiedene Merkmalstransformationen genutzt, die eine kompakte Repräsentation der charakteristischen Eigenschaften der verschiedenen Verkehrselemente erzielen. Auch hier wurde besonderer Augenmerk auf Laufzeitoptimierungen und Robustheit gelegt. Dazu wurden hierarchische SVMs vorgestellt, die eine schnellere Klassifikation bei einer großen Klassenanzahl ermöglichen als übliche Mehrklassen-Ansätze wie *one-against-one* und *one-against-all*. Das im Bereich des aktiven Lernens entwickelte Verfahren TPCB-ADR verbessert die Klassifikationsrate durch intelligente Auswahl von Lernbeispielen. Zusätzlich reduziert es den Annotationsaufwand und (bei gleicher Klassifikationsrate) die Laufzeit.

Die Berücksichtigung des zeitlichen Zusammenhangs in der Verarbeitung von Sensordaten wird durch das Tracking von Objekten und die temporale Fusion behandelt. Das Tracking wird in Fahrzeugkoordinaten durchgeführt, was bei gleichbleibender Fahrzeuggeschwindigkeit durch einen linearen Kalman-Filter erledigt werden kann. Es prüft die örtliche Konsistenz von Beobachtungen und ordnet sie Objekten der Verkehrssituation zu. Die Konsistenz des Zustands wiederum wird durch die temporale Fusion sichergestellt. Dazu wurden verschiedene Verfahren für statische und HMMs für semi-statische Verkehrselemente vorgestellt.

Die verschiedenen Methoden und Verfahren sind zur modularen Anwendung innerhalb der in Abbildung 3.1 dargestellten Prozesskette konzipiert. Für konkrete Verkehrselemente ergeben sich dadurch komplette Erkennungsverfahren. Entsprechende Parametrierungen wurden für die wichtigsten deutschen Verkehrselemente vorgestellt. Damit ist die Grundlage für die Interpretation von Verkehrssituationen gelegt, die im folgenden Kapitel untersucht wird.

4. Situationsinterpretation

In der Verkehrsdomäne treten eine Vielzahl unterschiedlicher Objekte auf, die miteinander in Beziehung stehen. Die Bedeutung eines Objekts für die Fahraufgabe kann nur dann richtig erfasst werden, wenn seine Beziehungen zu anderen Objekten berücksichtigt werden. Beispielsweise hat ein Stoppschild an einer Kreuzung bei aktiver Ampelschaltung keine Fahrrelevanz. Ein Tempolimit für die Nebenspur kann ignoriert werden, solange nicht auf diese Spur gewechselt wird. Für die Interpretation eines statischen Verkehrselements bedeutet dies, dass seine Bedeutung nur dann korrekt eingeordnet werden kann, wenn andere statische Verkehrselemente sowie weitere konkrete und abstrakte Objekte – der Kontext – und ihre Beziehungen untereinander in der Interpretation berücksichtigt werden. Im Sinne einer maschinellen Verarbeitung müssen dafür einerseits die verschiedenen Objekte (und ihre Eigenschaften) sowie ihre Beziehungen identifiziert werden. Andererseits müssen diese Objekte und Beziehungen in eine geeignete maschinelle Repräsentation gebracht und ausgewertet werden.

In diesem Kapitel werden geeignete Verfahren aus dem Bereich der Situationsinterpretation entwickelt. Zunächst wird in Abschnitt 4.1 untersucht, wie die räumliche Beziehung zwischen Verkehrszeichen und Fahrspuren ausgewertet werden kann, um festzustellen, auf welche Fahrspur bzw. Fahrspuren sich Verkehrszeichen beziehen. Die kontextsensitive Dempster/Shaffer-Fusion in Abschnitt 4.2 zeigt, dass Informationen über Baustellen, Tageszeit und andere Entitäten die Leistung eines Ansatzes zur Informationsfusion steigern. Markov-Logik-Netze werden als Framework zur Bestimmung der Fahrrelevanz in Abschnitt 4.3 eingeführt. Mit ihrer Hilfe werden Eigenschaften und Beziehungen zwischen Objekten verknüpft, um auf die Bedeutung von Verkehrselementen in der gegebenen Fahrsituation zu schließen.

4.1. Modellbasierte Bestimmung räumlicher Relationen

Um die Bedeutung von Verkehrselementen zu erfassen muss ihre räumliche Anbringung mit berücksichtigt werden. Der Gültigkeitsbereich von Halte- und Parkverboten ergibt sich beispielsweise aus ihrer Position, eventuell angebrachten Pfeilen und Zusatzzeichen und dem Verlauf der Fahrflächen in der Nähe. Verkehrselemente, die über einer Fahrspur montiert sind, gelten nur für diese Spur. In diesem Abschnitt wird ein modellbasierter Ansatz entwickelt, der die räumlichen Relationen *neben* und *über* zwischen Verkehrszeichen und Fahrspuren bestimmt. Diese Relationen gehen (direkt oder indirekt) in die Bestimmung der Fahrrelevanz von Verkehrszeichen ein.

In Deutschland sind zwei Konventionen zur Ausrichtung von Verkehrszeichen in Bezug auf Fahrspuren verbreitet: Stehen Verkehrszeichen rechts oder links der Straße, gelten sie für alle Fahrspuren. Werden mehrere Verkehrselemente auf Schilderbrücken über der Straße angebracht, beziehen sie sich nur auf die darunter befindliche Fahrspur (vergleiche Abbildung 4.1). Um ein Verkehrszeichen richtig einer oder



Abb. 4.1.: Unterschiedliche Geltungsbereiche von Verkehrszeichen abhängig von ihrer Anbringung.

mehreren Fahrspuren zuzuordnen, müssen daher die räumlichen Relationen zwischen den sichtbaren Verkehrszeichen und Fahrspuren analysiert werden. Der im Folgenden vorgestellte probabilistische Ansatz leistet dies [NGZ08].

Sei \mathbf{x}_B der Mittelpunkt einer positiv klassifizierten ROI Θ und b_B ihre Breite. Gemäß Formel [3.63] kann die Position des zugehörigen Verkehrszeichens \mathbf{x}_F in Fahrzeugkoordinaten berechnet werden als

$$\mathbf{x}_F = \mathbf{R}^\top \left(\begin{pmatrix} d \\ \frac{(p_x - x)d}{f_x} \\ \frac{(p_y - y)d}{f_y} \end{pmatrix} - \mathbf{t} \right) \quad [4.1]$$

mit

$$d = \frac{b_F}{b_B} f_x.$$

Hierbei ist b_F die Breite des Verkehrszeichens und f_x die Brennweite des Kamerasystems. Für die Zuordnung wird die Position $\mathbf{x}_F \in \mathbb{R}^3$ auf eine flache Welt $\mathbf{x}_E \in \mathbb{R}^2$ reduziert, also die z -Komponente von \mathbf{x}_F ignoriert. Sei $S \in \mathbb{R}^2$ eine Zufallsvariable, die diese Position darstellt. Sie wird mittels einer bivariaten Normalverteilung modelliert:

$$S \sim \mathcal{N}_2 \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \mathbf{x}_F, \mathbf{Y} \right) \quad [4.2]$$

mit Kovarianzmatrix \mathbf{Y} (vergleiche Abbildung 4.2). Die Varianz kann anhand der Varianz von Breiten- und Mittelpunktbestimmung von ROIs aus Formel [4.1] bestimmt werden. Für jede zu einem beliebigen Zeitpunkt von der Umfelderkennung detektierte und positiv klassifizierte ROI Θ_i , $i = 1, \dots, n$ wird eine

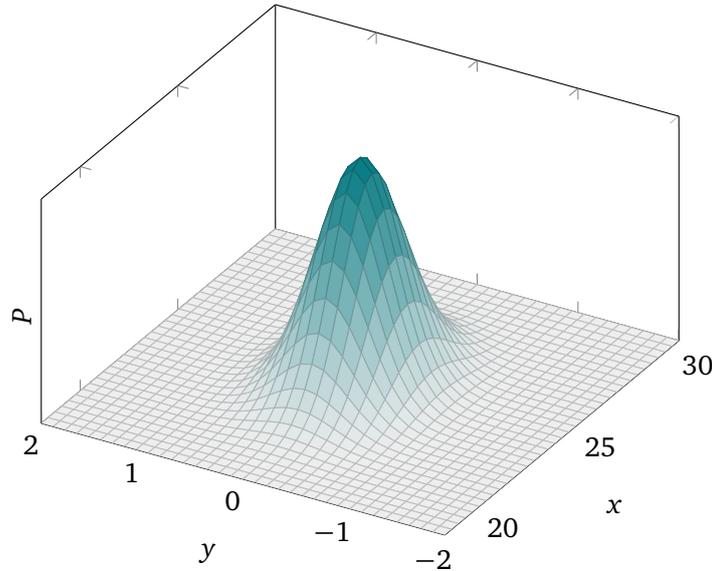


Abb. 4.2.: Beispielhafte Wahrscheinlichkeitsdichte für die Positionsschätzung eines Verkehrszeichens.

Zufallsvariable S_i gebildet.

Zur Erkennung von Fahrspuren wird z.B. der in [NGZ08] beschriebene Ansatz benutzt, der mehrere Fahrspuren gleichzeitig mit jeweils einem Kalman-Filter innerhalb eines festgelegten Sichtbereichs in einer Entfernung bis zu x_{\max} verfolgt. Jedes Kalman-Filter schätzt ein Tupel $\Psi = (o, w, c, \dot{c})$ bestehend aus dem horizontalen Abstand o des Fahrspurmittelpunkts zum Ursprung des Fahrzeugkoordinatensystems, der Fahrspurbreite w , der Krümmung c und der Krümmungsänderung \dot{c} . Zusätzlich zu den Fahrspuren werden die äußeren Fahrbahnrande mitgeführt, über denen sich seitlich von der Straße montierte Verkehrselemente befinden. Sie werden automatisch aus den beiden äußeren Fahrspuren geschätzt, indem deren Parameter übernommen und der seitliche Abstand vergrößert wird. Auf diese Weise erhält man m Tupel Ψ_j , die jeweils eine Fahrspur beziehungsweise einen Fahrbahnrand repräsentieren. Im Abstand $\ell \leq x_{\max}$ zum Fahrzeug auf der Mitte der Fahrspur ergibt sich aus einem Tupel Ψ der Punkt \mathbf{x}_E im Koordinatensystem auf der Bodenebene (vergleiche Abbildung 4.3) gemäß

$$\mathbf{x}_E = \begin{pmatrix} \ell \\ o + \frac{c}{2}\ell^2 + \frac{\dot{c}}{6}\ell^3 \end{pmatrix}. \tag{4.3}$$

Sei $L_j \in \mathbb{R}^2$ eine Zufallsvariable, die beschreibt, ob der zugehörige Punkt der Bodenebene Teil der Fahrspur Ψ_j ist. Ihre Dichte $f_L(\mathbf{x}_E)$ wird über eine Rechteckfunktion modelliert mit

$$f_L(\mathbf{x}_E) = \begin{cases} \frac{1}{(w - \kappa_m)x_{\max}}, & |o + \frac{c}{2}x^2 + \frac{\dot{c}}{6}x^3 - y| < \frac{w}{2} - \kappa_m, 0 \leq x \leq x_{\max} \\ 0, & \text{sonst} \end{cases} \tag{4.4}$$

und einem Rand der Breite $\kappa_m \geq 0$ zum Äußeren der Fahrspur.

Die Zufallsvariable $M_{ij} \in \mathbb{R}^2$ gibt an, ob sich an einem Punkt der Bodenebene sowohl die ROI Θ_i als

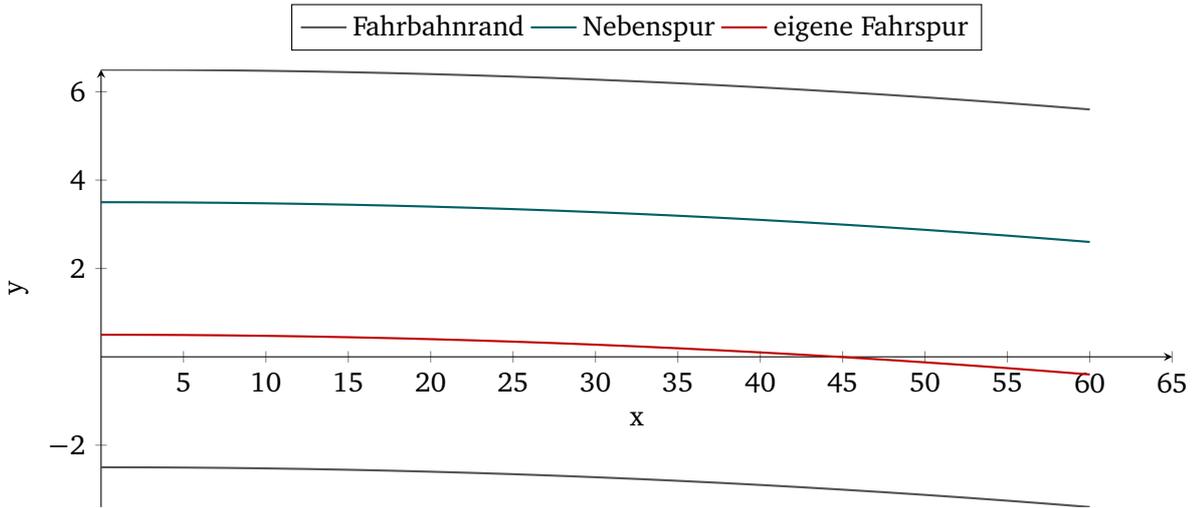


Abb. 4.3.: Beispielhafte Zustandsschätzung der Fahrspurerkennung für eine zweispurige Straße mit leichter Rechtskurve. Dargestellt ist jeweils der Fahrspurmittelpunkt x_E im Bereich $[0, x_{\max} = 60]$.

auch die Fahrspur Ψ_j befinden. Unter der Annahme stochastischer Unabhängigkeit ist

$$P(M_{ij}) = P(S_i)P(L_j). \quad [4.5]$$

Als diskrete Approximation der Wahrscheinlichkeitsdichte von M_{ij} wird die Häufigkeitsdichte $h_{M_{ij}}(x, y)$ eingeführt. Für eine gegebene Entfernung x und gegebene i, j ist

$$g_{M_{ij}}(x) := \max_y (h_{M_{ij}}(x, y)) \quad [4.6]$$

die Wahrscheinlichkeit der besten (wahrscheinlichsten) Zuordnung von Θ_i und Ψ_j . Die Zuordnung aller sichtbaren ROIs zu einer beliebigen Fahrspur maximaler Wahrscheinlichkeit für eine gegebene Entfernung x (eine Schilderbrücke in Entfernung x) ist damit

$$\sum_{i=1}^n \max_{j=1}^m (g_{M_{ij}}(x)). \quad [4.7]$$

Damit ergibt sich die Distanz der wahrscheinlichsten Zuordnung gemäß

$$\max_x \left(\sum_{i=1}^n \max_{j=1}^m (g_{M_{ij}}(x)) \right) \quad [4.8]$$

und die wahrscheinlichste Zuordnung M_{ij} für $i = 1, \dots, n$ entsteht als Nebenprodukt der Berechnung. Durch die Nutzung der Häufigkeitsdichte $h_{M_{ij}}$ lässt sich Formel [4.8] schnell berechnen, so dass der Laufzeitaufwand zur Berechnung der Zuordnung zu vernachlässigen ist im Vergleich zur Perzeption.

Aus den wahrscheinlichsten Zuordnungen ergibt sich, welches Verkehrszeichen aufgrund seiner Anbringung relevant ist: Jene, die einem Fahrbahnrand oder der eigenen Fahrspur (Ψ_j mit minimalem seitlichen Abstand) zugeordnet werden. In vielen Fällen sind diese Verkehrszeichen auch fahrrelevant (vergleiche Abbildung 4.4). Eine Ausnahme stellen durch Zusatzzeichen eingeschränkte Verkehrszeichen

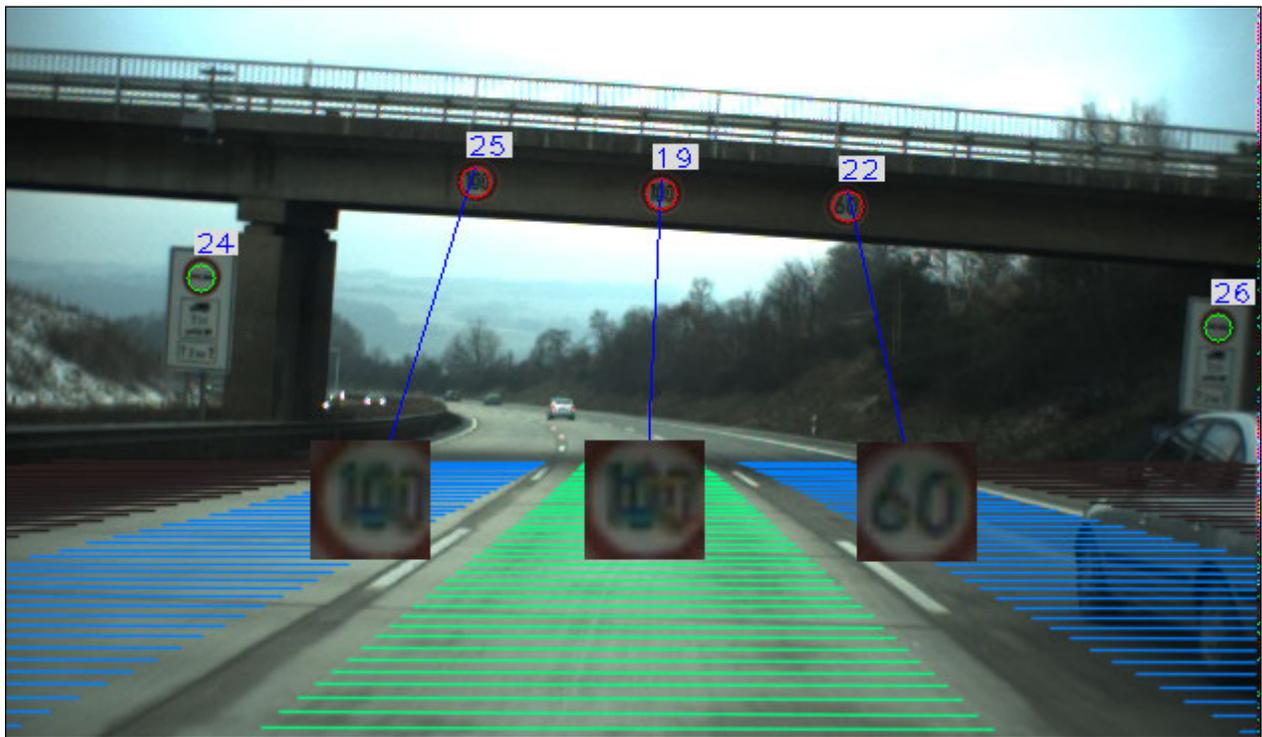


Abb. 4.4.: Zuordnung von Verkehrszeichen zu Fahrspuren gemäß Formel [4.8]. Die auf der Brücke angebrachten Tempolimits werden jeweils einer Fahrspur zugeordnet. Die neben den Fahrspuren aufgestellten Überholverbote werden dem Fahrbahnrand und damit allen Fahrspuren zugeordnet.

dar. Abschnitt 4.3.5 stellt einen Ansatz vor, der diesen Fall berücksichtigt, indem neben den räumlichen Relationen weitere Informationen ausgewertet werden.

4.2. Kontextsensitive Dempster/Shaffer-Fusion

Neben der bislang vornehmlich betrachteten Perzeption eignen sich digitale Karten in Verbindung mit Positionsdaten eines GPS-Sensors als Informationsquelle. Die exakte Kenntnis der eigenen Position und eine aktuelle, vollständige digitale Karte vorausgesetzt, kann zum Beispiel das gültige Tempolimit auf einfache Weise ermittelt werden. Aktuelle und vollständige Kartendaten gibt es in der Praxis aber nicht. In Deutschland sind beispielsweise nur etwa 75% der Tempolimits korrekt im Kartenmaterial verzeichnet [Jam+11, Table 1].

Am Beispiel der Bestimmung des gültigen Tempolimits wird in den folgenden Abschnitten aufgezeigt, wie mit Hilfe einer Fusion von Umfelderkennung und Kartendiensten zwei Informationsquellen zu einer neuen, robusteren kombiniert werden [NGZ09]. Das dem zugrunde liegende Fusionsverfahren muss dazu in der Lage sein, zwei Informationsquellen unter Berücksichtigung ihrer Zuverlässigkeit zu kombinieren. Außerdem muss es widersprüchliche Informationen erkennen und als Fehler melden. Die Fusion nach Dempster und Shafer [Sha02] stellt eine geeignete Basis für ein Fusionsverfahren dar, das diese Anforderungen erfüllt. Wie in den folgenden Abschnitten im Detail gezeigt wird, basiert die Fusion auf der Kombinationsregel von Dempster; der Konflikt-Indikator der Kombination wird zur Erkennung widersprüchlicher Informationen herangezogen. Die Zuverlässigkeit der beiden Informationsquellen



Abb. 4.5.: Kartendienste liefern in Baustellen und für Wechselverkehrszeichen unzuverlässige bzw. keine Informationen über das gültige Tempolimit. Die Perzeption weist bei Dunkelheit und schlechter Sicht eine höhere Fehlerwahrscheinlichkeit auf. Die Erkennung solcher Situationen gefolgt von der Adaption der Fusion verbessert die Ergebnisse.

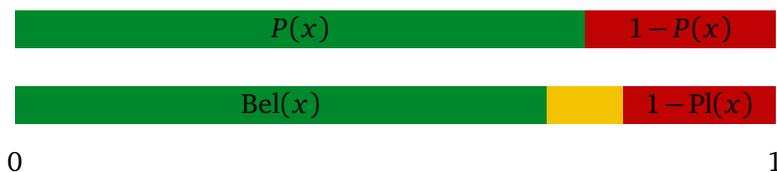


Abb. 4.6.: Gegenüberstellung von Wahrscheinlichkeitsmaß (oben) und Evidenz (unten). Evidenz verallgemeinert den Wahrscheinlichkeitsbegriff dahingehend, dass ein zusätzlicher, hier gelb dargestellter Unsicherheitsbereich eingeführt wird. Für $Pl(x) = Bel(x)$ sind Evidenzen mit dem klassischen Wahrscheinlichkeitsbegriff identisch.

wird nicht nur statisch in die Fusion eingebracht, sondern dynamisch an den Situationskontext adaptiert. Damit ist es möglich, auch dann gültige Fusionsergebnisse zu berechnen, wenn eine der Informationsquellen in einer für sie nachteiligen Situation (vergleiche Abbildung 4.5) fehlerhafte Informationen liefert.

4.2.1. Dempster/Shafer-Fusion

Die Evidenztheorie von Dempster und Shafer [Sha02] bildet die Grundlage für ein Verfahren der Informationsfusion zur Kombination verschiedener Quellen unter Berücksichtigung ihrer Glaubwürdigkeit. Das von ihr benutzte Maß *Evidenz* – nicht zu verwechseln mit denen im Kontext von Markov-Logik-Netze verwendeten Evidenzen – kann als zweidimensionale Erweiterung des eindimensionalen Maßes der Wahrscheinlichkeit aufgefasst werden (Abbildung 4.6). In der klassischen Wahrscheinlichkeitstheorie wird die angenommene Häufigkeit des Eintretens eines Ereignisses x durch das Wahrscheinlichkeitsmaß $P(x)$ angegeben. Die Evidenztheorie nutzt dagegen die zwei Maße $Bel(x)$ und $Pl(x)$ mit $Bel(x) \leq P(x) \leq Pl(x)$,

wobei $\text{Bel}(x)$ den Grad des Dafürhaltens (englisch *belief*) und $\text{Pl}(x)$ den Grad des Vertrauens (englisch *plausibility*) angibt. Ferner wird von einer Welt ausgegangen, die durch die diskreten, sich gegenseitig ausschließenden Zustände $\omega \in \Omega$ modelliert werden kann. Eine Informationsquelle wird durch eine Massefunktion m repräsentiert:

$$m : 2^\Omega \rightarrow [0, 1].$$

Die Massefunktion m muss die Bedingungen

$$\begin{aligned} m(\emptyset) &= 0 \\ \text{und } \sum_{X \in 2^\Omega} m(X) &= 1 \end{aligned} \quad [4.9]$$

erfüllen. Weiter ist der Grad des Dafürhaltens

$$\text{Bel}(X) := \sum_{Y \in X} m(Y) \quad [4.10]$$

die Summe der Massen aller Teilmengen von Zuständen sowie der Grad des Vertrauens

$$\text{Pl}(X) := \sum_{X \cap Y \neq \emptyset} m(Y) \quad [4.11]$$

die Summe der Massen aller sich nicht ausschließenden Zustände. Zwei über ihre Massefunktionen repräsentierte Informationsquellen m_1 und m_2 werden anhand der Kombinationsregel von Dempster zu einer neuen Massefunktion $m_{1 \otimes 2}$ kombiniert:

$$\begin{aligned} m_{1 \otimes 2}(\emptyset) &:= 0 \\ m_{1 \otimes 2}(X) &:= \frac{1}{1-k} \sum_{Y \cap Z = X} m_1(Y) m_2(Z) \quad \text{für } X \neq \emptyset \end{aligned} \quad [4.12]$$

mit

$$k := \sum_{Y \cap Z = \emptyset} m_1(Y) m_2(Z). \quad [4.13]$$

Der Wert k dient der Normalisierung, so dass $m_{1 \otimes 2}$ alle Anforderungen an eine Massefunktion erfüllt.

Wenn es mehrere mögliche Informationsquellen für eine diskrete Menge von zu erkennenden Zuständen gibt, kann man mittels Informationsfusion gemäß Formel [4.12] eine robustere Zustandsschätzung erzielen. Liefern die Informationsquellen sich widersprechende Informationen, spricht man von Konflikten. Die D/S-Fusion liefert über den Wert k (Formel [4.13]) hierfür einen Indikator, der die Summe der Evidenzen sich gegenseitig ausschließender Zustände enthält. Ein hoher Wert von k sollte zum Verwerfen des Fusionsergebnisses führen.

4.2.2. Bestimmung des gültigen Tempolimits

Sei $\Omega = \{5, 10, 20, 30, \dots, 130, \infty\}$ die Menge der möglichen Tempolimits, wobei ∞ stellvertretend für *kein Tempolimit* steht. Zur Bestimmung des aktuell gültigen Tempolimits sollen als Informationsquellen

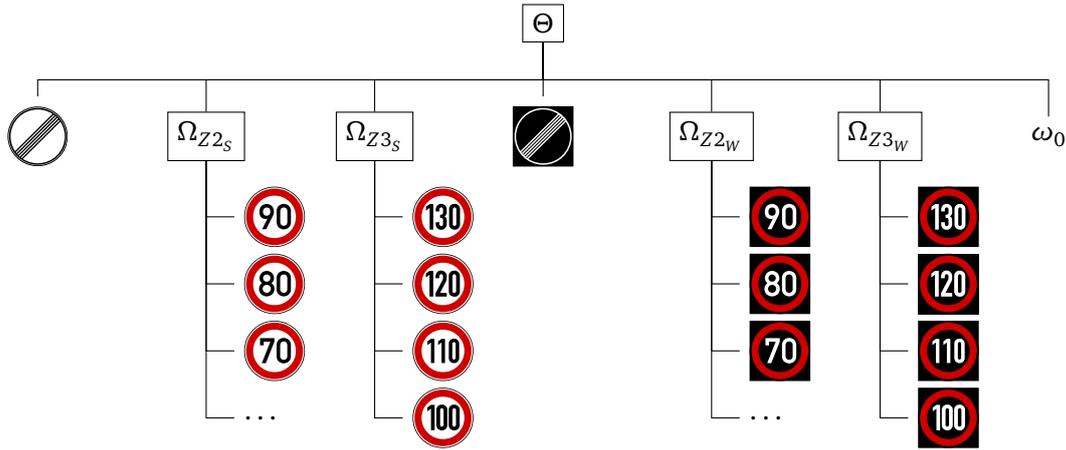


Abb. 4.7.: Struktur einer hierarchischen SVM zur Klassifikation von Tempolimits.

die Perzeption einerseits und Kartendienste andererseits verwendet werden. Kartendienste liefern das gültige Tempolimit, sofern die eigene Position bekannt ist und Kartendaten vorliegen. Die Perzeption erkennt explizite Tempolimits (Verkehrszeichen), nicht aber implizite, die aufgrund der Straßenart bzw. Umgebung gelten. Falls beiden Informationsquellen gleichzeitig Evidenz für gültige Tempolimits vorliegt, wird die im Folgenden beschriebene Fusion durchgeführt.

Wie in Abschnitt 3.3.6 vorgestellt eignet sich zur Klassifikation von Tempolimits eine hierarchische SVM; ihre Struktur sei die in Abbildung 4.7 dargestellte. Die Perzeption verwendet die in der Abbildung dargestellte Menge von Klassen $\Omega_{\text{Per}} = \{\omega_0, 5_S, 10_S, 20_S, \dots, 130_S, 5_W, 10_W, 20_W, \dots, 130_W, \infty_S, \infty_W\}$, die gegenüber Ω zusätzlich die Klasse ω_0 enthält sowie jeweils zwischen statischen Verkehrszeichen und Wechselverkehrszeichen unterscheidet. Sei $t : \Omega_{\text{Per}} \rightarrow \Omega$ eine surjektive, nicht injektive Funktion, die von der Perzeption erkannte Tempolimits aus Ω_{Per} auf sinngemäße Klassen in Ω abbildet. Zur Nutzung als Informationsquelle für die Fusion wird das Perzeptionsergebnis in eine Massefunktion m_p transformiert. Die SVM liefert für eine ROI Θ die Konfidenz $\hat{P}(\omega|f(\Theta))$ [3.49], die gemäß der korrespondierenden Klassen in Ω in die Massefunktion m_p übernommen wird:

$$m_p(X) := \begin{cases} 0, & X = \emptyset \\ \hat{P}(\omega_0|f(\Theta)), & X = \Omega \\ \sum_{t(\omega)=X} \hat{P}(\omega|f(\Theta)), & \text{sonst} \end{cases} \quad [4.14]$$

Wegen $m_p(\emptyset) = 0$ und $\sum_{X \in 2^\Omega} m_p(X) = \sum_{\omega \in 2^{\Omega_{\text{Per}}}} \hat{P}(\omega|f(\Theta)) \stackrel{P^{\text{Wkt.}}}{=} 1$ erfüllt m_p die Anforderungen an eine Massefunktion. Diese Definition von m_p bringt nicht nur Konfidenzen einzelner Klassen, sondern auch Konfidenzen für mehrere Klassen als Zwischenergebnisse der Hierarchiestufen ein. Tabelle 4.1 verdeutlicht das Vorgehen an einem Beispiel.

Zur Abbildung der Informationen des Kartendienstes in eine Massefunktion m_m soll neben explizit verzeichneten Tempolimits auch der Straßentyp eingehen, da beispielsweise auf einer Autobahn andere Geschwindigkeitsbereiche verwendet werden als in Innenstädten. Darüber hinaus kann die Ungenauig-

Klassifikation r	Massewerte
$\hat{P}(80_S f(\Theta)) = 0.91$	$m_p(80) = 0.91$
$\hat{P}(60_S f(\Theta)) = 0.07$	$m_p(60) = 0.07$
$\hat{P}(\Omega_{Z3} f(\Theta)) = 0.01$	$m_p(100, \dots, 130) = 0.01$
$\hat{P}(\omega_0 f(\Theta)) = 0.01$	$m_p(\Omega) = 0.01$

Tab. 4.1.: Beispielhafte Abbildung eines Klassifikationsergebnisses für eine ROI Θ auf die Massefunktion m_p .



Abb. 4.8.: Ungenauigkeiten der Positionsbestimmung können in die Massefunktion m_m einbezogen werden, indem Mehrdeutigkeiten der Karteneinpassung (hier durch A und B angedeutet) als unterschiedlich gewichtete Massewerte einbezogen werden. Illustration basierend auf [Ope12].

keit der Positionsbestimmung mit einbezogen werden (vergleiche Abbildung 4.8). Seien $R = \{r_1, \dots, r_k\}$ die von der Karteneinpassung gelieferten möglichen Positionen auf Straßen, wobei $r_i = (\psi_i, v_i, t_i)$ ein Tupel bestehend aus der Positionskonfidenz ψ_i , dem Tempolimit $v_i \in \Omega$ und dem Straßentyp $t_i \in T$ ist mit $\sum_{i=1}^k \psi_i = 1$ und $0 \leq \psi_i \leq 1 \forall i$. Sei $u : T \rightarrow 2^\Omega$ eine Abbildung von Straßentypen auf mögliche Tempolimits. Dann ist m_m mit

$$m_m(X) := \begin{cases} \kappa_A \psi_i, & X = \{v_i\} \\ 1 - \kappa_A \psi_i, & X = u(t_i) \\ 0, & \text{sonst} \end{cases} \quad [4.15]$$

eine Massefunktion mit $0 \leq \kappa_A \leq 1$ einem empirisch zu bestimmenden Parameter zur Modellierung der Aktualität der in den Kartendaten verzeichneten Tempolimits (vergleiche Abbildung 5.34). Aufgrund von $m_m(\emptyset) = 0$ und $\sum_{X \in 2^\Omega} m_m(X) = \sum_{i=1}^k (\kappa_A \psi_i + (1 - \kappa_A) \psi_i) = \sum_{i=1}^k \psi_i = 1$ erfüllt m_m die Anforderungen an eine Massefunktion. Tabelle 4.2 zeigt beispielhaft einen Auszug der Generierung der Massefunktion m_m für gegebene Tupel r .

Aus den beiden Massefunktionen m_p und m_m wird mit Hilfe der Kombinationsregel von Dempster

Tupel r	Massewerte
$r_1 = (0.95, 30, \text{residential})$	$m_m(30) = 0.7125$
	$m_m(\{5, 10, 20, \dots, 50\}) = 0.2375$
$r_2 = (0.05, 120, \text{motorway})$	$m_m(120) = 0.0375$
	$m_m(\{60, \dots, 130, \infty\}) = 0.0125$

Tab. 4.2.: Beispiel der Transformation von Tupeln r auf Massewerte m_m . Die Konfidenz für die Kartendaten wurde als $\kappa_A = 0.75$ festgelegt.

Quelle	Fehlerursache	Beschreibung
Kartendienst	Positionsungenauigkeit	Die Position des Fahrzeugs wird nicht auf die richtige Straße abgebildet
Kartendienst	Unvollständigkeit, Datenfehler	Daten sind in der Karte nicht eingetragen oder fehlerhaft
Perzeption	Schlechte Sicht	Ein Tempolimit wird nicht oder falsch erkannt, da die Sichtbedingungen ungünstig sind – typischerweise bei Nebel, Regen, Schnee, nachts, bei Gegenlicht oder in der Dämmerung
Perzeption	Fehlerkennung	Ein Tempolimit wird nicht oder falsch erkannt, da die Segmentierung oder Klassifikation fehlschlägt

Tab. 4.3.: Typische Fehlerursachen von Perzeption und Kartendaten.

(Formel [4.12]) das Fusionsergebnis $m_{p \otimes m}$ berechnet. Im Falle eines hohen Konfliktwerts – starken Widersprüchen zwischen beiden Informationsquellen – muss das Ergebnis allerdings abgelehnt werden. Da dieser Fall praktisch wichtig ist, wird er im Folgenden weiter untersucht.

4.2.3. Adaption an den Situationskontext

Konflikte zwischen den beiden Informationsquellen treten auf, wenn sich das von der Perzeption erkannte Tempolimit von dem in der Karte verzeichneten unterscheidet. Die Konfliktursachen können wie in Tabelle 4.3 dargestellt verschiedenen Kategorien zugewiesen werden: Fehler in der Positionsbestimmung, veraltete und unvollständige Kartendaten sowie ungünstige Sichtbedingungen.

Wenn es möglich ist, die in Tabelle 4.3 beschriebenen Situationen zu erkennen, kann die D/S-Fusion dahingehend erweitert werden: Die Zuverlässigkeit der einen oder anderen Informationsquelle wird in Abhängigkeit der vorliegenden Situation bestimmt und dieser Umstand in die jeweilige Massefunktion eingebracht. Dazu wird der Massewert $m(\Omega)$ als Repräsentation der Restunsicherheit angepasst:

$$m'(X) := \begin{cases} 0, & X = \emptyset \\ \kappa_S + m(\Omega), & X = \Omega \\ \frac{1 - \kappa_S - m(\Omega)}{1 - m(\Omega)} m(X), & \text{sonst} \end{cases} \quad [4.16]$$

Kontextelement	Wertebereich	Quelle
Wechselverkehrszeichen	ja, nein	Kartendaten, Perzeption
Umgebungslicht	Gegenlicht, Normal, Diffus, Schwach	Perzeption
Baustelle	ja, nein	Perzeption
Wetter	Trocken, Nebel, Regen	Perzeption

Tab. 4.4.: Modellierung des Situationskontexts S zur Adaption der Massefunktionen.

mit einem vom Situationskontext S (Tabelle 4.4) abhängigen, empirisch zu bestimmenden Wert κ_S und $0 \leq \kappa_S + m(\Omega) \leq 1$ (vergleiche Abbildung 5.34).

Beweis. m' ist eine Massefunktion, wenn m eine Massefunktion ist und $0 \leq \kappa_S + m(\Omega) \leq 1$ gilt, denn es gilt $m'(\emptyset) = 0$ und

$$\begin{aligned}
\sum_{X \in 2^\Omega} m'(X) &= m'(\emptyset) + m'(\Omega) + \sum_{X \in 2^\Omega \setminus (\Omega \cup \emptyset)} m'(X) \\
&= 0 + \kappa_S + m(\Omega) + \sum_{X \in 2^\Omega \setminus (\Omega \cup \emptyset)} \frac{1 - \kappa_S - m(\Omega)}{1 - m(\Omega)} m(X) \\
&= \kappa_S + m(\Omega) + \frac{1 - \kappa_S - m(\Omega)}{1 - m(\Omega)} \left(\left(\sum_{X \in 2^\Omega} m(X) \right) - m(\Omega) - m(\emptyset) \right) \\
&= \kappa_S + m(\Omega) + \frac{1 - \kappa_S - m(\Omega)}{1 - m(\Omega)} (1 - m(\Omega)) \\
&= 1.
\end{aligned}$$

□

Die adaptive Fusion verwendet nach Gleichung [4.16] berechnete Massefunktionen m'_p und m'_m um $m'_{p \otimes m}$ zu berechnen. Der dabei verwendete Wert κ_S ist abhängig vom Situationskontext S , der folglich erkannt und auf Werte für κ_S abgebildet werden muss (vergleiche Abbildung 5.34). Dazu wird im Folgenden die Erkennung der einzelnen in Tabelle 4.4 aufgeführten Kontextelemente behandelt.

Wechselverkehrszeichen Verkehrsbeeinflussungsanlagen setzen unter anderem Wechselverkehrszeichen ein, um das Tempolimit dynamisch anzupassen. Dafür kommen neben mechanischen Wechselverkehrszeichen mit Prismentechnik vermehrt dynamische Anzeigen mit LED-Technik zum Einsatz (Abbildung 4.9). Verkehrsbeeinflussungsanlagen sind in Kartendaten verzeichnet und diese Information kann für den Situationskontext extrahiert werden. Wechselverkehrszeichen mit LED-Technik unterscheiden sich im Aussehen deutlich von statischen Tempolimits. Sie werden von der Perzeption als eigene Klassen behandelt und sind daher einfach von statischen Tempolimits zu unterscheiden (vergleiche Abbildung 4.7).

Umgebungslicht Videokameras (eingeschlossen HDR-Kameras) besitzen einen begrenzten Dynamikbereich, der das Helligkeitsspektrum der Umgebung nicht vollständig abdecken kann. Insbesondere



Abb. 4.9.: Mechanisches Wechselverkehrszeichen in Prismentechnik (links) [Wik12] und Wechselverkehrszeichen mit LED-Technik.

Verschlusszeit	Verstärkungsfaktor	Umgebungslicht
$s = s_{\min}$		Gegenlicht
$s_{\min} < s \leq s_{\max}$	$g = g_{\min}$	Normal
$s = s_{\max}$	$g_{\min} < g < g_{\max}$	Diffus
	$g = g_{\max}$	Schwach

Tab. 4.5.: Aufgrund der Parameter der Belichtungsregelung der Kamera kann auf das Umgebungslicht geschlossen werden.

kommt es bei direkter Sonneneinstrahlung zu Überbelichtung. Fehlt die Sonne als Lichtquelle und ist kein ausreichendes Kunstlicht in der Umgebung vorhanden, entstehen unterbelichtete Bilder. Typischerweise ist dies während der Dämmerung, nachts und in Tunneln der Fall. Sowohl Unter- als auch Überbelichtung beeinflussen die Perzeption negativ, da der Kontrast von Verkehrselementen (zum Hintergrund und zu sich selbst) klein ist. Dadurch schlagen Farbklassifikation und Kantenextraktion häufiger fehl als bei idealem Umgebungslicht. Eine verwandte Fehlerquelle ist die Entstehung von Bewegungsunschärfe bei längeren Belichtungszeiten und hoher Eigengeschwindigkeit, die sich ebenfalls negativ auf Farbklassifikation und Kantenextraktion, aber auch negativ auf die Klassifikation auswirkt. Eine geeignete Heuristik zur Erkennung dieser Situationen ist die Auswertung der Parameter der Belichtungsregelung der Kamera. Tabelle 4.5 zeigt Regeln, die anhand der Verschlusszeit s der Kamera mit $s_{\min} \leq s \leq s_{\max}$ und dem Verstärkungsfaktor (englisch *gain*) g mit $g_{\min} \leq g \leq g_{\max}$ auf das Umgebungslicht schließen.

Baustellen Aufgrund ihrer begrenzten Dauer werden Baustellen und damit auch mit ihnen verbundene Änderungen der zulässigen Höchstgeschwindigkeit meist nicht in digitale Karten übernommen. Baustellen werden durch dreieckige Warnhinweise, Warnbaken und gelbe Fahrspurmarkierungen gekennzeichnet, die von der Perzeption erkannt werden können. Durch die Erkennung dieser Verkehrselemente kann dann auf das Vorhandensein einer Baustelle geschlossen werden. Ein geeigneter Ansatz hierfür wird in Abschnitt 4.3.4 vorgestellt.

Wetter Feine Wassertröpfchen in Bodennähe behindern in Form von Nebel die Sicht; sie schwächen Lichtstrahlen ab und vermindern dadurch den wahrgenommenen Kontrast von Objekten. Größere

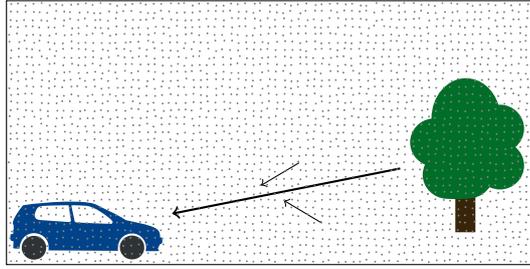


Abb. 4.10.: Das Koschmieder-Modell beschreibt den Effekt von Wassertröpfchen in der Luft auf die Farbwahrnehmung von Objekten durch zwei Einflussfaktoren: Überlagerung und Abschwächung (siehe Text).

Wassertröpfchen führen als Regen zu lokalen Störungen der Aufnahme von Objekten. Beide Wetterarten erhöhen damit die Fehlerwahrscheinlichkeit der Perzeption. Das Koschmieder-Modell hat sich als physikalisches Modell zur Beschreibung der durch Wassertröpfchen beeinträchtigte Sicht für den Beobachter etabliert [RM08]. Mit seiner Hilfe lässt sich ein Zusammenhang herstellen zwischen dem Farbwert $r(\mathbf{x})$ eines Bildpunkts und physikalischen Größen:

$$r(\mathbf{x}) = \underbrace{I_\infty \rho e^{-\beta(\lambda)d}}_{=: r_A(\mathbf{x})} + \underbrace{I_\infty (1 - e^{-\beta(\lambda)d})}_{=: r_U(\mathbf{x})} \quad [4.17]$$

mit der Umgebungsfarbe I_∞ , der normalisierten Abstrahlung des Objekts ρ , dem Streukoeffizienten $\beta(\lambda)$ und der Distanz des Objekts d . Laut Gleichung [4.17] setzt sich der wahrgenommene Farbwert aus zwei Teilen zusammen: Die *Abschwächung* $r_A(\mathbf{x})$ beschreibt, wie der wahrgenommene Farbwert des Objekts von Wassertröpfchen (modelliert durch den Streukoeffizienten) in der Luft abgeschwächt wird. Mit steigender Distanz verstärkt sich dieser Effekt; im Extremfall bei starkem Nebel und großer Distanz kann der Farbwert eines Objekts von einer Kamera nicht mehr wahrgenommen werden. Die *Überlagerung* $r_U(\mathbf{x})$ modelliert die Streuung des Umgebungslichts durch Wassertröpfchen; dieser Teil beschreibt die Wahrnehmung von Nebel als "graue Wand". Abbildung 4.10 illustriert die beiden Einflussfaktoren.

Vom Streukoeffizienten $\beta(\lambda)$ kann direkt auf das vorliegende Wetter geschlossen werden. Er ist jedoch ebenso wie ρ und d unbekannt und nicht direkt bestimmbar. Daher wird eine indirekte Bestimmung vorgenommen, indem Merkmale eingeführt werden, die den Einfluss von $\beta(\lambda)$ auf Farbwerte messen. Dafür werden wie von Roser und Moosmann vorgeschlagen die Merkmale minimale Intensität, Kontrast, Schärfe, Farbwert und Sättigung extrahiert. Um Regionen mit ähnlichen Entfernungswerten zusammenzufassen und Störungen durch andere Objekte zu vermeiden, werden die Merkmale gemäß der blockweisen Histogramm-Transformation (Abschnitt 3.3.3) extrahiert. Zusätzlich wird ein Histogramm für das Gesamtbild extrahiert. Für die Klassifikation wird eine SVM verwendet, die die Klassen *Trocken*, *Nebel* und *Regen* unterscheidet [WNZ08].

Zur kontextsensitiven Erkennung von Tempolimits wird das gültige Tempolimit $\hat{\omega} \in \Omega \cup \omega_0$ bestimmt als

$$\hat{\omega} := \begin{cases} \omega_0, & k > \kappa_k \\ \omega_0, & \max_{\omega \in \Omega} Bel'_{p\otimes m}(\omega) < \kappa_{Bel} \\ \omega_0, & \max_{\omega \in \Omega} Pl'_{p\otimes m}(\omega) < \kappa_{Pl} \\ \arg \max_{\omega \in \Omega} Pl'_{p\otimes m}(\omega), & \text{sonst} \end{cases} \quad [4.18]$$

wobei die Parameter κ_k den maximal akzeptierten Konfliktwert k , κ_{Bel} den minimalen Grad des Dahaltens, κ_{Pl} den minimalen Grad des Vertrauens und ω_0 den Fehlerzustand angeben. Bei hohem Konflikt bzw. geringer Sicherheit wird das Ergebnis folglich abgelehnt. In die Berechnung gehen die in den vorhergehenden Abschnitten eingeführten Massefunktionen m'_p und m'_m sowie der Situationskontext S ein.

Bei der praktischen Umsetzung ist zu beachten, dass häufig räumliche Differenzen zwischen der Position von Verkehrszeichen und der in den Kartendaten verzeichneten Höchstgeschwindigkeit für Streckensegmente auftreten. Dies muss in der Implementierung durch entsprechende zeitliche bzw. räumliche Toleranzen in der Fusion berücksichtigt werden. Des Weiteren besitzen Verkehrszeichen eine Gültigkeit, die über ihre Sichtbarkeit hinausgeht. Während der Fahrt muss ständig überprüft werden, ob aufgrund von Einmündungen in andere Straßen die Gültigkeit des zuletzt beobachteten Verkehrszeichens außer Kraft gesetzt wird. Ansonsten ist es für die Fusion als relevant anzusehen, auch wenn es nicht mehr im Videobild sichtbar ist. Schließlich empfiehlt sich die Einführung einer zeitlichen und räumlichen maximalen Gültigkeit von Verkehrszeichen. Zur Berechnung der Fusion betrachten viele der in diesem Abschnitt eingeführten Formeln die Potenzmenge 2^Ω . In der Implementierung kann man ausnutzen, dass fast alle Elemente $m_p(X)$ und $m_m(X)$ für $X \in 2^\Omega$ den Wert 0 annehmen. Anstatt über alle $2^{|\Omega|} = 32768$ Elemente der Potenzmenge zu iterieren, genügt die Betrachtung der Untermengen mit Werten größer 0. Die Laufzeit der Fusion ist dann im Vergleich zur Perzeption zu vernachlässigen.

4.3. Markov-Logik-Netze (MLNs)

Zur Interpretation einer Situation analysieren kognitive Assistenzsysteme die Eigenschaften und Beziehungen der vorhandenen Objekte. In Verbindung mit Hintergrundwissen, zum Beispiel in Form von Straßenverkehrsregeln, können daraus zusätzliche Informationen abgeleitet werden. Schlussfolgerungen müssen berücksichtigen, dass viele Informationsquellen verlässliche Informationen liefern. Eine probabilistische Betrachtung der Welt kann Unsicherheiten modellieren und in Schlussfolgerungen einbeziehen. Die manuelle Modellierung einer Welt durch ein bayessches Netz oder Markov-Netz ist aufwendig und fehleranfällig. Wünschenswert ist stattdessen eine automatische Generierung. Sie soll unsicheres Wissen aus der Perzeption – beispielsweise die Erkennungsunsicherheit bezüglich des Typs eines Verkehrszeichens – einbringen genauso wie sicheres Wissen über Situationen – etwa die Tatsache, dass sich ein Zusatzzeichen auf das über es montierte Verkehrszeichen bezieht. Markov-Logik-Netze (MLNs) [RD06] stellen hierfür ein methodisches Framework dar. Eine Wissensbasis bestehend aus prädikatenlogischen Formeln bildet das Situationswissen ab. Unsicherheit wird in Form von Wahrscheinlichkeiten repräsentiert, die über an die Formeln angehängte Gewichte eingebracht werden. Die Wissensbasis dient als Vorlage, um in Kombination mit erfassten Informationen zur modellierten Welt automatisch ein Markov-Netz zu generieren, auf das Inferenzverfahren angewendet werden können. Da die Theorie der Markov-Logik-Netze noch vergleichsweise jung ist, wird im Folgenden zunächst angelehnt an Richardson und Domingos [RD06] eine kurze allgemeine Einführung gegeben und danach ihre Anwendung im Rahmen dieser Arbeit aufgezeigt.

Name	Beschreibung	Beispiele
Konstanten		M, L
Variablen		x, y
Funktion	Abbildung von Objektupeln auf Objekte	DrivesOn
Prädikat	Repräsentation von Eigenschaften von Objekten und Relationen zwischen ihnen	LeftOf, Synchronized
Term	Ausdruck, der ein Objekt repräsentiert	L, x , DrivesOn(y)
Atomare Formel (Atom)	Prädikat mit Objektupel	LeftOf(x , Lane(y))
Grundterm	Variablenfreier Term	DrivesOn(L1)
Grundatom	Variablenfreie atomare Formel	LeftOf(M, Lane(L1))
Positives Literal	atomare Formel	
Negatives Literal	negierte atomare Formel	
Klausel	Disjunktion von Literalen	
Logische Symbole		NOT, AND, OR, =>, <=>
Quantoren		EXISTS, FORALL
Formel	Verbindung von atomaren Formeln und Formeln mit logischen Symbolen	LeftOf(M, L3) AND RightOf(M, L1)

Tab. 4.6.: Elemente von prädikatenlogischen Formeln in MLNs.

Definition 4.1. Ein Markov-Logik-Netz M ist eine Menge von geordneten Paaren (ω, F) bestehend aus einem Gewicht $\omega \in \mathbb{R}$ und einer prädikatenlogischen Formel F zusammengesetzt aus den in Tabelle 4.6 angegebenen Elementen:

$$M = \{(\omega_i, F_i)\} \quad i \in \mathbb{N}.$$



Ein Markov-Logik-Netz ist eine Wissensbasis bestehend aus gewichteten prädikatenlogischen Formeln. Ähnlich der aus Ontologien bekannten *A-Box* enthält es allgemeines Wissen unabhängig von einer konkreten Situation. Im Spezialfall $M = \{(\infty, F_i)\} \forall i$ – alle Gewichte nehmen den Wert unendlich an – ist ein MLN identisch mit einer deterministischen prädikatenlogischen Wissensbasis.

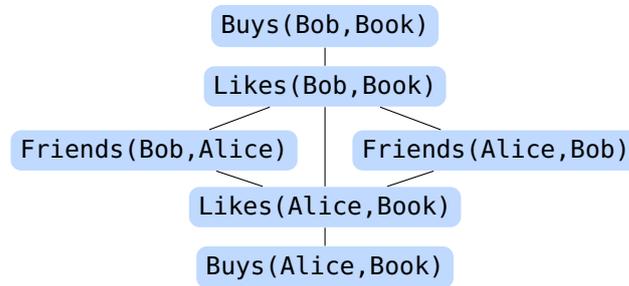
Beispiel 4.1. Die Aussagen “Die Vorliebe für ein Produkt verleitet Personen dazu, es zu kaufen” und “Freunde haben ähnliche Interessen” können in einem MLN wie folgt dargestellt werden:

- 1 0.23 Likes(**x, y**) => Buys(**x, y**)
- 2 0.42 Friends(**x1, x2**) **AND** Likes(**x1, y**) => Likes(**x2, y**).

Hierbei sind **x, x1, x2** und **y** typisierte Variablen und Likes, Buys sowie Friends sind Prädikate. Es wird die Notation verwendet, als Anfangsbuchstaben von Variablen Kleinbuchstaben und als Anfangsbuchstaben von Konstanten Großbuchstaben zu benutzen. Freie Variablen sind implizit durch Allquantoren gebunden und alle Formeln implizit durch Konjunktionen verbunden. Ein MLN kann also immer als eine Formel aufgefasst werden.

Ein Markov-Netz (MN) $N_{M,C}$ ist ein ungerichteter Graph zur Repräsentation stochastischer Abhängigkeiten von durch Knoten dargestellten Zufallsvariablen. Er kann bei gegebenen Konstanten $C = \{C_1, \dots, C_n\}$ aus einem MLN M konstruiert werden. Dazu werden die Konstanten gemäß ihres Typs in die atomaren Formeln eingesetzt und die Menge der Grundatome gebildet. Jedes Grundatom wird durch einen Knoten in $N_{M,C}$ repräsentiert. Zwei Knoten werden genau dann durch eine Kante verbunden, wenn sie in mindestens einer Formel in M zusammen auftreten.

Beispiel 4.2. Es sei das MLN aus Beispiel 4.1 gegeben sowie die typisierten Konstanten Alice, Bob und Book. Daraus werden alle gemäß der Typisierung möglichen Grundatome gebildet: Likes(Alice,Book), Likes(Bob,Book), Buys(Alice,Book), Buys(Bob,Book), Friends(Alice,Bob) sowie Friends(Bob,Alice). Sie bilden die Knoten des Markov-Netzes. Für alle Paare von Grundatomen, die in einer Formel zusammen auftreten, wird eine Kante erzeugt. Damit ergibt sich das Markov-Netz



Die aus einer Formel gebildeten Grundformeln bilden im Markov-Netz Cliques (Teilmengen von Knoten, bei denen jedes Knotenpaar verbunden ist), sind also benachbart und schnell zugreifbar. Ein MN ist eine kompakte Repräsentation der multivariaten Wahrscheinlichkeitsverteilung der zugrunde liegenden Welt gemäß

$$P(X = x) = \frac{1}{Z} \exp \left(\sum_{i=1}^n \omega_i n_i(x) \right), \quad [4.19]$$

wobei ω_i das Gewicht der i -ten Formel ist und $\frac{1}{Z}$ mit

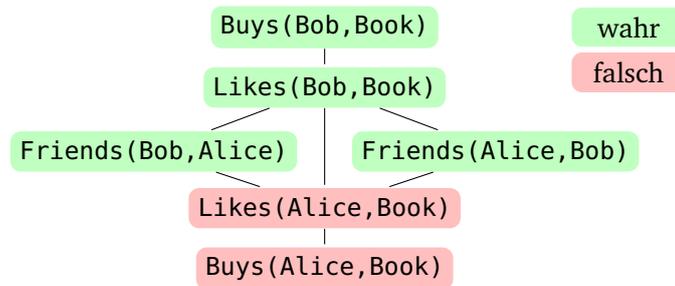
$$Z = \sum_{x \in X} \exp \left(\sum_{i=1}^n \omega_i n_i(x) \right) \quad [4.20]$$

eine Normalisierung der Gesamtwahrscheinlichkeit auf den Wert 1 bewirkt und

$$n_i(x) := \sum_{G_j \in F_{i,c}} \delta(G_j(x)) \quad [4.21]$$

die Anzahl der wahren Grundformeln zur Formel F_i ist.

Beispiel 4.3. Der Weltzustand des MNs des obigen Beispiels sei wie folgt gegeben:



Die Grundformeln sind

i	ω_i	G_j	$\delta(G_j(x))$	$n_i(x)$
1	0.23	Likes(Bob, Book) \Rightarrow Buys(Bob, Book)	1	2
		Likes(Alice, Book) \Rightarrow Buys(Alice, Book)	1	
2	0.42	Friends(Bob, Alice) AND Likes(Bob, Book) \Rightarrow Likes(Alice, Book)	0	1
		Friends(Alice, Bob) AND Likes(Alice, Book) \Rightarrow Likes(Bob, Book)	1	

und mit Formel [4.19] berechnet man seine Wahrscheinlichkeit als $P(X = x) = Z^{-1} \exp(2 \cdot 0.23 + 1 \cdot 0.42)$.

4.3.1. Inferenz

In praktischen Problemstellungen ist nicht der komplette Weltzustand bekannt, sondern nur als Evidenz bezeichnete Teile davon (nicht zu verwechseln mit dem Evidenz-Begriff der Dempster/Shافر-Theorie). Mittels Inferenz werden für ein gegebenes MLN Wahrscheinlichkeitsverteilungen für nicht beobachtbare Zustandsvariablen geschätzt. Wichtige Arten der Inferenz sind die Schätzung des wahrscheinlichsten Zustands der Welt (MAP/MPE-Schätzung) und die Schätzung der Randverteilung von Zufallsvariablen. Letztere Inferenz-Art wird im Rahmen dieser Arbeit benötigt und im Folgenden kurz als Inferenz bezeichnet.

Das anfänglich zur Inferenz in MLNs verwendete Gibbs-Sampling [GG84] ist nicht anwendbar, wenn harte Einschränkungen (Formeln mit unendlichem Gewicht) benutzt werden. Als Alternative wurde der Algorithmus MC-SAT als Kombination eines MCMC-Verfahrens und eines SAT-Solvers entwickelt [PD06]. Er ähnelt in seiner Struktur der MCMC-Vorgehensweise, eine vorgegebene Anzahl n zufällig generierter Zustände aufzustellen und die Randverteilung $P(Y = y | X = x)$ als Anteil der Zustände abzuschätzen, in denen Y erfüllt ist.

Der in Algorithmus 4.1 dargestellte MC-SAT Algorithmus wählt zunächst in Zeile 1 zufällig einen initialen Zustand $x^{(0)}$, in dem alle Formeln mit unendlichem Gewicht erfüllt sind. Gibt es keinen solchen Zustand, ist die Wissensbasis widersprüchlich. Anschließend werden n weitere Zustände jeweils ausgehend vom vorhergehenden Zustand bestimmt. Dazu wird eine Teilmenge von Formeln M bestimmt, die im nächsten Zustand erfüllt sein müssen. Basierend auf den im vorhergehenden Zustand $x^{(i-1)}$ erfüllten Formeln (Zeile 4) wird eine Teilmenge von ihnen zufällig ausgewählt. Die zufällige Auswahl verwendet

Algorithmus 4.1 MC-SAT(clauses, weights, n)

Quelle: Poon und Domingos [PD06]

-
- 1: $x^{(0)} \leftarrow \text{Satisfy}(\text{hard clauses})$
 - 2: **for** $i \leftarrow 1$ to n **do**
 - 3: $M \leftarrow \emptyset$
 - 4: **for all** $c_k \in \text{clauses satisfied by } x^{(i-1)}$ **do**
 - 5: With probability $1 - e^{-w_k}$ add c_k to M
 - 6: Sample $x^{(i)} \sim \mathcal{Q}_{\text{SAT}(M)}$
-

Slice Sampling, um Formeln mit einer Wahrscheinlichkeit proportional zu ihrem Gewicht auszuwählen (Zeile 5; Formeln mit unendlichem Gewicht werden so immer ausgewählt). Schließlich wird SampleSAT [WES04] (eine Kombination aus WalkSAT und Simulated annealing) benutzt, um zufällig einen neuen Zustand $x^{(i)}$ zu bestimmen, der M erfüllt.

4.3.2. Lernen

Parameterlernen für MLNs ist – bei gegebener Wissensbasis und gegebener Evidenz x (Trainingsdaten) – die Optimierung der Gewichte der Wissensbasis. Falls die für Inferenz-Anfragen benutzten Atome Y vorab bekannt sind, kann diskriminatives Lernen eingesetzt werden [SD05]. Die bedingte Wahrscheinlichkeit von Y gegeben X ist

$$P(Y = y|X = x) = \frac{1}{Z_x} \exp \left(\sum_{i \in F_Y} \omega_i n_i(x, y) \right) \quad [4.22]$$

mit F_Y der Menge der Formeln, die ein Atom aus Y enthalten. Die Maximierung dieser Wahrscheinlichkeit durch Anpassung der Gewichte $\mathbf{w} = (\omega_1, \dots, \omega_n)^\top$ ist ein konvexes Optimierungsproblem mit Richtungsableitung

$$\frac{\partial}{\partial \omega_i} \log P_{\mathbf{w}}(Y = y|X = x) = n_i(x, y) - E_{\mathbf{w}}(n_i(x, y)). \quad [4.23]$$

Es kann durch Gradientenabstiegsverfahren gelöst werden, indem die Gewichte iterativ in Richtung des Gradienten \mathbf{g} angepasst werden gemäß

$$\mathbf{w}^{[t+1]} := \mathbf{w}^{[t]} + \eta \mathbf{g} \quad [4.24]$$

mit einer Schrittweite η . Für stark unterschiedliche optimale Gewichte konvergiert dieses Verfahren nur sehr langsam. Des Weiteren ist $E_{\mathbf{w}}(n_i(x, y))$ nicht effizient exakt bestimmbar und muss approximiert werden. Lowd und Domingos [LD07] stellen effiziente Algorithmen vor, die beide Probleme lösen.

Oft werden in MLNs mehrere gleichartige Formeln verwendet, die sich nur durch ihre Gewichte und die Werte einzelner Konstanten unterscheiden, beispielsweise

- | |
|--|
| <ol style="list-style-type: none"> 1 $w1 \ x \text{ Likes}(x, \text{"little"}) \Rightarrow \text{Buys}(x)$ 2 $w2 \ x \text{ Likes}(x, \text{"much"}) \Rightarrow \text{Buys}(x)$ |
|--|

mit $c=\{\text{"little"}, \text{"much"}\}$. Im Zusammenhang mit Lernverfahren wird dafür eine Kurzschreibweise eingeführt: Ein einer Variable vorangestelltes $+$ führt beim Lernen von Gewichten zum Ausrollen in mehrere Formeln mit unterschiedlichen Gewichten gemäß des Wertebereichs der Variablen. Die Kurzschreibweise des obigen Beispiels ist

```
1 Likes(x,+c) => Buys(x).
```

Neben Parameterlernen kann Strukturlernen eingesetzt werden, bei dem nicht nur die Gewichte der Formeln, sondern auch die Formeln selbst gelernt werden. Es wird im Rahmen dieser Arbeit aber nicht weiter betrachtet.

4.3.3. Wissensrepräsentation

Markov-Logik-Netze (MLNs) stellen ein flexibles Framework für die Situationsinterpretation dar: Wissen über die Verkehrsdomäne und problemspezifische Regeln können als prädikatenlogische Formeln modelliert werden. Zusammen mit konkreten Informationen zu einer Verkehrssituation (beispielsweise von der Umfelderkennung erkannte Verkehrselemente) werden automatisch Markov-Netze (MNs) erzeugt, die eine kompakte Repräsentation der Wahrscheinlichkeitsverteilung darstellen. Auf sie können Inferenzverfahren angewendet werden, um neues Wissen abzuleiten. In den folgenden Abschnitten wird die Übertragung von MLNs auf die Situationsinterpretation entwickelt.

In Abbildung 4.11 ist ein Auszug aus der Taxonomie relevanter Objekte der Verkehrsdomäne dargestellt. Instanzen der von ihr modellierten Objekte werden inklusive Metainformationen wie der Abschätzung der Erkennungsunsicherheit von der Umfelderkennung erzeugt. Zur Nutzung des Wissens in einem MLN muss es geeignet transformiert werden. Ist Ch1 in der Taxonomie ein Kind von Obj1, werden folgende Formeln gebildet:

```
1 Ch1(x) => Obj1(x)
2 w Obj1(x) => Ch1(x).
```

Sind sowohl Ch1 als auch Ch2 Kinder von Obj1, wird eine zusätzliche Regel hinzugefügt, die ihre Eindeutigkeit sicherstellt:

```
1 NOT(Ch1(x) AND Ch2(x)).
```

Bei der Perzeption von Objekten treten Erkennungsfehler auf. Sie werden mittels Gewichten modelliert gemäß

```
1 PerObj(x,+c) => Obj(x)
```

wobei c die Diskretisierung der Konfidenz (Wahrscheinlichkeitsverteilung) des Klassifikationsergebnisses darstellt. Für den Diskretisierungsbereich $c = \{\text{"low"}, \text{"medium"}, \text{"high"}\}$ etwa ergibt das Ausrollen von $+c$ folgende Formeln:

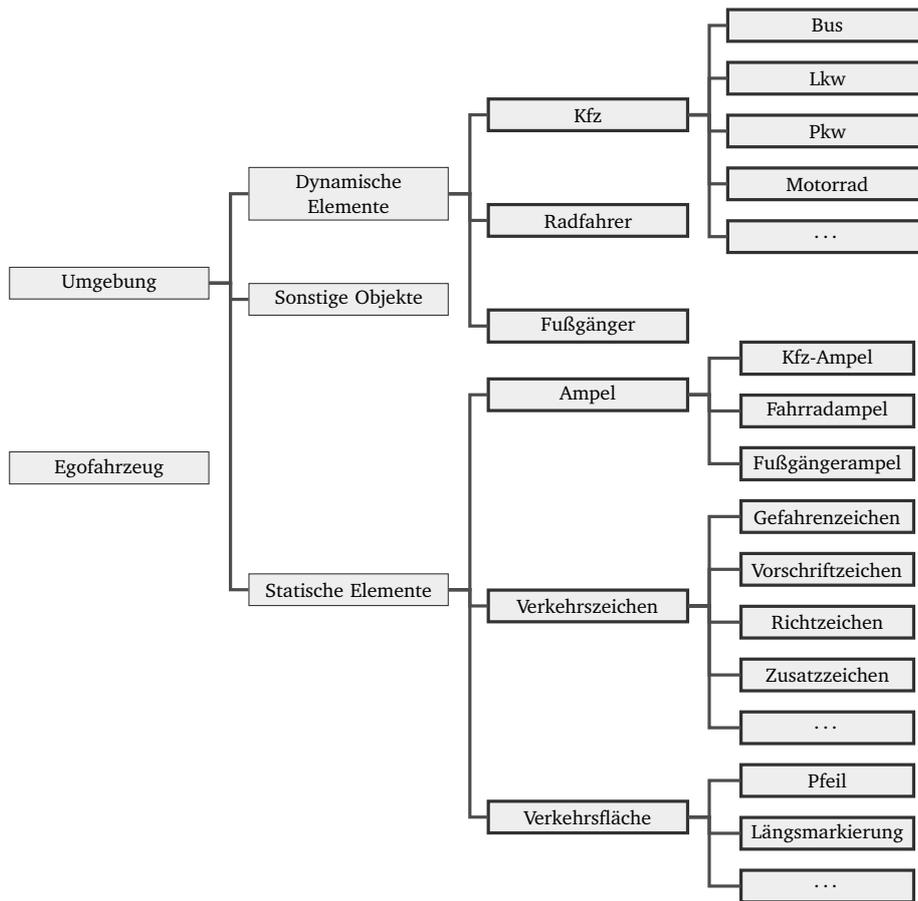


Abb. 4.11.: Auszug aus der Taxonomie von Objekten der Verkehrsdomäne.

- 2 w1 PerObj(x, "low") => Obj(x)
- 3 w2 PerObj(x, "medium") => Obj(x)
- 4 w3 PerObj(x, "high") => Obj(x).

Objekteigenschaften, die innerhalb von MLNs genutzt werden sollen, werden über Prädikate modelliert. Ihr Wertebereich wird gegebenenfalls diskretisiert, zum Beispiel durch

- 1 Height(x, +h).

Dieser generische Teil der Wissensrepräsentation kann auf alle bekannten Objekte der Verkehrsdomäne angewendet werden, um ein Basis-MLN zu bilden, das um problemspezifische Regeln erweitert wird.

4.3.4. Erkennung von Baustellen

Baustellen stellen durch die mit ihnen einhergehenden kurzfristigen Änderungen von Verkehrsregeln und Straßengeometrie eine Herausforderung für Assistenzsysteme dar (vergleiche Abschnitt 4.2.2). Verkehrsteilnehmer werden durch dreieckige Warnhinweise, rechteckige Warnbaken und gelbe Spurmarkierungen auf Baustellen aufmerksam gemacht. Meistens werden jedoch nicht alle Signale gleichzeitig

verwendet. Zur Erkennung der verschiedenen Elemente können die in den vorherigen Abschnitten vorgestellten Methoden und Verfahren wie folgt verwendet werden:

Element	Methoden	Details
Warnhinweis <i>Baustelle</i>	Erkennung von Warnhinweisen	Abschnitt 3.5.2
Warnbake	Viola/Jones-Verfahren	Abschnitt 3.2.6
Markierungsfarbe	Viola/Jones-Verfahren	Abschnitt 3.2.6
	Blockweise Histogramm-Transformation	Abschnitt 3.3.3
	SVM	Abschnitt 3.3.6

Aufgrund der Erkennung bzw. der Nicht-Erkennung der verschiedenen Merkmale kann mit Hilfe eines MLNs geschlossen werden, ob sich eine Baustelle in der näheren Umgebung befindet. Die genaue Anzahl der gefundenen Merkmale spielt dabei keine entscheidende Rolle. Zur Reduktion der Größe des Markov-Netzes wird daher zunächst eine Diskretisierung der Anzahl der gefundenen Elemente auf die Menge $\{none, some, many, lots\}$ durchgeführt. Darüber hinaus ist es sinnvoll, die zeitliche Historie mit einzubringen. Dies wird realisiert, indem neben Evidenzen für das aktuelle Bild auch solche für die kurzfristige Vergangenheit betrachtet werden. Die Evidenzen im MLN-Modell benutzen dafür den Typen *time* mit den Werten *now* stellvertretend für das aktuelle Bild, *just* für Beobachtungen der vergangenen sechs und *recently* für die letzten 18 Bilder:

```

1 // Types
2 RoadWorks(scene)
3 time = {"now", "just", "recently"}
4 color = {"white", "yellow"}
5 amount = {"none", "some", "many", "lots"}
6
7 // Evidence
8 Beacons(scene, time, amount)
9 RoadWorkSign(scene, time, amount)
10 Marking(scene, color, time, amount)
11
12 // Rule skeletons
13 RoadWorks(x)
14 Beacons(x, +t, +a) => RoadWorks(x)
15 RoadWorkSign(x, +t, +a) => RoadWorks(x)
16 Marking(x, +c, +t, +a) => RoadWorks(x)

```

Werden etwa wie in Abbildung 4.12 dargestellt im aktuellen Bild 133 viele Warnbaken erkannt, führt dies zur Evidenz `Beacons(133, "now", "many")`. Die Regeln wiederum verknüpfen die einzelnen Beobachtungen mit dem zu inferierenden Prädikat `RoadWorks`. Formeln, die Variablen mit vorangestelltem `+`



Abb. 4.12.: Typische Baustellenszene mit eingezeichneten Detektionsergebnissen.

enthalten, werden wie oben erwähnt ausgerollt, um Gewichte für alle Werte des zugehörigen Typs zu lernen. Die Notation Beacons (x, +t, +a) => RoadWorks (x) etwa ergibt ausgerollt 12 Formeln.

Zum Trainieren der Gewichte kommt das oben beschriebene diskriminative Lernen zum Einsatz. Die dafür benötigten Trainingsbeispiele bestehen jeweils aus von der Perzeption automatisch erzeugten Evidenzen und einem manuell annotierten RoadWorks Prädikat. Für die in Abbildung 4.12 dargestellte Verkehrsszene etwa ergibt sich

```

1 Beacons(133, "now", "many")
2 Beacons(133, "just", "many")
3 Beacons(133, "recently", "many")
4 RoadWorkSign(133, "now", "some")
5 RoadWorkSign(133, "just", "some")
6 RoadWorkSign(133, "recently", "some")
7 Marking(133, "white", "now", "none")
8 Marking(133, "white", "just", "some")
9 Marking(133, "white", "recently", "some")
10 Marking(133, "yellow", "now", "many")
11 Marking(133, "yellow", "just", "many")
12 Marking(133, "yellow", "recently", "lots")
13 RoadWorks(133)

```

mit der vom Benutzer vorgegebenen Annotation in Zeile 9 und den von den Detektionsverfahren gelieferten Evidenzen in den restlichen Zeilen.

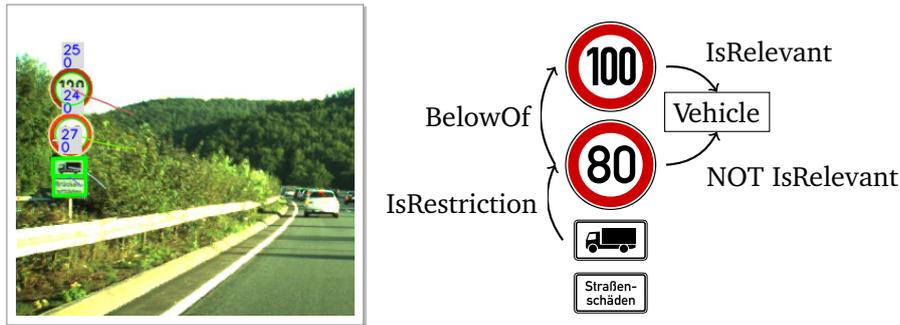


Abb. 4.13.: Zum Verständnis der Bedeutung von Schildgruppen muss neben der Perzeption die Informationen selbst und die Relationen zwischen den Objekten ausgewertet werden.

Information	Quelle
Tempolimit	Perzeption (Abschnitt 3.5.1)
Zusatzzeichen	Perzeption (Abschnitt 3.5.4)
Position zu Spuren	Perzeption (Abschnitt 4.2)
Fahrzeugtyp	Systemwissen
Datum und Uhrzeit	Systemuhr
Wetter (Regen)	Perzeption (Abschnitt 4.2)

Tab. 4.7.: Geeignete Informationsquellen zur Interpretation eingeschränkter Tempolimits.

Die Feststellung, ob eine Baustelle durchfahren wird, erfolgt durch Inferenz anhand der Wahrscheinlichkeit $P(\text{RoadWorks}|x)$ mit x der von den Detektionsverfahren gelieferten Evidenzen in der aktuellen Verkehrssituation. Die gewonnene Information kann z.B. zur Bestimmung des Situationskontexts in dem in Abschnitt 4.2 vorgestellten Ansatz verwendet werden.

4.3.5. Relevanzbestimmung für eingeschränkte Tempolimits

Wenn Tempolimits mit Zusatzzeichen kombiniert werden, ergeben sich zum Teil komplexe Schildgruppen (vergleiche Abbildung 4.13), deren Bedeutung nur dann richtig erfasst werden kann, wenn mehrere Informationen wie in Tabelle 4.7 dargestellt miteinander kombiniert werden. MLNs eignen sich zur Kombination der festgelegten semantischen Regeln mit den verrauchten Informationen der Perzeption. Im Folgenden werden Regeln zur Bestimmung der Fahrrelevanz von Tempolimits hergeleitet, die durch Zusatzzeichen eingeschränkt sind.

Ob ein Verkehrszeichen aufgrund seiner Anbringung relevant für eine Fahrspur ist, kann wie in Abschnitt 4.1 vorgestellt ermittelt werden. Die so ermittelten Zuordnungen M_{ij} stellen Evidenzen für ein Prädikat IsPosRel dar. Außerdem kann über die Fahrspurerkennung ermittelt werden, ob sich das eigene Fahrzeug auf einer bestimmten Fahrspur befindet, was auf ein Prädikat DrivesOn abgebildet wird.

4. Situationsinterpretation

```
1 IsPosRel(x,y,+c)
2 DrivesOn(y)
```

Erkannte Tempolimits werden als Prädikat PerSpeedLimit, Zusatzzeichen als PerSuppSign modelliert mit einer Eigenschaft PerHasType:

```
1 PerSpeedLimit(x,+c) => SpeedLimit(x)
2 PerSuppSign(x,+c) => SuppSign(x)
3 PerHasType(x,+c) => HasType(x)
```

Wenn mehrere Zusatzzeichen direkt übereinander montiert werden, adressieren sie jeweils das erste über ihnen angebrachte Tempolimit:

```
1 SuppSign(x1) AND SpeedLimit(x2) AND DirectlyBelowOf(x1,x2) ⊃
2   => IsRestriction(x1,x2)
3 SuppSign(x1) AND IsRestriction(x2,x3) AND BelowOf(x1,x2) ⊃
4   => IsRestriction(x1,x3)
```

Für einzelne Zusatzzeichen bzw. Gruppen von Zusatzzeichen mit korrespondierenden Bedeutungen werden eigene Regeln eingeführt. Beispielsweise stellt das Zusatzzeichen *Bei Nässe* eine Einschränkung dar, wenn es regnet:

```
1 IsRestriction(x1,x2) AND HasType(x1,"WhenRaining") AND IsWeather("Rain") ⊃
2   => RestrictionApplies(x1,x2)
```

Zuletzt werden die Information über die Relevanz aufgrund der Position sowie die Anordnung der Zusatzzeichen kombiniert:

```
1 NOT EXISTS x1,x2 IsRestriction(x1,x2) AND IsPosRel(x2,y) AND DrivesOn(y) ⊃
2   => IsRelevant(x2)
3 NOT RestrictionApplies(x1,x2) AND IsPosRel(x2,y) AND DrivesOn(y) ⊃
4   => NOT IsRelevant(x2)
```

Mit dieser Wissensbasis wird dann bei gegebener Evidenz x mittels Inferenz die Wahrscheinlichkeit $P(\text{IsRelevant}|x)$ berechnet.

4.3.6. Relevanzbestimmung für Ampeln

Ampeln sind fahrrelevant, wenn sie die Vorfahrt für die Spuren regeln, die auf der Fahrstrecke liegen. An Ampelkreuzungen sind in der Regel mehrere Ampeln sichtbar, von denen nicht alle fahrrelevant sind. Ihre Relevanz kann aufgrund der Ampelsymbole bzw. ihrer Position relativ zu Fahrspuren bestimmt werden. Die Erkennung der Ampelsymbole alleine reicht nicht aus, um in jedem Fall die Relevanz bestimmen zu können. Ein modellbasierter Ansatz (vergleiche Abschnitt 4.1) müsste zu komplexe Annahmen treffen, um eine rein bildbasierte Erkennung zu leisten. Daher wird anstelle der bildbasierten Fahrspurerkennung

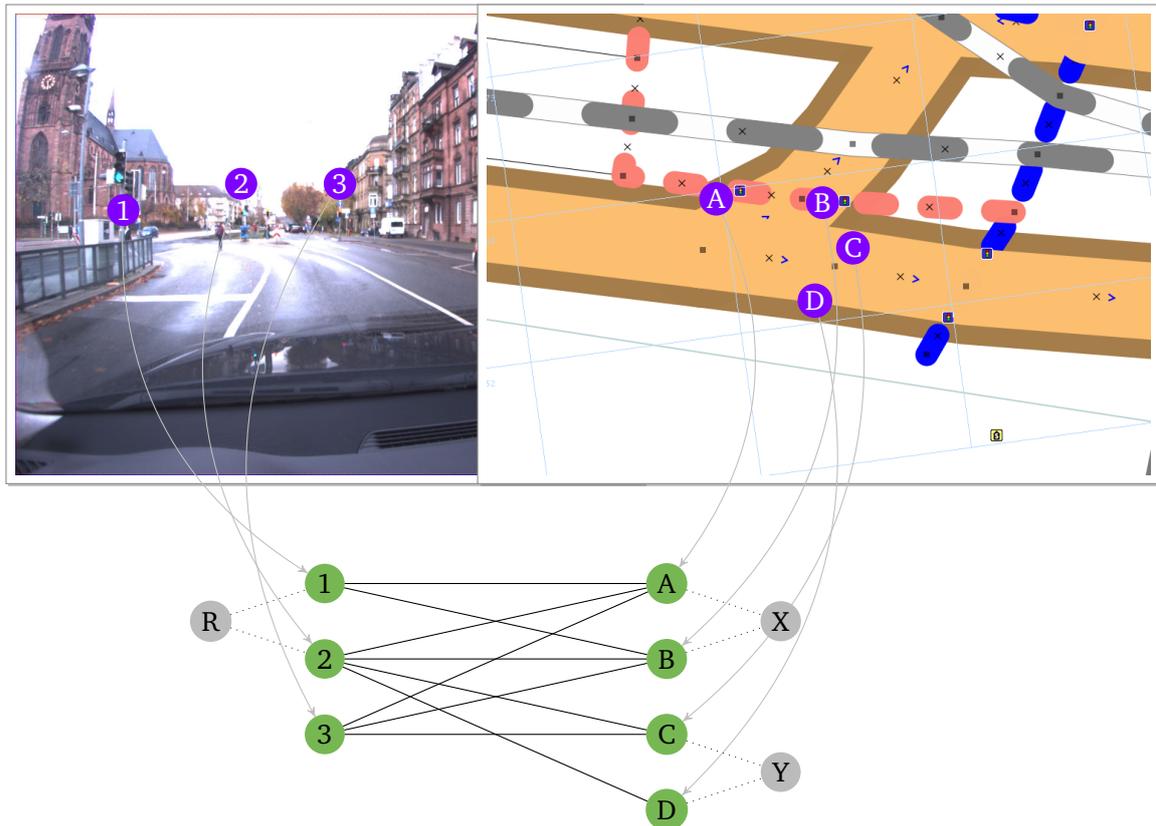


Abb. 4.14.: Zur Relevanzbestimmung werden Perzeption und Kartendaten kombiniert. Dazu wird per Inferenz eine plausible Zuordnung zwischen den Knoten bestimmt.

die Kenntnis der Fahrspur aus Kartendaten vorausgesetzt und die Problemstellung über die Fusion von Kartendaten und Perzeption gelöst. Über die Kartendaten ist in Verbindung mit der geplanten Route bekannt, welche Ampeln fahrrelevant sind (wenn der vorgegebenen Route gefolgt wird). Die Ampeln und der Ampelstatus wiederum müssen über die Perzeption erkannt werden. Um beide Informationen zu kombinieren, wird eine Zuordnung zwischen Ampeln sichtbar in der Kamera und Ampeln in den Kartendaten benötigt. Inferenz auf einem MLN kann diese Zuordnung ermitteln [NGZ11].

Zur Repräsentation der von der Perzeption erkannten Ampeln wird das Objekt `CamLight` eingeführt. Zeigen zwei Ampeln die gleiche Farbe, wird das durch das Prädikat `SameState` festgehalten. Über das Prädikat `LeftOf` wird die horizontale Reihenfolge (aus Fahrzeugsicht) von Ampeln modelliert. In Kartendaten verzeichnete Ampeln werden durch das Objekt `MapLight` repräsentiert. Das Prädikat `LeftOf` ist auch für sie anwendbar. Ob zwei Ampeln gleichzeitig geschaltet werden, wird über das Prädikat `Synchronized` modelliert. Um einen Zusammenhang zwischen Perzeption und Kartendaten herzustellen, wird das Prädikat `Compatible` eingeführt; eine erkannte Ampel ist `Compatible` mit einer Ampel in den Kartendaten, wenn beide kompatible Symbole und die gleiche Höhe (niedrig oder hoch) haben.

Die bisher vorgestellten Objekte und Prädikate stellen Evidenz dar. Es wird ein weiteres Prädikat `Matches` eingeführt, welches die Zuordnung zwischen Ampeln der beiden Informationsquellen modelliert (vergleiche Abbildung 4.14). Seine Wahrscheinlichkeit wird über Inferenz ermittelt und unterliegt

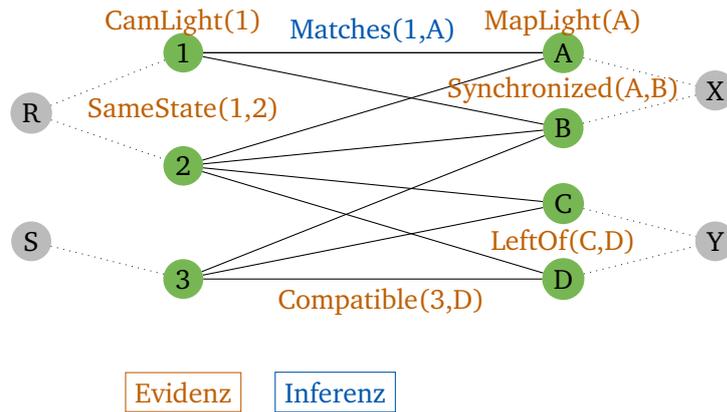


Abb. 4.15.: Zur Relevanzbestimmung für Ampeln wird per Inferenz eine Zuordnung zwischen Knoten des linken (Knoten 1,2,3) und des rechten Teilgraphen (Knoten A, B, C, D) bestimmt. Der linke Teilgraph repräsentiert sichtbare Ampeln. Aus Kartendaten entsteht der rechte Teilgraph. Die Gruppenschaltung der Ampeln (Prädikat Synchronized) ist durch die Knoten X und Y dargestellt. Die Kanten zwischen den beiden Teilgraphen entsprechen dem Prädikat Compatible.

weiteren Einschränkungen. Zunächst werden Regeln für die Zuordnung formuliert:

- 1 $\text{Matches}(x_1, y) \text{ AND } x_1 \neq x_2 \Rightarrow \text{NOT Matches}(x_2, y)$
- 2 $\text{Matches}(x, y_1) \text{ AND } y_1 \neq y_2 \Rightarrow \text{NOT Matches}(x, y_2)$
- 3 $\text{LeftOf}(x_1, x_2) \text{ AND } \text{LeftOf}(y_1, y_2) \text{ AND } \text{Matches}(x_1, y_2) \Rightarrow \text{NOT Matches}(x_2, y_1)$

Um überhaupt Zuordnungen erzeugen zu können, muss ein Zusammenhang zwischen der Kompatibilität von Ampeln und der Zuordnung bestehen. Außerdem sollen möglichst viele gültige Zuordnungen gefunden werden.

- 1 $w_1 \text{ Compatible}(x, y) \Rightarrow \text{Matches}(x, y)$
- 2 $w_2 \text{ EXISTS } x \text{ Matches}(x, y)$
- 3 $w_3 \text{ EXISTS } y \text{ Matches}(x, y)$

Als weitere Einschränkung wird der Status von Ampeln (Perzeption) bzw. die synchrone Schaltung von Ampeln (Kartendaten) berücksichtigt. Ampeln mit unterschiedlichem Status können nicht Ampeln der gleichen Schaltungsgruppe angehören:

- 1 $\text{NOT SameState}(x_1, x_2) \text{ AND } \text{Synchronized}(y_1, y_2) \text{ AND } \text{Matches}(x_1, y_1) \supset$
- 2 $\Rightarrow \text{NOT Matches}(x_2, y_2)$

Mit obigen Formeln kann die gesuchte Fahrrelevanz aus der Zuordnung gefolgert werden:

- 1 $\text{Matches}(x, y) \text{ AND } \text{IsRelevant}(y) \Rightarrow \text{IsRelevant}(x)$

Um die Relevanz der aktuell beobachteten Ampeln zu bestimmen, wird per Inferenz die Wahrscheinlichkeit $P(\text{IsRelevant}|x)$ bestimmt (vergleiche Abbildung 4.15).

4.4. Zusammenfassung

Die Situationsinterpretation leitet neue Informationen aus der Analyse von zwei oder mehr Entitäten einer Verkehrssituation ab. Damit kann der Einfluss der Umgebung auf den Erkennungsprozess berücksichtigt werden, wie in der kontextsensitiven Dempster/Shafer-Fusion gezeigt wird: Entitäten wie das Wetter, Dunkelheit oder Baustellen werden erkannt und als Einflussfaktoren der Zuverlässigkeit von Informationsquellen modelliert. Die so angepasste Fusion zur Erkennung des gültigen Tempolimits vermeidet Konflikte in Situationen, in denen andere Fusionsverfahren scheitern würden.

Die Fahrrelevanz von Verkehrselementen ergibt sich nicht immer aus dem Verkehrselement selbst. Ihre Anbringung muss berücksichtigt werden, um Verkehrselemente für andere Straßen oder Spuren als irrelevant einstufen zu können. Für außerstädtische Bereiche wurde dafür die probabilistische Zuordnung von Verkehrszeichen zu Fahrspuren vorgestellt. Sie fusioniert die Erkennung von Verkehrszeichen mit einer Fahrspurverfolgung und stellt räumliche Relationen zwischen Verkehrszeichen und Fahrspuren auf. Diese Relationen ordnen Verkehrszeichen entweder einzelnen Fahrspuren oder der gesamten Straße zu und können so zur Entscheidung über die Fahrrelevanz herangezogen werden.

Problemstellungen im Bereich der Situationsinterpretation zeichnen sich dadurch aus, dass Relationen zwischen verschiedenen Entitäten aufgestellt und ausgewertet werden müssen. Dabei müssen unsichere Informationen aus der Perzeption mit sicherem Hintergrundwissen wie Straßenverkehrsregeln kombiniert werden. Mit Markov-Logik-Netzen wurde ein Verfahren des statistischen relationalen Lernens auf die Situationsinterpretation übertragen. Seine flexible Einsatzmöglichkeit wurde exemplarisch durch Anwendungen zur Bestimmung von Baustellen, zur Bestimmung der Gültigkeit von Einschränkungen durch Zusatzzeichen sowie zur Bestimmung der Fahrrelevanz von Ampeln aufgezeigt.

Durch die vorgestellten Verfahren zur Umfelderkennung und Situationsinterpretation liegt ein Konzept vor, um die eingangs gestellte Forschungsfrage *Welche statischen Verkehrselemente sind in der aktuellen Verkehrssituation fahrrelevant?* durch ein kognitives Assistenzsystem automatisch beantworten zu können. Durch die Integration von Hintergrundwissen und die Anwendung von Lernverfahren zur Parameteroptimierung ist es universell einsetzbar. Die nun folgenden Experimente führen eine umfangreiche Evaluation verschiedener Aspekte und gesamter Systeme durch, um das Konzept empirisch zu validieren.

5. Experimente und Evaluation

Zur experimentellen Evaluation der Ansätze kommen verschiedene Versuchsplattformen zum Einsatz (Abbildung 5.1). Kraftfahrzeuge als Anwendungsplattform der kognitiven Assistenzsysteme in Form von Fahrerassistenzsystemen und Anwendungen zum (teil-)autonomen Fahren stellen im Großteil der Experimente den Versuchsträger dar. Darüber hinaus wurde ein Segway-Personentransporter verwendet, um die Übertragbarkeit der Ansätze auf eine weitere Transportart mit anderem Verhalten zu untersuchen (Abschnitt 5.6, Seite 148). Bei der Verwendung von Mobilgeräten in Abschnitt 5.7 (Seite 151) wiederum erfolgt der Einsatz im Fahrzeug.

Im Bereich der Umfelderkennung werden die einzelnen in Abschnitt 3.5 vorgestellten Anwendungen zur Erkennung von Ronden, Warnhinweisen, Vorwegweisern, Zusatzzeichen und Ampeln evaluiert. Neben der Untersuchung der Gesamtsysteme werden ihre Teilkomponenten evaluiert. Dabei wird ein Schwerpunkt auf Untersuchungen zur Skalierbarkeit der Klassifikation durch hierarchische SVMs und aktives Lernen gelegt. Anschließend wird in Abschnitt 5.6 mit der semi-automatischen Kartierung ein neuartiges Assistenzsystem beschrieben und evaluiert. Abschnitt 5.7 widmet sich Untersuchungen zur Übertragbarkeit der Ansätze auf Mobilgeräte.

Im Bereich der Situationsinterpretation wird in Abschnitt 5.8 das Verfahren zur Analyse der räumlichen Relationen zwischen Fahrspuren und Verkehrszeichen evaluiert. Die kontextsensitive Dempster/Shafer-Fusion zur Bestimmung des gültigen Tempolimits wird in Abschnitt 5.9 untersucht. Hierfür werden Perzeption und Kartendienste unter besonderer Berücksichtigung des Situationskontexts fusioniert. Der folgende Abschnitt 5.10 beschreibt Experimente zur Erkennung von Baustellen und zur Bestimmung der Fahrrelevanz von Ampeln durch Markov-Logik-Netze.



Abb. 5.1.: Als Versuchsträger kommen verschiedene Kraftfahrzeuge mit eingebauter Sensorik, Segway-Personentransporter und Mobilgeräte zum Einsatz.

5.1. Erkennung von Ronden

Innerhalb der Umfelderkennung wurde mit der Entwicklung der hierarchischen SVM und der TPCB-ADR-Auswahlstrategie zum aktiven Lernen ein Schwerpunkt auf Untersuchungen zur Skalierbarkeit der Klassifikation gelegt. Um sie zu evaluieren, werden große Datenmengen benötigt. Ronden und insbesondere Tempolimits werden besonders häufig aufgestellt und eignen sich folglich gut, um eine derartige Datenmenge bereitzustellen. In den Experimenten in diesem Abschnitt wird daher die in Abschnitt 3.5.1 beschriebene Erkennung von Ronden evaluiert mit Schwerpunkt auf Untersuchungen zur hierarchischen SVM und zum aktiven Lernen.

Zur Durchführung der Experimente wurden drei disjunkte Datenmengen mit einer Gesamtlänge von mehr als 21 Stunden zusammengestellt:

Datenmenge	Aufnahmen	Dauer	Bilder
Trainingsmenge	32	19:06:14	1921285
Testmenge A	1	00:20:32	29546
Testmenge B	3	02:11:07	232415

In Testmenge A sind alle ROIs aller Bilder annotiert, um sie für automatische Auswertungen nutzen zu können. Sie wird für Experimente zur hierarchischen SVM, zum aktivem Lernen und zur Ermittlung der Falschpositivrate genutzt. Die Testmenge B ist deutlich größer; sie wird zusammen mit Testmenge A zur Ermittlung von Erkennungsraten des Gesamtsystems genutzt.

Skalierbarkeit der Klassifikation

Als primäre Anforderung an die Klassifikation wird eine hohe Klassifikationsrate gestellt. Werden Daten online verarbeitet, ist zusätzlich eine maximale Laufzeit einzuhalten. Beim Einsatz in Steuergeräten und anderen eingebetteten Systemen mit beschränkter Speicherkapazität ist darüber hinaus die maximale Größe des trainierten Modells begrenzt. Wo genau diese Grenzen liegen, ist anwendungsabhängig. Die Auswahl der Merkmalstransformation, die Art des Umgangs mit Mehrklassen-Problemen (*one-against-one*-Verfahren oder hierarchische SVM) und der Kernel der SVMs (vergleiche Abschnitt 3.3.6) haben wesentlichen Einfluss auf die Performanz im Hinblick auf Laufzeit, Modellgröße und Klassifikationsrate. Um einen Klassifikator mit aus Sicht der Anwendung optimalen Eigenschaften zu erhalten, müssen daher verschiedene Kombinationen dieser Parameter evaluiert werden. Dies wird in diesem Experiment exemplarisch für Tempolimits durchgeführt; das Vorgehen lässt sich wie weiter unten diskutiert auf andere Arten von Verkehrselementen übertragen. Um die Einflüsse der Parameter zu quantifizieren und geeignete Werte zu selektieren, werden verschiedene Parameterkombinationen benutzt und anhand folgender Kriterien evaluiert:

- Die Klassifikationsrate,
- die durchschnittliche Laufzeit und

ω	Training	Test
background	3318	7516
end	966	
other	9544	425
10 km/h	78	
30 km/h	375	
40 km/h	125	
50 km/h	202	
60 km/h	2854	369
70 km/h	653	
80 km/h	2457	34
100 km/h	2255	72
120 km/h	3998	89
130 km/h	329	

Tab. 5.1.: Anteile der einzelnen Klassen in der Trainings- und Testdatenmenge.

- die Größe des trainierten Modells.

Zur Klassifikation von Tempolimits eignen sich als Merkmalstransformationen die Grauwerttransformation und die blockweise diskrete Kosinustransformation. Als Struktur der hierarchischen SVM wird die in Abbildung 4.7 dargestellte verwendet.

Mit dem in Abschnitt 3.5.1 beschriebenen Verfahren wurden insgesamt 27154 ROIs aus der Trainingsmenge extrahiert und manuell einer der in Tabelle 5.1 aufgelisteten Klassen zugewiesen. Die Testmenge von 8505 ROIs stammt aus Testmenge A.

Vergleich von Kernel-Typen und Grauwerttransformation Zunächst werden verschiedene Kernel-Typen bei Verwendung der Grauwerttransformation evaluiert. Die Ergebnisse sind in Abbildung 5.2 dargestellt. Die höchste Klassifikationsrate wird mit dem RBF-Kernel im *one-against-one*-Verfahren erzielt. Mit einer Laufzeit von durchschnittlich etwa 10 ms pro ROI stellt diese Kombination auch die langsamste dar. Die schnellste Kombination ist eine hierarchische SVM mit polynomiellen Kernel vom Grad 2. Die hierarchische SVM ist mit allen Kernel-Typen im Durchschnitt etwa doppelt so schnell wie das *one-against-one*-Verfahren, erzielt aber auch immer etwas geringere Klassifikationsraten. Dies liegt an einer höheren Falschpositivrate; die Unterscheidung positiver Klassen ist immer besser als die des *one-against-one*-Verfahrens.

Blockweise Diskrete Kosinustransformation und RBF-Kernel Zum Vergleich der Merkmalstransformationen wurde die Grauwerttransformation durch eine blockweise diskrete Kosinustransformation ersetzt. Abbildung 5.3 zeigt ihre Ergebnisse für verschiedene Parametrierungen. Analog zum vorherigen Experiment ist die Geschwindigkeit der hierarchischen SVM knapp doppelt so hoch wie die des *one-*

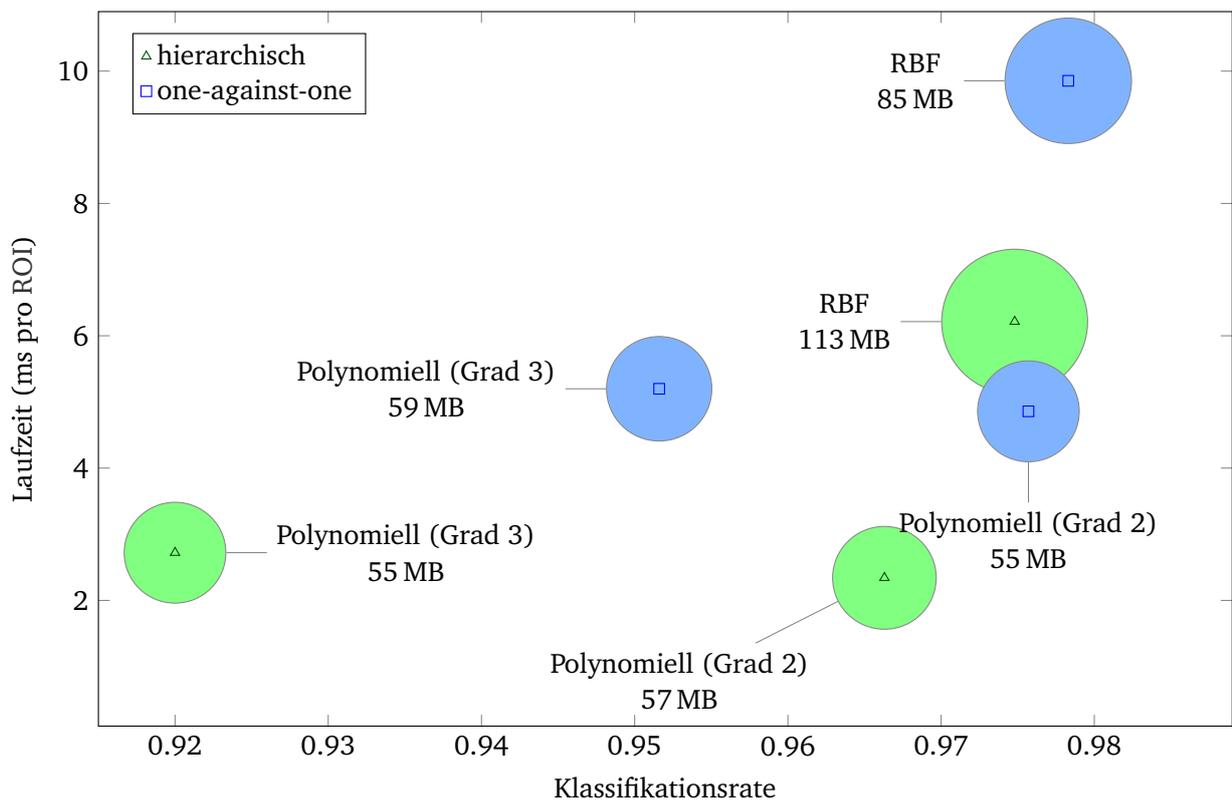


Abb. 5.2.: Performanz für verschiedene Kernel-Typen und Grauwerttransformation einer Größe von 20×20 Bildpunkten. Die Beschriftung gibt in der ersten Zeile die Art des Kernels und in der zweiten Zeile die Modellgröße an. Letztere wird zusätzlich durch den Flächeninhalt des Kreises veranschaulicht.

against-one-Verfahrens bei ansonsten gleicher Parametrierung. Über die Parameter s und insbesondere n (siehe Abbildung 5.3) wird analog zum JPEG-Verfahren die Stärke der Kompression gesteuert. Die höchste Klassifikationsrate wird allerdings nicht von einer Parametrierung minimaler Kompression erzielt, sondern für $n = 10$ Koeffizienten pro Blockmatrix bei Verwendung der hierarchischen SVM.

Blockweise Diskrete Kosinustransformation und Polynomieller Kernel Für den Einsatz auf Steuergeräten kann die Verwendung eines polynomiellen Kernels von Vorteil sein, um die Berechnung der Exponentialfunktion zu vermeiden. Aus diesem Grund wurde auch die Kombination von polynomiellen Kernel und blockweiser diskreter Kosinustransformation untersucht. Für polynomielle Kernel vom Grad 2 ergibt sich hier ein signifikanter Abfall der Klassifikationsrate auf maximal 93.35%. Polynomielle Kernel vom Grad 3 dagegen erzielen bessere Ergebnisse. Sie sind in Abbildung 5.4 dargestellt. Auch hier arbeitet die hierarchische SVM etwa doppelt so schnell bei gleichen Parametern. Die Klassifikationsraten sind insgesamt (zum Teil signifikant) geringer verglichen mit dem RBF-Kernel. Einige Kombinationen erzielen jedoch durchaus praxistaugliche Klassifikationsraten. Modellgröße und Laufzeit sind insgesamt geringer verglichen mit dem RBF-Kernel.

Diskussion Aus den durchgeführten Experimenten lassen sich wichtige Resultate ableiten:

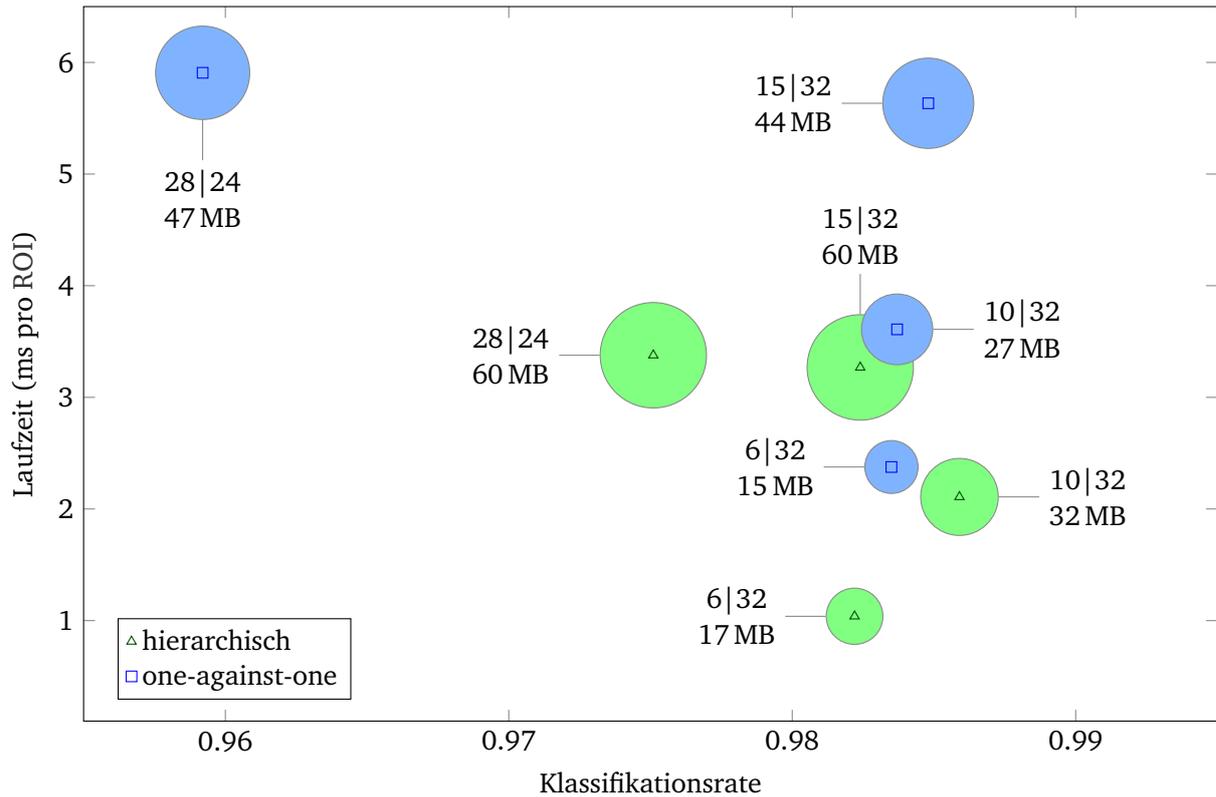


Abb. 5.3.: Performanz für Variationen der blockweisen DCT mit Blockmatrizen der Größe $s_x = s_y = 8$ und Verschiebung $o_x = o_y = 8$ angewendet auf ROIs der Größe $s \times s$. Für jede Blockmatrix werden die ersten n Koeffizienten extrahiert, die Dimension des Merkmalsraums ist folglich $n(s/8)^2$. Die Beschriftung gibt in der ersten Zeile n und s an und in der zweiten Zeile die Modellgröße an. Letztere wird zusätzlich durch den Flächeninhalt des Kreises veranschaulicht. Die SVMs verwenden einen RBF-Kernel.

- Die Laufzeit variiert bei vergleichbaren Klassifikationsraten um den Faktor 10. Die hierarchische SVM arbeitet im Durchschnitt etwa doppelt so schnell wie das *one-against-one*-Verfahren.
- Die Klassifikationsrate der hierarchischen SVM ist vergleichbar mit der des *one-against-one*-Verfahrens, aber meist etwas geringer.
- RBF-Kernel erzielen höhere Klassifikationsraten als polynomielle Kernel. Es gibt jedoch Parameterkombinationen, bei denen polynomielle Kernel vergleichbare Klassifikationsraten erzielen.
- Die höchste Klassifikationsrate wird nicht mit der Grauwerttransformation, sondern der blockweisen diskreten Kosinustransformation erzielt.

Die starken Unterschiede bezüglich der Laufzeit gehen auf zwei Ursachen zurück. Zum einen variiert die Dimension der verwendeten Merkmalsräume zwischen 104 und 400. Dies wirkt sich unmittelbar auf die Laufzeit aus, welche von der Kernelfunktion dominiert wird, die linear von der Dimension abhängt. Zum anderen wird bei der Verwendung der hierarchischen SVM aufgrund der gewählten Struktur ein Großteil der Daten bereits in der ersten Stufe als Hintergrunddaten klassifiziert und nicht weiter verarbeitet. Die Varianz der Laufzeit ist für die hierarchische SVM höher, da positiv klassifizierte Daten mehrere Level im Baum durchlaufen.

Die erste Stufe im Baum – die Unterscheidung positiver und negativer Daten – ist auch die Ursache für die in der Regel etwas geringere Klassifikationsrate der hierarchischen SVM. Hier werden mehr

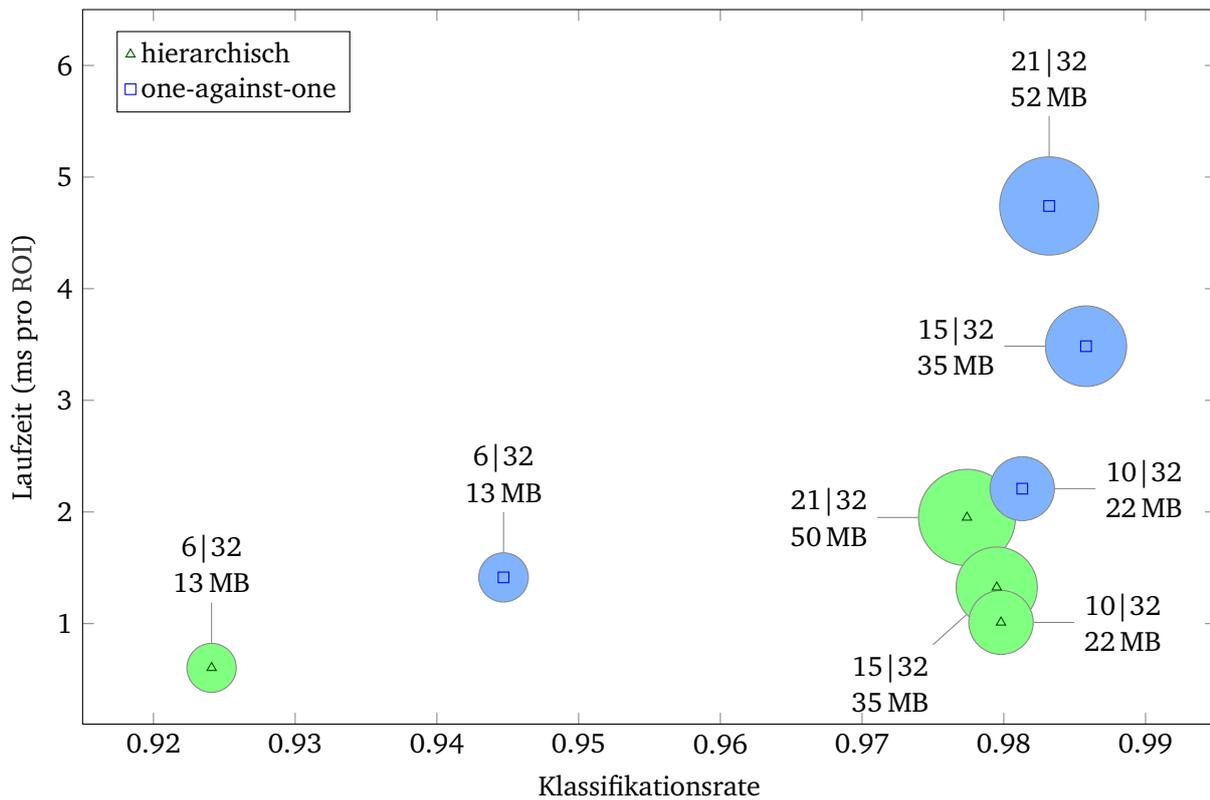


Abb. 5.4.: Performanz für Variationen der blockweisen DCT mit Blockmatrizen der Größe $s_x = s_y = 8$ und Verschiebung $o_x = o_y = 8$ angewendet auf ROIs der Größe $s \times s$. Für jede Blockmatrix werden die ersten n Koeffizienten extrahiert, die Dimension des Merkmalsraums ist folglich $n(s/8)^2$. Die Beschriftung gibt in der ersten Zeile n und s an und in der zweiten Zeile die Modellgröße an. Letztere wird zusätzlich durch den Flächeninhalt des Kreises veranschaulicht. Die SVMs verwenden einen polynomiellen Kernel vom Grad 3 (siehe Text).

Falschpositive zugelassen als im *one-against-one*-Verfahren. Andererseits ist die Verwechslungsrate positiver Beispiele geringer. Die tendentiell höhere Klassifikationsrate bei Verwendung von RBF-Kernen bestätigt die Erwartungen [HCL03]. Falls der Einsatz eines polynomiellen Kerns jedoch gewünscht ist (zur Vermeidung der Exponentialfunktion), lassen sich Parameterkombinationen mit praxistauglichen Klassifikationsraten finden.

Der Vergleich der Merkmalstransformationen weist die blockweise diskrete Kosinustransformation als besser geeignet aus. Sie führt zu einer geringeren Laufzeit, einer geringeren Modellgröße und einer höheren Klassifikationsrate. Die hierarchische SVM erzielt wie gewünscht eine deutliche Steigerung der Laufzeit bei einer vergleichbaren Klassifikationsrate. Sie eignet sich also insbesondere dann, wenn die Anzahl der zu unterscheidenden Klassen groß ist und die Laufzeit der Klassifikation reduziert werden soll.

Aktives Lernen

Nach der Bestimmung geeigneter Parameterkombinationen für die SVMs wird in diesem Abschnitt die Suche geeigneter Lerndaten evaluiert: Aktives Lernen und insbesondere die Auswahlstrategie track-pruned

class-biased angular diversity ranking (TPCB-ADR) wurden zur Verbesserung der Klassifikationsleistung und zur Reduktion des Annotationsaufwands entworfen, indem automatisch nicht annotierte Lerndaten ausgewählt werden (siehe Abschnitt 3.3.7). Zur experimentellen Evaluation der Leistungsfähigkeit des Ansatzes wird an dieser Stelle ein Vergleich verschiedener Auswahlstrategien, die aktives und passives Lernen implementieren, auf einem umfangreichen Datensatz vorgenommen [NZ13].

Auswahlstrategien und Aufbau der Experimente Das einfachste passive Lernverfahren basiert auf einer komplett zufälligen Auswahl der Trainingsdaten. Sinnvoller ist jedoch die Integration der Nebenbedingungen [3.56] – [3.58], also eine Auswahl der Trainingsdaten, die maximal t Beispiele pro Track mit einem Anteil $p/(p+n)$ positiv klassifizierter Beispiele zufällig auswählt. Zur besseren Vergleichbarkeit der Ergebnisse mit aktiven Auswahlstrategien wird diese Auswahl ebenfalls in Batches S mit $|S|=h$ durchgeführt. Diese passive Auswahlstrategie wird im Folgenden als track-pruned class-biased random sampling (TPCB-RS) bezeichnet.

Als aktive Auswahlstrategie wird die in Abschnitt 3.3.7 vorgestellte track-pruned class-biased angular diversity ranking (TPCB-ADR) evaluiert. Der Vergleich von TPCB-RS und TPCB-ADR zeigt die Unterschiede zwischen passivem und aktivem Lernen. Um auch den Einfluss von angular diversity ranking (ADR) zu bestimmen, wird außerdem eine weitere Auswahlstrategie track-pruned class-biased uncertainty sampling (TPCB-US) evaluiert: Sie kombiniert uncertainty sampling (US) [Set09] mit den Nebenbedingungen [3.56] – [3.58], ohne redundante Daten zu filtern.

Um ein realistisches Szenario abzubilden, kommt ein großer Datensatz von circa 19 Stunden Videoaufnahmen aus verschiedenen Jahren mit unterschiedlichen Jahres- und Tageszeiten sowie Wetterbedingungen zum Einsatz (Tabelle 5.2). Von den insgesamt über 100000 ROIs wurden etwa 25000 initial annotiert. Sie bilden die initiale Trainingsmenge. Dabei wurden vornehmlich positive Beispiele annotiert; die initiale Trainingsmenge eignet sich also nicht zur Abschätzung des Anteils von positiven Beispielen (vergleiche Tabelle 5.2b). Die übrigen circa 75000 nicht annotierten Trainingsbeispiele wurden den drei Auswahlstrategien jeweils als Pool zur Verfügung gestellt (englisch *pool-based active learning*).

Das Training der SVM erfolgte jeweils mit identischen Parametern auf der aktuellen Trainingsmenge, die Auswahl neuer Trainingsbeispiele durch die gegebene Auswahlstrategie. Zur Annotation wurden die Klassen $\Omega_l := \Omega \cup \{\omega_{-1}\}$ verwendet: Das ursprüngliche Klassifikationsproblem wurde um die Klasse ω_{-1} (*unsicher*) erweitert, mit dem Trainingsbeispiele annotiert wurden, die keiner Klasse eindeutig zugeordnet werden konnten (vergleiche Abbildung 5.5). Diese Beispiele wurden nicht in die Trainingsmenge übernommen.

Vergleich von Leistungskennwerten Im ersten Experiment wurden für alle drei Auswahlstrategien jeweils fünf Iterationen mit den in Tabelle 5.3 dargestellten Parametern durchgeführt. Für jede Iteration wurde eine Evaluation auf den Testdaten durchgeführt. Abbildung 5.6 zeigt die Klassifikationsraten auf den Testdaten in Abhängigkeit von der Anzahl der Trainingsbeispiele. Dabei fällt auf, dass nicht alle Iterationen die Trainingsdatenmenge um $h = 1000$ Beispiele erhöhen. Das liegt daran, dass die als ω_{-1} annotierten Beispiele nicht für das Training verwendet wurden.

Trainingsdaten			Testdaten		
Videos	32	1	background	10134	7522
Frames	1921285	29546	end of limit	963	0
Länge	19:05:56	00:20:32	10 km/h	78	0
ROIs	101439	8086	30 km/h	363	0
(a) Daten			40 km/h	119	0
			50 km/h	200	0
			60 km/h	2803	369
			70 km/h	636	0
			80 km/h	2406	34
			100 km/h	2236	72
			120 km/h	3984	89
			130 km/h	329	0
			(b) Klassenverteilung		

Tab. 5.2.: Übersicht der zur Evaluation des aktiven Lernens verwendeten Trainings- und Testdaten.

Parameter	Symbol	Wert
Batch-Größe	h	1000
Klassen-Vorzug	κ_p	500
	κ_n	500
Track-Pruning	κ_t	3
Ranking-Gewicht	λ	0.8

Tab. 5.3.: Verwendete Parameter für das aktive Lernen.

Ω_p (positiv)			
	80 km/h	80 km/h	100 km/h
ω_{-1} (unsicher)			
	100 km/h	60 km/h	80 km/h
Ω_n (negativ)			
	40 km/h	maximale Breite	80 km/h (ungültig)

Abb. 5.5.: Trainingsbeispiele aus der fünften Iteration von TPCB-ADR, die schwer einer Klasse zuzuordnen sind. Unterhalb der Beispiele ist die wahre Klasse angegeben, welche gegebenenfalls durch die Betrachtung zeitlich benachbarter Videobilder ermittelt wurde. Die erste Spalte jeder Reihe gibt die Art der Annotation an.

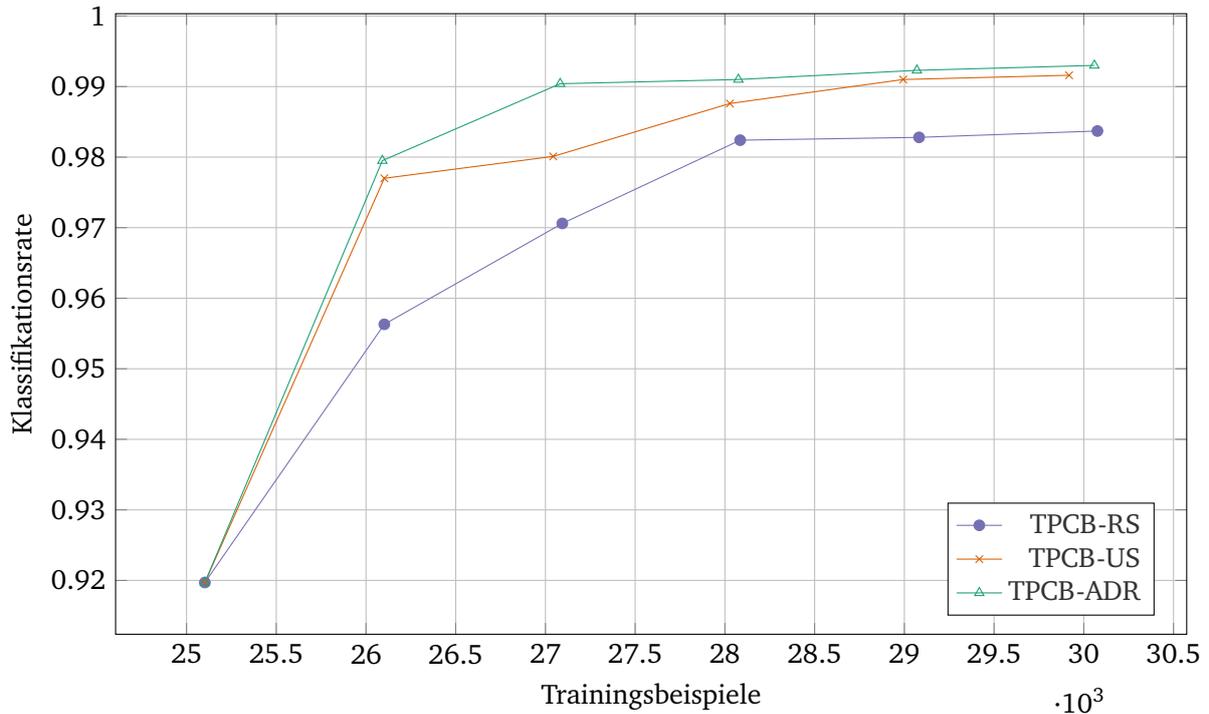


Abb. 5.6.: Klassifikationsrate verschiedener Auswahlstrategien während fünf Iterationen mit einer Batch-Größe von $h = 1000$. Die initiale Trainingsmenge besteht aus 25102 von Hand annotierten Trainingsbeispielen.

Die beiden aktiven Lernstrategien erzielen durchweg deutlich höhere Klassifikationsraten. Nach fünf Iterationen erreicht die passive Strategie TPCB-RS circa 98%, ein Wert, den TPCB-US und TPCB-ADR bereits nach einer Iteration fast erreichen. Die Strategie TPCB-ADR zeigt in den ersten beiden Iterationen einen besonders ausgeprägten Anstieg der Klassifikationsrate und erzielt in allen Iterationen die höchsten Werte.

Neben einer hohen gesamten Klassifikationsrate ist die Reduktion der Falschpositivrate ein Ziel des verwendeten Ansatzes. Abbildung 5.7 zeigt die Falschpositivrate in Abhängigkeit von der Anzahl negativer Trainingsbeispiele. Beim Übergang von der passiv gelernten initialen Trainingsmenge (bis 10134 Beispiele) zu den beiden aktiven Lernstrategien TPCB-US und TPCB-ADR fällt der starke Abfall der Falschpositivrate auf. Hierbei ist jedoch zu beachten, dass die initiale Trainingsmenge hauptsächlich andere Verkehrszeichen als Negativbeispiele enthält und damit eine hohe Redundanz enthält. Entsprechend wird durch den Übergang auf TPCB-RS ebenfalls eine merkliche (aber weniger starke) Reduktion der Falschpositivrate erzielt. Die TPCB-ADR-Strategie erreicht bereits in der zweiten Iteration eine sehr niedrige Falschpositivrate und erreicht in allen Iterationen die niedrigste Falschpositivrate. Des Weiteren fällt der starke Unterschied der Anzahl negativer Trainingsbeispiele auf; TPCB-US selektiert deutlich weniger negative Beispiele als die beiden anderen Strategien. Das geht einerseits auf die oben angesprochenen als *unsicher* aussortierten Trainingsbeispiele zurück, aber andererseits auch auf eine höhere Anzahl positiver Beispiele.

Abbildung 5.8 stellt die Anzahl der Supportvektoren im Modell in Abhängigkeit von der Anzahl an Trainingsbeispielen dar. Die nahezu konstante Steigung für TPCB-ADR ist ein weiteres Indiz dafür, dass diese Strategie redundante Trainingsbeispiele effektiv vermeidet. Bei zufälliger Auswahl der Trainingsbeispiele

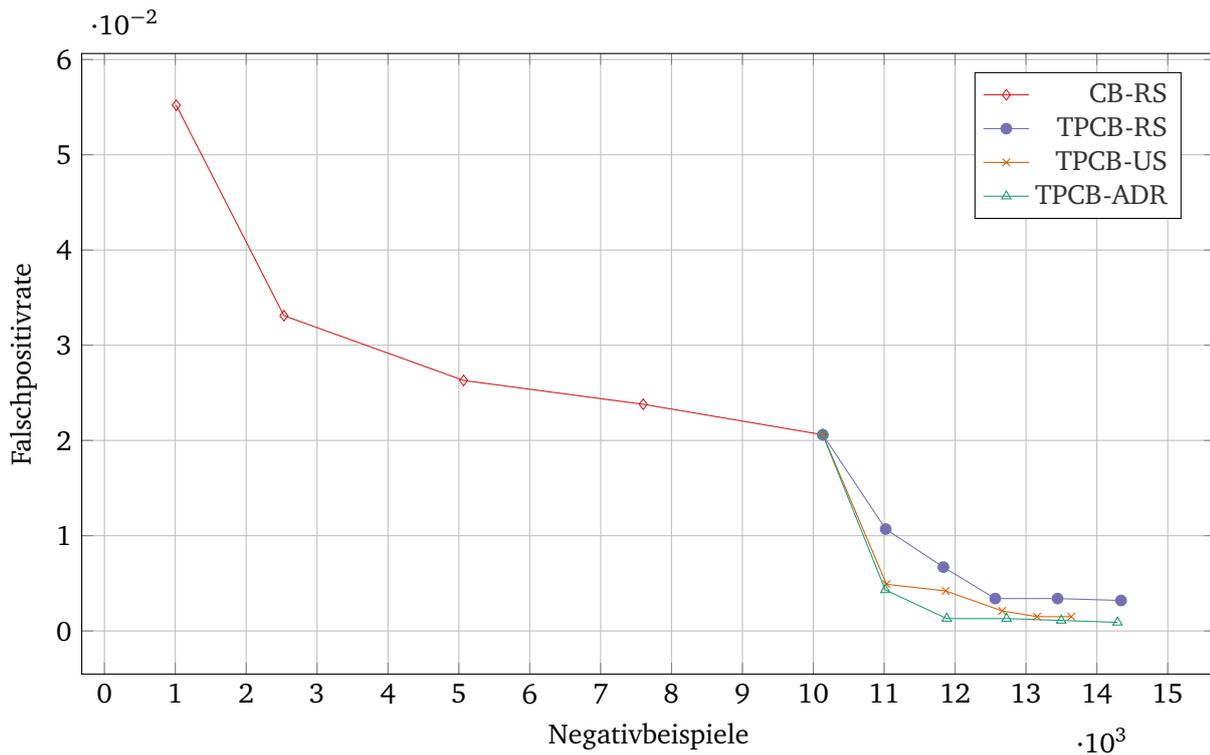


Abb. 5.7.: Falschpositivrate in Abhängigkeit von der Anzahl negativer Trainingsbeispiele. Die initiale Trainingsmenge (bis 10134 Beispiele) wurde per class-biased random sampling (CB-RS) generiert.

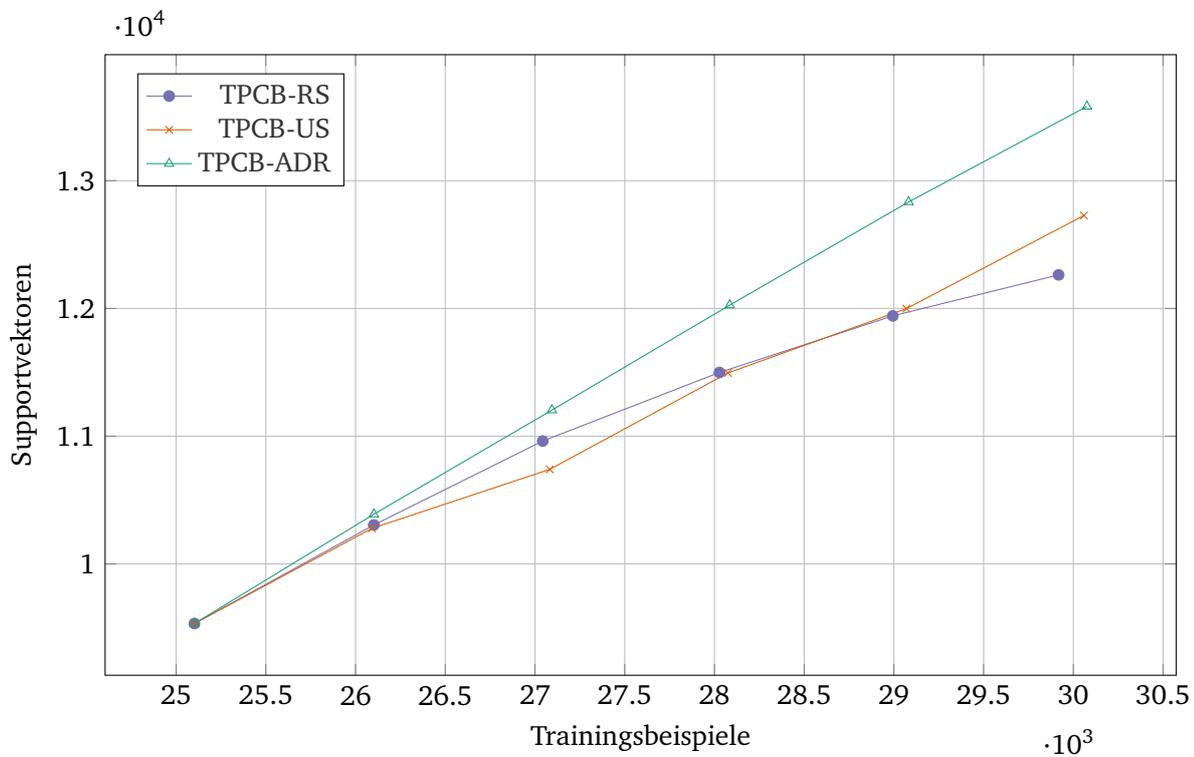


Abb. 5.8.: Anzahl der Supportvektoren im Modell in Abhängigkeit von der Anzahl an Trainingsbeispielen.

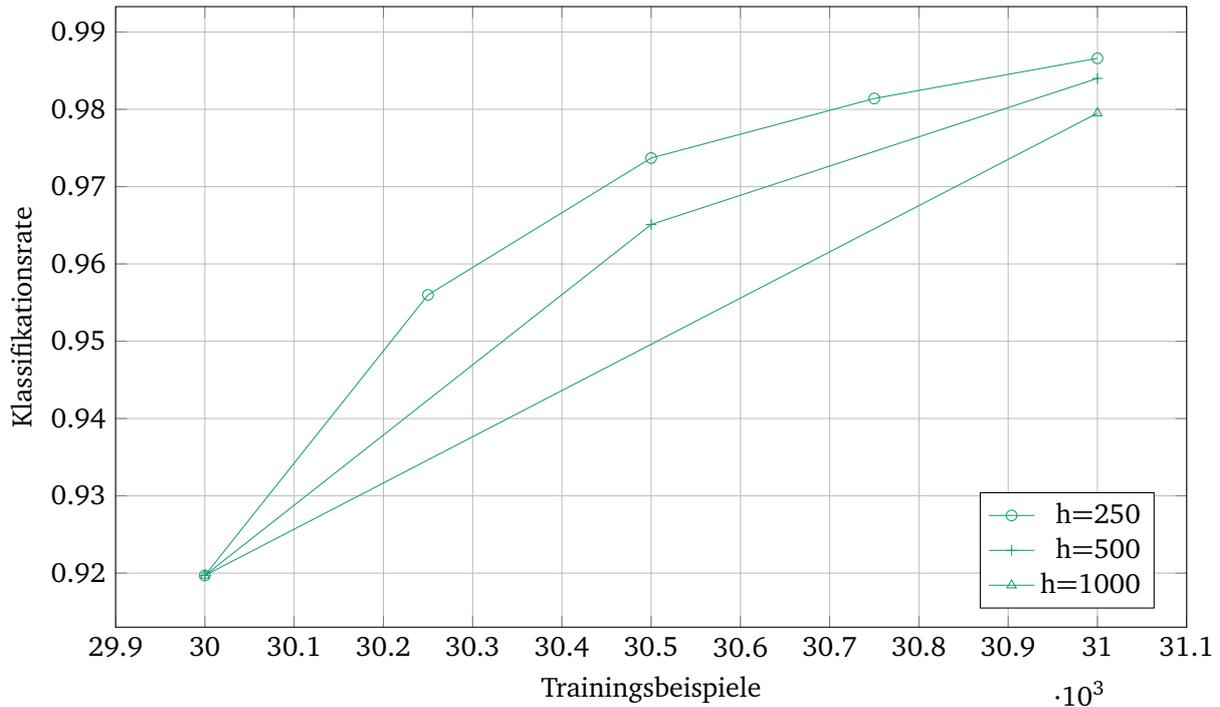


Abb. 5.9.: Vergleich der Klassifikationsrate der TPCB-ADR-Strategie bei Verwendung unterschiedlicher Batch-Größen.

durch TPCB-US dagegen nimmt die Steigung kontinuierlich ab. Dies lässt sich dadurch erklären, dass bei einer wachsenden Menge von Trainingsdaten ohne Redundanzbetrachtung die Wahrscheinlichkeit steigt, dass neue Trainingsbeispiele redundant zu den vorhandenen sind und nicht als Supportvektoren ausgewählt werden.

Die Größe eines Batches ist mit $h = 1000$ deutlich höher gewählt als in vergleichbaren Publikationen [Bri03; JPP10; DPB11]. Um die Auswirkung der Batch-Größe abzuschätzen, wurden weitere Versuche mit $h = 500$ und $h = 250$ durchgeführt. Abbildung 5.9 stellt die Ergebnisse für die TPCB-ADR-Strategie dar. Geringere Werte von h führen zu einem stärkeren Anstieg der Klassifikationsrate. Nach Verarbeitung der identischen Menge von Trainingsbeispielen erzielen Strategien mit geringeren Werten von h höhere Klassifikationsraten.

Auswirkungen auf die Laufzeit Überwachte Lernverfahren bestehen aus einer Trainingskomponente, die ein Modell erzeugt, und einer Klassifikationskomponente, die ausgehend vom gelernten Modell gegebenen Daten Klassen zuweist. Erstere wird einmalig ausgeführt bevor das System im Live-Betrieb läuft (*offline*), zweite im Live-Betrieb des Systems (*online*). Wie in Abbildung 3.23 skizziert ist diese Einteilung auch für das aktive Lernen gültig. Sie ist insofern wichtig, als dass sie in unterschiedlichen Laufzeit-Anforderungen mündet: Online muss die Verarbeitung in Echtzeit erfolgen, offline sind deutlich größere Verarbeitungszeiten tolerierbar. Tabelle 5.4 stellt die Laufzeiten der drei evaluierten Strategien in der ersten Iteration des ersten Experiments gegenüber. Sie unterscheiden sich jeweils in der Laufzeit der Berechnung des Batches. Insgesamt ergeben sich Laufzeiten von circa 40 bis circa 70 Minuten für eine Iteration des offline durchgeführten Trainings.

	TPCB-RS	TPCB-US	TPCB-ADR
$T' := T \cup S$	21 min	21 min	21 min
Training von T'	12 min	12 min	12 min
$A := A \setminus S$	8 min	8 min	8 min
Berechnung von S	0 min	17 min	26 min
Summe	41 min	58 min	67 min

Tab. 5.4.: Laufzeiten der verschiedenen Strategien für die einzelnen Arbeitsschritte einer Iteration im aktiven Lernen. Hierbei bezeichnet T die Trainingsmenge, A den Pool nicht annotierter Daten und S die Auswahlmenge (Batch).

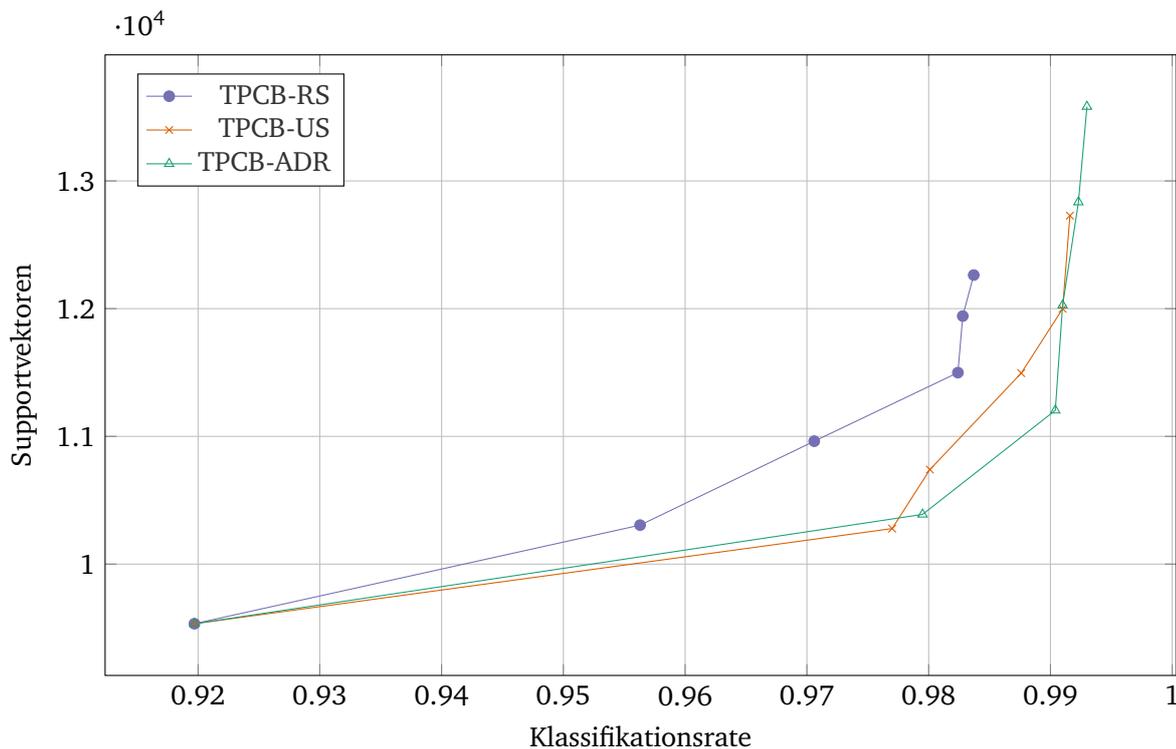


Abb. 5.10.: Die Anzahl der Supportvektoren in Abhängigkeit von der Klassifikationsrate eignet sich ebenfalls als Abbruchkriterium für das iterative Training: Eine hohe Steigung deutet auf eine sinkende Lernrate hin.

Die Laufzeit der online durchzuführenden Klassifikation hängt maßgeblich von der Anzahl der Supportvektoren im Modell ab. Mit steigender Zahl von Supportvektoren steigt die Laufzeit der Klassifikation linear an. Aus diesem Grund wird als Indikator für die Laufzeit der Klassifikation die Anzahl an Supportvektoren in Abhängigkeit von der Klassifikationsrate betrachtet. Sie ist in Abbildung 5.10 dargestellt. Diese Darstellung eignet sich auch als weiteres Abbruchkriterium: Das iterative Training kann abgebrochen werden, wenn die Steigung dieser Kurve zu hoch wird, also die Laufzeit der Klassifikation stark steigt, während die Klassifikationsrate sich kaum noch verbessert.

Diskussion Die durchgeführten Experimente zeigen deutlich die Vorteile von aktiven Lernstrategien. Die TPCB-ADR-Strategie erzielt die besten Resultate bei einer moderaten Erhöhung der Trainingsdauer. Durch die Verwendung einer sehr großen Batch-Größe von $h = 1000$ wird die Arbeitszeit des Annotati-

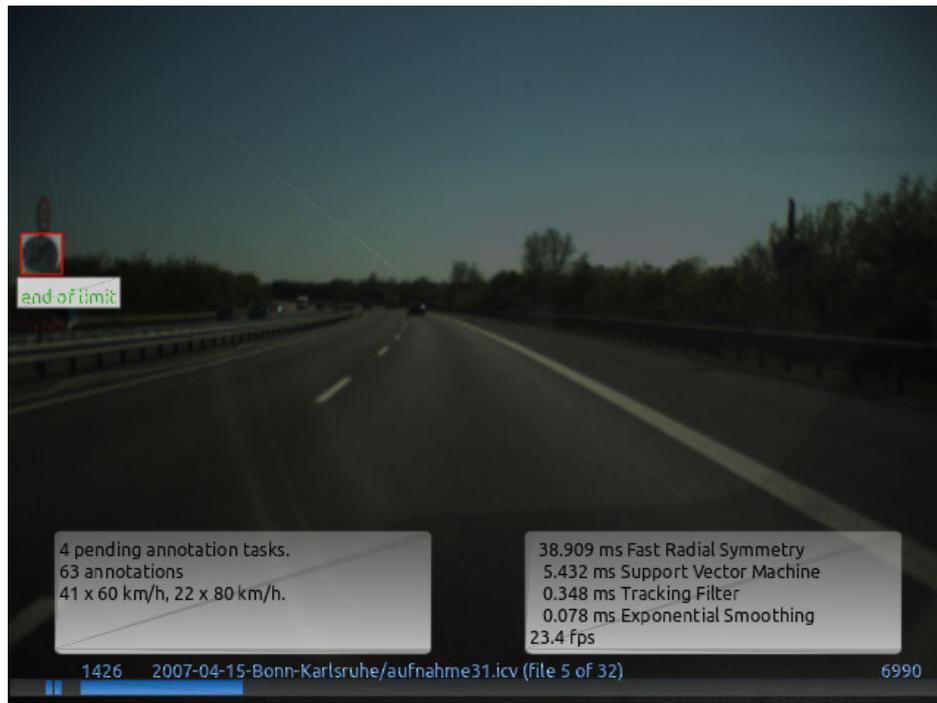


Abb. 5.11.: Benutzerschnittstelle zur Annotation. Durch die stetige Modellverbesserung während des aktiven Lernens werden auch die Annotationsvorschläge des Systems verbessert und dadurch der Zeitaufwand für Annotationen verkürzt.

onsexperten sinnvoll genutzt. Die in Abschnitt 3.3.7 entworfene Auswahlstrategie TPCB-ADR erzielt in den ersten Iterationen einen sehr starken Anstieg der Klassifikationsrate auf den Testdaten. Sie übertrifft sowohl die passive Strategie TPCB-RS als auch die aktive Strategie TPCB-US: Nach zwei Iterationen erreicht TPCB-ADR eine Klassifikationsrate von 99.04%. TPCB-US macht hier mit 98.01% etwa doppelt so viele Fehler und TPCB-RS mit 97.06% sogar drei mal so viele. Nach fünf Iterationen schließt TPCB-US mit 99.16% fast zu TPCB-ADR auf (99.3%), während die passive Strategie TPCB-RS nur 98.37% erreicht. Ein sehr ähnliches Bild ergibt sich bei der Analyse der Falschpositivitäten; TPCB-ADR übertrifft sowohl TPCB-RS als auch TPCB-RS. TPCB-ADR zeigt damit die besten Leistungskennwerte und bringt die gewünschten Eigenschaften einer in Bezug auf passive Lernstrategien höheren Lernrate und verbesserten Klassifikationsrate mit sich. Beide untersuchten aktiven Strategien erhöhen die Laufzeit des Trainings spürbar von circa 41 auf 58 bzw. 67 Minuten (erste Iteration, vergleiche Tabelle 5.4). Da die Auswahlstrategien keine Interaktion benötigen, wird allerdings nicht die Arbeitszeit des Annotationsexperten, sondern lediglich die Wartezeit erhöht. Wie Abbildung 5.10 entnommen werden kann, führt TPCB-ADR zu Modellen, die bei vergleichbaren Klassifikationsraten weniger Supportvektoren aufweisen und damit eine schnellere Klassifikation durchführen. Der Unterschied beträgt bei vergleichbaren Klassifikationsraten aber nur circa 10% und andere Methoden zur Laufzeitreduktion sind vorzuziehen (vergleiche Abschnitt 5.1). Der Trainingsprozess im passiven Lernen ist zweigeteilt bestehend aus einer langen Annotationsphase und einem abschließenden Training des Modells. Im aktiven Lernen ändert sich der Prozess stark dahingehend, dass nach einer kurzen Annotationsphase zur Erstellung einer initialen Trainingsmenge mehrere kurze (jeweils 2–3 Stunden) Iterationen folgen. In diesen Iterationen wird abwechselnd eine begrenzte Anzahl neuer Daten manuell annotiert und danach werden automatisch ein neues Modell trainiert und ein weiterer Batch zur Annotation erstellt.

	30	50	60	70	80	100	120	Aufhebung
erkannt	4	2	8	3	20	39	25	14
nicht erkannt			1		1			
Σ	4	2	9	3	21	39	25	14

Tab. 5.5.: Ergebnisse der Segmentierung. Die Detektionsrate beträgt 98.3%.

Je nachdem, wie die Abbruchkriterien definiert sind, müssen entweder genau so viele oder weniger Daten annotiert werden. Aufgrund der stetigen Modellverbesserung in jeder Iteration ist es dem System möglich, die Annotationsvorschläge für den Experten (vergleiche Abbildung 5.11) zu verbessern und dadurch die Annotationszeit zu verringern.

Evaluation des Gesamtsystems

Die Erkennungsrate des gesamten Systems hängt vom Zusammenspiel der einzelnen Komponenten ab. Nach der Evaluation der Klassifikation im vorherigen Abschnitt wird jetzt die Leistung der übrigen Komponenten und des Systems zur Erkennung von Ronden als Ganzes evaluiert. Dafür wird von der Betrachtung von ROIs als Evaluationseinheit gewechselt auf zusammengehörige Gruppen von Tempolimits. Eine Gruppe wird gebildet von einem oder mehreren Tempolimits, die zusammen aufgestellt sind und das gleiche Tempolimit anzeigen (beispielsweise bilden die beiden Tempolimits in Abbildung 5.12b eine Gruppe). Die Betrachtung von Schildgruppen ermöglicht es, die ermittelten Erkennungsraten auf die Fähigkeit des Systems zu übertragen, relevante Tempolimits zu erkennen. Da die Gesamtzahl evaluierter Objekte dadurch stark sinkt, werden die Testmengen A und B zusammen zur Evaluation verwendet, die insgesamt 117 Schildgruppen von Tempolimits enthalten.

Segmentierung Zur Evaluation der Segmentierung wird eine Schildgruppe als korrekt detektiert betrachtet, wenn mindestens eines der enthaltenen Tempolimits mindestens zwei Mal segmentiert wird. Tabelle 5.5 stellt die Ergebnisse zusammen. Mit 98.3% ist die Detektionsrate sehr hoch. Die beiden nicht detektierten Schildgruppen bestehen jeweils aus einem einzelnen Tempolimit, das mangels Kontrast oder aufgrund geringer Größe nicht von der radialen Symmetrie detektiert wird (vergleiche Abbildung 5.13a).

Tracking und temporale Fusion Tracking und temporale Fusion stabilisieren die Erkennungsergebnisse der Klassifikation. Eine Schildgruppe wird als korrekt erkannt gewertet, wenn die temporale Fusion die korrekte Klasse von mindestens einem der Tempolimits der Gruppe ermittelt. Werden unterschiedliche Tempolimits erkannt, wird dies als Fehler gewertet. Wie der Konfusionsmatrix 5.6 entnommen werden kann, ist die Klassifikationsrate mit 96.5% sehr hoch, fällt allerdings leicht schlechter aus als die Einzelbildklassifikation. Dies kann auf die Verwechslung der 50 km/h Schilder zurückgeführt werden, die in der Trainingsmenge unterrepräsentiert sind.

	30	50	60	70	80	100	120	Aufhebung
30	4							
50								
60			8					
70				3				
80		1			20			
100		1				38	1	
120							24	
Aufhebung								14
Hintergrund						1		
Σ	4	2	8	3	20	39	25	14

Tab. 5.6.: Konfusionsmatrix für die Klassifikation. Die Klassifikationsrate beträgt 96.5%.

	30	50	60	70	80	100	120	Aufhebung
30	4							
50								
60			8					
70				3				
80		1			20			
100		1				38	1	
120							24	
Aufhebung								14
Hintergrund			1		1	1		
Σ	4	2	9	3	21	39	25	14

Tab. 5.7.: Konfusionsmatrix für das Gesamtsystem. Die Erkennungsrate beträgt 94.9% und kann aufgrund der Verkettung der Komponenten aus den Einzelergebnissen in den Tabellen 5.5 und 5.6 berechnet werden.

Gesamtsystem Abschließend wird die Erkennungsrate des Gesamtsystems betrachtet. Eine Schildgruppe wird als erkannt gewertet, wenn das korrekte Tempolimit ermittelt wird. Die Konfusionsmatrix 5.7 stellt die Ergebnisse für die einzelnen Klassen zusammen. Die Erkennungsrate ist mit insgesamt 94.9% sehr hoch. Abbildung 5.12 zeigt einige Erkennungsergebnisse aus der Testmenge.

Zur Ermittlung der Falschpositivrate ist die Betrachtung aller Bilder der Testmenge notwendig, was deutlich mehr Annotationsaufwand bedeutet. Für diesen Teil der Auswertung wurde daher die kleinere Testmenge A verwendet. Bezogen auf ROIs und damit ohne Betrachtung von Tracking und temporalen Fusion ergeben sich hier 24 Falschpositive bei insgesamt 29546 Bildern. Bezogen auf Schildgruppen und unter Verwendung von Tracking und temporalen Fusion (exponentielle Glättung mit $\kappa_m = 0.85$) ergibt sich ein Falschpositiver. Der Großteil der falsch positiven ROIs wird also erfolgreich gefiltert.

Das System verarbeitet auf einem Intel Core i7-2820QM Prozessor mit 2.30 GHz 25–30 Bilder der Auflösung 640×480 pro Sekunde, wenn keine Objekte klassifiziert werden, und 20–25 Bilder pro Sekunde, wenn zusätzlich Tracking, Klassifikation und temporale Fusion aktiv sind.



Abb. 5.12.: Beispiele für korrekte Erkennungsergebnisse von Ronden.



(a) Das Tempolimit am rechten Bildrand wird von der Segmentierung nicht erkannt. (b) Klassifikation und temporale Fusion lehnen das Tempolimit ab.

Abb. 5.13.: Beispiele für Fehler bei der Erkennung von Ronden.



Abb. 5.14.: Eine Auswahl aus der Menge von Trainingsbeispielen für das Viola/Jones-Verfahren zur Erkennung dreieckiger Warnhinweise.

Parameter	Wert
Fenstergröße	19×21
Skalierungsfaktor	1.1
Anzahl Stufen	20
Minimale Trefferrate pro Stufe	0.995
Maximale Falschpositivrate pro Stufe	0.5

Tab. 5.8.: Parameter zum Training und zur Detektion mittels des Viola/Jones-Verfahrens zur Segmentierung von dreieckigen Warnhinweisen.

5.2. Erkennung von Warnhinweisen

Die Erkennung von Warnhinweisen wurde in der in Abschnitt 3.5.2 beschriebenen Variante mit Viola/Jones-Verfahren als Segmentierung umgesetzt. Zum Training der Kaskade wurden 4299 Beispiele für Warnhinweise sowie Hintergrunddaten annotiert, die aus 36 Fahrten einer Gesamtlänge von 10:25:59 (196606 Bilder) stammen (vergleiche Abbildung 5.14). Das Training der Viola/Jones-Kaskade erfolgte mit den in Tabelle 5.8 dargestellten Parametern. Vom Viola/Jones-Verfahren detektierte Dreiecke werden mit Hilfe einer SVM verifiziert und ihr Typ bestimmt. Als Merkmalstransformation kommt eine Grauwerttransformation zum Einsatz. Die Menge der zu unterscheidenden Klassen Ω wurde auf die fünf Typen dreieckiger Verkehrselemente beschränkt, zu denen in den Trainingsdaten ausreichend (mindestens 150) Trainingsbeispiele vorhanden waren: *Vorfahrt gewähren*, Warnhinweise *Radfahrer*, *Kinder* und *Baustelle* sowie Hinweis auf *Zebrastreifen*.

Das trainierte System erledigt wie in Abbildung 5.15 an Beispielszenen exemplarisch dargestellt die Erkennungsaufgabe sehr gut. Zur Evaluation des Gesamtsystems wurden die Erkennungsergebnisse auf einer Testfahrt der Länge 00:13:42 (9772 Bilder) durch die Karlsruher Innenstadt ausgewertet. Ein dreieckiges Verkehrszeichen wird als erkannt gewertet, wenn es in mindestens einem Bild korrekt erkannt wird. Von den 14 dreieckigen Verkehrszeichen im Testvideo werden alle erkannt. Es treten insgesamt zwei Falschpositive auf. Die Konfusionsmatrix 5.9 stellt die Ergebnisse der Klassifikation bezogen auf ROIs (ohne temporale Fusion) dar. Bei Verwendung einer Intel Core 2 Quad CPU Q8200 mit 2.33 GHz benötigt die Segmentierung durchschnittlich 60 ms zur Verarbeitung eines Bildes der Größe 1280×720 . Die Klassifikation benötigt pro Objekt durchschnittlich 1 ms. Daraus ergibt sich eine Verarbeitungsgeschwindigkeit des Gesamtsystems von 15 Bildern pro Sekunde.



Abb. 5.15.: Beispiele von Erkennungsergebnissen für dreieckige Verkehrszeichen in der Testfahrt.

	ω_0	Radfahrer	Kinder	Vorf. gew.	Baustelle	Zebrastrreifen
ω_0	376	1	20	20		10
Radfahrer	2	21	7			
Kinder	3	2	97			
Vorf. gew.	10			1539		
Baustelle	2				19	
Zebrastrreifen	1					134

Tab. 5.9.: Konfusionsmatrix zur Klassifikation von ROIs dreieckiger Warnhinweise. Die Klassifikationsrate ist 96.6%.

Diskussion Die Segmentierung mittels Viola/Jones-Verfahren hat gegenüber modellbasierten Ansätzen den Nachteil, dass Position und Größe weniger genau ermittelt werden und Rotationen unbekannt sind. Durch die Verwendung einer Kamera mit vergleichsweise hoher Auflösung von 1280×720 Bildpunkten und einer ausreichend großen Trainingsmenge ist es aber möglich, trotzdem sehr gute Detektionsergebnisse zu erhalten: Alle Verkehrszeichen in der Testmenge werden stabil über längere Zeiträume erkannt. Die Klassifikationsrate ist mit 96.4% sehr hoch. Hierbei muss jedoch beachtet werden, dass die vergleichsweise einfach zu erkennende Klasse *Vorfahrt gewähren* einen ausgeprägt hohen Anteil in den Testdaten hat, da entsprechende Verkehrszeichen besonders häufig aufgestellt werden. Insbesondere das Verkehrszeichen *Kinder* wird nur zu 78.2% erkannt. Es ist aber davon auszugehen, dass durch eine größere Menge von Trainingsdaten diese Werte deutlich gesteigert würden. Die Beschränkung der Klassenanzahl für die Klassifikation mittels SVM ist keine konzeptuelle Einschränkung: Um weitere Typen zu unterscheiden, muss lediglich die Trainingsmenge entsprechend erweitert werden und gegebenenfalls die in Abschnitt 5.1 evaluierten Methoden zur Parameterselektion und zum aktiven Lernen angewendet werden (siehe auch Abbildung 5.16).

5.3. Erkennung von Vorwegweisern

Zur Evaluation des in Abschnitt 3.5.3 beschriebenen Systems zur Erkennung von Vorwegweisern wurden Autobahnfahrten unter unterschiedlichen Bedingungen aufgezeichnet und in eine Trainings- und eine Testmenge aufgeteilt [XNZ09]. Die Testmenge enthält 15 Vorbeifahrten an Vorwegweisern, die jeweils eine Länge von 200–300 Bildern haben. Die Farbklassifikation wurde mit $h_{\min} = 200$ und $h_{\max} = 230$ parametrisiert. Zur Texterkennung wurde die OCR-Software *Tesseract* [Tes] eingesetzt. Die Ergebnisse der Texterkennung werden über ein Wörterbuch gemäß der minimalen Levenshtein-Distanz korrigiert.

Die erste Segmentierungsstufe zeigt sich unter den getesteten Bedingungen (trockenes Wetter, leichter Nebel, leichter Regen, vergleiche Abbildung 5.17) als robust und erkennt von den 20 Vorwegweisern im Testvideo 19 mindestens einmal. In der zweiten Segmentierungsstufe werden von den 66 insgesamt enthaltenen Textzeilen 58 mindestens einmal segmentiert und der Text nach Wörterbuchkorrektur richtig erkannt. Von den enthaltenen 6 Sinnbildern wurden 5 mindestens einmal korrekt erkannt. Bezogen auf ROIs entspricht das einer Klassifikationsrate von 87.9% (vergleiche Tabelle 5.10).

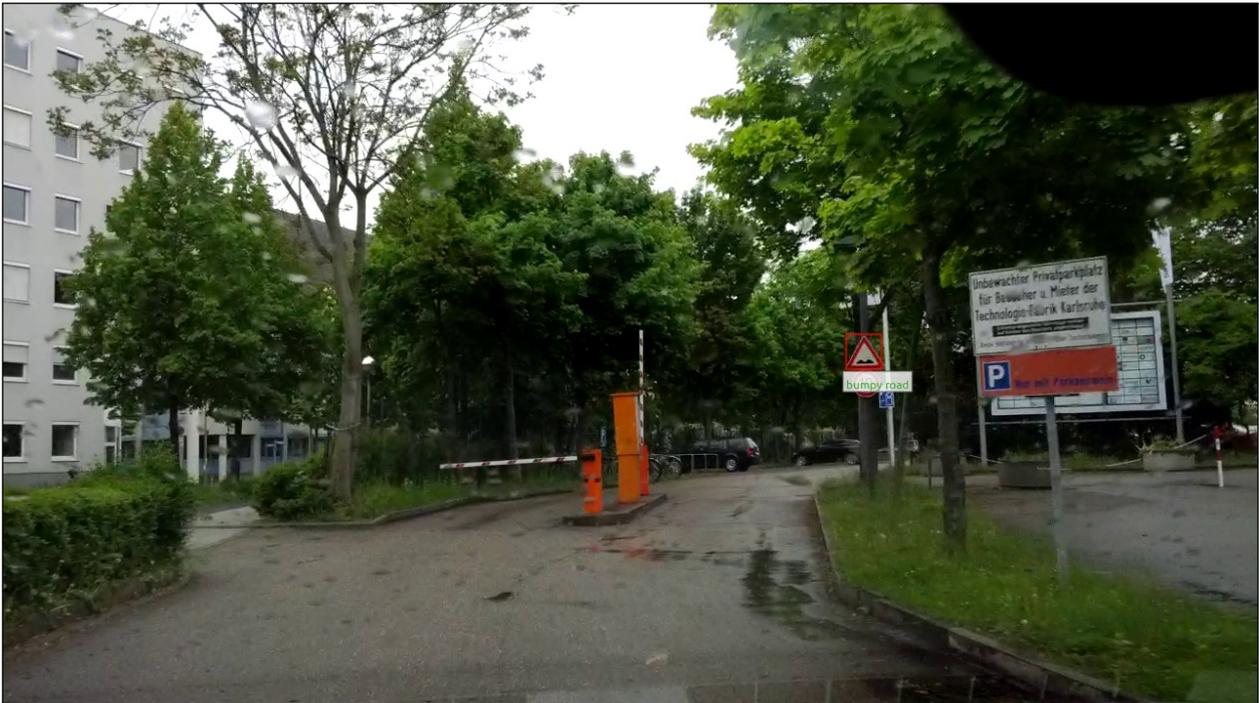


Abb. 5.16.: Die Erweiterung des Systems um weitere Arten von Warnhinweisen ist leicht möglich, indem zusätzliche Trainingsdaten für die SVM bereitgestellt werden.



Abb. 5.17.: Ergebnisse der ersten Segmentierungsstufe. Die Vorwegweiser in den Bildern (a) bis (e) und (h) werden wie gewünscht erkannt. Aufgrund des starken Gegenlichts schlägt in Bild (f) die Segmentierung fehl. Der Vorwegweiser in Bild (g) wird aufgrund seiner bei dieser Entfernung geringen Größe im Bild noch nicht erkannt. Die Bilder (g) bis (i) zeigen zudem jeweils Falschpositive hervorgerufen durch andere großflächige blaue Objekte. Durch den gewählten Ansatz ist dies zu erwarten; diese Objekte müssen von der Klassifikation abgelehnt werden.

	Kreuz	Ausfahrt	Knotenpunkt	sonstige	ω_0
Kreuz	83.3 %			12.5 %	4.2 %
Ausfahrt		92.3 %		7.7 %	
Knotenpunkt			76.1 %	22.8 %	1.1 %
sonstige				100 %	

Tab. 5.10.: Konfusionsmatrix für die Klassifikation von Sinnbildern auf Vorwegweisern.

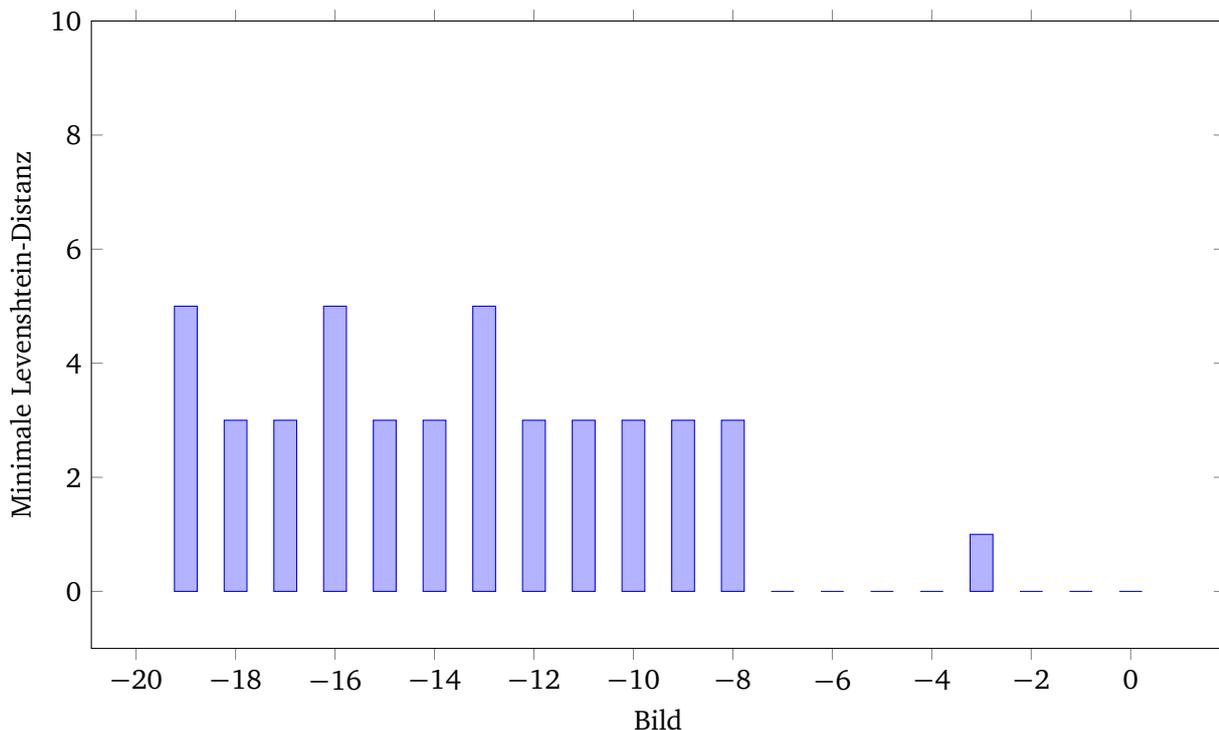


Abb. 5.18.: Bei der Annäherung an einen Vorwegweiser steigt die Größe der extrahierten Textregionen. Die höhere Auflösung wirkt sich positiv auf die Erkennungsleistung der OCR-Software aus, erkennbar an der sinkenden minimalen Levenshtein-Distanz zwischen dem erkannten Text und dem Wörterbuch.

Diskussion Während die erste Segmentierungsstufe gute Ergebnisse erzielt, bleibt die zweite Stufe etwas hinter den Erwartungen zurück. Fehler in der Symbolklassifikation können auf eine zu geringe Menge von Trainingsdaten zurückgeführt werden. Wie in Abschnitt 5.1 gezeigt wurde, werden für vergleichbare Daten deutlich höhere Klassifikationsraten erzielt, wenn die Trainingsmenge ausreichend groß ist. Bei der Texterkennung stellt sich die Auflösung der Kamera (640×480 Bildpunkte) als limitierender Faktor heraus. Dies lässt sich gut an den Werten der minimalen Levenshtein-Distanz bei der Vorbeifahrt an einem Vorwegweiser beobachten (Abbildung 5.18). Tatsächlich ist die Größe der an die OCR-Software übergebenen ROIs selbst unmittelbar während der Vorbeifahrt noch unterhalb der von Tesseract empfohlenen Minimalgröße.

Die Evaluation zeigt, dass das Konzept an sich funktionstüchtig ist. Um eine Leistung zu erzielen, die einen Einsatz als Assistenzsystem ermöglicht, ist allerdings die Verwendung einer höheren Auflösung sowie eine größere Trainingsmenge für die SVM notwendig.

5.4. Erkennung von Zusatzzeichen

Die Erkennung von Zusatzzeichen wird wie in Abschnitt 3.5 beschrieben in der Umgebung anderer Verkehrszeichen ausgeführt. Die Implementierung nutzt dazu die Erkennung von Ronden, um unterhalb erkannter Tempolimits und Überholverbote Zusatzzeichen zu detektieren (Abbildung 5.19). Eine zweistufige hierarchische SVM klassifiziert die Ergebnisse. Dabei wird in der ersten Stufe eine Validierung durchgeführt. Positiv klassifizierte Kandidaten werden weiter unterschieden in die Klassen



(a) Segmentierungsergebnis eines einzelnen Zusatzzeichens unterhalb eines Tempolimits.



(b) Segmentierungsergebnis für zwei direkt übereinander angebrachte Zusatzzeichen.

Abb. 5.19.: Ergebnisse der Segmentierung von Zusatzzeichen dargestellt als umgebendes Rechteck. Die farbigen Linien markieren Positionen der ROIs in vorhergehenden Bildern.

	ω_0	Lkw	Pkw	bei Nässe	7.5 t
ω_0	369	2			1
Lkw	1	65			1
Pkw	3	2	15		
bei Nässe	5				39
7.5 t	2				20

Tab. 5.11.: Konfusionsmatrix zur Evaluation der Erkennung von Zusatzzeichen. Die Klassifikationsrate ist 96.8%.

$\Omega_p = \{\text{Lkw, Pkw, bei Nässe, 7.5 t}\}$:

Stufe	Klassen	Merkmalstransformation
1	ω_0, Ω_p	Hybride Merkmalstransformation mit Grauwerttransformation und blockweiser Histogramm-Transformation (RGB-Farbraum)
2	$\omega \in \Omega_p$	Grauwerttransformation

Die erste Stufe verwendet ein Farbhistogramm, um die Unterscheidung zwischen den häufig farbigen Hintergrundobjekten und achromatischen Zusatzzeichen zu erleichtern. In der zweiten Stufe hingegen ist die Farbinformation kein typisches Merkmal der Klassen und dementsprechend wird eine Grauwerttransformation benutzt. Die Menge der Klassen entspricht den Typen von Zusatzzeichen, für die ausreichend Trainingsdaten zum Lernen eines SVM-Modells zur Verfügung standen. Insgesamt wurden 1633 ROIs manuell annotiert und aufgeteilt in eine Trainingsmenge der Größe 1108 und eine Testmenge mit den übrigen 525 ROIs. Wie der Konfusionsmatrix 5.11 entnommen werden kann, erzielt die trainierte hierarchische SVM eine Klassifikationsrate von 96.8%. Nutzt man in beiden Stufen die hybride Merkmalstransformation, sinkt die Klassifikationsrate auf 96%. Wenn nur eine Grauwerttransformation benutzt wird, sinkt die Klassifikationsrate auf 95.8%. Dies bestätigt die Erwartung, dass in der ersten Stufe Farbinformationen nützlich und in der zweiten Stufe redundant sind.

Abhängig von der Größe der Suchregion (die von der Größe und der Anzahl detektierter Tempolimits und Überholverbote bestimmt wird) beansprucht die Segmentierung eine Laufzeit zwischen 7 und 35 ms auf einem Intel Core 2 Quad Prozessor Q8200 mit einer Taktfrequenz von 2.33 GHz bei der Verwendung eines Prozessor-Kerns. Merkmalstransformation und Klassifikation benötigen pro Kandidat zusammen 4.4 ms. Insgesamt ergibt sich eine Verarbeitungsgeschwindigkeit von mehr als 25 Bildern pro Sekunde.

Diskussion Durch die Kombination robuster Verfahren – Hough-Transformation und SVMs – wird eine zuverlässige Erkennung von Zusatzzeichen erreicht, die aufgrund der starken Einschränkung der Suchregion auch die Echtzeitanforderung erfüllt. Um mehr als die hier untersuchten vier Klassen von Zusatzzeichen zu unterscheiden, müsste in erster Linie die Menge der Trainingsdaten erhöht werden. Um auch textbasierte Zusatzzeichen mit geringer Schriftgröße zuverlässig unterscheiden zu können, ist allerdings eine höhere Auflösung als die in diesem Experiment verwendeten 640×480 Bildpunkte notwendig (vergleiche Abbildung 5.19).

	FN	FP	TP
TopHat und Farbklassifikation	184	15219	2683
TopHat, Farbklassifikation und regelbasierte Validierung	187	6001	2680

Tab. 5.12.: Ergebnisse der Segmentierung von Ampeln.

5.5. Erkennung von Ampeln

Zur Erkennung von Ampeln wurde die in Abschnitt 3.5.5 beschriebene Prozesskette implementiert und Experimente mit Kameras mit Auflösungen von 512×384 und 1280×720 Bildpunkten durchgeführt.

Zur Evaluation der Segmentierung wurden manuell in vier Testvideos der Kamera mit Auflösung 512×384 ROIs annotiert. Wie in Abschnitt 3.5.5 vorgeschlagen ist die Farbklassifikation schwach filternd parametrisiert, so dass Falschnegative vermieden werden auf Kosten einer höheren Falschpositivrate. Dies wird von der Evaluation (Tabelle 5.12) bestätigt: Nach Durchführung von TopHat-Operator und Farbklassifikation sind 84 % der ROIs Falschpositive. Die anschließende regelbasierte Validierung filtert 60 % davon, ohne einen spürbaren Fehler einzubringen. Damit liegt die Falschpositivrate am Ende der Segmentierung bei 68 %. Von den tatsächlichen Ampellichtern werden 93 % detektiert.

Die Parameter der HMMs zur temporalen Fusion wurden mit Hilfe des Baum-Welch-Algorithmus trainiert. Dazu wurden Beobachtungssequenzen als Trainingsbeispiele extrahiert. Die ermittelten Übergangs- und Beobachtungswahrscheinlichkeiten sind in Abbildung 3.25 angegeben. Aus den Modellen kann die erwartete Aufenthaltsdauer in den einzelnen Zuständen berechnet werden. Beispielsweise ergibt sich für den gelb/roten Zustand des HMM λ_1 (dreifarbiges Ampel) ein Erwartungswert von $1/(1 - 0.982) = 55.6$ Zeiteinheiten und für den gelben Zustand $1/(1 - 0.991) = 111.1$. Dividiert durch die Frequenz der Kamera von 30Hz ergibt sich näherungsweise die tatsächliche Zustandslänge von zwei beziehungsweise vier Sekunden. Für die Zustände rot und grün ergeben sich kürzere Erwartungswerte als die tatsächlichen Zustandslängen, da diese Zustände bedingt durch die Anfahrt an die Ampel und das Überqueren der Kreuzung nicht komplett beobachtet werden.

Zur Evaluation der Klassifikation und des Gesamtsystems wurden Aufnahmen der Kamera mit Auflösung von 1280×720 Bildpunkten verwendet. Aus einer Trainingsmenge von zehn Fahrten wurden etwa 2000 manuell annotierte ROIs zum Training der SVM extrahiert. Die Evaluation erfolgt auf einer weiteren, zur Trainingsmenge disjunkten Fahrt durch die Karlsruher Innenstadt einer Länge von neun Minuten (8222 Bilder), die insgesamt 42 Ampeln für Fahrzeuge enthält. Zur Parametrierung der Merkmalstransformation wurden verschiedene Kombinationen von RGB- und HSV-Transformation (mit Größennormalisierung auf 6×21 Bildpunkte), blockweisem Farbhistogramm und Kantenhistogramm verglichen. Die höchste Klassifikationsrate erzielte die hybride Merkmalstransformation bestehend aus RGB-Transformation und Kantenhistogramm mit 95.3 % (siehe Konfusionsmatrix 5.14). Auf den ersten Blick überrascht das schlechtere Abschneiden des blockweisen HSV-Farbhistogramms. Es kann über Segmentierungsungenauigkeiten erklärt werden: Wenn die Bestimmung der Größe oder Position der Ampel ungenau ist, werden Hintergrund- und Ampelfarben im Histogramm vermischt. Wie der Konfusionsmatrix 5.14 entnommen werden kann, werden Hintergrundobjekte zuverlässig abgelehnt.

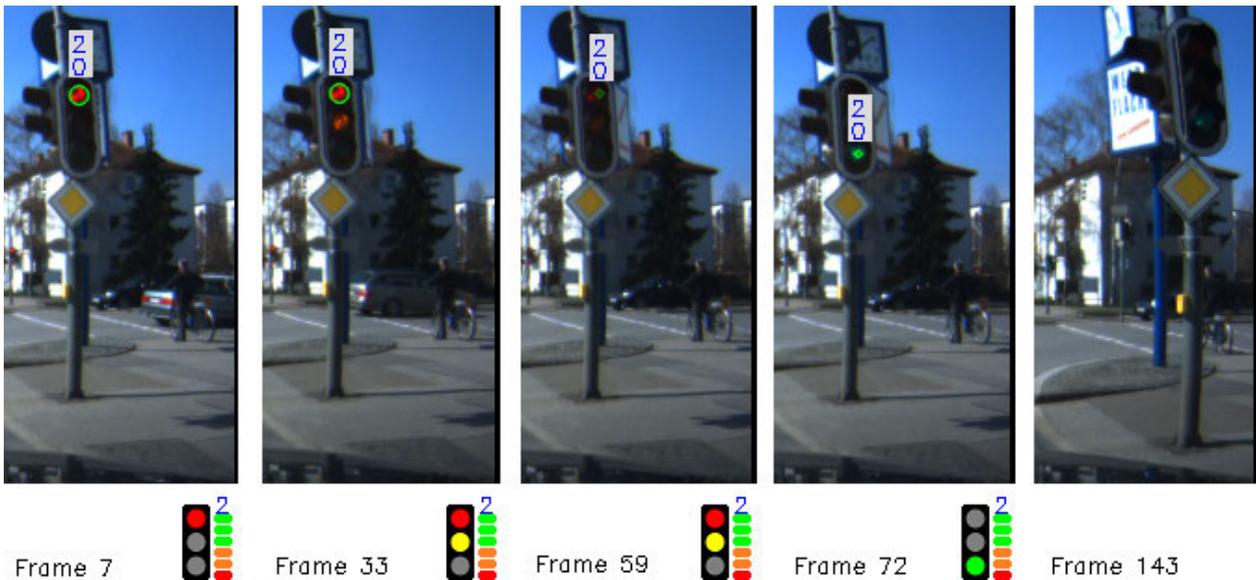


Abb. 5.20.: Verfolgung eines Zustandswechsels und Kompensation kurzzeitiger Störungen durch die temporale Fusion mittels HMMs.

Merkmalstransformation	Klassifikationsrate
A	94.6%
B	91.4%
A+C	95.3%
B+C	90.3%
A+B+C	93.9%

Tab. 5.13.: Klassifikationsraten verschiedener hybrider Merkmalstransformationen für die Klassifikation von Ampeln. Hier bezeichnet A eine RGB-Transformation, B ein HSV-Farbhistogramm und C ein Kantenhistogramm. Die übrigen, nicht aufgeführten Kombinationen erreichten weniger als 90%.

Zwischen grüner und roter Ampelfarbe gibt es keine Verwechslungen.

Die Testfahrt enthält 16 Ampeln ohne Pfeil, 10 mit Pfeil *geradeaus*, 12 mit Pfeil *links*, 3 mit Pfeil *rechts* und 1 mit Pfeil *geradeaus und links*. Alle dieser Ampeltypen werden von der Erkennung unterstützt, aber nicht in verschiedene Klassen unterschieden. In der Auswertung des Gesamtsystems wird daher analog zu [CN09] eine Ampel als korrekt erkannt gewertet, wenn sie mindestens einmal erkannt und ihre Farbe richtig bestimmt wird. Das System erkennt 41 der 42 Ampeln korrekt (97.6%). Dabei kommt es zu keiner falsch positiven Ampelerkennung, sämtliche sporadischen Fehlerkennungen werden erfolgreich von der temporalen Fusion abgelehnt.

	Hintergrund	grün	rot
Hintergrund	146	5	8
grün		73	
rot			47

Tab. 5.14.: Konfusionsmatrix zur Klassifikation von ROIs im Testvideo.

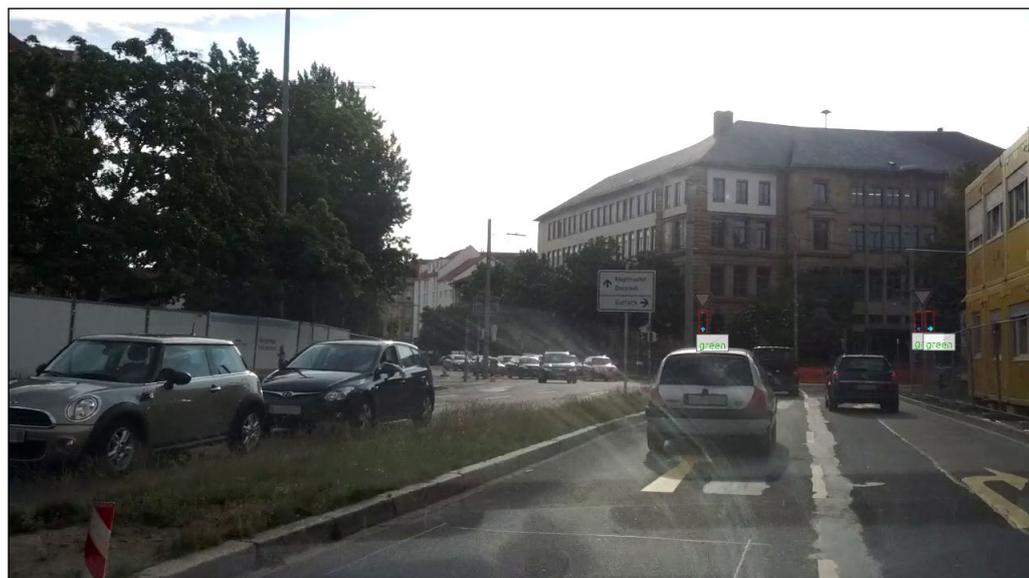
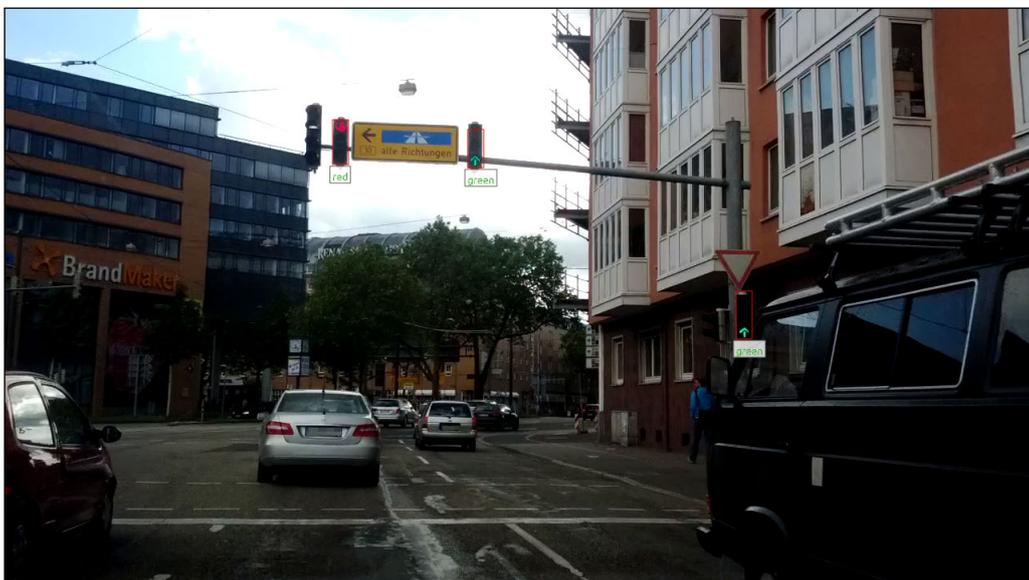


Abb. 5.21.: Ampeln mit beliebigen Lichtern (rund oder pfeilförmig) werden erkannt. Durch den Einsatz von Bildpyramiden wird eine effiziente Erkennung unabhängig von der Distanz (Größe im Bild) erreicht.



(a) Mangelnder Kontrast erschwert die Bestimmung des Ampelkastens.



(b) Kurze Belichtungszeiten hervorgerufen durch starkes Gegenlicht führen in diesem Beispiel dazu, dass nur die roten Ampeln der beiden rechten Spuren erkannt werden.



(c) Nachts ist die Bestimmung des Ampelkastens mangels Kontrast nicht möglich. Überstrahlung der Lichter und Bewegungsunschärfe erschweren seine Größenschätzung.

Abb. 5.22.: Die Erkennung von Ampeln kann auch bei ungünstigen Sichtverhältnissen durchgeführt werden. Es ist jedoch mit einer geringeren Erkennungsleistung zu rechnen.



Abb. 5.23.: Bei günstigen Sichtbedingungen ist die Erkennung auf große Distanzen möglich. Experimente mit einem Teleobjektiv ergaben Erkennungsweiten bis zu 950 m. Um hohe Falschpositivraten zu vermeiden, sollten derartige Ergebnisse aber nicht verwendet werden, sondern eine Mindestgröße für Kandidaten im Bild festgelegt werden.



Abb. 5.24.: Die Ampel am rechten Bildrand wird in der Evaluation als falsch negativ gewertet, da sie zwar in einigen Bildern korrekt erkannt, aber von der temporalen Fusion abgelehnt wird.

Im Schnitt verarbeitet das System 25 Bilder pro Sekunde auf einem Intel Core i7-2820QM Prozessor mit einer Taktfrequenz von 2.30 GHz bei Nutzung aller vier Prozessorkerne. Der Hauptanteil der Laufzeit wird mit 20–40 ms von der Segmentierung beansprucht. Die Klassifikation benötigt 0.8 ms pro ROI. Die Laufzeit der übrigen Komponenten ist zu vernachlässigen.

5.6. Semi-automatische Kartierung mit einem Segway-Personentransporter

Die Dempster/Shafer-Fusion (Abschnitt 4.3) benutzt Kartendaten, deren Gültigkeit zur Laufzeit bewertet wird, um das gültige Tempolimit zu bestimmen. Sie ist darauf ausgelegt, ohne Interaktion mit dem Fahrer korrekte Ergebnisse zu liefern. Die von ihr festgestellten Konflikte können auf Fehler in Kartendaten hinweisen. Damit liefern sie eine interessante Information für die Wartung von Karten, die typischerweise manuell von Firmen oder Freiwilligen in Projekten wie OpenStreetMap [Ope] erledigt wird. Allgemein ist es möglich, mit Hilfe der Umfelderkennung fehlerhafte und fehlende statische Verkehrselemente in Kartendaten zu erkennen. In diesem Experiment wird untersucht, inwiefern sich die im Rahmen dieser Arbeit entstandenen Algorithmen dazu eignen, ein semi-automatisches kognitives Assistenzsystem für die Kartierung von statischen Verkehrselementen zu realisieren [Web+11]. Ein solches Assistenzsystem kann den Zeitaufwand für die Wartung digitaler Karten signifikant senken. Beliebige, von der Perzeption erkannte Verkehrselemente werden an die korrekte Stelle in die Karte eingetragen. Konflikte müssen auch hier erkannt werden, um das doppelte Eintragen von Verkehrselementen zu verhindern. Die Ergebnisse werden in die OpenStreetMap-Datenbank übernommen und zur Qualitätssicherung vorher von einem menschlichen Benutzer kontrolliert. Es handelt sich also um eine semi-automatische Kartierung, bei der ein Mensch mit einem Segway-Personentransporter (Abbildung 5.25) die zu kartierenden Gebiete



(a) Frontansicht des Gesamtsystems



(b) Mensch-Maschine-Schnittstelle



(c) Sensorik (Kamera und GPS-Empfänger)

Abb. 5.25.: Ein für die semi-automatische Kartierung ausgerüsteter Segway-Personentransporter.

abfährt. Am Ende der Fahrt kontrolliert er die vom System erstellten Vorschläge und gibt sie zum Eintrag in die Karte frei. Die Wahl eines Segway-Personentransporters ist exemplarisch anzusehen; der benutzte Ansatz könnte ebenso mit einem anderen Verkehrsmittel (Fahrzeug, Fahrrad) oder von einem Fußgänger mit einem Smartphone verwendet werden. Derartige Systeme können auch die Grundlage zur Aufbereitung von Karten für kognitive Automobile darstellen, die für das autonome Fahren relevante Daten automatisch oder semi-automatisch übernehmen.

Das System verwendet die in Abschnitt 3.5.1 vorgestellte Erkennung von Ronden. Die Klassifikation ist mit typischerweise auftretenden Verkehrszeichen wie Tempolimits, Halteverboten, Rad- und Fußgängerwegen trainiert. Anhand der Kamerakalibrierung wird die Position eines erkannten und verfolgten Verkehrszeichens im Fahrzeugkoordinatensystem geschätzt (vergleiche Abschnitt 4.1) und über die per GPS-Sensor ermittelte Eigenposition ins Weltkoordinatensystem transformiert. In der lokalen Umgebung der geschätzten Position wird die Distanz zu allen bereits in den Kartendaten verzeichneten Verkehrszeichen berechnet. Anhand der Distanzen, den Straßenseiten und dem Typ der Verkehrszeichen wird entschieden, ob es sich um ein bereits kartiertes Verkehrszeichen, ein neues Verkehrszeichen oder einen

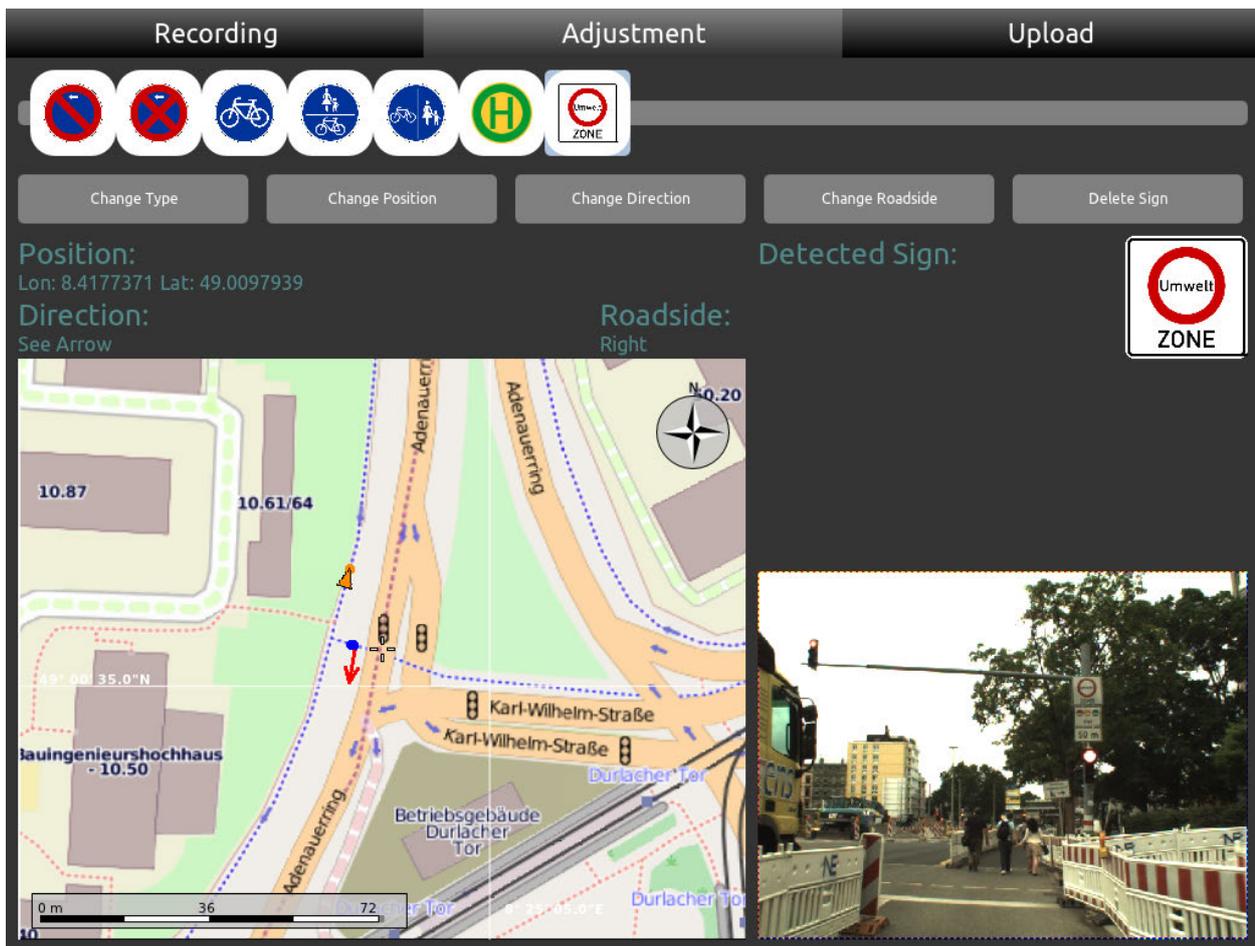


Abb. 5.26.: MMS zur semi-automatischen Kartierung mit Funktionen zur automatischen Sammlung neuer und zu korrigierender Verkehrselemente (*Recording*), manueller Korrektur der Systemvorschläge (*Adjustment*) und zum Eintragen in die OpenStreetMap-Datenbank (*Upload*).

Konflikt handelt. Noch nicht kartierte Verkehrszeichen werden inklusive weiterer Informationen wie dem aktuellen Videobild als Systemvorschlag gespeichert, den der Benutzer am Ende der Fahrt (gegebenenfalls korrigiert) in die Datenbank übernehmen kann. Abbildung 5.26 zeigt die Benutzerschnittstelle, die diesen Ablauf unterstützt.

Zur Evaluation des Systems wurden die beiden in Abbildung 5.27 dargestellten Versuchsfahrten A und B ausgewertet. Beide führen über Rad- und Fußgängerwege sowie Straßen der Karlsruher Innenstadt. Sie unterscheiden sich vornehmlich durch die Qualität der GPS-Positionierung, die für Fahrt A stabil und für Fahrt B ungenau mit häufigen Verlusten der Position ist. Zur Bewertung der Systemleistung werden die folgenden Kriterien benutzt:

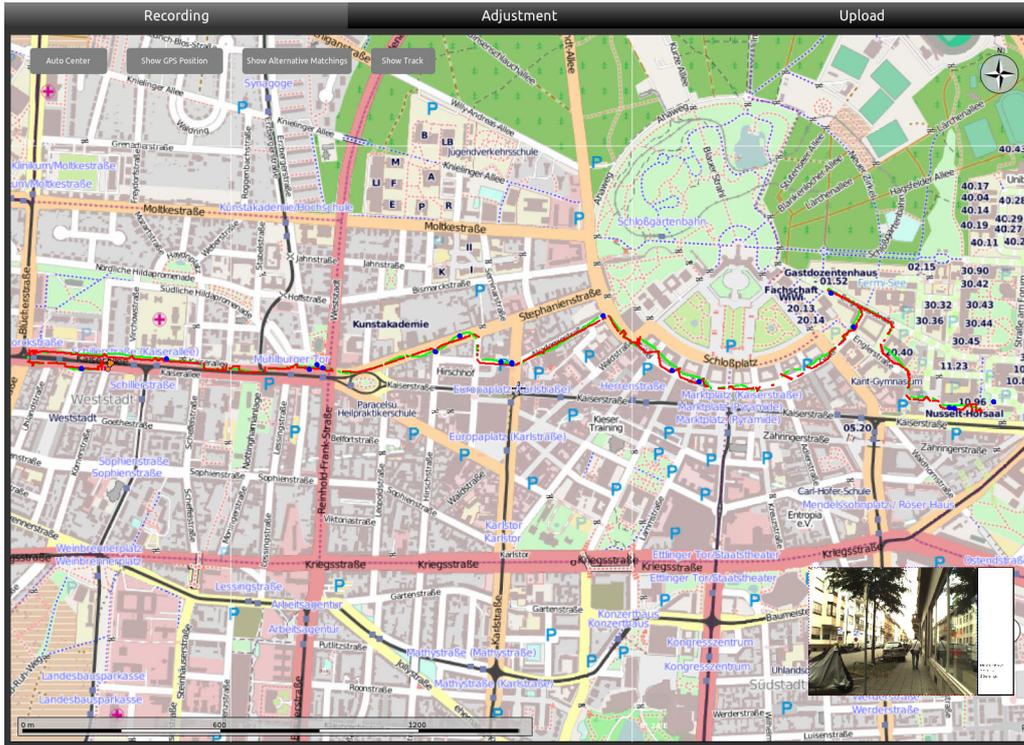
Kriterium	Beschreibung
Position	Die Differenz zwischen der automatisch ermittelten Position eines Verkehrselements und seiner tatsächlichen Position beträgt nicht mehr als 10 m.
Straße	Die ermittelte Straße ist korrekt.
Seite	Die ermittelte Straßenseite ist korrekt.
Übernahme	Alle Informationen (Typ des Verkehrselements, Position, Straße, Seite) sind korrekt, so dass der Benutzer das Verkehrselement ohne Korrekturen in OpenStreetMap übernehmen kann.

In Fahrt A wurden 16 und in Fahrt B 17 Verkehrszeichen kartiert. Die Auswertung der Kriterien ist in Abbildung 5.28 dargestellt. Die stabilere Positionierung in Fahrt A resultiert wie zu erwarten in besseren Werten in allen Kriterien. Der Großteil der Parameter der Systemvorschläge muss vom Benutzer nicht korrigiert werden; über 60% aller Verkehrszeichen können sogar direkt übernommen werden. Selbst für Fahrt B mit instabiler Positionierung kann noch fast jedes dritte Verkehrszeichen direkt übernommen werden. Der Großteil der bereits kartierten Verkehrszeichen wird korrekt als solche erkannt und nicht erneut zum Eintragen vorgeschlagen. Fehler entstehen vornehmlich durch ungenaue Positionsschätzungen, wenn die absolute GPS-Position oder die Ausrichtung des Fahrzeugs ungenau ist und dadurch Fehler in der Karteneinpassung entstehen. Solche Ungenauigkeiten sind im untersuchten Einsatzgebiet nicht zu vermeiden – in Innenstädten mit engen Straßen und hohen Häusern entstehen Signalreflexionen und nur wenige GPS-Satelliten sind frei sichtbar. Abhilfe könnte hier das Tracking der Position unter Zuhilfenahme von Odometrie-Daten des Fahrzeugs oder visueller Odometrie bringen. Eine weitere Fehlerquelle sind im Kartenmaterial fehlende oder fehlerhaft verzeichnete Straßen. Sie führen zu Fehlern in der Karteneinpassung.

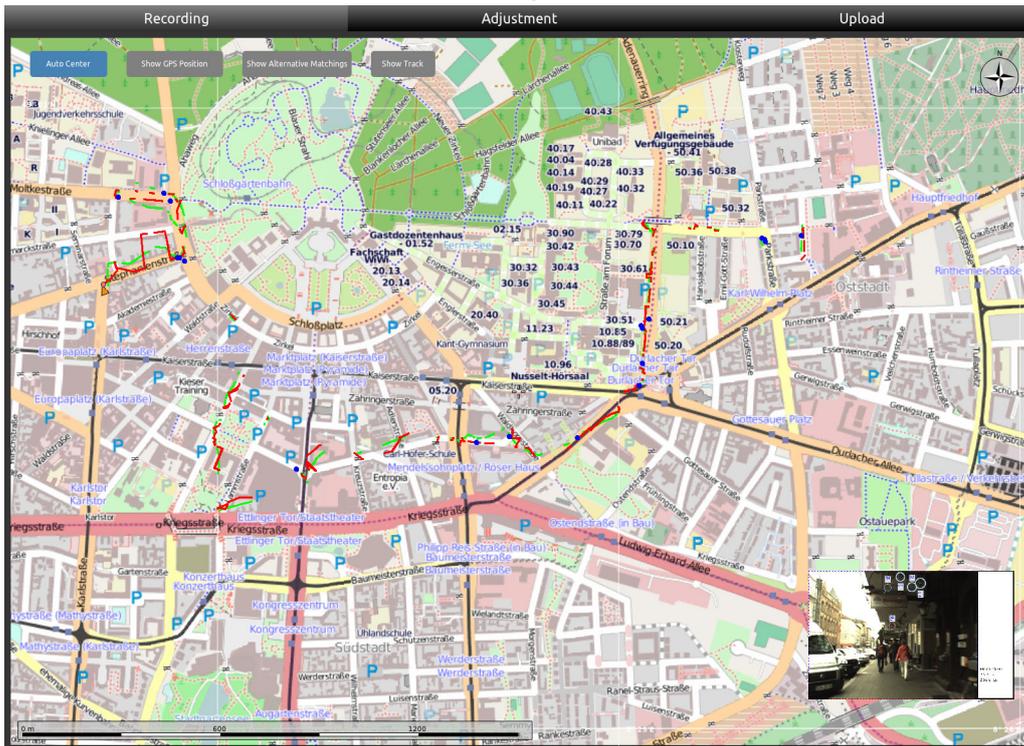
Diskussion Insgesamt ergibt sich für den Benutzer eine sehr gut geeignete Kartierungshilfe, die den Großteil der sonst manuell durchzuführenden Schritte automatisch erledigt. Durch die Sammlung von Vorschlägen und Kontrolle zu beliebigen Zeitpunkten wird die Kartierung vom Ort der Verkehrszeichen entkoppelt, wodurch sich die Arbeitsweise stark vereinfacht: Anstatt bei jedem neuen Verkehrszeichen anhalten zu müssen, um es in die Karte zu übernehmen, werden die Daten vom System aufgenommen, vorgehalten und können zu einem beliebigen späteren Zeitpunkt kontrolliert und übernommen werden. Durch Verbesserungen der Lokalisierung könnte die Robustheit gesteigert werden und durch die Erkennung weiterer Objektarten wie etwa Fahrflächen eine Generalisierung der Kartierungsfunktionen erreicht werden. Derartige Erweiterungen würden jedoch über den Rahmen dieser Arbeit hinausgehen.

5.7. Verkehrszeichenerkennung auf Mobilgeräten

Mobilgeräte erlebten in den vergangenen Jahren starke Leistungssteigerungen in Bezug auf ihre Hardware. Zu ihrer Standardausstattung gehören mittlerweile eine oder mehrere Kameras, GPS-Sensor und Lagesensoren. Damit eignen sie sich prinzipiell auch zur Ausführung der im Rahmen dieser Arbeit



(a) Stabile Positionierung in Versuchsfahrt (A).



(b) Ungenaue Positionierung in Versuchsfahrt (B).

Abb. 5.27.: Zwei Versuchsfahrten mit unterschiedlicher Qualität der GPS-Positionierung.

- GPS Position ● Verkehrselement
- - - Eingepasste Position

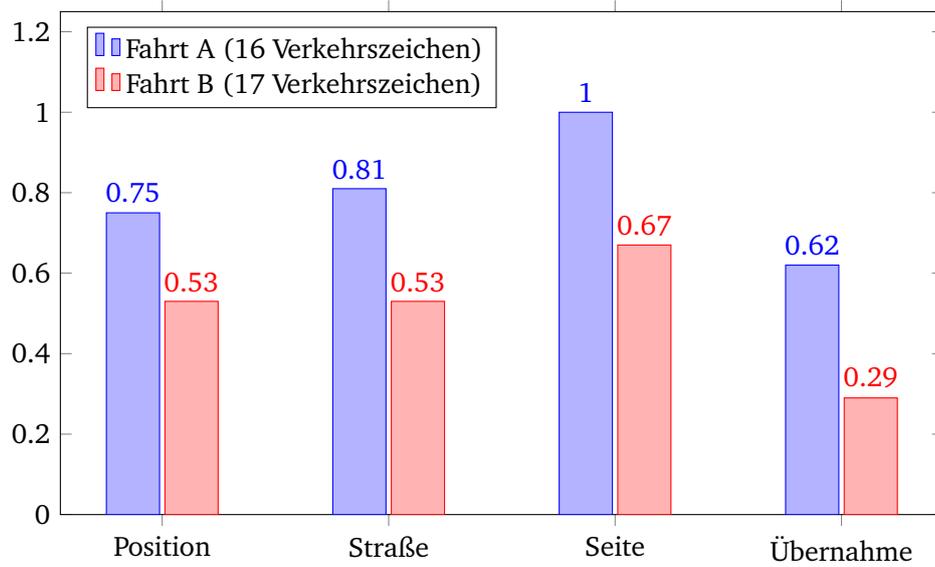


Abb. 5.28.: Evaluationsergebnisse der beiden Versuchsfahrten (siehe Text).



Abb. 5.29.: Typische Eingabedaten für das Assistenzsystem auf dem Android-Smartphone.



(a) Anbringung des Smartphones im Fahrzeug. (b) Nullpunktbestimmung während der Kalibrierung.

Abb. 5.30.: Einrichtung des Smartphones als Assistenzsystem im Fahrzeug.

entwickelten Methoden und Verfahren. Aus Anwendersicht ist eine derartige Umsetzung besonders attraktiv, da der Anschaffungspreis der Hardware deutlich geringer ausfällt als fest in einem Serienfahrzeug verbaute Lösungen. Als Anwendungsgebiet kommen für Mobilgeräte einerseits Assistenzsysteme in Frage, andererseits aber auch Kartierungslösungen. Letzteres ist vor dem Hintergrund der starken Verbreitung von Mobilgeräten auch für sogenannte Crowdsourcing-Ansätze interessant.

Um zu evaluieren, ob sich die entwickelten Verfahren für den Einsatz auf aktuellen Mobilgeräten eignen, wurden prototypisch zwei Perzeptionsverfahren – die Erkennung von Tempolimits und die Erkennung von Ampeln – auf einem Samsung Galaxy Nexus i9250 Smartphone umgesetzt. Es besitzt einen Texas Instruments OMAP 4460 ARM Cortex-A9 Zweikern-Prozessor, eine Kamera mit einer Auflösung von 1280×720 Bildpunkten, GPS-Sensor, Gyroskop und Beschleunigungssensor. Als Betriebssystem wird Android 4.0 eingesetzt.

Für die Positionsschätzung von Objekten werden die extrinsischen Kalibrierungsparameter (die Lage der Smartphone-Kamera in Bezug auf das Fahrzeugkoordinatensystem) benötigt. Um dem gewählten Anwendungsfall – eine einfach zu bedienende Smartphone-App – Rechnung zu tragen, wird auf ihre genaue Vermessung verzichtet und stattdessen ein Kalibrierungsmodus integriert. Seine Hauptaufgabe ist, den Gierwinkel zu bestimmen, da die Lagesensoren des Smartphones dafür keine Anhaltspunkte liefern. Roll- und Nickwinkel können dagegen von den Lagesensoren übernommen werden. Als Orientierungshilfe für den Benutzer wird im Kalibrierungsmodus ein künstlicher Horizont eingeblendet, anhand dessen das Smartphone auf die Straße (Fahrtrichtung) ausgerichtet wird (Abbildung 5.30). Nach der so erfolgten Nullpunktbestimmung kann der Benutzer das Smartphone in die von ihm gewünschte Position drehen und die Winkelabweichung vom Nullpunkt wird automatisch bestimmt. Der Abstand zum Ursprung des Fahrzeugkoordinatensystems verlangt keine hohe Genauigkeit und kann daher ohne Benutzerinteraktion geschätzt werden.

Im Kalibrierungsmodus werden relative Winkel benutzt, die von den Lagesensoren des Smartphones geliefert werden. Um deren Genauigkeit abzuschätzen, wurde der in Abbildung 5.31 skizzierte Versuchsaufbau eingesetzt, mit dem die Ausgaben der Lagesensoren bei Winkeländerungen um 15, 30, 45 und 60 Grad mehrfach gemessen wurden [MNZ12]. Für die drei Drehachsen ergaben sich dabei die in Tabelle 5.15 dargestellten Abweichungen. Um die Funktionalität der Applikation zu gewährleisten, muss die Kamera die Straßenszene beobachten. Für den Benutzer bedeutet das eine Einschränkung

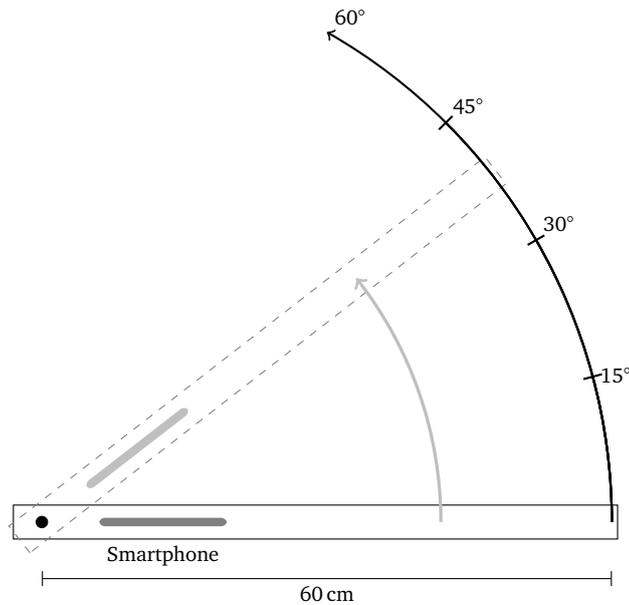


Abb. 5.31.: Versuchsaufbau zur Genauigkeitsmessung relativer Winkelangaben der Lagesensoren des Smartphones.

	15°	30°	45°	60°
Gierwinkel	0,3	1,4	4,2	6,2
Rollwinkel	1,2	0,9	1,0	1,6
Neigewinkel	1,7	2,7	3,4	5,2

Tab. 5.15.: Mittlere Abweichungen der von den Lagesensoren bestimmten relativen Winkel zur tatsächlich durchgeführten Rotation.

der maximal verwendbaren Gier- und Neigewinkel; Werte von 15°, maximal 30° können nicht überschritten werden. Die zu erwartenden Kalibrierungsungenauigkeiten durch falsche Drehwinkelausgaben beschränken sich daher auf wenige Grad. Damit sind sie geringer als die während der Fahrt dynamisch auftretenden Änderungen des Nickwinkels durch Bodenunebenheiten, die von den entwickelten Verfahren toleriert werden.

Zum Einsatz als Assistenzsystem auf dem Smartphone ist die Online-Verarbeitung der Daten notwendig. Die Auflösung der Kamera wird auf 640×480 Bildpunkte reduziert, um die Verarbeitung zu beschleunigen. Zur Bewertung der Laufzeit wurden zwei typische Szenen – eine Kreuzung mit zwei sichtbaren Ampeln und die Vorbeifahrt an einem Tempolimit – ausgewählt und die Verarbeitungsgeschwindigkeit des Systems gemessen. Die Ergebnisse finden sich in Tabelle 5.16. Die Segmentierung umfasst dabei die initiale Suche nach ROIs. Im Tracking wird die Verfolgung sowie eine Postsegmentierung durchgeführt, was den vergleichsweise hohen Aufwand hierfür erklärt. Die Klassifikation enthält neben der Klassifikation auch die temporale Fusion der Daten.

Diskussion Die relativen Laufzeiten der einzelnen Komponenten entsprechen in ihrer Größenordnung wie erwartet den Laufzeitanteilen des analogen Systems auf einem Desktop-System. Mit einer Verarbeitungsgeschwindigkeit von drei beziehungsweise sieben Bildern pro Sekunde auf einem Prozessorkern

	Tempolimit	Ampelkreuzung
Segmentierung	114 ms	45 ms
Tracking	53 ms	60 ms
Klassifikation	142 ms	26 ms
Gesamtsystem	318 ms	142 ms

Tab. 5.16.: Laufzeit des Gesamtsystems und einzelner Komponenten bei der Verarbeitung typischer Verkehrsszenen. Die Angaben beziehen sich auf die durchschnittliche Verarbeitungszeit pro Bild.

liegt die Echtzeitverarbeitung in greifbarer Nähe. Dazu könnten die in Abschnitt 5.1 untersuchten Laufzeitoptimierungen für die Klassifikation eingesetzt werden. Auch die Nutzung mehrerer Prozessorkerne oder der GPU könnte zukünftig zu einer starken Beschleunigung des Systems eingesetzt werden.

5.8. Probabilistische Zuordnung von Verkehrszeichen zu Fahrspuren

Verkehrselemente, die über einer Fahrspur angebracht sind, gelten nur für diese Spur. An den Fahrbahnrandern aufgestellt betreffen sie alle Fahrspuren. Um die Leistungsfähigkeit des in Abschnitt 4.1 vorgestellten probabilistischen Ansatzes zur Zuordnung von Verkehrszeichen zu Fahrspuren zu evaluieren, werden an dieser Stelle die ermittelten Zuordnungen auf einem circa 5 Minuten langen Video (8756 Einzelbilder) ausgewertet [NGZ08]. Dieser Datensatz wurde auf einem zwei- bis vierspurigen Teilstück einer Autobahn in der Nähe von Frankfurt aufgenommen, auf dem mit Verkehrsbeeinflussungsanlagen die Geschwindigkeit geregelt wird. Dafür kommen Wechselverkehrszeichen in LED-Technik zum Einsatz (vergleiche Abbildung 5.32).

Von den 46 auf diesem Abschnitt aufgestellten Tempolimits wurden 43 von der Perzeption korrekt erkannt. Die übrigen drei wurden von der Segmentierung nicht erfasst. Bei 38 Verkehrszeichen handelt es sich um Wechselverkehrszeichen. Die Zuordnung zwischen Verkehrszeichen und Fahrspur wurde insgesamt 627 Mal ausgeführt. Bei 263 dieser Zuordnungen war das Verkehrszeichen mehr als 60 m vom Fahrzeug entfernt und damit außerhalb der konfigurierten Reichweite der Fahrspurerkennung. Die Zuordnung gelang trotzdem in 74.9 % der Fälle (197 von 263) durch Extrapolation. Innerhalb der Reichweite der Fahrspurerkennung waren 92 % (335 von 364) aller Zuordnungen korrekt. Die Verteilung dieser Fehler in Abhängigkeit von der Distanz der beobachteten Verkehrszeichen ist in Abbildung 5.33 dargestellt. Kurz vor und bei der Vorbeifahrt treten keine Fehler mehr auf.

Diskussion Innerhalb der Sichtweite des Verfahrens wird mit 92 % korrekten Zuordnungen ein sehr gutes Ergebnis erzielt. Insbesondere treten auf kurzen Distanzen keine Fehler auf, so dass beim Passieren von Verkehrszeichen immer die korrekte Zuordnung zur Fahrspur bekannt ist. Dies kann genutzt werden, um zum Beispiel bei späteren Fahrspurwechseln auf das dort gültige Tempolimit hinzuweisen. Darüber hinaus eignet sich das Verfahren sehr gut als Grundlage für weitere Auswertungen wie der Relevanzbestimmung (siehe Abschnitt 4.3).

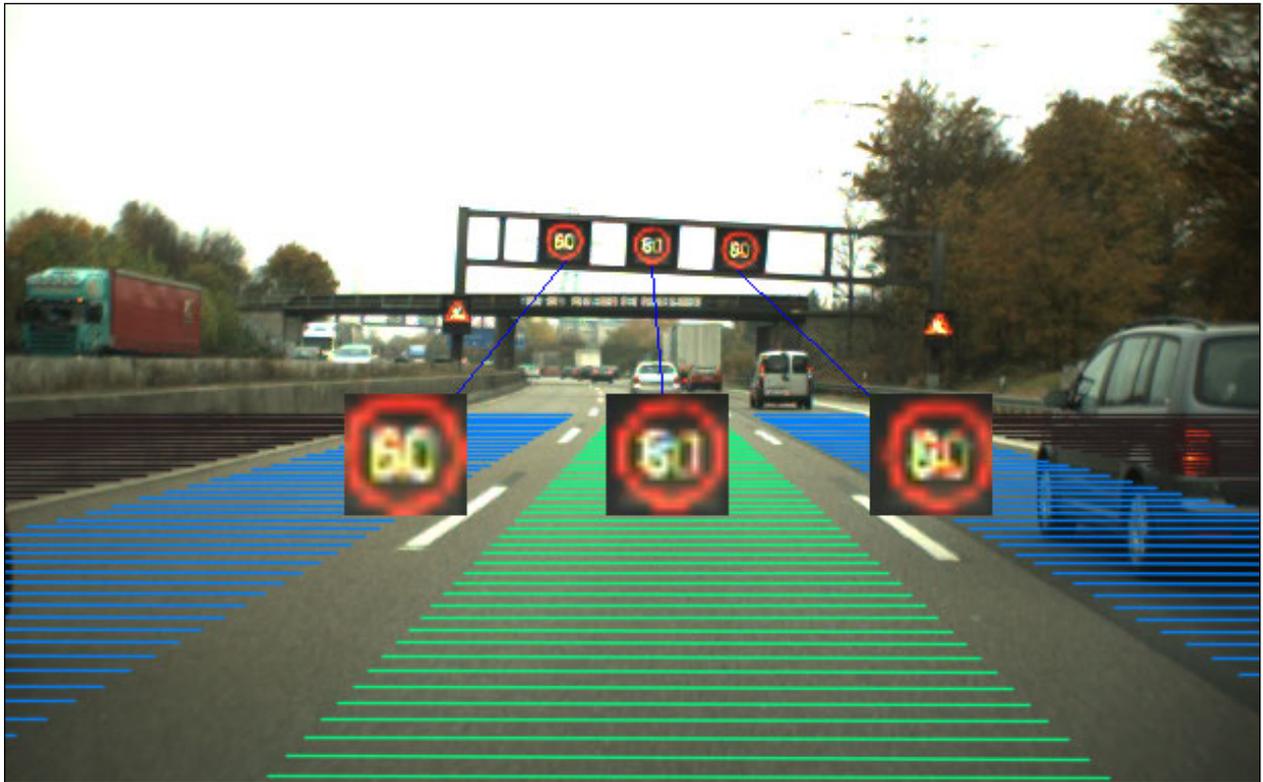


Abb. 5.32.: Mit Hilfe der probabilistischen Zuordnung von Verkehrszeichen zu Fahrspuren wird das für einzelne Fahrspuren gültige Tempolimit ermittelt.

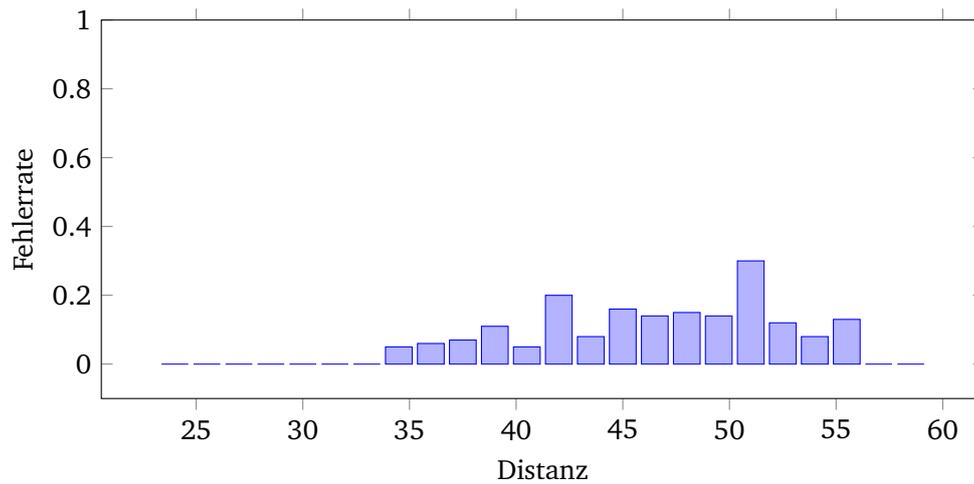


Abb. 5.33.: Distanzabhängige Fehlerrate bei der Zuordnung von Verkehrszeichen zu Fahrspuren.

5.9. Adaptive Dempster/Shafer-Fusion zur Bestimmung des Tempolimits

Mit Hilfe der in Abschnitt 4.2 vorgestellten Dempster/Shafer-Fusion kann das aktuell gültige Tempolimit durch eine Fusion von Perzeption und Kartendaten bestimmt werden. Um die Funktionalität des Ansatzes zu überprüfen, wurden folgende Methoden zum Aufbau des Situationskontexts verwendet:

Information	Methode
Tempolimit Perzeption	Erkennung von Ronden
Tempolimit Karte	GPS, OpenStreetMap
Wetter	SVM mit Klassen <i>Trocken, Nebel, Regen</i> [WNZ08]
Baustelle	Erkennung von Leitbaken [GNL09]
Dynamische Anzeigen	Perzeption
Umgebungslicht	Perzeption / Kamera

Die Parameter $\kappa_{S,p}$, $\kappa_{S,m}$ und κ_A zur Adaption der Massefunktionen an den Situationskontext wurden empirisch bestimmt und gemäß der in Abbildung 5.34 dargestellten Entscheidungsbäume festgelegt.

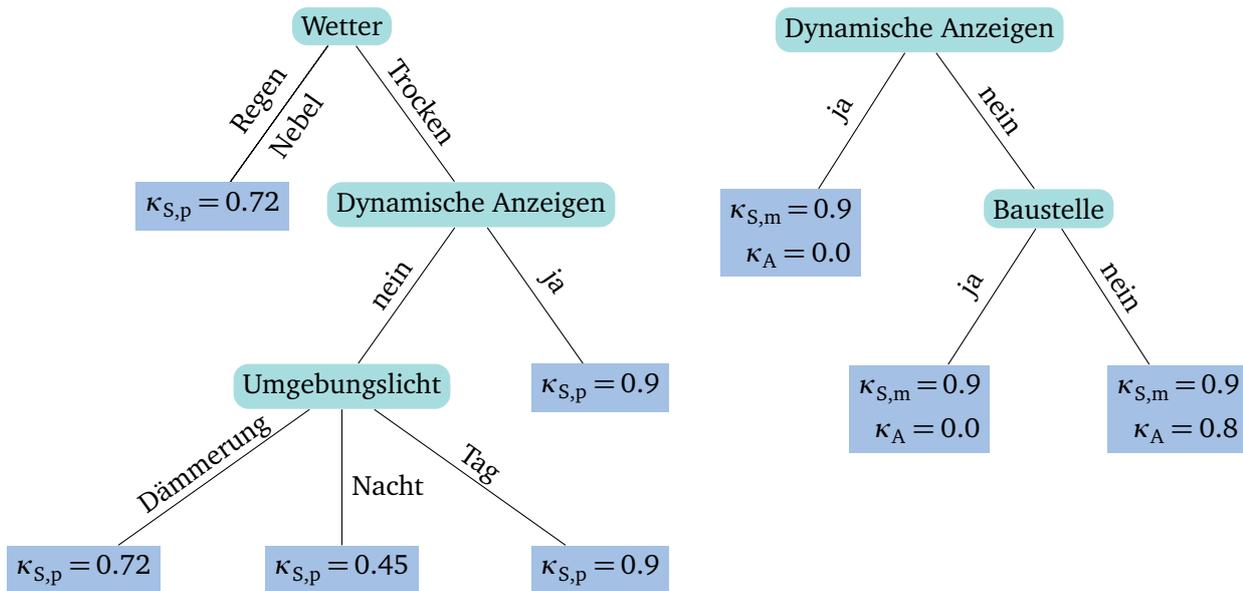
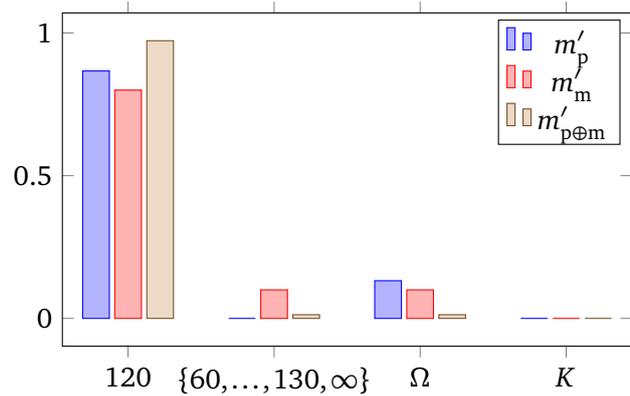


Abb. 5.34.: Entscheidungsbäume zur Bestimmung des Parameters $\kappa_{S,p}$ zur Adaption der Massefunktion der Perzeption an den Situationskontext (links) und der Parameter $\kappa_{S,m}$ und κ_A zur Adaption der Massefunktion des Kartendienstes an den Situationskontext (rechts).

Damit sind alle notwendigen Informationen für die Berechnung der nach Dempster/Shafer fusionierten, an den Situationskontext adaptierten Massefunktion $m'_{p \oplus m}$ gegeben. Im den Abbildungen 5.35–5.38 wird eine qualitative Evaluation des Verfahrens durchgeführt, in der die Ergebnisse der Fusion in vier unterschiedlichen Situationen betrachtet werden.



(a) Ein Tempolimit von 120 km/h bei guter Sicht.

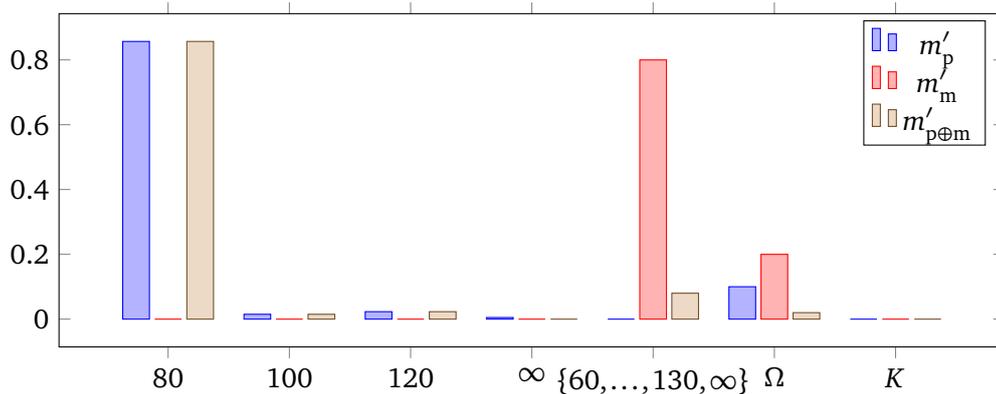


(b) Ergebnisse der Dempster/Shaffer-Fusion.

Abb. 5.35.: Situation A. Die Perzeption erkennt ein Tempolimit von 120 km/h. Es wird keine Baustelle, kein Wechselverkehrszeichen und das Wetter als trocken erkannt. Das gleiche Tempolimit ist in den Kartendaten verzeichnet. Die Ergebnisse der Fusion zeigen keinen Konflikt an ($K = 0$). Der Massewert $m'_{p\oplus m}(120) = 0.973$ ist höher als die Massewerte der beiden Informationsquellen.



(a) Ein Tempolimit von 80 km/h, das durch Wechselverkehrszeichen dargestellt wird.

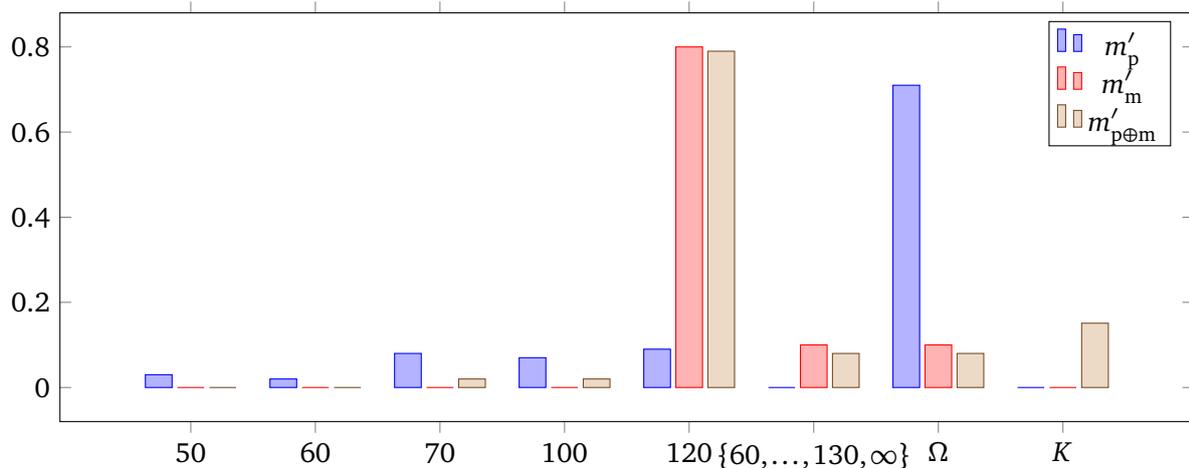


(b) Ergebnisse der Dempster/Shaffer-Fusion.

Abb. 5.36.: Situation B. Diese Situation unterscheidet sich von Situation A dadurch, dass Wechselverkehrszeichen erkannt werden. Das Wetter wird als trocken und es wird keine Baustelle erkannt. Aufgrund der Wechselverkehrszeichen enthält die Massefunktion für die Karte keine positiven Werte für einzelne Tempolimits. Auch hier geben die Ergebnisse der Fusion keinen Konflikt an ($K = 0$). Der Massewert $m'_{p\oplus m}(80) = 0.857$ ist identisch zu dem der Perzeption.



(a) Ein Tempolimit von 120 km/h bei Dunkelheit.

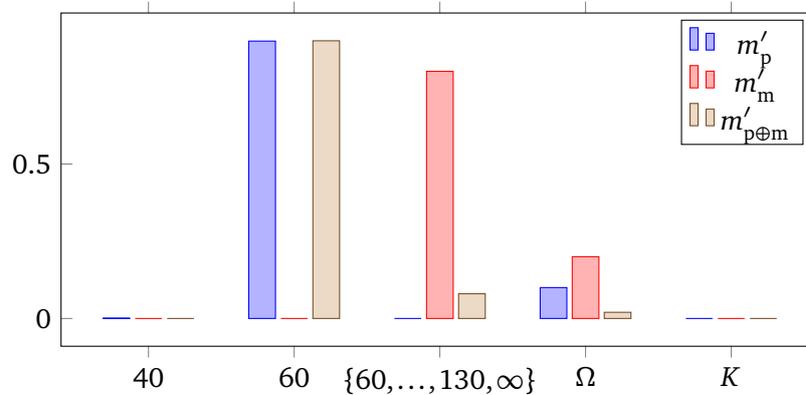


(b) Ergebnisse der Dempster/Shafter-Fusion.

Abb. 5.37.: Situation C. Bei Dunkelheit wird die Erkennung durch geringeren Kontrast und Bewegungsunschärfe erschwert. Dies äußert sich im wenig stark ausgeprägten Maximum für $m_p(120) = 0.09$. Aufgrund der schwachen Ausprägung würde dieses Ergebnis bei alleiniger Betrachtung der Perception als zu unsicher abgelehnt. Die Fusion ergibt folgende Werte: Der Massewert $m'_{p\oplus m}(120) = 0.79$ der Fusion ist demgegenüber ein ausgeprägtes Maximum bei geringem Konfliktwert und führt zur korrekten Bestimmung des Tempolimits.



(a) Ein Tempolimit von 60 km/h in einer Baustelle.



(b) Ergebnisse der Dempster/Shافر-Fusion.

Abb. 5.38.: Situation D. Es wird eine Baustelle erkannt. Die Perzeption erkennt mit hoher Sicherheit ein Tempolimit von 60 km/h. Das in der Karte verzeichnete Tempolimit von 120 km/h wird aufgrund der erkannten Baustelle nicht als positiver Wert in die Massefunktion m_p übernommen.

Diskussion Die Auswertung der vier Situationen zeigt die Funktionalität der Dempster/Shافر-Fusion und seine Fähigkeit, auch dann das korrekte Ergebnis zu berechnen, wenn eine der Informationsquellen für sich alleine betrachtet kein korrektes Ergebnis liefert. Die beiden letzten betrachteten Beispielsituationen zeigen die Vorteile des gewählten kontextsensitiven Ansatzes, der durch die Minimierung von Konflikten auch mit schwierigen Situationen umgehen kann. Eine Übertragung des Ansatzes auf andere Ländern sollte leicht möglich sein. Insbesondere müsste die Konfidenz für die Korrektheit der Kartendaten und die Abbildung von Straßenarten auf erlaubte Geschwindigkeiten an die nationalen Regelungen angepasst werden.

	keine Baustelle	Baustelle		keine Baustelle	Baustelle
keine Baustelle	1750	36	keine Baustelle	414	6
Baustelle	30	804	Baustelle	34	764

(a) Testvideo 1 (Klassifikationsrate 97.5 %). (b) Testvideo 2 (Klassifikationsrate 96.7 %).

Tab. 5.17.: Konfusionsmatrizen zur Erkennung von Baustellen auf zwei Videos (3838 Bilder). Die Klassifikationsrate liegt insgesamt bei 97.2%.

5.10. Markov-Logik-Netze

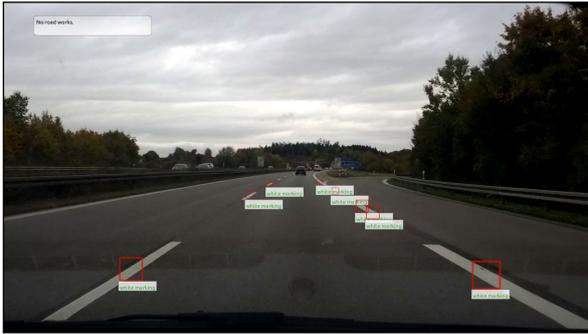
Erkennung von Baustellen

Per Inferenz auf dem in Abschnitt 4.3.4 vorgestellten Markov-Logik-Netz wird ermittelt, ob das Fahrzeug momentan eine Baustelle durchfährt. Zur Evaluation des Ansatzes wurden 18 Videos mit Baustellendurchfahrten ausgewählt. Davon werden 16 als Trainings- und 2 als Testmenge benutzt. Aus der Trainingsmenge wurden 426 Warnbaken als Trainingsbeispiele für den Viola/Jones Ansatz zur Erkennung von Warnbaken extrahiert, 635 Trainingsbeispiele für den Viola/Jones Ansatz zur Erkennung von Fahrspurmarkierungen sowie 327 Beispiele für gelbe und 245 Beispiele für weiße Markierungen zum Training der SVM. Nach dem Training der Umfelderkennung wurden in 403 Szenen aus der Trainingsmenge Evidenzen extrahiert und manuell das RoadWorks Prädikat annotiert, um die Gewichte des MLNs zu lernen. Abbildung 5.39 zeigt beispielhaft Ausschnitte aus den Testdaten mit eingezeichnetem Erkennungsergebnis. Die Ergebnisse der Evaluation sind in Tabelle 5.17 aufgelistet. Das Gesamtsystem verarbeitet durchschnittlich 7 Bilder der Größe 1280×720 pro Sekunde auf einem Intel Core 2 Quad Prozessor Q8200 mit einer Taktfrequenz von 2.33 GHz; der Großteil der Verarbeitungszeit wird für die beiden Viola/Jones-Kaskaden aufgewendet.

Diskussion Durch die Nutzung unterschiedlicher Informationen ergibt sich eine zuverlässige Erkennung, die in 97.2% der Bilder der Testmenge korrekte Ergebnisse liefert. Dank der Integration temporaler Informationen gelingt eine durchgängige Erkennung von Baustellen auch dann, wenn wenige Baustellenmerkmale (keine Warnbaken, gestrichelte gelbe Markierung) verfügbar sind. Erst, wenn die Erkennung dieser Merkmale mehrfach fehlschlägt, kommt es zu Fehlern (Abbildung 5.39 (e)). Der umgekehrte Fehlertyp – außerhalb einer Baustelle wird eine Baustelle erkannt – tritt auf, wenn fälschlich gelbe Markierungen detektiert werden oder wenn Baken erkannt werden, die nicht zur eigenen Spur gehören (Abbildung 5.39 (h)).

Bestimmung fahrrelevanter Ampeln

Um die Leistungsfähigkeit des in Abschnitt 4.3 vorgestellten Ansatzes zur Bestimmung relevanter Ampeln zu ermitteln, wurde ein Testdatensatz bestehend aus 12 Ampelkreuzungen zusammengestellt. Gemäß ihres Aufbaus werden die Ampelkreuzungen in fünf Kreuzungstypen unterschieden: Typ A weist vier Ampeln in zwei Gruppen, Typ B und D weisen drei Ampeln in zwei Gruppen und Typ C und E



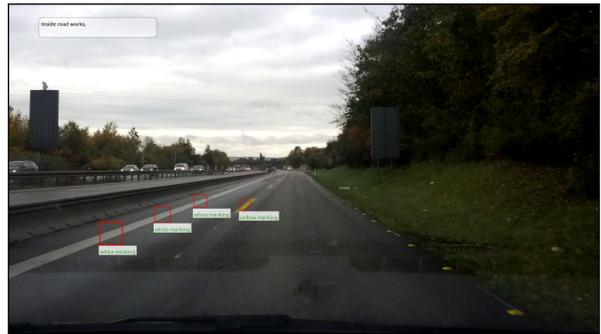
(a) Außerhalb von Baustellen treten typischerweise nur weiße Fahrspurmarkierungen auf.



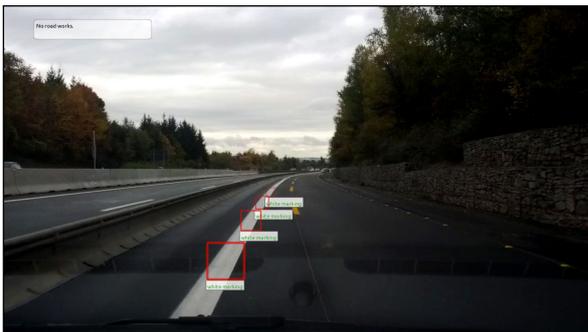
(b) Durchgezogene gelbe Markierungen und Warnbaken sind starke Indikatoren für Baustellen.



(c) Im Mittelteil von Baustellen nimmt die Anzahl der Merkmale häufig ab.



(d) Zur durchgängigen Erkennung spielen temporale Merkmale eine wichtige Rolle.



(e) Wenn über einen längeren Zeitraum zu wenige Merkmale gefunden werden, schlägt die Erkennung fehl.



(f) Das Ende von Baustellen ist nicht exakt festzulegen. In der Evaluation wird daher ein Toleranzbereich von 1 Sekunde verwendet.



(g) Am Ende dieser Baustelle ist auf der Gegenseite ebenfalls eine Baustelle vorhanden.



(h) Die Warnbaken der Gegenseite werden als Fortsetzung der Baustelle aus (g) interpretiert. Dies wird in der Evaluation als Fehler gewertet.

Abb. 5.39.: Auszüge der Erkennungsergebnisse von Baustellen aus den beiden Testvideos.

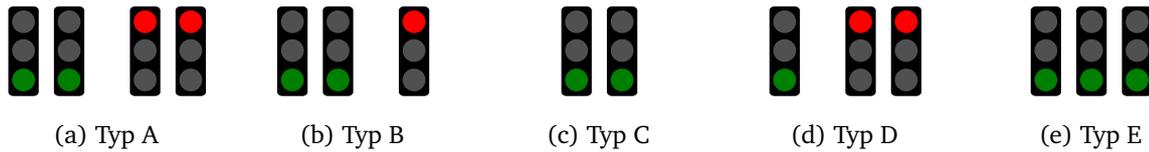


Abb. 5.40.: Verschiedene Konfigurationen von Ampelkreuzungen. Gleiche Farben symbolisieren die Zugehörigkeit zu einer Ampelgruppe (gleichzeitige Schaltung).

Typ	Szenen	Bilder	Korrekte Zuordnungen	Zuordnungsrate
A	2	2819	2801	99.4%
B	1	1819	1755	96.5%
C	3	708	708	100%
D	3	325	316	97.2%
E	3	2169	2169	100%

Tab. 5.18.: Ergebnisse der Zuordnung für die 12 untersuchten Kreuzungen.

weisen eine Gruppe mit zwei (C) bzw. drei (E) Ampeln auf. Die verschiedenen Konfigurationen sind in Abbildung 5.40 veranschaulicht. Für diese Kreuzungen wurde die aus Kartendaten zu extrahierende Evidenz (Ampelreihenfolge und -höhe, Zugehörigkeit zur Fahrroute) manuell erstellt und bereitgestellt. Sie dient zusammen mit den von der Perzeption automatisch extrahierten Informationen als Evidenz für das MLN. Mittels Inferenz wird in jedem Videobild eine Zuordnung zwischen den erkannten Ampeln und dem Kreuzungsmodell erstellt und die relevanten Ampeln ermittelt.

In der Evaluation wird ein Bild als korrekte Zuordnung gewertet, wenn alle ermittelten Zuordnungen zwischen sichtbaren Ampeln und Ampeln im Kreuzungsmodell korrekt sind (vergleiche Abbildung 5.41 und 5.42). Wie Tabelle 5.18 entnommen werden kann, ist die Fehlerrate sehr gering. Fehler treten in zwei Fällen auf: Wenn andere Ampeln in der Nähe erkannt werden, die nicht Teil des Kreuzungsmodells sind (typischerweise Fußgängerampeln und Warnlichter an Fußgängerübergängen) und gleichzeitig andere Ampeln nicht erkannt werden, kommt es zu fehlerhaften Zuordnungen. Wenn durch Segmentierungsfehler die geschätzte Höhe einer Ampel stark von seiner tatsächlichen Höhe abweicht, kann es ebenfalls zu falschen Zuordnungen kommen.

Diskussion Die Rate der korrekten Zuordnungen ist in allen Szenen unabhängig vom Kreuzungstyp sehr hoch. Die Kreuzungstypen C und E sind einfachere Zuordnungsprobleme, da es lediglich eine Ampelgruppe gibt und eine korrekte Zuordnung nur die Reihenfolge richtig bestimmen muss. Dementsprechend treten hier keine Fehler auf. Für die anderen, schwierigeren Kreuzungstypen ist die korrekte Zuordnungsrate mit mindestens 96.5% immer noch sehr hoch. Bei der Anfahrt des Fahrzeugs an eine Kreuzung werden typischerweise zuerst nicht alle Ampeln von der Perzeption erkannt. In diesem Fall können trotz korrekter Zuordnung mangels notwendiger Informationen nicht immer die relevanten Ampeln erkannt werden. Der gewählte Ansatz erkennt solche Situationen und führt in diesem Fall keine Zuordnung durch. Nähert sich das Fahrzeug der Kreuzung weit genug, um der Perzeption die Erkennung ausreichend vieler Ampeln zu ermöglichen, werden die relevanten Ampeln zuverlässig ermittelt.

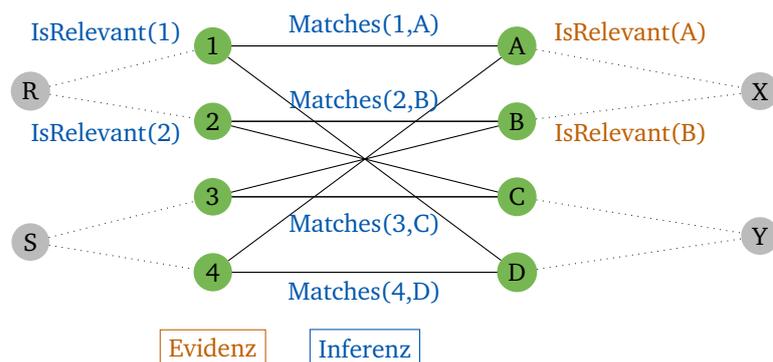


Abb. 5.41.: Die vier Ampeln für Fahrzeuge werden von der Perception erkannt. Die Inferenz erzeugt eine korrekte Zuordnung und ermittelt die richtigen Ampeln als relevant (im Bild hervorgehoben).

5.11. Zusammenfassung

Zur Evaluation der vorgestellten Konzepte und Methoden wurden insgesamt zehn Experimente durchgeführt. Sie untersuchen die Leistung der verschiedenen Anwendungen zur Umfelderkennung als kognitive Assistenzsysteme für Kraftfahrzeuge. Mit der semi-automatischen Kartierung wurde ein neuartiges Assistenzsystem auf einem Segway-Personentransporter erprobt. Auch die Anwendbarkeit der Verfahren auf einem Mobilgerät wurde untersucht. Schließlich wurden die Konzepte zur Situationsinterpretation durch Experimente zur Bestimmung der Fahrrelevanz mit Hilfe der probabilistischen Zuordnung von Verkehrszeichen zu Fahrspuren, der kontextsensitiven Dempster/Shafer-Fusion und MLNs validiert. Alle in den Experimenten verwendeten Anwendungen benutzen als primären Sensor eine Kamera, die in einigen Experimenten um Kartendienste und GPS-Sensoren ergänzt wird. Jede Anwendung verarbeitet mehrere Bilder pro Sekunde, so dass die Anforderungen an den Einsatz kostengünstiger Sensorik und

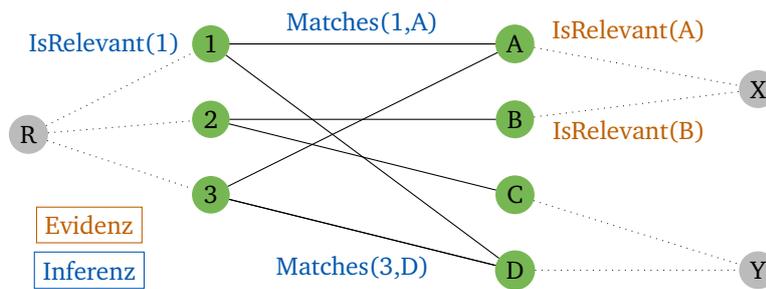


Abb. 5.42.: Drei der vier Ampeln werden von der Perzeption erkannt. Die beiden äußeren werden korrekt zugeordnet und die Relevanz der linken wird inferiert. Da nur eine der beiden mittleren Ampeln erkannt wird, ist ihre Zuordnung mehrdeutig.

eine echtzeitnahe Verarbeitung erfüllt ist.

Durch die Verwendung einer modularen Prozesskette ist es möglich, Komponenten wie die hierarchische SVM, das Tracking und die temporale Fusion in unterschiedlichen Anwendungen der Umfelderkennung einzusetzen. Damit konnten funktionsfähige Anwendungen zur Erkennung von Ronden, Warnhinweisen, Vorwegweisern, Zusatzzeichen und Ampeln demonstriert werden. Besonders umfangreiche Experimente wurden zur Validierung der Erkennung von Ronden durchgeführt, die bezogen auf ROIs Klassifikationsraten bis zu 99.3% erreicht und als Gesamtsystem in mehrstündigen Testfahrten 94.9% der Tempolimits erkennt. Dabei werden über 20 Bilder pro Sekunde verarbeitet bei einer Auflösung von 640×480 Bildpunkten. Die Erkennung von Ampeln erreicht sogar eine Verarbeitungsgeschwindigkeit von durchschnittlich 25 Bildern pro Sekunde bei einer Auflösung von 1280×720 Bildpunkten und erkennt 97.6% der Ampeln in der Testmenge ohne Fehlalarm. Die Evaluation der hierarchischen SVM angewendet auf die Erkennung von Tempolimits zeigt eine Verdopplung der Klassifikationsgeschwindigkeit bei nahezu gleicher Klassifikationsrate verglichen mit einer SVM mit *one-against-one* Mehrklassenentscheidung.

Ihr Einsatz ist immer dann anzuraten, wenn viele Klassen zu unterscheiden sind, die in Gruppen mit ähnlichen Merkmalen zusammengefasst werden können. Zur weiteren Optimierung der Klassifikation für sehr große Datensätze wurde das aktive Lernen und insbesondere die TPCB-ADR Auswahlstrategie evaluiert. Sie zeigt hervorragende Ergebnisse und führt zu einer deutlichen Steigerung der Lernrate verglichen mit passivem Lernen: Nach zwei Iterationen mit jeweils 1000 Trainingsbeispielen beträgt die Fehlerrate des aktiv gelernten Modells nur ein Drittel der Fehlerrate des passiv gelernten Modells. Darüber hinaus konnte die TPCB-ADR Auswahlstrategie erfolgreich außerhalb der Arbeit in den verwandten Problemstellungen der Erkennung von Fußgängern und der Erkennung von Fahrzeugen eingesetzt werden.

Im Bereich der Situationsinterpretation wurde die Zuordnung von Verkehrszeichen zu Fahrspuren auf einem zwei- bis vierspurigen Autobahnabschnitt validiert. Sie erzielt eine hohe Zuordnungsrate von 92%. Bei einer Distanz von 30 m und weniger gelingt die Zuordnung sogar fehlerfrei. Die Funktionalität der Dempster/Shafer-Fusion zur Bestimmung des gültigen Tempolimits wurde qualitativ an vier Beispielszenen demonstriert. Die Bestimmung des von ihr genutzten Situationskontexts erhöht die Komplexität der Verarbeitung zwar stark, lässt dafür aber eine deutliche Reduktion von Konfliktsituationen erwarten. Abschließend wurde der Einsatz von MLNs zur Situationsinterpretation durch die Anwendung zur Erkennung von Baustellen und zur Bestimmung der Fahrrelevanz von Ampeln gezeigt. Sie erweisen sich als flexibles Framework, das mit der durch die Umfelderkennung eingebrachten Unsicherheit umgehen kann und Fehler erkennen und zum Teil sogar kompensieren kann. Dadurch erzielten beide Anwendungen sehr hohe Erkennungsraten: Baustellen werden zu 97.2% und fahrrelevante Ampeln zu 98.8% korrekt erkannt.

6. Schlussbetrachtung

6.1. Zusammenfassung

Fahrerassistenzsysteme leisten wichtige Beiträge zur Erhöhung von Sicherheit, Komfort und Energieeffizienz im Straßenverkehr. Zukünftige Generationen kognitiver Systeme sollen immer mehr Fahraufgaben übernehmen und sich intelligent im Straßenverkehr verhalten können. Dies ist nur möglich, wenn alle relevanten Objekte im Umfeld erkannt und ihre Bedeutung richtig eingeschätzt wird. Diese Arbeit leistet einen Beitrag dazu, indem Methoden und Verfahren zur Erkennung statischer Verkehrselemente (Verkehrszeichen und Lichtsignale) untersucht wurden. Ansätze zur Situationsinterpretation wurden entwickelt, mit denen Beziehungen zwischen den Objekten analysiert werden, um neues Wissen abzuleiten.

Zur Erkennung von Objekten im Fahrzeugumfeld wurden Verfahren zur Segmentierung, Detektion, Klassifikation, zum Tracking und zur temporalen Fusion vorgestellt. Sie stützen sich auf charakteristische Eigenschaften der betrachteten statischen Verkehrselemente. Gemäß der Anforderung an eine zukünftige Umsetzbarkeit wurden dafür echtzeitfähige und robuste Verfahren entwickelt. Sie werden in einer modularen Prozesskette angeordnet, wodurch die Wiederverwendbarkeit der Verfahren in unterschiedlichen Problemstellungen gewährleistet ist. Darauf aufbauend wurden Prozessketten zur Erkennung von Ronden (kreisförmige Verkehrszeichen), Warnhinweisen (dreieckige Verkehrszeichen), Vorwegweisern, Zusatzzeichen und Ampeln präsentiert. Um mit der großen Anzahl verschiedener Arten deutscher Verkehrszeichen umgehen zu können, wurde ein Schwerpunkt auf Untersuchungen zur Skalierbarkeit der Klassifikation gelegt. Dazu gehört die Einführung der hierarchischen Support-Vektor-Maschine und die Nutzung von aktivem Lernen.

Vor dem Hintergrund der Bestimmung der Fahrrelevanz von Verkehrselementen wurden verschiedene Verfahren zur Situationsinterpretation untersucht. Da aus Sensordaten extrahierte Informationen immer verrauscht und auch andere Informationsquellen wie Kartendienste fehlerbehaftet sind, müssen dafür probabilistische Ansätze eingesetzt werden. Die Zuordnung von Verkehrszeichen zu Fahrspuren führt eine komplementäre Fusion mit einem Fahrspurerkenner durch, um die Fahrrelevanz anhand von räumlichen Relationen zu bestimmen. Die Dempster/Shafer-Fusion von Perzeption und Kartendiensten zur Bestimmung gültiger Informationen stellt eine sowohl kooperative wie auch konkurrierende Informationsfusion dar. Durch die zusätzliche Adaption an den Situationskontext wird hierbei das Konfliktpotential stark gesenkt und dadurch Fehler vermieden. Markov-Logik-Netze wurden als Framework zur Handhabung vielfältiger Aufgaben der Situationsinterpretation eingeführt. Ihre Flexibilität wird exemplarisch durch die Anwendung auf Problemstellungen zur Erkennung von Baustellen, zur Bestimmung der Gültigkeit von Einschränkungen durch Zusatzzeichen und zur Feststellung der Fahrrelevanz von Ampeln aufgezeigt.

Schließlich wurde das Konzept zur kontextsensitiven Umfelderkennung und Situationsinterpretation in

einer Vielzahl von Experimenten validiert. Dazu zählt die Evaluation der Anwendungen zur Umfelderkennung sowie ihrer einzelnen Komponenten. Zur Untersuchung neuartiger Anwendungsgebiete wurde die Übertragung der Ansätze auf Mobilgeräte untersucht und ein System zur semi-automatischen Kartierung von Verkehrselementen entwickelt und getestet. Die Bestimmung fahrrelevanter Verkehrselemente wurde durch Experimente im Bereich der Situationsinterpretation validiert: Die probabilistische Zuordnung von Verkehrszeichen zur eigenen Fahrspur, die kontextsensitive Dempster/Shafar-Fusion zur Bestimmung des gültigen Tempolimits und die Anwendung von Markov-Logik-Netzen zur Bestimmung fahrrelevanter Ampeln.

6.2. Diskussion

Das vorgelegte Konzept erweitert gängige Konzepte zur Umfelderkennung für Verkehrszeichen um die Berücksichtigung des Kontexts in Form der Betrachtung zusätzlicher relevanter Informationsquellen. Durch Informationsfusion und probabilistische Ansätze zur Auswertung von Relationen wird neues Wissen abgeleitet. Neuartig ist insbesondere die Nutzung von Markov-Logik-Netzen als flexibles Framework zur Situationsinterpretation, welches unsicheres Wissen aus der Perzeption mit sicherem Wissen (etwa Verkehrsregeln) vereint. Durch die Nutzung des Situationskontexts im Rahmen der Informationsfusion wird eine adaptive Fusion ermöglicht, bei der die Zuverlässigkeit von Informationsquellen ständig überprüft und angepasst wird, um Fehler zu vermeiden. Entitäten wie Fahrspuren, Wetter, Baustellen oder die Sichtbarkeit werden betrachtet, um Relationen zu analysieren und Verkehrselemente richtig in die Verkehrssituation einzuordnen. Ausgangsbasis ist die Anwendung robuster Perzeptionsverfahren. Hierfür wurde unter anderem die Skalierbarkeit von Klassifikationsverfahren verbessert durch Einführung der hierarchischen Support-Vektor-Maschine und der Entwicklung der TPCB-ADR Auswahlstrategie im Rahmen des aktiven Lernens. Das Potential der Ansätze wurde durch umfangreiche Experimente validiert. Alle Experimente wurden mit realen Daten durchgeführt und die implementierten Verfahren verarbeiten mehrere Bilder pro Sekunde. Damit wurde erstmalig ein Ansatz zur kontextsensitiven Situationsinterpretation für Verkehrselemente entwickelt, der eine geschlossene Kette von der Aufnahme von Sensordaten bis hin zur Entscheidung über die Fahrrelevanz umsetzt.

6.3. Ausblick

Diese Arbeit bildet die Grundlage für weitere Arbeiten zur Situationsinterpretation für kognitive Assistenzsysteme in der Verkehrsdomäne. Anknüpfungspunkte für zukünftige Erweiterungen und Ergänzungen gibt es wie folgt:

- Experimente im Rahmen der Arbeit wurden auf Deutschland beschränkt und die Verkehrsregeln der StVO zugrunde gelegt. Zum internationalen Einsatz müssen die Perzeptionsverfahren erweitert und neue Verfahren ergänzt werden. Zu untersuchen ist hier unter anderem die Übertragbarkeit bestehender (Teil-)Verfahren und der Austausch von Trainingsdaten. Im Bereich der Situationsinterpretation bringen MLNs sehr gute Voraussetzungen für die Internationalisierung mit, da in

vielen Fällen die Anpassung oder der Austausch der Wissensbasis genügen dürfte. Dieser Ansatz müsste umgesetzt und durch Experimente überprüft werden.

- Zur Weiterentwicklung der bestehenden MLNs und bei der Entwicklung von MLNs für neue Problemstellungen kann der Einsatz von Strukturlernen untersucht werden, um neben Formelgewichten auch die Formeln selbst oder Teile von ihnen zu lernen.
- Auf unterschiedlichen Ebenen werden Lernverfahren eingesetzt, z.B. in der Segmentierung, der Klassifikation und der Situationsinterpretation. Zur Optimierung der Ergebnisse kann es vorteilhaft sein, die Trainingsdaten der einzelnen Verfahren aufeinander abzustimmen. Dazu muss unter anderem untersucht werden, welche Daten ausgetauscht werden können und wie sich die Verfahren gegenseitig beeinflussen.
- An einigen Stellen werden Informationen auf höherem Level genutzt, um andere Komponenten auf niedrigerem Level zu parametrieren – beispielsweise die Änderung von Segmentierungsparametern in prädizierten Regionen nach positiver Klassifikation. Dieses Vorgehen kann an weiteren Stellen eingesetzt werden, um situationsgerecht die Laufzeit zu reduzieren oder Parameter zur Verbesserung der Erkennungsleistung anzupassen. Dabei müssen aber mögliche Wechselwirkungen beachtet werden zum Beispiel wenn verschiedene Anwendungen gleichzeitig Ressourcen wie Sensoren nutzen.
- Andere Verkehrsteilnehmer und der Fahrer wurden in dieser Arbeit nicht betrachtet. Ihre Berücksichtigung würde erweiterte Schlussfolgerungen ermöglichen, zum Beispiel zur Beurteilung von Vorfahrtsregelungen (muss Vorfahrt gewährt werden?) oder zur Hinweisgebung (hat der Fahrer ein fahrrelevantes Verkehrszeichen wahrgenommen?).

A. Quellennachweise

A.1. Veröffentlichungen

- [Bä+11] Tobias Bär, Dennis Nienhüser, Ralf Kohlhaas und J. Marius Zöllner. „Probabilistic Driving Style Determination by means of a Situation Based Analysis of the Vehicle Data“. In: *IEEE Intelligent Transportation Systems*. 2011.
- [GNL09] Thomas Gump, Dennis Nienhüser und Rebecca Liebig. „Recognition and tracking of temporary lanes in motorway construction sites“. In: *IEEE Intelligent Vehicles Symposium*. 2009.
- [GNZ10] Thomas Gump, Dennis Nienhüser und J. Marius Zöllner. „Physical road marker property estimation using monoscopic vision“. In: *IEEE International Conference on Robotics and Automation*. Mai 2010, S. 1022–1027. ISBN: 978-1-4244-5038-1.
- [Koh+11] Ralf Kohlhaas, Thomas Schamm, Dennis Nienhüser und J. Marius Zöllner. „Anticipatory Energy Saving Assistant for Approaching Slower Vehicles“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2011.
- [Miq+10] C. Miquet, J. Henning, B. Schick, Tobias Bär, Marcus Strand, J. Marius Zöllner, Dennis Nienhüser, Thomas Schamm, Thomas Gump und Marco Ziegenmeyer. „Continuous Evaluation of Methods for Sensor Data Fusion of Video-Based Driver Assistance Systems, Using Virtual Road Tests“. In: *10th International Symposium on Advanced Vehicle Control*. 2010.
- [Mü+10] Matthias Müller, Axel Braun, Joachim Gerlach, Wolfgang Rosenstiel, Dennis Nienhüser, J. Marius Zöllner und Oliver Bringmann. „Design of an automotive traffic sign recognition system targeting a multi-core SoC implementation“. In: *Design, Automation and Test in Europe (DATE)*. IEEE, 2010, S. 532–537. ISBN: 9783981080162.
- [NDZ10] Dennis Nienhüser, Markus Drescher und J. Marius Zöllner. „Visual state estimation of traffic lights using hidden Markov models“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2010, S. 1705–1710. ISBN: 9781424476589.
- [NGZ08] Dennis Nienhüser, Thomas Gump und J. Marius Zöllner. „Recognition and attribution of variable message signs and lanes“. In: *IEEE Intelligent Vehicles Symposium*. 2008, S. 55–60.
- [NGZ09] Dennis Nienhüser, Thomas Gump und J. Marius Zöllner. „A situation context aware Dempster-Shafer fusion of digital maps and a road sign recognition system“. In: *IEEE Intelligent Vehicles Symposium*. 2009.

- [NGZ11] Dennis Nienhüser, Thomas Gump, J. Marius Zöllner. „Relevance Estimation of Traffic Elements Using Markov Logic Networks“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2011, S. 1659–1664. ISBN: 9781457721960.
- [Nie+10] Dennis Nienhüser, Thomas Gump, J. Marius Zöllner und Koba Natroshvili. „Fast and reliable recognition of supplementary traffic signs“. In: *IEEE Intelligent Vehicles Symposium*. 2010, S. 896–901. ISBN: 9781424478682.
- [Nie+12] Dennis Nienhüser, Tobias Bär, Ralf Kohlhaas, Thomas Schamm, Jochen Zimmermann, Thomas Gump, Marcus Strand, Oliver Bringmann und J. Marius Zöllner. „Energy Efficient Driving and Operation Strategies Based on Situation Awareness and Reasoning“. In: *it - Information Technology* 54.1 (Feb. 2012), S. 5–16. ISSN: 1611-2776.
- [NZ13] Dennis Nienhüser und J. Marius Zöllner. „Batch-Mode Active Learning for Traffic Sign Recognition“. In: *IEEE Intelligent Vehicles Symposium*. 2013.
- [NZG07] Dennis Nienhüser, Marco Ziegenmeyer und Thomas Gump. „Kamera-basierte Erkennung von Geschwindigkeitsbeschränkungen auf deutschen Straßen“. In: *Autonome Mobile Systeme*. 2007.
- [Zö10] J. Marius Zöllner. „Advanced technical cognitive systems for mobility assistance“. In: *Emerging research directions in computer science : contributions from the young informatics faculty in Karlsruhe*. Hrsg. von Victor Pankratius und Samuel Kounev. Bd. 3. KIT Scientific Publishing, 2010. ISBN: 3866445083.

A.2. Studentische Arbeiten

- [BNZ08] Sebastian Brechtel, Dennis Nienhüser und J. Marius Zöllner. „Kamerabasierte Erkennung von Halteverbotszeichen in urbaner Umgebung“. Studienarbeit. 2008.
- [Dre+10] Markus Drescher, Dennis Nienhüser, J. Marius Zöllner und Rüdiger Dillmann. „Bildbasierte Erkennung situativ relevanter Ampeln für Assistenzsysteme“. Diplomarbeit. 2010.
- [MNZ12] Manuel Martin, Dennis Nienhüser und J. Marius Zöllner. „Traffic Sign Recognition for Mobile Devices“. Studienarbeit. 2012.
- [Web+11] Andreas Weber, Dennis Nienhüser, J. Marius Zöllner und Rüdiger Dillmann. „Semiautomatische OpenStreetMap-Kartierung mit einem Segway Personentransporter“. Bachelorarbeit. 2011.
- [WNZ08] Sebastian Wirkert, Dennis Nienhüser und J. Marius Zöllner. „Bildbasierte Bestimmung von Wetterverhältnissen im Straßenverkehr“. Studienarbeit. 2008.
- [XNZ09] Boxun Xi, Dennis Nienhüser und J. Marius Zöllner. „Kamera-basierte Erkennung von Richtungswegweisern auf deutschen Autobahnen“. Diplomarbeit. 2009.

A.3. Literatur

- [AA96] Yuji Aoyagi und Toshiyuki Asakura. „A study on traffic sign recognition in scene image using genetic algorithms and neural networks“. In: *IEEE International Conference on Industrial Electronics, Control, and Instrumentation (IECON)*. 1996, S. 1343–1348.
- [AJ06] Vavilin Andrey und Kang-Hyun Jo. „Road guidance sign recognition in urban areas by structure“. In: *International Forum on Strategic Technology*. IEEE, 2006, S. 293–296.
- [AKA96] Satoshi Azami, Shunji Katahara und Masayoshi Aoki. „Route guidance sign identification using 2-D structural description“. In: *Conference on Intelligent Vehicles*. IEEE, 1996, S. 153–158. ISBN: 0-7803-3652-6.
- [Ale+07] Bram Alefs, Guy Eschemann, Herbert Ramoser und Csaba Beleznai. „Road sign detection from edge orientation histograms“. In: *IEEE Intelligent Vehicles Symposium*. 2007, S. 993–998. ISBN: 1424410681.
- [Bah+05] Claus Bahlmann, Ying Zhu, Visvanathan Ramesh, Martin Pellkofer und Thorsten Koehler. „A system for traffic sign detection, tracking, and recognition using color, shape, and motion information“. In: *IEEE Intelligent Vehicles Symposium*. 2005, S. 255–260.
- [Bah+08] Claus Bahlmann, Martin Pellkofer, Jan Giebel und Gregory Baratoff. „Multi-modal speed limit assistants: Combining camera and GPS maps“. In: *IEEE Intelligent Vehicles Symposium*. 2008, S. 132–137. ISBN: 9781424425693.
- [Bal81] Dana H. Ballard. „Generalizing the Hough transform to detect arbitrary shapes“. In: *International Conference on Pattern Recognition 13.2* (1981), S. 111–122. ISSN: 00313203.
- [Bar+08] Alexandre Bargeton, Fabien Moutarde, Fawzi Nashashibi und Benazouz Bradai. „Improving pan-European speed-limit signs recognition with a new global number segmentation before digit recognition“. In: *IEEE Intelligent Vehicles Symposium*. 2008, S. 349–354.
- [Bar+09] Xavier Baro, Sergio Escalera, Jordi Vitria, Oriol Pujol und Petia Radeva. „Traffic Sign Recognition Using Evolutionary Adaboost Detection and Forest-ECOC Classification“. In: *IEEE Transactions on Intelligent Transportation Systems* 10.1 (März 2009), S. 113–126. ISSN: 1524-9050.
- [Bar+10] Alexandre Bargeton, Fabien Moutarde, Fawzi Nashashibi und Anne-Sophie Puthon. „Joint interpretation of on-board vision and static GPS cartography for determination of correct speed limit“. In: *CoRR abs/1010.3* (2010).
- [Bar+11] Matthew Barth, Sindhura Mandava, Kanok Boriboonsomsin und Haitao Xia. „Dynamic ECO-driving for arterial corridors“. In: *IEEE Forum on Integrated and Sustainable Transportation Systems* (2011).
- [Bar05] Nick Barnes. „Improved signal to noise ratio and computational speed for gradient-based detection algorithms“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. April. 2005, S. 4661–4666. ISBN: 078038914X.
- [BL06] Nick Barnes und Gareth Loy. „Real-time regular polygonal sign detection“. In: *Field and Service Robotics* (2006).

- [Bou+11] Katherine L. Bouman, Golnaz Abdollahian, Mireille Boutin und Edward J. Delp. „A Low Complexity Sign Detection and Text Localization Method for Mobile Applications“. In: *IEEE Transactions on Multimedia* 13.5 (Okt. 2011), S. 922–934. ISSN: 1520-9210.
- [BPF12] Jürgen Beyerer, Fernando Puente León und Christian Frese. *Automatische Sichtprüfung: Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung*. Springer, 2012, S. 963.
- [Bre65] Jack E. Bresenham. „Algorithm for computer control of a digital plotter“. In: *IBM Systems Journal* 4.1 (1965), S. 25–30. ISSN: 0018-8670.
- [Bri03] Klaus Brinker. „Incorporating diversity in active learning with support vector machines“. In: *20th International Conference on Machine Learning*. 2003, S. 59–66.
- [Bro+07] Alberto Broggi, Pietro Cerri, Paolo Medici, Pier Paolo Porta und Guido Ghisio. „Real time road signs recognition“. In: *IEEE Intelligent Vehicles Symposium*. section III. 2007, S. 981–986. ISBN: 1424410681.
- [Bur98] Christopher J.C. Burges. „A Tutorial on Support Vector Machines for Pattern Recognition“. en. In: *Data Mining and Knowledge Discovery* 2.2 (Juni 1998), S. 121–167. ISSN: 1573-756X.
- [BZ04] Nick Barnes und Alex Zelinsky. „Real-time radial symmetry for speed sign detection“. In: *IEEE Intelligent Vehicles Symposium*. 2004, S. 566–571. ISBN: 0780383109.
- [BZF08] Nick Barnes, Alex Zelinsky und Luck S. Fletcher. „Real-Time Speed Sign Detection Using the Radial Symmetry Detector“. In: *IEEE Transactions on Intelligent Transportation Systems* 9.2 (Juni 2008), S. 322–332. ISSN: 1524-9050.
- [Can86] John Canny. „A Computational Approach to Edge Detection“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), S. 679–698. ISSN: 0162-8828.
- [Car+08] Claudio Caraffi, Elena Cardarelli, Paolo Medici, Pier Paolo Porta, Guido Ghisio und Gianluca Monchiero. „An algorithm for Italian de-restriction signs detection“. In: *IEEE Intelligent Vehicles Symposium* (2008).
- [Car+09] Elena Cardarelli, Paolo Medici, Pier Paolo Porta und Guido Ghisio. „Road signs shapes detection based on Sobel phase analysis“. In: *IEEE Intelligent Vehicles Symposium*. Juni 2009, S. 376–381. ISBN: 978-1-4244-3503-6.
- [CGL13] Zixing Cai, Mingqin Gu und Yi Li. „Real-time Arrow Traffic Light Recognition System for Intelligent Vehicle“. In: *elrond.informatik.tu-freiberg.de* (2013).
- [Che+04] Xilin Chen, Jie Yang, Jing Zhang und Alex Waibel. „Automatic detection and recognition of signs from natural scenes.“ In: *IEEE Transactions on Image Processing* 13.1 (2004), S. 87–99.
- [Che+11] Long Chen, Qingquan Li, Ming Li und Qingzhou Mao. „Traffic sign detection and recognition for intelligent vehicle“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 908–913. ISBN: 9781457708893.

- [CL11] Chih-Chung Chang und Chih-Jen Lin. „LIBSVM: a library for support vector machines“. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* (2011). ISSN: 21576904.
- [CN09] Raoul de Charette und Fawzi Nashashibi. „Real time visual traffic lights recognition based on Spot Light Detection and adaptive traffic lights templates“. In: *IEEE Intelligent Vehicles Symposium*. Juni 2009, S. 358–363. ISBN: 978-1-4244-3503-6.
- [DDT11] Keisuke Doman, Daisuke Deguchi und Tomokazu Takahashi. „Estimation of traffic sign visibility considering temporal environmental changes for smart driver assistance“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 667–672. ISBN: 9781457708893.
- [Deg+11] Daisuke Deguchi, Mitsunori Shirasuna, Keisuke Doman, Ichiro Ide und Hiroshi Murase. „Intelligent traffic sign detector: Adaptive learning based on online gathering of training samples“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 72–77. ISBN: 9781457708893.
- [DH72] Richard O. Duda und Peter E. Hart. „Use of the Hough transformation to detect lines and curves in pictures“. In: *Communications of the ACM* 15.1 (Jan. 1972), S. 11–15. ISSN: 00010782.
- [DL11] Jérémie Daniel und Jean-Philippe Lauffenburger. „Conflict management in multi-sensor dempster-shafer fusion for speed limit determination“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 987–992. ISBN: 9781457708893.
- [DLB11] Jérémie Daniel, Jean-Philippe Lauffenburger und Michel Basset. „Multi-criterion Dempster-Shafer Fusion for Speed Limit Determination“. In: *IFAC World Congress* (2011).
- [Dom+10] Keisuke Doman, Daisuke Deguchi, Tomokazu Takahashi, Y. Mekada, Ichiro Ide, Hiroshi Murase und Y. Tamatsu. „Estimation of traffic sign visibility toward smart driver assistance“. In: *IEEE Intelligent Vehicles Symposium*. 2010, S. 45–50. ISBN: 9781424478682.
- [DPB11] Begüm Demir, Claudio Persello und Lorenzo Bruzzone. „Batch-Mode Active-Learning Methods for the Interactive Classification of Remote Sensing Images“. In: *IEEE Transactions on Geoscience and Remote Sensing* 49.3 (März 2011), S. 1014–1031. ISSN: 0196-2892.
- [DT05] Navneet Dalal und Bill Triggs. „Histograms of Oriented Gradients for Human Detection“. In: *IEEE Computer Society International Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 1. 2005, S. 886–893.
- [Eck07] Gerald Eckert. „Erkennung von Ampeln aus Onboard-Videosequenzen für Fahrerassistenzsysteme“. Dissertation. 2007, S. 136.
- [Est+94] S. Estable, J. Schick, F. Stein, R. Janssen, R. Ott, Werner Ritter und Y.J. Zheng. „A real-time traffic sign recognition system“. In: *IEEE Intelligent Vehicles Symposium*. 1994, S. 213–218.
- [FD05] Hasan Fleyeh und Mark Dougherty. „Road and traffic sign detection and recognition“. In: *10th EWGT Meeting and 16th Mini-EURO Conference*. 2005, S. 644–653.
- [FH10] Meng-Yin Fu und Yuan-Shui Huang. „A survey of traffic sign recognition“. In: *Analysis* July (2010), S. 11–14.

- [Fra+97] Uwe Franke, Steffen Görzig, Frank Lindner, D. Mehren und F. Paetzold. „Steps towards an intelligent vision system for driver assistance in urban traffic“. In: *IEEE Intelligent Transportation Systems*. 1997, S. 601–606. ISBN: 0780342690.
- [FS95] Yoav Freund und Robert E. Schapire. „A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting“. In: *Computational Learning Theory* (1995).
- [GG+11] M.A. Garcia-Garrido, M. Ocana, M. A. Llorca, A. Sotelo, E. Arroyo und A. Llamazares. „Robust traffic signs detection by means of vision and V2I communications“. In: *IEEE Intelligent Transportation Systems* (2011), S. 1003–1008.
- [GG84] Stuart Geman und Donald Geman. „Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6.6* (Nov. 1984), S. 721–741. ISSN: 0162-8828.
- [GJ+05] Pedro Gil-Jiménez, Sergio Lafuente-Arroyo, Hilario Gomez-Moreno, Francisco Lopez-Ferreras und Saturnino Maldonado-Bascón. „Traffic sign shape classification evaluation. Part II. FFT applied to the signature of blobs“. In: *IEEE Intelligent Vehicles Symposium*. 2005, S. 607–612. ISBN: 0780389611.
- [GJ+07] Pedro Gil-Jiménez, Hilario Gomez-Moreno, P. Siegmann, Sergio Lafuente-Arroyo und Saturnino Maldonado-Bascón. „Traffic sign shape classification based on Support Vector Machines and the FFT of the signature of blobs“. In: *IEEE Intelligent Vehicles Symposium*. 2007, S. 375–380. ISBN: 1424410681.
- [GJ+08] Pedro Gil-Jiménez, Saturnino Maldonado-Bascón, Hilario Gomez-Moreno, Sergio Lafuente-Arroyo und Francisco Lopez-Ferreras. „Traffic sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies“. In: *Signal Processing* (2008).
- [Gon+10] Jianwei Gong, Yanhua Jiang, Guangming Xiong, Chaohua Guan, Gang Tao und Huiyan Chen. „The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles“. In: *IEEE Intelligent Vehicles Symposium*. 2010, S. 431–435. ISBN: 9781424478682.
- [Gu+10] Yanlei Gu, Tomohiro Yendo, M.P. Tehrani, Toshiaki Fujii und Masayuki Tanimoto. „A new vision system for traffic sign recognition“. In: *IEEE Intelligent Vehicles Symposium*. 2010, S. 7–12. ISBN: 9781424478682.
- [GWF03] Stefan K. Gehrig, Sebastian Wagner und Uwe Franke. „System architecture for an intersection assistant fusing image, map, and gps information“. In: *IEEE Intelligent Vehicles Symposium*. 19. 2003, S. 144–149. ISBN: 0780378482.
- [GYT11] Yanlei Gu, Tomohiro Yendo und Mehrdad Panahpour Tehrani. „Traffic sign detection in dual-focal active camera system“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 1052–1057. ISBN: 9781457708893.
- [Ham+08] Omar Hamdoun, Alexandre Bargeton, Fabien Moutarde und Benazouz Bradai. „Detection and recognition of end-of-speed-limit and supplementary signs for improved European speed limit support“. In: *World Congress on Intelligent Transport Systems (ITS)* (2008), S. 7–10.

- [HCL03] Chih-Wei Hsu, Chih-Chung Chang und Chih-Jen Lin. „A practical guide to support vector classification“. In: *Bioinformatics* 1.1 (2003), S. 1–12.
- [Hou62] Paul V. C. Hough. *Method and means for recognizing complex patterns*. Dez. 1962.
- [HW12] Matthias Hillebrand und Christian Wöhler. „Semi-supervised training set adaption to unknown countries for traffic sign classifiers“. In: *First LAPR TC3 conference on Partially Supervised Learning*. Bd. 7081. Lecture Notes in Computer Science. Sep. 2012, S. 120–127. ISBN: 978-3-642-28257-7.
- [HZ09] Benjamin Hoferlin und Klaus Zimmermann. „Towards reliable traffic sign recognition“. In: *IEEE Intelligent Vehicles Symposium*. Juni 2009, S. 324–329. ISBN: 978-1-4244-3503-6.
- [Jam+11] Homayoun Jamshidi, Thomas Lukaszewicz, Amin Kashi, Ansel Berghuvud, Hans-Jürgen Zepernick und Siamak Khatibi. „Fusion of digital map traffic signs and camera-detected signs“. In: *International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, Dez. 2011, S. 1–7. ISBN: 978-1-4577-1180-0.
- [Jan+93] R. Janssen, Werner Ritter, F. Stein und S. Ott. „Hybrid approach for traffic sign recognition“. In: *IEEE Intelligent Vehicles Symposium*. 1993, S. 390–395.
- [Jan04] Holger Janssen. „Vehicle surround sensing based on information fusion of monocular video and digital map“. In: *IEEE Intelligent Vehicles Symposium*. 2004, S. 244–249. ISBN: 0780383109.
- [Jia+11] Yan Jiang, Shengyan Zhou, Jianwei Gong, Guangming Xiong und Huiyan Chen. „Traffic sign recognition using Ridge Regression and OTSU method“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 613–618. ISBN: 9781457708893.
- [JPP10] A. J. Joshi, F. Porikli und N. Papanikolopoulos. „Multi-class batch-mode active learning for image classification“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2010, S. 1873–1878. ISBN: 9781424450381.
- [JS04] C. R. Jung und Rodrigo Schramm. „Rectangle detection based on a windowed Hough transform“. In: *Symposium on Computer Graphics and Image Processing*. IEEE, 2004, S. 113–120.
- [Jä05] B. Jähne. *Digitale Bildverarbeitung*. Springer, 2005. ISBN: 9783540249993.
- [KAB09] Jesmin F. Khan, Reza R. Adhami und Sharif M. A. Bhuiyan. „Image segmentation based road sign detection“. In: *IEEE Southeastcon*. März 2009, S. 24–29. ISBN: 978-1-4244-3976-8.
- [Kal60] Rudolph Emil Kalman. „A new approach to linear filtering and prediction problems“. In: *Journal of basic Engineering* 82.Series D (1960), S. 35–45.
- [Kas+10] Robert Kastner, Thomas Michalke, Thomas Burbach, Jannik Fritsch und Christian Goerick. „Attention-based traffic sign recognition with an array of weak classifiers“. In: *IEEE Intelligent Vehicles Symposium*. 2010, S. 333–339. ISBN: 9781424478682.
- [KBA11] Jesmin F. Khan, Sharif M. A. Bhuiyan und Reza R. Adhami. „Image Segmentation and Shape Analysis for Road-Sign Detection“. In: *IEEE Transactions on Intelligent Transportation Systems* 12.1 (März 2011), S. 83–96. ISSN: 1524-9050.

- [Kel+08] C. G. Keller, Christoph Sprunk, Claus Bahlmann, Jan Giebel und Gregory Baratoff. „Real-time recognition of US speed signs“. In: *IEEE Intelligent Vehicles Symposium*. 2008, S. 518–523. ISBN: 9781424425693.
- [Kir+09] C.G. Kiran, Lekhesh V. Prabhu, Rahiman V. Abdu und K. Rajeev. „Traffic Sign Detection and Pattern Recognition Using Support Vector Machine“. In: *International Conference on Advances in Pattern Recognition*. IEEE, Feb. 2009, S. 87–90. ISBN: 978-0-7695-3520-3.
- [KKY07] Y.K. Kim, K.W. Kim und X. Yang. „Real time traffic light recognition system for color vision deficiencies“. In: *International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2007, S. 76–81. ISBN: 1424408288.
- [Koh+13] Ralf Kohlhaas, Thomas Schamm, Dominik Lenk und J. Marius Zöllner. „Towards driving autonomously: Autonomous cruise control in urban environments“. In: *IEEE Intelligent Vehicles Symposium*. 2013, S. 109–114.
- [Kou07] A. Z. Kouzani. „Road-sign identification using ensemble learning“. In: *IEEE Intelligent Vehicles Symposium*. 2007, S. 438–443. ISBN: 1424410681.
- [LA+05] Sergio Lafuente-Arroyo, Pedro Gil-Jiménez, R. Maldonado-Bascon, Francisco Lopez-Ferreras und Saturnino Maldonado-Bascón. „Traffic sign shape classification evaluation I: SVM using distance to borders“. In: *IEEE Intelligent Vehicles Symposium*. 2005, S. 557–562.
- [LA+07] Sergio Lafuente-Arroyo, Saturnino Maldonado-Bascón, Pedro Gil-Jiménez, J. Acevedo-Rodriguez und R. J. Lopez-Sastre. „A tracking system for automated inventory of road signs“. In: *IEEE Intelligent Vehicles Symposium*. 2007, S. 166–171. ISBN: 1424410681.
- [LA+08] Sergio Lafuente-Arroyo, Saturnino Maldonado-Bascón, Pedro Gil-Jiménez und Hilario Gomez-Moreno. „An intra-image tracking algorithm for traffic sign recognition“. In: *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. Sep. 2008, S. 259–264. ISBN: 978-1-4244-2359-0.
- [Lar] LaRA – La Route Automatisée. 2013. URL: <http://www.lara.prd.fr/>.
- [LB04] Gareth Loy und Nick Barnes. „Fast shape-based road sign detection for a driver assistance system“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Bd. 1. 2004, S. 70–75. ISBN: 0780384636.
- [LBB08] Jean-Philippe Lauffenburger, Benazouz Bradai und M. Basset. „Navigation and speed signs recognition fusion for enhanced vehicle location“. In: *International Federation of Automatic Control (IFAC) (2008)*, S. 2069–2074.
- [LD07] Daniel Lowd und Pedro Domingos. „Efficient weight learning for Markov logic networks“. In: *Knowledge Discovery in Databases: PKDD 2007*. Springer, 2007, S. 200–211.
- [Li+11] Yi Li, Zi-Xing Cai, Ming-Qin Gu und Qiao-Yun Yan. „Traffic lights recognition based on morphology filtering and statistical classification“. In: *International Conference on Natural Computation*. IEEE, Juli 2011, S. 1700–1704. ISBN: 978-1-4244-9950-2.
- [Liu+11] Wei Liu, Jin Lv, Haihua Gao, B. Duan, Huai Yuan und Hong Zhao. „An efficient real-time speed limit signs recognition based on rotation invariant feature“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 1000–1005. ISBN: 9781457708893.

- [LK04] Frank Lindner und Ulrich Kressel. „Robust recognition of traffic signals“. In: *IEEE Intelligent Vehicles Symposium* (2004), S. 49–53.
- [LLX02] H. Liu, D. Liu und J. Xin. „Real-time recognition of road traffic sign in motion image based on genetic algorithm“. In: *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*. Bd. 1. IEEE, 2002, S. 83–86.
- [IVPLAC10] I. Landesa-Vazquez, F. Parada-Loira und J. L. Alba-Castro. „Fast real-time multiclass traffic sign detection based on novel shape and texture descriptors“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2010, S. 1388–1395. ISBN: 9781424476589.
- [LY10] Ching-Hao Lai und Chia-Chen Yu. „An Efficient Real-Time Traffic Sign Recognition System for Intelligent Vehicles with Smart Phones“. In: *International Conference on Technologies and Applications of Artificial Intelligence*. IEEE, Nov. 2010, S. 195–202. ISBN: 978-1-4244-8668-7.
- [Mar+11] Javier Marinas, Luis Salgado, Jon Arróspide und Marcos Nieto. „Detection and Tracking of Traffic Signs Using a Recursive Bayesian Decision Framework“. In: *IEEE International Conference on Intelligent Transportation Systems* (2011), S. 1942–1947.
- [MB+07] Saturnino Maldonado-Bascón, Sergio Lafuente-Arroyo, Pedro Gil-Jiménez, Hilario Gomez-Moreno und Francisco Lopez-Ferreras. „Road-Sign Detection and Recognition Based on Support Vector Machines“. In: *IEEE Transactions on Intelligent Transportation Systems* 8.2 (Juni 2007), S. 264–278. ISSN: 1524-9050.
- [MC08] Paolo Medici und C. Caraffi. „Real time road signs classification“. In: *IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (2008).
- [McL+08] S. McLoughlin, C. Deegan, C. Mulvihill, C. Fitzgerald und C. Markham. „Mobile mapping for the automated analysis of road signage and delineation“. In: *Intelligent Transport Systems, IET* 2.1 (2008), S. 61–73.
- [Meu+10] Mirko Meuter, Stefan Müller-Schneiders, Christian Nunn, Stephanie Hold, Steffen Goermer und Anton Kummert. „Decision fusion and reasoning for traffic sign recognition“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2010, S. 324–329. ISBN: 9781424476589.
- [MKS00] Jun Miura, Tsuyoshi Kanda und Yoshiaki Shirai. „An active vision system for real-time traffic sign recognition“. In: *IEEE International Conference on Intelligent Transportation Systems*. Mi. 2000, S. 52–57. ISBN: 0780359712.
- [Mob] *Traffic Sign Detection - Mobileye*. 2013. URL: <http://www.mobileye.com/technology/applications/traffic-sign-detection/>.
- [Mou+07] Fabien Moutarde, Alexandre Bargeton, Anne Herbin und Lowik Chanussot. „Robust on-vehicle real-time visual detection of American and European speed limits signs, with a modular Traffic Signs Recognition system“. In: *IEEE Intelligent Vehicles Symposium*. Bd. 51. 33. 2007, S. 1122–1126. ISBN: 1424410681.

- [MTM12] Andreas Mogelmoose, Mohan Manubhai Trivedi und Thomas B. Moeslund. „Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey“. In: *IEEE Transactions on Intelligent Transportation Systems* 13.4 (Dez. 2012), S. 1484–1497. ISSN: 1524-9050.
- [NKMS08] Christian Nunn, Anton Kummert und Stefan Muller-Schneiders. „A two stage detection module for traffic signs“. In: *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. Sep. 2008, S. 248–252. ISBN: 978-1-4244-2359-0.
- [Ope] *OpenStreetMap*. URL: <http://openstreetmap.org>.
- [Ove11] Gary Overett. „Large scale sign detection using HOG feature variants“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 326–331. ISBN: 9781457708893.
- [PC07] Carlos Filipe Paulo und Paulo Lobato Correia. „Automatic Detection and Classification of Traffic Signs“. In: *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*. IEEE, Juni 2007, S. 11–11. ISBN: 0-7695-2818-X.
- [PD06] Hoifung Poon und Pedro Domingos. „Sound and efficient inference with probabilistic and deterministic dependencies“. In: *AAAI*. Bd. 6. 2006, S. 458–463.
- [Pic+94] G. Piccioli, E. De Micheli, P. Parodi und M. Campani. „Robust road sign detection and recognition from image sequences“. In: *IEEE Intelligent Vehicles Symposium*. 1994, S. 278–283.
- [PLAC10] F. Parada-Loira und J. L. Alba-Castro. „Local Contour Patterns for fast traffic sign detection“. In: *IEEE Intelligent Vehicles Symposium*. 2010, S. 1–6. ISBN: 9781424478682.
- [PMN12] Anne-Sophie Puthon, Fabien Moutarde und Fawzi Nashashibi. „Subsign Detection with Region-Growing from Contrasteds Seeds“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2012. ISBN: 9781467330633.
- [PMN13] Anne-Sophie Puthon, Fabien Moutarde und Fawzi Nashashibi. „Recognition of Supplementary Signs for Correct Interpretation of Traffic Signs“. In: *Workshop on Environment Perception and Navigation for Intelligent Vehicles, IEEE Intelligent Vehicles Symposium*. 2013.
- [PNB10] Anne-Sophie Puthon, Fawzi Nashashibi und Benazouz Bradai. „Improvement of multi-sensor fusion in speed limit determination by quantifying navigation reliability“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2010, S. 855–860. ISBN: 9781424476589.
- [PNB11] Anne-Sophie Puthon, Fawzi Nashashibi und Benazouz Bradai. „A complete system to determine the speed limit by fusing a GIS and a camera“. In: *IEEE International Conference on Intelligent Transportation Systems (2011)*, S. 1686–1691.
- [PND06] P. Paclik, J. Novovicova und R.P.W. Duin. „Building Road-Sign Classifiers Using a Trainable Similarity Measure“. In: *IEEE Transactions on Intelligent Transportation Systems* 7.3 (Sep. 2006), S. 309–321. ISSN: 1524-9050.
- [PR93] L. Priese und V. Rehrmann. „A fast hybrid color segmentation method“. In: *DAGM Symposium Mustererkennung*. Citeseer, 1993, S. 297–304.

- [Pri+93] Lutz Prieße, Volker Rehrmann, Rainer Schian und Raimund Lakmann. „Traffic Sign Recognition Based on Color Image Evaluation“. In: *IEEE Intelligent Vehicles Symposium*. 1993, S. 95–100.
- [Rab89] Lawrence R. Rabiner. „A tutorial on hidden Markov models and selected applications in speech recognition“. In: *Proceedings of the IEEE* 77.2 (1989), S. 257–286.
- [RD06] Matthew Richardson und Pedro Domingos. „Markov logic networks“. In: *Machine Learning* 62.1-2 (Jan. 2006), S. 107–136. ISSN: 0885-6125.
- [Rit92] Werner Ritter. „Traffic sign recognition in color image sequences“. In: *IEEE Intelligent Vehicles Symposium*. 1992.
- [RLL10] Andrzej Ruta, Yongmin Li und Xiaohui Liu. „Real-time traffic sign recognition from video by class-specific discriminative features“. In: *International Conference on Pattern Recognition* 43.1 (Jan. 2010), S. 416–430. ISSN: 00313203.
- [RM08] Martin Roser und Frank Moosmann. „Classification of weather situations on single color images“. In: *IEEE Intelligent Vehicles Symposium*. 2008, S. 798–803. ISBN: 9781424425693.
- [Rö09] D.P.D. Rösler. „Fahrrelevanz von Elementen des Straßenverkehrs - Beschreibung, Erfassung und Anwendung“. In: *tu-chemnitz.de* (2009).
- [SB85] Satoshi Suzuki und Keiichi A. Be. „Topological structural analysis of digitized binary images by border following“. In: *Computer Vision, Graphics, and Image Processing* 30.1 (Apr. 1985), S. 32–46. ISSN: 0734189X.
- [Sch00] H. Scharf. „Optimale Operatoren in der digitalen Bildverarbeitung“. In: (2000).
- [SD05] Parag Singla und Pedro Domingos. „Discriminative training of Markov logic networks“. In: *AAAI*. Bd. 5. 2005, S. 868–873.
- [SD06] George K. Siogkas und Evangelos S. Dermatas. „Detection, Tracking and Classification of Road Signs in Adverse Conditions“. In: *IEEE Mediterranean Electrotechnical Conference (MELECON)*. 2006, S. 537–540. ISBN: 1-4244-0087-2.
- [Seg+10] S. Segvic, K. Brkic, Z. Kalafatic, V. Stanisavljevic, M. Sevrovic, D. Budimir und I. Dadic. „A computer vision assisted geoinformation inventory for traffic infrastructure“. In: *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, Sep. 2010, S. 66–73. ISBN: 978-1-4244-7657-2.
- [Set09] Burr Settles. *Active learning literature survey*. Techn. Ber. University of Wisconsin–Madison, 2009.
- [Sha02] Glenn Shafer. *Dempster-Shafer Theory*. Hrsg. von Stuart C Shapiro. 2002.
- [Shi+02] T. Shioyama, H. Wu, N. Nakamura und S. Kitawaki. „Measurement of the length of pedestrian crossings and detection of traffic lights from image data“. In: *Measurement Science and Technology* 13 (2002), S. 1450.
- [Shn05] Michael Shneier. „Road sign detection and recognition“. In: *IEEE Computer Society International Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 100. June. 2005.

- [SHW05] Piyanuch Silapachote, Allen Hanson und Richard Weiss. „A hierarchical approach to sign recognition“. In: *IEEE Workshops on Application of Computer Vision (WACV/MOTIONS)*. Bd. 1. 2005, S. 22–28. ISBN: 0769522718.
- [Sil+05] Piyanuch Silapachote, Jerod Weinman, Allen Hanson, M.A. Mattar und Richard Weiss. „Automatic sign detection and recognition in natural scenes“. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005, S. 27–27.
- [SO09] Y. Shen und U. Ozguner. „A robust video based traffic light detection algorithm for intelligent vehicles“. In: *IEEE Intelligent Vehicles Symposium (2009)*.
- [SPJ04] Christin Seifert, Lucas Paletta und Andreas Jeitler. „Visual object detection for mobile road sign inventory“. In: *Mobile Human-Computer Interaction (MobileHCI)*. 2004, S. 491–495.
- [SSD12] George G. Siogkas, Evangelos Skodras und Evangelos S. Dermatas. „Traffic lights detection in adverse conditions using color, symmetry and spatiotemporal information“. In: *wcl.ece.upatras.gr*. 2012.
- [ST02] Lukas Sekanina und Jim Torresen. „Detection of Norwegian speed limit signs“. In: *European Simulation Multiconference (ESM)*. Bd. 90. 2002, S. 337–340.
- [ST94] J. Shi und C. Tomasi. „Good features to track“. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 1994, S. 593–600.
- [Sta] Bundesministerium für Verkehr Bau und Stadtentwicklung. *Allgemeine Verwaltungsvorschrift zur Straßenverkehrs-Ordnung*. URL: http://www.verwaltungsvorschriften-im-internet.de/bsvwvbund_26012001_S3236420014.htm.
- [Sta+11] Johannes Stallkamp, Marc Schlipsing, Jan Salmen und Christian Igel. „The German Traffic Sign Recognition Benchmark: A multi-class classification competition“. In: *International Joint Conference on Neural Networks*. IEEE, Juli 2011, S. 1453–1460. ISBN: 978-1-4244-9635-8.
- [Sta13] Bundesministerium für Verkehr Bau und Stadtentwicklung. *Straßenverkehrs-Ordnung (StVO)*. 2013. URL: http://www.gesetze-im-internet.de/stvo_2013/index.html.
- [STB09] Ludovic Simon, J.P. Tarel und R. Bremond. „Alerting the drivers about road signs with poor visual saliency“. In: *IEEE Intelligent Vehicles Symposium*. 2009, S. 48–53. ISBN: 9781424435043.
- [TB04] Jim Torresen und J. W. Bakke. „Efficient recognition of speed limit signs“. In: *IEEE Intelligent Transportation Systems Conference*. Bd. 90. 2004, S. 652–656. ISBN: 0780385004.
- [Tes] *tesseract-ocr*. 2013. URL: <http://code.google.com/p/tesseract-ocr/>.
- [TZG09] Radu Timofte, Karel Zimmermann und Luc Van Gool. „Multi-view traffic sign detection, recognition, and 3D localisation“. In: *Workshop on Applications of Computer Vision (WACV)*. IEEE, Dez. 2009, S. 1–8. ISBN: 978-1-4244-5497-6.
- [VA08] Allgemeine Deutsche Automobil-Club e. V. (ADAC). *ADAC Umfrage - Verkehrsschilder in Europa 2008*. 2008. URL: <http://www.adac.de/infotestrat/tests/strassen/schilderwald/2008/default.aspx>.

- [VA12] Allgemeine Deutsche Automobil-Club e. V. (ADAC). *ADAC Motorwelt, Heft 9 2012*. 2012.
- [Vis] VisLab. 2013. URL: <http://vislab.it/>.
- [VJ01] Paul Viola und Michael Jones. „Robust real-time object detection“. In: *International Journal of Computer Vision* 57.2 (2001), S. 137–154.
- [VR05] A. Vázquez-Reina. „Traffic sign shape classification based on correlation techniques“. In: *WSEAS International Conference on Signal Processing, Computational Geometry and Artificial Vision* (2005).
- [WCY05] Wen Wu, Xilin Chen und Jie Yang. „Detection of text on road signs from video“. In: *IEEE International Conference on Intelligent Transportation Systems* 6.4 (2005), S. 378–390.
- [WES04] Wei Wei, Jordan Erenrich und Bart Selman. „Towards efficient sampling: Exploiting random walk strategies“. In: *AAAI*. Bd. 4. 2004, S. 670–676.
- [Wie] *Vienna convention on road signs and signals*. 1968.
- [WLW04] Ting-Fan Wu, Chih-Jen Lin und Ruby C. Weng. „Probability Estimates for Multi-class Classification by Pairwise Coupling“. In: *The Journal of Machine Learning Research* 5 (Dez. 2004), S. 975–1005. ISSN: 1532-4435.
- [XK08] G. Xiaohong und H. Kunbin. „Colour vision model-based approach for segmentation of traffic signs“. In: *EURASIP Journal on Image and Video Processing* (2008).
- [YCW10] Zhang Yankun, Hong Chuyang und Charles Wang. „An efficient real time rectangle speed limit sign recognition system“. In: *IEEE Intelligent Vehicles Symposium*. 2010, S. 34–38. ISBN: 9781424478682.
- [YH05] Qixiang Ye und Qingming Huang. *A New Text Detection Algorithm in Images/Video Frames*. Hrsg. von Kiyoharu Aizawa, Yuichi Nakamura und Shin’ichi Satoh. Bd. 3332. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. ISBN: 978-3-540-23977-2.
- [Yue+90] H. K. Yuen, J. Princen, J. Illingworth und J. Kittler. „Comparative study of Hough transform methods for circle finding“. In: *Image and Vision Computing* 8.1 (1990), S. 71–77.
- [Zhe+12] Zhihui Zheng, Hanxizi Zhang, Bo Wang und Zhifeng Gao. „Robust Traffic Sign Recognition and Tracking for Advanced Driver Assistance Systems“. In: *IEEE International Conference on Intelligent Transportation Systems*. 2012. ISBN: 9781467330633.
- [ZRJ94] Y.J. Zheng, Werner Ritter und R. Janssen. „An adaptive system for traffic sign recognition“. In: *IEEE Intelligent Vehicles Symposium*. 1994, S. 165–170.
- [ZS11a] Fatin Zaklouta und Bogdan Stanculescu. „Segmentation Masks for Real-time Traffic Sign Recognition using Weighted HOG-based Trees“. In: *IEEE International Conference on Intelligent Transportation Systems* (2011), S. 1954–1959.
- [ZS11b] Fatin Zaklouta und Bogdan Stanculescu. „Warning traffic sign recognition using a HOG-based Kd tree“. In: *IEEE Intelligent Vehicles Symposium*. Iv. 2011, S. 1019–1024. ISBN: 9781457708893.
- [Goo13] Google. *Google Scholar*. 2013. URL: <http://scholar.google.com/>.

A.4. Patente

- [Bah+06] Claus Bahlmann, Ying Zhu, Visvanathan Ramesh, Martin Pellkofer und Thomas Koehler. „METHOD FOR TRAFFIC SIGN DETECTION“. Pat. WO/2006/022910. März 2006.
- [Bec09] Lars-Peter Becker. „METHOD AND APPARATUS FOR RECOGNIZING TRAFFIC SIGNS“. Pat. WO/2009/013223. Jan. 2009.
- [BG10] Jochen Bauer und Jan Giebel. „DRIVER ASSISTANCE SYSTEM FOR TRAFFIC SIGN RECOGNITION“. Pat. 20100141476. Juni 2010.
- [Jan03] Holger Janssen. „Method and device for traffic sign recognition and navigation“. Pat. 6560529. Mai 2003.
- [NM11] Koba Natroshvili und Ayyappan Mani. „Method and device for classifying a traffic sign“. Pat. EP2363827. Sep. 2011.
- [Ste+08] Gideon P Stein, Uri Worlfovitz, Ori Shachar und Yoav Taieb. „Detecting and recognizing traffic signs“. Pat. EP1930863. Juni 2008.
- [Ste+11] Gideon P Stein, Ori Shachar, Yoav Taieb und Uri Wolfvitz. „Detecting and recognizing traffic signs“. Pat. 8064643. Nov. 2011.
- [WCH13] Charles Wang, Zhang Yankun Chuanming und Chuyang Hong. „SYSTEM AND METHOD FOR DETECTING AND RECOGNIZING RECTANGULAR TRAFFIC SIGNS“. Pat. WO/2013/026205. Feb. 2013.

A.5. Abbildungen

- [Com10] Wikimedia Commons. *Category:Diagrams of road signs of Germany* — Wikimedia Commons. 2010. URL: http://commons.wikimedia.org/w/index.php?title=Category:Diagrams_of_road_signs_of_Germany.
- [Art13] Arthur Shlain. *The Noun Project – Traffic Light Icon*. 2013. URL: <http://thenounproject.com/noun/traffic-light/#icon-No7813>.
- [Ben13] Benjamin Orlovski. *The Noun Project – Car*. 2013. URL: <http://thenounproject.com/noun/car/#icon-No7499>.
- [Chr13] Chris Robinson. *The Noun Project – Camera Icon*. 2013. URL: <http://thenounproject.com/noun/camera/#icon-No8835>.
- [Jas13] Jason D. Rowley. *The Noun Project – Connection Icon*. 2013. URL: <http://thenounproject.com/noun/connection/#icon-No19129>.
- [Mic13a] Michael Rowe. *The Noun Project – Camera Rotation*. 2013. URL: <http://thenounproject.com/noun/camera-rotation/#icon-No5205>.
- [Mic13b] Michael Rowe. *The Noun Project – Eye Icon*. 2013. URL: <http://thenounproject.com/noun/eye/#icon-No19791>.
- [Ope12] OpenStreetMap-Mitwirkende. *OpenStreetMap Open Data*. Lizenz: Open Data Commons Open Database License (ODbL), <http://opendatacommons.org/licenses/odbl/1.0/>. 2012.
- [Ré13] Rémy Médard. *The Noun Project – Map*. 2013. URL: <http://thenounproject.com/noun/map/#icon-No10319>.
- [Ste13] Steve Laing. *The Noun Project – Motion Camera*. 2013. URL: <http://thenounproject.com/noun/motion-camera/#icon-No9332>.
- [Val13] Valentina Piccione. *The Noun Project – Tree*. 2013. URL: <http://thenounproject.com/noun/tree/#icon-No1535>.
- [Vol13] Volodin Anton. *The Noun Project – Speedometer*. 2013. URL: <http://thenounproject.com/noun/speedometer/#icon-No17191>.
- [Wik12] Wikimedia Commons. *File:Proměnná dopravní značka, limit 100.jpg* — Wikimedia Commons. 2012. URL: http://commons.wikimedia.org/w/index.php?title=File:Prom%C4%9Bnn%C3%A1_dopravn%C3%AD_zna%C4%8Dka,_limit_100.jpg.