TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Verkehrswissenschaften „Friedrich List “, Institut für Verkehrstelematik

Studienarbeit

Untersuchungen zur Objektklassifikation in digitalen Bildfolgen des Straßenverkehrs

eingereicht von

Manuela Knaak

geb. am 10. Juni 1987 in Erlangen

Hochschullehrer: **Prof. Dr.-Ing. O. Michler**
Betreuer: **Dipl. Inf. H. Saul (DLR)**
Dipl. Ing. U. Gosda (TUD)

Dresden, den 08. Mai 2013 _____

Manuela Knaak

Bibliographischer Nachweis

Manuela Knaak
Untersuchungen zur Objektklassifikation in digitalen Bildfolgen des Straßenverkehrs
08. Mai 2013
Technische Universität Dresden
Fakultät Verkehrswissenschaften „Friedrich List“
Institut für Verkehrstelematik
Professur Informationstechnik für Verkehrssysteme
Studienarbeit

Autorenreferat:
Ziel dieser Arbeit ist die Untersuchung verschiedener Ansätze zur Klassifikation von Verkehrsobjekten in Bildfolgen aus Verkehrsvideos und die Implementierung ausgewählter Verfahren. Der Prozess der Objektklassifikation als Ganzes wird grob skizziert und die einzelnen Arbeitsschritte Merkmalsbestimmung, Merkmalsdeskription sowie Klassifikation geschildert. Der Stand der Technik wird auf Basis einer Literaturrecherche dargelegt. Verfahren der Merkmalsdeskription, die hierbei zur Anwendung kamen, werden vorgestellt. Basierend auf den Erkenntnissen der Literaturrecherche, werden Fourier Deskriptoren für die Merkmalsbeschreibung und zwei Clusteringverfahren sowie drei Klassifikationsverfahren (MDC, k-NN, SVM) ausgewählt. Grundlegende Verfahren, auf die sich die Arbeit stützt, wie z. B. die Fourier Transformation, werden erläutert. Die ausgewählten Verfahren werden in Matlab implementiert und ihre Leistungsfähigkeit sowie ihr Potential durch Experimente abgeschätzt. Als Datenbasis für die Experimente dienen 1.757 Objektkonturen, die aus Bildfolgen realer Verkehrsszenen extrahiert und von Hand vorklassifiziert (annotiert) wurden.

Thesen zur Studienarbeit

1. Herkömmliche Technologien zur Objekterkennung und -klassifikation sind zu unflexibel und teuer.
2. Kamerabasierte Verfahren haben viel Potential, müssen aber noch zuverlässiger werden.
3. Die Klassifikation von Verkehrsobjekten auf Basis ihrer Kontur ist möglich und sinnvoll.
4. Shape Signatures eignen sich gut als Repräsentanten der Objektkontur.
5. Fourier Deskriptoren können als Basis für eine robuste Objektklassifikation fungieren, die auch unter realen Bedingungen im Verkehrsumfeld leistungsfähig ist.
6. Bereits einige wenige Fourier Deskriptoren spannen einen Merkmalsraum auf, in dem ähnliche Objekte geringe Distanzen zueinander aufweisen.
7. Clustering Verfahren können die inneren Strukturen der Objektdaten aufzeigen.
8. Die Klassifikatoren Minimum-Distance-Classifier, k-Nearest-Neighbor Klassifikator und Support-Vector-Machine sind in der Lage den mehrdimensionalen Merkmalsraum so zu unterteilen, dass ähnliche Objekte als solche erkannt und Klassen mit ähnlichen charakteristischen Eigenschaften zugeordnet werden.
9. Anhand von Bildfolgen aus realen Verkehrsvideos kann der Ansatz der Objektklassifikation auf Basis von Fourier Deskriptoren überprüft und seine Eignung bestätigt werden.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Abkürzungen

3D	Dreidimensional
Acc	Accuracy
BOF	Bag of Features
BV	Bildverarbeitung
CPO	Classifier Performance Object
DFT	Diskrete Fourier Transformation
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DoG	Difference of Gaussians
FD	Fourier Deskriptor
FFT	Fast Fourier Transformation
FPR	False Positive Rate/Ratio
fps	frames per second
FT	Fourier Transformation
GLOH	Gradient Location-Orientation Histogram
HOG	Histogram of Oriented Gradients
k-NN	k-Nearest-Neighbor
KV	Kreuzvalidierung
LOC	Local Orientation Coding
LOO-KV	Leave-One-Out-KV
LS	Least Squares
MDC	Minimum Distance Classifier
MLP	Multilayer Perceptron
OCR	Optical Character Recognititon
OHS	Optimal Separating Hyperplane
OVA	One-Versus-All
OVO	One-Versus-One
PCA	Principal Component Analysis
PNG	Portable Network Graphiks Format
PPV	Positive Predictive Value
QP	Quadratische Programmierung
RBF	(Gaussian) Radial Basis Function
ROC	Receiver Operating Characteristic
RoI	Regions of Interest
SIFT	Scale-Invariant Feature Transform
SMO	Sequentielle Minimale Optimierung
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TAR	Triangle-Area-Representation
TNR	True Negative Rate/Ratio
TPR	True Positive Rate/Ratio
WT	Wavelet Transformation

Verzeichnis der Formelzeichen

$\mathbb{R}$	Reelle Zahlen
j	Imaginäre Einheit
e	Eulersche Zahl
Im	Imaginärteil einer komplexen Zahl
Re	Realteil einer komplexen Zahl
π	Kreiszahl Pi
$\mathcal{O}$	Landau Symbol
θ	Winkel
ϕ	Winkel
σ	Skalierungsfaktor des RBF Kernels für SVM
$\mathscr{H}$	Hyperebene
$\vec{\omega}$	Normalenvektor
ζ	Schlupfvariable (SVM)
κ	Kappa-Koeffizient

Kapitel 1

Einleitung

Das menschliche visuelle System bewältigt tagtäglich eine große Menge und Bandbreite an Aufgaben. Gestik und Mimik der Mitmenschen erkennen und interpretieren, verschiedene Handschriften lesen, Personen wiedererkennen, die rote Ampel bemerken, ein Fahrrad von einem Motorrad unterscheiden - die Liste könnte nahezu unbegrenzt fortgeführt werden. Für den Menschen ist all das kein Problem, weshalb diese Fähigkeiten als selbstverständlich empfunden werden. Für ein technisches System jedoch stellt die Erkennung und Einordnung von Situationen und Objekten eine große Herausforderung dar. Trotz der Komplexität der Aufgabe treibt das Streben nach Automatisierung seit einigen Jahren die Forschung im Bereich der computergestützten, automatischen Objekterkennung voran. Ob in der Robotik, der industriellen Fertigung, in Überwachungssystemen, Augmented Reality Anwendungen, medizinischen Assistenzsystemen, der Sportanalyse oder im Verkehrswesen - die Vielfalt der Einsatzgebiete und das Interesse an dieser Technologie sind sehr groß. Allein im Verkehrswesen finden sich viele Anwendungen, wie im Bereich der Fahrerassistenzsysteme, der Verkehrsüberwachung, Verkehrsdatenerfassung oder dem Verkehrsmanagement. Ganz konkrete Beispiele für die Nutzung automatischer Objektklassifikation sind die Mautberechnung an sogenannten Mautbrücken, die Erkennung von Fahrzeugen auf LKW Stellplätzen an Autobahnen, Gefahrraumfreimeldung an Bahnübergängen und Bahnsteigen, Stauwarnsysteme oder automatisierte Verkehrszählungen.

1.1 Motivation und Zielsetzung

Im Verkehrswesen kommen für die Objekterkennung und -klassifikation verschiedene Technologien zum Einsatz, wie Induktionsschleifen, Infrarotsensoren, Radar oder Ultraschallsensoren. Auch das große Potential von Kamerasystemen für das Verkehrswesen wurde erkannt. Obwohl die Bildverarbeitung bereits enorme Fortschritte gemacht hat, so gilt eine zuverlässige Objektklassifikation im Verkehrssektor noch immer als Herausforderung. Als Gründe hierfür können folgende Punkte genannt werden:

- Schwierige Umgebungseigenschaften (Im Freien, daher unbeständige und z. T. schwierige Wetterbedingungen, wechselnde Lichtverhältnisse, Schatten)
- Mannigfaltigkeit an Farben und Formen
- Vielfalt an Blickwinkeln (va. im Individualverkehr)
- Häufig geringe Bildqualität (Kostengründe)
- Oft Echtzeitfähigkeit nötig
- Bewegliche Objekte (Bewegungsunschärfe, Änderungen in Perspektive und Größe eines Objektes)
- (Teil-)Verdeckungen
- Spiegelnde Oberflächen/Lichtreflexe
- Verändertes Szenario bei Dunkelheit (Blendung durch Scheinwerfer bei unzureichender Straßenausleuchtung, Fußgängererkennung erschwert)
- Bei sicherheitskritischen Anwendungen hohe Anforderungen an Zuverlässigkeit und Verfügbarkeit

Den besonderen Anforderungen stehen jedoch die bedeutenden Vorteile der Technologie gegenüber. Zu diesen zählen Kosteneffizienz, Flexibilität, Möglichkeit der Extraktion verschiedener Informationen aus den Bilddaten (z. B. räumlich-zeitliche Verkehrskenngrößen, Erkennung auch von nicht-metallischen Objekten), unkomplizierte Erweiterung, einfache Installation (i. d. R. keine Baumaßnahmen nötig) und das große Potential, dass sich durch die laufende Weiterentwicklung der Bildverarbeitung ergibt.

Da in vielen Anwendungsfällen kostenintensive Verfahren unwirtschaftlich sind und der Druck, Einsparpotentiale zu finden und Verfahren effizienter zu gestalten,

kontinuierlich steigt, stellt die preiswerte Objektklassifikation mittels Kameradaten eine attraktive Alternative zu herkömmlichen Verfahren dar. Aus diesem Grund wird seit einigen Jahren vermehrt an Ansätzen geforscht, welche diese Technologie zuverlässiger und leistungsfähiger machen.

Diese Arbeit soll einen Beitrag zur Weiterentwicklung der kamerabasierten Objektklassifikation leisten. Konkret ist das Ziel dieser Studienarbeit die Untersuchung verschiedener Ansätze zur Klassifikation von Verkehrsobjekten in Bildfolgen aus Verkehrsvideos und die Implementierung von ausgewählten Verfahren. Hierfür werden zunächst geeignete Merkmale der Verkehrsobjekte im Bildraum identifiziert und Deskriptoren zur Erfassung dieser Merkmale abgeleitet. Auf Basis der Deskriptoren werden verschiedene Ansätze zur Objektklassifikation geprüft. Die Deskriptor-Klassifikator-Kombinationen werden anhand von Bildfolgen aus realen Verkehrsszenen getestet und bewertet.

1.2 Aufbau der Arbeit

Nachdem grundlegende Überlegungen sowie das Ziel der Arbeit vorgestellt wurden, werden in Kapitel 2 Grundlagen in Bezug auf Prozesse der Objektklassifikation, Merkmale, Merkmalsbeschreibung und Klassifikation behandelt. Zudem werden Ergebnisse der Literaturrecherche vorgestellt und ein Überblick über den aktuellen Stand der Technik, bezogen auf Deskriptor-Klassifikator-Kombinationen, vermittelt. Das Kapitel schließt mit der Benennung der für diese Arbeit gewählten Merkmale und Verfahren. Kapitel 3 dient der Erläuterung aller in der Arbeit verwendeten Verfahren - von der Konturrepräsentation bis hin zu Verfahren zur Bewertung der Klassifikation. Die Implementierung der Verfahren aus Kapitel 3 sowie Experimente werden in Kapitel 4 dargestellt. In Kapitel 5 werden die Ergebnisse der Experimente analysiert und Gründe für Fehlklassifikationen aufgezeigt. Kapitel 6 dient der Zusammenfassung und beinhaltet ein abschließendes Fazit.

Objektklassifikation ist ein international diskutiertes Thema, weshalb der Gebrauch englischer Fachbegriffe üblich ist. In dieser Arbeit werden Termini in der Regel nicht ins Deutsche übersetzt, da die Bedeutung dadurch verfälscht werden könnte.

Kapitel 2

Grundlagen und Stand der Technik

2.1 Einordnung in den Prozess der Objektklassifikation

Die Aufgabe der Objektklassifikation auf Basis von digitalen Bildfolgen des Straßenverkehrs lässt sich in eine Reihe von Arbeitsschritten unterteilen (vgl. Abb. 2.1).

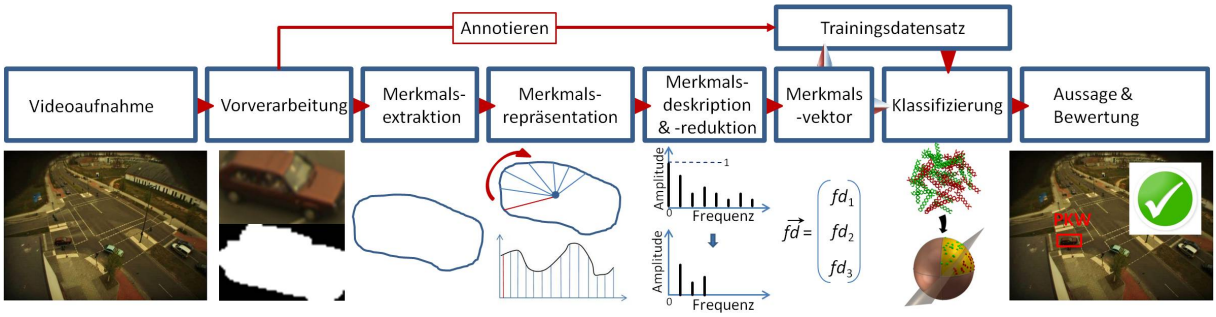

Abbildung 2.1: Prozess der Objektklassifikation.

Zunächst wird eine Datenbasis erzeugt, indem Verkehrsszenen mit Kameras aufgenommen werden. Die Bildsequenzen werden dann einer Vorverarbeitung unterzogen, die in erster Linie der Beseitigung bzw. Reduktion von Störungen in den Bildsignalen dient. Dieser Schritt kann eine große Bandbreite an Operationen beinhalten, die alle dem Ziel dienen, das Bildmaterial für die weitere Verarbeitung aufzubereiten. So werden durch die Bildaufnahme bedingte radiometrische[1] (z. B. Rauschen) und geometrische Fehler (z. B. Verzerrungen) oder inhomogene Beleuchtung korrigiert. Des-

[1] Die Messung elektromagnetischer Strahlung betreffend.

weiteren umfasst die Vorverarbeitung häufig eine Modifikation des Farbmodells der Signale. Anschließend wird das Objekt vom Hintergrund separiert. Dies kann mit Hilfe eines Hintergrundschätzers realisiert werden und wird als Segmentierung bezeichnet. In Anwendungsfällen mit bewegten Objekten lässt sich der Hintergrund über Differenzbildanalyse ermitteln. Für die weitere Bearbeitung wird häufig ein rechteckiger Bildausschnitt um das Objekt erzeugt - die sogenannte „Bounding Box“. Ausgewählte Eigenschaften werden nun extrahiert (Feature Extraction) und als Merkmal dargestellt (Feature Representation). Merkmale könnten beispielsweise Farbinformationen, Abmessungen oder die Kontur sein. Bei der sich anschließenden Merkmalsdeskription (Feature Description) werden die Merkmale in eine Form gebracht, in der sie gewünschte Eigenschaften wie Invarianz oder Robustheit besitzen und vor allem eine eindeutige Klassenzuordnung ermöglichen. Es folgt die Reduktion der Merkmale (Feature Reduction), mit dem Ziel nur so viele Merkmalsdeskriptoren wie nötig zu verwenden. Dies reduziert die Rechenintensität und hilft Überanpassung (Overfitting) zu vermeiden. Die reduzierten Merkmalsdeskriptoren werden üblicherweise in einem sogenannten Merkmalsvektor (Feature Vector) zusammengefasst. Im nächsten Schritt erfolgt die Klassifizierung. Zunächst muss der Klassifikator jedoch justiert werden, indem sie ein sogenanntes „Training“ durchlaufen. Für Klassifikatoren mit überwachtem Lernen müssen Merkmalsvektoren mit bekannter Klassenzuordnung als Trainingsdaten verwendet werden. Mit dem trainierten (angelernten) Klassifikator können nun die Klassen der Testobjekte geschätzt werden. Der letzte Arbeitsschritt ist die Analyse der Ergebnisse und die Bewertung der Klassifikation. Gegebenenfalls müssen nun noch Nachjustierungen vorgenommen werden. [J̈10]

Diese Arbeit setzt beim Arbeitsschritt der Merkmalsdeskription ein. Als Datengrundlage dienen Bildsequenzen aus Verkehrsszenen, die von Kameras des DLR in Berlin-Adlershof stammen und bereits vorverarbeitet wurden. Zur Verfügung stehen bereits die vom Hintergrundschätzer ermittelte Objektkontur, die konvexe Hülle, teilweise Skalierungsinformationen sowie die Bounding Boxes mit dem Objekt als Farb- und Grauwertbild. Basierend auf den vorliegenden Informationen werden im Rahmen dieser Arbeit ein Merkmal ausgewählt sowie die Merkmalsdeskription, Merkmalsreduktion, Klassifikation und eine Auswertung durchgeführt.

2.2 Objektmerkmale im Bild

Die Basis einer erfolgreichen Objektklassifikation ist die Extraktion distinkter Merkmale, die in geeigneter Form repräsentiert und beschrieben werden. Anhand der Datengrundlage und der Problemstellung müssen also geeignete Merkmale gefunden werden, die die charakteristischen Eigenschaften der Klassen widerspiegeln und somit eine Abgrenzung der Klassen ermöglichen [BÖ2] . Merkmale stellen somit eine komprimierte Form der Bildinformationen dar. Es gibt verschiedene Merkmale die ein Objekt im Bild beschreiben können. Hierzu zählen beispielsweise Form, Größe, Farbe, Textur, Geschwindigkeit sowie daraus abgeleitete Beschreibungsvektoren.

Die folgenden Merkmale sind wichtiger Bestandteil der menschlichen Wahrnehmung und werden daher auf ihre Anwendbarkeit im Hinblick auf das vorliegende Klassifikationsproblem geprüft.

2.2.1 Kontur

Die Kontur eines Objektes ist eines der Kernerkennungsmerkmale sowohl für die menschliche Wahrnehmung als auch für viele Anwendungen im Bereich der digitalen Bildverarbeitung. Die Kontur kann auf verschiedene Art und Weise repräsentiert werden. Besonders beliebt sind „Chain Codes“ (Kettenkodierung) und „Shape Signatures“. Chain Codes stellen die Kontur als Folge gleichlanger gerader Segmente dar. Abb. 2.2a zeigt den Chain Code nach Freeman, der acht Richtungen unterscheidet. Shape Signatures sind eindimensionale Funktionen, die die Kontur repräsentieren. Es gibt viele Möglichkeiten diese Funktionen zu erzeugen. Eine davon ist die Centroid Distance Function, die in Abb. 2.2b dargestellt ist und das Objekt über die Abstände von Kontur zum Schwerpunkt definiert. Problematisch sind bei allen Konturbasierten Merkmalen Effekte wie Rauschen, Verdeckungen und der Informationsverlust durch die Projektion der dreidimensionalen Wirklichkeit auf ein zweidimensionales Bild.

2.2.2 Einfache konturbasierte Merkmale

Von der Kontur abgeleitete Merkmale, die auf Grund ihrer Einfachheit auch als „Low Level Features“ bezeichnet werden sind beispielsweise Umfang, Ausdehnung des Seg-

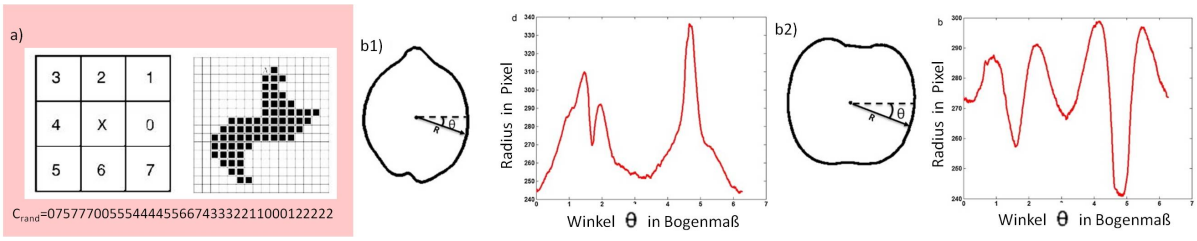

Abbildung 2.2: Kontur a) Freeman[Pan11] b) Centroid Distances[MBG+11]

mentes entlang seiner Hauptachsen, Größe des kleinsten umschließenden Rechtecks (Bounding Box), Exzentrizität (Kreisähnlichkeit) und abgeleitete Merkmale wie die Kompaktheit, Flächeninhalt oder die Flächendifferenz zwischen Segmentfläche und konvexer Hülle (vgl. Abb. 2.3) [Poh06].

Merkmale dieser Art werden fast ausschließlich in Kombination verwendet. [Bro04] vereint beispielsweise Größenmerkmale wie Umfang, Fläche und Hauptachsen mit der Geschwindigkeit zu einem Merkmalsvektor. Diese Merkmalskombination soll der Blickwinkelunabhängigen Erkennung von Fußgängern und Fahrzeugen dienen. In [KB12] werden Höhe, Breite und der Winkel zwischen Heckleuchten und Nummernschild zur Erkennung von Fahrzeugtypen von hinten verwendet. Die Objektabmessungen werden hierbei auf die Größe des Nummernschilds normiert.

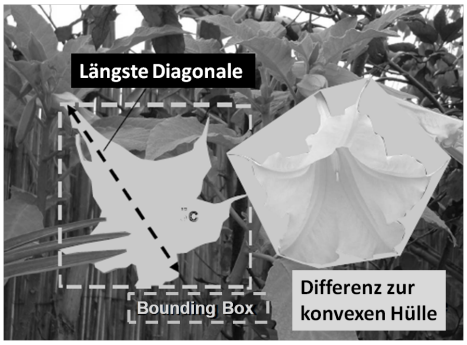

Abbildung 2.3: Einfache konturbasierte Merkmale. [Poh06]

2.2.3 Farbe

Für das menschliche Auge ist Farbe ein sehr wichtiges, einfach zu erfassendes und einprägsames Objektmerkmal. Die Farbe von Objekten in Bildfolgen automatisch zu

erkennen ist hingegen nicht trivial, sodass für diesen Schritt an sich bereits eine Klassifikation notwendig ist [CPFA09] . Der am weitesten verbreitete Ansatz zur Repräsentation der Farben innerhalb eines Bildes sind Farbhistogramme [KÖ5] . In dieser Wahrscheinlichkeitsverteilung ist kodiert, wie oft die verschiedenen Farbwerte in einem Bild auftreten [KÖ5] . Die nahezu unbegrenzte Farbvielfalt in Bezug auf Fußgänger und Fahrzeuge führt jedoch dazu, dass dieses Merkmal zur Unterscheidung von Fahrzeugklassen weitestgehend ungeeignet ist. In der Literatur wird die Fahrzeugfarbe teilweise unterstützend als Merkmal verwendet, um beispielsweise bestimmte Fahrzeuge innerhalb eines Kreuzungsbereichs mit einer gewissen Wahrscheinlichkeit wiedererkennen zu können (vgl. [KRM05]). In [USS01] wird die Information „Hautfarbe“ im Bild zur Fußgängererkennung verwendet. Das dazu nötige hochaufgelöste Bildmaterial steht in der Praxis jedoch häufig nicht zur Verfügung.

Insgesamt findet das Merkmal Farbe in erster Linie im Bereich Tracking Anwendung und weniger im Bereich der Klassifikation von Verkehrsobjekten [Bro04] . Aus diesem Grund wird das Merkmal in dieser Arbeit im Folgenden nicht weiter betrachtet.

2.2.4 Textur

Die Textur ist eine wichtige Komponente der menschlichen Wahrnehmung. Menschen können sie meist problemlos einordnen, es ist aber sehr schwierig sie eindeutig zu definieren [HR04] . Dies erschwert die Nutzung von Textureigenschaften als Merkmal. Um die Textur numerisch beschreiben zu können, können auf Basis von sogenannten „Grey Level Co-occurence Matrices“ Kennzahlen wie die Entropie, Energie, Kontrast oder Homogenität berechnet werden [HR04] . In einigen Anwendungsfällen wird das Merkmal auch im Bereich des Straßenverkehrs genutzt. So verwendet [GM07] beispielsweise texturbasierte Merkmale zur binären Klassifikation von Fußgängern.

2.2.5 Topologische Merkmale

Topologische Merkmale weisen umfangreiche Invarianzeigenschaften auf. Sie sind völlig unabhängig von Verzerrungen und anderen Veränderungen, solange das Objekt nicht zertrennt oder ergänzt wird. Merkmale dieser Art sind beispielsweise die An-

zahl an Löchern in einem Objekt oder die Eulerzahl, die sich aus der Differenz der verbundenen Teile des Objektes und der Lochzahl ergibt (vgl. Abb. 2.4). Diese Merkmale eignen sich nur zur Erkennung sehr einfacher Formen, wie Stanzteile beispielsweise. [Poh06]

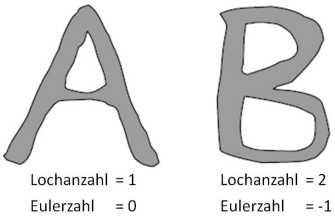

Abbildung 2.4: Topologische Merkmale. [Poh06]

2.2.6 Bewegung

Bewegung ist ein sehr starkes Merkmal und eignet sich hervorragend um Fahrzeuge und Fußgänger von der statischen Umgebung abzugrenzen. In erster Linie wird die Bewegung der Objekte daher zur Detektion der Objekte genutzt - in dieser Arbeit macht sich beispielsweise der Hintergrundschätzer dieses Merkmal zu Nutze um zu ermitteln wo im Frame ein interessierendes Objekt vorhanden ist. Aber auch für die Klassifikation ist die Berücksichtigung der Bewegung hilfreich. So kann man Geschwindigkeitsgrenzen festlegen, ab denen bestimmte Verkehrsteilnehmer wie Fußgänger oder Radfahrer ausgeschlossen werden können. Auch lässt sich vermuten, dass Objekte, die sich über einen längeren Zeitraum in Fuß- bzw. Fahrrad-üblichen Geschwindigkeiten bewegen, auch der entsprechenden Klasse angehören. [Bro04] nennt zudem die periodischen Bewegungen von gehenden oder rennenden Personen als markantes Merkmal.

Die Ermittlung der Objektgeschwindigkeit ist für viele Anwendungen jedoch problematisch. Eine einzelne Kamera kann lediglich eine Projektion der Wirklichkeit erfassen. Dies bewirkt den Effekt der Bewegungsparallaxe, der zu einer verfälschten Geschwindigkeitswahrnehmung führt. So muss bei der Bestimmung der Objektgeschwindigkeiten beachtet werden, dass weit entfernte Objekte sich bei gleicher Geschwindigkeit langsamer durch das Bild bewegen, als nahgelegene Objekte. Auch un-

terschiedliche Bewegungsrichtungen führen zu einer verzerrten Geschwindigkeitsmessung im Bild. Zum Beispiel würde für ein Fahrzeug, dass auf die Kamera zufährt die Geschwindigkeit Null ermittelt werden, während für den quer vor der Kamera kreuzenden Fußgänger eine deutlich höhere Geschwindigkeit festgestellt würde. Für Videos mit großen Entfernungs- und Blickwinkeldiskrepanzen ist die Berücksichtigung dieser Problematik essentiell.[BÖ2]

Da in dieser Arbeit mit Bildfolgen gearbeitet wird, die jeweils als vom Video losgelöstes Szenario betrachtet werden, liegen keine Geschwindigkeitsinformationen vor, sodass das Merkmal keine Anwendung finden kann. Die Verwendung der Geschwindigkeit als Ergänzung zu anderen Merkmalen kann aber als vielversprechend bezeichnet werden.

2.2.7 Fazit zu Objektmerkmalen im Bild

Merkmale können global, auf das ganze Bild bzw. Objekt bezogen, oder lokal für bestimmte Regionen ermittelt werden. Die menschliche Wahrnehmung orientiert sich bei der Erkennung von Fahrzeugen vor allem an der globalen Form des Objektes und an lokalen Schlüsselpunkten wie Scheiben, Scheinwerfer oder Rädern. Bedingt durch die beweglichen Gliedmaßen, können Fußgänger beispielsweise anhand ihrer veränderlichen Kontur identifiziert werden. Bei der Suche nach geeigneten Objektmerkmalen für die automatische (computergestützte) Objekterkennung orientiert man sich oft an diesen vom Menschen als charakteristisch empfundenen Merkmalen. Aus diesem Grund basiert ein Großteil der Algorithmen zur Erkennung von Verkehrsteilnehmern auf Konturmerkmalen oder auf Verfahren die lokal Bildmerkmale wie Kontrast extrahieren.

Die ausgewählten Merkmale könnten nun direkt als Merkmalsdeskriptor verwendet werden. Bei Verwendung von mehreren Merkmalen werden diese in einem Merkmalsvektor (engl. Feature Vector) zusammengefasst. In der Regel erfüllen die genannten Merkmale die Anforderungen an Merkmalsdeskriptoren nicht hinreichend gut und werden daher noch zu sogenannten „High-Level-Feature-Descriptors“ weiterverarbeitet. So kann für ein Objekt, das sich anhand seiner Centroid-Distance-Function von anderen Objekten abgrenzt, beispielsweise durch Transformation der Funktion in den Frequenzbereich eine robuste und teilinvariante Beschreibung des Merkmals „Kontur“

erreicht werden.

Im Folgenden soll die Idee der Merkmalsdeskriptoren kurz erläutert werden.

2.3 Merkmalsdeskriptoren

Merkmalsdeskriptoren sind Verfahren um klassencharakteristische Objektmerkmale in eine Form zu bringen, in der sie den Anforderungen der jeweiligen Klassifikationsaufgabe gerecht werden. Wichtig ist, Deskriptoren so zu gestalten, dass zwei Objekte nur dann einen identischen Deskriptor aufweisen, wenn sie (bezogen auf die interessierenden Eigenschaften) auch identisch sind. Deskriptoren von Objekten unterschiedlicher Klassen sollten daher möglichst große Unähnlichkeit aufweisen. Diese essentielle Eigenschaft wird als Kongruenz bezeichnet. [NA08]

Für eine möglichst universelle Nutzbarkeit ist es von Vorteil, wenn der Deskriptor invariant ist gegenüber Transformation und Deformation. In fast allen Fällen wird Skalierungs-, Translations- und Rotationsinvarianz gefordert. Für die Klassifikation von Fahrzeugen ist es wichtig Invarianzen so zu verwenden, dass Eigenschaften welche unabhängig von der Erkennungsaufgabe sind, herausgefiltert bzw. ignoriert werden. Da sich die Fahrzeuge in dem hier betrachteten Fall in unterschiedlichen Winkeln auf die Kamera zu bzw. weg bewegen, kommt es zu perspektivischen Verzerrungen sowie zu Veränderungen in Bezug auf Skalierung und Translation. Ein Deskriptor der nicht invariant gegenüber diesen Veränderungen ist, würde also für ein und dasselbe Objekt in jedem Frame einen anderen Deskriptor erstellen. Invarianzen führen somit zu einem teilweise gewollten Informationsverlust, der in einigen Situationen die Unterscheidung von Objekten aber auch erschwert oder gar verhindert. Bei der Erkennung von Verkehrsobjekten spielt die Objektgröße beispielsweise eine durchaus wichtige Rolle. Nutzt man einen skalierungsinvarianten Deskriptor, so ist es nicht mehr möglich, einen LKW von einem PKW anhand ihrer Größen zu unterscheiden. Reichen die verbleibenden Eigenschaften nicht zur Erkennung aus, muss auf die Invarianz ggf. verzichtet oder mehrere Klassifikatoren kombiniert werden. [NA08]

Neben Invarianz und Unterscheidbarkeit (Distinktivität) sollte ein Merkmalsdeskriptor die folgenden Eigenschaften aufweisen: [ZL04]

- Leichte Extrahierbarkeit der Merkmale
- Kompaktheit (Basis für Echtzeitfähigkeit)
- Stabilität (wenige unsichere/ad-hoc Faktoren)
- Geringe Rechenkomplexität, Recheneffizienz
- Effizientes Matching (z. B. durch grobes Vorfiltern vor feinem Matching mit allen Details)
- Speichereffizienz
- Robustheit gegenüber Bildrauschen, Belichtungsveränderungen und Entartung
- Breite Anwendbarkeit

Eine geschickte Wahl der betrachteten Merkmale ist entscheidend für gute Klassifikationsergebnisse. Die Merkmale müssen genau diejenigen Eigenschaften des Objektes widerspiegeln, die es von anderen Objekten unterscheidet und abgrenzt und durch die das Objekt eindeutig identifizierbar ist. Merkmale, die klassenintern unterschiedlich sind oder sich für ein Objekt verändern, sollen nicht in den Deskriptor integriert sein. Für ein möglichst gutes Klassifikationsergebnis ist daher ein individueller, aufgabenspezifischer Entwurf der Deskriptoren unabdingbar. [NA08]

Die Güte der Wahl der Deskriptoren lässt sich für die jeweiligen Anwendungsfälle kaum isoliert bestimmen, sondern nur in Abhängigkeit der in den vor- und nachgelagerten Arbeitsschritten verwendeten Verfahren. Das beste Ergebnis wird nicht durch Kombination der besten Einzelverfahren erreicht, sondern durch geschickte Kombination von Einzelverfahren zu einem geeigneten Gesamtkonzept. [NA08]

Aus der großen Zahl existierender Merkmalsdeskriptoren werden in Abschnitt 2.5 einige bereits für Anwendungen im Verkehrsbereich getestete Verfahren vorgestellt.

2.4 Klassifikatoren

Sind Merkmale und Merkmalsdeskriptor gut gewählt, so spannen sie einen Merkmalsraum auf, in welchem ähnliche Objekte nahe beieinander liegen und sogenannte „Cluster“ bilden. Die Dimension des Merkmalsraums entspricht der Dimension des Merkmalsvektors und somit der Anzahl an Deskriptoren. Um ein unbekanntes Objekt

klassifizieren zu können, müssen dem Klassifikator die charakteristischen Eigenschaften der Klassen bekannt sein. Auf Basis der Merkmalsdeskriptoren von sogenannten Trainingsobjekten generiert jeder Klassifikator die klassenspezifischen Merkmale auf seine Weise - dies wird als Training bezeichnet. Die Objektklassifikation unterscheidet Verfahren, welche kein Klassenvorwissen bezogen auf die Trainingsdaten benötigen und solche, bei denen Vorwissen im Trainingsprozess genutzt wird. Diese beiden Verfahren sind auch als unüberwachtes und überwachtes Lernen bekannt. Den trainierten bzw. angelernten Klassifikatoren werden die zu klassifizierenden Testdaten übergeben. Die Merkmalsvektoren der Testdaten werden auf Ähnlichkeit zu den klassenspezifischen Merkmalen überprüft. Ähnlichkeit kann nicht eindeutig berechnet werden. Um dennoch eine Aussage hierüber treffen zu können, verwendet man als Hilfsmittel Distanzmaße. In der Regel wird ein Testobjekt demnach jener Klasse zugeordnet, zu der die Distanz am geringsten ist. Das Ergebnis der Klassifizierung (Prozess) nennt man Klassifikation (Resultat). [ME03]

Es sollen an dieser Stelle kurz die wichtigsten Distanzmaße vorgestellt und ein grober Überblick über Klassifikationsverfahren mit unüberwachtem und überwachtem Lernen gegeben werden. Detaillierte Informationen sind den Abschnitten 3.3 und 3.4 zu entnehmen, in denen alle im Rahmen dieser Arbeit verwendeten Verfahren erläutert werden.

2.4.1 Distanzmaße

Es gibt verschiedene Distanzmaße, die zur Beurteilung von Ähnlichkeit herangezogen werden können. Grundsätzlich kann der Abstand zwischen zwei Punkten $d_{p,q}$ im n-dimensionalen Raum über die Minkowski Gleichung (2.1) bestimmt werden.

$$d_{p,q} = \left[\sum_{i=1}^{n} |p_i - q_i|^k \right]^{\frac{1}{k}} \tag{2.1}$$

wobei i der Index der Koordinaten ist. Die Wahl des Parameters k bestimmt die die Art des Abstands. Für k=1 ergibt sich die City-Block-Distanz (Manhattan-Distanz), die für binäre Daten als Hamming-Distanz bezeichnet wird. Die Wahl von k=2 ergibt die wohl bekannteste und meist genutzte Distanz - die euklidische Distanz.

Beispiele für weitere Distanzmaße, sind

- Minkowski Distanzen mit Werten $k > 2$
- Chebychev Distanz: Größte Distanz zwischen den Elementen zweier Vektoren. $d_{chebychev}(p, q) = \min_i (p_i - q_i)$.
- Jaccard Distanz: Verhältnis der Schnittmenge zweier Merkmalsvektoren zur Summe der Merkmale. Merkmale die in beiden Objekten nicht vorkommen, werden nicht berücksichtigt.
- Mahalanobis-Distanz: Entspricht der euklidischen Distanz, erweitert um die inverse Kovarianzmatrix. Eignung daher für korrelierte Daten

[Loh12]

2.4.2 Clustering

Clusteranalysen, oft auch Clustering-Algorithmen oder Clusterverfahren genannt, basieren auf unüberwachtem Lernen. Dies unterscheidet sie von der „Klassifikation“ im eigentlichen Sinne.

Der Kern des unüberwachten Lernens ist, dass dem Klassifikator beim Training keine Information über die Zielausgabe übergeben wird und keine Rückmeldung über die Korrektheit der Klassifikation erfolgt. Die Algorithmen versuchen in den Eingangsdaten Strukturen zu erkennen, beispielsweise um Datencluster zu identifizieren und eine Dimensionsreduktion zu erreichen. Vorteil dieser Lernverfahren ist, dass sie selbst dann anwendbar sind, wenn keinerlei Klassenvorwissen, wie Klassenanzahl oder Klassenzuordnung, vorhanden ist.

Ziel von Clusteranalysen ist es, Ähnlichkeitsstrukturen zu detektieren und Klassen von Objekten zu generieren, deren Merkmale Ähnlichkeiten aufweisen und die daher einen Cluster bilden. [ME03]

Es gibt eine Vielzahl verschiedener Clustering-Algorithmen, die grob in vier Typen unterteilt werden können: [Kle12]

- Hierarchische Verfahren (z. B. Hierarschische Clusteranalyse, Darstellung mittels Dendrogramm)
- Partitionierende Verfahren (z. B. k-Means)

- Graphentheoretische/Dichtebasierte Verfahren (z. B. DBSCAN)
- Optimierungsverfahren/Andere (z. B. Fuzzy Clustering, neuronale Netze)

2.4.3 Klassifikatoren

Eine effizientere Methode des Lernens als das unüberwachte Lernen stellt das überwachte Lernen dar.

Hierbei wird dem Klassifikator die korrekte Klassifikation zu den Trainingsdaten mitgeteilt. Das Ergebnis der Klassifizierung wird über ein Fehlermaß mit der Zielausgabe verglichen. Das Fehlermaß, oft der mittlere quadratische Fehler, wird minimiert indem die Parameter des Klassifikators modifiziert werden. Bei Verfahren dieser Klasse sind bessere Ergebnisse zu erwarten als bei Verfahren mit unüberwachtem Lernen.

Als Beispiele für sehr einfache Klassifikatoren können der „Nearest Neighbor Classifier“ und der „Mean Distance Classifier“ angeführt werden. Sie ordnen ein Objekt der Klasse seines nächsten Nachbarn bzw. der Klasse des nächsten Klassenmittelpunktes zu.

Zu den fortgeschritteneren Verfahren zählen die folgenden Klassifikatoren:

- k-Nearest-Neighbor-Classifier (Zuordnung zur Klasse der k nächsten Nachbarn.)
- Polynomklassifikator (nicht-lineare Klassentrennung, z. B. durch Kreise.)
- Künstliche Neuronale Netze (Dem biologischen neuronalen Netz nachempfundene Strukturen, die einen Lernprozess ermöglichen.)
- Support-Vector-Machine (Dimensionserhöhung ermöglicht lineare Separierung nicht-linear trennbarer Daten. Trennebenen werden so gelegt, dass sich zwischen den Klassengrenzen ein möglichst breiter Rand ergibt.)

[TK09] und [ME03]

2.5 Stand der Technik

Die Objektklassifikation ist ein wichtiger Teilbereich der Bildverarbeitung. Auf Grund der breitgefächerten Anwendungsmöglichkeiten, die von Gesten- und Schrifterken-

nung über Tumordetektion bis hin zur Fahrzeugklassifikation reichen, wurde auf diesem Gebiet bereits einiges an Forschungsarbeit geleistet. Und dies, wie die Revolutionierung des Postverkehrs durch OCR („Optical Character Recognition“/automatische Schrifterkennung) zeigt, teilweise mit großem Erfolg. Für die meisten Anwendungsgebiete wurde bereits eine Vielzahl unterschiedlicher Kombinationen von Verfahren getestet und angewandt. Da es aber kein universell bestes Verfahren gibt, muss für jeden Anwendungsfall ein geeignetes Verfahren ermittelt werden. Zu beachten ist, dass die Kombination der Verfahren der einzelnen Arbeitsschritte auf dem Weg zu Objektklassifikation eine wichtige Rolle spielt. Bei der Recherche lag der Fokus auf unterschiedlichen Kombination von Merkmalsdeskriptoren und Klassifikatoren. Hierfür wurden Arbeiten aus allen Fachbereichen betrachtet. Der Schwerpunkt lag allerdings auf der Fahrzeug- und Fußgängerklassifikation.

In den nachfolgenden Abschnitten werden sechs Verfahren zur Merkmalsdeskription kurz erläutert und jeweils ein bis drei Arbeiten vorgestellt, die das jeweilige Verfahren zur Fahrzeug- und Fußgängerklassifikation nutzen. Soweit möglich, werden vorgelagerte Arbeitsschritte und der verwendete Klassifikator vorgestellt. Die Schlussfolgerungen für die hier vorliegende Aufgabenstellung werden für jeden Merkmalsdeskriptor kurz in einem Zwischenfazit zusammengefasst. Das Kapitel wird mit einer Schlussfolgerung in Bezug auf diese Arbeit abgeschlossen.

2.5.1 Scale-Invariant Feature Transform (SIFT)

Ein Deskriptor, der im Zusammenhang mit Fahrzeugklassifikation in den letzten Jahren häufig Verwendung fand, ist die skaleninvariante Merkmalstransformation, kurz SIFT (Scale-Invariant Feature Transform) genannt.

Theorie

SIFT ermöglicht das Erkennen und Extrahieren lokaler Merkmale im Bild (vgl. Abb. 2.5). Das Verfahren ist invariant gegenüber Skalierung, Rotation, Intensität und gegenüber moderaten affinen Transformationen. Die durch SIFT erkannten Merkmale weisen eine hohe Unverwechselbarkeit auf, weshalb selbst ein einzelnes Merkmal erfolgreich gegen eine große Datenmenge an Merkmalen gematcht werden kann. Diese

Eigenschaft macht SIFT zu einem geeigneten Deskriptor für Objekt- oder Szenenerkennung. [Bar12]

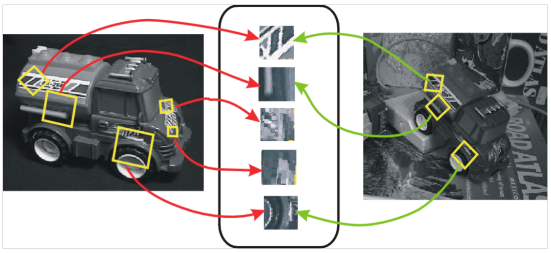

Abbildung 2.5: SIFT. [Bar12]

Die Generierung von Merkmalsdeskriptoren mittels SIFT gliedert sich in mehrere Arbeitsschritte, die der Detektion und Deskription der Merkmale dienen. [dB11]

1. Finden potentieller Schlüsselpunkte im Bild

   a) Erzeugen des Ort-Skalen-Raums mittels DoG (Difference of Gaussians)

   Das Bild wird durch Faltung mit Gaußschen Kernels unterschiedlich stark geglättet. Es werden die Differenzbilder der so erzeugten unterschiedlich scharfen Bilder berechnet. Dies wird auf mehreren Skalierungsniveaus (Bildgrößen) durchgeführt. Der DoG hat hierbei die Wirkung eines Bandpassfilters.

   In Abb. 2.6 ist dies schematisch und am Bildbeispiel visualisiert.

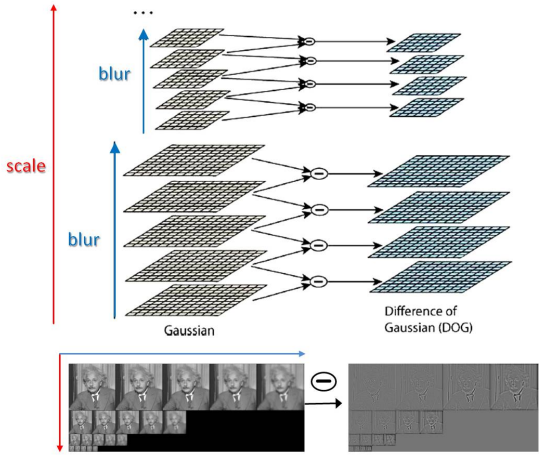

Abbildung 2.6: Erzeugen des Ort-Skalen-Raums mittels DoG. (Basis: [dB11])

b) Identifizierung lokaler Extrempunkte

Die potentiellen Schlüsselpunkte sind jene Stellen im Bild, die ein lokales Maximum bilden. Diese Extrempunkte werden ermittelt indem jedes Pixel mit seinen 26 Nachbarn verglichen wird. Abb.2.7(a1) zeigt in grüner Farbe die Nachbarpixel des mit einem Kreuz markierten Pixels. Die Nachbarpixel befinden sich also zum einen im Bild selber und zum anderen in den Bildern vor- und nachgelagerter Glättungsstufen.

Die lokalen Maxima im Beispielbild sind Abb. 2.7(a2) zu entnehmen.

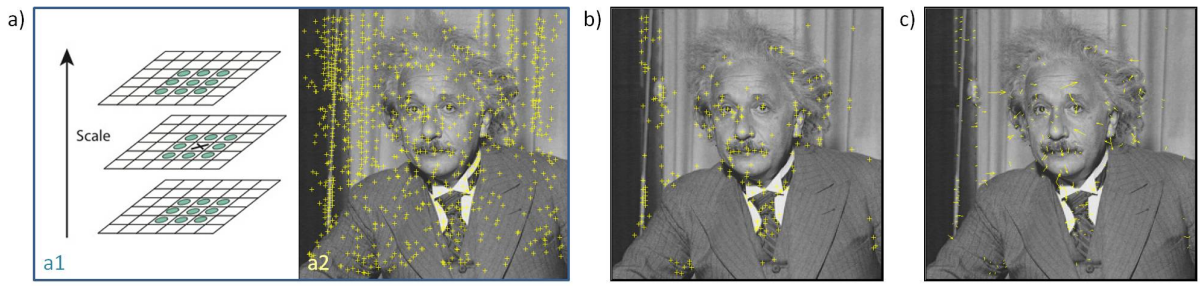

Abbildung 2.7: SIFT a) Lokale Maxima b) Schlüsselpunkte c) Gradienten.[dB11]

2. Bestimmung von Schlüsselpunkten

Da der erste Arbeitsschritt eine große Zahl an Kandidaten für Schlüsselpunkte erzeugt, wird deren Zahl im zweiten Schritt reduziert indem instabile Punkte nicht weiter betrachtet werden. Zu diesen gehören beispielsweise auf Kanten liegende Punkte oder solche mit geringem Kontrast. Abb. 2.7b zeigt das Beispielbild mit den gelb markierten Schlüsselpunkten.

3. Erzeugung von Gradientenhistogrammen

Aus der durch Betrag und Gaußfunktion gewichteten Gradientenausrichtung der Punkte, die sich in einem bestimmten Bereich um den Schlüsselpunkt befinden, wird ein sogenanntes Gradientenhistogramm erstellt (vgl. Abb. 2.7c).

4. Generierung des Schlüsselpunktdeskriptors

Die Merkmalsdeskriptoren dienen nicht der Beschreibung des Schlüsselpunktes selbst, sondern einer Region relativ zum Schlüsselpunkt. Die betrachtete Region umfasst in der Regel eine Fläche von 16x16 Pixel um den Schlüsselpunkt.

Wie in Abb. 2.8 ersichtlich ist, besteht der Deskriptor aus 4x4 Subregionen, deren Histogramme 8 Klassen für 8 unterschiedliche Richtungen enthalten. Für jeden Schlüsselpunkt entsteht somit ein Vektor der Größe $4x4x8 = 128$, der durch Normierung zum SIFT-Deskriptor wird.

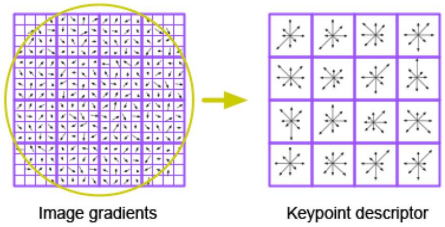

Abbildung 2.8: SIFT - Erzeugung der Deskriptoren. [Low04]

Der auf vielen Anwendungsgebieten geeignete SIFT Algorithmus wurde unter anderem zu den Algorithmen SURF, PCA-SIFT und GLOH weiterentwickelt.

So verwendet der sogenannte „Speeded Up Robust Features“ Algorithmus, kurz SURF, im Gegensatz zu SIFT Mittelwertfilter anstelle von Gauß-Filtern. Der Algorithmus zeichnet sich durch kurze Rechenzeiten und hohe Robustheit aus, hat dafür aber Nachteile in Bezug auf Invarianzeigenschaften.

PCA-SIFT hat eine, im Vergleich zu SIFT, andersartige Konstruktion der Schlüsselpunktdeskriptoren und bietet dadurch Verbesserungen in Bezug auf Rechendauer, Accuracy, Kompaktheit und Unverwechselbarkeit. PCA steht für „Principal Component Analysis“ und wird genutzt um den Bereich der Gradienten um den Schlüsselpunkt effizient darzustellen, sodass die Dimension des Feature Vektors deutlich geringer ist als bei SIFT. [KS06]

GLOH steht für „Gradient Location-Orientation Histogram“. Der Algorithmus ist robust und erhöht die Unverwechselbarkeit. Die Klasseneinteilung erfolgt nicht über Quadrate wie in Abb.2.8 , sondern in Form einer logarithmisch-polaren Struktur (Abb. 2.9). Es entsteht dabei ein größerer Deskriptor (272 Dimensionen), der dann mittels PCA wieder verkleinert wird. Der Deskriptor führt in der Regel zu besseren Ergebnissen als SIFT.

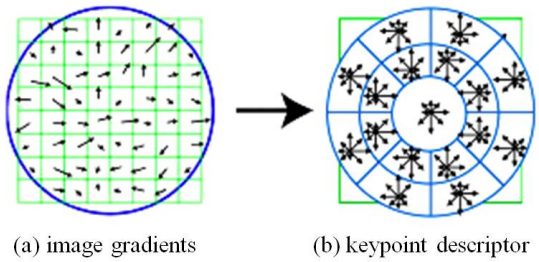

Abbildung 2.9: GLOH - Erzeugung der Deskriptoren. [dB11]

Beispielanwendungen

Nachfolgend soll anhand einiger Beispiele gezeigt werden, inwiefern der SIFT Algorithmus bereits zur Fahrzeugklassifikation herangezogen wurde.

In [NT12] werden markante Bildpunkte (engl. Keypoints) mit Hilfe einer SIFT beschrieben und für die Klassifikation von Fahrzeugen in Videosequenzen für die Mautberechnung genutzt. Obwohl nur eine midrange Überwachungskamera verwendet wurde, konnten Experimente eine sehr gute Erkennungsrate für Autos und Transporter nachweisen. Selbst Fahrzeuge größerer Ähnlichkeit, wie Taxen und Limousinen, konnten mit einer Fehlerquote von 7% noch gut unterschieden werden. Die Experimente wurden auf Basis von 530 Fahrzeugbildern aus einem schräg-seitlichen, jedoch konstanten, Blickwinkel durchgeführt.

[KS12] stützt die Fahrzeugklassifikation für ein Sicherheitssystem auf die Erkennung von Nummernschild, Form, Farbe und Markenlogo. Für Letzteres eignet sich laut [KS12] SIFT sehr gut. Basis hierfür ist entsprechend hochaufgelöstes Bildmaterial.

In [HJS09] kommt im Rahmen einer Objekterkennung, die auf der sogenannten „Bag of Features“ (BOF) basiert, die Merkmalsbeschreibung mittels SIFT zum Einsatz. Dem Prinzip der BOF liegt die Idee zu Grunde, dass jedes Objekt durch seine Teile darstellbar ist. Diese Teile, bei Autos z.B. könnten es unter anderem die Räder sein, werden in [HJS09] mit Hilfe von SIFT-Merkmalsvektoren beschrieben. Im Vergleich mit HOG führte BOF mit SIFT trotz höheren Rechen- und Speicheraufwands nur zu geringfügig besseren Ergebnissen. Die Kombination aus beiden (zusammen mit Chi^2 Kernel) ergab

jedoch sehr gute Ergebnisse. In dieser Betrachtung wurden beispielsweise Autos und Kühe erkannt.

Weitere Betrachtungen zu SIFT Deskriptoren für die Fahrzeugklassifikation sind in [MG05] und [DC10] (SURF) zu finden.

Zwischenfazit

Die Anwendungsbeispiele zeigen, dass sich SIFT Deskriptoren durchaus als Deskriptoren zur Fahrzeugklassifikation eignen. Jedoch sind Einschränkungen in Bezug auf die Eingangsdaten zu beachten. So wurde in den Testszenarien mit annähernd konstanten Blickwinkeln in Bezug auf die Fahrzeugachse bzw. hochaufgelöstem Bildmaterial gearbeitet. Es ist somit unklar, ob SIFT auch in weniger idealisierten Szenarien noch zu hinreichend guten Ergebnissen führt.

2.5.2 Histogram of oriented gradients (HOG)

Theorie

Das „Histogram of oriented gradients“ (HOG) nach [DT05] beschreibt Objekte in Bildern mit Hilfe der Intensitätsverteilung ihrer Gradienten. Ähnlich wie SIFT baut auch das HOG auf Histogrammen lokaler Gradienten auf. Hierzu wird das Bild in rechteckige (R-HOG) oder kreisförmige (C-HOG) Zellen unterteilt, für die jeweils über die Ausrichtung von Kanten, ein Histogramm erstellt wird. Um die Illuminationsinvarianz und die Genauigkeit zu erhöhen, wird anschließend eine Kontrast-Normierung durchgeführt. Die Histogramme bilden zusammen den Deskriptor. Vorteil von HOG sind Invarianz gegenüber einigen photometrischen und geometrischen Transformationen, da HOG mit lokalisierten Zellen arbeitet.

Die Vorgehensweise zur Erzeugung des Deskriptors ist in Abb. 2.10 dargestellt.

Im Gegensatz zu vielen anderen Deskriptoren, ist für die Berechnung des HOG Deskriptors eine Normierung der Farb- und Gammawerte vor Durchführung der eigentlichen Berechnungschritte nicht nötig, da die spätere Normierung des Deskriptors dasselbe bewirkt. Im Einzelnen gliedert sich die Vorgehensweise wie folgt:

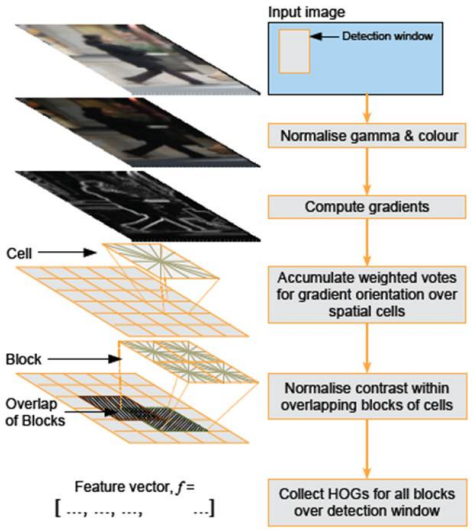

Abbildung 2.10: Ablauf HOG Deskriptor. [Hut09]

1. Berechnung der Gradienten

   Zunächst werden die Werte der Gradienten bestimmt. Hierzu eignet sich die horizontale und vertikale Anwendung einer eindimensionalen, zentrierten, punktdiskreten Maske in eine oder beide Richtungen. Das heißt, die Farb- und Intensitätsdaten des Bildes werden mit den Kernels $[-1, 0, 1]$ und $[-1, 0, 1]^T$ gefiltert.

2. Richtungs-Klasseneinteilung (engl. Orientation Binning)

   Im zweiten Schritt werden die Zellhistogramme erstellt. Jedes Pixel innerhalb einer Zelle (rechteckig oder kreisförmig) gibt eine gewichtete Stimme für eine Richtungsklasse ab. Diese befinden sich gleichmäßig verteilt zwischen $0°$ und $180°$ für vorzeichenlose Gradienten und zwischen $0°$ und $360°$ für vorzeichenbehaftete Gradienten. Zur Gewichtung eignet sich in der Regel der Betrag der Gradienten am besten.

3. Deskriptor Blöcke

   Um Invarianz gegenüber Belichtungsänderungen und Kontrast zu erzeugen, ist eine lokale Normierung nötig. Hierzu werden die Zellen zu größeren, räumlich zusammenhängenden Blöcken zusammengefasst. Der HOG Deskriptor ist der Vektor der Komponenten der normierten Zellhistogramme aller Blockregionen.

Da die Blöcke überlappend angeordnet sind, trägt jede Zelle mehrfach zum Gesamtdeskriptor bei.

4. Block Normierung

   Die Normierung kann beispielsweise mittels L2-Norm (2.2) realisiert werden.

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \tag{2.2}$$

   Mit v als nicht-normierten Vektor, der alle Histogramme eines bestimmten Blocks enthält und e als sehr kleinen, konstanten Wert.

Als Abgrenzung zu SIFT sei erwähnt, dass der R-HOG zwar große Ähnlichkeiten zu SIFT aufweist, die R-HOG Blocks jedoch in engem Raster in einer einzigen Skalierung berechnet werden. SIFT Deskriptren hingegen, werden an einigen wenigen, ausgewählten größeninvarianten Schlüsselpunkten berechnet und werden an ihrer Hauptrichtung ausgerichtet. Desweiteren werden bei R-HOG Blöcke als Gesamtheit gemeinsam genutzt, wohingegen bei SIFT die Informationen von den einzelnen Deskriptoren verwendet werden. [DT05]

Beispielanwendungen

Eine übliche Anwendung für HOG ist die Fußgängererkennung. Ein Beispiel dafür findet sich in [DT05]. Hier dient ein 3D HOG als Deskriptor. Für die Klassifikation wird eine lineare SVM verwendet. Es werden Schwierigkeiten bei der Erkennung in Fällen mit Verdeckung erwähnt. Außerdem ist eine aufrechte Haltung der Fußgänger für eine erfolgreiche Klassifikation nötig. Für den betrachteten Anwendungsfall schnitt das HOG Verfahren besser ab, als die im Vergleich betrachteten Wavelets. Es wurden keine Videosequenzen, sondern Bilder mit Fußgängern betrachtet. Es ist anzunehmen, dass mit HOG auch andere Verkehrsteilnehmer klassifiziert werden können. Abb. 2.11 zeigt wie die HOG Deskriptoren generiert wurden.

[HJS09] vergleicht die Eignung von HOG und BOF („Bag of features“) zur Erkennung von verschiedenen Objekten in Bildern. Die Objekte der verwendeten Daten-

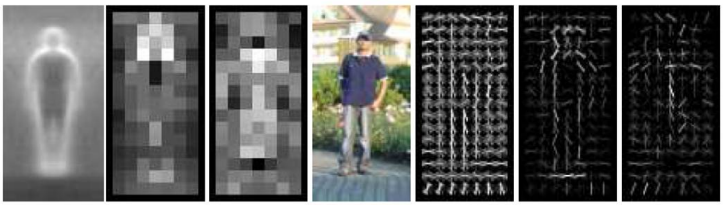

Abbildung 2.11: HOG Deskriptor für Fußgängererkennung. [DT05]

bank „PASCAL VOC 2007“[2] umfassen unter anderem verschiedene Verkehrsobjekte wie Autos, Fahrräder und Motorräder. HOG schnitt bei Tests etwas schlechter ab als BOF. Die besten Klassifikationsergebnisse brachte die Kombination aus HOG und BOF unter Verwendung einer nicht-linearen SVM mit Chi^2-Kernel als Scoring Klassifikator.

[MXHZ10] nutzt HOG Deskriptoren zur Fahrzeugklassifikation als Unterstützung für Fahrerassistenzsysteme. Die Kamera ist hierbei in ein Fahrzeug eingebaut und soll erkennen, ob es sich bei Objekten im Kamerabild um Fahrzeuge handelt oder nicht. Eine genauere Klassifikation wird in diesem Fall nicht angestrebt. Die Objektrepräsentation auf der die Objektbeschreibung basiert, erfolgt in [MXHZ10] mittels PCA, Local Orientation Coding (LOC), Haar-Wavelet und Gaborfiltern. Nach Generierung des Deskriptors mittels HOG wird die Objektklassifikation durchgeführt. Hierbei werden zwei Verfahren getestet: Lineare SVM und Neuronale Netze. Bei Tests in unterschiedlichen Verkehrsumgebungen, wie auf Autobahnen, in komplexer urbaner Umgebung oder in Situationen mit lokaler Verdeckung, erzielte das untersuchte Verfahren gute Klassifikationsergebnisse. Vorteil des Verfahrens ist, dass alle möglichen Bereiche untersucht werden und in allen Bereichen sofort entschieden wird, ob Fahrzeuge enthalten sind oder nicht.

Zwischenfazit

Die Ergebnisse der betrachteten Anwendungsbeispiele sind nicht direkt auf das in dieser Arbeit betrachtete Szenario übertragbar. Die Klassifikationsaufgaben in [DT05] und

[2] Datensatz der „Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Visual Object Classes(VOC) Challenge 2007“ des EU geförderten Exzellenz-Netzwerks PASCAL [EVGW+07]

[MXHZ10] waren auf das Erkennen eines bestimmten Verkehrsobjektes beschränkt. In [DT05] und [HJS09] wurde außerdem mit Einzelbildern statt mit Videosequenzen gearbeitet. [HJS09] zeigt, dass die isolierte Nutzung von HOG zur Objektbeschreibung unterschiedlicher Verkehrsobjekte Optimierungspotential besitzt und zur alleinigen Nutzung nicht uneingeschränkt geeignet ist.

2.5.3 Hauptkomponentenanalyse (PCA)

Theorie

Die Hauptkomponentenanalyse, auch bekannt unter den Begriffen Hauptachsentransformation, Karhunen-Loève-Transformation, Singulärwertzerlegung oder engl. Principal Component Analysis (PCA), ist ein Verfahren der multivariaten[3] Statistik. Dies bedeutet, dass Abhängigkeiten zwischen mehreren Variablen erkannt werden können. Mit der PCA können umfangreiche Datensätze strukturiert, vereinfacht und veranschaulicht werden. Idee des Verfahrens ist, eine große Menge an Variablen durch eine geringe Zahl von Linearkombinationen, den sogenannten Hauptkomponenten (Principal Components), anzunähern. Dies wird erreicht, indem korrelierte Variablen des ursprünglichen Datensatzes durch nichtkorrelierte neue Variablen ersetzt werden, sodass die Varianz des neuen Datensatzes möglichst groß wird. Der Abstand der in den Eigenraum projizierten Bilder ermöglicht eine Aussage über die Ähnlichkeit der Bilder und bildet somit die Basis für die Klassifizierung. [Sti06]

Der grobe Ablauf der PCA kann wie folgt skizziert werden: [ZCC06]

1. Normierung der Trainingsbilder.
2. Darstellung der mxn Pixel großen Bilder als Vektoren der Größe mnx1.
3. Bestimmung des mnx1 großen Vektors mit den Mittelwerten der Intensitäten der Pixel aller Bilder.
4. Berechnung der Abweichung der jeweiligen Bildvektoren vom Mittel.
5. Berechnung der Eigenvektoren und Eigenwerte der Kovarianzmatrix.
6. Auswahl der Hauptkomponenten, indem nur noch jene Eigenvektoren betrachtet werden, die über die größten Eigenwerte verfügen.

[3] Verwendung von mehr als einer Variablen/Dimension.

7. Durchführung der Schritte 1 und 4 für das zu klassifizierende Objekt und Zerlegung in seine Hauptkomponenten.
8. Klassifikation.

Die Punkte 2,3 und 4 können je nach Implementierung abweichen. Die hier verwendete mittlere Bildintensität stellt eine simple, aber beliebte Option dar.

Das Verfahren ist schnell und effizient, erfordert jedoch einen großen Aufwand zur Erzeugung von Invarianz.

Beispielanwendungen

Eine prominente Anwendung der PCA in der Bildverarbeitung ist die Gesichtserkennung. Hier werden mit Hilfe der PCA sogenannte Eigenfaces generiert, welche die invarianten Merkmale von Gesichtern widerspiegeln. Eigenfaces bestehen aus Eigenvektoren, die aus der Kovarianzmatrix der Wahrscheinlichkeitsverteilung der höherdimensionalen Vektorräume von möglichen Gesichtern abgeleitet sind. Der Eigenvektor mit dem größten Eigenwert repräsentiert die Objektklasse am besten.

In [ZCC06] wird die Hauptkomponentenanalyse als Bestandteil zweier Algorithmen zur Klassifikation von Fahrzeugtypen in Videosequenzen (2943 Frames) verwendet.

Die Kameraperspektive und somit die Bilddaten in [ZCC06] weisen große Ähnlichkeit zu den Bilddaten dieser Arbeit auf. So zeigen die Kamerabilder eine Kreuzung von schräg oben aus einer recht großen Distanz. Die Fahrzeuge bewegen sich in verschiedenste Richtungen. Um Vergleichbarkeit zu erzeugen, wurden die Fahrzeuge gegenüber Veränderungen von Skalierung und Fahrtrichtung sowie gegenüber perspektivischer Verzerrung invariant gemacht. Die Bestimmung der Fahrtrichtung bei Betrachtung eines einzelnen Frames gestaltet sich schwierig. Da sich die Fahrzeuge nicht zuverlässig Fahrspuren zuordnen lassen, wurden die Videoframes in Phasen unterteilt, die an die Lichtsignalphasen geknüpft sind. Somit sind die möglichen Bewegungsrichtungen der Fahrzeuge in der betrachteten Phase für bestimmte Kreuzungsbereiche bekannt. Die normierten Objekte werden nun im ersten Schritt mit Hilfe ihrer Abmaße in die Klassen LKW und Auto eingeteilt. Anschließend werden Formmerkmale und invariante Fahrzeugmerkmale auf Basis der PCA identifiziert und darüber

eine genauere Einteilung in weitere Klassen ermöglicht. [ZCC06] testet hierfür zwei unterschiedliche Verfahren: „Eigenvehicles“ und „PCA-SVM“.

Das Verfahren „Eigenvehicles“ funktioniert analog zu den, für Gesichtserkennung weit verbreiteten, „Eigenfaces“. Mehr Informationen finden sich in [Sze11] (S.589). Es sei an dieser Stelle kurz erwähnt, dass die „Eigenvehicles“ bzw. Eigenfaces den Eigenvektoren entsprechen. Sie geben die Richtungen an, in welche die Bilder vom gemittelten Bild abweichen.

Bei der PCA-SVM wird die PCA zur Merkmalsextraktion genutzt. Das heißt mit Hilfe der PCA wird für jedes Objekt eine Menge an Eigenvektoren erzeugt, die die Merkmale des Fahrzeugs und damit das Fahrzeug selber repräsentieren. Die signifikantesten Eigenvektoren (Principal Components) bilden die Basis für die Klassifikation mittels binärer SVM, die jeweils für die Klassen „passenger-cars“, „pickup/trucks“ und „vans/SUVs“ durchgeführt wird.

Experimente ergaben für PCA-SVM eine Trefferquote (Recall) von durchschnittlich 72% für die drei Fahrzeugklassen und eine Präzision (Precision) von rund 59%. Die Resultate für Eigenvehicles waren mit 68% und 56% geringfügig schlechter.

Zwischenfazit

In[ZCC06] handelt es sich um einen interessanten Anwendungsfall, da Videosequenzen einer Kreuzung mit vier möglichen Fahrtrichtungen als Basis für die Klassifikation dienen. PCA scheint hierbei ein vielversprechendes Verfahren zur Extraktion gemeinsamer Merkmale einer Objektklasse und somit auch zur Klassifikation von Fahrzeugtypen. Nachteilig ist jedoch der hohe Aufwand zur Angleichung der einzelnen Objekte in Bezug auf Ausrichtung, Auflösung, Skalierung und Lichtverhältnisse.

2.5.4 Template Matching

Theorie

Beim Template Matching wird im Bild nach einer gegebenen Vorlage gesucht. Es wird somit das Vorwissen über ein Objekt genutzt. Diese Vorlage kann man auch als Schablone oder engl. Template beschreiben. Das Template wird in Form einer Filtermaske in

allen Orientierungen über alle Positionen im Bild geschoben. Mit Hilfe einer Faltungsoperation wird dabei für jeden Schritt das Maß an Übereinstimmung von Template Filtermaske und darunter liegender Bildfunktion bestimmt. Das Verfahren ist sehr aufwendig und eignet sich daher für viele Anwendungsfälle nur bedingt. [Ste93]

Beispielanwendungen

In [TTR01] werden Fahrzeuge beim Einfahren in eine Parkgarage seitlich von einer Kamera erfasst. Die Objekte werden durch Farbveränderungen im Bild erkannt und vom Hintergrund separiert. Die dadurch entstandenen Bildsegmente werden dann mit hinterlegten Bildern verschiedener Fahrzeugtypen verglichen. Mit Hilfe eines Ähnlichkeitswertes wird die Ähnlichkeit zwischen Template und Bild ermittelt. Die Ergebnisse der Tests wurden als gut beschrieben. Das Paper weist allerdings darauf hin, dass es auch zu Fehlerkennungen kommt. Ohne weitere Bearbeitung der Bilder vor dem Matching und unter Berücksichtigung, dass die Fahrzeuge konstant aus dem gleichen seitlichen Blickwinkel betrachtet wurden, kann man davon ausgehen, dass das Verfahren für komplexere Anwendungsfälle weniger geeignet ist als andere betrachtete Verfahren.

Zwischenfazit

Template Matching ist per se ein aufwändiges Verfahren. Das Template muss unterschiedlich skaliert, transliert und rotiert werden und für eine Klassifikation sind für alle Klassen Templates nötig. [TÖ8]

Da in [TTR01] mit dem Ansatz des Template Matchings für den vereinfachten Anwendungsfall der Fahrzeugklassifikation keine vielversprechenden Ergebnisse nachgewiesen werden konnten, wird der Ansatz in dieser Arbeit nicht weiter verfolgt.

2.5.5 Fourier Deskriptor (FD)

Eine sehr beliebte Methode der Merkmalsdeskription in der Bildverarbeitung sind die Fourier Deskriptoren (FD). Mit Ihnen wird ein Objekt anhand seiner Kontur beschrieben. Diese wird in der Regel mit Hilfe von „Shape Signatures“ als eindimensionale

Funktion dargestellt, um eine mehrdimensionale FT zu umgehen. Beliebte Shape Signatures sind die „Complex Coordinates“, bei denen die Konturkoordinaten (x, y) als komplexe Zahlen $(x + jy)$ interpretiert werden oder die “Centroid-Distance-Function“ (Schwerpunktdistanz-Funktion). Die Fourier Transformation (FT) wird dann auf die angepasste Darstellung der Objektkontur angewandt.

Theorie

Als Fourier Deskriptoren bezeichnet man die (normierten) Fourier Koeffizienten. Diese sind das Resultat der Fourier Transformation. Die Idee der Fourier Transformation ist es, beliebige Signale als Überlagerung gewichteter Sinusoide verschiedener Frequenzen darzustellen. Es gibt mehrere Formen der Fourier Transformation (vgl. Abschnitt 3.2), von denen die sogenannte Diskrete Fourier Transformation (DFT) für die digitale Bildverarbeitung von Bedeutung ist. Die komplexwertigen Fourier Koeffizienten $a(u)$ der DFT des Signals $s(k)$ berechnen sich gemäß Formel 2.3 .

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) e^{-\frac{j2\pi uk}{N}} \tag{2.3}$$

mit $u = 0, ..., N-1$. Die Berechnung der Fourier Transformation wird in der Praxis meist mit der sogenannten Fast Fourier Transformation (FFT) realisiert. Die FFT ist ein sehr effizienter Algorithmus zur computergestützten Berechnung der DFT. Zur Rekonstruktion des Signals $s(k)$ ist es ausreichend, die Koeffizienten zu kennen. Für ein N-fach abgetastetes Signal interessieren nur die ersten $\frac{N}{2}$ Frequenzen. Für alle höheren Frequenzen ist das Abtasttheorem nicht erfüllt. Abgesehen von der 0. Frequenz, sind jeder Frequenz zwei Koeffizienten zugeordnet - einer für die positive Frequenz und der andere für die negative Frequenz. Diese beiden Koeffizienten bilden ein Koeffizientenpaar. Koeffizienten niedriger Frequenzen enthalten Informationen über die grobe Form des Signals und höhere Frequenzen beinhalten Details. Um eine Form hinreichend gut zu beschreiben, ist es somit oft unnötig die hohen Frequenzen zu beachten, weshalb auch für die Klassifikation eines Objektes anhand seiner Form nicht alle Koeffizienten berücksichtigt werden müssen. Deskriptoren sollten möglichst unempfindlich gegenüber Veränderungen wie Rotation, Translation oder Größenänderungen (Skalierung) sein. Da Fourier Koeffizienten an sich nicht ausreichend invariant sind,

müssen sie noch normiert werden, um als Deskriptor verwendet werden zu können. Dank der Einfachheit der Rechenregeln der FT (Dualitätsprinzip) ist es mit wenigen Rechenschritten möglich, aus den Fourier Koeffizienten normierte Fourier Deskriptoren zu erzeugen. Weitere Informationen zu Fourier Deskriptoren werden in Kapitel 3.2 bereitgestellt. [Smi97]

Beispielanwendungen

Von den vielen Anwendungen der Fourier Deskriptoren im Bereich der Objekterkennung werden drei Papers vorgestellt, die das Verfahren auf vergleichbare Problemstellungen angewandt haben.

In [TA03] werden Fourier Deskriptoren zur Klassifikation von Verkehrsteilnehmern in Bildsequenzen verwendet. Das verwendete Bildmaterial umfasst mehrere hundert Bilder aus Verkehrsszenen, in denen Fahrzeuge und Personen aus unterschiedlichen Perspektiven zu sehen sind (vgl. Abb 2.12). Mit Hilfe eines Algorithmus, der über die

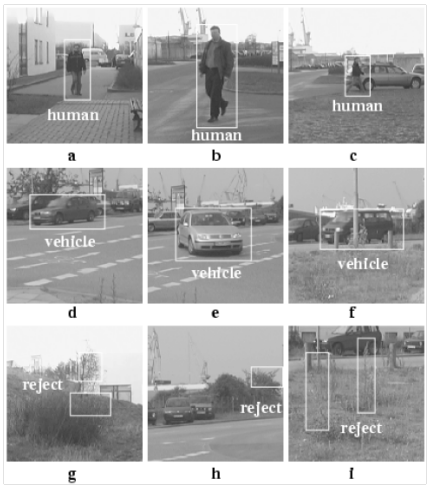

Abbildung 2.12: Klassifikation von Verkehrsobjekten. [TA03]

Differenz aufeinanderfolgender Frames Bewegungen erkennt, werden Objektkandidaten ermittelt. Es wird hierfür das sogenannte "Context-adaptive motion detection" Verfahren verwendet, welches laut [TA03] im Gegensatz zu herkömmlichen Verfahren dieser Art so gute Objektmasken erzeugt, dass eine zuverlässige Fahrzeugklassifikation damit möglich ist. Als Grundlage für die Fourier Transformation wurden beide

bereits erwähnten Boundary Tracings verwendet: die komplexen Koordinaten und die reellwertige "Centroid-Distance"-Funktion. Die Fourier Koeffizienten wurden ähnlich der Beschreibung in [Kapitel Methodik FD] normiert, so dass sich die Fourier Deskriptoren für den komplexen Fall zu

$$fd_{komplex} = [\frac{|a(n)|}{|a(1)|} * e^{-j\phi_1 n}], n = 2, 3, ..., N-1 \tag{2.4}$$

ergeben und für den reellen Fall zu

$$fd_{reell} = [\frac{|b(n)|}{|b(0)|}], n = 1, 2, ..., \frac{N}{2}. \tag{2.5}$$

Für die Merkmalsbeschreibung werden nur die ersten zehn Fourier Deskriptoren berücksichtigt. Es wurde empirisch ermittelt, dass die restlichen, höheren Frequenzen in diesem Anwendungsfall keine relevanten Informationen für die Klassenzuordnung enthalten. Die Klassifikation wurde mit einem „Feed-Forward Neural Net", also einem vorwärtsgerichtetem neuronalen Netz, mit vier Schichten realisiert. Die Schichten setzen sich aus einem Input-Layer mit einem Neuron je Merkmal, zwei sogenannte "Hidden Layers"mit je sieben Neuronen und einem Output-Layer mit einem Neuron je Klasse zusammen. Das Training wurde mit 400 Fußgängermerkmalsvektoren und 400 Fahrzeugmerkmalsvektoren durchgeführt. Es wurden nur die beiden Klassen *Person* und *Fahrzeug* unterschieden. Nach 10.000 Trainingszyklen stellten sich Erkennungsraten von $96 - 98\%$ für Fahrzeuge und $87 - 96\%$ für Personen ein. Die Klassifikation basierend auf komplexen Koordinaten schnitt dabei deutlich besser ab. Insbesondere Fußgänger wurden auf Basis der Centroid-Distances mit nur 87% vergleichsweise schlecht erkannt. Als Hauptgründe für Fehlklassifikationen wurden verdeckte Objekte und Schatten genannt. Auch die geringe Geschwindigkeit der Fußgänger beeinträchtigte das Ergebnis, da die Masken dadurch weniger exakt ermittelt werden konnten. Alle Ergebnisse wurden offline generiert, da der Algorithmus (noch) nicht echtzeitfähig ist.

[THM07] verwendet Fourier Deskriptoren zur Konturbeschreibung und -erkennung von Fußgängern. Motivation ist eine Verbesserung und Erweiterung von Fahrerassistenzsystemen in Bezug auf Fußgängererkennung. Es wird dennoch sowohl mit Fußgängerkonturen als auch mit Fahrzeugkonturen gearbeitet. Art und Güte des Bildmaterials sind dem Paper nicht zu entnehmen. Als Shape Signature werden wie im

zuvor vorgestellten Paper komplexe Koordinaten und Centroid Distances verwendet. Die Fourier Deskriptoren resultieren in [THM07] aus den Berechnungen 2.6 und 2.7.

$$fd_{komplex} = [\frac{|A(n)|}{|A(1)|}], n = 2, 3, ..., N-1 \tag{2.6}$$

$$fd_{reell} = [\frac{|B(n)|}{|B(1)|}], n = 2, 3, ..., N-1. \tag{2.7}$$

[THM07] gibt an, die ersten und letzten zehn Koeffizienten für die Objektdeskription verwendet zu haben. Als Klassifikator wird eine Support Vektor Maschine (SVM) mit RBF Kernel verwendet. Als Datengrundlage für die binäre Klassifikation stehen [THM07] die Merkmalsvektoren von 500 Fußgängerkonturen und 300 Fahrzeugkonturen zur Verfügung. Es wurde eine 5-fache Kreuzvalidierung angewendet. Die Klassifikation hat Fahrzeuge in 94% der Fälle korrekt erkannt und Fußgänger in 97% der Fälle. Die Kappa-Koeffizienten betrugen 0,93 für die komplexen Koordinaten und 0,92 für die Centroid Distances. Beide Verfahren brachte somit ähnlich gute Ergebnisse. Fehlklassifikationen wurden insbesondere auf Verdeckung zurückgeführt.

Eine weitere Anwendung der Fourier Deskriptoren im Kontext der Fahrzeugklassifikation findet sich in [YNGR09]. Es wird zwischen den Klassen Flugzeug, Helikopter, Rakete, Panzer, LKW und PKW unterschieden (vgl. Abb. 2.13). Die 180 verwendeten Objektkonturen stammen nicht aus einem Video, sondern von Einzelbildern, die aus dem Internet heruntergeladen wurden. Für die Merkmalsdeskription wurden 20 Fou-

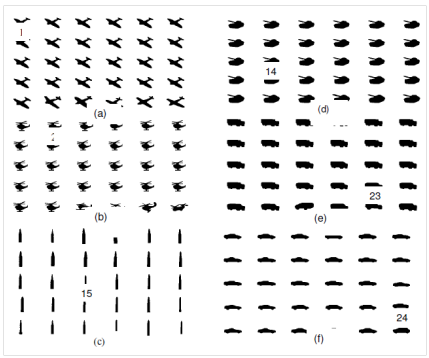

Abbildung 2.13: Datengrundlage für die Klassifikation. [YNGR09]

rier Deskriptoren verwendet, die invariant sind gegenüber Skalierung, Rotation und

Translation. Für die Klassifikation werden jeweils zwei Klassen mit Hilfe der euklidischen Distanz auf Ähnlichkeit überprüft. Als Performance wurde über 90% für alle Klassen angegeben. Die euklidischen Distanzen zwischen Objekten einer Klasse waren für alle Klassen klar geringer als die zu anderen Klassen. Im Vergleich zu den rechenintensiveren Wavelet Deskriptoren wurde den Fourier Deskriptoren die bessere Eignung zur Fahrzeugklassifikation zugesprochen.

Zwischenfazit

Fourier Deskriptoren erwiesen sich in allen betrachteten Anwendungen als äußerst gute Basis für die Fahrzeugklassifikation. Sie sind vergleichsweise wenig rechenintensiv und invariant gegenüber wichtigen geometrischen Transformationen. Als weitere Vorteile wurde Robustheit gegenüber Rauschen und gute Erkennungsraten in einer Vielzahl von Anwendungsfällen genannt. Die sehr guten Erkennungsraten der drei betrachteten binären Klassifikationen (mit nur zwei Klassen) zeigen das Potential des Verfahrens. Ob Fourier Deskriptoren auch für Klassifikationsaufgaben mit mehr als zwei Klassen (Mehrklassen-/Multiclass-Probleme) gute Ergebnisse generieren kann, bleibt offen.

2.5.6 Wavelet Deskriptoren

Theorie

Eine Wavelet-Transformation (WT) ist eine lineare Zeit-Frequenz-Transformation, mit der die Wavelet Deskriptoren einer Funktion berechnet werden können. Die ursprüngliche Funktion kann wiederum mit Hilfe der Deskriptoren dargestellt und rücktransformiert werden. Wavelets sind oszillierende, in der Regel zu den Seiten hin auslaufende Funktionen. Es gibt verschiedene Arten von Wavelets. Die einfachste und älteste Form eines Wavelets ist das sogenannte Haar Wavelet.

Die Idee der Wavelet Transformation ist, ähnlich wie bei der Fourier Transformation, die Darstellung einer Funktion mit Hilfe von Basisfunktionen, den sogennanten Wavelets. Wavelets sind somit das Äquivalent zu Sinus- und Kosinusfunktionen bei der Fourier Transformation. Der große Unterschied zwischen Wavelet- und Fourier Transformation ist, dass letztere nur Lokalität im Frequenzbereich besitzt, die WT hin-

gegen im Zeit- und Frequenzbereich. Das heißt, bei der Fourier Transformation hat das Signal im Zeitbereich keinerlei Frequenzinformation und im Frequenzbereich keinerlei Zeitinformation[4] . Man weiß also nicht, wann die Frequenzen auftreten, sondern nur, dass sie auftreten. Handelt es sich bei dem betrachteten Signal um ein nicht-stationäres Signal, so ist der Zusammenhang zwischen Zeitinformation und Frequenzinformation durchaus von Relevanz. Dies ist in Abb. 2.14 veranschaulicht, die zeigt, dass zwei sehr unterschiedliche Signale das gleiche Spektrum im Fourier-transformierten Raum ergeben können. Mit Hilfe der WT kann im Bildbereich der Unterschied zwischen den Signalen aufgezeigt werden. [Pol96]

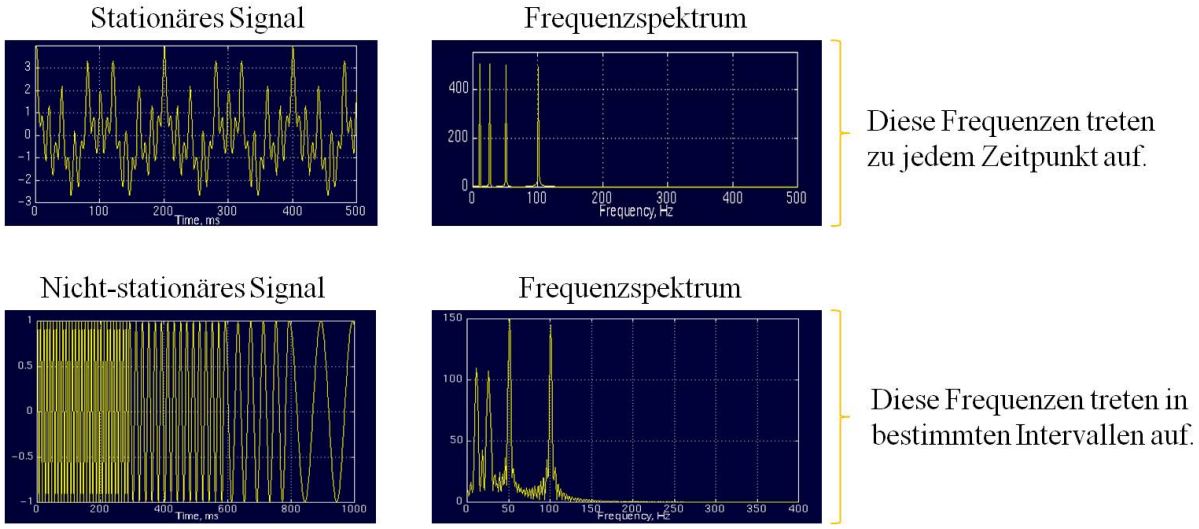

Abbildung 2.14: Zwei unterschiedliche Signale mit gleichem Frequenzspektrum. [Pol96]

Über die Wavelet Transformation ist also es gelungen eine Darstellung im Frequenz- und Zeitbereich zugleich zu realisieren. Zwar gilt die Unschärferelation der Nachrichtentechnik[5] selbstverständlich auch hier, mit der WT kann jedoch ein guter Kompro-

[4]Anmerkung: Analog dazu, dass der Zeitbereich in der Bildverarbeitung einem Ortsbereich entspricht, entspricht die Zeitinformation im Kontext der BV einer Information in Bezug auf den Ort.

[5]Die Unschärferelation der Nachrichtentechnik besagt, dass ein Ereignis nicht gleichzeitig im Zeit- und Frequenzbereich mit beliebiger Genauigkeit lokalisiert werden kann. Eine Aussage welche Frequenzen zu einem bestimmten Zeitpunkt vorliegen ist folglich nicht möglich. Es ist aber möglich diese Aussage für ein bestimmtes Zeitintervall zu treffen, wobei der Konflikt in Bezug auf die Auflösung ersichtlich wird: je kleiner das betrachtete Zeitintervall und je höher die daraus folgende zeitliche Auflösung, desto ungenauer wird die Auflösung im Frequenzbereich. [Pol96]

miss in Bezug auf die Auflösung im Zeit- und im Frequenzbereich erzielt werden. Auf diese Weise ist es möglich, Informationen darüber zu generieren, welche Frequenzen in welchem Zeitraum auftreten. [Pol96]

Die WT macht sich hierfür folgenden Zusammenhang zunutze:

- Im Zeitbereich sind hohe Frequenzen besser aufgelöst als Niedrige.
- Im Frequenzbereich sind niedrige Frequenzen besser aufgelöst als Hohe.

Die Konsequenz hieraus ist, dass man die Frequenzinformation im Bereich niedriger Frequenzen höher auflöst als im Bereich hoher Frequenzen und die Zeitinformation genau umgekehrt im Bereich hoher Frequenzen höher auflöst als im Bereich niedriger Frequenzen. Die Analyse des Signals mit unterschiedlicher Auflösung für unterschiedliche Frequenzen wird im englischen Sprachraum auch als „Multiresolution Analysis“ bezeichnet. [Pol96]

Die Funktionsweise der WT wird im Folgenden grob umrissen:

1. Das Eingangssignal wird durch Filterung mit speziellen Hoch- und Tiefpässen in ein niederfrequentes und ein hochfrequentes Teilsignal geteilt.
2. Das niederfrequente Teilsignal (manchmal auch das hochfrequente Teilsignal) wird dann nochmals auf die gleiche Art und Weise geteilt.
3. Diese Aufsplittung, auch Dekomposition genannt, wird nun wiederholt bis die gewünschte Auflösung erreicht ist.

Ergebnis der WT ist eine Menge an Signalen, die alle dasselbe Signal repräsentieren, jedoch in jeweils anderen Frequenzbändern. In der 3D Darstellung dieser Signale, kann man erkennen, in welchen Zeitintervallen welche Frequenzbänder mit welchem Betrag vorhanden sind (vgl. Abb. 2.15). [Pol96]

Als verbesserte Verfahren sollen hier beispielhaft die Curvelet und die Contourlet Transformationen genannt werden, da beide im Bereich der Fahrzeugklassifikation Anwendung finden.

Die Curvelet Transformation ist eine höherdimensionale Verallgemeinerung der Wavelet Transformation. Das bedeutet, dass sie im Gegensatz zu Wavelets auch Singulariäten höherer Ordnung erfassen können, wie sie in hochdimensionalen Signalen

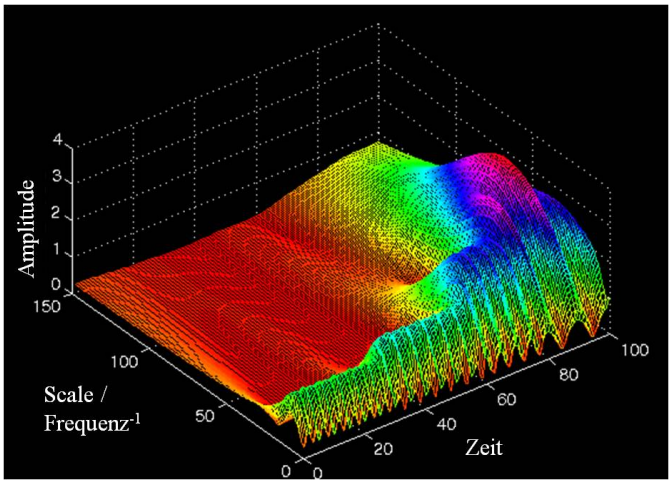

Abbildung 2.15: Wavelet Transform in 3D. [Pol96]

wie Bildern vorkommen. Somit können nicht nur beim Kreuzen von Kanten Bildmerkmale erfasst werden, sondern auch entlang der Kanten. Wie der Name „Curvelet“ bereits andeutet, können mit diesem Verfahren Kurven im Bild erkannt werden, weshalb es in der Praxis häufig für Gesichtserkennungsaufgaben verwendet wird. Ein Anwendungsbeispiel aus dem Bereich der Fahrzeugklassifikation findet sich in [KSPAT07]. [ZWZX11]

Da die Implementierung von Curvelets für diskrete Bilder problematisch ist, wurde mit den Contourlets eine Transformation speziell für diskrete Anwendungen geschaffen. Man kann sagen, dass die Contourlet Transformation die diskrete Form einer bestimmten Curvelet Transformation darstellt. In [AM10] wird das Verfahren der Contourlets für die Fahrzeugklassifikation getestet. [MP09]

Beispielanwendungen

[Mor08] vergleicht die Performance mehrer Verfahren zur Klassifikation von Fahrzeugtypen. Das Gabor Wavelet schneidet im Vergleich mit HoG, Texture Information und SIFT sehr gut ab. Als Klassifikator wurden k-nn und SVM getestet, wobei sich SVM als deutlich geeigneter herausstellte, da dieser die Daten skaliert und in einen höherdimensionalen Raum transformiert. Dort ist die Separierung der Daten einfacher. Es wurde mit einer SVM mit 5-facher Kreuzvalidierung gearbeitet. Die Klassifikation wurde für fünf verschiedene Blickwinkel auf die Fahrzeuge durchgeführt. Für das Ga-

bor Wavelet war die Genauigkeit mit fast 92% mit Blick auf das Heck des Fahrzeugs am besten, gefolgt von der Seitansicht mit knapp 91%. Etwas schlechter schnitten waren die Klassifikationsergebnisse für die frontale (ca. 89%) und schräg seitliche (ca. 85%) Sicht.

In [SBM02] werden Wavelet Features als besonders geeignet zur Fahrzeugerkennung beschrieben. Als Gründe werden die kompakte Darstellungsform, kurze Rechenzeiten und die Kodierung von, zur Fahrzeugerkennung wichtigen, Kanteninformationen genannt. Der Anwendungsfall in [SBM02] bezieht sich auf Bilder einer Kamera, die auf einem Fahrzeug montiert ist. Die Erkennung und Klassifikation dient in erster Linie der Verbesserung der Sensordaten für Fahrerassistenzsysteme und hat den Schwerpunkt auf der einfachen Unterscheidung der zwei Klassen *Fahrzeug* und *keinFahrzeug*. Hierfür werden quantisierte Haar-Wavelet Features verwendet, welche dann mit einer SVM klassifiziert werden. Mit dem Verfahren konnten Fahrzeuge mit einer durchschnittlichen Accuracy von 93,94% erkannt werden. In [WYY+07] dieser Ansatz in adaptierter Form aufgegriffen und auf ein binäres Problem angewandt. Aus [SBM02] und [WYY+07] geht nicht hervor, inwiefern sich der Wavelet-basierte Ansatz auch zur Klassifikation unterschiedlicher Fahrzeugtypen eignet.

[KSPAT07] vergleicht die Eignung von Fast Fourier Transformation, Wavelet Transformation (Haar Wavelet mit drei Auflösungsstufen) und Curvelet Transformation zur Erkennung und Klassifikation von fünf unterschiedlichen Fahrzeugmodellen auf Basis von Kameraaufnahmen aus dem Heck eines Fahrzeugs. Als Klassifikator wird ein k-nearest-neighbor Klassifikator eingesetzt. Versuche ergaben für die Curvelet Transformation mit einer Erkennungsrate von 100% die besten Ergebnisse. Das gute Ergebnis konnte sowohl bei Berücksichtigung aller Koeffizienten, als auch bei Betrachtung von nur 10% aller Koeffizienten erzielt werden. Unter Nutzung aller Koeffizienten ergab die Verwendung der FFT eine Erkennungsrate von 97%, die Wavelet Transformation erzielte nur 92%. Bei Verwendung von weniger als 75% aller Koeffizienten übertrifft die Wavelet Transformation allerdings die Ergebnisse der FFT. Für einen ähnlichen Anwendungsfall kommt [AM10] zu dem Ergebnis, dass eine Kombination aus Wavelet Transformation und Contourlet Transformation klassifiziert mittels SVM sich sehr

gut (97% Erkennungsrate) zur Erkennung von Fahrzeugmodellen eignet.

[YNGR09] vergleicht Wavelet Deskriptoren mit Fourier Deskriptoren. Der hier beschriebene Ansatz zur Klassifikation unterschiedlicher Fahrzeuge wie Hubschrauber, Flugzeug und Auto. Die Merkmalsvektoren werden mit Hilfe der Shape Signature „Centroid Distance Function“ generiert. Die Ähnlichkeit der Merkmalsvektoren des gesuchten Objektes und den Traingsdaten werden über die euklidische Distanz ermittelt und daraus dann die entsprechende Fahrzeugklasse abgeleitet. Die Ergebnisse der Versuche bescheinigen den Fourier Deskriptoren eine bessere Eignung zur Konturbeschreibung von Fahrzeugen. Die These wird auch durch [WRS+00] unterstützt, die Wavelet Deskriptoren in ihrem Artikel als weniger geeignet darstellen, was sie auf die fehlende Rotationsinvarianz sowie den hohen Aufwand für das Matching zurückführen.

Zwischenfazit

Eine allgemeine Aussage zur Eignung der Wavelet Transformation zur Fahrzeugklassifikation kann aus den betrachteten Papers nicht gefolgert werden. Zum einen decken sich die Aussagen in den einzelnen Arbeiten zur Eignung des Verfahrens für die Fahrzeugklassifikation nicht und zum anderen weichen die Anwendungsfälle vom in dieser Arbeit betrachteten Fall ab. So werden Wavelets in erster Linie zur bloßen Erkennung eines Objektes als Fahrzeug oder zur Unterscheidung von Fahrzeugmodellen - beides aus einer fahrzeugbasierten Kameraperspektive - eingesetzt. Da eine Erkennung von Fahrzeugmodellen möglicherweise eine ähnlich komplexe Aufgabe darstellt, wie das Erkennen des Fahrzeugtyps, ist eine Eignung des Verfahrens aber denkbar. Da Wavelets Zeit-Frequenz-Transformationen sind, kann grundsätzlich davon ausgegangen werden, dass dieses Verfahren im Falle zeitvarianter Signale mehr Informationen und damit eine größere Basis zur Differenzierung der Klassen bereitstellen kann, als vergleichsweise die Fourier Transformation. Es ist allerdings fraglich, ob diese Informationen für die Erkennung von Verkehrsobjekten tatsächlich nötig sind - insbesondere im Hinblick auf die Nachteile, die das Verfahren mit sich bringt, wie eine erschwerte Erzeugung von Invarianzeigenschaften [TK09] .

2.5.7 Fazit zum Stand der Technik

Fourier Deskriptoren scheinen für Anwendungen im Verkehrsbereich grundsätzlich gut geeigent zu sein. Sie sind vergleichsweise wenig rechenintensiv und haben, wie die Papers zeigen, großes Potential. Fourier Deskriptoren basieren auf dem Merkmal Kontur. Laut Literatur eignen sich zur Repräsentation des Merkmals die Shape Signatures „Centoid Distance Function“ und „Complex Coordinates“ am besten.

In dieser Arbeit sind daher Fourier Deskriptoren die Merkmalsdeskriptoren der Wahl. Das Verfahren wird auf Basis beider oben aufgeführter Shape Signatures ausgeführt. Zur Klassifikation wurden neben den in der Literatur häufig verwendeten Verfahren der SVM und des k-nn noch zwei Clusterverfahren sowie der simple Minimum-Distance-Classifier getestet.

Kapitel 3

Methodik und Vorgehensweise

Im Kapitel *Methodik* werden alle Verfahren, die in dieser Arbeit in Bezug auf Formrepräsentation, Merkmalsdeskriptor, Klassifikatoren sowie Bewertung der Klassifikation Verwendung finden, vorgestellt und erläutert.

3.1 Shape Signatures

Auf Basis grundlegender Überlegungen und einer Literaturrecherche, wurden in Kapitel 2 Fourier Deskriptoren als Merkmalsdeskriptor ausgewählt. Diese gehören zur Gruppe der Kontur-basierten Verfahren. Das heißt, die Deskriptoren werden durch Fourier Transformation der Objektkontur generiert und repräsentieren auch selber wieder eine Kontur. Es gibt verschiedene Möglichkeiten die Kontur darzustellen. Naheliegend und zunächst ohne jeglichen Aufwand wäre die direkte Verwendung der Koordinaten. Dies hätte den Nachteil, dass die Transformation im zweidimensionalen Raum durchgeführt werden müsste. Es werden daher „Shape Signatures“ als Konturrepräsentanten bevorzugt, da sie die Kontur in den 1-D Raum abbilden (vgl. Abschnitt 2.2.1). Übliche Shape Signatures sind Complex Coordinates, Centroid Distance Function, Tangent Angle (Turning Angles), Curvature Function, Area Function, Triangle-Area Representation and Chord Length Function. Diese werden nachfolgend kurz vorgestellt. Die Ausführungen basieren auf [MKJ08].

Complex Coordinates (Komplexe Koordinaten)

Im einfachsten Fall wird die Dimensionenreduktion erreicht, indem die (zweidimensionalen) Konturkoordinaten (x, y) als (eindimensionale) komplexe Zahlen $(x + jy)$ interpretiert werden (vgl. Formel 3.1). Die Koordianten können auf den Schwerpunkt oder einen anderen Punkt, z.B. den Koordinatenursprung, bezogen werden.

$$z(n) = [x(n) - g_x] + j[y(n) - g_y] \tag{3.1}$$

mit (g_x, g_y) als Bezugspunkt der Koordinaten. Complex Coordinates sind translationsinvariant für den Fall, dass der Bezugspunkt der Schwerpunkt ist.

Centroid Distance Function (Schwerpunktsdistanzfunktion)

Die Centroid Distance Function repräsentiert eine Kontur anhand ihres Abstandes zum Schwerpunkt (g_x, g_y) (vgl. Abb. 3.1). Berechnet wird die Funktion gemäß Formel 3.2.

$$r(n) = \sqrt{(x(n) - g_x)^2 + (y(n) - g_y)^2} \tag{3.2}$$

Diese Shape Signature ist Translationsinvariant.

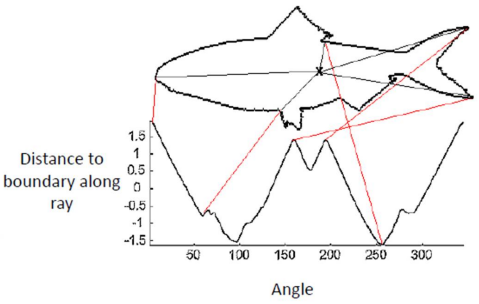

Abbildung 3.1: Centroid Distance. [Par11]

Tangent Angle (Turning Angles)

Die Tangent Angle Funktion $\theta(n)$ ist durch die Richtung der Tangenten an den einzelnen Konturpunkten definiert. Das Verfahren hat den Nachteil, dass es empfindlich gegenüber Rauschen ist und es eine Unstetigkeit durch die Definition des Winkels an der Stelle $theta = 0 = 2\pi$ gibt (vgl. Abb. 3.2). Abhilfe schafft die sogennante „Cumulative Angular Function“, bei der die Differenz der jeweiligen Tangenten zur Tangente eines Bezugspunktes betrachtet wird.

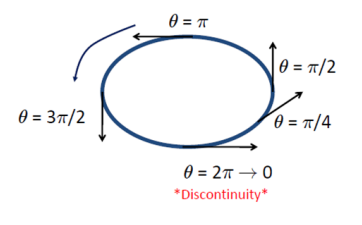

Abbildung 3.2: Tangent Angle. [Par11]

Contour Curvature (Krümmung der Kontur)
Für die menschliche Wahrnehmung von Ähnlichkeiten spielt die Krümmung eine wichtige Rolle und auch für die automatische Formerkennung hat sich das Verfahren als brauchbar erwiesen. Die Berechnung der Contour Curvature Funktion ist Formel 3.3zu entnehmen.

$$K(n) = \frac{\dot{x}(n)\ddot{y}(n) - \dot{y}(n)\ddot{x}(n)}{(\dot{x}(n)^2 + \dot{y}(n)^2)^{\frac{3}{2}}} \tag{3.3}$$

Area Function (Gebiets-/Flächenfunktion)
Die Area Function beschreibt die Kontur anhand der Fläche, die jeweils zwischen zwei Konturpunkten P_1 und P_2 sowie dem Schwerpunkt aufgespannt wird. Weichen die Konturpunkte von der Kreislinie ab, so verändert sich die Fläche. Abb. 3.3 visualisiert die Area Function am Beispiel „Apfel“.

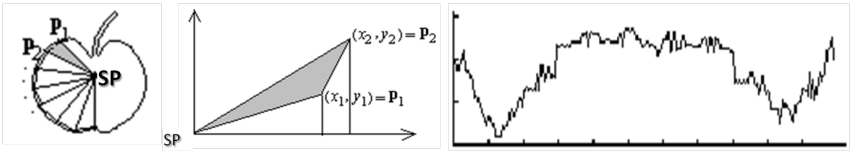

Abbildung 3.3: Area Function eines Apfels. [LZ13]

Triangle-Area-Representation (TAR - Dreiecksflächenrepräsentation)
Die TAR Signature ergibt sich durch die Fläche, die von drei aufeinanderfolgenden Konturpunkten aufgespannt wird. Liegen die Punkte auf einer Gerade, wäre die Fläche und damit der TAR Wert beispielsweise Null. Positive TAR Werte bedeuten, dass die Kontur an dieser Stelle konvex ist, negative Werte stehen für eine konkave Form (vgl. Abb. 3.4).

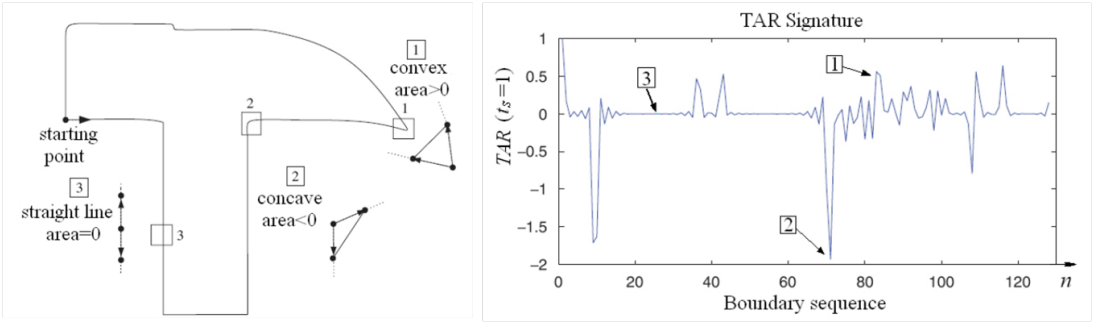

Abbildung 3.4: Triangle-Area-Representation. [MKJ08]

Chord Length Function

Gebildet wird sie durch die Distanz eines Konturpunktes zu einem bestimmten an-

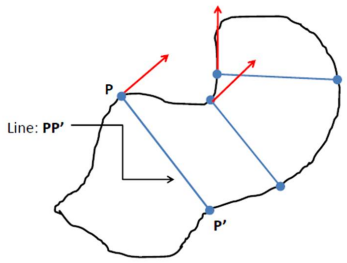

Abbildung 3.5: Chord Length Function. [Par11]

deren Konturpunkt. Dieser muss auf derjenigen Gerade durch den betrachteten Konturpunkt liegen, die senkrecht zur Tangente an diesem Punkt verläuft. Gibt es mehrere mögliche Punkte, so wird der Nächstgelegene gewählt. In Abb. 3.5 ist dies veranschaulicht. Die Chord Length Function ist unabhängig von einem Bezugspunkt. Dies ist ein Vorteil, weil dieser durch Konturfehler oder Rauschen oft mit einem Fehler (Bias) behaftet ist. Der große Nachteil des Verfahrens ist jedoch eine sehr große Sensibilität gegenüber Rauschen.

Fazit

Shape Signatures haben den großen Vorteil wenig rechenintensiv zu sein und häufig besitzen sie bereits wichtige Invarianzeigenschaften. Da diese Verfahren jedoch empfindlich gegenüber Rauschen sind und selbst kleine Änderungen der Kontur die Signature massiv verändern kann, eignen sie sich nicht direkt als Shape Descriptor. Als

Basis für die Beschreibung mittels Fourier Deskriptoren sind sie hingegen sehr gut. Wie die Literaturrecherche zeigte, haben sich in vielen Anwendungsfällen insbesondere Centroid Distance Functions als günstig erwiesen - da es diesbezüglich jedoch auch abweichende Meinungen gibt, soll in dieser Arbeit neben der Centroid Distance Function auch der Ansatz der Complex Coordinates implementiert werden.

3.2 Fourier Transformation und Fourier Deskriptoren

3.2.1 Grundlagen Transformation

Transformation Durch eine Transformation wird eine Menge von Daten mit Hilfe eines festgelegten Verfahrens in eine andere Menge von Daten umgewandelt. Transformationen können beispielsweise nützlich sein, wenn eine Berechnung mit den Originaldaten schwierig ist. Man transformiert die Daten in eine Form, in der sie besser handhabbar sind. Anschließend kann das Ergebnis wieder zurücktransformiert werden. Auf diese Weise erhält man das Ergebnis ohne die komplizierte Rechnung im Originalbereich durchführen zu müssen. Transformationen in den Frequenzbereich ermöglichen außerdem Aufschlüsse über im Signal vorkommende Frequenzen.

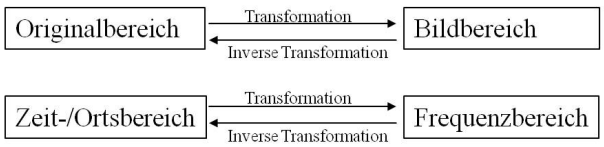

Abbildung 3.6: Begriffe zur Transformation.

Begriffsklärung Mit den in Abb. 3.6 erklärten Begriffen kann ausgedrückt werden, ob auf das Ursprungssignal oder auf seine Transformierte Bezug genommen wird. Der Begriff Zeitbereich ist in der Digitalen Signalverarbeitung üblich und wird auch im Bereich der digitalen Bildverarbeitung verwendet. In der Bildverarbeitung liegen jedoch keine zeitabhängigen Funktionen im eigentlichen Sinne vor, weshalb auch vom Ortsbereich gesprochen wird.

3.2.2 Fourier Transformation

Eine der bekanntesten und sehr häufig genutzten Transformationen ist die Fourier Transformation. Die folgenden Ausführungen basieren auf [Smi97].

Die Idee der Fourier Transformation ist es, ein beliebiges Signal als Summe von Kosinus- und Sinusfunktionen verschiedener Frequenzen und Amplituden darzustellen. Die Transformation erzeugt Amplitudenwerte, die jeweils den Basisfunktionen, Sinus und Kosinus, und einer bestimmten Frequenz zugeordnet werden. Addiert man die gewichteten Basisfunktionen, so erhält man das rekonstruierte Ursprungssignal (Zeitbereichssignal). Abb. 3.7 zeigt wie ein Rechteck-Signal (Mitte) durch Überlagerung von Sinuskurven (links) dreier Frequenzen (ganz rechts) angenähert wird.

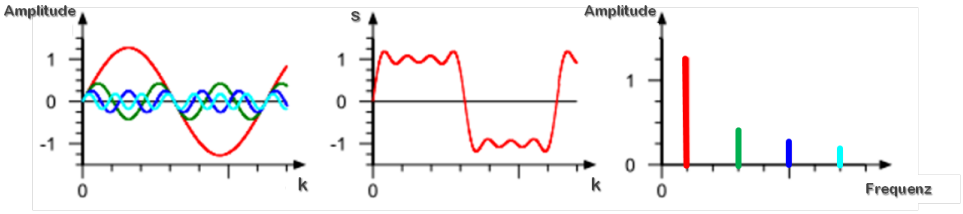

Abbildung 3.7: Fouriersynthese. (Basis:[Sch10])

Die diskrete Fourier Transformation (DFT) von $s(k)$ ergibt die komplexwertigen Fourier Koeffizienten $a(u)$, die entsprechend Formel 3.4 ermittelt werden:

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) e^{-\frac{j2\pi uk}{N}} \quad mit\ u = 0, ..., N-1^{1}. \tag{3.4}$$

Fourier Koeffizienten stellen die Basis der Fourier Deskriptoren dar[2].

Durch inverse Fourier Transformation der Fourier Koeffizienten, ist das Zeitbereichssignal $s(k)$ rekonstruierbar:

$$s(k) = \sum_{u=1}^{N-1} a(u) e^{\frac{j2\pi uk}{N}} \quad mit\ k = 0, ..., N-1. \tag{3.5}$$

Für die vollständige Rekonstruktion müssen alle N Fourier Koeffizienten verwendet werden. Insbesondere für die Beschreibung und Klassifikation einer Kontur ist es jedoch nicht nötig, zum Teil sogar hinderlich, jedes Detail eines Objektes zu beachten.

[1] Wie noch gezeigt wird, geht $a(u)$ für die reelle DFT nur bis $u = \frac{N}{2}$.

[2] In der Literatur werden die Begriffe teilweise sogar synonym verwendet.

Aus diesem Grund werden oft nur jene Fourier Koeffizienten verwendet, die niedrigen Frequenzen zugeordnet sind und somit grobe Formen beschreiben. Die Koeffizienten, die den höheren Frequenzen zugeordnet sind, beschreiben Details und werden ignoriert. Unabhängig von der Anzahl verwendeter Koeffizienten für die Rücktransformation, entsteht immer ein rekonstruiertes Signal aus N Punkten, das sich mit steigender Anzahl berücksichtigter Koeffizienten an die ursprüngliche Kontur annähert. Dieser Effekt wird in Abb. 3.8 am Beispiel der in Matlab durchgeführten Rekonstruktion einer Fahrzeugkontur visualisiert. Für die Rekonstruktion wurde schrittweise jeweils ein Koeffizientenpaar mehr einbezogen.

Abbildung 3.8: Rekonstruktion Fahrzeugkontur.

Die Anzahl an Abtastpunkten N wird in der Regel als Potenz von Zwei gewählt, da sich diese Werte für die digitale Datenspeicherung gut eignen und sich der effiziente FFT Algorithmus, der in Abschnitt 3.2.3 vorgestellt wird, auf solche Werte besonders gut anwenden lässt.

Genau genommen teilt sich die Familie der Fourier Transformationen in vier Kategorien, die Abb. 3.9 zu entnehmen sind. Welche der Transformationen benötigt wird, hängt davon ab, ob das betrachtete Signal periodisch oder nicht periodisch und ob es kontinuierlich oder diskret ist.

Da die Bildverarbeitung mit Hilfe von Digitalrechnern erfolgt, eignet sich von den vorgestellten Transformationen lediglich die Diskrete Fourier Transformation. Das liegt zum einen daran, dass Digitalrechner nur mit diskreten Informationen arbeiten können. Zum anderen können aperiodische Signale nur mit Hilfe von unendlich vielen Sinuskurven dargestellt werden, was die Berechnung mittels Computeralgorithmen unmöglich macht.

Für jede der Transformationen gibt es eine reelle und eine komplexe Version, je nachdem ob die Eingangsdaten reell oder komplexwertig sind. In dieser Arbeit werden sowohl die reelle, als auch die komplexe Diskrete Fourier Transformation verwendet.

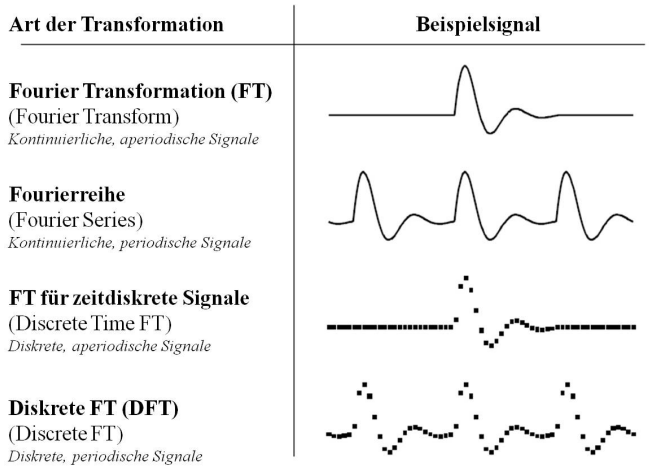

Abbildung 3.9: Arten von Fourier Transformationen. (Basis:[Smi97])

Reelle DFT Bei der reellen DFT wird ein im Originalbereich mit N Punkten abgetastetes Signal in $\frac{N}{2}+1$ Kosinus- und $\frac{N}{2}+1$ Sinuskurven transformiert. Auf Grund der mathematischen Zusammenhänge, die in Formel 3.6 beschrieben sind, können Sinus und Kosinus zu einer komplexen Zahl zusammengeführt werden. Das Ergebnis der reellen DFT sind somit $\frac{N}{2}+1$ komplexe Zahlen, aus denen Betrag und Phase der der Sinus- und Kosinusfunktion gemäß 3.6 bestimmt werden können.

$$z = Re + jIm = |z|cos(\phi) + j|z|sin(\phi) = |z|e^{j\phi} \tag{3.6}$$

wobei $|z| = \sqrt{(Re^2 + Im^2)}$ und $\phi = arctan(\frac{Im}{Re})$.

Die Koeffizienten $a(0)$ bis $a(\frac{N}{2})$ sind jeweils den Basisfunktionen der entsprechenden Frequenzen 0 bis $\frac{N}{2}$ zugeordnet. Beispielsweise ist der Koeffizient $a(2)$ den Basisfunktionen der Frequenz 2 zuzuordnen. Das heißt, die Funktionen durchlaufen über die N Abtastpunkte zwei komplette Zyklen. Ein besonderer Fall ist die 0. Frequenz. Der Betrag des Koeffizienten $a(0)$ entspricht dem Mittelwert oder Offset des Signals im Originalbereich. In Bezug auf die Centroid-Distance-Funktion kann dieser Wert daher als Radius der gemittelten Kontur verstanden werden. Da eine Sinusschwingung im Mittel immer Null ergibt, ist der Imaginärteil für die Frequenz Null immer gleich Null. Daher ist der Wert von $Im_{a(0)}$ irrelevant, sodass $a(0) = Re_{a(0)}$ ist. Die zweite Besonderheit ist die höchste Frequenz. Auch hier ist der Imaginärteil irrelevant. Hintergrund ist, dass eine Sinusschwingung der Frequenz $\frac{N}{2}$, die bei Null angefangen an

N Punkten abgetastet wird, an jedem Abtastpunkt den Wert Null hat. Zur Veranschaulichung sind in Abb. 3.10 beispielhaft die Basisfunktionen der Frequenzen 0,2,10 und 16 für eine DFT mit $N = 32$ Abtastpunkten visuell dargestellt.

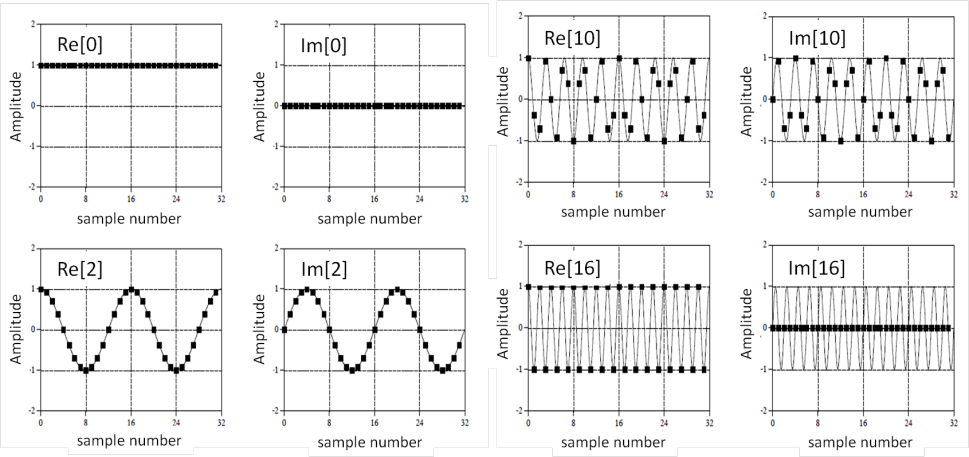

Abbildung 3.10: Basisfunktionen für Frequenzen 0, 2, 10, 16 (N=32). [Smi97]

Komplexe DFT Sind die Eingangsdaten der DFT komplexwertig, so verwendet man die komplexe DFT. Im Gegensatz zur reellen DFT transformiert die komplexe DFT ein N-Punkt Signal aus dem Originalbereich auch in ein N-Punkt Frequenzspektrum. Die FT ergibt also die N Fourier Koeffizienten $a(0)$ bis $a(N-1)$. Wie bei der reellen DFT entspricht jeder Punkt im Frequenzbereich einer komplexen Zahl mit Real- und Imaginärteil. Der Grund, dass das Signal im Frequenzbereich nun die Länge N hat anstelle von $\frac{N}{2}$ liegt darin, dass bei der komplexen DFT auch die negativen Frequenzen im Spektrum enthalten und von Bedeutung sind. Da die Frequenzen somit nicht von 0 bis $N-1$ sondern von -$\frac{N}{2}$ bis +$\frac{N}{2}$ gehen, bleibt die höchste Frequenz analog zur reellen DFT $\frac{N}{2}$. Da es sich um ein periodisches Signal handelt, spielt es keine Rolle, ob man das Signal in den Intervallen $[-\frac{N}{2}, +\frac{N}{2}]$ oder $[0, N-1]$ betrachtet. Die positiven Frequenzen, die auch für die reelle DFT verwendet werden, betrachtet man gewöhnlich im Intervall $[0, \frac{N}{2}]$. Die negativen Frequenzen liegen durch die Periodizität zwar sowohl im Intervall $[\frac{N}{2}+1, N-1]$ als auch im Intervall $[-\frac{N}{2}, -1]$ (vgl. Abb. 3.11), werden in der Regel im Ergebnisvektor aber im erstgenannten Intervall angegeben.

Analog zu Abb. 3.11a ist der Ergebnisvektor der komplexen DFT und somit auch

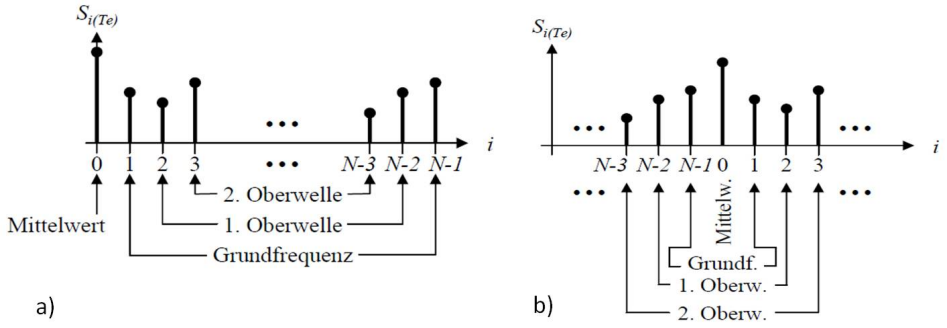

Abbildung 3.11: Frequenzanordnung a) ohne Shift b) mit Shift. [Lan13]

der reellen und komplexen FFT (vgl. Abschnitt 3.2.3), entsprechend Tabelle 3.1 zu interpretieren.

Re_0	$Re_1 + jIm_1$	...	$Re_{(\frac{N}{2}-1)} + jIm_{(\frac{N}{2}-1)}$	$Re_{\frac{N}{2}}$	$Re_{-(\frac{N}{2}-1)} + jIm_{-(\frac{N}{2}-1)}$	...	$Re_{-1} + jIm_{-1}$

Tabelle 3.1: Anordnung Fourier Koeffizienten im Ergebnisvektor.

Die Indizes sind hierbei als die zugeordneten Frequenzen zu verstehen. Für die FFT/DFT eines Beispielsignals mit $N = 8$ Abtastwerten ergibt sich die Ergebnisinterpretation gemäß 3.2.

Re_0	$Re_1 + jIm_1$	$Re_2 + jIm_2$	$Re_3 + jIm_3$	Re_4	$Re_{-3} + jIm_{-3}$	$Re_{-2} + jIm_{-2}$	$Re_{-1} + jIm_{-1}$

Tabelle 3.2: Beispiel Ergebnisvektor für N=0.

Im Falle der reellen FFT entsprechen die Koeffizienten ab $\frac{N}{2} + 1$ den konjugiert Komplexen der positiven Frequenzen. Sie beinhalten folglich keinerlei zusätzliche Information und werden daher nicht beachtet. Für die komplexe DFT bzw. FFT sind die negativen Frequenzen jedoch von Bedeutung. Die Koeffizienten von jeweils einer positiven und einer negativen Frequenz bilden ein Koeffizientenpaar.

Die geometrische Bedeutung der Koeffizienten kann wie folgt beschrieben werden. Der Koeffizient der Frequenz Null fd_0 ergibt den Mittelpunkt der Kontur und beschreibt somit die Translation des Objektes. Der Koeffizient fd_1 beschreibt einen Kreis. Auch der Koeffizient der zugehörigen negativen Frequenz fd_{-1} beschreibt einen Kreis.

Dieser wird jedoch in entgegengesetzter Richtung durchlaufen. Die Addition der beiden Koeffizienten ergibt eine Ellipse. Dies trifft auch auf alle weiteren Koeffizientenpaare zu, wobei die Ellipsen der jeweiligen Frequenz entsprechend häufig durchlaufen werden. [JÏ0]

Abbildung 3.12 zeigt wie durch Überlagerung von Ellipsen verschiedener Frequenzen die ursprüngliche Kontur rekonstruiert werden kann. Die Bilder (a) bis (d) zeigen die Rekonstruktion unter Verwendung der Nullfrequenz und zusätzlich einem, zwei, drei und zehn Fourier Deskriptorpaaren. Dies zeigt, dass die Verwendung von nur wenigen, niederfrequenten FDs, nur die grobe Form abbildet. Je mehr FDs für die Rekonstruktion verwendet werden, desto detaillierter wird die Kontur wiederhergestellt. Die Verwendung aller FDs bewirkt die komplette Rekonstruktion.

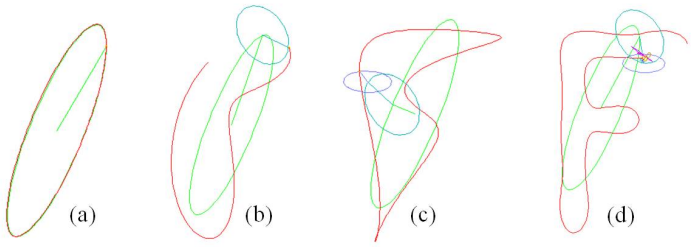

Abbildung 3.12: Rekonstruktion mit Fourier Deskriptoren. [Bur11]

3.2.3 Fast Fourier Transformation (FFT)

Die DFT kann auf unterschiedliche Art und Weise berechnet werden. Die bei Weitem schnellste Methode ist die sogenannte Schnelle Fourier Transformation oder auf Englisch Fast Fourier Transform, kurz FFT. Die Idee des Algorithmus ist es eine N-Punkt-DFT (also eine DFT mit N Abtastpunkten) in N 1-Punkt-DFTs zu zerlegen.

Die FFT basiert sowohl für reelle als auch für komplexe Eingangswerte auf der komplexen DFT. Um eine FFT für reelle Eingangswerte durchführen zu können, müssen aus den reellen Werten komplexe Werte generiert werden. Dies wird ganz unkompliziert realisiert, indem die reellen Werte als Realteil interpretiert und die Imaginärteile gleich Null gesetzt werden.

Die Funktionsweise der FFT kann wie folgt beschrieben werden:

1. Zerlegung des ursprünglichen Signals der Länge N in N Signale der Länge Eins nach dem "divide-and-conquer" Prinzip (Abb. 3.13):

   Die Abtastpunkte werden so lange nach Gruppen mit geradem und ungeradem Index aufgeteilt, bis jede Gruppe nur noch ein Element enthält. Dies entspricht einer Sortierung der Punkte in umgedrehter Bitfolge.

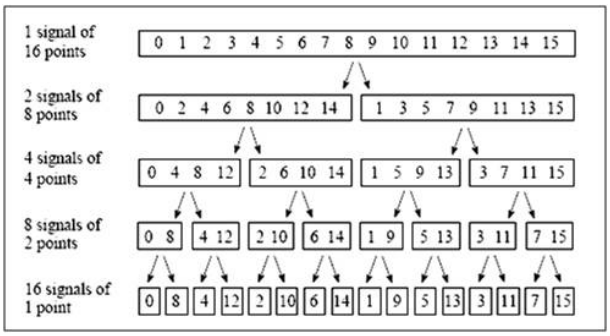

Abbildung 3.13: Divide-And-Conquer-Prinzip der FFT.[Smi97]

2. Bestimmung der Frequenzspektren der N Signale aus Schritt 1:

   Die N Signale im Originalbereich entsprechen jeweils ihrem Frequenzspektrum. Eine Berechnung im eigentlichen Sinn ist daher nicht nötig.

3. Zusammenfassen der N Frequenzspektren zu einem Frequenzspektrum:

   Die N Frequenzspektren werden Schritt für Schritt fusioniert bis es nur noch ein Frequenzspektrum mit N Punkten gibt. Dies geschieht in genau umgekehrter Reihenfolge als in Schritt 1. Realisiert wird dies mit Hilfe des sogenannten „butterfly", dem Kernelement der Berechnung der FFT, der zwei komplexe Werte in zwei andere komplexe Werte transformiert (vgl. Abb. 3.14)[3]. Wendet man den „butterfly" beispielsweise auf zwei 1-Punkt-Spektren an, so erhält man ein 2-Punkt-Spektrum.

Als Ergebnis ergeben sich im Frequenzbereich analog zur komplexen DFT N komplexe Zahlen. Wurde die FFT auf ursprünglich reelle Werte angewandt, so sind von diesen N komplexen Werten nur die ersten $\frac{N}{2} + 1$ Werte von Bedeutung. Die restlichen

[3]Die Operation xS bedeutet, dass das Signal mit dem Sinusoid einer bestimmten Frequenz multipliziert wird.

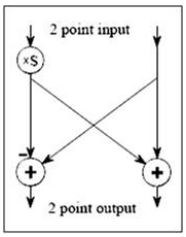

Abbildung 3.14: Butterfly-Prinzip der FFT. [Smi97]

Werte entsprechen den negativen Frequenzen, die für den Fall, dass alle Imaginärteile der Eingangsfunktion Null waren, die konjugiert Komplexen der ersten $\frac{N}{2}$Werte darstellen und somit keinerlei zusätzliche Information enthalten. Sie werden daher für den reellen Fall einfach ignoriert. Für komplexe Eingangswerte erhält man N komplexe Fourierkoeffizienten, also $\frac{N}{2} - 1$ Koeffizientenpaare sowie fd_0 und $fd_{\frac{N}{2}}$.
[Smi97]

3.2.4 Fourier Deskriptoren

Fourier Deskriptoren sind normierte Fourier Koeffizienten. In der Literatur ist die Definition der Fourier Deskriptoren nicht konsistent. Es können somit jegliche aus der FT resultierende Beschreibung des Signals gemeint sein. In dieser Arbeit werden für die Deskription normierte Beträge von Fourier Koeffizienten verwendet - diese bilden die Fourier Deskriptoren. Wie bereits erläutert, können mit der reellen DFT maximal $\frac{N}{2} + 1$ Fourier Deskriptoren erzeugt werden. Aus der komplexen FT ergeben sich N FDs, wobei jeweils zwei ein Koeffizientenpaar bilden und somit nicht einzeln verwendet werden.

Die Normierung der Koeffizienten wird für die reelle bzw. die komplexe DFT wie folgt durchgeführt:

Reelle DFT: Die Fourier Koeffizienten der reellen DFT auf Basis von Centroid Distances $s_r(k)$

$$a_r(u) = \frac{1}{N} \sum_{k=0}^{N-1} s_r(k) e^{-\frac{j2\pi uk}{N}} \quad mit \;\; u = 0, ..., \frac{N}{2}. \tag{3.7}$$

werden in ein Set invarianter Fourier Deskriptoren umgewandelt. Durch die Verwendung der Centroid Distances sind bereits die Eingangswerte translationsinvariant. Um

Rotationsinvarianz und Startpunktinvarianz ([TK09] , p.437f) zu erzeugen, werden ausschließlich die Beträge der komplexen Fourier Koeffizienten betrachtet, da die Rotationsinformation in der Phase enthalten ist. Skalierungsinvarianz wird erzeugt, indem die Beträge der Koeffizienten mit dem Betrag des Koeffizienten der Nullfrequenz $|a_r(0)|$ normiert werden.

Dies ergibt gemäß [TA03] die Fourier Deskriptoren

$$fd_{real} = [\frac{|a_r(u)|}{|a_r(0)|}] \quad mit \;\; u = 1, 2, ..., \frac{N}{2} \, . \tag{3.8}$$

Komplexe DFT: Auch die Fourier Koeffizienten der komplexen DFT auf Basis von Koordinaten, die als komplexe Zahlen $s_c(k)$ interpretiert werden,

$$a_c(u) = \frac{1}{N} \sum_{k=0}^{N-1} s_c(k) e^{-\frac{j2\pi uk}{N}} \quad mitu = 0, ..., N-1 \tag{3.9}$$

werden in ein Set invarianter Fourier Deskriptoren umgewandelt.

Wiederum werden Rotations- und Startpunktinvarianz erzeugt, indem ausschließlich die Beträge der komplexen Fourier Koeffizienten betrachtet werden. Um Skalierungsinvarianz zu erreichen, wird mit dem Betrag des Koeffizienten der ersten Frequenz $|a_c(1)|$ normiert. Da Information über die Translation lediglich in der Nullfrequenz enthalten ist, reicht es den Koeffizienten $|a_c(0)|$ zu ignorieren um die Deskriptoren translationsinvariant zu machen.

Die resultierenden invarianten Fourier Deskriptoren sind gemäß [THM07]

$$fd_{complex} = [\frac{|a_c(u)|}{|a_c(1)|}] \quad mit \;\; u = 2, 3, ..., N-1. \tag{3.10}$$

Um ein Signal vollständig beschreiben zu können, werden alle Fourier Deskriptoren benötigt. Für die Erkennung von Objektklassen ist es in der Regel ausreichend und sinnvoll (Gefahr des Overfitting) nur einen Teil der Deskriptoren zu verwenden. In der Regel werden nur die FDs der niedrigen Frequenzen (die grobe Forminformation enthalten) betrachtet. Dies muss jedoch für jeden Anwendungsfall individuell entschieden werden. Die ausgewählten Deskriptoren bilden gemeinsam den Merkmalsvektor eines Objektes. Sind die Merkmale bzw. Deskriptoren gut gewählt, so spannen sie einen Merkmalsraum auf, in welchem ähnliche Objekte nahe beieinander liegen

und sogenannte „Cluster“ bilden. Diese Eigenschaft ist die Basis für ein erfolgreiches Clustering bzw. eine erfolgreiche Klassifikation.
[TA03]

3.3 Clusteringverfahren

In dieser Arbeit sollen Verfahren der Clusteranalyse genutzt werden, um die Anzahl differenzierbarer Klassen zu ermitteln und um abschätzen zu können, ob die anvisierte Klassenanzahl mit den vorhandenen Daten realisierbar ist. Zudem bieten diese Verfahren den großen Vorteil, dass die sehr aufwendige Annotation der Videosequenzen, das heißt die händische Klassifikation, nicht unbedingt nötig ist. In dieser Arbeit wurden hierfür Dendrogramme und das k-Means Clustering verwendet.

3.3.1 Hierarchische Clusteranalyse (Dendrogramm)

Hierarchische Clusterverfahren sind distanzbasierte Verfahren zur Strukturanalyse von Daten. Das heißt, Objekte hoher Ähnlichkeit und damit geringer Distanz bilden einen Cluster. Die meisten Clusterverfahren fraktionieren die Objekte in eine konkrete Zahl an Clustern, wobei jedes Objekt eindeutig einem Cluster zugeordnet ist. Das hierarchische Verfahren hingegen erzeugt eine Folge von Partitionen der Objektmenge. Die Clusterzuordnung ist je nach Partitionsebene unterschiedlich. Folglich existiert keine global eindeutige Clusterzuordnung für die einzelnen Objekte, sodass die gewählte Partition letztendlich vom gewählten Partitionslevel bzw. der Abbruchbedingung abhängt. Abhängig davon, ob das Verfahren divisiv (top-down) oder agglomerierend (bottom-up) angewendet wird, erzeugt man aus einem groben Cluster, der alle Objekte beinhaltet, Schritt für Schritt eine detaillierte Aufteilung der Daten auf mehr und mehr Cluster bzw. genau andersherum. Beim agglomerativen Verfahren bildet zu Beginn also jedes Objekt einen Cluster. Anschließend werden Schritt für Schritt ähnliche Cluster zusammengefasst. Dies kann solange durchgeführt werden, bis nur noch ein einziger Cluster mit allen Objekten existiert. Mögliche Parameter der hierarchischen Clusteranalyse stellen das Distanzmaß (z. B. euklidische Distanz) und die Wahl der Bezugspunkte deren Distanz betrachtet wird (z. B. Schwerpunkt, nächster/entferntester

Punkt einer Klasse) dar. [Kle12]

Hierarchische Verfahren generieren zwar keine klare Clusterzuordnung, ermöglichen dafür jedoch aufschlussreiche Einblicke in die Clusterhierarchie samt ihrer Unterstrukturen. Dies lässt sich mit Hilfe eines Dendrogramms visualisieren.

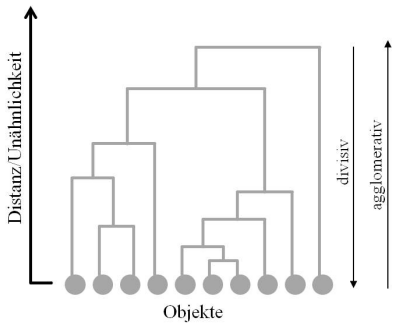

Abbildung 3.15: Dendrogramm

Ein Dendrogramm ist eine Baumstruktur, welche die hierarchische Zerlegung einer Menge von Daten (ein Cluster) in immer kleinere Cluster visuell darstellt. Die Objekte innerhalb der Cluster sollen hierbei möglichst ähnlich sein. Die horizontalen Ebenen des Dendrogramms entsprechen jeweils einer Clustereinteilung/Partitionsebene. Die Kantenlänge (vertikal) repräsentiert die Unähnlichkeit der Cluster. Die optimale Anzahl der Cluster kann bestimmt werden, indem die Ebene mit der längsten Kante und damit der größten Unähnlichkeit, als Clustereinteilung gewählt wird. Es kann jedoch auch einfach die Clustereinteilung für eine bestimmte Clusteranzahl betrachtet werden. Dendrogramme ermöglichen das intuitive Erkennen von Strukturen und Clustern in multidimensionalen Daten. Horizontal verbundene Objekte oder Subcluster bilden auf der übergeordneten Partitionsebene ein neues Cluster, die Höhe der senkrechten Verbindung zeigt wie groß die Distanz, also die Unähnlichkeit der Subcluster oder Objekte ist.

3.3.2 k-Means Clustering als Partitionierendes Clusterverfahren

Das k-Means Clustering soll in dieser Arbeit Verwendung finden, um die Einteilung der Merkmalsvektoren in die angedachten Klassen auf ihre Eignung hin zu überprü-

fen.

Das k-Means Clustering unterteilt die betrachteten Objekte in k Cluster. Die Zuordnung erfolgt jeweils zu dem Cluster, dessen Mittelwert/Klassenmitte die geringste Distanz und damit größte Ähnlichkeit zu den Merkmalen des betrachteten Objektes aufweist. Der Merkmalsraum wird dabei in Voronoi-Zellen[4] untergliedert. Da die Distanz zu den Klassenmitten verglichen wird, ist der Rechenaufwand unabhängig von der Anzahl der verwendeten Trainingsdaten. Die Startwerte für die Klassenmitten werden in der Regel zufällig gewählt. Da die Wahl der Startwerte das Resultat immens verändern kann, wird das Verfahren in der Regel mehrmals mit unterschiedlichen Startwerten durchlaufen und das beste Ergebnis genutzt. Zur Reduzierung dieser Schwäche, wurden zudem verschiedene adaptierte Verfahren entwickelt, wie beispielsweise „k-means++“. Weitere Schwachstellen des Verfahrens sind, neben dem großen Einfluss der gewählten Startwerte auf Laufzeit und Ergebnis, die Notwendigkeit zur Festlegung der Clusterzahl k, der Einfluss von Ausreißern sowie die Beschränkung auf konvexe Cluster (welche durch die Minimierung des Abstandes zum Schwerpunkt begründet ist). Dennoch zählt das Verfahren, dank seiner Effizienz und einfachen Implementierung, zu den populärsten partitionierenden Clustering Verfahren. [BAB+07]

Das Vorgehen kann wie folgt skizziert werden:

1. Festlegung der Klassenzahl k
2. Zufällige Verteilung der Klassenmitten: Abb. 3.16 (a)
3. Zuordnung der Objekte zu demjenigen Cluster, für den die (z.B. euklidische) Distanz zur Klassenmitte minimal ist: Abb. 3.16 (b)
4. Neuberechnung des Mittelwerts für jeden Cluster: Abb. 3.16 (c)
5. Basierend auf den neu berechneten Klassenmitten werden die Objekte wieder wie in Schritt 3 auf die Cluster verteilt: Abb. 3.16 (d)

Der Algorithmus endet entweder mit Erreichen einer festgelegten Iterationstiefe oder wenn Schritt 5 zu keiner Veränderung in der Klassenzuordnung mehr führt.

Die optimale Clusteranzahl ist meist nicht trivial bestimmbar. Als Orientierung dient ein Vergleich der Güte des Clustering für die unterschiedlichen Clusterzahlen.

[4] Voronoi-Zellen unterteilen einen Raum mit Elementen so, dass jedes Element von jedem nächsten Nachbarn durch die Mittelsenkrechten separiert wird.Es ergibt sich eine Art Wabenstruktur.

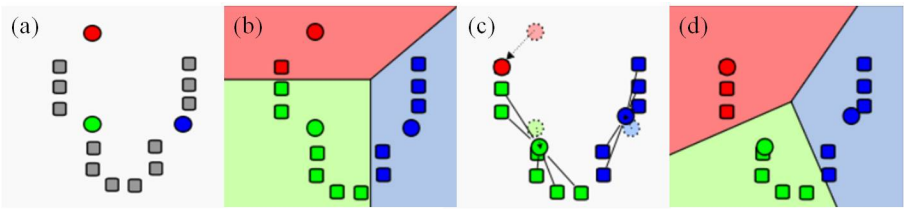

Abbildung 3.16: Ablauf 3-Means. (Basis: [Pac07])

Das Maß für die Güte darf nicht direkt von der Klassenanzahl abhängen. Geeignet ist daher beispielsweise der Silhouetten-Koeffizient. Dieser beinhaltet Informationen zur clusterinternen Kompaktheit (intraclass) sowie zur Trennschärfe (interclass) und berechnet sich gemäß [KR90] wie folgt:

$$s(o) = \frac{b(o) - a(o)}{max\{a(o), b(o)\}} \qquad mit \quad -1 \leq s(o) \leq +1 \tag{3.11}$$

wobei $a(o)$ der Abstand eines Objektes o zum Repräsentanten seines Clusters und $b(o)$ der Abstand zum Repräsentanten des „zweitnächsten“ Clusters ist.

Ein Silhouettenwert nahe Null bedeutet, dass das Objekt ungefähr zwischen zwei Clustern liegt. Ist der Wert kleiner Null, so ist das Objekt den Objekten des nächsten Clusters ähnlicher als dem eigenen Cluster - dies deutet auf ein nicht ausgeschöpftes Optimierungspotential hin.

Der Silhouettenkoeffizient s_c entspricht dem durchschnittlichen Silhouettenwert aller Objekte. Gemäß [KR90] ist der Koeffizient wie folgt zu interpretieren:

$0,7$	$< s_c \leq$	1	starke Clusterstruktur
$0,5$	$< s_c \leq$	$0,7$	brauchbare Clusterstruktur
$0,25$	$< s_c \leq$	$0,5$	schwache oder künstliche Clusterstruktur
	$s_c \leq$	$0,25$	keine Clusterstruktur vorhanden

Ein kleiner s_c Wert deutet bei kleinen Clusterzahlen auf falsche Clusterzuordnung hin, bei großen Clusteranzahlen ist es ein Hinweis auf künstliche Trennung von zusammengehörigen Clustern.

3.4 Klassifikatoren

Für die Zuordnung von Verkehrsobjekten zu Klassen werden in dieser Arbeit drei Klassifikatoren verwendet: Minimum Distance Classifier, k-Nearest-Neighbor Klassi-

fikator und SVM Klassifikator. Diese Verfahren sollen im Folgenden erläutert werden.

3.4.1 Minimum Distance Classifier

Der Minimum Distance Classifier, auch Minimum Distance to Means Classifier genannt, stellt ein einfaches Verfahren zur Klassifizierung von Objekten anhand des nächstgelegenen Klassenprototyps dar. Als Prototyp wird häufig der Klassenmittelpunkt gewählt. Dieser wird für jede Klasse aus den Trainingsdaten bestimmt, für die die Klassenzuordnung bekannt ist. Ein Testobjekt wird jener Klasse zugeordnet, zu deren Prototyp (Mittelpunkt) es die geringste Distanz aufweist (vgl. Abb. 3.17). Als Abstandsmaß wird meist die euklidische Distanz gewählt. [HU05]

Vorgehen:

1. Berechnung der Klassenmittelpunkte für die Trainingsdaten
2. Berechnung der Distanzen der Testobjekte zu den Klassenmitten
3. Zuordnung des Testobjektes zu der Klasse für die die Distanz aus 2) minimal ist

Das Verfahren ist sehr einfach zu implementieren und generiert in der Praxis oft gute Ergebnisse. Dies setzt jedoch voraus, dass die Interklassendistanz im Vergleich zur Intraklassendistanz groß ist. Zu berücksichtigen ist außerdem, dass es zu einem Bias/Verzerrungen in der Klassifikation kommen kann, wenn die natürliche Auftretenswahrscheinlichkeit markant von der im Trainingsset abweicht. [FPWW04]

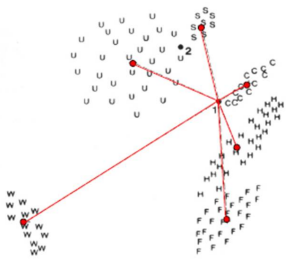

Abbildung 3.17: Minimum Distance Classifier.[HU05]

3.4.2 k-Nearest-Neighbor Klassifikator (k-NN)

Der Nearest Neighbor Klassifikator weist einem zu klassifizierenden Punkt die Klasse seines nächsten Nachbarn zu. Beim k-NN wird ein Merkmalsvektor entsprechend der Klasse zugeordnet, der die Mehrheit der k nächsten Nachbarn angehört (vgl. Abb. 3.18). Als nächste Nachbarn bezeichnet man jene Trainingsdaten, die sich innerhalb einer bestimmten (aber nicht fixen) Distanz zum zu klassifizierenden Objekt befinden. Als Distanzvektor können beispielsweise die euklidische, Manhattan oder Mahalanobis Distanz gewählt werden. Standardmäßig wird das Mehrheitsprinzip als Entscheidungsregel für die Zuordnung eines Objekts zu einer Klasse genutzt. Eine andere Option ist die Konsensusregel, bei der ein Objekt einer Klasse nur dann zugeordnet wird, wenn alle k nächsten Nachbarn dieser Klasse angehören. Besteht unter den nächsten Nachbarn kein Konsens, so wird das Objekt keiner Klasse zugeordnet.

Vorgehen:

1. Suche der k Merkmalsvektoren (mit bekannter Klassenzuordnung) deren Distanz zum Merkmalsvektor des betrachteten Objekts minimal ist.
2. Das Objekt wird der Klasse zugeordnet, der die Mehrheit der k Nachbarn angehört (Mehrheitsregel). Sind mehrere Klassen zu genau gleichem Anteil vertreten, erfolgt die Zuordnung zufällig.

Die Wahl von k hat einen großen Einfluss auf die Güte der Klassifikation. Die optimale Festlegung dieses Wertes ist nicht ganz trivial und nicht allgemein bestimmbar. Jedoch sollte k kein Vielfaches der Klassenanzahl sein. Im Regelfall wird die Klassifizierung mit unterschiedlichen Werten für k durchgeführt und das k, welches die geringste Fehlerrate ergibt, gewählt.

Das Prinzip ist einfach, birgt jedoch die Gefahr des Overfitting (selbst Rauschpunkte werden korrekt klassifiziert) und ist den Speicher- und Rechenaufwand betreffend nicht effizient. Als problematisch erweist sich insbesondere die Suche nach den nächsten Nachbarn. Ohne effiziente Suchschemata/-algorithmen ist der Rechenaufwand insbesondere für die Klassifizierung mit einer großen Zahl an k Nachbarn und einer großen Trainingsdatenmenge N groß: $\mathcal{O}(k * N)$. Für die Güte der Klassifikation ist ein großes N (viele Trainingsdaten) jedoch unabdingbar. Im Allgemeinen führt die Klassifikation mit dem 1-NN Klassifikator zu weniger zuverlässigen Ergebnissen als mit

einem $k > 1$. [PH03]

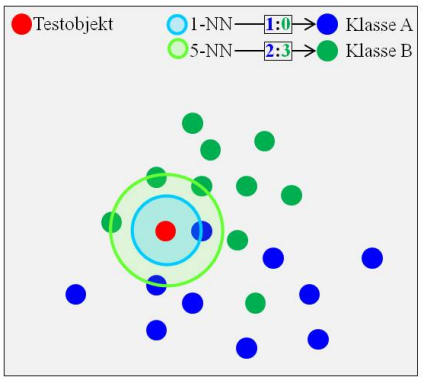

Abbildung 3.18: Beispiel 1-NN & 5-NN.

3.4.3 SVM

Die Klassifikation auf Basis der „Support Vector Machine“, kurz SVM, ist ein vergleichsweise neues Werkzeug. Der Begriff Maschine ist in Bezug auf das maschinelle Lernen zu verstehen. Wie eine Vielzahl an Papers (z. B. [THM07]) zeigen, wird die SVM seit einigen Jahren häufig und meist erfolgreich im Bereich der Objektklassifikation eingesetzt. Ziel der SVM ist es, binäre Klassifikationsprobleme zu lösen, indem die optimale Hyperebene zur Trennung der beiden Klassen gefunden wird. Wie Abb. 3.19 zeigt, gibt es selbst für die lineare Trennung von Daten eine große Zahl möglicher Trennebenen.

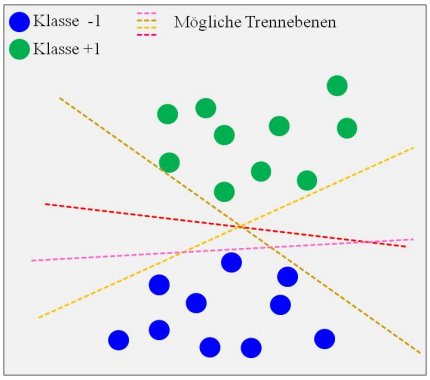

Abbildung 3.19: Beispiele möglicher Hyperebenen.

Die SVM versucht die Hyperebene mit Hilfe der Trainingsdaten so zu legen, dass der Abstand (Margin) zu den beiden Klassen maximal wird, also das Band zwischen den Klassen möglichst breit wird (vgl. Abb. 3.20). Je breiter die Margin, desto größer ist die Wahrscheinlichkeit der korrekten Zuordnung eines zu klassifizierenden Objektes.

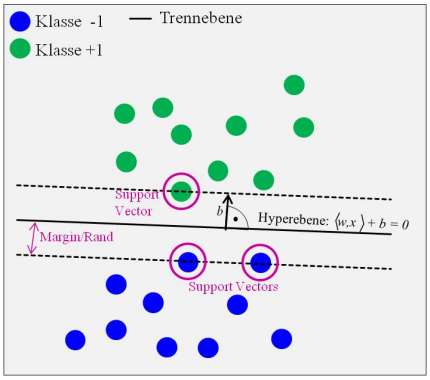

Abbildung 3.20: Trennebene mit maximierter Margin.

Als Stützvektoren bzw. Support Vectors werden jene Datenpunkte bezeichnet, die genau auf dem Rand der Margin liegen. Allein durch sie ist die Lösung (Trennebene) eindeutig bestimmt. Dies hat den Vorteil, dass auch mit großen Datensätzen effizient gearbeitet werden kann, da in der Klassifizierung bis auf die Stützvektoren keine weiteren Datenpunkte mehr berücksichtigt werden.

Die Hyperebene, auch OHS (Optimal Separating Hyperplane) genannt, definiert sich folglich über die Stützvektoren. Für eine Trainingsmenge aus N Vektoren $\vec{x}_i$ der Dimension d und ihrer Klassenzuordnung y_i (mit $\vec{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$ und $i \in [1, N]$) wird die trennende Hyperebene $\mathscr{H}$ zu

$$\mathscr{H} = \{\vec{x} | \langle \vec{w}, \vec{x} \rangle + b = 0\} \tag{3.12}$$

definiert mit Normalenvektor $\vec{w}$, Verschiebung (Bias) b und $\langle \vec{w}, \vec{x} \rangle$ als Skalarprodukt von $\vec{w}$ und $\vec{x}$. Stützvektoren sind jene $\vec{x}_i$, für die gilt $\langle \vec{w}, \vec{x} \rangle + b = 1$. Da alle Elemente auf der richtigen Seite der Hyperebene liegen sollen, gilt für alle Punkte $y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1$.

Die Bestimmung von $\vec{w}$ und b erfolgt durch Lösen des Optimierungsproblems der

Klasse Quadratischer Programmierung[5]

$$\min \frac{1}{2}\|\vec{w}\|^2 \tag{3.13}$$

mit den Nebenbedingungen $y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1$ für alle $1 \leq i \leq N$.

Das Optimierungsproblem ergibt sich aus dem Ziel, die Breite der Margin $\frac{2}{\|\vec{w}\|}$ zu maximieren, was eine Minimierung von $\vec{w}$ bedingt. Die Nebenbedingungen besagen, dass alle Datenpunkte eindeutig einer der Klassen $+1$ oder -1 zuordenbar sein sollen. Dies bedeutet, dass selbst Ausreißer in die Trennebene einbezogen werden. Eine solche Überanpassung ist für eine robuste Klassifizierung hinderlich und sollte vermieden werden. Der Effekt wird in Abb. 3.21 veranschaulicht.

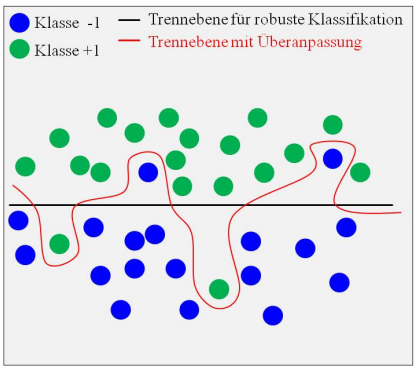

Abbildung 3.21: Overfitting.

Um einen robusten Klassifikator zu kreieren, verwendet man eine „Soft Margin“, die Ausreißer bei der Bestimmung der Trennebenen zu einem gewissen Grad ignoriert, aber bestraft. Dies ergibt das Optimierungsproblem

$$\min \frac{1}{2}\|\vec{w}\|^2 + C\sum_i \zeta_i \tag{3.14}$$

mit den Nebenbedingungen $y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1 - \zeta_i$ für alle $1 \leq i \leq N$ und $\zeta_i \geq 0$. ζ_i ist dabei der Abstand zum Rand der Margin (vgl. Abb. 3.22) bzw. die sogenannte Schlupfvariable (engl. Slack Variable). Sie ermöglicht zum einen, dass Elemente auch jenseits der Margin liegen können, die ihre Klasse begrenzt (realisiert durch Berücksichtigung

[5] Als Quadratische Programmierung werden Optimierungsprobleme bezeichnet, bei denen eine multivariate quadratische Funktion mit linearen Gleichungen oder Ungleichungen als Nebenbedingungen minimiert bzw. maximiert wird.[Mat13a]

ζ_i in Zwangsbedingung). Zum anderen sorgt sie für eine Bestrafung entsprechend der Stärke der Abweichung(realisiert durch Berücksichtigung ζ_i und C in Minimierungsfunktion). Der Faktor C dient der Gewichtung der Bestrafung.

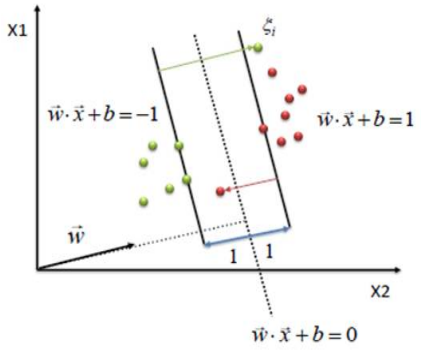

Abbildung 3.22: Soft Margin SVM. [Say12]

Grundsätzlich liegt hier ein konvexes, quadratisches Optimierungsproblem vor. Es ist jedoch auch möglich die Optimierungsaufgabe mit Hilfe von linearen Gleichungen zu lösen - dies bezeichnet man als „Least Squares Support Vector Machine“ (mehr dazu in [SV99]).

Die bisherigen grundlegenden Betrachtungen bezogen sich, von Ausreißern abgesehen, auf linear separierbare Fälle. In der Praxis ist eine lineare Separierbarkeit der Daten jedoch häufig nicht gegeben. Für die Support Vektor Maschine stellt dies kein Problem dar. Die SVM transformiert die Daten so lange in höher dimensionale Räume, bis eine lineare Trennbarkeit gegeben ist und die OHS bestimmt werden kann [THM07]. Bei der Rücktransformation in den ursprünglichen Raum wird die Hyperebene zu einer nicht-linearen Trennfläche.

Abbildung 3.23 zeigt ein Beispiel für eine binäre Klassifikationsaufgabe, deren Elemente im zweidimensionalen Raum (zwei Merkmale) nicht linear separierbar sind (3.23 a). Transformiert man den Merkmalsraum jedoch vom Zweidimensionalen ins Dreidimensionale, so ist eine lineare Trennung der Klassen problemlos möglich (3.23 b). Realisiert wird dies, indem aus den beiden Merkmalen (x_1, x_2) durch geschickte Transformation drei Merkmale (z_1, z_2, z_3) erzeugt werden. Konkret ergibt sich der neue Merkmalsvektor $\vec{z}$ im Beispiel aus $(z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. Die Rücktransformation der linearen Hyperebene im $\mathbb{R}^3$ ergibt hier eine kreisförmige Klassentrennung im $\mathbb{R}^2$ (3.23 c). [Mar03]

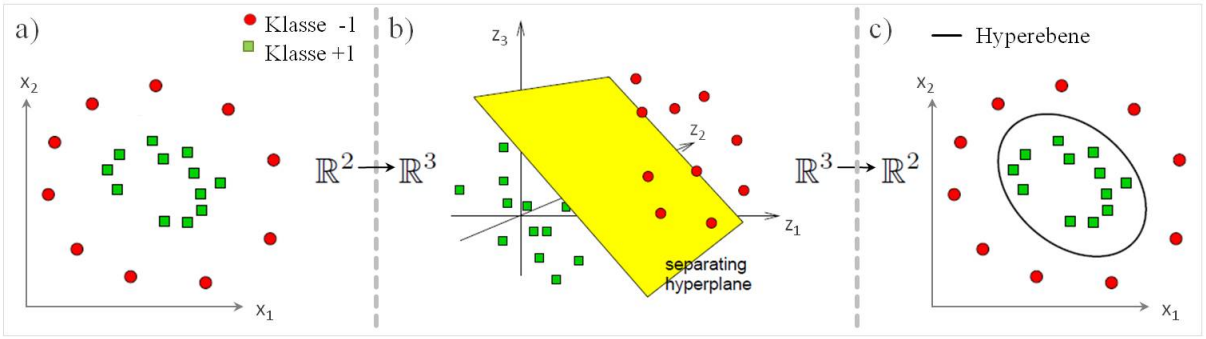

Abbildung 3.23: Linearer Separierbarkeit durch Dimensionserhöhung. (Basis: [Mar03])

Die Transformationen realisieren Support Vektor Maschinen mit Hilfe von Kernel Funktionen, welche die Daten in einen höher dimensionalen Raum abbilden. Es gibt eine Vielzahl solcher Funktionen, von denen keine nachweislich dominant ist. In der Objektklassifikation im Kontext der Bildverarbeitung wird häufig der universell einsetzbare „Gaussian Radial Basis Function Kernel“ (RBF Kernel) gewählt. [THM07]

Mehrklassen-SVM Der Nachteil der SVM ist, dass sie nur für binäre Klassifikationsprobleme $(A, \neg A)$ geeignet sind. Um, wie im vorliegenden Fall, Mehrklassen-Probleme zu lösen, müssen mehrere SVM verschachtelt und kombiniert werden. Dies kann auf verschiedene Art und Weise umgesetzt werden. Beliebte Ansätze sind das One-Versus-All (OVA) und das One-Versus-One (OVO) Prinzip. OVA bedeutet, dass für jede Klasse und für jedes Objekt geprüft wird, ob es zu der Klasse gehört oder zum Rest. Es werden somit für M Klassen M binäre SVM Klassifikatoren verwendet. Hier kann es vorkommen, dass die Zuordnung nicht eindeutig ist und ein Testobjekt mehreren oder keiner Klasse zugeordnet wird. Beim OVO Prinzip wird für jede Kombination von zwei Klassen geprüft, zu welcher der Klassen ein Testobjekt eher gehört. Für M Klassen werden somit $\frac{M(M-1)}{2}$ binäre Klassifikatoren trainiert. Das Testobjekt wird jener Klasse zugeordnet, die von den binären Klassifikatoren mehrheitlich vorhergesagt wurde. Dieses Verfahren ist durch die hohe Zahl an Klassifikatoren sehr aufwendig. [TK09]

3.5 Training und Bewertung der Klassifikatoren

3.5.1 Kreuzvalidierung

Die Kreuzvalidierung (engl. cross validation) nach [RTL09] ist eine statistisches Verfahren um Lernalgorithmen zu bewerten und zu vergleichen, indem die Daten in Partitionen unterteilt werden.

Hintergrund ist, dass für die Bewertung von Klassifikation und Klassifikator neben vorklassifizierten Trainingsdaten auch ein Testdatensatz benötigt wird, dessen korrekte Klasseneinteilung bekannt ist. Das Annotieren von Datensätzen ist sehr zeitaufwendig, weshalb vorklassifizierte Datensätze meist nur in stark begrenztem Umfang zur Verfügung stehen. Einerseits sollen die Daten daher so effizient wie möglich genutzt werden, andererseits dürfen unter keinen Umständen Trainingsdaten zugleich auch als Testdaten verwendet werden. Um diese Trennung von Trainings- und Testdaten zu gewährleisten und dabei dennoch möglichst viele der vorklassifizierten Daten für das Training nutzen zu können, wendet man eine Kreuzvalidierung (KV) an.

Für eine k-fache KV bedeutet dies, dass der annotierte Datensatz in k (ungefähr) gleichgroße Gruppen unterteilt wird. Anschließend wird die Klassifizierung k Mal durchgeführt, wobei in jedem Durchlauf eine andere Gruppe als Testdatensatz dient. Die verbleibenden Gruppen bilden zusammen jeweils die Trainingsdaten. Für jeden der k Durchläufe wird verglichen, ob die Testdaten das mit Hilfe der Trainingsdaten aufgestellte Modell bestätigen.In der Praxis hat sich gezeigt, dass sich die 10-fache KV besonders gut eignet.

Zu den verbesserten Varianten der Kreuzvalidierung gehören die stratifizierte Kreuzvalidierung sowie die Leave-One-Out Kreuzvalidierung.

Stratifizierte Kreuzvalidierung Die k-fache stratifizierte Kreuzvalidierung unterscheidet sich von der herkömmlichen k-fachen Kreuzvalidierung nur in der Aufteilung der Daten in die k Gruppen. Während Letztere die Daten zufällig auf die k Gruppen verteilt, stellt Erstgenannte sicher, dass in jeder Gruppe mindestens ein Repräsentant jeder Klasse vertreten ist.

Leave-One-Out-Kreuzvalidierung Die LOO-KV entspricht einer N-fachen Kreuzvalidierung, wobei N der Anzahl der Elemente im Datensatz entspricht. Dies bedeutet, dass N Durchläufe erfolgen, jeweils mit N-1 Trainingsdaten und einem Testelement. Der Rechenaufwand für dieses Verfahren ist entsprechend höher als bei der herkömmlichen Kreuzvalidierung.

3.5.2 Konfusionsmatrix

Die Bewertung eines Klassifikators erfolgt durch Vergleich der Klassifikation (Ergebnis der Klassifizierung) mit den tatsächlichen Klassenzugehörigkeiten. Dieser Vergleich ergibt die sogenannte Konfusionsmatrix, auch als Wahrheitsmatrix bekannt. Abb. 3.24 zeigt die Konfusionsmatrix für ein Zweiklassenproblem. Sie bezieht sich also auf eine Frage wie „Klasse A oder nicht Klasse A“.

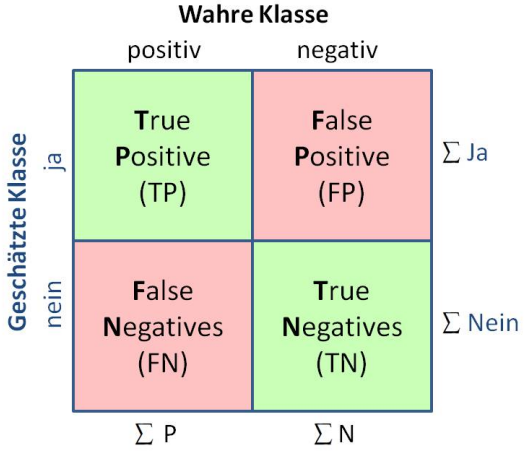

Abbildung 3.24: Konfusionsmatrix. (Basis: [Faw06])

Die Zeilensummen in Abb. 3.24 stehen für das Ergebnis der Klassifikation, die Spaltensummen für das wahre Ergebnis. Wäre die Klassifikationsaufgabe „Erkennung eines PKW“, so bezeichnet man alle als PKW klassifizierten Objekte, die auch tatsächlich PKW sind als „True Positive“. Analog fallen Objekte, die keine PKW sind und auch als „kein PWK“ erkannt wurden in die Rubrik „True Negative“. Entspricht die Klassifikation nicht der wahren Klassenzuordnung, so spricht man von „False Negative“ wenn

ein PKW nicht als solcher erkannt wurde und von „False Positive“ wenn ein Objekt fälschlicherweise für einen PKW gehalten wird.

Da in dieser Arbeit ein Mehrklassen-Problem vorliegt, liegt für M zu unterscheidende Klassen anstelle der $2x2$ Matrix eine MxM Matrix vor. Bei Mehrklassen-Problemen ist eine Einteilung gemäß der Konfusionsmatrix nur sinnvoll, wenn es auf die einzelnen Klasse bezogen als OVA Problem betrachtet wird $(A, \neg A)$.

Tabelle 3.3 zeigt die Interpretation der Konfusionsmatrix für Mehrklassen-Probleme bezogen auf alle Klassen.

		wahre Klasse			
		PKW	LKW	Zweirad	Fußgänger
geschätzte Klasse	PKW	korrekt	falsch	falsch	falsch
	LKW	falsch	korrekt	falsch	falsch
	Zweirad	falsch	falsch	korrekt	falsch
	Fußgänger	falsch	falsch	falsch	korrekt

Tabelle 3.3: Konfusionsmatrix für Mehrklassen-Probleme.

Arbeitet ein Klassifikator für alle Klassen fehlerfrei, so ergibt sich als Konfusionsmatrix eine Diagonalmatrix.

Aus den Werten der Konfusionsmatrix lassen sich einige Kennzahlen ableiten, die für die Bewertung der Klassifikation entscheidend sind. Einige wichtige Kennzahlen, die in dieser Arbeit für Vergleich und Bewertung der Klassifikatoren verwendet wurden, werden im nächsten Abschnitt vorgestellt.

3.5.3 Kennzahlen zur Bewertung einer Klassifikation

In dieser Arbeit wurden insbesondere die Kennzahlen Sensitivität und Ausfallrate als klassenindividuelles Bewertungsmaß sowie „Overall Accuracy“ und Kappa-Koeffizient für die Gesamtbewertung verwendet. Es werden die Definitionen der Kennzahlen nach [Sze11] und [Faw06] vorgestellt.

Sensitivität (TPR) [True Positive Rate, Empfindlichkeit, Trefferquote/Hit Rate]
Anteil der korrekt als positiv klassifizierten Objekte an der Gesamtheit der tatsächlich

positiven Objekte: $TPR = \frac{TP}{TP+FN} = \frac{TP}{P}$

Spezifität (TNR) [True Negative Rate oder Correct Rejection Rate]
Anteil der korrekt als negativ klassifizierten Objekte an der Gesamtheit der in Wirklichkeit negativen Objekte. $TNR = \frac{TN}{TN+FP} = \frac{TN}{N}$

Falsch-Positiv-Rate (FPR) [Ausfallrate/Fall Out, Falschalarmrate]
Anteil der fehlerhaft als positiv klassifizierten Objekte an der Gesamtheit der negativen Objekte: $FPR = \frac{FP}{TN+FP} = \frac{FP}{N} = 1-$ Spezifität

Präzision (PPV) [Precision, Positive Predictive Value, Relevanz, Wirksamkeit, Genauigkeit]
Anteil korrekt als positiv erkannter Objekte an der Gesamtheit der als positiv erkannten Objekte: $PPV = \frac{TP}{TP+FP}$

Accuracy (Acc) [Treffergenauigkeit]
Anteil der korrekt klassifizierten Objekte an der Gesamtzahl der Objekte:
$Acc = \frac{TP+TN}{P+N}$. Die Kennzahl eignet sich sowohl für binäre als auch für Mehrklassen-Probleme.

Kappa-Koeffizient
Differenz der beobachteten Übereinstimmung und der erwarteten zufälligen Übereinstimmung im Verhältnis zum Fall der kompletten Übereinstimmung (100% korrekt klassifiziert). Der Kappa-Koeffizient ermöglicht die Bewertung der Klassifikation unter Einbeziehen von zufällig korrekten Klassenzuordnungen. Für eine fehlerfreie Klassifikation ergibt sich $\kappa = 1$ und für eine systematisch falsche Klassifikation $\kappa = -1$. Gemäß [HU05] berechnet sich der Kappa-Koeffizient für eine Konfusionsmatrix der Größe MxM bestehend aus $k_{1,1}$ bis $k_{M,M}$ für eine Klassifikation mit n Objekten wie folgt:

$$\kappa = \frac{\frac{\sum_{i=1}^{r} k_{i,i}}{n} - \frac{\sum_{i=1}^{r}(k_{i,:} \cdot k_{:,i})}{n^2}}{1 - \frac{\sum_{i=1}^{r}(k_{i,:} \cdot k_{:,i})}{n^2}} \tag{3.15}$$

wobei „:“ für ganze Zeile bzw. ganze Spalte steht. Die Kennzahl eignet sich sowohl für binäre als auch für Mehrklassen-Probleme.

3.5.4 Receiver Operating Characteristic (ROC)

Die Zuordnung von Merkmalsvektoren zu Klassen erfolgt anhand ihrer Distanz zu Klassenrepräsentanten. Je nachdem wie die Distanz bis zu der ein Objekt noch zu einer Klasse gehört, gewählt wird, werden einem Repräsentanten tendenziell zu wenige (Schwellwert klein) oder zu viele (Schwellwert groß) Merkmalsvektoren und damit Objekte zugeordnet (vgl. Abb. 3.25).

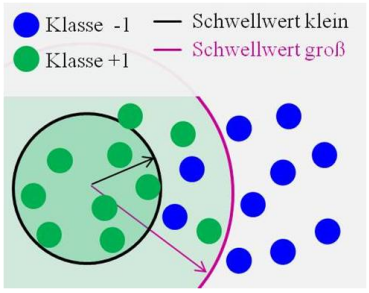

Abbildung 3.25: Schwellwerte.

Der erste Fall führt dazu, dass sowohl die TPR, als auch die FPR groß sind. Im zweiten Fall verhalten sich die Werte genau umgekehrt. Angestrebt wird jedoch eine hohe TPR und eine möglichst geringe FPR. Für welche Schwellwerte man hier das beste Ergebnis erreicht, ist in sogenannten ROC Kurven ersichtlich. Diese sind primär für probabilistische Klassifikatoren (die mit Schwellwerten arbeiten) und binäre Entscheidungsprobleme geeignet. Das Verfahren ist so in dieser Form daher für diskrete Klassifikatoren und Mehrklassen-Probleme, wie sie Gegenstand dieser Arbeit sind, nicht anwendbar. [Faw06]

Die ROC Kurven können jedoch auch für die Justierung diskreter Klassifikatoren hilfreich sein. Hier werden anstelle unterschiedlicher Schwellwerte verschiedene Parametereinstellungen miteinander verglichen. So können beispielsweise die optimale Zahl an Partitionen für die Kreuzvalidierung oder klassifikatorspezifische Parameter wie die Anzahl nächster Nachbarn bei der k-NN Klassifikation ermittelt werden.

Um eine ROC für ein Mehrklassen-Problem zu erzeugen, muss Letzteres auf mehrere Zweiklassenprobleme $(A, \neg A)$ herunter gebrochen werden. Es ergibt sich dann für jede Klasse eine eigene ROC Kurve.

Abb. 3.26 zeigt, wie ROC Kurven anhand ihrer Lage zu interpretieren ist. Dies soll

nachfolgend grob erläutert werden: Auf der Abszisse wird die FPR und auf der Ordinate die TPR abgetragen. Sowohl Definitions- als auch Wertebereich liegen zwischen Null und Eins. Die Diagonale zwischen Ursprung (0,0) und dem Punkt (1,1) entspricht dem Ergebnis einer rein zufälligen Klassifikation. Ein Klassifikator dessen Performance unterhalb dieser Diagonalen liegt, ist somit ungeeignet. Den Bereich akzeptabler Performance kann man wiederum in die Abschnitte konservative und liberale Performance teilen. Bei konservativen Klassifikatoren wird der Anteil an *FalsePositives* so gering gehalten wie möglich. Man möchte sicher sein können, dass ein erkanntes Objekte auch tatsächlich zu den gesuchten Objekten gehört. Dafür nimmt man eine geringere Anzahl an *TruePositives* in Kauf. Liberale Klassifikatoren verhalten sich entsprechend andersherum und nehmen für mehr *TruePositives* einen größeren β-Fehler (Fehler 2.Art) in Kauf. Der Klassifikator ist optimal eingestellt, wenn der Punkt (0,1) erreicht wird.
[Sze11]

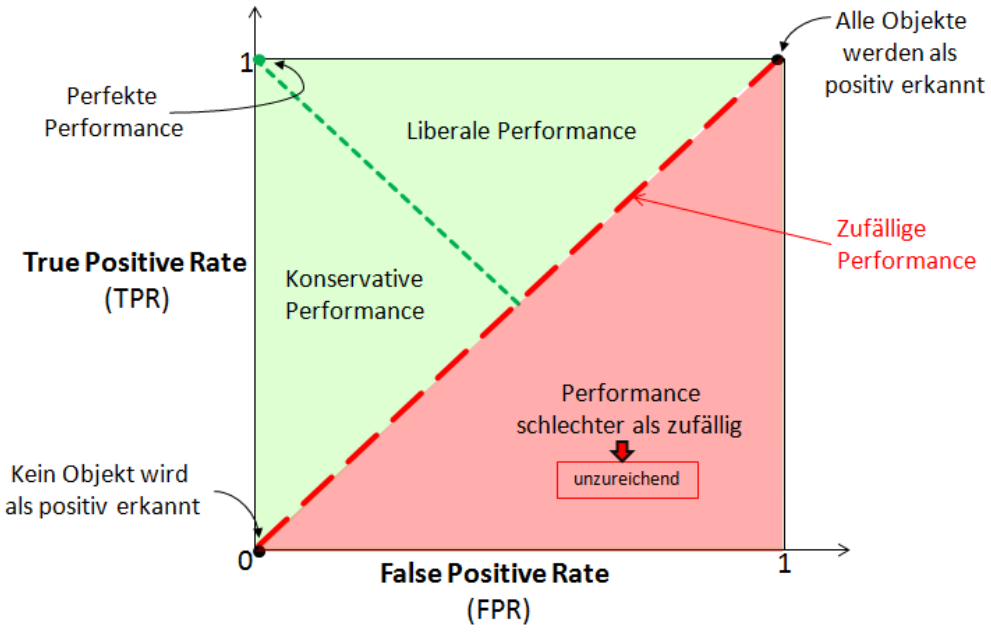

Abbildung 3.26: Interpretation ROC.

Kapitel 4

Implementierung und Experimente

4.1 Datengrundlage und Vorverarbeitung

Als Datengrundlage dienen fünf Verkehrsvideos, die von drei SmartCams des DLR (Deutsches Zentrum für Luft- und Raumfahrt) in Berlin aufgenommen wurden. Die Kameras befinden sich an der Kreuzung Rudower-Chaussee/Wegedornstraße in Berlin Adlershof. Die SmartCams sind in einer Höhe von 20 Metern angebracht. Sie haben eine Auflösung von 1392 x 1040 Pixel und eine Bildrate von 20 fps (frames per second). Detailliertere Informationen zu technischen Spezifika sind in [Leu10] verfügbar. Die Kameras sind mit einem Videoserver verbunden, über den die Signale in unterschiedlicher Größe und Qualität abgerufen werden können. Mit dem Videoserver ist ein Bildverarbeitungsserver verbunden, dessen Systemarchitektur in Abb. 4.1 abgebildet ist. Ein Objekterkennungsmodul detektiert mit Hilfe eines Hintergrundschätzers[1] bzw. optischem Fluss Bewegung im Bild und segmentiert die entsprechenden Regionen. Diese inhaltlich zusammenhängenden Regionen stellen im Weiteren die Objekte dar, die es zu klassifizieren gilt.

Aus den fünf Verkehrsvideos wurden insgesamt 85.517 Einzelbilder gewonnen. Aufeinanderfolgende Frames die keine oder kaum Änderungen aufweisen wurden nicht verwendet. Da der Hintergrundschätzer nicht zuverlässig nur Fahrzeuge und Fußgänger als Objekte erkennt, musste ein beträchtlicher Teil der Bilder aussortiert werden, da in ihnen keine (erkennbaren) Verkehrsobjekte enthalten waren (vgl. Abb.

[1] Es wurde der „BackgroundSubtractorMOG2“ nach Zivkovic von openCV verwendet. [Bra12]

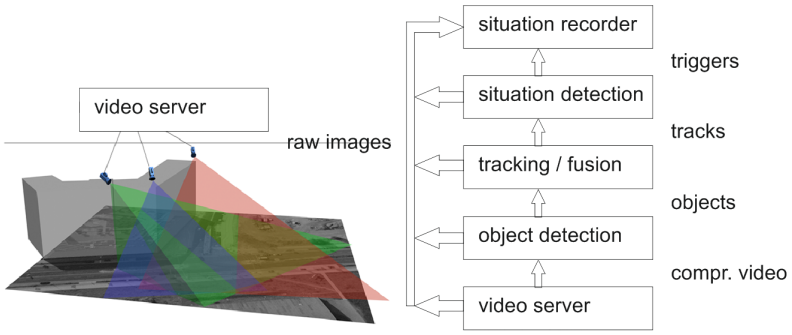

Abbildung 4.1: Vorverarbeitung. [KJ13]

4.2). Weitere Frames konnten nicht verwendet werden, weil mangelhafte Belichtungsverhältnisse oder unzureichende Bildauflösung eine Klassenzuordnung auch für das menschliche Auge nicht zuließen (vgl. Abb. 4.2). Die übrigen Bilddaten können als Test- und Trainingsdaten genutzt werden.

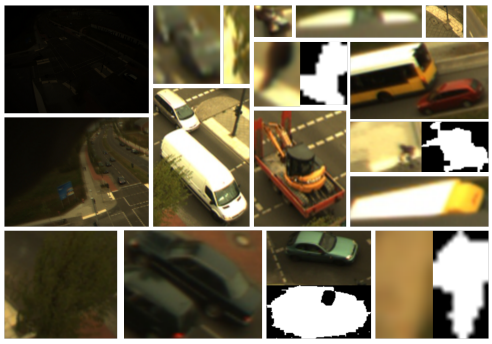

Abbildung 4.2: Negative Bildbeispiele.

Für die Verwendung überwacht lernender Klassifikatoren und zur Beurteilung der Klassifikationsgüte sind annotierte, also bereits klassifizierte Daten nötig - die sogenannte Ground Truth. Da die automatische Einteilung von Objekten aus Videosequenzen in Klassen unterschiedlicher Verkehrsobjekte das Ziel dieser Arbeit ist, muss die Vorklassifizierung von Hand durchgeführt werden. In einem ersten Schritt wurden die Bilder den Klassen PKW, Liefer-PKW, Kleinbus/Lieferwagen, Bus, LKW, andere Nutzfahrzeuge, motorisierte Zweiräder, Fahrräder, Fußgänger und Fußgängergruppen zugeordnet (vgl. Abb. 4.3) - diese sehr detaillierte Klassifikation ermöglicht maximale Flexibilität in Bezug auf die durchzuführenden Versuche. Realisiert wird die Klassenzuordnung durch Aufteilung der Bilder in Unterordner, die jeweils die Bilder einer

Klasse beinhalten. Der Basisdatensatz, in dem die wichtigsten Fahrzeugarten sowie Fußgänger enthalten sind, umfasst 1.757 Objekte. Nicht verwendet werden Fußgängergruppen und ungewöhnlich geformte Nutzfahrzeuge.

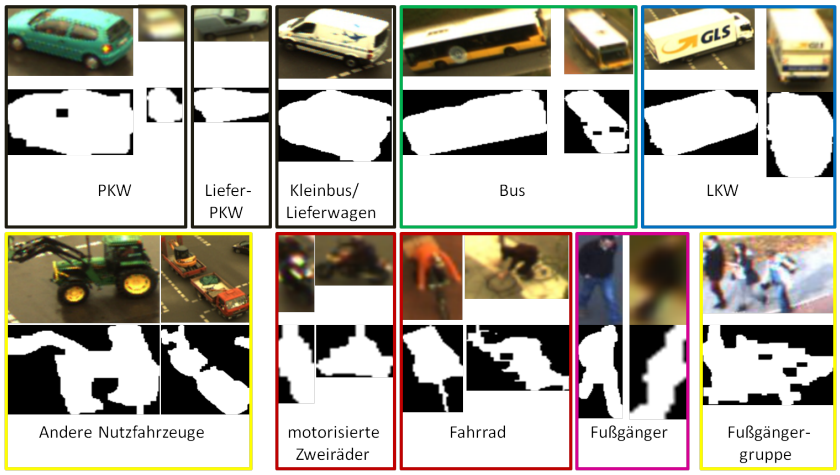

Abbildung 4.3: Beispielbilder der Objektklassen.

Desweiteren wurden mehrere adaptierte Datensätze erstellt. Sie unterscheiden sich beispielsweise in der Anzahl der unterschiedenen Klassen, in der Güte der Konturbeschreibung oder der Perspektive aus der die Fahrzeuge zu sehen sind.

Folgende Datensätze wurden für die Experimente verwendet:

- *DatensatzI*: Datensatz aus 1712 Objekten eingeteilt in fünf Klassen:
  805 Klasse 1 „PKW“, 44 Klasse 2 „LKW“, 68 Klasse 3 „Bus“, 266 Klasse 4 „Zweirad“, 529 Klasse 5 „Fußgänger“.
  Größe, Lichtverhältnisse und Perspektive der Objekte bzw. Bilder unterscheiden sich zum Teil erheblich.
- *DatensatzII*: Datensatz aus 713 Objekten aus schräg seitlichem Blickwinkel, eingeteilt in fünf Klassen:
  291 Klasse 1 „PKW“, 21 Klasse 2 „LKW“, 17 Klasse 3 „Bus“, 76 Klasse 4 „Zweirad“, 308 Klasse 5 „Fußgänger“.
  Größe und Lichtverhältnisse der Objekte bzw. Bilder unterscheiden sich zum Teil erheblich. Perspektive relativ konstant.
- *DatensatzIII*: Datensatz aus 1712 Objekten eingeteilt in vier Klassen:
  849 Klasse 1 „PKW/LKW“, 68 Klasse 2 „Bus“, 266 Klasse 3 „Zweirad“, 529 Klasse

4 „Fußgänger“.
Größe, Lichtverhältnisse und Perspektive der Objekte bzw. Bilder unterscheiden sich zum Teil erheblich.

Für jedes einzelne Objekt gibt es vier Dateien. Zwei Bilddateien im PNG-Format (Portable Network Graphics Format), wobei eines in Farbe und eines in Graustufen ist, und zwei Textdokumente, in denen die Koordinatenpunkte der extrahierten Objektkontur bzw. der konvexen Hülle der Kontur, sowie teilweise Informationen zur Skalierung enthalten sind. Kontur und konvexe Hülle sind die Ergebnisse der Vorverarbeitung in C++. Mit Hilfe eines Hintergrundschätzers wurden die Objekte detektiert und mit Hilfe der *openCV* Funktionen „findContours“ und „convexHull“ wurden die Koordinatenpunkte erzeugt, auf denen die Klassifikation dieser Arbeit basiert. Der große Unterschied zwischen den beiden Konturrepräsentationen ist, dass bei Fußgängern, Spezialfahrzeugen und durch Fehler in der Erkennung von Objektkonturen konkave Objektformen entstehen (vgl. Abb. 4.4), was in der weiteren Bearbeitung zum Teil problematisch ist. Zudem kompensiert die konvexe Hülle teilweise die Fehlerkennung bei Fahrzeugen recht gut, sodass zumindest eine Fahrzeugähnliche Form entsteht. Es wurde dennoch mit beiden Formrepräsentanten gearbeitet, da die Fehler in der Fahrzeugerkennung oft ähnlich sind und möglich ist, dass PKW auf Grund dieser vermeintlich falschen Kontur zuverlässiger erkannt werden als mit der konvexen Hülle, die oft weniger markant ist.

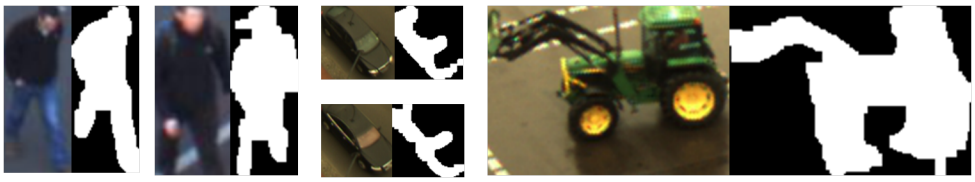

Abbildung 4.4: Objektkonturen.

Die extrahierten Objektkonturen werden in ein Matlab Programm eingelesen. Dort wird das Objektmerkmal "Kontur", bzw. seine Repräsentanten, mit Hilfe von Fourier Deskriptoren (FD) beschrieben. Diese bilden zusammen den Merkmalsvektor des Objektes, der im Klassifikator mit den Klasseneigenschaften verglichen wird. Die Matlab Implementierung in dieser Arbeit erzeugt für drei Klassifikatoren die Klassenzuordnungen der aus den Videos extrahierten Objekte.

4.2 Fourier Deskriptoren

Im ersten Schritt muss die Kontur abgetastet werden, eine Repräsentationsart gewählt und die Fourier Transformation (FT) gebildet werden. Durch Normierung werden die Fourier Koeffizienten der FT zu Fourier Deskriptoren (FD). Das Vorgehen wird im Folgenden etwas näher beschrieben.

Die Implementierung erfolgt in Matlab unter Verwendung sowohl eigener, als auch von Mathworks bereitgestellter Matlab-Funktionen.

Zunächst werden die Bilddaten samt ihrer Klassenzugehörigkeit eingelesen. In Abhängigkeit der Wahl der Parameter „*fw-abt-N-b*“ unterscheidet sich der weitere Ablauf. Die Parameter sollen hier kurz vorgestellt werden, da Sie wichtige Stellhebel für die Justierung der Klassifikatoren darstellen und daher in diesem Kapitel häufig erwähnt werden.

„fw“ bzw. „formWahl“ Dieser Parameter bestimmt, ob die Koordinaten der Objektkontur ($formWahl$=0) oder der konvexen Hülle ($formWahl$=1) verwendet werden. Zu beachten ist, dass für $formWahl$=0 keine winkelgleiche Abtastung möglich ist, da es hier bei konkaven Formen zu Uneindeutigkeiten kommen kann.

„abt“ bzw. „Abtastung“ *Abtastung* legt die Art der Abtastung und die Shape Signature fest, also wie die Form repräsentiert wird.

- $Abtastung = 0$ Die Form wird äquidistant an N Punkten abgetastet. Es werden die Centroid Distances (Schwerpunktdistanzen) erzeugt und eine reelle FT durchgeführt.
- $Abtastung = 1$ Wie 0, aber mit winkelgleicher Abtastung der konvexen Hülle.
- $Abtastung = 2$ Äquidistante Abtastung der Form. Koordinaten werden als komplexe Zahlen interpretiert und eine komplexe FT durchgeführt.
- $Abtastung = 3$ Wie 2, aber mit winkelgleicher Abtastung der konvexen Hülle.

„N“ Anzahl der Abtastpunkte. Da die Algorithmen mit der FFT arbeiten, sollte hier eine Zweierpotenz gewählt werden. In den Experimenten werden zwar auch unterschiedliche Einstellungen für N betrachtet, jedoch ist die Relevanz dieses Parameters

als niedrig einzustufen. Ein sehr niedriges N sollte vermieden werden, da es einen großen Informationsverlust bewirken kann. Hohe Abtastraten sind dafür rechenintensiver, erhalten aber vorerst die Informationen, die später durch Reduktion der FD Anzahl noch verringert werden kann.

„b“ Anzahl verwendeter Fourier Deskriptoren. Da nicht alle hochfrequenten Informationen (Details) benötigt werden, bestimmt man mit diesem Parameter die Anzahl der FD, die für die Klassifikation als Objektmerkmale dienen sollen.

In Abhängigkeit der Wahl des Parameters *formWahl* werden die Koordinatenpunkte aus der Datei mit den ursprünglichen Objektkonturen oder der Datei mit den konvexen Hüllen ausgelesen. Anschließend wird die Kontur an N Punkten in gleichen Abständen abgetastet. Wurde die äquidistante Abtastung gewählt, so wird die Kontur in N gleichlange Konturstücke unterteilt. Für die winkelgleiche Abtastung bilden die N Schnittpunkte von Kontur und $\frac{N}{2}$ sternförmig angeordneten Geraden die Abtastpunkte.

Werden komplexe Zahlen als Shape Signature verwendet, so bilden die Koordinaten direkt die Eingangswerte für die komplexe Fourier Transformation. Für die reelle Fourier Transformation müssen die Koordinaten erst in Centroid Distances umgerechnet werden, indem von jedem Konturpunkt die Distanz zum Polygonschwerpunkt bestimmt wird. Diese eindimensionalen Daten bilden die Eingangswerte für die reelle FT.

Die reelle Fourier Transformation wurde selbst implementiert und mit der Matlabfunktion *fft()* verifiziert, die komplexe Fourier Transformation wurde direkt über *fft()* ermittelt. Die Algorithmen der Matlabfunktion entsprechen, wie der Name *fft* schon sagt, einer Fast Fourier Transformation.

Normierung Die Normierung der Fourier Koeffizienten erfolgt für komplexe und reelle FT gemäß der Beschreibung aus Abschnitt 3.2 . Die nun translations-, skalierungs-, rotations- und startpunktinvarianten Fourier Koeffizienten bilden die Fourier Deskriptoren. Deren Anzahl wird nun auf b reduziert.

Bei den FDs der reellen FT können nun einfach die ersten b Elemente des Vektors

mit den $\frac{N}{2}$ FDs (nullte Frequenz fällt durch Normierung weg) als Merkmalsvektor für die Klassifikation verwendet werden. Hintergrund ist, dass die negativen Frequenzen nicht betrachtet werden, da sie den komplex Konjugierten der positiven Frequenzen entsprechen und somit keine zusätzliche Information enthalten.

Der Vektor mit den FDs der komplexen FT enthält sowohl positive als auch negative Frequenzen und somit N Elemente, von denen nach der Normierung noch N-2 FDs übrig bleiben. Eine positive und die entsprechende negative Frequenz bilden jeweils ein Koeffizientenpaar. Der Merkmalsvektor für die Klassifikation wird daher von b Koeffizientenpaaren gebildet.

Anhand der so ermittelten Merkmalsvektoren, auch Feature Vectors genannt, ordnen Klassifikatoren Objekte in die Klasse ein, zu der sie die größte Ähnlichkeit aufweisen.

4.3 Bewertungsmechanismen

Bevor die Implementierung der Klassifikatoren samt Experimenten erläutert wird, soll an dieser Stelle noch kurz die Implementierung der Methoden vorgestellt werden, die hauptsächlich zur Bewertung herangezogen wurden. Zu nennen sind in diesem Zusammenhang die Kreuzvalidierung, ROC Kurve, Kappa-Koeffizient und Accuracy.

Kreuzvalidierung

Die Indizes der Test- und Trainingsdaten für die K-fache Kreuzvalidierung werden mit Hilfe der Matlabfunktion *Index=crossvalind('Kfold', N, K)* generiert. In dieser Arbeit wurde eine k-fache stratifizierte Kreuzvalidierung verwendet, indem der Funktion anstelle der Objektanzahl N, ein Vektor mit Klassenzugehörigkeiten übergeben wird. Der optimale Wert K wurde für jeden Klassifikator mit Hilfe einer ROC Kurve ermittelt. In weniger eindeutigen Fällen wurde k=10 angenommen, da sich dieser Wert in der Praxis bewährt hat. [RTL09]

Performance Bewertung/Konfusionsmatrix

Für die Beurteilung der Performance der Klassifikatoren wurde die Matlab Funktion

classperf(truelabels, classout) aus der Bioinformatics Toolbox genutzt. Der Funktion werden die wahren und die vom Klassifikator geschätzten Klassenzuordnungen übergeben. Mit diesen Informationen werden eine Konfusionsmatrix sowie Kennzahlen, wie beispielsweise Fehlerrate, Sensitivität oder Spezifität erzeugt. Wie in Abschnitt 3.5.1 bereits erläutert, werden die Klassifikatoren unter Zuhilfenahme des Verfahrens der Kreuzvalidierung traininiert und bewertet. Aus diesem Grund wird vor Durchführung der Kreuzvalidierung mittels *cp = classperf(truelabels)* ein leeres „Classifier Performance Object (CPO)“ erzeugt und initialisiert. Da das Klassifikationsergebnis aller Kreuzvalidierungsdurchläufe von Interesse ist, muss das Classifier Performance Object in jedem Durchlauf mit Hilfe von *cp = classperf(cp,. . .)* aktualisiert werden.

ROC Kurve

Die Receiver Operating Characteristic findet in dieser Arbeit Anwendung um Parameter der Klassifikatoren zu testen, um die optimale Anzahl der Partitionen der Kreuzvalidierungen zu ermitteln und um diskrete Klassifikatoren (die in der ROC jeweils nur einen Punkt ergeben) zu vergleichen.

Die ROC arbeitet eigentlich mit Matrizen, in denen jedem Objekt ein Score zugeordnet ist. Sie sucht dann den Entscheidungsschwellwert (Threshold), für den die TPR maximal und die FPR minimal wird. Da die hier verwendeten diskreten Klassifikatoren keine Scores ausgeben, sondern direkt die Klassenzuordnungen, wurde auf die Verwendung der ROC Standardfunktionen aus der Neural Network Toolbox verzichtet. Stattdessen wurde eine eigene Funktion *plotROC()* kreiert. Die Funktion plottet eine Graphik, in der es für jede Klasse eine ROC Kurve gibt. Die Kurven bzw. Punkte definieren sich über die True Positive und False Positive Ratios für die jeweiligen Parametereinstellungen. Die ROC Kurven werden ausgewertet und der Parameterwert mit dem besten Ergebnis in die jeweilige Klassifikatorimplementierung übernommen.

4.4 Clustering-Verfahren

4.4.1 Dendrogramm

Es wird die Funktion *dendrogram(linkage(pdist(X,distance),method),parameter)* mit euklidischer Distanz als *distance* und der *method* „Single Linkage“[2] verwendet.

Die Matlab-Funktion $dendrogram(Z, parameter)$ erzeugt einen Dendrogramm-Plot für den hierarchischen, binären Clusterbaum Z. Der Clusterbaum kann mit Hilfe der Funktionen *pdist*, die eine Distanzmatrix der Merkmalsvektoren erzeugt, und $linkage$ generiert werden. Letztere Funktion erzeugt aus der Distanzmatrix einen agglomerierenden, bottom-up generierten, hierarchischen Clusterbaum Z. Das heißt, sie fusioniert schrittweise Cluster mit großer Ähnlichkeit, also kleinster Distanz. Der Betrachter kann mit Hilfe des Dendrogramm-Plots auf den ersten Blick erkennen, ob es eine klare Abgrenzung von Clustern gibt oder nicht. Auch die konkrete Zuordnung der Objekte zu Clustern kann für die jeweiligen Partitionslevels (horizontale Ebene) abgelesen werden. Die Wahl des Partitionslevels kann an der angestrebten Clusteranzahl oder einer Mindestdistanz (Höhe der vertikalen Linien) zwischen Clustern orientiert werden.

Experimente Es wurden Dendrogramme für unterschiedliche Einstellungen in Bezug auf die Generierung der FDs erzeugt und bewertet. Für $DatensatzI$ konnten die fünf Klassen für keine Kombination der Parameter fw-*abt*-*N*-*b* als Cluster erkannt werden. Besonders schlecht war das Clustering auf Basis der extrahierten Fahrzeugkonturen fw=0. Etwas mehr Struktur ergab sich für die Parameter 1-2-*N*-*b*, wobei nur ein Fourier Deskriptor (b=1) oft bessere Ergebnisse brachte als die Verwendung mehrerer FDs. Dies ist bereits ein Hinweis darauf, dass die vorliegenden Daten keine klaren Clusterstrukturen enthalten. Zwar konnten gewisse Strukturen erkannt werden, die geschätzten Clusterzuordnungen waren jedoch durchwegs mangelhaft. So wurden für alle Parametereinstellungen circa 99% aller Objekte demselben Cluster zugeordnet. Den restlichen Clustern gehörten jeweils nur sehr wenige Objekte an, die vermutlich Ausreißer sind und sich daher vom Rest der Objekte abheben. Einige Ergebnisse der

[2] Distanz bezieht sich auf den kleinsten Abstand aller Objektpaarungen der beiden Cluster.

Experimente sind Abb. 4.5 zu entnehmen. Zum Vergleich ist ganz links ein idealisiertes Dendrogramm abgebildet, was das schwache Ergebnis der beiden rechten Dendrogramme aus den Experimenten nochmals verdeutlicht Das mittlere Dendrogramm in

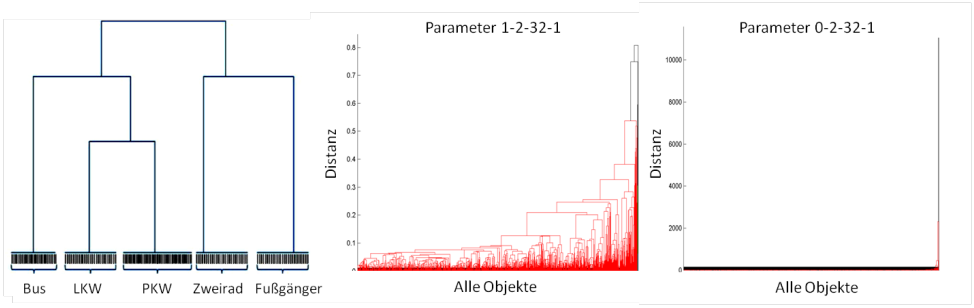

Abbildung 4.5: Dendrogramme.

4.5 ist Ergebnis eines Clustering mit den Einstellungen 1-2-32-1, das Rechte zeigt die innere Struktur für 0-2-32-1. Im Letzteren lassen sich die erkannten Cluster sogar nur erahnen, da bis auf Einzelfälle (Peak ganz rechts) alle Objekte als ähnlich (schwarzer flacher Balken über fast alle Objekte) erachtet wurden. Insgesamt konnte für keine Parametereinstellung eine Clusterstruktur entdeckt werden, die einen Zusammenhang mit den tatsächlichen Objektklassen aufweist. Die Analyse der Daten mittels Dendrogramm zeigt somit, dass es nicht leicht sein wird, eine sinnvolle Abgrenzung der Klassen zu finden. Die Daten bilden keine klaren, abgegrenzten und eindeutigen Gruppierungen. Es sind, zumindest mit diesem Verfahren, keine Strukturen in den Daten erkennbar, die eine Einteilung in mehr als zwei Klassen unterstützen würden.

4.4.2 k-Means

Das k-Means Clustering wird in Matlab mit der Funktion *IDX = kmeans(X,k)* realisiert. Hierbei werden die Objekte in k Cluster geteilt, indem die Summe der clusterinternen (kumulierten) euklidischen Distanzen zwischen Merkmalsvektoren und Clusterschwerpunkt über alle Cluster minimiert wird.

Für das Clustering der Fahrzeugkonturmerkmale wird getestet, mit welcher Clusterzahl der Silhouettenkoeffizient maximal wird. Der k-Means Algorithmus wird mit zufällig gewählten Startschwerpunkten mehrfach durchlaufen und die Clusterzahl mit dem größten Silhouettenkoeffizient ausgegeben. Für die Berechnung dieses Gütema-

ßes dienen die Matlabfunktion *silhouette()* , welche die Silhouettenwerte der einzelnen Objekte enthält, und die Funktion *mean()*, die den Durchschnitt aller Silhouettenwerte und somit den Silhouettenkoeffizient des Clusterings ausgibt.

Das Clustering wird auch mit der angestrebten Klassenzahl getestet und kann für diesen Fall mit der entsprechenden Zielklassifikation verglichen werden. Um das Risiko lokaler Optima zu reduzieren, werden die Startwerte für den Clusterschwerpunkt nicht komplett zufällig aus der Menge der Merkmalsvektoren bestimmt. Stattdessen werden Merkmalsvektoren von Objekten aus der jeweiligen Klasse gewählt. Die Wahl erfolgt innerhalb der Klassen zufällig und wird gemäß der gewünschten Wiederholungszahl mehrfach (in dieser Implementierung zehnfach) durchgeführt.

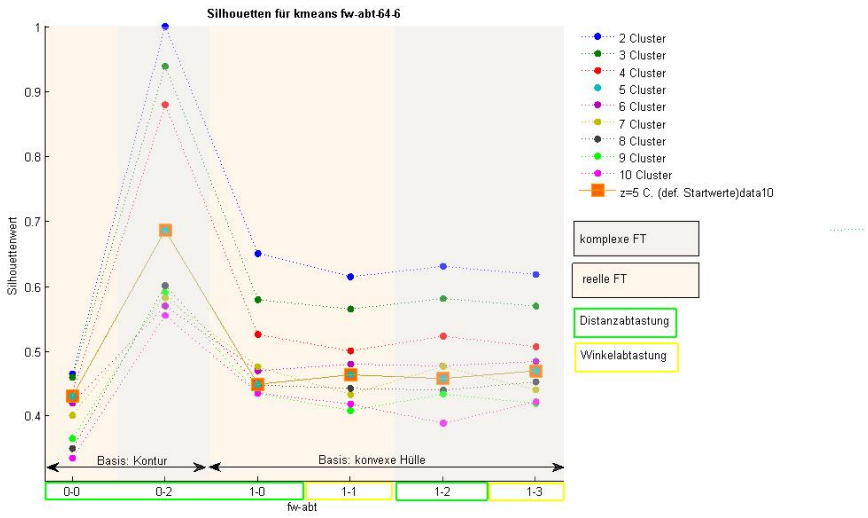

Abbildung 4.6: Silhouettenwerte k-Means fw-abt-64-6.

Experimente Tests zeigten für alle Clusterzahlen k ein klar dominantes Ergebnis für die Einstellung 0-2-N-b, also basierend auf extrahierten Konturen und komplexer Fourier Transformation (siehe Abb. 4.6). Es wurde außerdem festgestellt, dass die Ergebnisse für k=5, was in diesem Fall der tatsächlichen Klassenzahl entsprach, unabhängig davon war, ob die Startwerte zufällig aus der Menge aller Objekte oder gezielt aus der jeweiligen Klasse genommen wurden (an Abb. 4.6 orange und türkis gekennzeichnet). Durchläufe mit niedrigen Clusterzahlen schnitten deutlich besser ab als solche mit Höheren. So können ganz klar zwei Cluster unterschieden werden. Auch für drei und vier

Cluster wird mit einem Silhouettenwert von rund 0,9 (bei einem Maximum von 1) eine starke Clusterstruktur erkannt. Für die angestrebten fünf Cluster fällt das Ergebnis mit knapp 0,7 durchschnittlich aus, kann aber noch als brauchbare Clusterstruktur bezeichnet werden. Die dominanten Parametereinstellungen wurden anschließend für weitere Tests verwendet. Abb. 4.7 zeigt die Ergebnisse der Clusteringdurchläufe mit den Einstellungen 0-2-*N*-*b* mit N=2^i (i=2,...,8) und b=1, ..., $\frac{N}{2}$ (maximal 15). Auch hier ergeben sich für zwei bis vier Cluster mit Abstand die besten Ergebnisse. Während die Zahl der Abtastpunkte kaum Einfluss auf den Silhouettenwert zu haben scheint, dominiert vor allem bei den höheren Clusterzahlen klar das Clustering mit nur einem Fourier Deskriptor. Da es mit nur einem Deskriptor nicht möglich sein kann, zwischen mehreren Verkehrsobjekt-Klassen zu differenzieren, kann angenommen werden, dass die Elemente der Objektklassen keine Cluster bilden, die mit dem k-Means Verfahren zu erkennen sind.

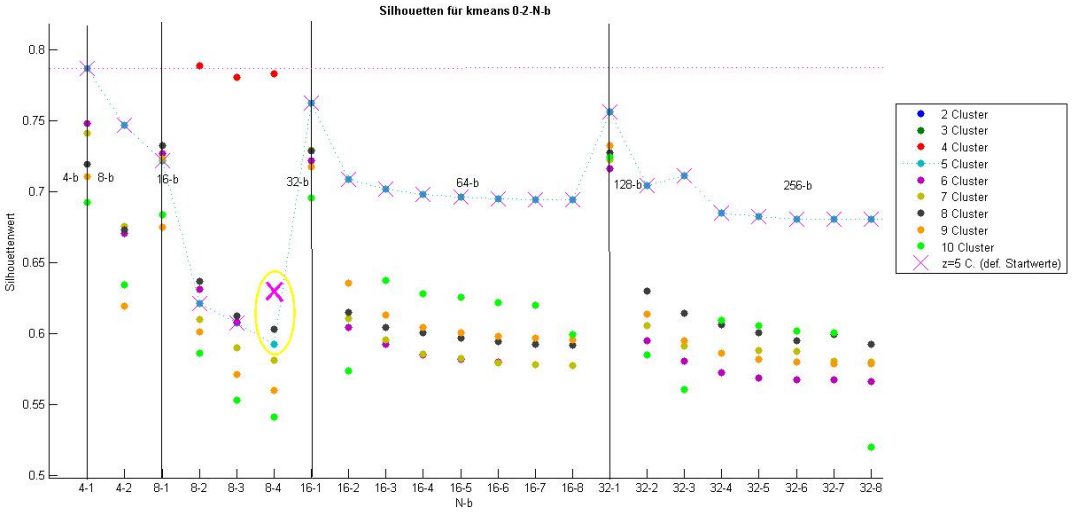

Abbildung 4.7: Silhouettenwerte k-Means 0-2-N-b.

Für die anvisierte Klassenzahl „5“, erreicht der Silhouettenwert mit der Einstellung 0-2-4-1 sein Maximum. Dies ist bezeichnend, da bei Verwendung von vier Abtastpunkten und nur einem FD ein großer der Konturinformation verloren sind. Dennoch ist auch die Accuracy für diese Einstellung mit 36,7% am höchsten. Eine maximale Erkennungsrate von deutlich unter 50% kann jedoch nicht als ausreichend gewertet werden. Der Plot der Silhouettenwerte (vgl. Abb. 4.8 (b)) aller Objekte für die genannten

Einstellungen lässt Aufschlüsse auf den Hintergrund des Ergebnisses zu. So zeigt die Graphik, dass der Großteil aller Objekte in derselben Klasse zugeordnet werden, der zweitgrößte Cluster beinhaltete einen Teil vermutlich falsch zugeordneter Objekte (erkennbar an vorhandenen negativen Silhouettenwerten) und den verbelibenden drei Clustern wurden kaum Objekte zugeordnet. Die Accuracy von 36,7% ist darauf zurückzuführen, dass der Cluster dem fast alle Objekte angehören, auch einen Großteil der Elemente der stark vertretenen Klasse *PKW* (40% aller Elemente) enthält, und diese als „True Positive“ gewertet wurden.

Zur Verdeutlichung ist Abb. 4.8 (a) zu entnehmen, wie der Silhouettenplot im Idealfall aussehen sollte. Die Ergebnisse in (b) könnten ein Hinweis darauf sein, dass die Klassifikation mit Unterscheidung von drei oder vier statt fünf Klassen bessere Ergebnisse bringen könnte, da hier etwas klarere Clusterstrukturen erkannt werden.

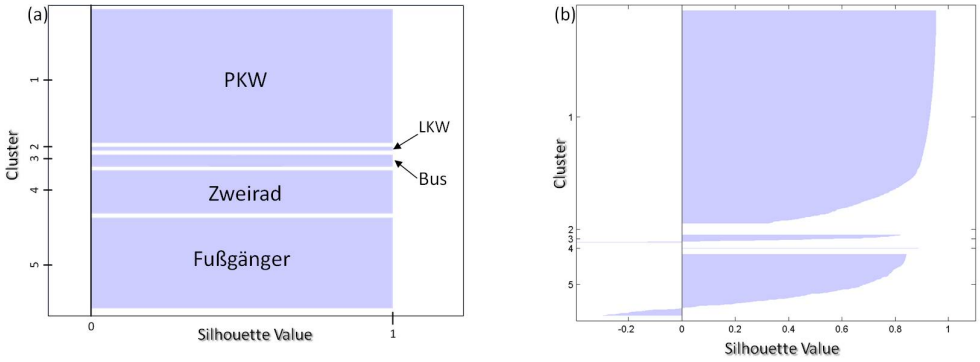

Abbildung 4.8: Silhouette 5-Means(a)idealisiert (b)0-2-4-1.

4.5 Klassifikatoren

Eine erfolgreiche Klassifikation setzt voraus, dass alle Parameter in Bezug auf den Klassifikator, aber auch in Bezug auf die vorgelagerten Prozesse geeignet gewählt sind. Die Parameter der jeweiligen Verfahren werden im Zusammenhang mit der Erläuterung der Implementierung vorgestellt. Bevor im Versuch Erkennungsraten ermittelt werden können, erfolgen eine Reihe von Vorversuchen, die gewährleisten sollen, dass alle Stellhebel so gut wie möglich eingestellt und aufeinander abgestimmt sind. Hierfür wurde für alle Klassifikatoren nach folgendem Schema vorgegangen:

1. Vorversuche zur Ermittlung möglicher Werte für die Parameter *fw-abt-N-b* und *K* nach dem „Trail and Error"-Prinzip und unter Verwendung von Erfahrungswerten und Hinweisen aus der Literatur. Für klassifikatorspezifische Parameter werden zunächst die Standardwerte verwendet, wenn nicht auf Grundlage der Literaturrecherche anders entschieden wurde.
2. Bewertung der sechs Kombinationsmöglichkeiten für die Parameter *fw* und *abt*. Die restlichen Parameter werden gemäß (1) festgelegt. Auswahl der besten Kombination für weitere Tests.
3. Finden des besten *K* für die *K*-fache Kreuzvalidierung mit Hilfe von ROC Kurven.
4. Bestimmung geeigneter Werte für *N-b* mittels ROC Kurven und teilweise Performance Diagramm.
5. Ggf. Wiederholung von Punkt(2) mit neu ermittelten Parametern *N-b* und *K*.
6. Ggf. Tests zur klassifikatorspezifische Parameter.

4.5.1 Minimum Distance Classifier (MDC)

Der in dieser Arbeit implementierte MDC ordnet Objekte derjenigen Klasse zu, zu deren Mittelpunkt die euklidische Distanz minimal ist. Die entsprechende Funktion wurde selber erzeugt und nutzt die Matlabfunktion *pdist*() zur Generierung der Distanzmatrix. Training und Test wurden im Rahmen einer *K*-fachen Kreuzvalidierung durchgeführt, wobei die Klassifikationsgüte für Werte *K*=2 bis 20 mittels ROC Kurven verglichen wurde.

Experimente

fw-abt Nachdem erste Vorversuche bereits ergaben, dass die Klassifikation mit den Parametern *N*=64, *b*=4 und *K*=18 brauchbare Ergebnisse liefert, wurden diese Werte für Experimente zum Finden der besten Kombination von *fw* und *abt* verwendet (vgl. Tabelle 4.1). Es wurden jeweils die Accuracy(*Acc*), der Kappa-Koeffizient (κ) sowie das

beste K ermittelt. Da sich in vielen Fällen die 20-fache Kreuzvalidierung als die Vorteilhafteste herausstellte, wurden einige Tests nochmal mit K=20 durchgeführt. Mit einer Accuracy von 37% und einem Kappa-Koeffizienten von 0,24 erwies sich 1-3-64-4 (*fw-abt-N-b*) als beste Wahl. Noch ausstehende Experimente werden für den MDC daher basierend auf winkelgleich abgetasteten konvexen Hüllen und Complex Coordinates als Shape Signature durchgeführt.

Test	1	2	3	4	5	6	7	8	9	10	11	12	13
K	18	20	18	20	18	20	18	20	20	20	18	20	18
fw/abt	1/0	1/0	1/1	1/1	1/2	1/2	1/3	1/3	1/3	1/3	0/0	0/0	0/2
N/b	64/4	64/4	64/4	64/4	64/4	64/4	64/4	64/4	128/4	32/4	64/4	64/4	64/4
κ	0,21	0,23	0,21	0,24	0,23	0,24	0,21	0,24	0,24	0,24	0,15	0,19	-0,01
Acc	0,33	0,35	0,33	0,37	0,34	0,36	0,34	0,37	0,37	0,37	0,29	0,32	0,31
Besseres K	20	-	20	-	20	-	20	-	-	-	20	-	-[3]

Tabelle 4.1: Ergebnisse MDC für verschiedene *fw-abt*.

Mit den Parametern 1-3-64-4 und K=20 werden im Anschluss die, in den Vorversuchen nur grob getesteten Parameter, nun nochmals systematisch überprüft.

K Zunächst wurden ROC Kurven für verschiedene Werte K einer K-fachen Kreuzvalidierung erzeugt und bewertet (vgl. Abb. 4.9a). Auf den ersten Blick ist ersichtlich, dass es für jede Klasse Werte außerhalb des rötlich hinterlegtem Bereichs gibt, der eine unzureichende Performance kennzeichnet. Der ROC Plot wurde außerdem in Matlab stark vergrößert betrachtet, um die zugehörigen Parameter zu den geplotteten Datenpunkten erkennen zu könne. Die Analyse ergab eine gute Eignung von K=17,18,19,20, wobei die 20fache Kreuzvalidierung meist die besten Ergebnisse brachte. Aus diesem Grund wurde für den Minimum Distance Classifier pauschal eine 20-fache Kreuzvalidierung für weitere Experimente festgelegt. Zu beachten ist, dass die Ergebnisse bei Verwendung der konvexen Hülle als Kontur (*fw*=1) deutlich besser waren als jene mit der Kontur vom Hintergrundschätzer (*fw*=0). Die Kombination mit komplexer FT (0-2-64-4) führte sogar zu so schlechten Ergebnissen, dass die ROC Kurve für fast alle K unterhalb der Diagonalen lag und der Klassifikator somit schlechter arbeitet, als eine Zufallsmaschine (vgl. Abb. 4.9b). Der κ-Koeffizient für diese Parametereinstellung war entsprechend auch kleiner Null.

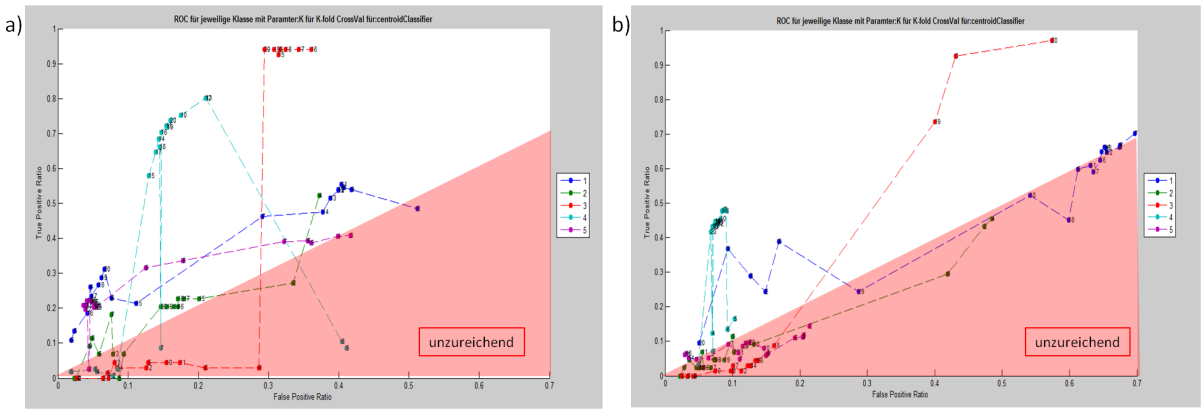

Abbildung 4.9: ROC für Parameter K mit (a) 1-3-64-4 und (b) 0-2-64-4.

N-b Mit den geprüften Parametern wurde dann die Klassifikationsgüte für unterschiedliche Abtastraten in Kombination mit der Anzahl betrachteter FDs bewertet. Aus der Gegenüberstellung der Accuracy und Kappa Werte für alle Kombinationen *N-b* mit N=2^i (i=2, ...,8) und b=1, ...,$\frac{N}{2}$ (maximal aber bis b=20) sind folgende Erkenntnisse möglich (vgl. Abb. 4.10):

- Kein Einfluss durch Veränderung der Abtastrate, solange diese nicht irrational klein gewählt wird ($N < 16$).
- Extrem schlechte Ergebnisse mit nur einem FD (b=1).
- Für steigende Anzahl berücksichtigter FDs (b) konvergiert die Klassifikationsgüte stets (logarithmisch) gegen ihr Maximum.
- Da aus Gründen der Recheneffizienz ein möglichst kleiner b Wert angestrebt wird, verwendet man anstelle des Maximums den Wert, ab dem die Verbesserung nur noch minimal ist.

Aus den Tests und Analysen wurde eine Standardeinstellung für den MDC von 1-3-64-9 mit 20-facher Kreuzvalidierung als geeignete Parameterkombination gefolgert. Die Klassifikationsgüte befindet sich mit einer Accuracy von unter 40% jedoch auf einem nicht zufriedenstellenden Niveau. Dies kann mehrere Ursachen haben. Wie das Clustering bereits gezeigt hat, liegen keine klaren Klassengrenzen vor. Deshalb überrascht es nicht, dass ein so simples Verfahren wie der MDC Schwierigkeiten hat, die Objekte korrekt zu klassifizieren. Das schlechte Ergebnis kann konkret darin begrün-

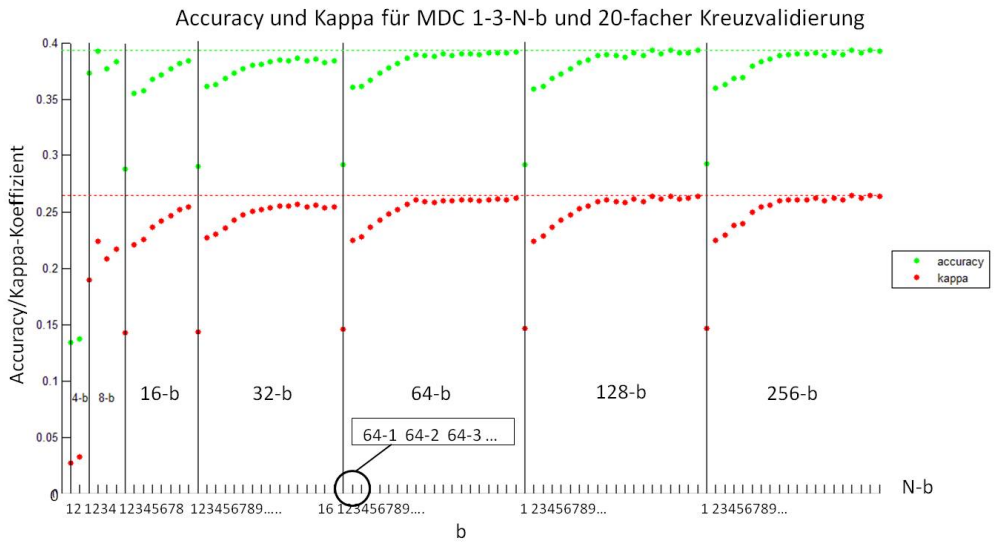

Abbildung 4.10: Accuracy und κ-Koeffizient für 1-3-N-b und K=20.

det sein, dass Ausreißer eine Verschiebung des Schwerpunktes ihrer Klasse bewirken. Schon wenige Ausreißer können, gerade bei kleinen Klassen, den Prototyp so verzerren, dass er die Eigenschaften der Klasse nur noch unzureichend widerspiegelt. Für das Training sollte daher ein von Ausreißern bereinigter Datensatz verwendet werden.

4.5.2 k-Nearest-Neighbor Klassifikator (k-NN)

Der *k*-Nearest-Neighbor (*k*-NN) Klassifikator wird in Matlab mit der Funktion *knnclassify()* realisiert. Als Distanzmaß wurde die euklidische Distanz und als Entscheidungsregel das Mehrheitsprinzip gewählt.

Experimente Zur Bestimmung der Parameter wurden mehrere Versuche durchgeführt.

fw-abt Zunächst wurde die Performance aller Kombinationen *fw-abt*-64-4 überprüft und jeweils die am besten geeigneten Werte für k (k-NN) und K (K-fache Kreuzvalidierung) ermittelt (vgl. Tabelle 4.2). Das beste Ergebnis brachte mit einer Accuracy von rund 76% die 7-NN Klassifikation mit der Parametereinstellung 0-2-64-4 und 14-facher

Kreuzvalidierung.

Test	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k/K	7/9	7/14	7/9	9/6	7/9	7/8	7/9	7/9	5/8	7/9	3/12	7/14	7/14	7/14
fw/abt	1/0	1/0	1/1	1/1	1/2	1/2	1/3	0/0	0/0	0/2	0/2	0/2	0/2	0/2
N/b	64/4	64/4	64/4	64/4	64/4	64/4	64/4	64/4	64/4	64/4	64/4	64/4	128/4	32/4
κ	0,55	0,56	0,51	0,54	0,61	0,61	0,55	0,47	0,46	0,62	0,60	0,62	0,62	0,62
Acc	0,71	0,72	0,69	0,71	0,75	0,75	0,71	0,66	0,65	0,76	0,74	0,76	0,76	0,76
better k	-	-	9	-	-	-	-	5	-	3	7	-	-	-
better K	14	-	6	-	8	-	-	8	-	12	14	-	-	-

Tabelle 4.2: Ergebnisse k-NN für verschiedene fw-abt.

Nicht immer führten Parametereinstellungen, die laut ROC Kurve geeignet sind, auch zur besseren Accuracy. Hintergrund ist, dass es in der Regel keine universell „beste“ Lösung gibt. In Abhängigkeit davon, wie konservativ die Klassifikation sein soll und wie stark Fehlklassifikation kleiner Klassen (LKW,Bus) gewichtet werden sollen, ergeben sich durchaus sehr unterschiedliche Parametereinstellungen für das „beste“ Ergebnis. Eine hohe Accuracy kann sich beispielsweise auch ergeben, wenn eine wenig vertretene Klasse (hier LKW und Bus) geschlossen nicht erkannt wurde. Sollen auch kleine Klassen erkannt werden, wird dafür eine niedrigere Accuracy möglicherweise in Kauf genommen.

K Wie Abb. 4.11a zeigt, liegen die Ergebnisse für unterschiedliche Werte von K nahe zusammen. Häufig ergaben die 9-fache oder 14-fache Kreuzvalidierung hierbei für alle Klassen gute Ergebnisse. Insgesamt zeigt Abb. 4.11a akzeptable Ergebnisse für alle K und alle Klassen. Allerdings schnitt Klasse 2 *LKW* vergleichsweise schlecht ab.

k Mit den Einstellungen 0-2-64-4 und K=14 werden in Abb.4.11 b) die ROC Kurven der einzelnen Klassen für den Parameter k gezeigt. Erkennbar ist hier der Zielkonflikt durch die unterschiedlichen Klassen. So ist beispielsweise zur Erkennung von LKW und PKW ein sehr kleines k gut. Um Fußgänger, Radfahrer und Busse zu erkennen jedoch ein möglichst großer Wert für k. Einen für alle Klassen akzeptablen Kompromiss bietet zum Beispiel der Wert k=7.

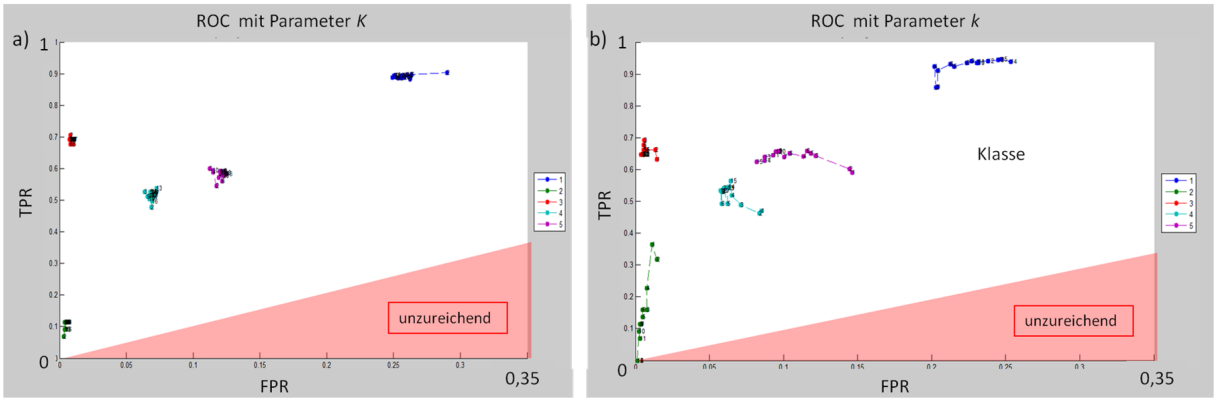

Abbildung 4.11: KNN (a)ROC für K mit 1-3-64-4 & k=5 (b) ROC für k mit 0-2-64-4 & K=14

N-b Ähnlich wie die Auswertung für den MDC, beeinflusst die Abtastrate die Performance nicht, solange gilt $N \geq 16$. Der Performance Vergleich für verschiedene Werte b ergab ähnlich gute Ergebnisse für alle $b \geq 3$ (vgl. Abb 4.12). Da es kein klares Optimum in Bezug auf die Kombination N-b gibt, wurden im Folgenden Tests mit 0-2-128-18, 0-2-64-10, 1-2-32-11 und 0-2-32-6 durchgeführt.

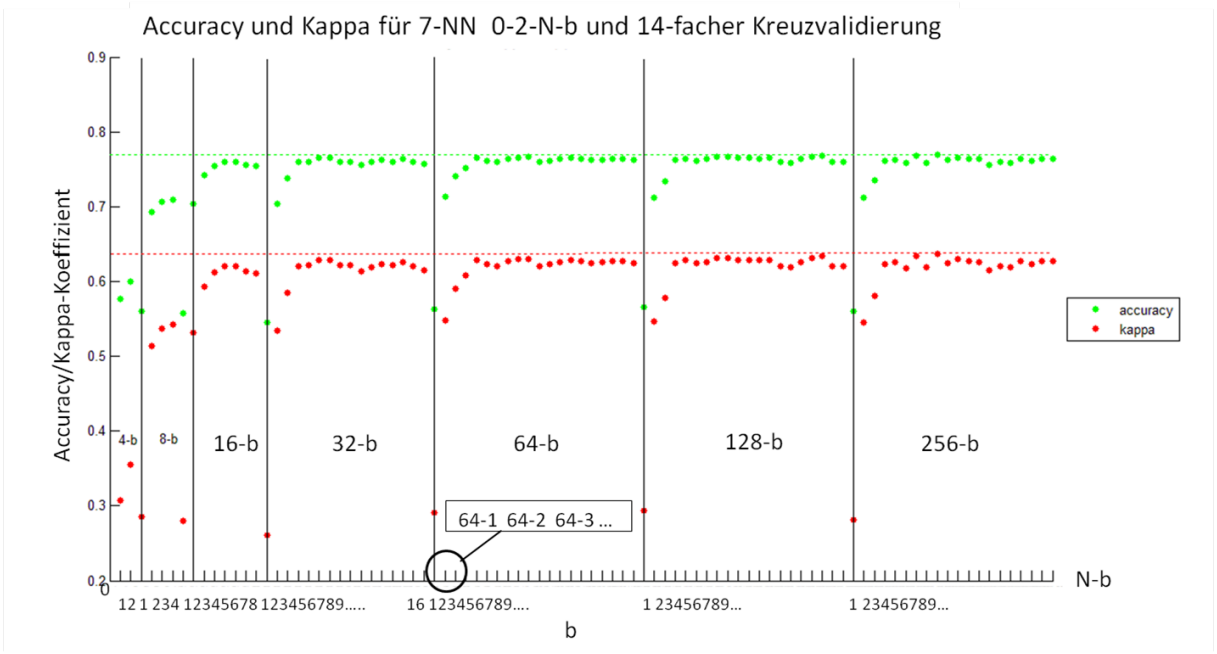

Abbildung 4.12: Performance für eine 7-NN mit 0-2-N-b & K=14.

Der 7-NN Klassifikator zeigt mit einer Accuracy von rund 75% deutlich bessere Ergeb-

nisse als der zuvor betrachtete MDC.

4.5.3 Support Vector Machine (SVM)

Für eine binäre SVM bietet die „Matlab Statistics Toolbox“ die Funktion *svmclassify(SVMStruct, Sample)*. Die Funktion klassifiziert die Objekte des Vektors *Sample* mit Hilfe der Daten der SVM-Klassifikator-Struktur $SVMStruct$ und gibt die Klassen in einem Vektor aus. *SVMStruct* wird mit Hilfe der Funktion *svmtrain()* erzeugt und übergibt Daten wie Stützvektoren, Bias oder Informationen zur Kernel Funktion an *svmclassify*.

Die Funktion $svmtrain()$ trainiert einen SVM Klassifikator mit den annotierten Trainingsdaten. $svmtrain()$ ermöglicht die Einstellung verschiedener Parameter, wie *Kernel, Method* oder *BoxConstraint C*.

Die Kernel Funktion hat die Aufgabe, die Trainingsdaten in den Kernel Raum zu mappen. Folgende Kernel Funktionen stehen in Matlab zur Verfügung:

- Linearer/Punkt- Kernel
- Quadratischer Kernel
- Gaussian Radial Basis Function (RBF) Kernel mit Skalierungsfaktor σ (standardmäßig σ=1)
- Polynomieller Kernel (standardmäßig 3. Ordnung)
- Multilayer Perceptron (MLP) Kernel

Gewählt wurde in dieser Arbeit der Gaussian Radial Basis Function (RBF) Kernel, da dieser in der Literatur für vergleichbare Aufgaben als geeignet befunden wurde und für seine universelle Einsetzbarkeit bekannt ist. [THM07] [ZAW07]

Dem Finden der Hyperebene dienen in Matlab die folgenden Methoden:

- Quadratische Programmierung (QP): Soft-Margin SVM, minimiert L^2-Norm Problem (Schlupfvariable ζ_i^2).
- Sequentielle Minimale Optimierung (SMO): Soft-Margin SVM, minimiert L^1-Norm Problem (Schlupfvariable ζ_i^1).

- Least-Squares/Kleinste Quadrate (LS): im Gegensatz zu den anderen Methoden minimiert LS ein Set linearer Gleichungen anstelle eines konvexen Problems der Quadratischen Programmierung.

Experimente

fw-abt Die Bestimmung geeigneter Parametereinstellungen für fw-abt-N-b und K erfolgte analog zum Vorgehen bei MDC und k-NN. Mit den Parametern aus den Vorversuchen N-b=64-4, K=10, RBF-Kernel und LS Methode wurden Accuracy und κ-Koeffizient für die verschiedenen Kombinationen fw-abt ermittelt. Tabelle 4.3 sind die Ergebnisse zu entnehmen.

Test	1	2	3	4	5	6
K	10	10	10	10	10	10
fw/abt	1/0	1/1	1/2	1/3	0/0	0/2
N/b	64/4	64/4	64/4	64/4	64/4	64/4
κ	0,55	0,51	0,65	0,56	0,46	0,59
Acc	0,69	0,67	0,77	0,70	0,61	0,72

Tabelle 4.3: Ergebnisse SVM für verschiedene fw-abt.

K Die ROC Kurven mit den unterschiedlichen Werten für den Parameter K, ergaben kein klares Ergebnis. Einige Werte wie z. B. K=2 konnten ausgeschlossen werden. Für weitere Versuche wurde $K = 10$ ausgewählt, da sich dieser Wert in vielen Anwendungsfällen bewährt hat - sowohl in der Literatur als auch in eigenen Tests.

N-b Für die Ermittlung der Parameter N und b wurden ein Performance-Diagramm und ROC-Kurven erzeugt. Die Erkenntnisse der ROC-Analyse der jeweiligen Klassen (vgl. Abb. 4.13) sind zum einen, wie bei den anderen Klassifikatoren auch, dass für die Wahl von N, solange $N >$16, keinen Einfluss auf die Klassifikationsgüte hat. Zum anderen konnte festgestellt werden, dass die Klassifikation mit steigender Anzahl b FDs

immer konservativer wird. Einzige Ausnahme sind hier die Klasse "Fußgänger", bei denen zwischen b=3 und b=9 keine Ordnung erkannt werden konnte. Abgesehen von Klasse 2 „LKW“, erreichen alle Klassen hohe TPRs von über 80% bei vergleichsweise kleinen FPRs von unter 20%. Für die unterschiedlichen Klassen wurden folgende Parameterwerte als geeignet befunden:

Klasse	1	2	3	4	5
N-b	64-6	128-8	128-8	64-6	64-6

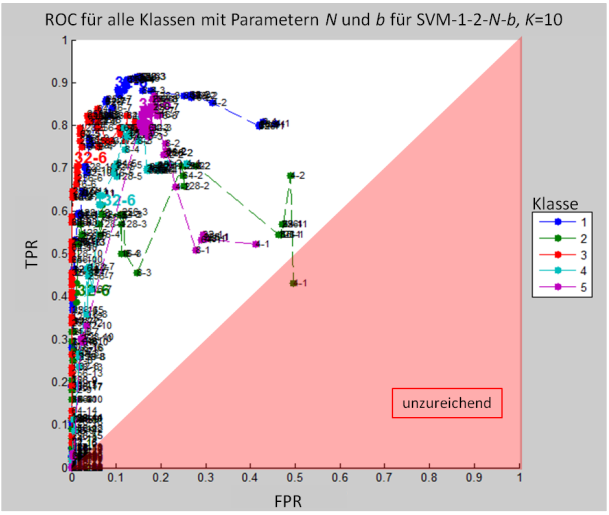

Abbildung 4.13: ROC für *N-b* mit 1-2-*N-b* & $K = 10$.

Kernel Abb. 4.14a zeigt die Klassifikationsgüte für unterschiedliche Kernels in Form einer ROC Kurve. Der Vergleich wurde mit Standardparametereinstellungen durchgeführt und kann daher nur als grobe Orientierung dienen. Wie aus der Abbildung ersichtlich ist, übertrifft der RBF-Kernel in den meisten Fällen die anderen Kernel, was die Wahl des RBF-Kernel im betrachteten Anwendungsfall bestätigt. Der Kernel kann mit Hilfe des Parameters σ skaliert werden. Dieser wird gemeinsam mit dem Parameter C bestimmt.

***C*, σ** Wie Abb.4.14b zeigt, gibt es für die Wahl von σ mehrere ungefähr gleichwertige Möglichkeiten. Wählt man σ=1 und C=*BoxConstraint*=1, so erhält man für alle Klassen ein vergleichsweise gutes Klassifikationsergebnis. Ähnlich gute Ergebnisse bringt die Kombination σ=1, C=0,1 bzw. C=0,01. Der Wert C gewichtet die Schlupfvariable und dient somit der Bestrafung von Ausreißern. Ein hoher C Wert bedeutet eine strenge

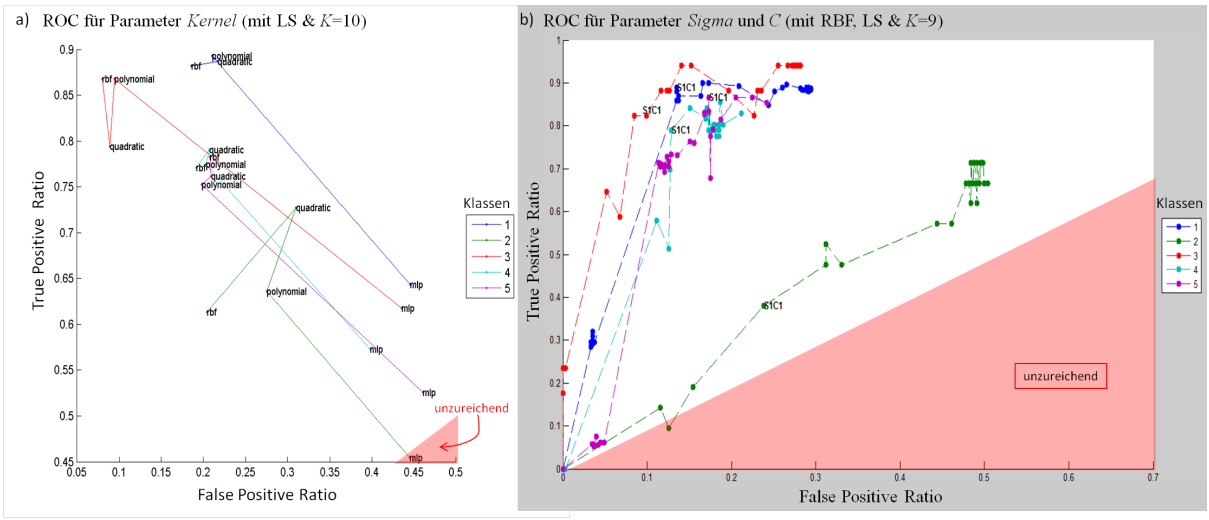

Abbildung 4.14: ROC für (a)*Kernel* (b)σ.

Trennung zwischen den Klassen - die „soft margin“ ist in diesem Fall wenig ausgeprägt.

Method Abbildung 4.15 zeigt eine ROC Kurve, welche die Leistungsfähigkeit des Klassifikators unter Verwendung von LS, QP und SMO im Vergleich für jede Klasse aufzeigt. Generell brachten QP und LS stets vergleichsweise konservative Ergebnisse.

In dieser Arbeit wurde die Methode *LeastSquares* gewählt. Der Klassifikator ist mit dieser Methode deutlich schneller als mit *QP* und etwas schneller als *SMO*. Die Klassifikationsgüte war vergleichbar mit der unter Nutzung von *QP* und die False Positive Rate war generell besser als unter Nutzung von *SMO*. Das Programm läuft mit *LS* außerdem deutlich stabiler und ist weniger speicherintensiv. Für die Nutzung von *QP* ist außerdem eine „Optimization Toolbox“ Lizenz nötig.

Mehrklassen-SVM Um SVM als Klassifikatoren für Mehrklassen-Probleme nutzen zu können müssen die Ergebnisse der binären SVM verschmolzen werden. Konkret wurde hier das one-versus-all (OVA) Prinzip gewählt. Es wird also für jede Klasse getestet, ob ein Objekt zu der jeweiligen Klasse gehört oder zu den restlichen Klassen. Ungünstig ist, dass es somit möglich ist, dass ein Objekt keiner oder aber gleich mehreren Klassen zugewiesen wird. In dieser Arbeit wurden daher zwei SVM Ergebnisse generiert. Eines, das nur eindeutige Klassenzuweisungen anzeigt und alle anderen

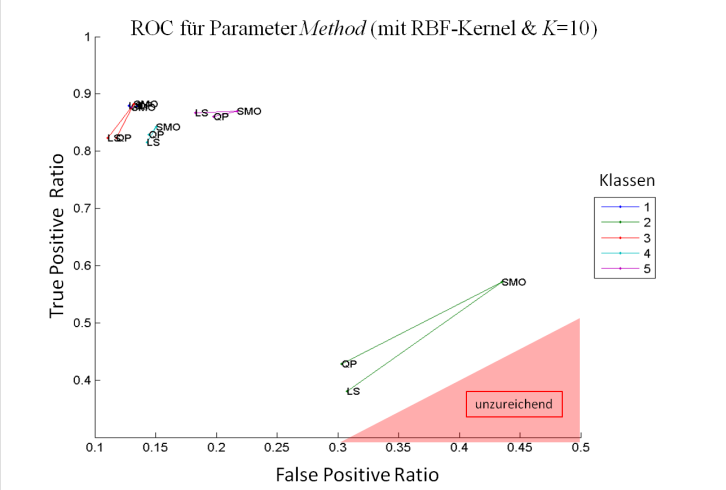

Abbildung 4.15: ROC für *Method*.

Objekte als nicht klassifiziert ausweist - also auch jene, die mehreren Klassen zugeordnet wurden. Das zweite Ergebnis weist allen Objekten, die in mindestens eine Klasse passen, ein Klassenlabel zu - die restlichen Elemente bekommen das Klassenlabel 0. Die Entscheidung der Klassenzuordnung erfolgt bei Objekten die mehreren Klassen zugeordnet wurden anhand eines Distanzmaßes. Die Funktion *svmclassify()* aus der Matlab Toolbox ist nicht in der Lage eine Distanz auszugeben, weshalb eine eigene Funktion *mySVMdistance* eingeführt wurde. Diese berechnet mit Hilfe der Ergebnisse der Funktion *svmtrain* (Kernelfunktion, Stützvektoren, Bias, etc.) die Distanzen, deren Vorzeichen für die Klassenzuordnung verantwortlich sind. Je weiter die Distanzwerte von Null entfernt sind, umso eindeutiger ist die Klassenzugehörigkeit. Der Algorithmus ordnet einem Objekt folglich jene Klasse zu, für die das Distanzmaß maximal ist. Es soll an dieser Stelle darauf hingewiesen werden, dass es sich bei diesem Distanzmaß nicht um tatsächliche Distanzen handelt. [Mat13b]

Für die Mehrklassen-SVM wurde ein Performance Diagramm ausgewertet, das in Abb. 4.16 zu sehen ist. Das Diagramm zeigt Accuracy und Kappa-Koeffizient für die Mehrklassen-SVM mit den Parametern 1-2-*N*-*b* und 10-facher Kreuzvalidierung. Es konnte gezeigt werden, dass die Verwendung von sehr wenigen (1-2) oder sehr vielen (>7) Fourier Deskriptoren schlechtere Ergebnisse bringt, als ein b zwischen drei und sieben. Die maximale Accuracy von knapp 80% wird beispielsweise mit sechs FDs erreicht, weshalb für die Mehrklassen-SVM die Parameter 1-2-32-6 mit $K = 14$ festgelegt

wurden.

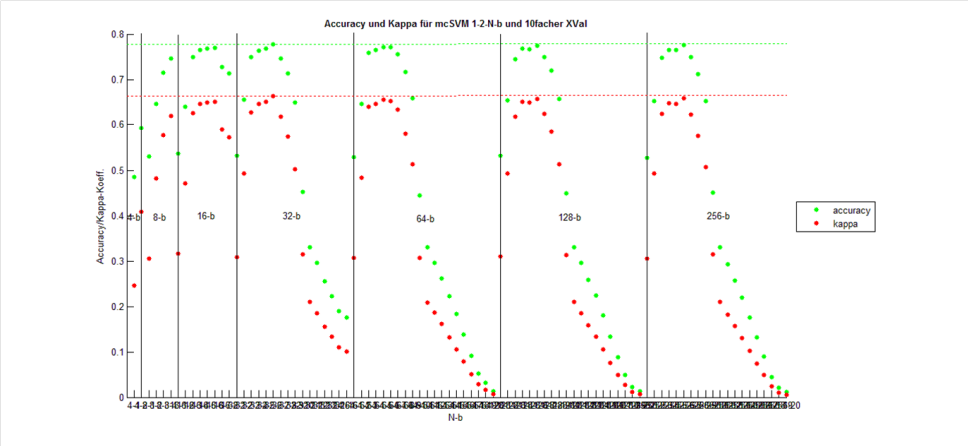

Abbildung 4.16: Performance für Mehrklassen-SVM mit 1-2-N-b & $K = 10$.

4.6 Gesamtvergleich der Klassifikatoren

Nachdem für alle Klassifikatoren geeignete Parametereinstellungen ermittelt wurden, erfolgt ein Vergleich der drei Klassifikatoren MDC, k-NN und (Mehrklassen-)SVM. Die Tests wurden mit *Datensatz I* und unter Verwendung der jeweiligen geprüften Parametereinstellungen durchgeführt. Abb. 4.17 zeigt die Erkennungsraten (Accuracy) der drei Klassifikatoren, sowie der binären SVM für die einzelnen Klassen. Letztere ist insbesondere für die stark repräsentierten Klassen, wie *PKW* oder *Fußgänger* aussagekräftig. Bei kleinen Klassen führen konservative Klassifikationen sehr schnell zu extrem guten Accuracy Werten, weshalb die guten Ergebnisse für *LKW* und *Bus* in Abb. 4.17 täuschen (Beispielrechnung für eine Klasse die nur 2,5% der Gesamtdaten ausmacht: TPR=0,6 und FPR=0,02 ergibt Accuracy=0,97).

Um die Eignung der unterschiedlichen Konturformen, Shape Signatures und Abtastarten bewerten zu können, wurden die Ergebnisse der Klassifikationen mit den sechs unterschiedlichen Kombinationen der Parameter fw und abt verglichen. Wie aus Abb. 4.18 ersichtlich ist, kann für k-NN und SVM Klassifikatoren eine leichte Dominanz der äquidistant abgetasteten und als komplexe Koordinaten dargestellten konvexen Hülle (fw-abt=1-2)erkannt werden. Der MDC zeigt auf Basis der konvexen Hüllen

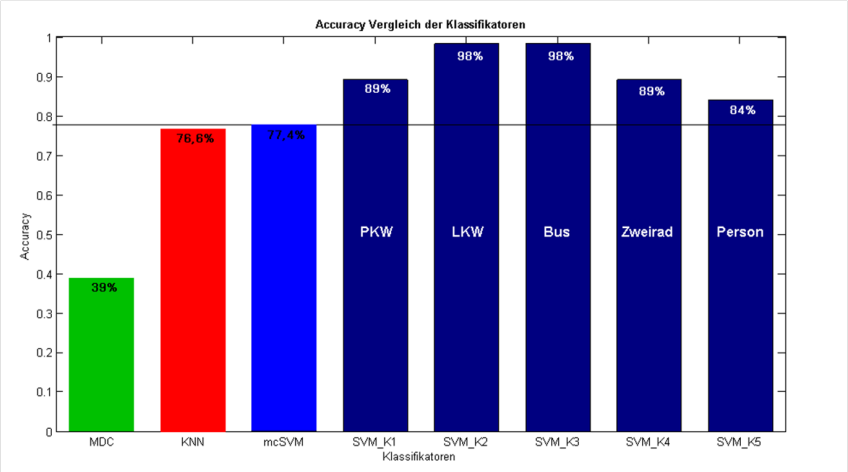

Abbildung 4.17: Performance MDC, k-NN, multiclass/binary SVM.

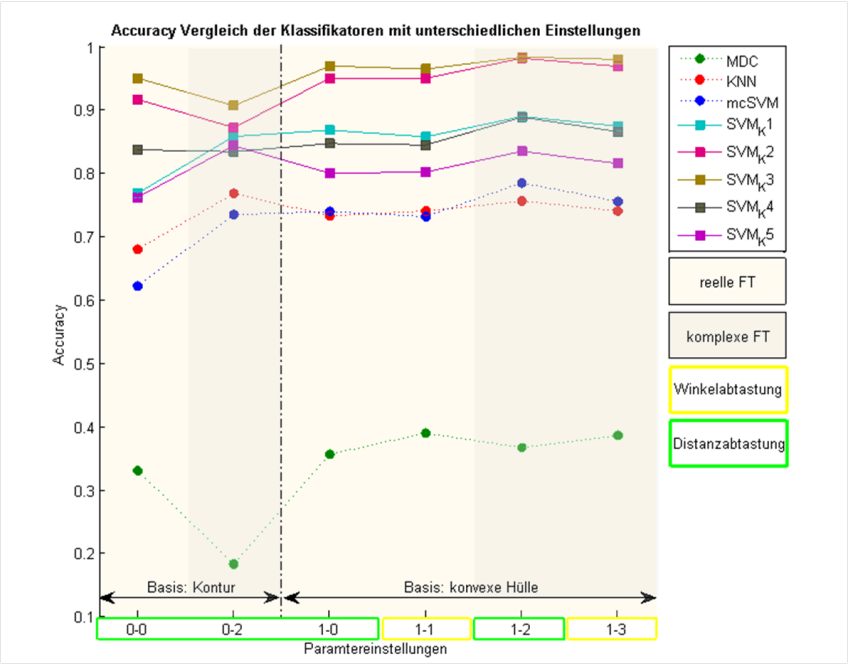

Abbildung 4.18: Accuracy Vergleich der Klassifikatoren.

der Objekte relativ konstant die besten Ergebnisse, allerdings auf einem deutlich niedrigeren Niveau als die anderen beiden Verfahren.

Um zu prüfen inwiefern die Verwendung von individuell angepassten Parametern tatsächlich einen Einfluss auf die Klassifikationsgüte hatte, wurde probehalber der gleiche Vergleich unter Verwendung einheitlicher Parameterwerte N=64 bzw. N=128, b=6 und K=9 durchgeführt. Die Accuracy Werte erreichten ein ähnliches Niveau wie mit individuellen Parametereinstellungen. Dies kann zum einen darauf zurückgeführt werden, dass die Abtastrate ab einem Wert von 32 Abtastpunkten keine Auswirkungen auf die Güte hat. Zum anderen stellen drei bis zehn Fourier Deskriptoren meist eine gute Wahr lar, da somit ausreichend Informationen und dennoch nicht zu viele Details enthalten sind.

Mit einheitlichen Parameterwerten von 1-2-128-6 und K=9 für alle drei Klassifikatoren, wurden auf Basis der drei Datensätze I, II und III weitere Tests durchgeführt. Zum Vergleich wurde auch der bisher betrachtete Datensatz I verwendet. Abbildung 4.19a) zeigt die ROC Kurve der fünf Objektklassen. Als Parameter wurden die Klassifikatoren verwendet. Es zeigt sich wieder, dass der MDC deutlich schwächer ist als die anderen beiden Verfahren. Der 7-NN Klassifikator erreicht für die meisten Klassen leicht schlechtere Ergebnisse als die binäre SVM. Diese Tendenz wurde auch durch die Accuracy Werte der Verfahren bestätigt, die in Abb. 4.19b) dargestellt sind.

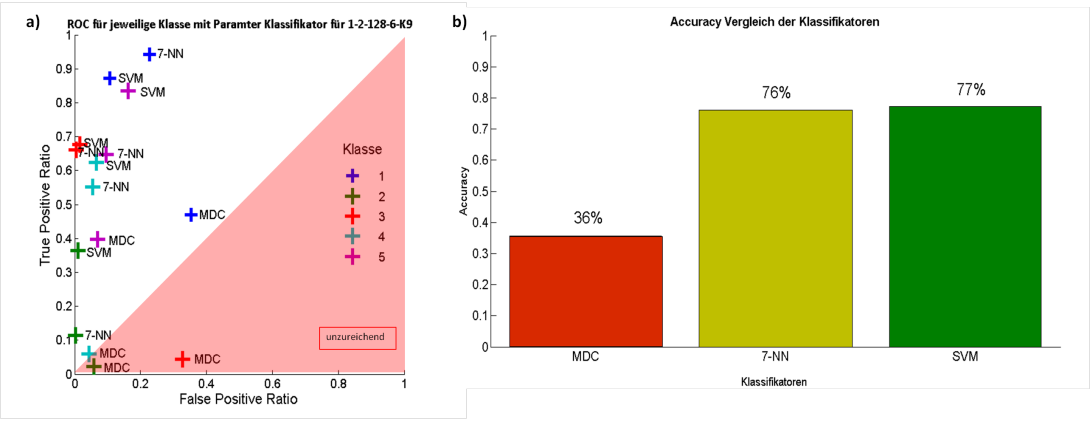

Abbildung 4.19: Datensatz I - ROC und Accuracy der Klassifikatoren.

Der bisher verwendete Datensatz enthält Objekte, die aus den verschiedensten Blickwinkeln zu sehen sind, was die Klassifikationsaufgabe erschwert. Um zu klären inwie-

fern sich das Ergebnis verbessert, wenn diese Vielfalt etwas reduziert wird, wurden auch Tests mit Datensatz II durchgeführt. Der Datensatz beinhaltet nur Objekte, die aus einer schräg seitlichen Perspektive zu sehen sind. Wie die Ergebnisse in Abb. 4.20 zeigen, konnten nur begrenzt Verbesserungen erreicht werden. PKW (Klasse 1) konnten mit TPR Werten von ca. 90% wieder sehr gut erkannt werden (vgl. Abb. 4.20a), bei der Erkennung von Fußgängern (Klasse 5) kann sogar eine deutliche Verbesserung beobachtet werden. Zweiräder (Klasse 4) und Busse (Klasse 3) hingegen wurden schlechter erkannt als mit Datensatz I. Die TPR der Klasse LKW (Klasse 2) verbesserte sich nur für den SVM KLassifikator und blieb auf einem verhältnismäßig niedrigen Niveau. Insgesamt konnte die Accuracy des MDC von 36% auf 50% erhöht werden, was eine erhebliche Verbesserung darstellt (vgl. Abb. 4.20b). Das Niveau von k-NN und SVM liegt, wie auch mit Datensatz I, im Bereich von knapp 80%. Der k-NN arbeitete mit Datensatz II etwas zuverlässiger als vorher und konnte so eine leicht höhere Accuracy erreichen, als die SVM, welche keine Verbesserung durch den vermeintlich einfacher zu klassifizierenden Datensatz zeigt. Die Ergebnisse deuten darauf hin, dass ein komplexer Klassifikator wie die SVM keine Schwierigkeiten mit den verschiedenen Blickwinkeln haben. Einfachere Methoden hingegen konnten mit dem leicht idealisierten Datensatz deutlich bessere Ergebnisse erzielen, als mit Datensatz I.

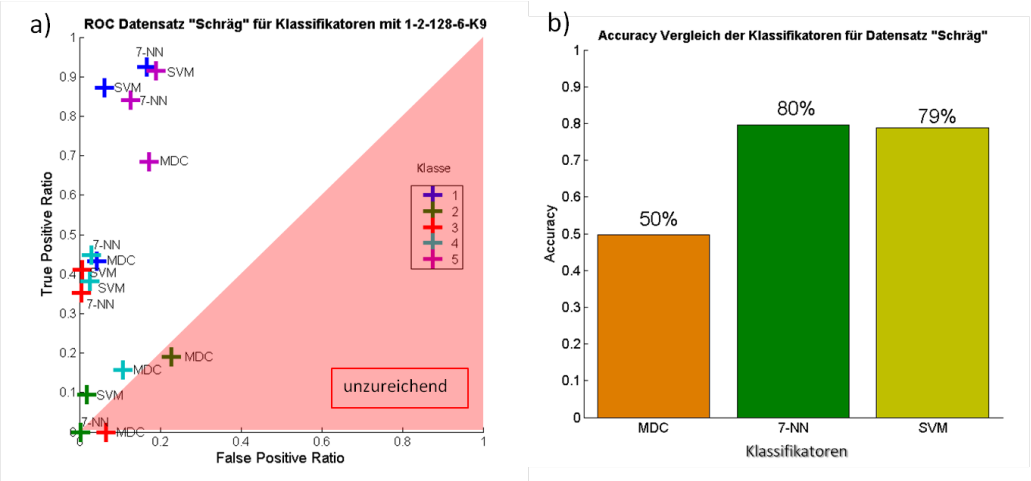

Abbildung 4.20: Datensatz II - ROC und Accuracy der Klassifikatoren.

Experimente mit Clustering-Verfahren in Abschnitt 4.4 hatten auf eine Struktur mit maximal vier Clustern hingedeutet. Um diese These zu überprüfen, wurde Datensatz I

von fünf auf vier Klassen reduziert. Aus Analysen der Konfusionsmatrizen aus Experimenten mit dem ursprünglichen Datensatz, konnte geschlussfolgert werden, dass die Klassen LKW und PKW im Merkmalsraum sehr nahe beieinander liegen. So wurden Objekte der Klasse LKW zu 32% korrekt zugeordnet. Genauso oft wurden sie jedoch der Klasse PKW zugeordnet. Dies legt die Vermutung nahe, dass eine Gemeinschaftsklasse PKW&LKW zu besseren Ergebnissen führen könnte. Zu beachten ist jedoch, dass die Klasse LKW mit unter 3% an der Gesamtobjektzahl nur schwach vertreten ist und daher durch eine verbesserte Zuordnung diser Objekte keine große Erhöhung der Accuracy zu erwarten ist. Die Ergebnisse, die in Abb. 4.21a dargestellt sind, bestätigen diese Vermutung nur zum Teil. So verbesserte sich zwar die Accuracy der Klassifikatoren k-NN und SVM nur um 2%-Punkte im Vergleich zu Tests mit Datensatz I, der MDC hingegen konnte von der Klassenfusion profitieren und erhöhte die Erkennungsrate auf 48% (vgl. Abb. 4.21b). Abbildung 4.21a zeigt insgesamt gute und durchaus verbesserte TPR Werte. Zu einem gewissen Maß geht dies jedoch zu Lasten der FPR, die insgesamt etwas zugenommen hat.

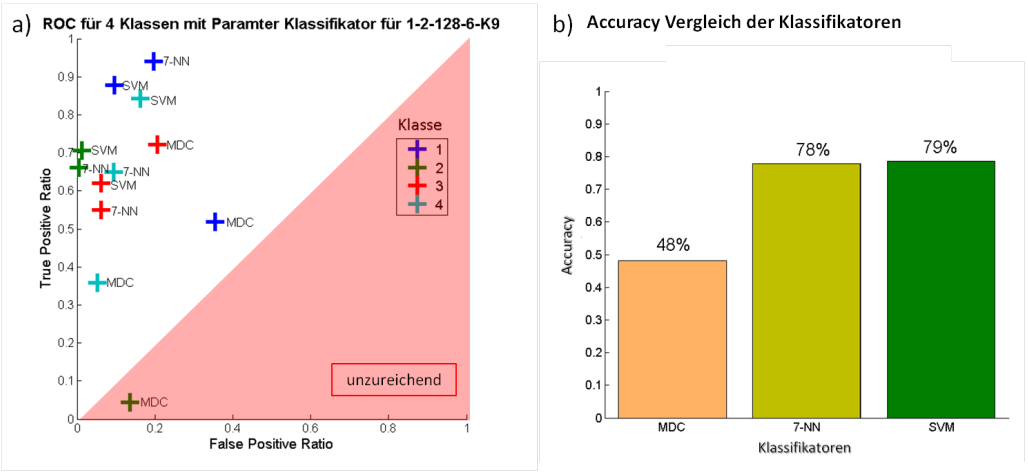

Abbildung 4.21: Datensatz III - ROC und Accuracy der Klassifikatoren.

Insgesamt zeigten die Tests mit den drei unterschiedlichen Datensätzen überraschend ähnliche Ergebnisse in Bezug auf k-NN und SVM Klassifikator. Die etwas weniger komplexen Datensätze II und III konnten die Erkennungsraten zwar teilweise anheben, jedoch beim k-NN um maximal 4%-Punkte und bei der SVM um nur 2%-Punkte. Für den vergleichsweise simplen Klassifikationsansatz des MDC konnte eine

Steigerung um 14%-Punkte erreicht werden, was einer Verbesserung von knapp 40% entspricht. In Tabelle 4.4 sind die Accuracy Werte der Klassifikatoren für die drei Datensätze zusammenfassend dargestellt. Einen Überblick über die Konfusionsmatrizen der einzelnen Klassifikatoren für den jeweiligen Datensatz sind Abb. 4.22 zu entnehmen.

Klassifikator	MDC	7-NN	SVM
Datensatz I	35,7	76,0	77,3
Datensatz II	49,5	79,7	78,5
Datensatz III	48,2	77,9	78,7

Tabelle 4.4: Accuracy auf Basis verschiedener Datensätzen.

Klassifikation mit den Einstellungen 1-2-128-6, K=9 für drei Datensätze

	Konfusionsmatrix SVM					Konfusionsmatrix 7-NN					Konfusionsmatrix MDC				
Klasse	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
	Datensatz I					Datensatz I					Datensatz I				
1	699	14	17	16	40	753	34	16	44	104	377	24	64	28	204
2	5	14	1	2	1	4	7	0	0	3	80	1	1	4	14
3	10	2	46	0	3	4	0	46	0	3	270	17	3	161	86
4	14	2	0	146	51	10	1	0	142	76	49	1	0	22	12
5	57	5	4	85	424	34	2	6	80	343	29	1	0	51	213
-	20	7	0	17	10	0	0	0	0	0	0	0	0	0	0
	Datensatz II					Datensatz II					Datensatz II				
1	247	8	4	5	9	269	18	5	15	32	126	8	1	0	9
2	3	2	0	1	1	1	0	0	0	1	74	4	0	49	35
3	2	1	6	0	1	1	1	6	0	1	36	1	0	6	3
4	5	1	0	26	8	4	0	0	34	15	16	2	0	12	50
5	23	4	4	42	279	16	2	6	27	259	39	6	16	9	211
	11	5	3	2	10	0	0	0	0	0	0	0	0	0	0
	Datensatz III					Datensatz III					Datensatz III				
1	742	17	22	38		796	18	49	106		440	64	31	211	
2	10	46	0	3		3	45	0	4		203	3	12	14	
3	15	0	146	50		12	0	145	74		176	1	191	113	
4	58	4	83	422		38	5	72	345		30	0	32	191	
5	24	1	15	16		0	0	0	0		0	0	0	0	

Abbildung 4.22: Konfusionsmatrizen der Klassifikatoren.

Insgesamt können diese Ergebnisse als akzeptabel eingestuft werden. Durch Maßnahmen zur Optimierung könnte jedoch eine Verbesserung erreicht werden, was Gegenstand der Diskussion in Kapitel 5 ist.

Kapitel 5

Bewertung und Diskussion

Die Experimente in Kapitel 4 konnten zeigen, dass sich Fourier Deskriptoren in Kombination mit Klassifikatoren wie k-NN oder Mehrklassen-SVM durchaus als alleinstehendes Verfahren zur Verkehrsobjektklassifikation in Bildfolgen realer Verkehrsvideos eignen. Für viele Anwendungen sind jedoch Erkennungsraten notwendig, die deutlich über den hier erreichten 80% liegen. Im Folgenden wird erörtert, inwiefern der Ansatz dennoch geeignet ist, wo Stärken und Schwächen liegen und welche Möglichkeiten es bezogen auf Verbesserungen gibt.

Es wurden in dieser Arbeit zwei Clustering-Verfahren auf die Daten von über 1.700 Verkehrsobjekten angewandt. Das Dendrogramm dient der Visualisierung innerer Strukturen von Datensätzen. Für Datensatz I konnten keine Strukturen erkannt werden. Die verwendeten Fourier Deskriptoren spannen folglich keinen Merkmalsraum auf, der so klare Gruppierungen aufweist, dass sie mit dem Verfahren der hierarchischen Clusteranalyse erkennbar wären. Es wurde daraufhin mit dem k-Means Clustering ein zweites Verfahren angewendet. Obwohl das Verfahren für fünf Klassen „brauchbare Strukturen“ ermittelte, konnten doch keine bedeutenden Parallelen zu den tatsächlichen Klassenzuordnungen erkannt werden. Das akzeptable Ergebnis des Silhouettenwerts für 5 Cluster und die noch besseren Resultate für weniger Clusterzahlen relativieren sich bei genauerer Betrachtung. Es wurden eigentlich stets nur zwei Klassen unterschieden und einige, offensichtlich starke, Ausreißer führten zu kleinen aber starken zusätzlichen Clusterstrukturen. Diese entsprechen jedoch nicht den erhofften Clustern der Verkehrsteilnehmergruppen. Das Accuracy Ergebnis von 37% ist wenig zufriedenstel-

len spiegelt vermutlich in erser Linie den Anteil, der am meisten vertretenen Klasse wider und weniger eine Erkennungsleistung für alle Klassen. D.h. die TPR für die mit 40% am stärksten repräsentierte Klasse lag bei nahezu 100%, was durch eine unakzeptabel hohe FPR vonentsprechend fast 60% erreicht wurde. Insgesamt kann auf Basis der beiden Clustering-Verfahren noch nicht auf die Eignung der Fourier Deskriptoren als charakteristische Merkmale geschlossen werden. Wie sich zeigen wird, lag dies in erster Linie daran, dass der Datensatz keine linearen und mit simplen Methoden separierbaren Klassenstrukturen aufweist. Möglicherweise hätten höher entwickelte Verfahren wie Self-Organizing-Maps die gesuchten Strukturen besser erkannt.

Auch weniger anspruchsvolle „low-level“ Klassifikatoren, wie der MDC, können die schwierige Klassifikationsaufgabe, die Datansatz I generiert nicht zufriedenstellend lösen. Dies kann im Falle des MDC eine Folge von Verzerrungen der Klassenmitten durch Ausreißer sein. Wie Experimenete zeigten, erhöht sich die Klassifikationsgüte solch einfacher Verfahren bei Reduktion der Komplexität der Klassifikationsaufgabe zum Teil erheblich. Dies wurde überprüft, indem der MDC auf Datensätze mit weniger Klassen bzw. auf Bildfolgen angewandt wurde, in denen die Objekte nur aus einer Perspektive zu sehen sind. Wie am Beispiel von k-NN und SVM gezeigt werden konnte, wirkt sich diese Verbesserung umso weniger aus, je höher entwickelt ein Klassifikator ist. Dies kann darauf zurückgeführt werden, dass Verfahren wie SVM weniger Schwierigkeiten haben, nicht lineare und weniger offensichtliche Klassengrenzen zu finden. Insgesamt konnten für k-NN und SVM akzeptable Erkennungsraten von ca. 80% für fünf Fahrzeugklassen beobachtet werden.

In Bezug auf die Parameter kann festgestellt werden, dass die Abtastrate groß genug gewählt werden sollte, dass nicht schon zu Beginn möglicherweise wichtige Informationen verloren gehen. Die Anzahl der Fourier Deskriptoren betreffend scheinen sieben Deskriptoren ausreichend um die Form zu beschreiben. Die Wahl von konvexer Hülle oder der eigentlichen Kontur beeinflusste zwar das Ergebnis, jedoch kann keine universelle Dominanz festgestellt werden. Abbildung 5.1 zeigt Beispiele für Objekte, die jeweils nur auf Basis von konvexer Hülle oder der eigentlichen Kontur erkannt werden konnten. In Bezug auf die Shape Signature scheinen sowohl Centroid Distances als auch Complex Coordinates geeigent, wobei für k-NN und SVM leicht bessere Ergebnisse mit Centroid Distances nachgewiesen werden konnten. Auch das Abtast-

verfahren stellt keine wichtige Einflussgröße auf das Ergebnis dar, da die erzeugten Konturen bei Abtastraten von über 64 fast identisch sind. Im Vergleich zu anderen For-

Abbildung 5.1: Klassifikationsgüte abhängig von Parameter *fw*.

schungsbeiträgen auf diesem Gebiet erscheinen knapp 80% als Erkennungsrate eher wenig. Zu beachten ist jedoch, dass, im Gegensatz zu vielen anderen in der Literatur vorgestellten Ansätzen, hier keine binäre Klassifikation vorliegt sondern fünf Klassen unterschieden wurden. Betrachtet man lediglich die Entscheidung „PKW“ und „kein PKW“ so können deutlich bessere ergebnisse erzielt werden, wie anhand der binären SVM dargestellt werden konnte. Desweiteren wird häufig mit idealisierten Bildern gearbeitet, sodass die wirklichen Herausforderungen, die in der Verkehrsobjekterkennung durch die schwierigen Umgebungsbedigungen entstehen, umgangen werden. In dieser Arbeit hingegen, wurde mit Daten gearbeitet, die nicht von Störungen wie Schatten, Lichteffekten oder leichten Verdeckungen (starke Verdeckungen wurden auch hier aussortiert) befreit wurden. Außerdem ist die Auflösung der Daten teilweise sehr schlecht und in erheblichem Maß waren Konturen durch Fehler des Hintergrundschätzers gestört. Wie Abb. 5.2 zeigt, konnten Bilder ohne Störungen dieser Art meist gut erkannt werden. Teilweise zu Fehlklassifikationen führten hingegen Störungen, eine sehr geringe Bildauflösung, fehlerhafte Konturen und ungünstige Blickwinkel (vgl. Abb. 5.2, in der ein Radfahrer frontal nicht von einem Fußgänger unterschieden werden konnte). Diese zufällig ausgewählten Beispiele zeigen, dass Fehlklassifikationen oft aus Mängeln in der Bildqualität oder den schwierigen Umfeldbedingungen realer

Verkehrsdaten resultieren.

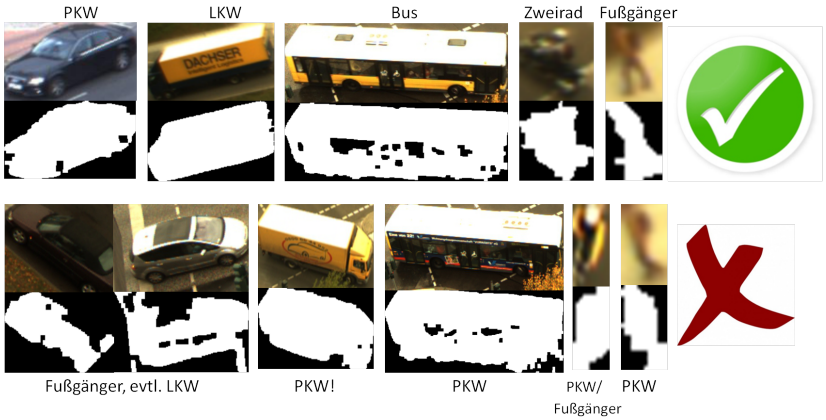

Abbildung 5.2: Beispiele für korrekt und falsch klassifizierte Objekte.

Um die Erkennungsraten trotz der anspruchsvollen, suboptimalen Datengrundlage weiter zu erhöhen und um auch in schwierigen Situationen, wie bei Lichtveränderungen, Teilverdeckungen oder fehlerhaft extrahierten Konturen noch eine ausreichende Klassifikationsgüte gewährleisten zu können, wird eine Fusionierung des vorgestellten konturbasierten Verfahrens mit weiteren Merkmalen empfohlen. Auch eine geschickte Kombination mehrerer Klassifikatoren würde die Ergebnisse verbessern. Im Folgenden werden konkrete Überlegungen zur Optimierung des vorgestellten Verfahrens dargelegt.

Eine Verbesserung der Klassifikationsgüte könnte beispielsweise durch eine gewichtete Fusion der Klassenzuordnungen mehrerer Klassifikatoren erreicht werden. Im Rahmen dieser Arbeit wurden unterschiedliche Klassifikatoren getestet, sie wurden jedoch nicht miteinander kombiniert. Bei Nutzung von Klassifikatoren mit unterschiedlichen Schwachstellen wäre so eine Verbesserung der Klassifikationsgüte auf unkomplizierte Art und Weise möglich. Großes Potential kann des Weiteren in der kombinierten Nutzung mehrerer Merkmale gesehen werden. Durch die Invarianzeigenschaften der Fourier Deskriptoren bleiben wichtige charakteristische Eigenschaften der Objekte ungenutzt. Rotationsinvarianz bewirkt beispielsweise, dass Fußgänger nicht mehr anhand ihrer schmalen, hohen Kontur erkannt werden können, da ein um 90° rotierter PKW möglicherweise eine ähnliche Kontur aufweist. Ein Merkmal mit viel Potential

ist die Objektgröße, die bedingt durch die Skalierungsinvarianz der FD,in dieser Arbeit unberücksichtigt blieb. Auch die Nutzung von Geschwindigkeitsinformationen aus den Videos wird als sinnvoll erachtet. Bei Überschreiten festgelegter Geschwindigkeitsgrenzen, könnten so bestimmte Klassen ausgeschlossen werden. Werden die Objekte über mehrere Frames hinweg getrackt, könnten außerdem die Klassenzuordnungen der letzten Frames als unterstützende Information verwendet werden. So wäre es möglich, eine Änderung der Klassenzuordnung erst zuzulassen, wenn diese Klasse mehrere Frames hintereinander erkannt wurde. Auf diese Weise könnte Robustheit gegenüber Störungen wie temporärer Verdeckung oder kurze Lichteffekte generiert werden. Auch die Kombination der FD mit Deskriptoren, die lokale Bildmerkmale beschreiben, erscheint vielversprechend. Mittels SIFT-Deskriptor könnten die Räder von Fahrzeugen, Nummernschilder, Leuchten oder Aufdrucke auf LKW Planen erkannt werden.

Fourier Deskriptoren stellen einen vergleichsweise robusten, wenig rechenintensiven Ansatz dar und erreichen somit auch bei einer weniger optimalen Datengrundlage akzeptable Ergebnisse. Die Kombination des vorgestellten Verfahrens mit anderen Merkmalsdeskriptoren kann als vielversprechend eingestuft werden, sodass mit relativ geringem Aufwand ein deutlicher Anstieg der Zuverlässigkeit zu erwarten ist.

Kapitel 6

Zusammenfassung und Ausblick

Objektklassifikation ist ein bedeutendes Gebiet der Bildverarbeitung. Durch die vielfältigen Anwendungsmöglichkeiten, Vorzüge wie Kosteneffizienz oder Flexibilität sowie durch stetige Weiterentwicklungen, ist das Interesse an kamerabasierter Objektklassifikation in den letzten Jahren stark gestiegen. Auch im Verkehrswesen wird die Technologie in den unterschiedlichsten Bereichen eingesetzt. In den letzten Jahren hat die Forschung im Bereich der visuellen Klassifikation von Verkehrsobjekten beträchtliche Fortschritte erzielt, doch das Potential ist noch lange nicht ausgeschöpft. So stellt die automatische Erkennung verschiedener Verkehrsteilnehmer auf Grundlage realer Verkehrsvideos noch immer eine Herausforderung dar.

In dieser Arbeit wurde ein möglicher Ansatz der Objektklassifikation untersucht, implementiert und experimentell verifiziert. Eine Literaturrecherche ergab, dass konturbasierte Fourier Deskriptoren für den betrachteten Anwendungsfall ein vielversprechender Ansatz sind. Da es keinen universell geeigneten Klassifikator gibt, wurde die Zuordnung der Objekte zu Klassen mit mehreren Verfahren durchgeführt.

Fourier Deskriptoren sind die normierten Fourier Koeffizienten der transformierten Objektkontur und beschreiben somit die Objektkontur als Summe von gewichteten Sinus- und Kosinusfunktionen verschiedener Frequenzen. Die Objektkontur wird i. d. R. nicht direkt verwendet. Stattdessen wird die Fourier Transformation auf einen Repräsentanten der Kontur angewendet. Hierfür eignen sich beispielsweise Shape Signatures, welche die zweidimensionale Kontur in den eindimensionalen Raum abbilden. Konkret wurde in dieser Arbeit parallel mit zwei verschiedenen Verfahren gear-

beitet, der „Centroid-Distance-Function“ und den „Complex Coordinates“. Aus den Fourier Transformierten dieser Funktionen wurden die Fourier Deskriptoren berechnet. Um bei der Klassifizierung nicht objektspezifische Merkmale, sondern klassenspezifische Merkmale zu vergleichen, wurden nur jene Fourier Deskriptoren verwendet, die niedrigen Frequenzen zugeordnet sind und somit grobe Forminformationen enthalten.

Die Datengrundlage für die Klassifikation bildete ein Datensatz aus 1.712 Objektkonturen, die aus Videoaufnahmen an einem dreiarmigen Knotenpunkt extrahiert und den Klassen PKW, LKW, Bus, Zweirad und Fußgänger zugeordnet wurden. Mit Hilfe von Clustering-Verfahren wurde der Merkmalsraum auf innere Strukturen untersucht, um Aufschlüsse darüber zu gewinnen, ob und wie viele Klassen unterscheidbar sind. Es ergaben sich hierbei jedoch keine Hinweise auf Strukturen, die einen Zusammenhang mit den tatsächlichen Klassen der Verkehrsobjekte aufweisen. Die Klassifikation wurde mit den drei Verfahren Minimum-Distance-Classifier, k-Nearest-Neighbor Klassifikator und OVA-SVM durchgeführt. Im Rahmen einer Kreuzvalidierung wurden die Klassifikatoren mit Vertretern der fünf Objektklassen traininiert und getestet. Für den Minimum Distance Classifier ergab sich eine sehr niedrige Erkennungsrate von nur 40%. Für die anderen beiden Verfahren konnte eine Accuracy von 77% beobachtet werden.
Insgesamt kann festgestellt werden, dass Fourier Deskriptoren gemeinsam mit einem k-NN Klassifikator oder Mehrklassen-SVM, auch unter schwierigen Bedingungen akzeptable Ergebnisse liefern. Für viele Anwendungen ist die hierbei erreichte Erkennungsrate jedoch nicht ausreichend. Um eine Verbesserung zu erreichen, kann entweder die Komplexität der Klassifikationsaufgabe reduziert oder Schwachstellen des Verfahrens durch Kombination mit anderen Verfahren ausgeglichen werden. Da stets eine breite Anwendbarkeit angestrebt wird, sollte Erstgenanntes nicht der bevorzugte Ansatz sein. Bei konkreten Anwendungen in der Praxis kann es jedoch sinnvoll sein, die Komplexität durch Anpassung an die jeweiligen Randbedingungen zu verringern um bessere Ergebnisse zu erreichen. Da kein Verfahren perfekt ist, ist es ein naheliegender Ansatz, unterschiedliche Verfahren so zu kombinieren, dass die Stärken des einen Verfahrens die Schwächen des anderen ausgleichen und umgekehrt. So kann die Kombination des vorgestellten Verfahrens mit anderen Merkmalsdeskriptoren als viel-

versprechend eingestuft werden, sodass mit relativ geringem Aufwand ein deutlicher Anstieg der Zuverlässigkeit zu erwarten ist.

Da der Ansatz der Fourier Deskriptoren eine solide Basis für die Klassifikation von Verkehrsobjekten bildet, sollte dieser Ansatz weiterverfolgt und optimiert werden. Als Gegenstand weiterer Untersuchungen auf diesem Gebiet bieten sich außerdem Verfahren der SIFT- und Wavelet-Familien an. In Bezug auf Clustering- und Klassifikationsverfahren könnte geprüft werden, ob das große Potential der Künstlichen Neuronalen Netze und verbesserter Verfahren der Mehrklassen-SVM für die Klassifikation von Verkehrsobjekten genutzt werden kann.

Literaturverzeichnis

[AM10] M.M. Arzani and M.Jamzad. Car type recognition in highways based on wavelet and contourlet feature extraction. In *2010 International Conference on Signal and Image Processing (ICSIP)*, pages 353 –356, 2010.

[BÖ2] Andreas Böckert. Vehicle detection and classification in video sequences. Master's thesis, Tekniska Hogskolan i Linkoping, 2002.

[BAB+07] Christian Böhm, Johannes Assfalg, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Jörg Sander, and Matthias Schubert. Clustering. 2007.

[Bar12] Kai Uwe Barthel. Keypointdetektion der scale-invariant feature transform. 2012.

[Bra12] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2012.

[Bro04] Lisa M. Brown. View independent vehicle/person classification. In *VSSN '04: Proceedings of the ACM 2nd international workshop on Video surveillance and sensor networks*, pages 114–123, 2004.

[Bur11] H. Burkhardt. Reconstruction of a closed curve from its elliptic descriptor. 2011.

[CPFA09] Zezhi Chen, N. Pears, M. Freeman, and J. Austin. Road vehicle classification using support vector machines. In *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, volume 4, pages 214 –218, 2009.

[dB11] Alberto del Bimbo. Sift, surf gloh descriptors. 2011.

[DC10] Piotr Dalka and Andrzej Czyzewski. Vehicle classification based on soft computing algorithms. In Marcin Jensen, and Qinghua Hu, editors, *Rough Sets and Current Trends in Computing*, volume 6086 of *Lecture Notes in Computer Science*, pages 70–79. Springer Berlin Heidelberg, 2010.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, 2005.

[EVGW+07] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 results. 2007.

[Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006.

[FPWW04] Robert Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. Classification. 2004.

[GM07] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. J. Comput. Vision*, 73(1):41–59, 2007.

[HJS09] Hedi Harzallah, Frederic Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *International Conference on Computer Vision*, 2009.

[HR04] Peter Howarth and Stefan Rüger. Evaluation of texture features for content-based image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval, Springer-Verlag*, 2004.

[HU05] Joachim Hill and Thomas Udelhoven. Analyse von bilddaten und überwachte klassifikationsverfahren. 2005. Abteilung Fernerkundung, Universität Trier.

[Hut09] Tobias Hutzler. seam carving. Seminararbeit, Department Informatik, Fakultät Technik und Informatik, Hochschule für Angewandte Wissenschaften Hamburg, 2009.

[JÏ10] Bernd Jähne. *Digitale Bildverarbeitung*. Springer, 7 edition, 2010.

[KÖ5] Thomas Käster. *Intelligente Bildersuche durch den Einsatz inhaltsbasierter Techniken*. PhD thesis, Technischen Fakultät der Universität Bielefeld, 2005.

[KB12] Mehran Kafai and Bir Bhanu. Dynamic bayesian networks for vehicle classification in video. *IEEE Trans. Industrial Informatics*, pages 100–109, 2012.

[KJ13] Karsten Kozempel and Marek Jungshans. A system for video-based detection, classiffication and automatic storage of unusual and dangerous situations. 2013.

[Kle12] Ingo Klein. Datenanalyse nach w. stier. 2012.

[KR90] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 9th edition, 1990.

[KRM05] Uwe Knauer, Ralf Reulke, and Beate Meffert. Fahrzeugdetektion und -erkennung mittels mehrdimensionaler farbhistogrammanalyse. 2005.

[KS06] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. 2006.

[KS12] T. Senthil Kumar and S.N. Sivanandam. A modified approach for detecting car in video using feature extraction techniques. *European Journal of Scientific Research*, 77(1):134–144, 2012.

[KSPAT07] Farhad Mohamad Kazemi, Saeed Samadi, Hamid Reza Poorreza, and Mohamad-R. Akbarzadeh-T. Vehicle recognition based on fourier, wavelet and curvelet transforms - a comparative study. In *Information Technology, 2007. Fourth International Conference on*, pages 939 –940, 2007.

[Lan13] B. Lang. Vorlesung bildverarbeitung. 2013.

[Leu10] Vision GmbH Leutron. Picsight p141b-smart502 smart gige camera. July 2010.

[Loh12] Hans Lohninger. *Grundlagen der Statistik - Distanzmaße*. 2012.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[LZ13] Guojun Lu and Dengsheng Zhang. Shape based image retrieval using fourier descriptors. 2013.

[Mar03] Florian Markowetz. Klassifikation mit svm. 2003.

[Mat13a] MathWorks. Quadratische programmierung. 2013.

[Mat13b] MathWorks. Svmclassify in bioinformatics toolbox 3.3 (r2009a). 2013.

[MBG+11] H.K. Mebatsion, F. Boudon, C. Godin, C. Pradal, M. Genard, C. Goz-Bac, and N. Bertin. A novel profile based model for virtual representation of quasi-symmetric plant organs. *Computers and Electronics in Agriculture*, 75(1):113 – 124, 2011.

[ME03] D. Meyer-Ebrecht. Bildanalyse. 2003.

[MG05] Xiaoxu Ma and W.E.L. Grimson. Edge-based rich representation for vehicle classification. In *Computer Vision, 2005. Tenth IEEE International Conference on*, volume 2, pages 1185 –1192 Vol. 2, 2005.

[MKJ08] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph. A survey of shape feature extraction techniques. In *Pattern Recognition Techniques, Technology and Applications*, chapter 3. A Survey of Shape Feature Extraction Techniques. Peng-Yeng Yin, 2008.

[Mor08] Daniel Morlock. Vision based recognition of vehicle types. Master's thesis, Faculty of Computer Science, University of Karlsruhe, Germany und Carnegie Mellon University of Pittsburgh, USA, 2008.

[MP09] Jianwei Ma and Gerlind Plonka. A review of curvelets and recent applications. In *IEEE Signal Processing Magazine*, 2009.

[MXHZ10] Ling Mao, Mei Xie, Yi Huang, and Yuefei Zhang. Preceding vehicle detection using histograms of oriented gradients. In *Communications, Circuits and Systems (ICCCAS), 2010 International Conference on*, pages 354 –358, 2010.

[NA08] M.S. Nixon and A.S. Aguado. *Feature Extraction & Image Processing*. Academic Press, 2008.

[NT12] Jun Yee Ng and Yong Haur Tay. Image-based vehicle classification system. *CoRR*, abs/1204.2114, 2012.

[Pac07] Weston Pace. kmeans. 2007.

[Pan11] Robert Pansch, David und Wieczoreck. Seminar intelligent robotics: Objekterkennung. 2011.

[Par11] Fred Park. Shapedescriptor/feature extraction techniques. 2011.

[PH03] D.W.R. Paulus and J. Hornegger. *Applied Pattern Recognition: Algorithms and Implementation in C++*. Vieweg IT. Gwv-Vieweg, 4 edition, 2003.

[Poh06] Regina Pohle. Graphische datenverarbeitung und bildverarbeitung: Merkmale und klassifikation. 2006.

[Pol96] Robi Polikar. The wavelet tutorial part 1: Fundamental concepts and an overview of the wavelet theory. (Second Edition), 1996.

[RTL09] Payam Refaeilzadeh, Lei Tang, and Huan Liu. *Encyclopedia of Database Systems*, chapter Cross Validation. Springer, 2009.

[Say12] Saed Sayad. Support vector machine - classification. 2012.

[SBM02] Zehang Sun, G. Bebis, and R. Miller. Quantized wavelet features and support vector machines for on-road vehicle detection. In *7th International Conference on Control, Automation, Robotics and Vision.*, volume 3, pages 1641 – 1646 vol.3, 2002.

[Sch10] Rene Schwarz. Fourier analysis of a square wave. 2010.

[Smi97] Steven W. Smith. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, 1997.

[Ste93] R. Steinbrecher. *Bildverarbeitung in der Praxis*. Oldenbourg R. Verlag GmbH, 1993.

[Sti06] Stefan Stiene. Konturbasierte objekterkennung aus tiefenbildern eines 3d-laserscanners. Master's thesis, Universität Osnabrück, Fachbereich Informatik, 2006.

[SV99] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[Sze11] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 1 edition, 2011.

[TÖ8] Klaus-Dietz Tönnies. Template matching. 2008.

[TA03] D. Toth and T. Aach. Detection and recognition of moving objects using statistical motion detection and fourier descriptors. In *Image Analysis and Processing, 2003.Proceedings. 12th International Conference on*, pages 430 – 435, 2003.

[THM07] M.N. Tahir, A. Hussain, and M.M. Mustafa. Fourier descriptor for pedestrian shape recognition using support vector machine. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 636 –641, 2007.

[TK09] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, 4 edition, 2009.

[TTR01] Andre Thiang, Guntoro Teguh, and Lim Resmana. Type of vehicle recognition using template matching method. Electrical Engineering Department, Petra Christian University, 2001.

[USS01] K. Uchida, Y. Shirai, and N. Shimada. Probabilistic method of real-time person detection using color image sequences. In *International Conference*

on Intelligent Robots and Systems, 2001., volume 4, pages 1983–1988 vol.4, 2001.

[WRS+00] Andreas Wimmer, Georg S. Ruppert, Oliver Sidla, Harald Konrad, and Floris M. Gretzmacher. Fft-descriptors for shape recognition of military vehicles. pages 81–87, 2000.

[WYY+07] Xuezhi Wen, Huai Yuan, Chunyang Yang, Chunyan Song, Bobo Duan, and Hong Zhao. Improved haar wavelet feature extraction approaches for vehicle detection. In *Intelligent Transportation Systems Conference 2007. IEEE*, pages 1050 –1053, 2007.

[YNGR09] Raj Bahadur Yadav, Naveen K Nishchal, Arun K Gupta, and Vinod K Rastogi. Vehicular shape-based objects classification using fourierdescriptor technique. pages 484–495, 2009.

[ZAW07] Guohui Zhang, Ryan P. Avery, and Yinhai Wang. A video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. 2007.

[ZCC06] Chengcui Zhang, Xin Chen, and Wei-Bang Chen. A pca-based vehicle classification framework. In *22nd International Conference on Data Engineering Workshops, 2006.*, page 17, 2006.

[ZL04] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 1 2004.

[ZWZX11] Jiulong Zhang, Yinghui Wang, Zhiyu Zhang, and Chunli Xia. Comparison of wavelet, gabor and curvelet transform for face recognition. *Optica Applicata*, XLI(1), 2011.

Erklärung

Hierdurch erkläre ich, daß ich die von mir am heutigen Tage eingereichte Studienarbeit selbständig verfasst und andere als die angegebenen Hilfsmittel nicht benutzt habe.

Manuela Knaak
Dresden, 08. Mai 2013

Anhang