

ERMITTLUNG DER MITTLEREN REISEGESCHWINDIGKEIT MIT METHODEN DER DIGITALEN BILDVERARBEITUNG

Thomas Weber, Bakk. techn.

Betreuer und Begutachter:
Univ. Prof. DI Dr. Horst Bischof

**Masterarbeit am Institut für Maschinelles Sehen und Darstellen,
Fakultät für Informatik**



Technische Universität Graz
Erzherzog Johann Universität

April, 2007

Abstract

The aim of this work is to provide a method for automatic estimation of traffic parameters (like travel time for example) using digital image processing. The central problem in this field of research is the *reliable* recognition and (re-)identification of as many vehicles as possible. For the application mentioned, this is of prior significance compared to a *complete* identification of *all* vehicles. In this diploma thesis, an algorithm for colour object recognition is proposed, which will be applied in a real traffic analysis system. The novelty of the approach lies in the object pre-selection stage, where all object candidates are tested to estimate the richness of the colour structure; only from the residual candidates, features are extracted and finally matched. The selected (colour) features show to be - to some extent - insensitive to illumination changes. Eventually, matching is considered an association problem, which is solved by the well known stable marriage algorithm.

Extensive evaluation will examine the effectiveness especially of the preselection stage, while examining the sensitivity to changes of internal parameters.

Zusammenfassung

Das Ziel dieser Arbeit ist es, eine Methode zur automatischen Schätzung von Verkehrsparametern (wie z.B. der Reisezeit) mithilfe der digitalen Bildverarbeitung vorzustellen. Das zentrale Problem in diesem Forschungsgebiet ist die *zuverlässige* Erkennung und Identifizierung von so vielen Fahrzeugen als möglich. Dies hat in der vorliegenden Anwendung Vorrang vor einer *vollständigen* Identifizierung *aller* Fahrzeuge. In dieser Diplomarbeit wird ein Algorithmus zur Farb-Objekterkennung vorgestellt, der in einem realen System zur Verkehrsanalyse Anwendung findet. Der neue Ansatz ist eine Objektselektion, wo alle Objekte auf ihren Farbgehalt überprüft werden; nur von den erfolgreich getesteten Objekten werden Features extrahiert und gematcht; die verwendeten (Farb-) Features sind zu einem bestimmten Grad unempfindlich gegen Beleuchtungsänderungen. Der Matchingprozess wird als ein Zuordnungsproblem definiert und über den bekannten stable marriage Algorithmus gelöst. Eine ausführliche Evaluierung analysiert die Leistungsfähigkeit insbesondere der Objektselektion; daneben wird untersucht, wie kritisch die internen Parameter des Verfahrens auf Änderungen reagieren.

Danksagung

Für die große Geduld bis zur Fertigstellung dieser Arbeit sowie die wertvolle Unterstützung sowohl bei der schriftlichen Arbeit als auch der Implementierung, gilt mein Dank besonders meinem Betreuer Horst Bischof. Den Mitarbeitern der Firma Frequentis, insbesondere Andreas Kröpfl, möchte ich für die Möglichkeit danken, eine Diplomarbeit in diesem für mich sehr interessanten Gebiet verfassen zu können, sowie für die regen Fachdiskussionen rund um Computer Vision. Meinem Kollegen Clemens Arth, der mich während eines großen Teils der Implementierung des Gesamtsystems unterstützt hat, bin ich dankbar für die fruchtbare Zusammenarbeit und fortlaufende Motivierung, die die Implementierung sehr beschleunigt hat. Weiters möchte ich mich bei meinen Kollegen Christian Schindler und Martin Lechner bedanken, die vor allem in der frühen Projektphase stark am Aufbau des Environments und der Organisation mitgeholfen haben, und auch durch nicht fachbezogene Gespräche den „Büroalltag“ auf angenehme Weise mitgestaltet haben.

Für bleibendes Interesse, auch trotz Verzögerungen und Unterbrechungen bei der Fertigstellung, danke ich allen Freunden und Arbeitskollegen, die mir gezeigt haben, dass ein komplexes Thema wie Computer Vision auch bei Nicht-Fachkollegen Neugier zu wecken vermag.

An letzter und wichtigster Stelle gilt mein Dank meiner Familie, die mich immer unterstützt und an mich geglaubt hat: ohne sie wäre diese Arbeit nicht möglich gewesen. Vor allem Ihnen widme ich diese Arbeit !

Inhaltsverzeichnis

Einführung.....	9
1.1 Motivation.....	9
1.2 Lösung Verkehrstelematik ?.....	10
2 Problemstellung.....	14
2.1 Das Projekt InfoA10.....	14
2.2 Detailbeschreibung der Aufgabenstellung.....	16
2.3 Randbedingungen und Einschränkungen.....	18
2.4 Rahmen der Problemstellung.....	19
3 Literaturüberblick über Fahrzeugidentifizierung.....	20
3.1 Wiedererkennung von Fahrzeugen und mittlere Reisezeit.....	20
3.2 Abschließende Bemerkungen.....	40
4 Fahrzeugwiedererkennung und Matching.....	42
4.1 Allgemeines.....	42
4.2 Struktur des Verfahrens.....	42
4.3 Objektselektion.....	43
4.4 Die Featuregenerierung.....	59
4.5 Feature- und Objektmatching.....	66
4.6 Zusammenfassung.....	71
5 Evaluierung.....	72
5.1 Allgemeines.....	72
5.2 Wesentliche Parameter des Verfahrens.....	76
5.3 Einführendes Beispiel zur Illustration.....	77
5.4 Untersuchung der Featureselektions-Stufe.....	81
5.5 Untersuchung der Featuregenerierungs-Stufe.....	89
5.6 Untersuchung der Matching Stufe.....	96
5.7 Zusammenfassung der Ergebnisse für die Datensets.....	103
5.8 Ermittelte Durchschnittsgeschwindigkeiten aus den Matches.....	105
5.9 Zusammenfassung.....	107
6 Fazit und Ausblick.....	108
6.1 Zusammenfassung.....	108
6.2 Ausblick.....	108
6.3 Fazit.....	111
7 Anhang.....	112
7.1 Implementierungsbeschreibung.....	112
7.2 Eingesetzte Tools.....	119
7.3 Nachtrag: Fallbeispiel aus der Evaluierung.....	121
8 Literaturverzeichnis.....	124

Abbildungsverzeichnis

Fahrbahnabschnitt.....	15
Aufgabenstellung.....	16
Signatur.....	21
3D Modelfitting.....	26
Autocolor Objektdetektion.....	28
Mehrstufiges Matching.....	30
Farbkategorien.....	31
Sequence Matching.....	32
Signatur und Farbe als Features.....	33
Temporal Fusion.....	35
Graphenstruktur von Platoons.....	36
Matching Strategie (Überblick).....	38
Einzelfeatures bei probabilistischem Matching.....	39
Beispielbild zur Segmentierung.....	49
Featurespace des Beispielbildes.....	49
Cluster des Beispielbildes.....	50
Essentielle Moden (1).....	54
Essentielle Moden (2).....	54
Gitterverfahren Übersicht.....	58
Gitterverfahren Detail.....	58
kd – Baum Struktur.....	58
Zerlegung eines Bildes in Patches.....	61
Signatur eines Bildes.....	64
Color Constancy.....	65
Features im Vergleich.....	68
Fahrzeug-Beispiele aus Set1.....	73
Fahrzeug-Beispiele aus Set2.....	73
Fahrzeug-Beispiele aus Set3.....	74
Exemplarische Bildmenge 1.....	78
Exemplarische Bildmenge 2.....	78
Selektion aus Bildmenge 1.....	79
Selektion aus Bildmenge 2.....	79
Ergebnis des Matchingverfahrens.....	81
Schwieriges Matchingproblem (PKW).....	81
Verlauf der Erkennungsrate in Abhängigkeit von Mindestscore und Mindestgröße (relativ)	83/84
Verlauf der Erkennungsrate in Abhängigkeit des Mindestscore (absolut).....	84
Selektion mit Bildgröße als Kriterium.....	85
Selektion mit Bildgröße und Mindestscore als Kriterium.....	86
Verlauf der Erkennungsrate in Abhängigkeit von Radius und Buntheit.....	87/88
Texturgrößen.....	90
Effekt von unterschiedlichen Pruningparametern.....	92
Verlauf der Erkennungsrate in Abhängigkeit vom Pruning.....	93/94
Analyse der Beleuchtungsinvarianz.....	95/96
Analyse des Matching-Thresholds und des Einflusses externer Parameter.....	98/99

Analyse der Verifikation.....	101
Beispiele zu Verifikation bzw. false positives.....	102
Ermittelte Durchschnittsgeschwindigkeiten.....	106
Software Architektur.....	112
Region of Interest als Polygonzug.....	116
Approximiertes Koordinatensystem.....	117
Diskretisierung der Straengeometrie.....	117
Scanline.....	118
Blobgrowing.....	118
Beispielfahrzeuge aus Set 1 fr Fallstudie.....	121
Beispielfahrzeuge aus Set 2 fr Fallstudie.....	121

Tabellenverzeichnis

Tabelle 1: Abhängigkeit der Matchingergebnisse von der Kachelungsgröße bei Set 1.....	89
Tabelle 2: Abhängigkeit der Matchingergebnisse von der Kachelungsgröße bei Set 2.	90
Tabelle 3: Abhängigkeit der Matchingergebnisse von der Kachelungsgröße bei Set 3.....	90
Tabelle 4: Verteilung der Fahrzeugtypen bei den Sets 1-3.....	98
Tabelle 5: Ergebnisse für das Matching von Set 1.....	104
Tabelle 6: Ergebnisse für das Matching von Set 2.....	104
Tabelle 7: Ergebnisse für das Matching von Set 3.....	104
Tabelle 8: Verwendeten Parameter für die Ergebnisse aus Tabelle 1-3.....	105
Tabelle 9: Eingesetzte Software Tools.....	120
Tabelle 10: Fallbeispiel: Score-Matrizen nach dem Matching.....	122

Einführung

1.1 Motivation

Das Phänomen Verkehr am Beispiel Europa

Der Wunsch nach individueller Mobilität, einhergehend mit einer Steigerung des allgemeinen Wohlstandes, schlägt sich in beeindruckenden Zahlen nieder [1]:

- die Zahl der Fahrzeuge weltweit ist von 1950-1990 von 75 auf 675 Millionen angestiegen
- das Autobahnnetz der EU-15 Staaten hat sich in den Jahren 1990-1999 um 25% auf insgesamt knapp 50000 km gesteigert
- der EU-15 Durchschnitt für das Jahr 2000 belief sich auf 469 Fahrzeuge je 1000 Einwohner

Offensichtlich ist der Ausbau des Straßenverkehrs (und hier vor allem der Hochgeschwindigkeitsstrecken) bisher stark gegenüber alternativen Transportmitteln wie der Bahn vorangetrieben worden [2]. Leider sind die Lebensqualität des Menschen, sowie die Umwelt im Allgemeinen, davon unmittelbar betroffen. Staus sind beinahe unweigerlich die Folge dieser Verkehrsbewegungen, und Staus bedeuten Kosten - täglich über 100 Millionen Euro laut einer Studie der EU Kommission [3]. Verkehrsmanagement ist daher verständlicherweise zu einem wichtigen Anliegen unter praktisch allen Staaten mit entsprechender technischer Infrastruktur geworden.

Das EU-Weißbuch für Verkehr erklärt, dass durch den Einsatz von Verkehrstelematik bzw. ITS (**Intelligent Transport Systems**) im allgemeinen eine Reduzierung der Reisezeit um bis zu 20%, eine Verbesserung der Verkehrsnetzauslastung um 5-10% sowie eine

Verminderung bei gewissen Unfallarten (Auffahrunfall) um 10-15% erreicht wird. Im allgemeinen ist eine Erhöhung der Überlebensrate bei Unfällen aller Art festzustellen, die durch die Entwicklung vor allem im Bereich der automatischen Detektion von ungewöhnlichen Vorfällen und des Managements von Notfallszenarien begründet ist [4].

1.2 Lösung Verkehrstelematik ?

Um einen möglichst umfassenden Einblick in das Verkehrsgeschehen zu erhalten, ist eine möglichst vollständige Überwachung der kritischen Zonen erforderlich. Für ein optimales Zusammenschalten der vielen, zum Teil sehr unterschiedlichen, Informationen und deren Auswertung, ist ein modernes Datenverarbeitungssystem notwendig [5].

Dieser Einsatz von Methoden der Informationsverarbeitung im Verkehrswesen wird mit **Verkehrs-Telematik** bezeichnet.

Dazu werden etwa Leit- und Bordrechner eingesetzt, die heterogene Peripherie (Sensoren wie Induktionsschleifen, Infrarotsensoren, Videokameras, ...) zur Beurteilung der Verkehrssituation und ggf. zur Einleitung von Maßnahmen zur Gegensteuerung verwenden (z.B. Aktoren wie Licht- und Signalanlagen). Auch die Ausnutzung von globalen, satellitengestützten Ortungssystemen wie GPS, modernen Kommunikationsnetzen (Mobilfunk und Corporate Networks) oder intelligenten Verkehrsinformationssystemen im Allgemeinen gehören heute zu den immer häufiger eingesetzten Mitteln zur Beherrschung der Verkehrsproblematik. Technologien wie „Drive-by-Wire“ oder Begriffe wie „Floating Car Data“ belegen den Trend zum Einsatz von Hi Tech in der Verkehrsdomäne.

Seit einigen Jahren nützt man nun zunehmend auch die Methoden der **digitalen Bildverarbeitung**.

Ausgangspunkt der Verarbeitung sind dabei die Bildsequenzen einer Videokamera, die üblicherweise überkopf montiert wird (auf „Gantries“, also Überbauten über der Fahrbahn), bzw. in Tunnels auch an der Tunneldecke.

Die Bildanalyse stützt sich dabei auf Aufnahmen des menschlich sichtbaren Spektrums des Lichtes (für Tageslichtsituationen) oder des Infrarotbereiches (für Nachtsituationen), die für die automatisierte Weiterverarbeitung notwendige Digitalisierung der Daten kann (lokal) in der Kamera selbst oder (remote) am Analyserechner erfolgen.

Entscheidende Vorteile gegenüber vergleichbaren Sensoren sind:

- der hohe Informationsgehalt visueller Daten (andere Sensoren liefern oft nur eindimensionale Daten, sind also Punktsensoren oder -detektoren)
- die kostengünstige und unkomplizierte Datenerfassung
- die Tatsache dass die Natur der Daten parallele Kontrolle und Analyse durch den Menschen praktisch ohne zusätzliche Aufbereitung möglich macht.

Dadurch werden vielfältige Methoden der Verarbeitung eröffnet. Der Einsatzbereich reicht von informativen Diensten wie Stau Erkennung über die Detektion ungewöhnlicher Ereignisse (z.B.: Geisterfahrer) bis zur automatisierten Geschwindigkeitskontrolle (Section Control) [6].

Überblick über aktuelle Systeme im Einsatz geben z.B. das Rosetta [7] Projekt für die EU. Folgende Beispiele daraus seien hier angeführt:

- Das Verkehrsüberwachungssystem in Finnland (über die Finnish National Road Administration FINRA), wo vor allem die Wetterbedingungen auf Überlandstraßen das Hauptinteresse darstellen. Dazu werden u.a. 180 Videokameras eingesetzt.

- Vor allem in Italien, wo alle Autobahnen unter Gebühr stehen, wird stark auf Videounterstützung gesetzt (1334 Kameras sind im Einsatz). Hier werden Kameras und Bildverarbeitung dazu verwendet, Anomalien im Verkehr zu entdecken und dem verantwortlichen Mitarbeiter in der Überwachungszentrale zu melden. Fahrzeuge werden beim Eintreten in das und beim Verlassen des Autobahnnetzes kontrolliert, womit auch eine recht gute Abschätzung des Verkehrsflusses in Echtzeit möglich ist. Diese Videosysteme werden ebenfalls in Großstädten eingesetzt (z.B. Rom und Turin).
- In den Niederlanden wird nach einer langen Tradition des Einsatzes von Schleifendetektoren nun zunehmend Videoverarbeitung eingesetzt. Dies dient vor allem der der Geschwindigkeitskontrolle durch Aufnahme und Speicherung der Fahrzeugkennzeichen auf jeweils zwei Stationen, die einige Kilometer voneinander entfernt sind. Somit ist also keine punktuelle Geschwindigkeitsübertretung, sondern eine erhöhte Durchschnittsgeschwindigkeit Ziel der Messung.
- Österreich verwendet ein ähnliches System auf der Autobahn unter dem Namen „Section Control“, das ebenfalls auf Kennzeichen Identifizierung zurückgreift.

Projekte wie IN_RESPONSE [7](Monitoring von Straßen und Autobahnen, vor allem zur automatische Fahrzeugidentifikation und Unfalldetektion mittels Bildverarbeitung), CLEOPATRA [8] (Reisezeitinformationen und -leitung, Messung von Fahrzeitverzögerungen und Verkehrsflüssen) sind zwei schon etwas ältere Beispiele für Projekte in größerem (d.h. EU-weitem) Rahmen.

Die Hauptmotive für diese und andere Projekte sind im wesentlichen:

- Einhaltung der Straßenverkehrsordnung („law enforcement“), z.B. durch automatische Geschwindigkeitskontrollen

- Erhebung langfristiger statistischer Daten (Schätzung der Umweltbelastung, Fahrbahnabnutzung, etc.)
- Optimierung von Verkehrsbewegungen durch Stau- und Reisezeitinformationen

Die vorliegende Arbeit beschäftigt sich mit der letztgenannten Problematik, indem eine Methode zur Schätzung der Reisezeit vorgestellt wird. Der Aufbau ist wie folgt: Im nächsten Kapitel wird das zugrunde liegende Projekt InfoA10 beschrieben und daraus die Problemstellung herausgearbeitet. Im Kapitel 3 wird dann ein Überblick über aktuelle Veröffentlichungen bzw. bestehende Lösungen im Bereich der Problemstellung gegeben. Kapitel 4 beschreibt den eigenen Ansatz zur Lösung des Problems. Eine anschließende Evaluierung in Kapitel 5 analysiert das Verhalten des Systems im Einsatz mit den erhobenen Daten. Die Arbeit schließt mit einem Ausblick und Überlegungen über mögliche Erweiterungen. Im Anhang wird außerdem noch das implementierte System zur Datenerhebung beschrieben.

2 Problemstellung

In diesem Kapitel wird die zu lösende Aufgabenstellung im Detail besprochen, und die Randbedingungen werden definiert.

2.1 Das Projekt InfoA10

Ziel des Projektes ist der Aufbau eines Pilotsystems zur automatischen Erzeugung von Verkehrsinformationen. Dazu werden zwei Videokameras eingesetzt, die am Anfangs- und Endpunkt eines Baustellenprojekt auf einer österreichischen Autobahn montiert sind (A10 Tauern-Autobahn, s. Abbildung 1 für den Streckenplan). Da es aufgrund des bekannt starken Verkehrsaufkommens der Strecke durch die Baustelle (vor allem in der typischen Urlaubszeit, da die Strecke eine Nord-Süd Achse zwischen Deutschland und Italien darstellt) zu erwarteten Verzögerungen und Staus kommen wird, ist eine Überwachung und Kontrolle des Verkehrs erwünscht [9].

Die folgenden Firmen arbeiten am Gesamtsystem mit:

- **FREQUENTIS**, die die Aufgabe der Videoverarbeitung und der Generierung von Verkehrsinformationen übernimmt
- **Mobilkom Austria**, die die Verkehrsinformationen zur Bereitstellung von Services an Endkunden anbietet
- **Arsenal Research**, die eine Fahrzeugzählung über Lasersensoren implementiert
- **Team**, welches die Datenbank zur Archivierung der erhobenen Daten und Protokolle zur Übertragung der Verkehrsdaten implementiert.

In enger Kooperation befinden sich die Autobahnen- und Schnellstraßen-Finanzierungs-Aktiengesellschaft (**ASFINAG**), die einen Teil der Infrastruktur zur Verfügung stellt (Kameras und Anbindung an das

interne Corporate Network), sowie der **ATTAC** (Austrian Traffic Telematics Cluster).

Eine Aufgabe dieses Projektes, die in dieser Arbeit vorgestellt wird, ist die Schätzung der streckenbezogenen Reisezeit (*“link travel time”*). Damit ist die Zeitdauer gemeint, die ein Fahrzeug für das Zurücklegen einer definierten Strecke (den *„link“*) benötigt; die Strecke ist hier definiert als der Abschnitt zwischen den Kameras, also das Baustellengebiet. Die *link travel time* ist einer der wichtigsten Parameter aus der Verkehrstelematik, und wird in erster Linie zur Verkehrsflussanalyse herangezogen; dazu werden meist die Durchschnittszeiten mehrerer Fahrzeuge auf derselben Strecke gemittelt (*„mean (link) travel time“*). Dies soll Fahrern, die diese Strecke benutzen wollen, Hinweise auf die zu erwartende Verzögerung geben.



Abbildung 1: Der Fahrbahnabschnitt der Tauernautobahn, an dem die Kameras montiert sind (© ASFINAG)

2.2 Detailbeschreibung der Aufgabenstellung

Datenerhebung und Infrastruktur

Als einzige optische Sensoren sind zwei Videokameras im Einsatz, die in einem Abstand von ca. 6 km voneinander und ca. 5m über der Fahrbahn fest montiert sind. Beide Kameras sind mit Blickrichtung in Fahrtrichtung des Verkehrs ausgerichtet, überwachen also dieselbe Fahrtrichtung. Abbildung 2 veranschaulicht das Grundprinzip:

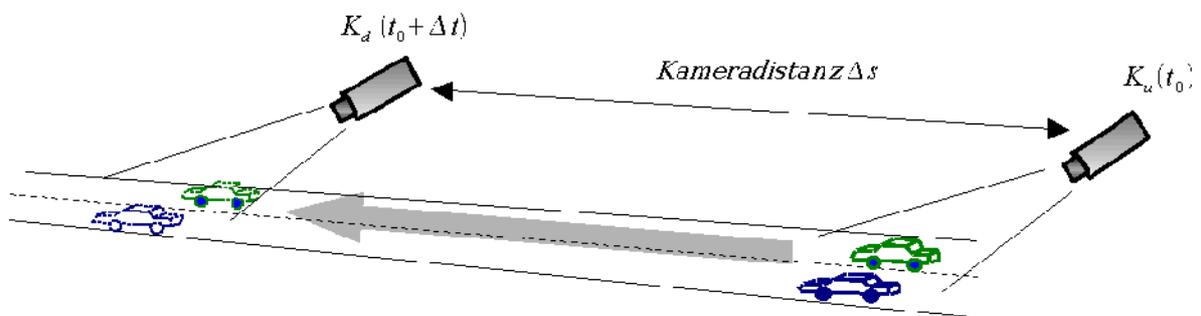


Abbildung 2: Aufstellung der Videokameras. Kamera K_a ist stromabwärts, Kamera K_u stromaufwärts in einer Distanz von Δs montiert. Eine Szene zum Zeitpunkt t_0 auf K_u wird zum Zeitpunkt $t_0 + \Delta t$ auf K_a beobachtet

Die Videoverarbeitung wird auf Ausgangsdaten der bereits in Betrieb befindlichen Kameras durchgeführt. Eine weitere Datenerhebung erfolgt parallel dazu mit herkömmlichen Punktdetektoren (Lasersensoren), wobei letztendlich eine Validierung/Verifizierung der Verarbeitungsergebnisse der unterschiedlichen Sensordaten gegeneinander geplant ist.

Die ursprünglich analogen Videodaten der Kameras werden digitalisiert und in einen MJPEG Stream kodiert, dann über das Corporate Network der ASFINAG an eine Workstation geschickt, wo die eigentliche Verarbeitung und Auswertung stattfindet. Die gewonnene Information wird als Nutzdaten an einen Server übertragen, dort in einer Datenbank gespeichert, und soll in weiterer Folge für Endkunden als

Verkehrsinformation angeboten werden. Die Daten (aktuelle streckenbezogene Verkehrsinformation) sollen später als ein Service über Mobiltelefone abgerufen werden können.

Berechnung der mittleren Reisezeit

Für die folgenden Überlegungen wird diejenige Kamera, die „stromaufwärts“ montiert ist, K_u (für **upstream**) genannt, und die weiter „stromabwärts“ montierte Kamera K_d (für **downstream**). Die Distanz d zwischen K_u und K_d ist konstant und bekannt. Wir definieren die mittlere Reisezeit für ein Fahrzeug auf einem Fahrbahnabschnitt als den Quotienten aus durchfahrener Distanz und dafür benötigter Zeit. Bei bekannter Distanz bleibt somit noch die Ermittlung der benötigten Zeit. Dazu müssen kontinuierlich die Bilder beider Kameras analysiert werden. Es ist folglich das Ziel, eine Situation S (~ ein oder mehrere Fahrzeuge) die zum Zeitpunkt t_0 auf K_u beobachtet wird, zu einem späteren Zeitpunkt $t_1 = t_0 + \Delta t$ auf K_d wieder zu erkennen. Damit ist die für das Durchfahren der Distanz Δs benötigte Zeit Δt bekannt

und es kann die Geschwindigkeit $v = \frac{\Delta s}{\Delta t}$ berechnet werden. Für die konkrete Problemstellung ist es gefordert, laufend die aktuelle mittlere Geschwindigkeit abzuschätzen. Daher wird die mittlere Reisegeschwindigkeit *des Streckenabschnittes* definiert als der gleitende Mittelwert über die letzten M ermittelten Einzelgeschwindigkeiten:

$$\bar{v}(t_i) = \frac{1}{M} \times \sum_{j=0}^{M-1} t_{i-j}$$

wobei für die ersten Schätzungen für $j < M - 1$ die Einzelgeschwindigkeiten selbst oder auch die gleitenden Mittelwerte mit entsprechender Mittelperiode j angenommen werden können.

2.3 Randbedingungen und Einschränkungen

Um die Aufgabenstellung präziser zu fassen bzw. zu vereinfachen, werden gewisse Randbedingungen definiert und Prioritäten vereinbart:

1. Es ist nicht erforderlich, alle Fahrzeuge in der Bildverarbeitung zu berücksichtigen, die an einer Kamera vorbeifahren. Der Output soll dennoch angemessen dynamisch die wesentlichen Verkehrsänderungen wiedergeben. Dafür sollte für die tatsächlich betrachteten Fahrzeuge eine höhere Erkennungssicherheit erreicht werden [10].
2. Es wird nur eine Fahrtrichtung betrachtet, die jedoch aus bis zu 3 Spuren bestehen kann. Die gleichzeitige Erfassung von Verkehrs- und Gegenverkehrsspur würde lediglich Komplexität in die Software einführen, die durch Betrieb je eines Systems für jede Fahrtrichtung vermieden werden kann.
3. Die Bildverarbeitung arbeitet nur unter durchschnittlichen Tageslichtbedingungen. Teile der verwendeten Algorithmen verwenden Farbinformation, und sind bei Dunkelheit sehr unzuverlässig. Das bedeutet in weiterer Folge, dass für extreme Lichtverhältnisse (wie starke Spiegelungen oder dichter Nebel), keine korrekten Ergebnisse zu erwarten sind. Für diese Verhältnisse sind spezielle, aufwändigere Verfahren erforderlich, die nicht Teil dieser Implementierung sind.
4. Robustheit gegen Änderungen an der Kameraausrichtung ist keine Anforderung. Falls die Kamera durch mechanische äußere Einwirkung bewegt wird, ist es nicht erforderlich, dass das implementierte System sich dieser Änderung anpasst (Bildstabilisierung), und weiterhin korrekt arbeitet. Durch eine Änderung der Kameraausrichtung wird die relative Position der Straße im Videobild geändert. Diese wird aber

beim Konfigurieren des Systems fest vorgegeben, daher ist das Funktionieren der Gesamtapplikation von dieser Vorgabe abhängig.

2.4 Rahmen der Problemstellung

Im wesentlichen lässt sich die Problemstellung der Geschwindigkeitsermittlung in 2 Grundprobleme aufteilen. Aus diesen ergeben sich - je nach gewähltem Lösungsansatz - eine Reihe von Teilproblemen, die in Kapitel 3 diskutiert werden. Folgende 2 Grundprobleme liegen vor:

1. die Fahrzeugdetektion (*also Objektdetektion*), und
2. die *Wiedererkennung* detektierter Fahrzeuge.

Die Fahrzeugdetektion wird üblicherweise meist über unspezifische Verfahren wie Motion Detection, Moving Object Detection (MVOD) oder Feature Point Detection, oder aber spezifischere Verfahren wie Appearance Based Recognition oder Model Based Recognition durchgeführt. Die Unterteilung in spezifische und unspezifische Verfahren soll ausdrücken, ob grundsätzlich bewegte Objekte, oder speziell Fahrzeuge mit bestimmten Erscheinungsformen detektiert werden (z.B. **Klassifikation**).

Die ersten beiden Verfahren sind dabei die häufigsten, weil sie bei einfacher Implementierung in vielen Fällen akzeptable Ergebnisse liefern. Zur Datenerfassung für die vorliegende Arbeit wurde ein System entwickelt, das mit einer MVOD arbeitet (s. Anhang für eine Implementierungsbeschreibung).

Der Fokus dieser Arbeit liegt aber klar auf dem zweiten Gebiet, der Wiedererkennung; daraus wird das Hauptverfahren in Kapitel 4 entwickelt.

3 Literaturüberblick über Fahrzeugidentifizierung

Im folgenden wird ein Überblick gegeben, welche Arbeiten über das zentrale Thema dieser Arbeit, die Fahrzeugidentifizierung existieren. Die Möglichkeiten reichen dabei von einem einfachen Detektorpaar an 2 hinreichend weit entfernten Strecken-Endpunkten bis zu einem „Multisensor Fusion Network“ [11][12] mit vielen Stationen, wo Sensoren für unterschiedlichste physikalische Größen miteinander verknüpft werden.

3.1 Wiedererkennung von Fahrzeugen und mittlere Reisezeit

Allgemeines

Es wird hier ganz grob unterschieden zwischen optischen und nicht optischen Methoden. Im Detail werden nur die optischen Methoden besprochen; aus den nicht-optischen Methoden stammt aber das wichtige Prinzip der Signatur, das kurz angesprochen wird.

Nicht-optische und Hybride Methoden

Die große Mehrheit der bestehenden Systeme ist mit Punktdetektoren ausgestattet. Dazu zählen in erster Linie Laser- und Schleifendetektoren nach dem Induktionsprinzip. Aus dem zeitlichen Signalverlauf der Messgröße werden eindeutige Merkmale berechnet (die *Signatur* eines Fahrzeugs bzw. Fahrzeugtyps, s. Abbildung 3) und anschließend gematcht.

Schließlich gibt es auch Systeme, die mit *Tags* (Identifikationsträgern) arbeiten, die in den Fahrzeugen mitgeführt werden müssen (z.B. auf RFID Basis oder über Mobiltelefone.) Nachteil ist aber die Beschränkung der Erkennung auf entsprechend ausgestattete Fahrzeuge.

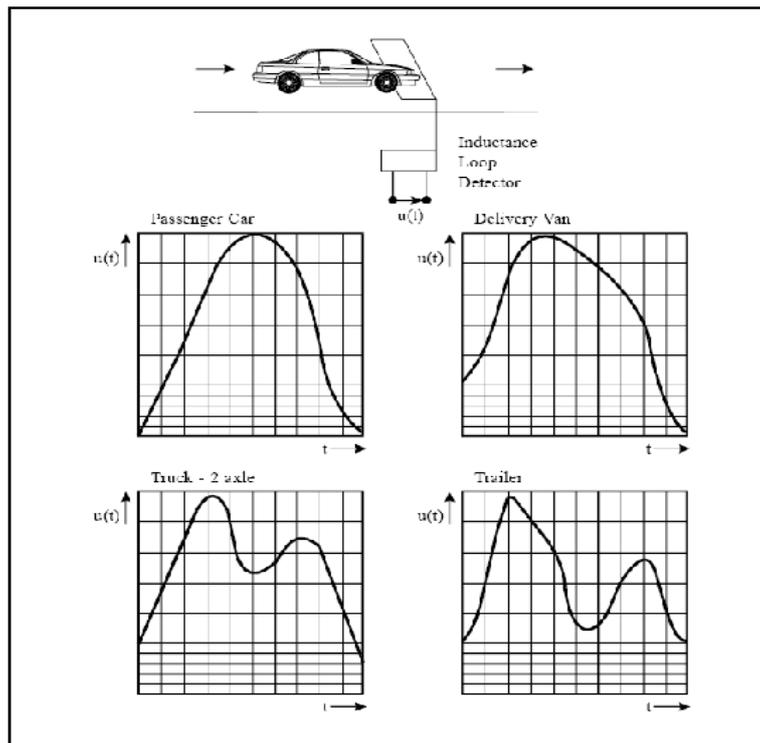


Abbildung 3: Signaturen verschiedener Fahrzeugtypen eines Schleifendetektors [14]

Neben den zuvor beschriebenen Methoden, gibt es zunehmend auch Ansätze, die Sensoren unterschiedlicher physikalischer Messgrößen gleichzeitig einsetzen, sodass ihre jeweiligen Schwächen teilweise kompensiert werden können [13]. Neben Kombinationen von Laser, induktiven Sensoren und Videokameras werden z.B. auch akustische Signale mit Mikrofonen aufgezeichnet, von denen ebenfalls (wie bei induktiven Aufnehmern) eine Signatur gewonnen werden kann.

Übersicht über Methoden mithilfe der Bildverarbeitung

Die folgenden optischen Methoden basieren auf der Nachbildung eines zuvor besprochenen Schleifendetektors mit Kameras anstelle der induktiven Aufnehmer, der so genannten **optischen Schleife**. Werden

mehrere Schleifen zur Ermittlung streckenbezogener Verkehrsdaten herangezogen, spricht man von der **erweiterten optischen Schleife**; auf dieser basieren alle hier besprochenen Verfahren.

Fast alle Verfahren stammen aus den folgenden zwei Gebieten:

1. Erkennung von Nummerntafeln
2. Feature Based Identification („Merkmalsbasierte Identifizierung“)

Erkennung von Nummerntafeln

Eine häufig angewandte [15] und nahe liegende Methode basiert auf der Erkennung und dem Vergleich von Nummerntafeln; die Nummern darauf stellen ein eindeutiges Merkmal für jedes Fahrzeug dar, das noch dazu in einer vorgeschriebenen Norm kodiert ist. Diese Eindeutigkeit macht den Matching Prozess selbst trivial.

Die Aufgabenstellung besteht im Wesentlichen aus 1. Lokalisation der Nummerntafel am Fahrzeug 2. (geometrisches) Entzerren und Transformieren in eine normalisierte Darstellung und 3. Erkennung der Buchstabencodes und Speicherung als ASCII Zeichen. Dieser Schritt kann auch durch direktes Matching der Bilder der Nummerntafeln (z.B. über Template Matching) erfolgen. Die Information der Nummerntafel kann (als Ziffernfolge) sehr effizient gespeichert werden.

Wichtige Voraussetzung ist die gute Lesbarkeit von Nummerntafeln. Probleme bereiten dabei der notwendige flache Aufnahmewinkel, da dann die Häufigkeit von Verdeckungen durch andere Fahrzeuge sehr groß ist, rechtliche Gründe (vor allem der Datenschutz) und die (zu) geringe Auflösung der herkömmlichen Überwachungskameras. Wenn ausschließlich die Nummerntafeln als Features extrahiert werden, sind

diese Verfahren sehr stark von Verdeckungen (vor allem bei Verkehr „Stoßstange an Stoßstange“) betroffen [16][17][18][19].

Feature based Verfahren

Allgemeines

Bei diesen Verfahren werden die Fahrzeuge durch eine Anzahl repräsentativer Merkmale beschrieben. Dazu werden n Einzelmerkmale aus dem Fahrzeugbild extrahiert und in Form eines Merkmalsvektors gespeichert. Dieser Vektor hat n Koordinaten, wobei jeder Koordinate ein Merkmal zugeordnet wird. Ein bestimmter Merkmalsvektor

$V = (v_1, v_2, v_3, \dots, v_n)^T$ wird als ein Punkt mit diesen Koordinaten im n -dimensionalen Raum interpretiert. Die **Featurespace Analysis** untersucht Anordnung und Lage der Merkmale in dem Raum, der durch die Merkmalsvektoren aufgespannt wird. Ziel der Featurespace Analysis ist es, eine üblicherweise große Anzahl an Merkmalsvektoren durch wenige „repräsentative“ Vektoren möglichst gut zu beschreiben, sodass Speichern und Matchen dieser Darstellung effizient sind.

Diejenigen Regionen im Raum, die über eine hohe Dichte an Merkmalen verfügen, werden Cluster oder Moden genannt. Das Finden dicht besiedelter Regionen (repräsentiert durch ihre Zentren) im Featurespace wird in der Statistik „Multivariate Location Problem“ genannt. Es wird angenommen, dass die in diesem Raum abgebildeten Datenpunkte eine Dichtefunktion einer Verteilung (Probability Density Function oder **PDF**) darstellen. Dadurch wird impliziert, dass die Datenpunkte im Featurespace Stichproben der wahren PDF sind, und für diese somit nur ein Schätzwert gefunden werden kann. Das Ergebnis ist eine Klassifizierung des Featurespace in der Art, dass jeder Datenpunkt der Verteilung einem der Moden der zugrunde liegenden PDF zugeordnet wird; die Menge aller zu einem Mode (oder „Cluster“) gehörenden Datenpunkte wird dessen Support genannt.

Werden zwei Merkmalsvektoren gematcht, wird das Ergebnis meist als (normierter) Score quantitativ erfasst. Damit wird prinzipiell eine Hypothese über eine Objektkorrespondenz aufgestellt. Durch zusätzliche Constraints wie z.B. temporale Randbedingungen (meist eine Bedingung für die erwartete Ankunftszeit von Fahrzeugen auf den Sensoren) kann diese Hypothese verifiziert werden.

Die in den Veröffentlichungen beschriebenen primären (=nur aus dem Fahrzeugbild gewonnenen) *und* sekundären (=abgeleiteten, spatio-temporalen oder aus dem Kontext gewonnenen) Merkmalen sind meist:

- Farbe bzw. Farbstruktur
- Größe bzw. Dimension
- Form, Fahrzeugtyp
- Momentangeschwindigkeit
- Spurverhalten etc.

Farbmerkmale

Als häufigstes Merkmal bei Fahrzeugen wird **Farbe** verwendet; Grund ist u.a. der hohe Informationsgehalt. Verschiedene Fahrzeuge desselben Fabrikats können immer noch durch unterschiedliche Farbe unterschieden werden.

Die Farbe eines Objektes kann auf verschiedene Art repräsentiert werden. Die einfachste Form ist die Speicherung des **dominanten Mode**, also des Modes dem die größte Anzahl an Pixeln zugeordnet wird.

Diesen Weg geht [20], wo es primär um Überwachungsaufgaben und Absicherung gegen Anschläge statt um zivile Verkehrsanalyse geht. Die Merkmale sind Farbe, Größe und geschätzter Ankunftszeit des Fahrzeugs auf der stromabwärts gelegenen Kamera. Vor allem ist hier ein

Selektionsschritt bemerkenswert, der die Eignung der Fahrzeuge für die Extraktion *guter* Merkmale schätzt und nur die entsprechend bewerteten Fahrzeugkandidaten weiterverwendet. Die Autoren begründen diesen Schritt mit der Beobachtung, dass große und auffallend bunte Fahrzeuge selten vorkommen, und somit die Wahrscheinlichkeit eines zuverlässigen Matches größer ist. (Dieser selektiven Ansatz wird auch im eigenen Verfahren im Kapitel 4 angewendet, allerdings werden aufwendigere Farb-Features berechnet.) Die Farbstruktur selbst wird über k-means Clustering analysiert, und der dominante Mode (= derjenige mit dem größten Support) wird als Merkmal gespeichert. Der Support wird auch zur Schätzung der Fahrzeuggröße herangezogen. Es werden nur Fahrzeuge akzeptiert, deren Farb-Sättigung und Größe über bestimmten Schwellwerten liegt. Für das Matching wird der gewichtete Abstand zwischen den Features bewertet, außerdem werden stromabwärts nur Kandidaten innerhalb eines Zeitfensters (geschätzte Ankunftszeit mit Konfidenzintervallen) betrachtet.

Woesler beschreibt in [21] die Farbe eines Fahrzeuges durch den Median der Farbwerte aller Pixel, vor allem um Effekte wie spiegelnden Reflexionen und Schatten abzuschwächen. Laut den Autoren soll das extrahierte Merkmal dem dominanten monochromen Farbeindruck entsprechen; der Median soll also auch nicht-relevante Farben im Unterbau und den Fensterscheiben des Fahrzeugs unterdrücken. Die Arbeit beschäftigt sich in weiterer Folge primär mit dem Verfolgen von Fahrzeugen über mehrere (> 2) Kameras mittels Multiple Hypothesis Tracking (MHT).

Eine Kombination aus Model based Recognition und Farbmatching wird in [22] angewendet. Zuerst werden die Kanten aus dem über Moving Object Detection lokalisierten Fahrzeug extrahiert, dann verschiedene 3D Modelle in die Szene (über das Fahrzeug) projiziert und gefittet. Aus einem genau spezifizierten, hinteren Teil des Fahrzeugdachs wird die

Farbe entnommen und in den HSV Raum transformiert; laut den Autoren ist hier vor allem der Hue-Wert (der Farbton) invariant gegen Helligkeitsänderungen. Aus dem Modell wird die Länge des Fahrzeugs als Merkmal extrahiert, ebenso wie erwartete Fahrzeit und Farbton(Hue), Sättigung(Saturation) und Helligkeit (Brightness) des vom Fahrzeugdach ermittelten Farbwertes. In einer stochastischen Fehlerfunktion werden die Differenzen von Farbe (H,S und V Komponente), Fahrzeuglänge und Ankunftszeit verknüpft mit (empirisch ermittelten) Standardabweichungen der Parameter sowie Gewichtungsfaktoren, die die Relevanz für den Score widerspiegeln.

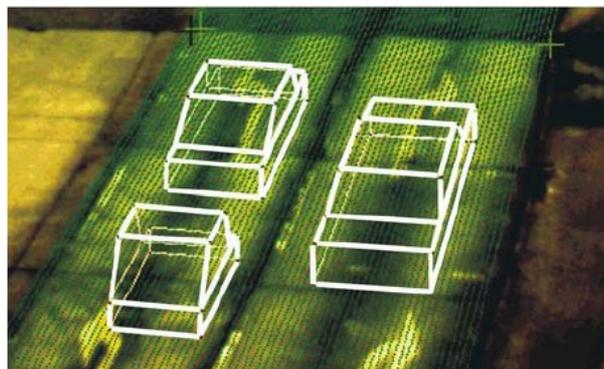


Abbildung 4: Projektion und Fitting von 3D Fahrzeug Modellen in [22].

Die Berechnung des **Histogramms** beschreibt die relativen Häufigkeiten der Pixelfarben und bietet eine differenziertere Repräsentation eines Objekts als der dominante Mode. Um die Datenmenge klein zu halten, wird der Farbraum immer quantisiert, und in so genannte Bins eingeteilt (Beispiel: Der RGB-Farbraum wird von der Auflösung 256x256x256 in jeweils 32 Bins zerlegt); dies ist auch sinnvoll, weil im quantisierte Histogramm Rauschen stärker unterdrückt wird [23]. Das Histogramm entspricht im wesentlichen einer einfachen **Signatur** (manchmal auch „fingerprint“ genannt) eines Fahrzeuges; die $N \times N \times N$ Bins können konkateniert und als Merkmalsvektor interpretiert werden, oder -wie in [24] - in Form einzelner Kanäle verwaltet werden. [10] beschäftigt sich mit Wiedererkennen von Fahrzeugen in unterschiedlichen Bildbereichen

einer Kamera. Es werden verschiedene Metriken zum Vergleichen von Farbhistogrammen analysiert und hinsichtlich der Fehler-Kenngrößen **False Acceptance Rate** (= Anzahl an gefundenen, aber falschen matches) und **False Rejection Rate** (also Anzahl von nicht gefundenen, aber richtigen matches) evaluiert. Der Schluss wird gezogen, dass Maßnahmen zu Verringerung des einen Fehlers typischerweise den jeweils anderen Fehler wiederum erhöhen, also hier immer nur ein Kompromiss möglich ist. Als Erkenntnis geht auch hervor, dass Histogramme alleine nicht ausreichen, Fahrzeuge (auf einem Kamerabild!) zuverlässig wieder zu erkennen, und es wird erwogen, bei Wiedererkennung *mit einer zweiten Kamera* weitere Merkmale einzubeziehen.

Eine frühe und wichtige Arbeit ist [24], die *nur* auf Farbmerkmalen basiert, um Fahrzeuge auf zwei Kameras zu identifizieren. Im vorgestellten *Autocolor* System wird das Farbhistogramm in Form der einzelnen Farbkanäle berechnet und in eine Datenbank eingetragen. Außerdem werden Farbwerte der Straße rund um das Fahrzeug sowie Ort und Zeitpunkt der Detektion mit gespeichert. Die Zeitpunkte werden später verwendet, um mithilfe des erwarteten Ankunftszeitpunktes der Fahrzeuge auf der zweiten Kamera, Kandidaten für den Matching Prozess zu isolieren.

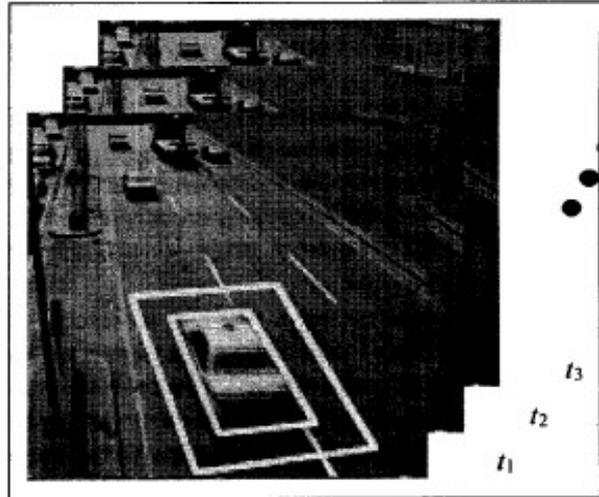


Abbildung 5: Bei der Objektdetektion in [24] wird neben dem Objekt auch ein Teil der Fahrbahn ausgeschnitten, um eine Angleichung der Licht- und Kameraverhältnisse zu ermöglichen.

Da Matching über Histogrammvergleich nicht robust gegen Beleuchtungsänderungen ist, verwenden die Autoren die Samples der Straße, um die Änderung der Beleuchtung (und auch unterschiedliche Kamerasensoren) zu kompensieren (s. Abbildung 5). Eine lineare Transformation, bestehend aus einem Verschiebungsvektor $t = (t_R, t_G, t_B)^T$ und einer Skalierung $s = (s_R, s_G, s_B)^T$, wird aus den Samples der Straße von Kamera 1 und Kamera 2 berechnet. Dazu werden Mittelwerte (für t) und Varianzen (für s) der Samples von den zwei Kameras verwendet. Diese Transformation wird auf die Histogramme der Kandidaten angewendet, bevor diese gematcht werden. Das Matching selbst wird über eine normalisierte Distanzfunktion zwischen 2 Histogrammen implementiert. Der Score wird aus den normalisierten Distanzen der Histogrammbins zwischen Zielobjekt und Kandidaten berechnet; der höchste Score wird als potentieller Match akzeptiert. Die Autoren unterstreichen den nicht akzeptablen Einfluss von false positives; daher werden Matches nur dann akzeptiert, wenn der (1) der Score einen Threshold T_1 übersteigt, und (2) der nächstbeste Score zumindest einen Abstand von T_2 hat. Trotz der guten dokumentierten Genauigkeit erscheint jedoch die (enge) Kopplung der

Kandidatenauswahl an die erwartete Ankunftszeit (die statistischen Parameter werden offline berechnet und *nicht* laufend aktualisiert) problematisch, sodass bei einer erheblichen Änderung der Verkehrssituation mit einer schlechten Performance zu rechnen ist.

Das Matching von Fahrzeugen mittels Farbmerkmalen wird auch in [15] untersucht. Allerdings wird hier ein **komplexer mehrstufiger Ansatz** verfolgt. Es wird eine Farbsignatur erzeugt, die einen „Fingerabdruck“ der dominanten Farben des Fahrzeugs repräsentieren soll; auf Details über diese Signatur gehen die Autoren leider nicht ein.

Neben Farbmerkmalen werden an jeder Kamera noch zusätzliche Verkehrsinformationen mit erhoben, die das Matching weiter absichern:

- Länge, Breite und 3D-Modell des Fahrzeugs (mit kalibrierten Kameras)
- Fahr(er)verhalten wie Spurtreue oder Abstandhalten zwischen Fahrzeugen
- Fahrzeuggeschwindigkeit für eine Schätzung der Ankunftszeit an der stromabwärts stationierten Kamera

Der resultierende, vierstufiger Matchingprozess wird in Abbildung 6 gezeigt. Die Autoren weisen schließlich darauf hin, dass eine sichere Erkennung einer kleinen Anzahl von Fahrzeugen einer Erkennung von mehr Fahrzeugen bei entsprechend größerer Unsicherheit vorzuziehen ist (eine Einschätzung, die deutlich auch in [24] geäußert wird und für diese Arbeit von grundlegender Bedeutung ist).

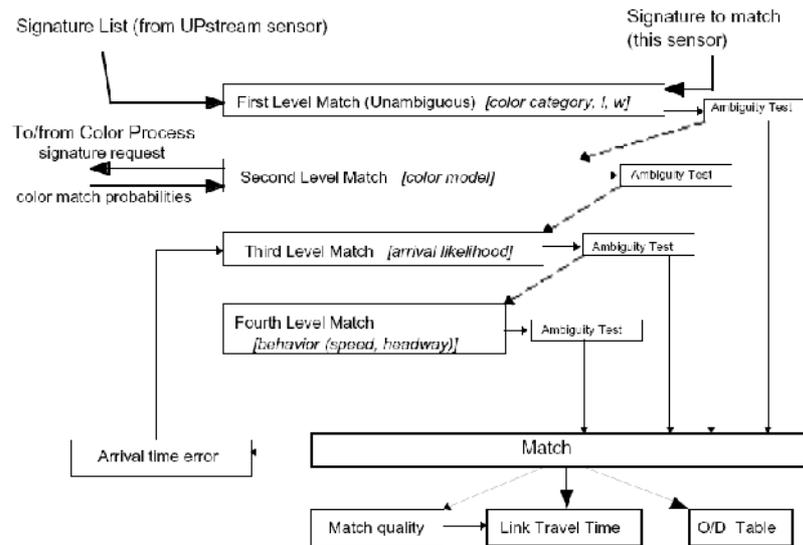


Abbildung 6: Ein mehrstufiger Matching Prozess verknüpft in [15] primäre Farbmerkmale mit sekundären Features, um eine möglichst stabile Erkennung zu gewährleisten

In [25] wird ebenfalls der Ansatz verfolgt, Fahrzeuge zur Ermittlung der Reisezeit zu matchen; die extrahierten Merkmale sind wieder Farbe, Fahrzeuggröße und -typ. Interessant ist hier die Tatsache, dass die Kamerabilder zuerst geometrisch entzerrt werden, wobei die projektiven Abbildungen der Kameras auf eine normalisierte Ansicht transformiert werden, um gleiche Ausrichtung und Größe zu erhalten. Die Fahrbahnsuren werden offline erstellt und in einem File abgelegt. Diese werden benutzt, um das Tracking und Data Association Problem stabiler zu lösen.

Die Farbmerkmale werden anhand ihres Farbgehalts im HSL (Hue-Saturation-Lightness) Modell unterschieden: bunte Fahrzeuge werden dabei noch genauer über ihren Farbton (Hue) modelliert (s. Abbildung 7). Zum Bestimmen des Fahrzeugtyps werden die (radiometrisch und geometrisch normalisierten) Fahrzeugbilder mittels Template Matching einer von 4 Kategorien zugeordnet.

Das Ergebnis des Matchings zwischen 2 Fahrzeugen ist das gewichtete Mittel aus den einzelnen Scores für Typ, Farbe und Größe. Für den Score des Farbhistogrammes wird dabei das innere Produkt der Bins berechnet. Zusätzlich wird auch das Verhalten von Fahrzeugverbänden für das

Matching herangezogen; diesen Ansatz findet man in anderen Veröffentlichungen häufig unter dem Begriff „Platoon Matching“ (Abbildung 8).



Abbildung 7: Einteilung der Fahrzeugfarbe in chromatische und achromatische Kategorien in [25]

Das Matching erfolgt über dynamische Programmierung (DP-Matching). Dieses Verfahren wird auf gewichtete Gitterstrukturen bzw. Matrizen angewendet, um die Kosten für den optimalen Pfad zwischen zwei Endpunkten zu finden. Der optimale Pfad selbst wird über Backtracking vom Endpunkt aus gefunden. In den rekursiven Formeln für das DP Matching werden die bereits erwähnten Spurinformatoren berücksichtigt, wie auch mögliche Richtungsänderungen bei Kreuzungen. Die Evaluierung zeigt, dass eine grobe Erkennung von z.B. Stauphasen mit diesem System möglich ist, was die Autoren auf die ausreichende Korrelation der Fahrzeugsequenzen im Testsystem zurückführen.

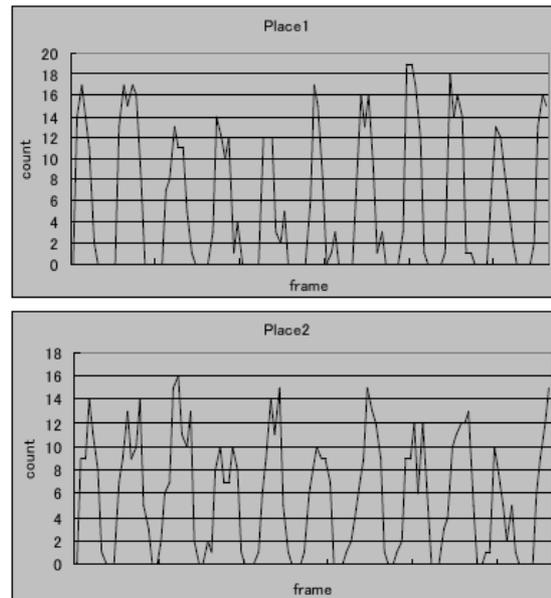


Abbildung 8: Sequence Matching nutzt Information über die Ankunft der Fahrzeuge auf Kamera 1 und Kamera 2 in [25].

Einen kombinierten Ansatz verfolgen Sun et al. in [26]: die individuell eruierten Merkmale sind die induktive Signatur (s. Abbildung 3) und Amplitude, Farbe, Länge, und Geschwindigkeit eines Fahrzeugs. Darüber hinaus wird als abgeleiteter aber wesentlicher Parameter noch die Platoon Reisezeit (auf einer Kamera!) ermittelt. Allerdings muss man bei dieser Arbeit schon von Multisensor Fusion sprechen muss, da neben Videokameras auch herkömmliche induktive Schleifendetektoren mitverwendet werden. Die extrahierten Merkmale werden mit einer Multi Object Optimization (**Lexikographische Analyse**, siehe auch [27]) gematcht; dazu werden 3 verschiedene Ziele optimiert:

1. Das Zeitfenster zwischen den zwei Kameras innerhalb dessen Matches akzeptiert werden, auf ein ganzes *Platoon* bezogen
2. Dem gewichteten Mittelwert aus den Scores aller Merkmalsvektoren aller möglichen Platoons als dem Gesamtscore. Dies entspricht einer linearen Fusion der Einzel-Scores aus den unterschiedlichen Sensordaten.

3. Das Zeitfenster zwischen den zwei Kameras innerhalb dessen Matches akzeptiert werden, auf ein einzelnes *Fahrzeug* bezogen

Der Score mit dem besten Resultat gemäß der Ziele wird jeweils ausgewählt (Nearest Neighbor Classifier).

Es werden nur 2 Fahrbahns Spuren einzeln überwacht und damit die Komplexität um einen kombinatorischen Faktor reduziert; das Umspuren von Fahrzeugen ist mit 2 % ohnehin selten. Die Performance des Verfahrens mit stets mehr als 90 % Erkennungsrate bei Verwendung von mehr als 90 % aller Fahrzeuge ist sehr gut zu bewerten. Als weiteres Ergebnis wurde (für die verwendeten Daten) eine optimale Platoongröße von 3 Fahrzeugen ermittelt. Leider geben die Autoren keine Details über die verwendeten Farb-Merkmale bekannt.

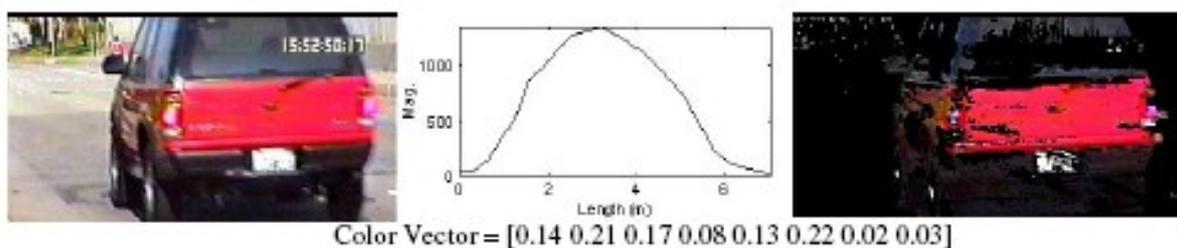


Abbildung 9: Beispiele für verwendete Features in [26]: Signatur und Farbe.

Von denselben Autoren stammt jedoch eine weitere Veröffentlichung [23], in der die Fusion von induktiven- und Farbmerkmalen zur Wiedererkennung vorgeschlagen werden. Es wird im Wesentlichen ein Farbhistogramm berechnet, wobei eine Quantisierung in $5 \times 5 \times 5 = 125$ Bins im RGB-Farbraum erfolgt; die Bins ergeben auch den Farbmerkmalsvektor. Als Vorverarbeitungsschritt wird zusätzlich noch eine Spreizung des Kontrastumfangs durchgeführt. Die Berechnung der Scores und das Matching sind wie in [26]. Bei der Evaluierung zeigt sich, dass die induktiven Aufnehmer gegenüber den gewählten Farbmerkmalen allerdings weit performanter sind (ca. 91% vs. 78%). Das beste Resultat wird dennoch unter Einbeziehung der Bildinformation erzielt.

Kogut und Trivedi verwenden ebenfalls Farbhistogramme als Merkmale in [11]. Diese werden zusätzlich normalisiert, um den Bedingungen auf freiem Feld (Beleuchtungsänderungen) Rechnung zu tragen; es wird mit 32 Bins pro Kanal quantisiert. Die Implementierung zielt vor allem auf eine Entlastung der Kommunikation durch lokale Vorverarbeitung ab, es soll ein intelligenter Verkehrssensor entstehen, der mit dem Zentralmodul und anderen Knoten nur mehr High-Level Informationen austauscht.

Das beschriebene System betont die Modellierung der Unsicherheit in den eingesetzten Modulen (wobei hier ausschließlich Kameras als Sensoren eingesetzt werden):

- Bei der Moving Object Detection wird das Aktualisieren des Referenzbildes über Gewichtungsfaktoren gesteuert, die aus Rauschen der Trainingsdaten berechnet werden
- Bei der Berechnung der Distanz zwischen Farbhistogrammen wird die Mahalanobis-Distanz verwendet, die Abstände in Einheiten von Erwartungswert und Kovarianzmatrix ausdrückt; die Kovarianzmatrix modelliert hier die Variabilität durch Rauschen und Beleuchtungsänderung und wird aus Trainingsdaten offline berechnet
- Aus der allgemeinen Bewegungsgleichung wird von den Agenten stromaufwärts eine Ankunfts-wahrscheinlichkeit der Fahrzeuge für die weiter stromabwärts liegenden Sensoren generiert und über das Netzwerk propagiert. Diese Ankunftszeit wird mit einem unscharfen Intervall beschrieben, die Unsicherheit dabei mit der Theorie der fuzzy sets beschrieben (die sog. **Domain Occurrence Possibility**). Laufende Beobachtungen fließen in das Modell ein und ermöglichen die Auflösung von bestimmten Mehrdeutigkeiten (durch **temporal fusion**, siehe Abbildung 10). Mehrdeutigkeiten werden durch Multiple Hypothesis Testing modelliert.

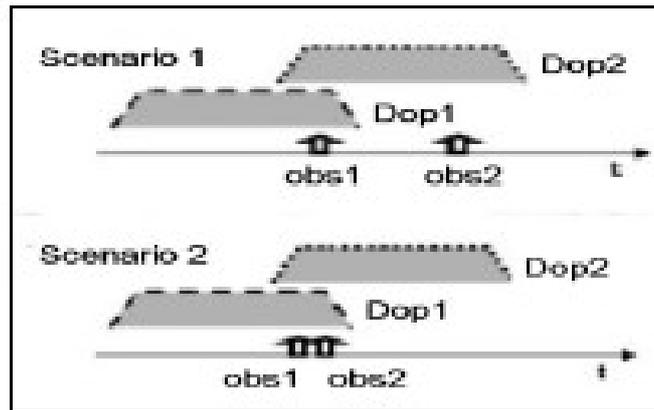


Abbildung 10: Durch unscharfe Intervalle der Ankunftszeit an den stromabwärts gelegenen Sensoren in [11] entstehen Mehrdeutigkeiten, wenn ähnliche Objekte 1,2 auf einem Sensor beobachtet (Obs1, Obs2) werden. DOP1 und DOP2 sind die Verteilungen der Ankunftszeit für die Objekte 1,2. Das erste Szenario kann durch temporale Fusion aufgelöst werden, sobald Beobachtung Obs2 gemacht wird. Szenario 2 kann nicht eindeutig aufgelöst werden.

Zusammenfassend kann gesagt werden, dass bis zu moderatem Verkehrsaufkommen (ca. 20 Fahrzeuge pro Minute) durch die temporal fusion gute Ergebnisse erzielt werden, bei höheren Verkehrsdichten aber schnell zu viele Mehrdeutigkeiten entstehen, die nicht mehr aufgelöst werden können und deren Hypothesen abgelehnt werden müssen (d.h. der Match kann nicht entschieden werden).

In [28] werden Analyse der Farbstruktur und der räumlichen Positionierung der Fahrzeuge zueinander kombiniert. Im Gegensatz zu [26] werden aber die Fahrbahnen nicht isoliert betrachtet, Platoons können mehrere Spuren umfassen und müssen nicht immer strikt dieselbe Reihenfolge einhalten.

Das (Low Level) Vision Modul selbst verbindet kanten- und regionenbasierte Segmentierung, wobei erstere auch Schatten detektieren kann, und durch die zweite Methode falsch segmentierte Blobs besser getrennt werden können. Die extrahierten Farbfeatures entsprechen denen in [24] und werden mit statistischen Parametern der Farbkanäle beschrieben (μ , σ). Ein einfaches Trackingmodul identifiziert einzelne Fahrzeuge und unterstützt das Finden der Platoons. Neue

Platoons werden gebildet, wenn Fahrzeuge eine vordefinierte Bildregion verlassen oder in diese eintreten. Mehrfache Zuordnungen sind dabei möglich, nicht aber isolierte Fahrzeuge ohne Platoon. Die Beziehung der Fahrzeuge in den Platoons wird über Graphen modelliert (s. Abbildung 11); die Graphen sind vollständig, daher ist aus Performancegründen die Anzahl der Fahrzeuge in einem Platoon auf 6 beschränkt. Top down betrachtet, werden immer Platoons gematcht (mit dem Spezialfall, dass ein Platoon auch nur aus 1 Fahrzeug bestehen kann). Das Graph Matching berechnet den Matching Score als Summe der Kosten, die entstehen, wenn ein Graph A mittels definierter Operationen in einen Graph B transformiert werden soll. Diese Operationen sind:

- das Umbenennen von Kanten bzw. Knoten
- das Löschen von bzw. Knoten
- das Hinzufügen von bzw. Knoten

Die Summe aller notwendigen Operationen ergibt die Gesamtkosten; aus 2 Kandidaten-Graphen mit minimalen Kosten zum Test-Graphen wird der optimale Match ausgewählt, wenn sowohl ein absoluter als auch ein relativer Threshold (zwischen bestem und zweitbestem Match) eingehalten werden.

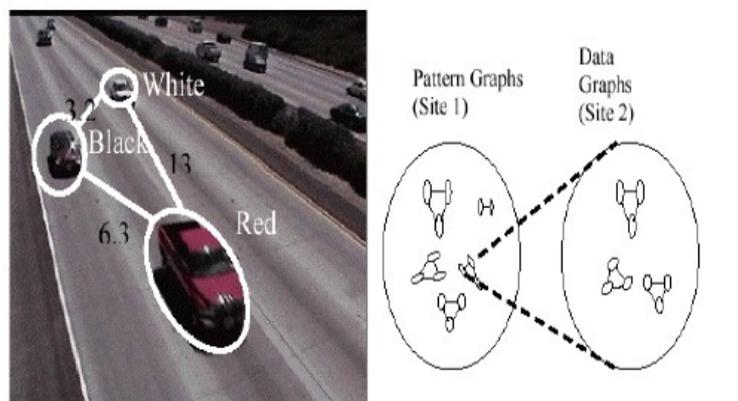


Abbildung 11: Platoons werden in [28] zu Graphen aufgebaut. Die Knoten sind die Fahrzeuge mit ihren Features, die Kanten werden als räumliche Distanz bewertet. Matching wird als approximiertes Graph Matching gelöst.

Interest Points

Diese Verfahren haben in der Bildverarbeitung zunehmend an Aufmerksamkeit gewonnen; neuere Algorithmen wie SIFT (Scale Invariant Features) [29] oder MSERS (Maximum Stable Extremal Regions) [30] werden sehr erfolgreich in der Object Recognition eingesetzt. Dennoch gibt es kaum Arbeiten im Bereich der streckenbezogenen Verkehrsanalyse.

In [31] wird in erster Linie ein sehr robustes Trackingsystem entwickelt. Als „Nebenprodukt“ wird das diskontinuierliche Matching über mehrere Standorte hinweg beschrieben. Es werden unterschiedliche Merkmale extrahiert (Linien, Eckpunkte und Regionen („Blobs“)), um unterschiedliche Ziele zu erreichen: Linien und Eckpunkte werden dabei hauptsächlich für Lokalisation und Ausrichtung verwendet, da sie verhältnismäßig genau bestimmt werden können; die Regionen werden aufgrund ihrer besseren Abdeckung des gesamten Objekts für das Matching verwendet.

Beim Tracking von Frame zu Frame (auf *einer* Kamera) werden Blob Features benutzt (ähnlich MSER), das Matchen der Blobs wird über die Earth Movers Distance (EMD) gelöst, die die Distanz der Blobs in örtlicher und visueller Hinsicht berücksichtigt. Aus diesem Tracking wird laufend ein Modell gelernt. Beim Matching zwischen verschiedenen Standorten kann die Ausrichtung („Pose“) sehr unterschiedlich sein, daher werden orthogonale Linien als Merkmale extrahiert, um Testkandidaten in die gleiche Ausrichtung bringen zu können. Über Ähnlichkeitstransformationen wird versucht, eine Korrespondenz zwischen den Linien des Test- und Modellbildes herzustellen. Für das eigentliche Matching werden Merkmale wie das *Farb Korrelogramm*, die Chamfer-Distanz oder normalisierte Korrelation benutzt.

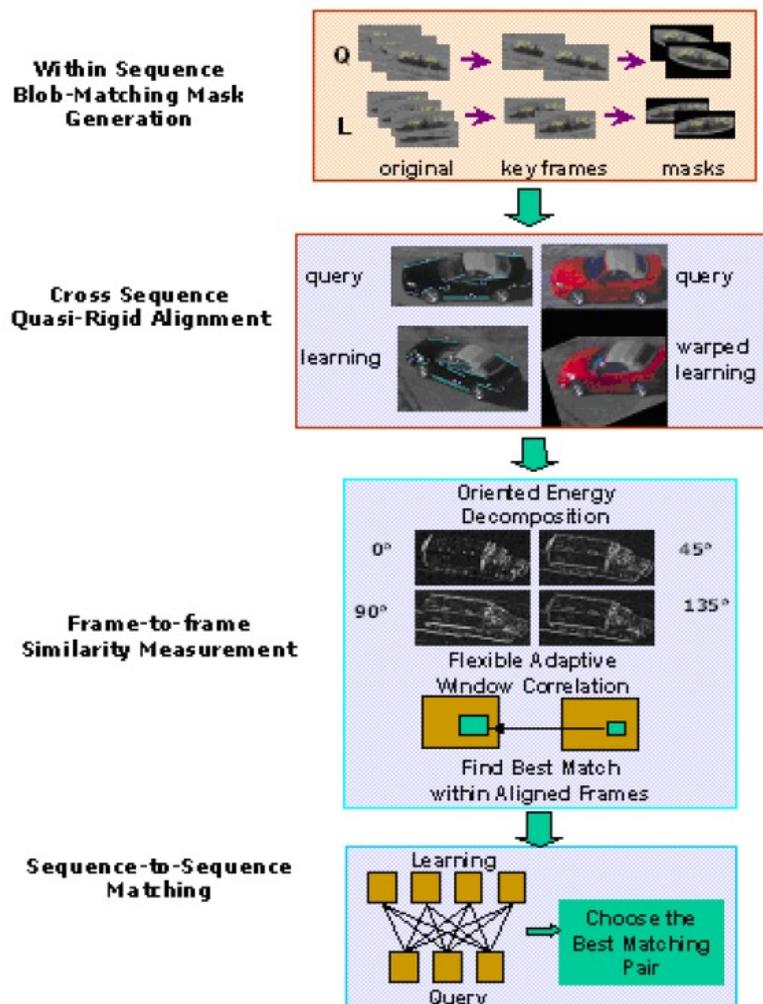


Abbildung 12: Überblick über die Matching Strategie in [31].

Probabilistische Verfahren

Die folgenden Verfahren extrahieren zwar auch Features aus den Bildern, der Ansatz beim Matching ist jedoch unterschiedlich, da über statistische Verfahren die erwartete „Appearance“ geschätzt wird.

In [32], wird aus den Beobachtungen eines Fahrzeuges auf einer Kamera, die zu erwartende Beobachtung des Fahrzeugs auf einer anderen Kamera abgeleitet. Es wird ein Identitätskriterium eingeführt, das die Beobachtungen der „Lebensgeschichte“ zweier Fahrzeuge miteinander

verknüpft und die Wahrscheinlichkeit der Identität aus der Wahrscheinlichkeit von Beobachtungen bestimmter Ereignisse herleitet. Jede Beobachtung o_j^d eines Fahrzeugs j wird als ein Tupel (r_j^u, f_j^u) (upstream) bzw. die Beobachtung o_i^u eines Fahrzeugs i als Tupel (r_i^d, f_i^d) (downstream) betrachtet, in der die Position (r) und ein allgemeiner Featurevektor (f) eingetragen werden. Die Erscheinungswahrscheinlichkeit (Appearance Probability) wird berechnet als $P(o_j^d | o_i^u)$, d.h., wie wahrscheinlich ist es, dass eine Beobachtung o_j^d des Objektes i stromabwärts von einem Objekt stammt, das ursprünglich die Beobachtung o_i^u stromaufwärts gemacht hat ?

Die Erscheinungswahrscheinlichkeit wird als Produkt von 4 verschiedenen Verteilungsfunktionen berechnet, in der Spur (x), Größe (w,l), Farbe (h,s,v) und prognostizierte Ankunftszeit berücksichtigt sind. Aus den Matches werden ständig die Parameter der Verteilungen neu gelernt. Das Matching wird als gewichtetes bipartites Zuordnungsproblem gelöst, in dem neue und „fehlende“ Fahrzeuge berücksichtigt werden.

Name	Description
t	time of observation
x	lane position (1, 2, 3, etc.)
y	distance along lane
\dot{x}	lateral velocity
\dot{y}	forward velocity
w	vehicle width
l	sum of vehicle length and height
h	mean vehicle color hue
s	mean vehicle color saturation
v	mean vehicle color value
C	histogram of color distribution over vehicle pixels

Abbildung 13: Die in [32] verwendeten Einzel-Features des Featurevektors.

Aufbauend auf die Arbeit von Huang und Russel [32], modifiziert [33] jedoch das Konzept der paarweisen Erscheinungswahrscheinlichkeiten, da sich dieses nicht auf die Anwendung mit vielen Kameras skalieren lässt. Über intrinsische Variablen wird ein unveränderlicher Zustand der Objekte beschrieben, sodass Beobachtungen nicht mehr von früheren Beobachtungen abhängen; damit kann das globale Zustandsmodell aufgespalten und einfacher beschrieben werden. Das Berechnen der wahrscheinlichsten Zuordnungen von n Objekten wird über Markov Chain Monte Carlo (MCMC) Inferenz gelöst. Bei der Evaluierung wird nur Farbe als intrinsischer Parameter benutzt; es zeigt sich eine sehr stabile Performance der Erkennungsrate bei zunehmendem normalverteiltem Rauschen der Sensoren gegenüber [32].

3.2 Abschließende Bemerkungen

Aus den vorgestellten Arbeiten lassen sich bei allen Unterschieden in den Algorithmen folgende zentrale Aussagen formulieren:

1. Es ist nicht ausreichend bzw. ratsam, zu einfache Merkmale zu verwenden (z.B. nur ein Histogramm). Erst durch Kombination mehrerer Merkmale (Größe, Form, ...) bzw. Verwendung komplexerer Merkmale (z.B. Textur) wird eine akzeptable Erkennungsrate erreicht.
2. Das Einbeziehen von Informationen aus dem Kontext (z.B. zur Schätzung der Ankunftszeit) ist ein wesentlicher Faktor zur Erhöhung der Performance und Erkennungsrate (Einschränkung der Anzahl zu matchender Objekt-Kandidaten).
3. Durch Berücksichtigung von Platoons kann die Performance weiter erhöht werden.

4. Für Zwecke wie die mittlere Reisezeit, ist es wichtiger, weniger Objekte zuverlässig zu erkennen, als möglichst viele Objekte mit größerem Fehler.
5. Farbe bzw. Farbstruktur ist ein Merkmal, das sich gut für Objektidentifizierung eignet.

Im nächsten Kapitel wird ein Verfahren vorgestellt, das diese Aspekte (außer (3)) berücksichtigt, um so ein robustes Fahrzeug Matching zu implementieren.

4 Fahrzeugwiedererkennung und Matching

4.1 Allgemeines

In diesem Kapitel wird ein Verfahren zum Matchen von Fahrzeugen vorgestellt. Die Aufgabe besteht darin, aus zwei Mengen von Fahrzeugbildern, die jeweils von einer Kamera stromaufwärts und stromabwärts aufgenommen wurden, diejenigen Bilder zu finden, die mit hoher Wahrscheinlichkeit das gleiche Objekt darstellen. Die Bilder sind das Ergebnis zweier (als gegeben betrachteten) Segmentierungs- bzw. Detektionsstufen, wie sie im Anhang beschrieben wird. Die wesentliche Idee des Verfahrens besteht in einer Selektionsstufe, die nur möglichst gut für das Matching geeignete Objekte berücksichtigt, sowie der Extraktion komplexer Farbmerkmale.

4.2 Struktur des Verfahrens

Das präsentierte Verfahren geht in 3 Schritten vor:

1. die Eignung der Objektbilder für eine Extraktion von Merkmalen wird bestimmt.
2. aus denjenigen Objekten, die sich im ersten Schritt als geeignet erwiesen haben, werden Merkmale bzw. Merkmalsvektoren berechnet.
3. die berechneten Merkmalsvektoren werden miteinander gematcht, um die wahrscheinlichsten Paarungen zu erhalten.

Während die eigentliche Merkmalsextraktion (Stufe 2) mit einer Methode durchgeführt wird, die an eine schon existierende - der **Multimodal Neighborhood Signature (MNS)** [34] angelehnt ist, besteht der neue Ansatz dieser Arbeit in der ersten Stufe, der Objektselektion, die im

wesentlichen eine Verknüpfung aus mehreren Kriterien vorschlägt, um die Tauglichkeit eines Bildes zur Merkmalsextraktion zu bestimmen. Dieser Schritt trägt vor allem zur Performance und Stabilität bei. Zur Stabilisierung werden außerdem noch zusätzlich alle Fahrzeuge in eine von 4 Fahrzeugklassen eingeteilt, und eine zeitliche Bedingung für die mögliche Ankunftszeit auf der zweiten Kamera gestellt. Im folgenden werden nun die 3 Schritte des Verfahrens detaillierter herausgearbeitet.

4.3 Objektselektion

Allgemeines

Die Berechtigung für eine Objektselektion ergibt sich aus der Überlegung, dass im vorliegenden Einsatzgebiet keine vollständige Verarbeitung aller Eingangsdaten notwendig ist. Daher genügt es, nur von Fahrzeugen mit „günstigen“ Eigenschaften Merkmale zu berechnen, vorausgesetzt dass diese einen akzeptablen Anteil der tatsächlich beobachteten Fahrzeuge ausmachen.

Zwei Kriterien werden für die Selektion herangezogen:

- Größe der Bilder. Aus zu kleinen Fahrzeugen können nicht genügend verlässliche Features berechnet werden. Hier wird also bereits Top-Level Vorwissen aus der spezifischen Anwendung in das System integriert, um die Erkennungsrate zu erhöhen.
- Da die Merkmale auf Farbe basieren, spielt die „Buntheit“, also die Anzahl an Farben im Bild, die eine gute Sättigung aufweisen, eine entscheidende Rolle.

Wesentlich bei der Überlegung, welche Merkmale man aus Objekten berechnen soll, ist die Bedingung, dass die zu analysierenden Objekte durch diese Merkmale auch gut repräsentierbar sein müssen. Für Objekte, deren Aussehen durch geschmeidige Flächen und weiche Farbübergänge dominiert wird, kann ein Kantendetektor nur wenig nützliche Ergebnisse liefern. In analoger Weise liefert ein auf Farbkompositionen beruhendes Verfahren bei einfarbigen Objekten oder solchen, die keine deutlichen Farbübergänge aufweisen, schlechte Resultate. Daher ist es auch gar nicht sinnvoll, für diese Objekte Merkmale zu berechnen, da nur die Wahrscheinlichkeit des Auftretens von mismatches erhöht würde (vor allem durch Zunahme an false positives). Es wird also ein Filter implementiert, der

1. schnell im Vergleich zur Merkmalsextraktion ist
2. eine Aussage über die Tauglichkeit der Bilder zur Merkmalsextraktion liefert

Dafür werden verschiedene Constraints eingeführt. Das bereits weiter oben erwähnte Kriterium der Bildgröße ist leicht nachzuvollziehen; zum einen kommen Bewegungsartefakte bei kleinen Bildgrößen stärker zum Tragen, da ohnehin oft schwach ausgeprägte Farbstrukturen damit vollends verwaschen werden; schließlich ist die Geschwindigkeit bei kleinen Fahrzeugen (PKW Klasse) erwartungsgemäß höher als bei der Klasse der LKW oder Busse. Außerdem sind in dieser Kategorie auch die Erscheinungsformen sehr ähnlich: Busse oder LKW tragen für gewöhnlich Firmenlogos und sind farblich eher reich texturiert. PKW hingegen sind fast ausschließlich einfarbig und es ist beispielsweise praktisch kaum möglich, ein rotes Fahrzeug zweier unterschiedlicher Hersteller auf einem Bild der Größe 50x50 Pixel zu unterscheiden.

Da später Farbkompositionen berechnet werden, sollte das Objekt also möglichst „bunte“ Farben aufweisen, die im noch dazu Bild durch eine

nennenswerte Anzahl an Pixeln repräsentiert sind. Je mehr dieser Farben existieren, umso größer die erwartete Anzahl geeigneter Features; außerdem ist die Auftretenswahrscheinlichkeit derartiger Objekte klein im Vergleich zu Objekten mit monochromer Farbstruktur (siehe z.B. [20], für die relative Häufigkeit von Fahrzeugfarben) sodass die Wahrscheinlichkeit einer erfolgreichen Wiedererkennung größer ist. Der wichtigste Schritt besteht also im Ermitteln dieser dominanten Farben; dazu wird eine Clusterung im Merkmalsraum (s. Kapitel 3) durchgeführt.

Ein besonders einfaches und flexibles Verfahren zur Cluster-Analyse ist der **Mean Shift** Algorithmus, ursprünglich von Fukunaga und Hostetler entwickelt, jedoch erst von Comaniciu und Meer [35] im Bereich der Bildsegmentierung eingesetzt. Es ist unter die nicht-parametrischen Verfahren im Bereich der nicht-überwachten Klassifikation einzuordnen, da bezüglich der Daten keine Annahmen gemacht werden bzw. keine Parameter eines Modells geschätzt werden.

Allgemeines

Featurespace Analysis mit dem Mean Shift Verfahren

Der Mean Shift gehört zur Gruppe der Kernel-basierten Verfahren. Unter einem Kernel $K_{h,A}(x_0, x_i)$ versteht man eine Gewichtungsfunktion, die die Daten (Samples) x_i entsprechend ihrer Distanz zu einem Punkt x_0 unterschiedlich gewichtet [36]; durch die Einführung eines Kernels können glatte Schätzfunktionen realisiert werden, da der Einfluss der Samples auf den Schätzwert an einem Punkt x_0 durch ihre Entfernung zu diesem Punkt bestimmt wird, und „sanft“ abgeschwächt werden kann.

A_{dxd} ist hier eine positiv semidefinite Matrix zur Gewichtung der einzelnen Koordinaten („Bandwidth“-Matrix des Kernels), λ die Größe der Nachbarschaft um x_0 , innerhalb der die Daten betrachtet werden

(Breite des Kernels). Um die Komplexität der Berechnung zu vermindern, wird anstatt eines voll parametrisierten Kernels die Einheitsmatrix $A_{d \times d} = I$ verwendet, und die Bandbreite des Kernels mit einem einzelnen Parameter $h > 0$ definiert: $K_h(x_0, x_i)$. Dies ist natürlich nur unter der Voraussetzung eines isotropen Raumes (beispielsweise \mathbb{R}^d mit euklidischer Metrik) sinnvoll.

Seien nun $x_1 \dots x_n$ n Samples im d -dimensionalen Raum, die zufällig aus einer PDF $f(x)$ gezogen wurden. Dann stellt die Funktion

$$\hat{f}(x) = \frac{1}{n} \cdot \sum_{i=1}^n K_{\lambda, A}(x - x_i)$$

eine multivariate Schätzung der Dichtefunktion bzgl. des Kernels $K_{\lambda, A}$ am Punkt x dar.

Vereinfacht man für die Bandbreite (s.o.), so erhält man - nun mit dem Parameter h als der Fenstergröße bzw. Radius:

$$\hat{f}(x) = \frac{1}{n \cdot h^d} \cdot \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

Der Ausdruck

$$K\left(\frac{x - x_i}{h}\right)$$

bezeichnet die Kernelfunktion, zentriert um den Wert x , und normiert mit der Fenstergröße h . Vereinfacht gesagt, wird die Dichtefunktion als gewichtete Summe der Anzahl an Samples in einem gewissen Volumen (definiert durch die Fenstergröße) dividiert durch die Gesamtanzahl der Samples, geschätzt.

Viele verschiedene Kernel sind in der Literatur zu finden, wie z.B. Gaußkernel oder uniforme Kernel. Da Kernel Dichtefunktionen darstellen, müssen sie folgende Bedingung erfüllen:

$$\int_{\mathbb{R}^d} K(x) \cdot dx = 1$$

d.h. der Kernel hat ein Einheitsvolumen in \mathbb{R}^d .

Ein spezieller Kernel ist der *Epanechnikov* Kernel $K_{E,h}$ mit der allgemeinen Form für \mathbb{R}^d mit

$$K_{E,h} = \begin{cases} \frac{1}{2} c_{d,h}^{-1} (d+2) (1-x^T x) & \text{für } x^T x < h^2 \\ 0 & \text{sonst} \end{cases} .$$

Für diesen Kernel wird $\hat{f}(x)$ optimal bzgl. des **AMISE (Approximated Mean Integrated Square Error)**. Der Support des Kernels ist die d-dimensionale Hyperkugel mit Radius h , $c_{d,h}$ ist das Volumen der d-dimensionalen Hyperkugel; damit wird die oben angegebene Bedingung für das Integral über das Kernelvolumen erfüllt.

Aus der Schätzung der Dichtefunktion $\hat{f}(x)$ wird über Ableitung die Schätzung des *Gradienten* der Dichtefunktion ermittelt (dies setzt natürlich einen differenzierbaren Kernel voraus). Nach Ableitung von $\hat{f}(x)$ mit $K=K_E$ erhält man den Gradienten:

$$\hat{\nabla} f(x) = \frac{n_x}{n (h^d c_d)} \frac{d+2}{h^2} \left(\frac{1}{n_x} \sum_{x_i \in S_h(x)} [x_i - x] \right)$$

Separiert man den rechten Term und schreibt

$$M_h(x) = \frac{1}{n_x} \sum_{x_i \in S_h(x)} x_i - x$$

dann wird dieser Teil der Sample Mean Shift genannt. n_x bezeichnet die Anzahl von Samples, die in der Umgebung $S_h(x)$ liegen. $M_h(x)$ zeigt immer in Richtung des stärksten Wachstums der Dichte, daher kann daraus ein „Pfad“ abgeleitet werden, der zu einem stationären Punkt in der geschätzten Dichtefunktion führt, i.e. $\hat{\nabla} f(x) = 0$; diese Punkte stellen Moden dar. Aus dieser Eigenschaft lässt sich eine Methode zur Detektion von Moden entwickeln:

1. Berechne den Mean Shift $M_h(x)$ im Punkt x
2. Verschiebe das Fenster $S_h(x)$ um den Vektor $M_h(x)$

Werden die Schritte 1 und 2 bis zur Konvergenz – d.h. $\|M_h(x)\| < \epsilon$ – iteriert, so bewegt man sich über stetig dichter werdende Regionen Schritt für Schritt auf die Moden zu, in deren Einflussbereich sich die untersuchten Punkte befinden („Basins of attraction“ genannt.) Die Konvergenz des Verfahrens für diskrete Daten wurde u.a. in [37] bewiesen.

Abbildung 14 zeigt ein Farbbild, das stark ausgeprägte Farbregionen hat. In Abbildung 15 ist der dazugehörige Featurespace abgebildet (in unserem Fall sind die Merkmalsvektoren RGB-Farbkomponenten, dementsprechend liegt ein dreidimensionaler Raum mit den Koordinatenachsen R,G,B vor). Man sieht die gut ausgeprägten Moden, die auf die (auch visuell) recht gut identifizierbaren Farbregionen zurückzuführen sind. Prinzipiell ist die Darstellung des Featurespace (wie in Abbildung 2) als ein dreidimensionales Farbhistogramm zu interpretieren; die Farbe eines Datenpunktes gibt die Anzahl der Pixel an, die diesen Farbwert im Bild besitzen. (Die Regionen höchster Dichte befinden sich in den Clusterzentren, daher sind sie verdeckt.) Abbildung 16 schließlich zeigt das Ergebnis der Clusterung, wobei der Algorithmus selbst in Schema 3 am Ende des Unterkapitels „Implementierung“

dargestellt ist.



Abbildung 14 - Bild mit kompakten
Farbregionen

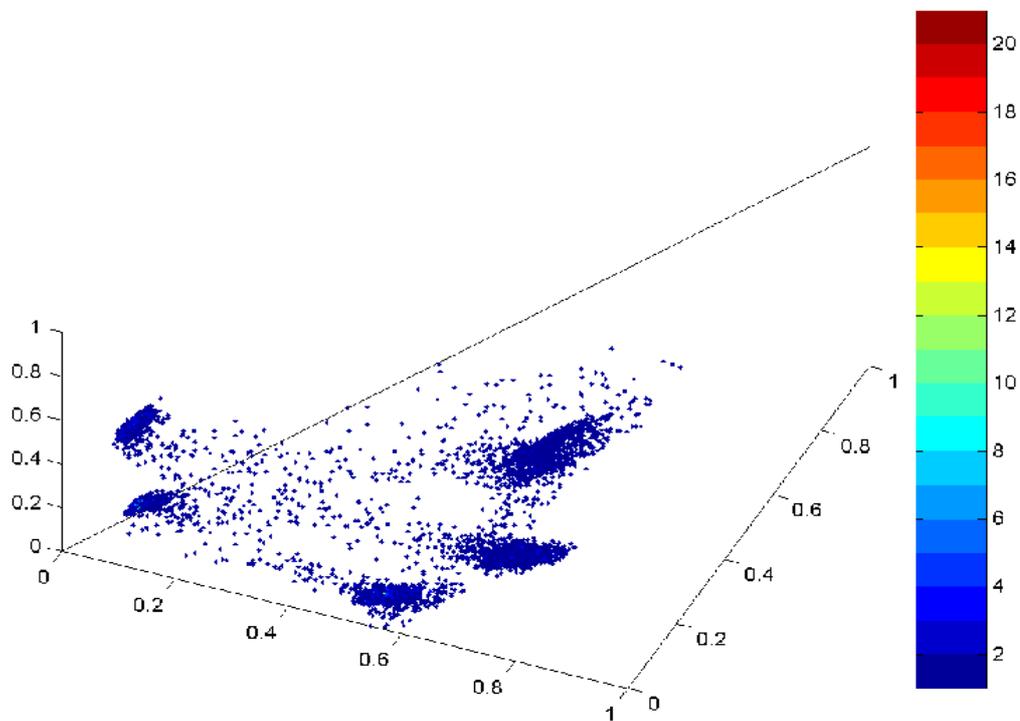


Abbildung 15 - Featurespace aus Abbildung 1 (die dazugehörige Colormap rechts im Bild gibt an, welche Farbe der Anzahl an Pixeln entspricht, die denselben RGB-Farbwert im Bild haben.)

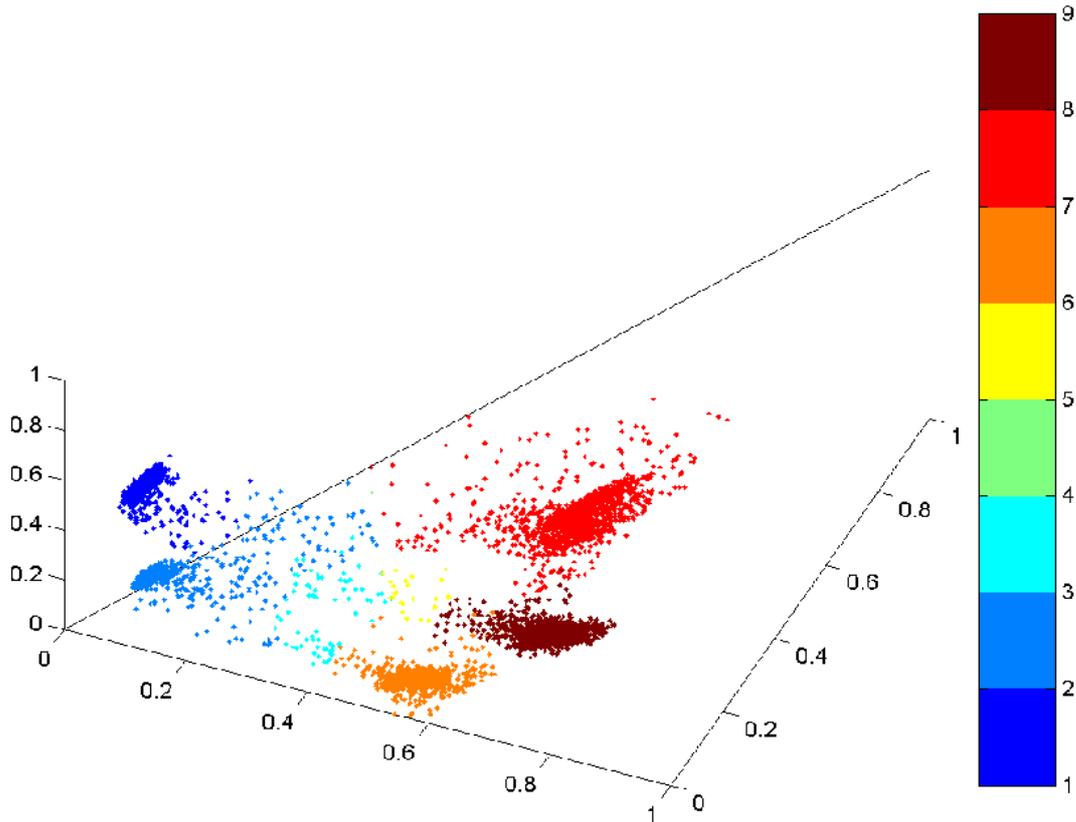


Abbildung 16 - Cluster aus Abbildung 1, detektiert mit dem Algorithmus aus Schema 3

Bestimmung der Buntheit

Aus Gründen der Effizienz werden nicht alle Pixel eines Bildes betrachtet, sondern lediglich eine Teilmenge davon. Diese Teilmenge entsteht durch Subsampling der Original-Bildfunktion mit einer quasi-zufälligen Samplingfunktion. Dazu wird ein Dithermuster verwendet, welches durch folgende - rekursiv definierte - Rechenvorschrift generiert wird [38]:

$$D^{(2)} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

$$D^{(2n)} = \begin{bmatrix} 4D^{(n/2)} + D_{00}^2 U^{(n/2)} & 4D^{(n/2)} + D_{01}^2 U^{(n/2)} \\ 4D^{(n/2)} + D_{10}^2 U^{(n/2)} & 4D^{(n/2)} + D_{11}^2 U^{(n/2)} \end{bmatrix}$$

$U^{(k)}$ bezeichnet die $n \times n$ -Matrix, in der alle Elemente 1 sind, und D_{ij}^2 das Element in Zeile i und Spalte j der Matrix D^2 .

Die auf diese Weise erzeugten Dithermatrizen der Größe 2x2, 4x4, 8x8 usw. können offline berechnet, und als Datenfiles abgespeichert werden, zur Laufzeit müssen sie lediglich in der richtigen Größe nachgeladen werden.

Durch eine Dithermatrix wird eine *Reihenfolge* bestimmt, in der Pixel aus einem Bild selektiert werden; mit einem Threshold kann zusätzlich bestimmt werden, wie viele Pixel insgesamt verwendet werden. Soll mit der Matrix D^2 beispielsweise ein Subsampling mit 50 % der Pixel durchgeführt werden, so würden die Pixel bis zum Median, also die Pixel 0 und 1, selektiert werden.

Es wird nun davon ausgegangen, dass für alle *selektierten* Pixel die Moden über den Mean Shift Algorithmus bestimmt wurden, und alle Pixel einem Mode zugeordnet sind. Dann ist $M = \{ m_1, \dots, m_l \}$ die Menge aller Moden mit $m_i = (R_i, G_i, B_i)$ als den Tripeln mit den RGB-Werten der Clusterzentren. $S = \{ s_1, \dots, s_l \}$ werden Support eines Mode genannt, d.h., die Anzahl an Pixeln, die jedem Mode zugeordnet sind. Es gilt also:

$$\sum_{i=1}^l s_i = n, \quad n \dots \text{Gesamtanzahl der selektierten Pixel des Bildes}$$

, da alle Pixel einem Mode zugeordnet werden.

Moden mit einem zu geringen Support werden verworfen. Diese entsprechen entweder unwesentlichen Bildregionen, oder sind überhaupt Bildstörungen. Die verbleibende Menge an Moden

$$M_{ess} = \{ m_i \mid m_i \in M \wedge s_i > T_{support} \}$$

werden die essentiellen Moden genannt ($T_{support}$ ist der Threshold für den Support).

Sehr dunkle Moden tragen erfahrungsgemäß wenig (verlässliche) Farbinformation, sodass durch diese keine wertvollen Features erwartet werden können. Daher werden Moden verworfen, wenn die Helligkeit, die über die Intensität

$$L = \frac{(R+G+B)}{3}$$

berechnet wird, unter einem Threshold T_L liegt.

Die verbleibenden Moden stellen somit eine minimale Palette der bildbestimmenden Farben dar. Es muss nun jedoch zusätzlich eine Quantifizierung der Qualität der Moden gefunden werden; dafür sind grundsätzlich zwei Kriterien ausschlaggebend:

1. Die Anzahl unterschiedlicher Moden. Diese korreliert mit der Wahrscheinlichkeit, aus diesen Moden (für das Objekt) möglichst viele repräsentative Kombinationen bilden zu können.
2. Die Sättigung der Farben. Umso weiter die RGB Werte der Moden von der Unbuntachse (die Hauptdiagonale $r=g=b$ des RGB-Würfels) entfernt sind, umso stärker sind die Features, und umso besser ist die Unterscheidbarkeit.

Die folgende Berechnungsmethode berücksichtigt diese Kriterien:

Es wird zuerst der Abstand aller Moden $m_i \in M_{ess}$ von der Unbuntachse berechnet. Danach werden von den Moden jene aussortiert, deren Abstand $d_i > \Delta_{min}$ ist; diese Moden entsprechen den ungesättigten Farben. Mit dem letzten Schritt wird verhindert, dass eine höherer

Anzahl ungesättigter Farben ein ähnliches Resultat erzielt, wie eine geringere Anzahl kräftigerer Farben. Der Abstand eines 3D Punktes $P = (r, g, b)^T$ von der in Parameterdarstellung gegebenen Gerade

$$\vec{g} = \vec{a} + t * \vec{b}$$

kann über folgende Formel berechnet werden:

$$d = \frac{|\vec{b} \times (\vec{P} - \vec{a})|}{|\vec{b}|}$$

Im Spezialfall der Unbuntachse ergibt sich mit $\vec{a} = (0, 0, 0)^T$ und $\vec{b} = (1, 1, 1)^T$ die einfachere Formel:

$$d = \frac{|(1, 1, 1)^T \times \vec{P}|}{\sqrt{3}}$$

Im Anschluss wird ein Score s als endgültige Maßzahl für die Qualität der Buntheit berechnet:

$$s = \sum d_i$$

Damit ist der Score proportional der Anzahl *geeigneter* Moden, sowie proportional zu deren *Abstand* vom nächsten Grauwert.

Alle Bilder, die gematcht werden sollen, werden dieser Analyse unterzogen. Für die Featuregenerierung werden nur mehr jene Bilder verwendet, die den Kriterien der Schärfe und Buntheit entsprechen.

Der Algorithmus ist im Detail noch einmal in Schema 1 zusammengefasst. Abbildung 18 und 17 zeigt typische Farbmoden von Fahrzeugen als Ergebnis dieses Schrittes.

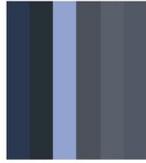


Abbildung 17: Oben: Alle essentiellen Moden. Mitte: Alle Moden, die nach Schritt (6) übrig geblieben sind. Unten: Das Fahrzeug. Der erzielte Score ist 88.



Abbildung 18: Bildaufbau wie in Abbildung 17. Der Score ist hier 253.

1. Verwirf den Bildausschnitt, falls es weniger als $M_{min} \times N_{min}$ Pixel hat
2. Subsampling über Dithermatrix ergibt Farbsamples
3. Wende den Mean Shift Algorithmus an und bestimme die Farbcluster in den Farbsamples:
 - Bestimme die Moden des Farbraumes
 - Ordne alle Pixel genau einem Mode zu; die Anzahl der Pixel pro Mode sei der **Support** S des Mode
4. Verwirf alle Moden mit Support $S < T_{support}$
5. Verwirf alle Moden (R,G,B) für die gilt:
 - 1) $\frac{(R+G+B)}{3} < T_L$
 - 2) $\delta_i < \Delta_{min}$
6. Berechne den Score als $s = \sum_i^M d_i$
 (mit M = Anzahl der Moden nach Schritt 5)
7. Verwirf das Bild für die Featuregenerierung, falls $s < T_{score}$

Schema 1 - Preselect Algorithmus

Implementierung des Mean Shift

Der Mean Shift Algorithmus ist in seiner Anwendung intuitiv nach zu vollziehen und einfach zu implementieren. Prinzipiell besteht die einfachste Umsetzung darin, alle Datenpunkte in den Featurespace abzubilden, und für jeden Datenpunkt über Mean Shift den Konvergenzpunkt zu finden.

Der große Nachteil ist die hohe Laufzeit für „ernsthafte“ Datenmengen, wie sie in der Bildverarbeitung häufig anfallen. Zwar konvergiert der Mean Shift für einzelne Datenpunkte sehr schnell (typischerweise 3-8 Iterationen pro Punkt bei RGB-Farbpixeln in natürlichen Bildern). Das wesentliche Problem ist zum einen die Berechnung der Funktion $S_h(x_j)$, d.h. das Finden derjenigen Punkte, die innerhalb der Kugel mit Radius h um x_j liegen. Bei naiver Implementierung muss dazu die Distanz zwischen x_0 und allen anderen Punkten $x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n$ berechnet werden (also $n-1$ Punkte), wodurch sich **je Iteration** eine Komplexität von $O(n)$ ergibt!

Glücklicherweise existieren Ansätze zur effizienten Lösung diese Problems. Eine einfache Methode besteht in der gitterförmigen Unterteilung des Raumes. Dabei wird der – hier dreidimensionale – Raum in Subvolumina unterteilt (Abbildung 19). Für jedes Subvolumen wird eine Liste aller darin enthaltenen Punkte angelegt. Wird die Größe der Volumina geeignet gewählt, so brauchen zur Bestimmung der Punkte $S_h(x_j)$ nur die Subvolumina untersucht werden, die unmittelbar an jenes angrenzen, in dem sich der Punkt x_j befindet (Abbildung 20). Dies wird lineare Indizierung bzw. Hashing genannt, da das betreffende Subvolumen in konstanter Zeit bestimmt werden kann.

Eine andere Möglichkeit ist die multidimensionalen Bereichssuche [39][40]; damit lässt sich die Komplexität stark reduzieren. Eine Implementierung kann beispielsweise effiziente Datenstrukturen wie kd-Bäume zur räumlichen Dekomposition der Daten verwenden; dadurch

werden die Punkte - anstatt in einem linearen Feld ohne Berücksichtigung der räumlichen Ordnung - entsprechend ihrer Koordinaten in eine Baumstruktur eingefügt (s. Abbildung 21). Damit ist eine Bereichssuche (*Range Query*) in $O(\log n+k)$ möglich, wobei n die Gesamtanzahl der Punkte, und k die Anzahl der tatsächlich benachbarten Punkte für den Query-Punkt darstellt [39].

Diese Methode wurde implementiert, und damit eine deutliche Steigerung der Performance erreicht. Dabei wird **einmalig** vor dem eigentlichen Verfahren ein größerer Zeitaufwand in die Strukturierung der Eingangsdaten investiert, um dann jedoch die vielen auftretenden Zugriffe im Laufe des Verfahrens sehr schnell durchführen zu können.

Folgende Erkenntnisse wurden außerdem verwendet, um die Performance weiter zu steigern:

- Punkte, die während der Suche nach „ihrem“ Mode besucht werden, werden als so genannte Spur gespeichert. Falls ein Punkt, dessen Mode bestimmt werden soll, in unmittelbarer Nähe ($\frac{1}{4}$ des Radius des Suchfensters) zu einem der Punkte einer Spur liegt, so wird er dem Mode zugeordnet, zu dem die Spur gehört, da eine große Wahrscheinlichkeit besteht, dass bei der Suche ohnehin auf dieser Spur vorgegangen wird.
- Alle Punkte, die in großer Nähe („**basin of attraction**“ genannt) zu einem bereits bestimmten Mode liegen, werden diesem automatisch zugeordnet, ohne dass der Konvergenzpunkt für diesen Datenpunkt wirklich berechnet wird.

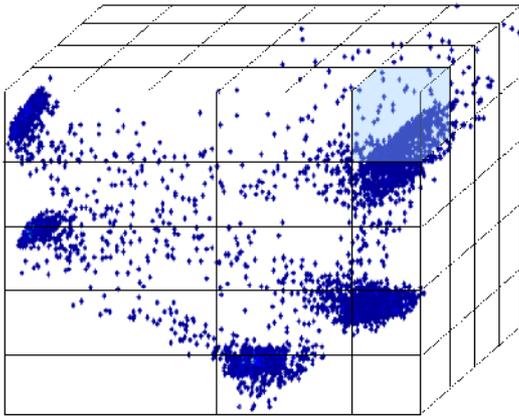


Abbildung 19 - Partitionierung des Featurespace nach dem Gitterverfahren. Der gesamte Datenbereich wird in Subvolumina unterteilt.

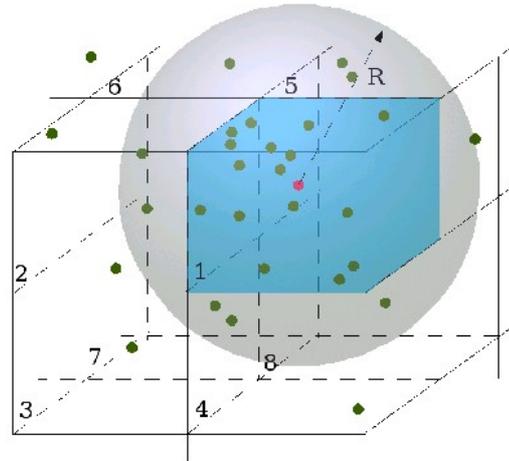


Abbildung 20 - Nachbarschaftsbeziehung zwischen Subvolumina. Sind die Subvolumina in der Größe (R) entsprechend gewählt, kann die Bereichssuche auf die umliegenden Würfel beschränkt werden.

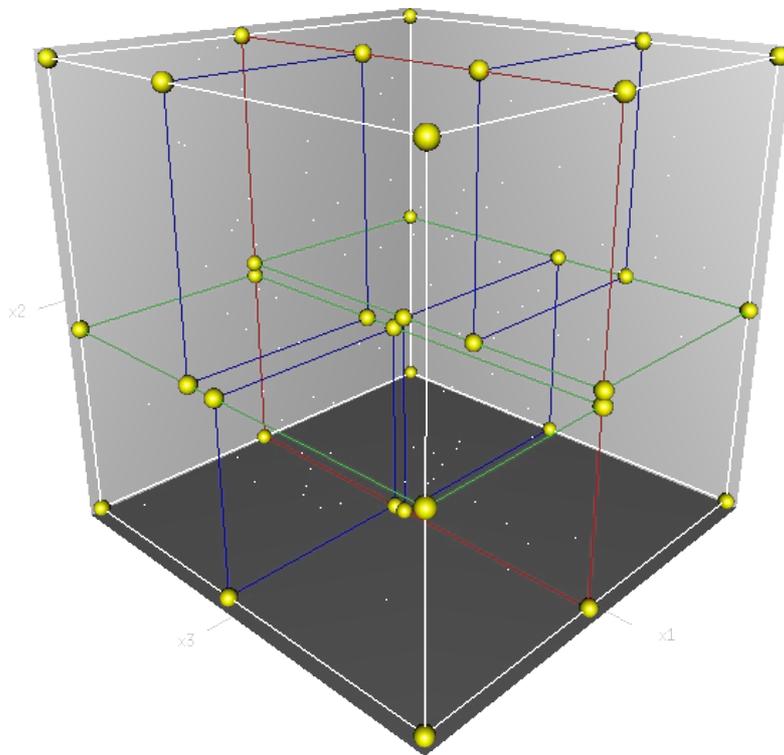


Abbildung 21: 3-dimensionaler kd-Baum. Die ursprüngliche Zelle (weiss) wird in einem ersten „Schnitt“ in 2 Teile unterteilt (rot). Diese werden im zweiten Schnitt (grün) jeweils wieder in 2 Teile geteilt, und in einem dritten Schritt nochmals (blau). Die gelben Punkte sind die Knoten des somit aufgespannten Baumes. (© Benjamin Tyner)

4.4 Die Featuregenerierung

Allgemeines

Objekte können über eine Vielzahl von Merkmalen charakterisiert werden.

Ein Möglichkeit besteht darin, lokale Farbkompositionen kleiner Bildregionen als Merkmale zu betrachten, und damit in weiterer Folge das Objekt im Bild zu repräsentieren. Diesen Weg geht die sog. Multimodal Neighborhood Signature. Das Verfahren wurde von Matas, Koubaroulis und Kittler als neuer Ansatz zu Color Image *Retrieval* und Color Object *Recognition* präsentiert [34]. Die vorliegende Implementierung der Feature Generierung und Feature Matching lehnt sich stark an diese Methode an.

Die Merkmalsextraktion über die MNS selbst liefert eine kompakte Beschreibung der Farbverteilung von Objekten. Dazu wird ein Bild in eine Anzahl kleiner überlappender Teilbilder („Patches“) zerlegt, und von diesen Patches werden die dominanten Farben bestimmt und alle Kombinationen daraus gebildet. Anschließend werden invariante Merkmale berechnet. Die so gefundenen Farbnachbarschaften stellen – in komprimierter Form - die Merkmalsvektoren dar. Diese Nachbarschaften werden in Form eines Vektors von Farbtupeln gespeichert.

Der Algorithmus

Vorverarbeitung

Die Eingangsbilder werden mit einem Gauß-Kernel der Größe 3x3 gefiltert, um Störungen und Rauschen abzuschwächen.

MNS

Als erster Schritt wird das Bild in eine überlappende Menge von kleinen Nachbarschaften zerlegt. Die Form ist dabei nicht kritisch; da sich rechteckige Bildbereiche besonders leicht behandeln lassen, werden diese im weiteren Verlauf verwendet; die resultierenden Patches partitionieren somit das gesamte Bild in Teilbilder.

Sei $I_{M \times N}$ das Originalbild, welches das Objekte enthält, für das Features berechnet werden sollen. Ein Maskenbild $B_{M \times N}$ wird mit folgenden Eigenschaften eingeführt:

$$B(x, y) = \begin{cases} 1 & \text{für Pixel } I(x,y) \text{ gehört zum Objekt} \\ 0 & \text{sonst} \end{cases}$$

Das Maskenbild resultiert aus der Bewegungssegmentierung; aus dem dabei entstehenden Differenzbild zwischen statischem Hintergrund und bewegtem Vordergrund(-objekt) wird ein binäres Bild erzeugt, in dem alle Vordergrundpixel auf „1“ und alle Hintergrundpixel auf „0“ gesetzt werden.

Die Zerlegung eines Bildes in Teilbilder wird definiert mit

$$P^{i,j} = (r_0, c_0, r_1, c_1) = (r_0, c_0, \min(r_0+n, M), \min(c_0+m, N))$$

mit

$$\begin{aligned} r_0 &= \text{round}((i-1) \cdot (1-o) \cdot n) + 1 \\ &\quad \text{und} \\ c_0 &= \text{round}((j-1) \cdot (1-o) \cdot m) + 1 \end{aligned}$$

und

$$i=1 \dots \text{ceil}\left(\frac{M}{n \cdot (1-o)}\right) \text{ und } j=1 \dots \text{ceil}\left(\frac{N}{m \cdot (1-o)}\right)$$

wobei die $P^{i,j}$ die Partitionierungen in Form von rechteckigen Bereichen mit der linken oberen Ecke $(r0, c0)$ und der rechten unteren Ecke $(r1, c1)$ darstellen. m, n sind Breite und Höhe des Patches in Pixeln, o der Grad der Überlappung ($0 \leq o < 1$). Abbildung 22 zeigt ein Beispiel für eine Zerlegung.

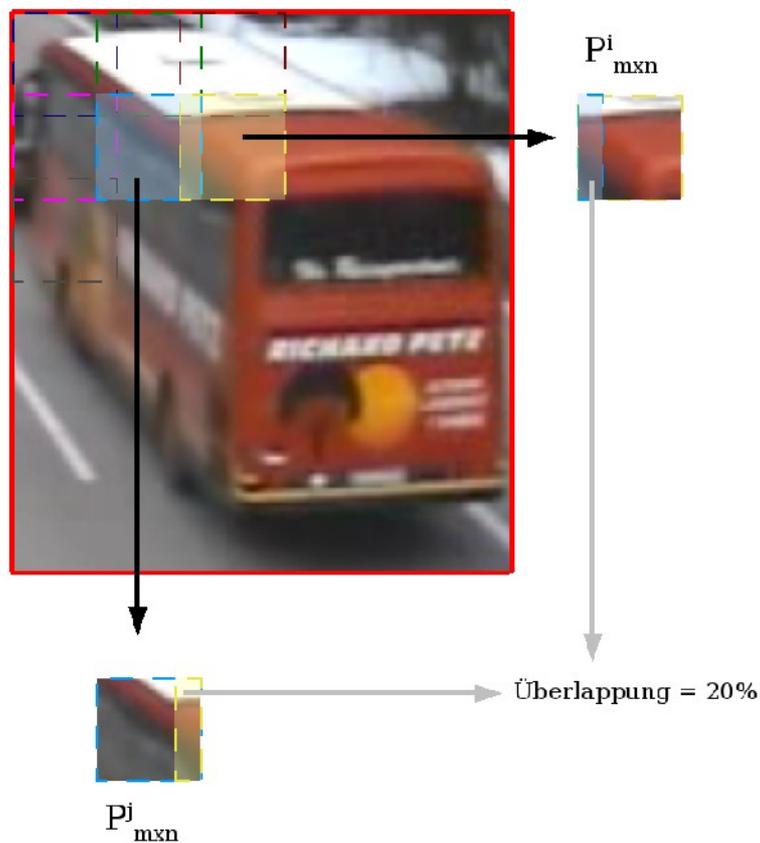


Abbildung 22 - Zerlegung eines (vergrößerten) Bildes in Patches

Als erster Schritt wird für ein Eingangsbild die Zerlegung $P^{i,j}$ durchgeführt. Für alle Teilbilder wird anschließend ein Subsampling mit demselben Verfahren wie in der Objektselektion beschrieben,

durchgeführt. Damit wird die Datenmenge für die Featuregenerierung reduziert. Danach werden ungünstige Kandidaten aus den Patches aussortiert. Dies soll verhindern, dass der - gemessen an den Rechenoperationen – teure Schritt der Featuregenerierung für Patches durchgeführt wird, die von vornherein keine interessanten Features liefern können. Daher werden nur Patches akzeptiert, deren Anteil an *Objektpixeln* über einem Mindestbetrag liegt. Dazu wird die Anzahl an gesetzten Pixeln im Maskenbild mit der Gesamtanzahl der Pixel in Relation gesetzt. Dies betrifft Patches, die hauptsächlich Maskenpixel beinhalten, ebenso wie Patches am Rand des Bildes, die insgesamt eine zu geringe Anzahl an Pixeln besitzen.

Erfüllt ein Patch dieses Kriterien, so werden über den Mean Shift Algorithmus die Moden bestimmt. Wie auch bei der Objektselektion, sind auch hier nur die Moden mit einem Mindestsupport interessant.

Wenn nun $m_i, i=1\dots l, l \geq 3$ die l ermittelten Moden eines beliebigen Patches sind (uni- und bimodale Patches werden ignoriert), so werden aus diesen Moden Features im vereinten 9-dimensionalen Raum gebildet, indem alle Kombinationen aus dreidimensionalen Moden jeweils konkateniert werden:

$$f_k = (m_i | m_j | m_k) \forall i, j, k < l, i \neq j \neq k$$

Die Anzahl der gebildeten Mode Kombinationen pro Patch ist somit

$$\frac{l!}{(l-3)!3!} \cdot$$

Während des Berechnungsschrittes werden die Modekombinationen zunächst alle in derselben Menge abgelegt. Da global über das Bild immer wieder ähnliche Kombinationen auftreten, die auf die Eindeutigkeit der Repräsentation jedoch kaum Einfluss haben, wird eine

Featureselection durchgeführt, um ähnliche Kombinationen zu verwerfen. Die Ähnlichkeit zwischen Features wird über eine spezielle Distanzfunktion berechnet, die berücksichtigt, dass die Repräsentation eines Feature f_k nicht eindeutig ist: durch die nicht-spezifische Reihenfolge beim Bilden der Kombinationen gilt nämlich

$$f_k = (m_i | m_j | m_k) = (m_i | m_k | m_j) = (m_k | m_i | m_j) = \dots$$

wodurch sich für ein Feature in Summe immer jeweils 6 mögliche Repräsentationen (aus 3 Moden) ergeben.

Da die euklidische Distanz $d(f_i, f_j)$ zwischen 2 Features im *vereinten* 9-dimensionalen Raum nicht eindeutig ist, wird stattdessen folgende Distanzfunktion eingeführt:

$$d(f_i, f_j) = d((m_r | m_s | m_t), (m_u | m_v | m_w)) = \min(\{d_k\})$$

wobei d_k die Menge der 6 möglichen Distanzen zwischen f_i und allen Repräsentationen von f_j ist:

$$d_1 = \|m_r - m_u\| + \|m_s - m_v\| + \|m_t - m_w\|$$

$$d_2 = \|m_r - m_u\| + \|m_s - m_w\| + \|m_t - m_v\|$$

$$d_3 = \|m_r - m_w\| + \|m_s - m_v\| + \|m_t - m_u\|$$

$$d_4 = \|m_r - m_w\| + \|m_s - m_u\| + \|m_t - m_v\|$$

$$d_5 = \|m_r - m_v\| + \|m_s - m_u\| + \|m_t - m_w\|$$

$$d_6 = \|m_r - m_v\| + \|m_s - m_w\| + \|m_t - m_u\|$$

Die so ermittelten Features stellen die „Signatur“ des Objektes dar, also eine komprimierte Kodierung seiner farblichen Zusammensetzung. Die Abbildung 23 zeigt eine typische Signatur und das dazu gehörende

Objekt. Im folgenden Abschnitt wird erklärt, warum und wie auch noch invariante Features gewonnen werden können.



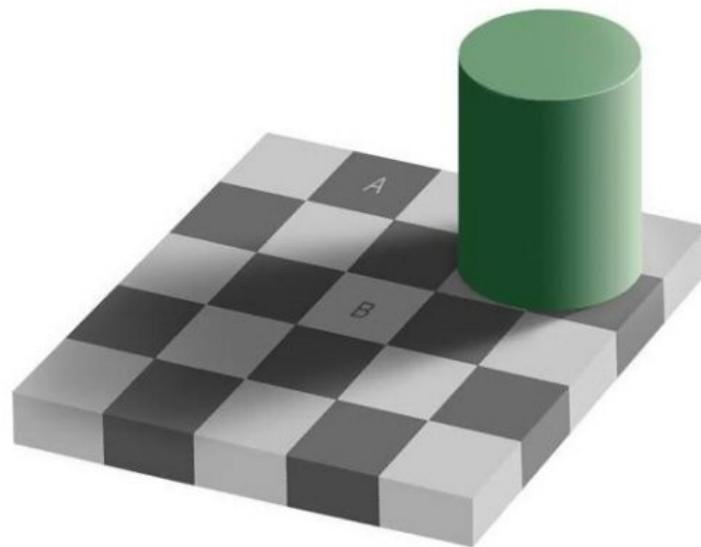
Abbildung 23 - Signatur eines Farbbildes (nur wesentliche Features nach dem Selektionsschritt)

Invariante Features

Das große und für den allgemeinen Fall weitgehend ungelöste Problem der farbbasierten Verfahren ist die so genannte „color constancy“. Damit wird die Fähigkeit des menschlichen Sehapparates bezeichnet, die Farbe eines Objektes auch bei einer Beleuchtungsänderung als die ursprünglich vorhandene wahr zu nehmen (vgl. Abbildung 24). Das Ziel robuster Verfahren muss es natürlich sein, zumindest geringe Beleuchtungsschwankungen zu tolerieren. Dennoch kann hier immer nur ein Kompromiss zwischen Diskriminanz und Invarianz erreicht werden.

Viele Ansätze wurden hierzu veröffentlicht, die im Wesentlichen Normalisierungen (meist über die rechnerische Modellierung und dann

Umkehrung des Beleuchtungseinflusses, z.B. **Comprehensive Color Normalization** wie bei [41]), Farbraumtransformationen (RGB nach HSV) und Berechnung invarianter Eigenschaften (z.B. **Cross Ratios** aus den Farbkanälen [42],[43]) aus den Merkmalen umfassen; der letzte Ansatz geht auf Untersuchungen von Kimmel und Jayanta zurück, wonach die Spektralkanäle korreliert und Cross Ratios der Farbbänder für kleine Umgebungen glatt sind.



*Abbildung 24: Beispiel für Farbkonstanz. Das menschliche Auge läßt alle hellen Quadrate gleich erscheinen, obwohl das Quadrat B durch den Schatten den gleichen Grauton wie das (dunkle) Quadrat A erhält !
(© Adrian Pingstone)*

Nach den Schritten im vorigen Kapitel können aus den vorliegenden Modekombinationen Features mit invarianten Eigenschaften berechnet werden. Je nachdem, welche Eigenschaften über die Beleuchtung und Geometrie zwischen Patches gelten bzw. bekannt sind, sind verschiedene Zusammenhänge zwischen den Farbkanälen invariant bei Beleuchtungs- und/oder Geometrieänderungen. Versuche mit den ursprünglich bei Koubaroulis et al. [43] verwendeten Features verliefen leider wenig viel versprechend. Offensichtlich können die dort angeführten, physikalischen Modelle der Beleuchtungsänderung die Phänomene in den vorliegenden

Daten (u.a. Reflexionen) nicht ausreichend beschreiben. Anstatt nun diese Invarianz mithilfe von Farbverhältnissen im RGB-Raum zu suchen, wird hier der Ansatz verfolgt, die Moden in einen geeigneten (Farb-)Raum zu transformieren und dort invariante Features zu gewinnen. In dieser Arbeit wird der CIE-LUV-Raum verwendet, da dieser isotrope Eigenschaften aufweist und für die Berechnung euklidischer Abstände gut geeignet ist. Da der Einfluss der Helligkeitsschwankungen auf die Features minimiert werden soll, können die Komponenten der LUV-Tripel mit der allgemeinen Gewichtungsmatrix

$$W = w_{ij} = \begin{bmatrix} w_{11} & 0 & 0 \\ 0 & w_{22} & 0 \\ 0 & 0 & w_{33} \end{bmatrix}$$

multipliziert werden; für den hier verwendeten Fall ergibt sich damit:

$$\begin{pmatrix} \tilde{l} \\ \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{bmatrix} w & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} l \\ u \\ v \end{pmatrix} = \begin{pmatrix} l \cdot w \\ u \\ v \end{pmatrix}$$

wobei für w empirisch Werte zwischen 0.7 und 0.9 bestimmt wurden. Features werden damit stärker durch die Chrominanz u und v als durch die Luminanz l bestimmt. Im experimentellen Teil dieser Arbeit (s. Kapitel 5) wird schließlich auch der Einfluss verschieden großer Werte des Parameters w untersucht.

4.5 Feature- und Objektmatching

Für das Matching ist ein Verfahren notwendig, welches für je 2 Mengen an Merkmalsvektoren (welche von diesem Zeitpunkt an die Objekte ausschließlich repräsentieren) ein Ähnlichkeitsmaß berechnet. Hier fiel

die Entscheidung auf das Prinzip der „Stable Marriage“, wie es auch in [38] vorgeschlagen wird. Damit wird das Matching als Zuordnungsproblem definiert, welches für jedes Feature eines Sets das am Besten passende Feature des anderen Sets zu finden versucht. Aus den Abständen der einander zugeordneten Merkmale wird ein Maß für die Übereinstimmung berechnet; die Anzahl der nicht zu matchenden Merkmale wird in diesen Score miteinbezogen.

Nachdem die Merkmale der Bildes extrahiert sind, werden diese mit den Merkmalen der anderen Bilder gematcht. Das Matching bedeutet im vorliegenden Fall auf die Lösung der Frage: Gegeben die Signaturen zweier Eingangsbilder I_1, I_2 ; mit welcher Wahrscheinlichkeit beschreiben diese Featuresets dasselbe Objekt ?

Zur Lösung dieser Frage wird ein robustes Matchingverfahren angewendet, das in 2 Schritten durchgeführt wird:

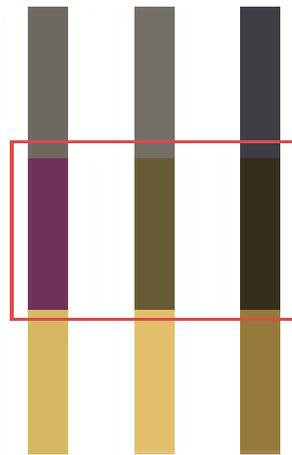
1. Vergleichen von Zeitstempel und Fahrzeugtyp der zu matchenden zweier Objekte. Daraus ergibt sich eine Vorentscheidung, ob diese Fahrzeuge überhaupt zu matchen sind
2. Ein auf dem Stable Marriage Algorithmus ([44][40]) basierendes Matching der nach (1) übrig geblieben Kandidaten.

Die Zeitstempel der Fahrzeuge müssen dabei realistische Werte ($< T_t$) aufweisen, sodass das Fahrzeug die Strecke zwischen den Kameras überhaupt in dieser Zeit zurücklegen kann.

Der Stable Marriage Algorithmus betrachtet die Frage des Matchens zweier Featuresets als ein Zuordnungsproblem: jeweils die Features aus den beiden Sets mit dem geringsten Abstand zueinander werden einander zugeordnet und aus der Distanzmatrix „entfernt“. Diese Distanzen werden zum Score addiert; für Distanzen, die über dem Schwellwert liegen (und somit nicht gemacht werden), wird statt der Distanz der Schwellwert addiert. Schließlich wird der Score noch auf das größere

Featureset normiert, sodass sich für den Score immer ein Wert zwischen 0 und 1 ergibt. Dabei bedeutet 0 eine perfekte Übereinstimmung, und 1 eine vollkommene Unähnlichkeit der Featuresets. (Offensichtlich kann der Extremwert 0 für den optimalen Match nur bei symmetrischen Verhältnissen erreicht werden, also wenn beide Featuresets gleich groß sind). Als Distanzfunktion wird dieselbe Funktion wie zur Featureselektion verwendet (s. Seite 63)

Aus Gründen der Robustheit wird eine Einschränkung für die Berechnung der Distanzfunktion eingeführt. Wie auf Seite 63 gezeigt, wird die Distanz zweier Features über die Summe der Distanzen der einzelnen Moden gebildet. Um zu gewährleisten, dass **alle** 3 Moden eines zweier Features ähnlich sind, wird ein zusätzlicher Schwellwert eingeführt, sodass jede Einzeldistanz kleiner als dieser Schwellwert sein muss. Damit wird verhindert, dass zwei Features matchen können, obwohl jeweils einer der Moden jedes Features eine deutlich unterschiedliche Farbe zu allen Moden des anderen Features hat (s. Abbildung 25).



*Abbildung 25: Das mittlere Feature wird gegen das linke und rechte Feature gematcht. Während beide Features dieselbe Gesamtdistanz zum mittleren Feature haben, matcht das linke Feature aufgrund der zu großen Einzeldistanz (vgl. mittleren Mode) **nicht**.*

Folgender Algorithmus wird verwendet (für die Implementierung des robusten Featurematching siehe [45]):

Seien F_1 und F_2 zwei (i.A. asymmetrische) Featuresets mit jeweils m, n Features:

1. Wenn die Differenz der Zeitstempel von F_1 und F_2 kleiner T_t oder die Fahrzeugtypen unterschiedlich sind: kein match dieser Featuresets

2. Bilde die Distanzmatrix $D_{m \times n}$ mit $d_{ij} = \text{dist}(F_1(i), F_2(j))$

3. Bis zur Konvergenz (keine Distanz $d_{ij} < \text{Threshold } T$):

- Bilde $C = \{c_{ij}\}$, mit $c_{ij} = \begin{cases} 1 & \text{für } d_{ij} < d_{ik} \forall k \neq j \\ 0 & \text{sonst} \end{cases}$

, die *forward matching Matrix*

- Bilde $B = \{b_{ij}\}$, mit $b_{ij} = \begin{cases} 1 & \text{für } d_{ij} < d_{kj} \forall k \neq i \\ 0 & \text{sonst} \end{cases}$

, die *backward matching Matrix*

- Suche die Elemente $d_{ij} \in D$, für die $c_{ij} = 1$ und $b_{ij} = 1$

- Addiere die Distanzen d_{ij} zum score

- Entferne die Distanzen d_{ij} aus der Distanzmatrix D

4. $score = score + \sum_{i=1}^{\max(m, n) - \text{nscores}} T$ wobei nscores die Gesamtanzahl

der in (3) gefundenen d_{ij} ist.

5. $score = \frac{score}{\max(m, n) * T}$

Schema 2 - Robustes Feature Matching

Das oben beschriebene Matching liefert also ein Maß für die Ähnlichkeit zweier Featuresets. Im allgemeinen ist es der Fall, dass nicht die Featuresets von nur 2 Objekten, sondern von 2 *Mengen von Objekten* gematcht werden. In der konkret vorliegenden Problemstellung ist zusätzlich davon auszugehen, dass nicht alle Objekte gematcht werden können; da die Strecke zwischen den Kamera Standorten nicht quellen- und senkenfrei ist (z.B. Raststationen, Autobahnauffahrten und -abfahrten), und die MVOD nicht ideal ist, muss man immer mit einer Anzahl zu verwerfender Features rechnen (für den Fall dass ein Fahrzeug nicht auf beiden Kameras beobachtet wurde).

Betrachtet man nun die beiden *Objektmengen* S_1 und S_2 mit k bzw. l Elementen, so ergeben sich durch Berechnung des Score für alle Objekte der zwei Mengen $l \cdot k$ Scores, die in der Scorematrix $S_{l \times k}$ zusammengefasst werden. Um die gültigen Matches aus S zu bestimmen, wird ein greedy-selection Algorithmus durchgeführt: Jeweils der beste Score wird aus der Matrix entfernt und als Match gewertet, solange ein Threshold nicht überschritten ist. Daraus resultiert ein Vektor von Objektpaaren, die gemäß dem Algorithmus einander zugeordnet werden.

Da bedingt durch die Struktur des Verfahrens, Farbübergänge die Ähnlichkeitsfunktion zwischen Objekten bestimmen, kann es vorkommen, dass Objekte, deren Farbverteilung (aus der Sicht eines eines Histogramms) sehr unterschiedlich ist (die wesentlichen Farbcluster, also die dominanten Bildfarben, sind nicht übereinstimmend), aber in kleinen Teilstrukturen ähnliche Farbübergänge aufweisen, dennoch matchen. Um diesen Fall abzufangen, der zu einer erhöhten Anzahl an false positives führt, wird ein zusätzlicher Verifikationsschritt eingeführt, der nach dem Matching auf die potentiellen korrespondierenden Objekte angewendet wird:

- (1) es werden die N stärksten Moden (bezogen auf den Support) aus der Stufe der Objektselektion für jedes Objekt herangezogen
- (2) Aus diesen werden – wie schon für die eigentlichen Features beschreiben - invariante Features berechnet.
- (3) Diese Features werden miteinander gematcht, und der erzielte Score muss über einem Threshold liegen; sonst wird der Match abgelehnt.

Mit dem letzten Schritt wird also ein globales Kriterium eingeführt, mit dem „offensichtliche“ falsche false positives reduziert werden können; eine Evaluierung dieses Schrittes erfolgt im nächsten Kapitel.

4.6 Zusammenfassung

In diesem Kapitel wurde eine Methode zum robusten Matchen von zwei Bildmengen vorgestellt. Dazu wurden zuerst Kriterien und ein Algorithmus zur Auswahl geeigneter Objekte definiert, die die Stabilität des anschließenden Matching erhöhen. Für das Matching selbst wurde die Extraktion von farbbasierten Merkmalen über die Multimodal Neighborhood Signature beschrieben, sowie die Berechnung von invarianten Eigenschaften. Das Matchen zweier Bildmengen anhand dieser Eigenschaften und einfacher Constraints wurde schließlich mit dem Stable Marriage Verfahren gezeigt. Das Verfahren für das Matching wurde schließlich mit einfachen Constraints wie Zeitstempel und Fahrzeugtyp stabilisiert.

5 Evaluierung

5.1 Allgemeines

In diesem Kapitel wird das im vorigen Kapitel beschriebene Verfahren zur Objekterkennung evaluiert. Es werden 3 verschiedene Datensätze verwendet, und die Performance des Verfahrens bei diesen Datensätzen analysiert. Die wesentlichen Parameter werden auf ihren Einfluss auf die Performance des Verfahrens untersucht, wobei vor allem dem Einfluss vor allem der ersten Stufe – der Objektselektion – besonderes Augenmerk gilt.

Testdaten

Für die generelle Einschätzung der Performance unter „normalen“ Bedingungen werden zwei größere Hauptsets mit jeweils ca. 400 bzw. 500 Bildern von Fahrzeugen untersucht:

- Das erste Set (554 vs. 461 Fahrzeuge) enthält nur relativ wenige Objekte mit guten Features, sodass sich hier vor allem der Vorteil der Objektselektion zeigt, da die Anzahl zu vergleichender Fahrzeuge sehr stark reduziert wird (Abbildung 26).
- Das zweite Set (546 vs. 390 Fahrzeuge) enthält eine weitaus größere Anzahl an geeigneten Objekten. Es sind auch viele Objekte mit ähnlichen Features enthalten, sodass vor allem die diskriminativen Eigenschaften des Algorithmus gefordert sind (Abbildung 27).
- Ein weiteres kleineres Set (111 bzw. 160 Fahrzeuge) beinhaltet Aufnahmen, bei denen das Bild der zweiten Kamera durch Regen stark beeinträchtigt ist, und ermöglicht die Analyse der Robustheit unter erschwerten Verarbeitungsbedingungen (Abbildung 28).

Die Testdaten wurden dabei so gewählt, dass nur ein Bruchteil aller tatsächlich in den Videos vorkommenden Fahrzeuge ausgeschnitten wurde. Von den Fahrzeugen, die ohnehin aufgrund ihrer Größe in der späteren Objektselektion heraus gefiltert würden, wurde daher nur ein sehr kleiner Teil verwendet. Von den Fahrzeugen mit geeigneter Größe wurden auch jene verwendet, die „offensichtlich“ schlechte bis gar keine Eignung zur Weiterverarbeitung haben, um die Wirkung der Objektselektion zu bewerten.



Abbildung 26: Das Set 2 enthält einige Objekte mit guten Features, und auch die Beleuchtungsänderung zwischen den Aufnahmen ist akzeptabel.



Abbildung 27: Die Fahrzeuge aus dem Set 2 haben zum Teil sehr ähnliche Features (hier z.B. rot-gelb Übergänge). Außerdem sind oft nur wenige starke Features vorhanden (aufgrund schwacher Texturierung ergeben sich wenige markante Farbübergänge.)



Abbildung 28: Die Bilder der zweiten Kamera (unten) aus dem Set 4 sind durch Regen verwaschen und damit qualitativ viel schlechter als die von Kamera 1 (oben).

Teststrategie

Da kein vollständiges „Livesystem“ zur Verfügung steht, erfolgt die Objekterkennung offline mit einer MatLAB Implementierung. Dazu werden von den Videostreams der beiden Kameras einzelne Fahrzeuge ausgeschnitten und als Bitmap abgespeichert. Der Hintergrund der Bilder wurde mit einer speziellen Farbe maskiert. Diese Maskenfarbe ist so gewählt, dass sie nicht in den Aufnahmen vorkommt; beim Subsampling der Bildausschnitte werden explizit nur Nicht-Maskenpixel selektiert. Dadurch werden Features aus dem Hintergrund nicht dem Objekt zugeordnet und in weiterer Folge auch nicht verglichen.

Zu jedem Bild wird der Zeitstempel (Präzision: 1 Sekunde) mit gespeichert. Als weiteres Feature wird – von Hand – jedes Fahrzeug einer der 4 Klassen LKW, PKW, BUS und „SONSTIGE“ zugeordnet. Dies soll vor allem die Robustheit und Effizienz verbessern und wird noch in einem eigenen Abschnitt getestet.

Abgeleitete Vergleichszahlen

Als *total true positives* wird die Anzahl der in zwei Sets enthaltenen möglichen Matches definiert. In den Tabellen werden meist statt der

absoluten Werte für die Anzahl der gefundenen, korrekten Matches (*true positives*), der gefundenen, aber nicht korrekten Matches (*false positives*) und der durch den Verifikationsschritt ausgefilterten *false positives* (*rejected positives*) zwei aussagekräftige Relationen angegeben: α als der Quotient aus **true positives / total true positives** und β als der Quotient aus **true positives / false positives**.

α gibt dabei an, wie viele der möglichen Matches tatsächlich gefunden werden. Je höher dieser Wert, umso mehr Fahrzeuge und damit Zeitwerte können für die Geschwindigkeitsschätzung herangezogen werden, und umso präziser (und dynamischer) ist der resultierende Schätzwert.

Der Parameter β beschreibt die Verfälschung des Resultates durch den Anteil von falschen Matches und damit Zeitwerten.

Ziel des Verfahrens ist es, möglichst hohe Werte für beide Parameter zu erreichen, damit Dynamik und Zuverlässigkeit gleichzeitig maximiert werden. Die Erfahrung zeigt jedoch, dass i.A. nur ein Kompromiss erreicht werden kann [24] .

Exemplarisch werden auch für die mit den Standardparametern durchgeführten Matchings die Geschwindigkeitsschätzungen, resultierend aus den Zeitstempeln der Objektbilder und der bekannten Kameradistanz, in Diagrammform präsentiert und daraus das Diagramm mit den Durchschnittsgeschwindigkeiten ermittelt. Diese werden mit den Geschwindigkeitsverläufen aus der Ground Truth verglichen.

5.2 Wesentliche Parameter des Verfahrens

Im folgenden wird beschrieben, welche Parameter die Stufen des Verfahrens im wesentlichen beeinflussen; ihr Einfluss soll durch die Testdaten in den weiteren Experimenten evaluiert werden:

1. Objektselektion

Hier bestimmen vor allem die Einstellungen **Mindestbildgröße** und der **Mindestscore**, welche Objekte weiter verwendet werden. Der Score wird wiederum über Parameter wie **Buntheit** (also Abstand der Moden von der Unbuntachse) und den Support der Moden bestimmt. Als wichtiger Faktor geht hier vor allem der **Radius** (=Bandbreite) des Mean Shift ein. Die Vermeidung zu dunkler oder zu heller Farben trägt im Gegenzug nur wenig zur Stabilität bei und wird hier nicht weiter untersucht. Es wird schließlich noch untersucht, inwiefern die Buntheit (so wie sie in dieser Arbeit ermittelt wird) mit der Größe der Objekte korreliert.

2. Featuregenerierung

Die wesentliche Einflussfaktor auf die Signatur ist in erster Linie natürlich der **Radius** des Mean Shift. Zusätzlich wirken sich aber noch die Größe der Kachelung der MNS-Patches sowie der **Pruning** Parameter (Mindestabstand der endgültigen Features), und in besonderem Maße auch die Berechnung helligkeitsinvarianter Features aus den Modekombinationen aus. Hier ist entscheidend, wie stark die Helligkeitsinformation unterdrückt wird, um das geeignete Mass zwischen Invarianz und Diskriminanz zu erreichen.

3. Featurematching

Wichtig ist der **Threshold** der Distanzfunktion zwischen Features, mit dem die erlaubte Änderung der Farbinformation zwischen den Aufnahmen quantifiziert wird. Zudem ist interessant, wie sehr die Erkennung durch die Zuhilfenahme von **Zeitstempel** und **Fahrzeugtyp** stabilisiert wird. Schließlich wird noch die Wirksamkeit des **Verifikationsschrittes** analysiert, der false positives reduzieren soll.

5.3 Einführendes Beispiel zur Illustration

Zur Illustration werden die Ergebnisse des Matchings für eine repräsentative Teilmenge aus Set 1 gezeigt. Von den beiden Kameras sind 555 bzw. 462 Bilder vorhanden (Abbildungen 29 und 30 zeigen die verwendeten Teilmengen aus den Sets, jeweils aus den Aufnahmen einer Kamera). Die verwendeten Parameter für die jeweiligen Ergebnisse sind nahe bei den optimalen Werten, die Ergebnisse variieren jedoch nur wenig bei kleinen Parameter-Variationen. In späteren Versuchen wird untersucht, inwieweit die Wahl der Parameter die Resultate des Algorithmus beeinflusst. Am Ende dieses Kapitels werden auch noch die Ergebnisse für „optimale“ Parameter angegeben.

Von den Streams von Set 1 wurden 44 bzw. 48 Fahrzeugen entnommen (Abbildungen 29 und 30). Für diese wurden Objektselektion und Matching durchgeführt, und die selektierten (Abbildungen 31 und 32) und gematchten Fahrzeuge (Abbildung 33) gezeigt. In diesem Fall zeigt einerseits die Selektion das erwartete Verhalten (Auswahl der besser geeigneten Objekte, die Eingangs-Datenmenge wird stark reduziert von 44 auf 16 bzw. von 48 auf 19 Fahrzeuge).

Andererseits liefert auch das Matching sehr gute Werte (von den 12 existierenden Matches in der Selektion werden 11 korrekt erkannt, somit ein sehr guter Wert von $\alpha = \beta = 91.67 \%$ erzielt).



Abbildung 29: Beispielbilder aus dem Set1. Die Maskenfarbe ist mit $[255,0,255]$ gewählt, eine Farbe, die -aufgrund des Sättigungsgrades- in den Aufnahmen nicht vorkommt. Man sieht, dass im Set sowohl deutlich texturierte Fahrzeuge, als auch solche mit eher homogenen Farbverläufen bzw. ohne deutliche Farbstruktur enthalten sind.

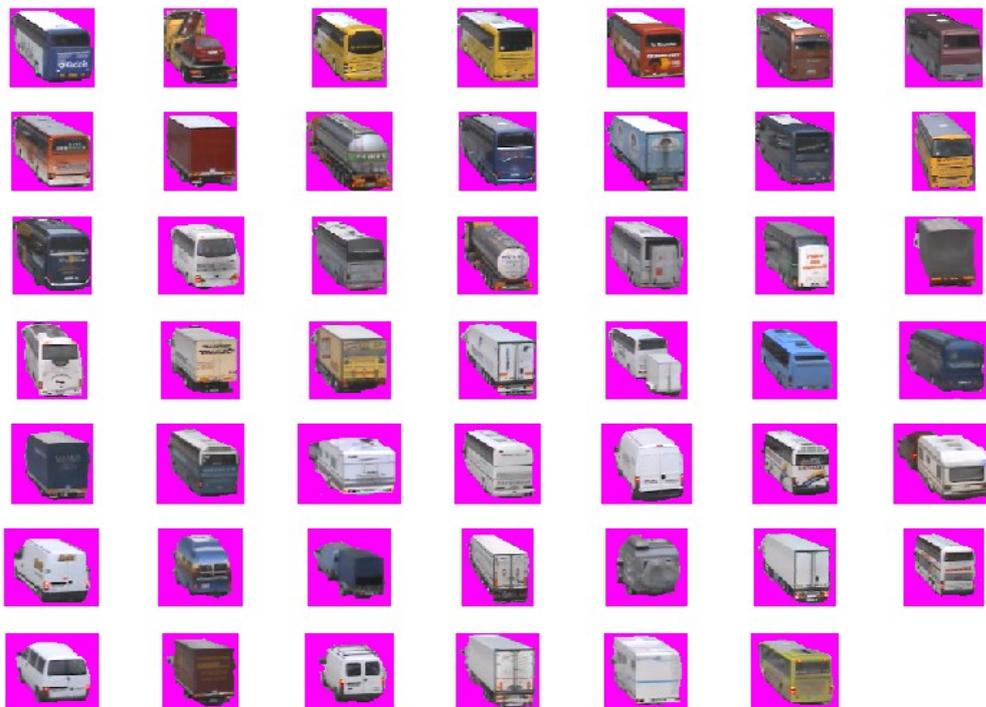


Abbildung 30: Beispielbilder aus dem Set2. Einige Fahrzeuge aus Set 1 (s. Abbildung oben) sind auch in diesem Set enthalten ($total_true_positives = 12$)



Abbildung 31: Die ausgewählten Kandidaten aus der den Fahrzeugen aus Abbildung 29. Diese Objekte genügen den Anforderungen an Größe und Farbigkeit.



Abbildung 32: Selektierte Kandidaten aus den Fahrzeugen aus Abbildung 30

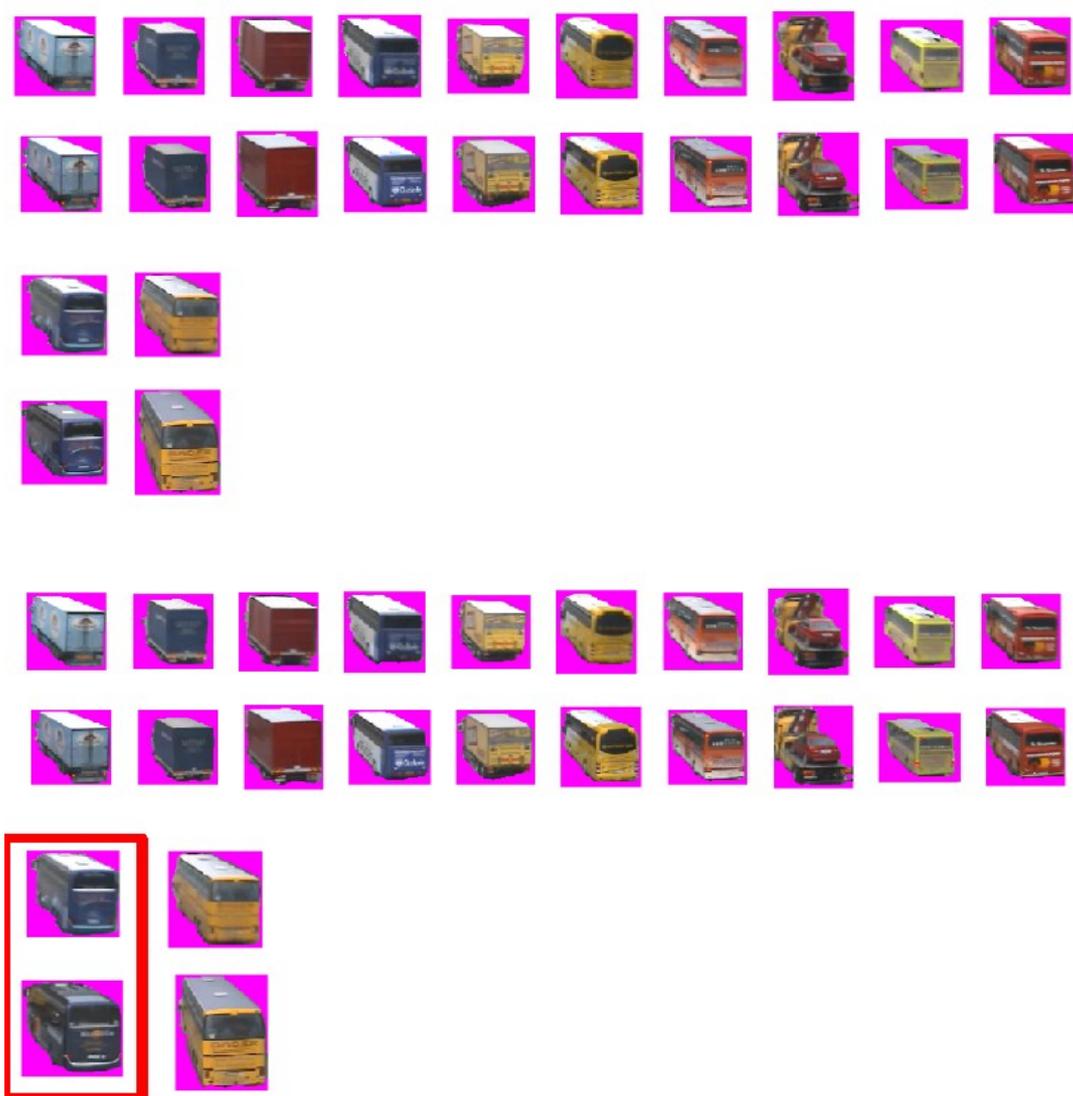


Abbildung 33: Oben: Alle möglichen Matches aus den selektierten Objekten. Es gibt 12 korrespondierende Fahrzeuge, 11 davon wurden korrekt erkannt (unten). Die rot umrandeten Fahrzeuge wurden nicht richtig gematcht, allerdings wurde die falsche Zuordnung im Verifikationsschritt erkannt und als false positive gewertet.

5.4 Untersuchung der Featureselektions-Stufe

Variation von Mindest-Bildgröße und Mindest-Score

Die Objektselektion hat große Auswirkungen auf die Laufzeit der nachfolgenden Stufen, aber sie beeinflusst auch den (möglichen) Fehler der Erkennungsstufe. Die Beobachtung zeigt, dass eben nur wenige Fahrzeuge gut wiedererkannt werden können (selbst vom menschlichen Betrachter), weil sie über aussagekräftige Features verfügen. Kleine tatsächlich bestehende Unterschiede sind durch die immanente Unschärfe (Bewegung, Aufnahmequalität) sowie Bildgröße nicht mehr erkennbar. Daher ist es auch für das vorliegende Verfahren nicht möglich, diese Objekte (ausreichend) eindeutig zuzuordnen, und der erwartete Fehler entsprechend groß. (Aus diesem Grund ist auch die Erstellung der Ground Truth für diese Fahrzeuge nicht vollständig möglich, weil auch ein manueller Matching-Versuch scheitert, wie die Abbildung 34 zeigt!)

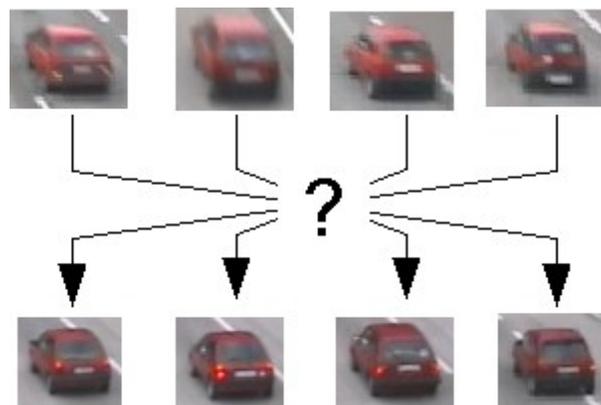


Abbildung 34: Welche Fahrzeuge aus den Bildern der ersten Kamera jenen auf den Bildern der zweiten Kamera entsprechen, ist - insbesondere für PWK - oft nicht entscheidbar.

Die sinnvolle Mindestgröße der Bilder ist stark abhängig von den speziellen Abbildungseigenschaften und Montage der Kameras, sodass dieser Wert – für jede Kamera - experimentell bestimmt werden muss. In den Abbildungen 35-37 ist die Abhängigkeit der Erkennungsrate von der Fahrzeuggröße dargestellt. Die starken Schwankungen am rechten Skalenende stammen daher, dass dort nur mehr sehr wenige Fahrzeugkandidaten übrig sind. Generell zeigt sich über weite Bereiche keine nennenswerte Sensitivität des Matchings auf die Bildgröße (blaue Linie = α , rote Linie = β); Abbildung 42 zeigt, dass bei ausschließlicher Verwendung der Bildgröße zu viele ungünstige Objekte übrig bleiben, die insbesondere die Performance von β senken. Den typischen Verlauf von true positives und false positives sowie der zurückgewiesenen Matches in absoluten Zahlen zeigt Abbildung 44. Man erkennt, dass sich bei Erhöhung des Mindestscore die Erkennungsrate deutlich mit-erhöht. Das beste Resultat ergibt sich jedoch, wenn beide Filter kombiniert werden. Der Filter für die Farbstruktur eliminiert dann von den großen Fahrzeugen (s. Abbildung 42) diejenigen, die kaum Features besitzen (Abbildung 43).

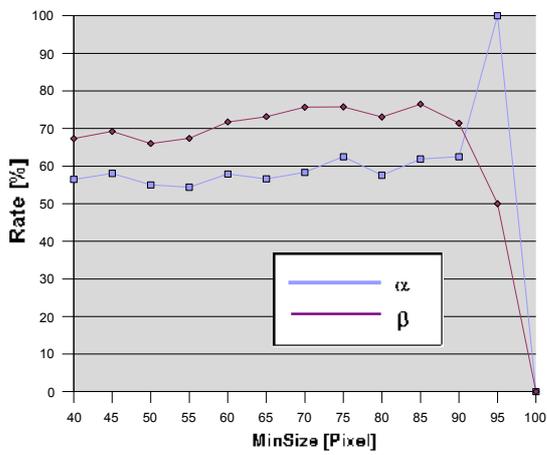


Abbildung 35: Abhängigkeit der Erkennungsrate von der Mindestbildgröße in Set 1.

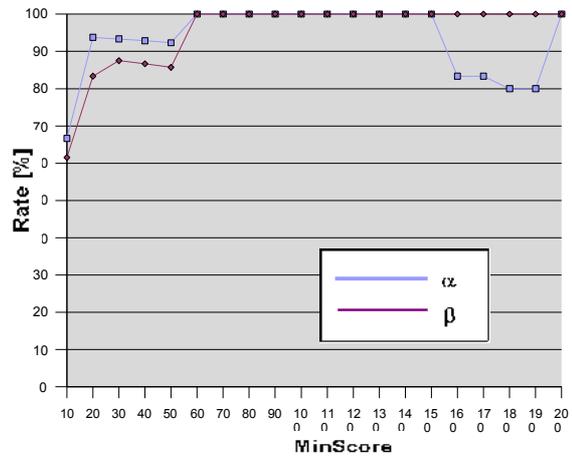


Abbildung 36: Abhängigkeit der Erkennungsrate von der Mindestscore in Set 1.

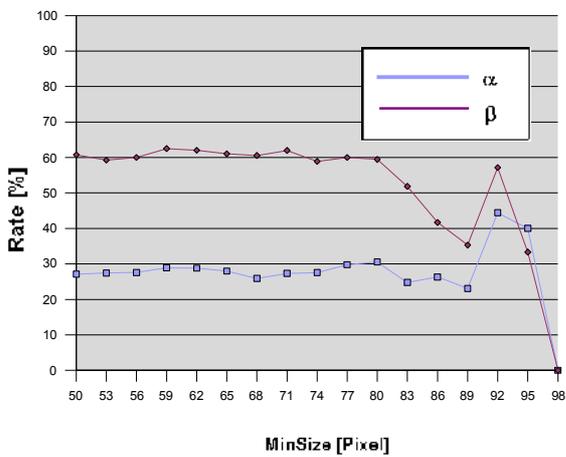


Abbildung 37: Abhängigkeit der Erkennungsrate von der Mindestbildgröße in Set 2

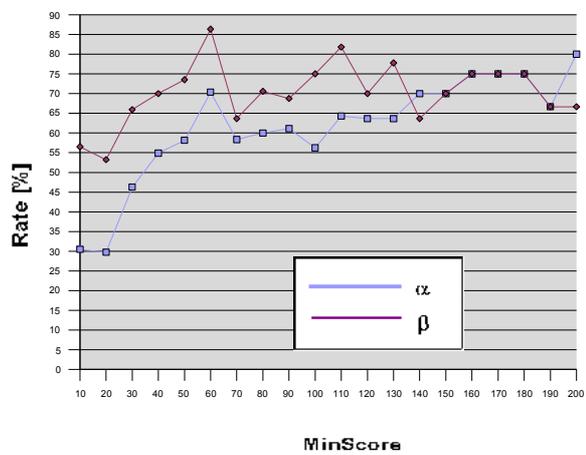


Abbildung 38: Abhängigkeit der Erkennungsrate von der Mindestscore in Set 2.

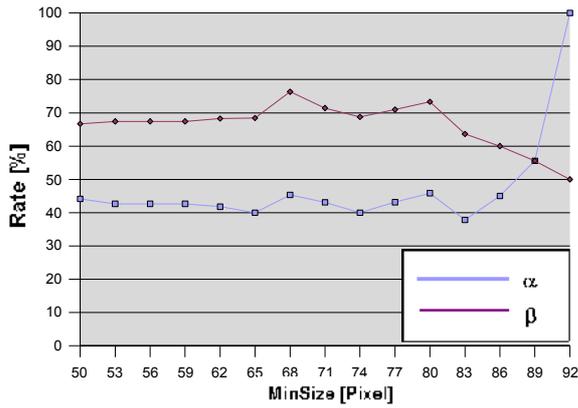


Abbildung 39: Abhängigkeit der Erkennungsrate von der Mindestbildgröße in Set 3

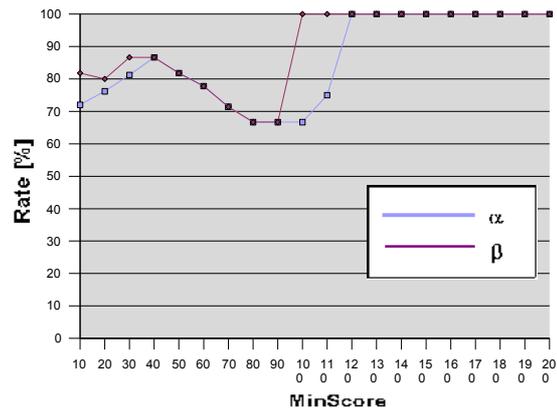


Abbildung 40: Abhängigkeit der Erkennungsrate von der Mindestscore in Set 3.

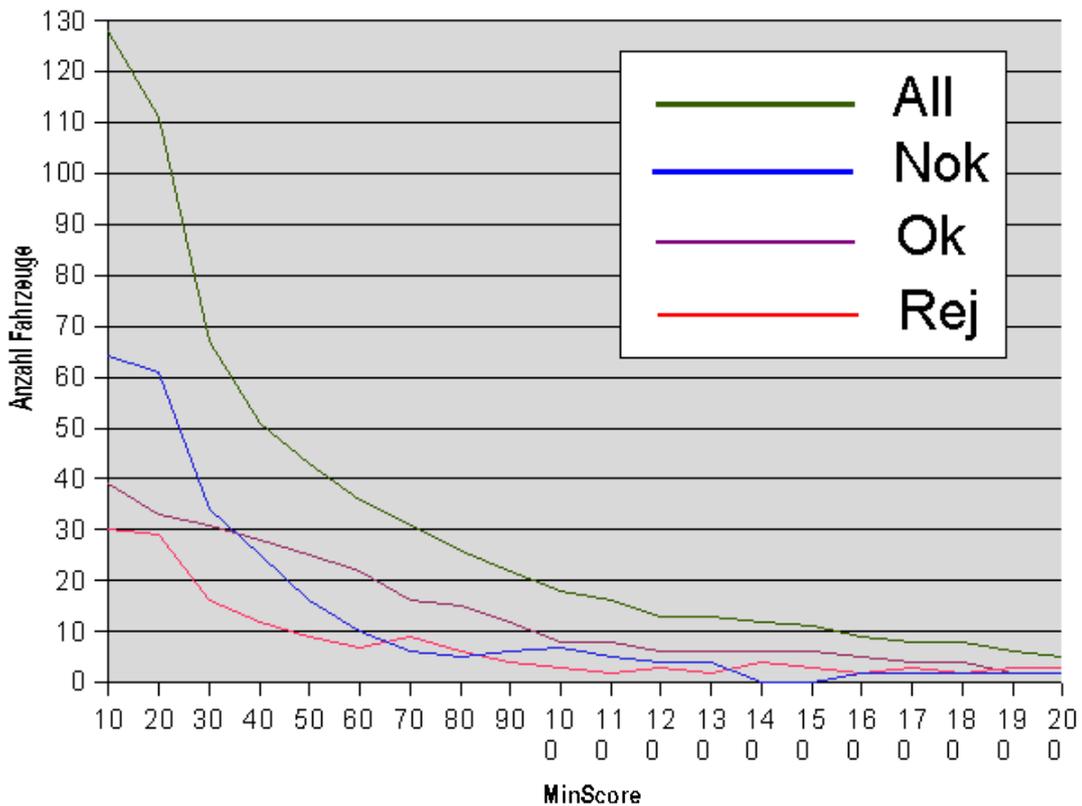


Abbildung 41: Typischer Verlauf der Erkennungsrate bei Variation des Mindestscore der Preselection Stufe am Beispiel von Set 2. Es werden hier bewusst statt den Verhältniszahlen α, β die absoluten Werte gezeigt, da diese hier besser den qualitativen Zusammenhang zeigen. Anmerkung: Für viele (kleine) Fahrzeuge kann hier keine Ground-Truth angegeben werden; für sehr kleine Scores ist daher der Wert von OK nicht sehr genau.



Abbildung 42: Wenn nur die Bildgröße als Filter für die Selektion herangezogen wird, bleiben zu viele ungeeignete Objekte über (hier viele weiße Fahrzeuge in beiden Sets, rot markiert). Die Fahrzeuge stellen die Kandidaten aus Set 1 bei Bildgröße 80x80 Pixel dar.

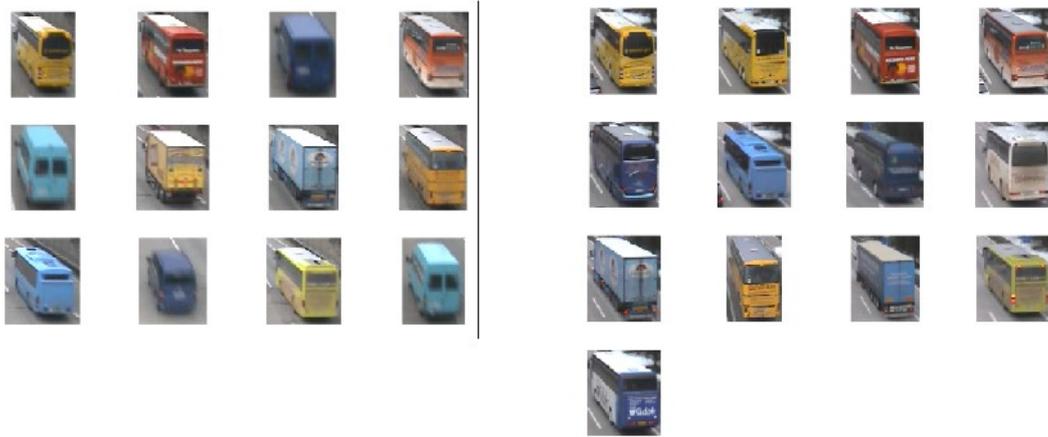


Abbildung 43: Die Kombination aus Größenselektion und Farbselektion ergibt effektiv fast ausschließlich Kandidaten mit den guten Features. Der „weiße“ Bus am rechten Set ist durch den Schmutz auf dem Fahrzeug in die Auswahl gekommen

Variation der Parameter der Farbvielfalt

Die Farbvielfalt wird über die Parameter des Mean Shift Verfahrens (Radius) sowie der nachfolgenden Selektion der Moden bestimmt.

Über den Radius kann beeinflusst werden, wie fein unterschiedliche Moden noch aufgelöst werden können; bei einem großen Radius wird ein Mode im Allgemeinen als Mittelwert einer größeren Punktmenge bestimmt, was zur Verschmelzung eng benachbarter Moden führen kann, die bei kleinerem Radius noch getrennt werden könnten. Für die Vorselektion ist aber das grobe Farbhistogramm eines Objekts von Interesse. Daher ist eine zu feine Clusterung (kleiner Radius) nicht zielführend, ganz abgesehen von der längeren Rechenzeit. Neben dem Radius wirkt sich auch der Mindestabstand aus, den die Moden von ihrem Grauwert haben, denn nur Farben über diesem Schwellwert tragen zum Score bei.

Abbildungen 45 bis zeigen die Abhängigkeit der Erkennungsrate von Radius bzw. Mindestabstand. Im ersten Set sind die Farbfeatures am stärksten, daher bewirkt eine Vergrößerung von Radius und Abstand, dass die Kandidaten mit den schwächeren Features wegfallen, bei den

Kandidaten mit guten Features aber immer noch ausreichend viele „bunte“ Moden mit genügend großem Abstand zueinander vorhanden sind. Bei Sets 2 und 3 zeigt sich, dass aufgrund vieler Fahrzeuge mit großer Ähnlichkeit ein zu „grobes“ Clustering ($R \gg$) viele benachbarte Farbmoden zusammenfasst, und damit nicht genügend Moden (für die Selektion müssen mindestens 3 Moden gefunden werden) übrig bleiben. Für den Abstand der Moden von der Unbuntachse gilt kein so ausgeprägter Zusammenhang. Bei Set 1 zeigt sich erwartungsgemäß ein ähnlicher Verlauf wie für den Radius, bei Set 2 und 3 ergibt sich jedoch ein anderes Bild: Zwar ist bei Berücksichtigung des Abstandes eine deutlich bessere Erkennungsrate zu erwarten sind als ohne, allerdings sinkt die Kurve nach einem kleinen Optimum bald wieder ab. Grund sind die nicht so ausgeprägten Features aus Set2 und Set3.

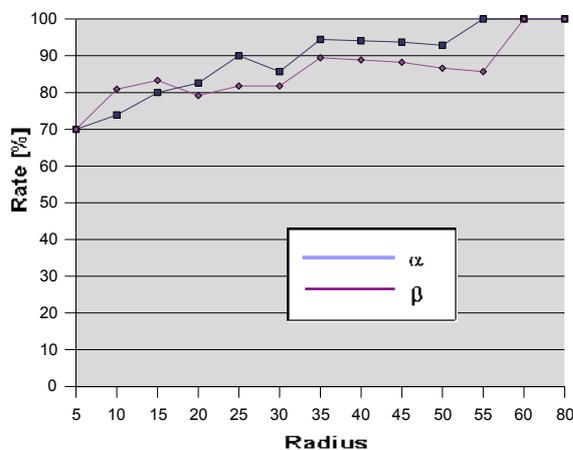


Abbildung 44: Abhängigkeit des Matchings vom Mean Shift Radius der Vorselektion bei Set 1

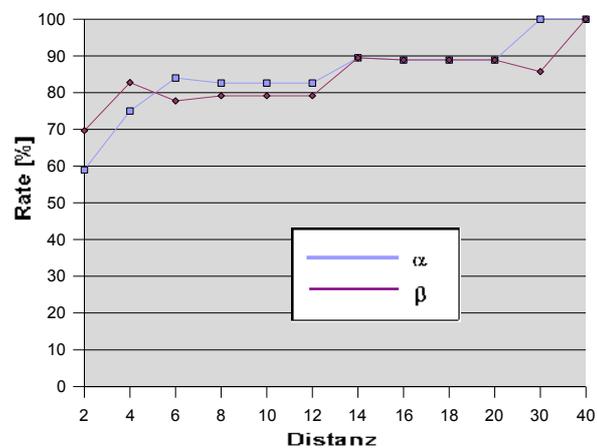


Abbildung 45: Abhängigkeit des Matchings vom Mindestabstand der Moden vom Grauwert in der Vorselektion bei Set 1

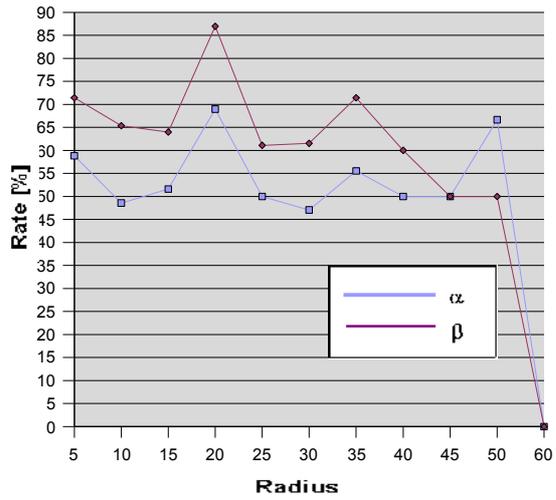


Abbildung 46: Abhängigkeit des Matchings vom Mean Shift Radius der Vorselektion bei Set 2

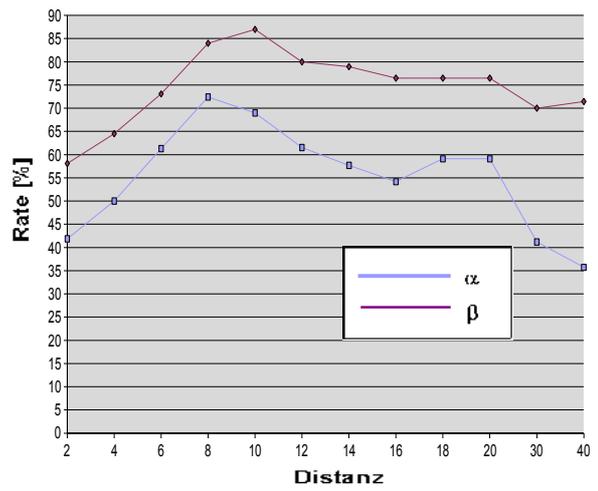


Abbildung 47: Abhängigkeit des Matchings vom Mindestabstand der Moden vom Grauwert in der Vorselektion bei Set 2

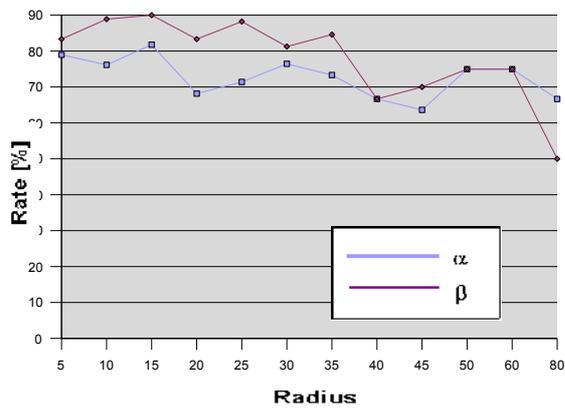


Abbildung 48: Abhängigkeit des Matchings vom Mean Shift Radius der Vorselektion bei Set 3

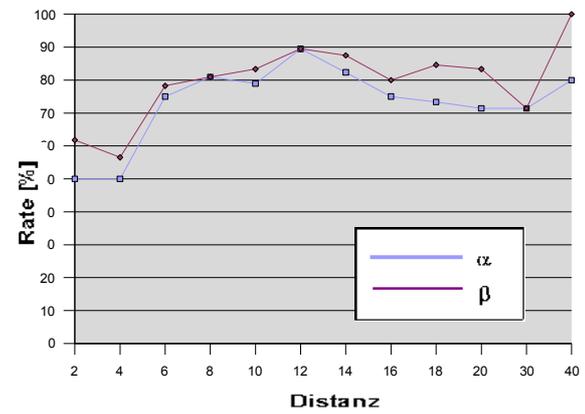


Abbildung 49: Abhängigkeit des Matchings vom Mindestabstand der Moden vom Grauwert in der Vorselektion bei Set 3

5.5 Untersuchung der Featuregenerierungs-Stufe

Kachelungsgröße

Wie schon im Kapitel 4.4 beschrieben, erfolgt als erster Schritt der MNS eine Kachelung des Eingangsbildes. Die Größe der Kacheln ist standardmäßig 16x16 Pixel und wird im Normalfall nicht verändert. Durch die Größe wird eine gewisse Lokalität und ein innerer Zusammenhang der Features definiert. Da nur die Moden **eines** Patch zu Farbnachbarschaften kombiniert werden, sollte die Größe des Patch in Zusammenhang mit der Größe der erwarteten Texturen stehen (daher auch der Name **neighbourhood**; die Farbregionen müssen nicht direkt aneinander liegen, sich jedoch in einer gewissen Nachbarschaft zueinander befinden.) Steigt die Größe der Patches an, wird dieses Lokalitätsprinzip zunehmend verletzt, außerdem entsteht das Problem der kombinatorischen Explosion, sodass sehr viele Modekombinationen betrachtet werden müssen. Bei sehr kleinen Patches werden möglicherweise komplexere Strukturen nicht mehr vollständig beschrieben, weil sie sich über mehrere Patches erstrecken; gerade diese sind meist aber die stärksten Features der Objekte. In den folgenden Experimenten werden nur diejenigen Objekte verwendet, die von der Objektselektion ausgewählt wurden.

	5	10	15	20	25	30
OK	13	14	15	13	15	14
NOK	3	2	2	3	2	2
REJ	2	2	2	2	2	2
α	81,25	87,5	93,75	81,25	93,75	87,5
β	81,25	87,5	88,24	81,25	88,24	87,5

Tabelle 1: Abhängigkeit der Matchingergebnisse von der Kachelungsgröße bei Set 1

	5	10	15	20	25	30
OK	20	19	23	16	14	15
NOK	9	10	8	8	8	7
REJ	6	5	3	7	6	7
α	68,97	65,52	79,31	55,17	48,28	51,72
β	68,97	65,52	74,19	66,67	63,64	68,18

Tabelle 2: Abhängigkeit der Matchingergebnisse von der Kachelungsgröße bei Set 2.

	5	10	15	20	25	30
OK	11	10	13	15	17	16
NOK	3	4	3	2	2	2
REJ	13	14	11	10	10	8
α	50	45,45	59,09	68,18	77,27	72,73
β	78,57	71,43	81,25	88,24	89,47	88,89

Tabelle 3: Abhängigkeit der Matchingergebnisse von der Kachelungsgröße bei Set 3



Abbildung 50: Die Größe der relevanten Details der Fahrzeuge ist sehr unterschiedlich. Im linken Bild reicht die Patchgröße von 24x24 Pixel, um die wichtigen Details mit einem Patch abzudecken. Im mittleren Bild würde eine größere Patchgröße bessere Features erzielen. Im rechten Bild ist die Patchgröße eher unkritisch, das gesamte Fahrzeug hat eine gleich verlaufende Textur.

Offensichtlich ist die Größe der Patches ein kritischer Parameter; da die Texturen der Fahrzeuge jedoch von der Größe her recht unterschiedlich sind (s. Abbildung 50), kann nur schwer ein für alle Fahrzeuge optimaler

Wert angegeben werden. Der Standardwert mit 16x16 Pixel liefert jedoch oft gute bis sehr gute Werte.

Pruning bzw. Featureselection

Nach der Kachelung kann auch noch durch den Pruning Parameter ein gewisser Steuereffekt erreicht werden. Je kleiner dieser Wert ist, umso feinere Abstufungen der Modekombinationen werden in die finale Signatur übernommen (und werden später gematcht). Dieser Wert darf jedoch nicht zu klein werden, da sonst künstliche Farbstufen (z.B. Verwischungen durch Bewegungsunschärfe) einen zu hohen Anteil an den Features ausmachen und das Matching verfälschen können. Der Wert des Parameters hängt mit der durchschnittlichen Sättigung der Farben im Objekt zusammen, und sollte bei gut gesättigten Farben größer, sonst kleiner gewählt werden. Abbildung 51 zeigt die berechneten Features in Abhängigkeit vom Pruning Parameter (damit ist der notwendige Mindestabstand zweier Features gemeint); die Features wurden noch nicht in den L^*u^*v -Raum transformiert. Die Diagramme (Abbildung 52-54) bis zeigen die Ergebnisse für die 3 Sets, wobei jeweils nur α (blaue Linie) und β (rote Linie) eingetragen sind. Bei Set1 zeigt die Featureselektion nur wenig Wirkung, da die Features größeren Abstand zueinander haben und somit nicht so stark von einem Pruning betroffen sind. Die Erkennungsrate ist in weiten Bereichen um 90 %. Bei Set 2 und 3 ist die Abhängigkeit größer, da hier die Features/Moden näher beieinander liegen; es zeigt sich jedoch bei allen Sets, dass

1. die optimale Performance *nicht* unter Berücksichtigung *aller* Features ($D=0$) erreicht wird
2. ab einem gewissen Wert ($D \approx 55-60$) die Performance im Trend sinkt.

Für sehr kleine Werte von D werden auch kleinste Farbabstufungen in die finalen Features übernommen, die aber für das Aussehen des Objektes

nicht relevant sind, sondern von Beleuchtung und Bewegung induziert; damit werden beim darauf folgenden Matching viele Pseudo-Korrespondenzen geschaffen zwischen „echten“ Features und künstlichen Features erzeugt. Wird D zu groß gewählt, steigt die Anzahl an möglichen Mismatches, da durch die Selektion zu wenige finale Features bleiben, um die Objekte ausreichend genau zu beschreiben.



Abbildung 51: Die Mode-Kombinationen mit unterschiedlichen Pruning Abständen. Man erkennt, dass auch mit sehr starker Pruning ($D=90$) die wesentlichen Farbübergänge weiß-blau-grau repräsentiert sind. „Reines“ Weiß (z.B. am Fahrzeugdach) kommt nicht vor, da zu helle Farben vom Algorithmus geclipped werden.

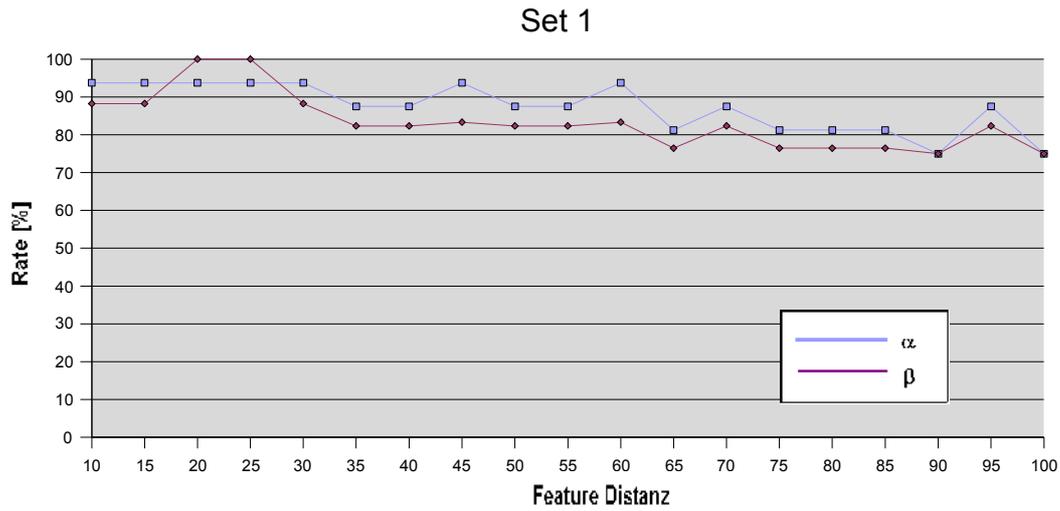


Abbildung 52: Bei Set 1 ist die Abhängigkeit vom Pruning Parameter nicht sehr ausgeprägt, es werden in weiten Bereichen gute bis sehr gute Ergebnisse erzielt.

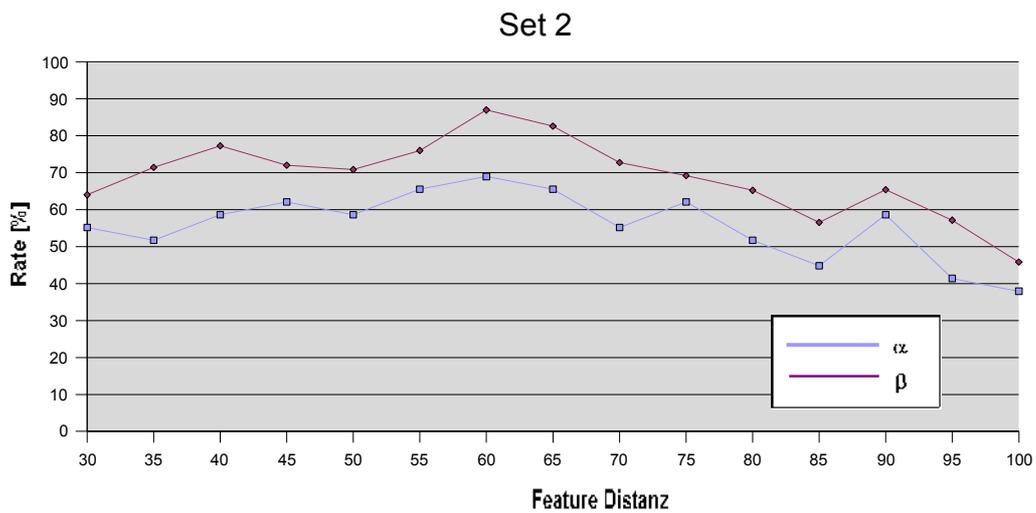


Abbildung 53: Bei Set 2 ist zwar die Abhängigkeit vom Pruning Parameter stärker, man sieht aber auch hier die Tendenz, dass die Performance für sehr große und sehr kleine Pruning Parameter deutlich sinkt.

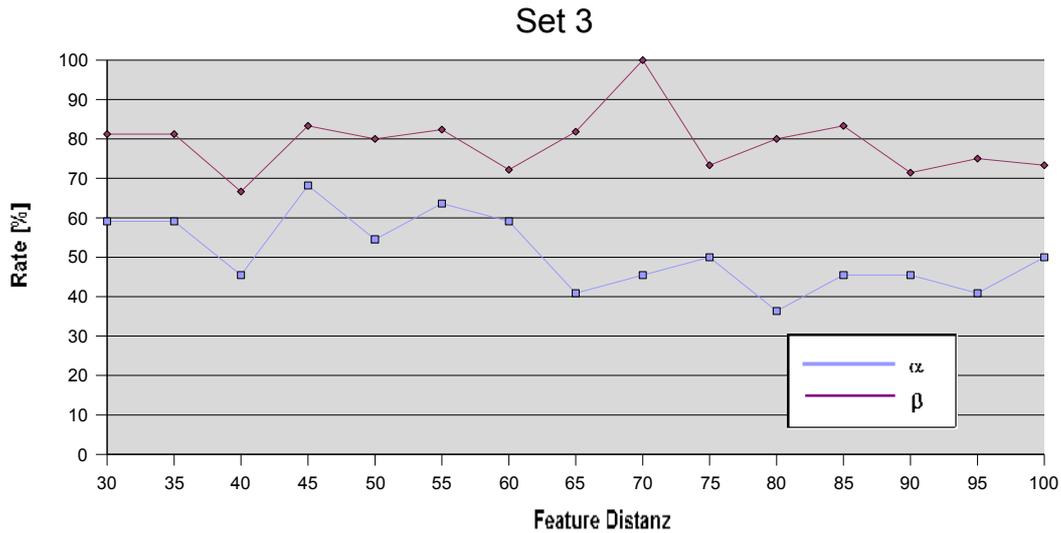


Abbildung 54: Der Verlauf des Diagramms bei Set 4 zeigt, dass bei diesem Set die unterschiedliche Bildqualität der zweiten Kamera die Bestimmung eines optimalen Wertes für den Pruning Parameter erschwert.

Unterschiedliche Gewichtung der Intensitätsinformation der Luv-transformierten Moden

Als letzter Schritt der Featuregenerierung erfolgt eine Berücksichtigung von Intensitätsschwankungen. Diesen wird durch eine Gewichtung der in den $L \cdot u \cdot v$ -Raum transformierten Farbkanäle Rechnung getragen, wobei allein der Gewichtungsfaktor für die Helligkeitsinformation (w) variiert wird. (Da der Gewichtungsfaktor über einen Durchlauf des Matchingverfahrens konstant bleibt, wird auf Variationen des Beleuchtungseinflusses, die sich innerhalb des Aufnahmezeitraumes ergeben, damit natürlich nicht reagiert.)

Der sinnvolle Bereich für Werte von w wurde empirisch ermittelt und liegt im Bereich um 0.6, wie auch in den Abbildungen 55 - 57 zu erkennen ist (blaue Linie = α , rote Linie = β); im Bereich dieses Werts ist die Performance für alle drei Sets optimal. Werte von $w \approx 1$ sind kaum sinnvoll, da dann die Intensität praktisch unverändert in die Features eingeht, und somit der Hauptvorteil der Transformation in diesen Farbraum verloren geht.

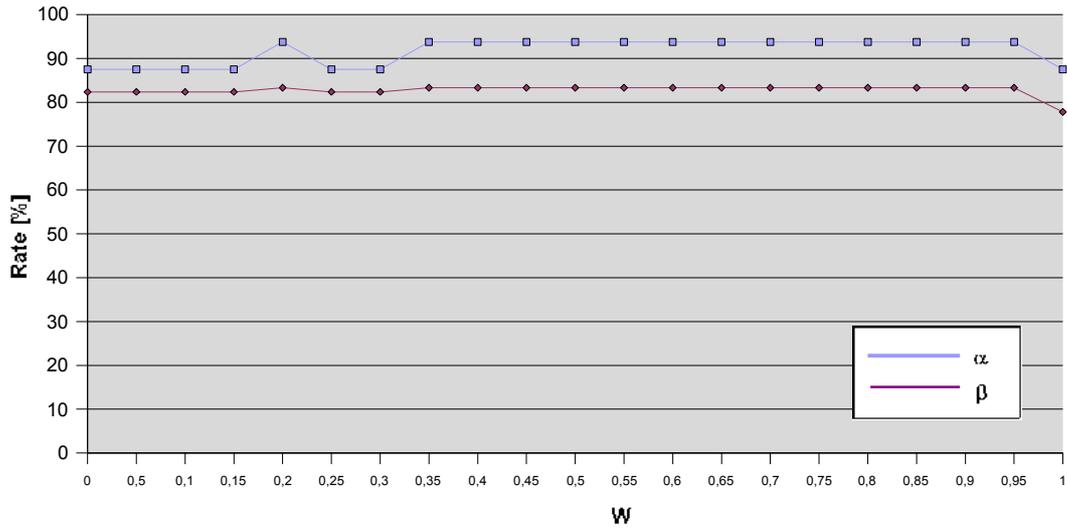


Abbildung 55: α und β bei Variation von w für das Set 1. Es zeigt sich keine nennenswerte Abhängigkeit, der optimale Wert liegt jedenfalls über 0.35. Die guten Features und der relativ schwache Einfluss der Beleuchtungsänderung sind der Grund für das stabile Verhalten.

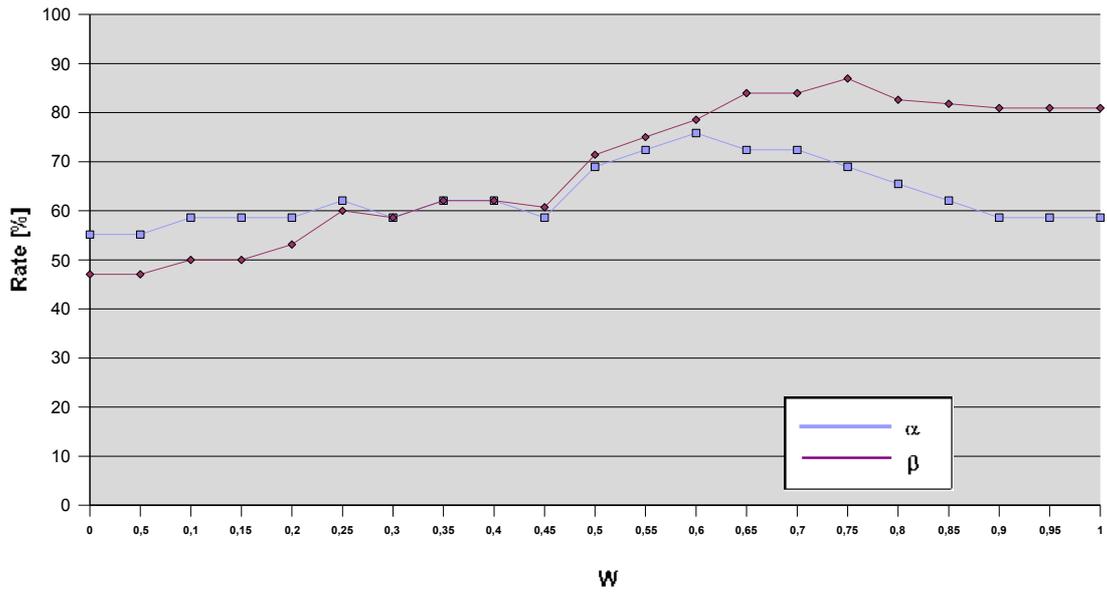


Abbildung 56: Bei Set 2 ist ein relativ gut ausgeprägtes Optimum um $w \approx 0.6 - 0.8$ zu sehen. Hier wird offensichtlich ein optimaler Kompromiss zwischen Invarianz und Diskriminanz erzielt.

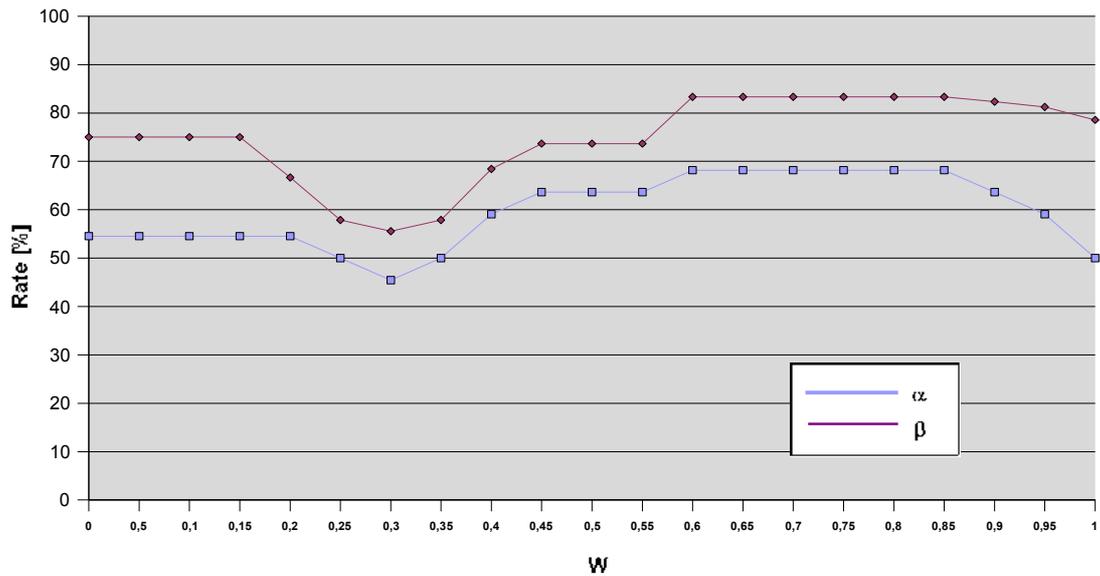


Abbildung 57: Auch bei Set 3 ist ein stabiler Bereich ab $w \approx 0.6$ vorhanden.

5.6 Untersuchung der Matching Stufe

Beim Featurematching dieses Verfahrens werden interne und externe Parameter verwendet. Interne Parameter beeinflussen das Matching zweier Merkmalsvektoren. Für die externen Parameter werden Informationen externer Quellen genutzt, um Bedingungen für die Notwendigkeit des Matchings einzuführen. Bei den verfahren-internen Parametern ist die Wahl der Distanzfunktion der Features wesentlich, die jedoch von der Berechnung der Features abhängt. Für die hier verwendeten Luv-Tripel wird aber stets die bereits auf Seite 63 beschriebene Distanzfunktion verwendet. Wesentlich ist die Wahl der Maximaldistanz für zwei Features, um noch als zusammengehörig klassifiziert zu werden. Damit lässt sich die Toleranz verändern, innerhalb der 2 Features noch als „gleich“ betrachtet werden. Durch die Größe des Wertes kann das Maß, in dem die Lichtverhältnisse zwischen den Aufnahmen eines Objektes sich ändern dürfen, werden.

Variation der Feature Maximaldistanz/ externen Parameter

In den Abbildungen 58-63 zeigt sich, wie bei einer Veränderung der Feature-Maximaldistanz für das Matching sich die Parameter α und β ändern, wobei für jedes Set jeweils ein Graph für α und β gezeigt wird. In jedem Graphen sind die Ergebnisse für

1. Matching ohne externe Parameter (Zeitstempel, Fahrzeugtyp)
(Legende: **NO_VT_NO_TS**)
2. Matching mithilfe von Zeitstempel (Legende: **TS**)
3. Matching mithilfe von Fahrzeugtyp (Legende: **VT**)
4. Matching mithilfe von Zeitstempel und Fahrzeugtyp
(Legende: **VT_TS**)

dargestellt. Damit wird gleichzeitig die Abhängigkeit von der Feature-Maximaldistanz mit der Auswirkung der externen Parameter gezeigt.

Durch den Zeitstempel wird ein einfacher aber wichtiger Constraint eingeführt, der sowohl Performance als auch Stabilität des Matching erhöht; da der Aufnahmezeitpunkt der Objekte bekannt ist, brauchen manche Objekte von vornherein nicht gematcht werden. Es wurde bewusst kein eigentliches Zeitfenster implementiert. Zählflussiger Verkehr ist schließlich möglich, und damit kann die Zeitdauer zwischen den Aufnahmen (und damit der obere Grenzwert des Zeitfensters) relativ groß werden. Daher wird nur eine Mindestdauer eingeführt, die Fahrzeuge für die Strecke einhalten müssen.

Als zweiter Parameter wird der Fahrzeugtyp betrachtet. Die Wirkung dieses Parameters ist stark davon abhängig, wie unterschiedlich die Typen der Fahrzeuge sind. Je mehr Matches durch diesen Parameter verhindert werden können, umso kleiner die Gefahr von Mismatches.

Damit ist natürlich auch eine höhere Anforderung an die Zuverlässigkeit des Parameters verbunden; bei der vorliegenden Anwendung wurde eine

manuelle Klassifikation vorgenommen, der erreichbare Fehler für automatische Klassifikation liegt aber bereits bei ca. 5% [46].

Beide Parameter dienen in jedem Fall der Vermeidung unnötiger Matchings, tragen also sowohl zum Laufzeitverhalten als auch zur Verbesserung der Erkennungsrate bei. Das erwartete Verhalten ist, dass die Resultate für $\text{NO_VT_NO_TS} < \text{TS} / \text{VT} < \text{VT_TS}$, d.h. ohne Verwendung externer Informationen am schlechtesten, mit Verwendung aller externer Informationen am besten sind und mit Verwendung von Fahrzeugtyp oder Zeitstempel jeweils dazwischen liegen.

	Set1	Set2	Set3
Kamera 1			
PKW	353	127	0
LKW	17	318	88
BUS	18	5	5
Sonstige	156	96	18
Kamera 2			
PKW	248	30	24
LKW	17	270	98
BUS	29	4	5
Sonstige	167	86	33

Tabelle 4: Verteilung der Fahrzeugtypen bei den Sets 1-3

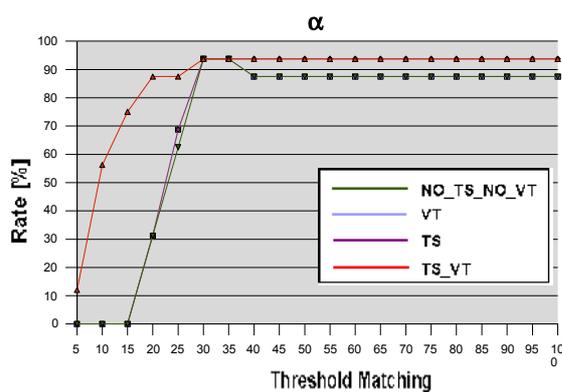


Abbildung 58: Der Wert von α für das Set 1. Ab einer Matching Threshold von ca. 35 ändert sich der Verlauf nicht mehr.

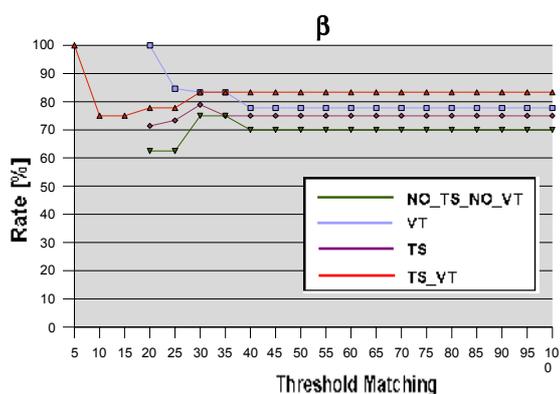


Abbildung 59: Der Wert von β für das Set 1. Wie bei α ist bei Werten ab ca. 35 ein stabiles Resultat erreicht.

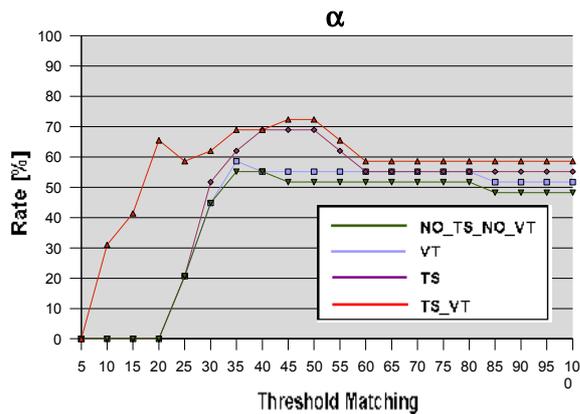


Abbildung 60: Der Wert von α für das Set 2. Hier stellen sich erst ab Threshold Werten von ca. 60 relativ stabile Endwerte ein.

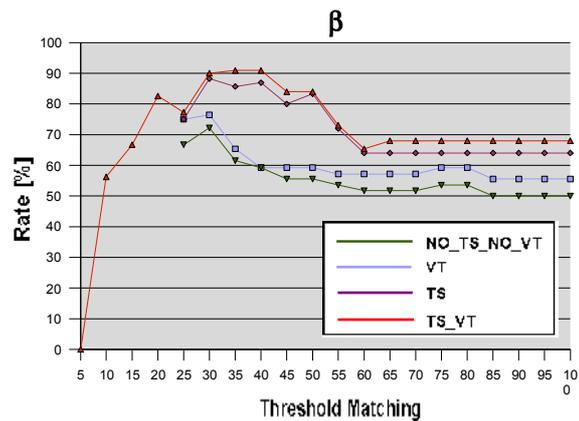


Abbildung 61: β von Set 2. Für den Verlauf gilt dasselbe wie für Alpha.

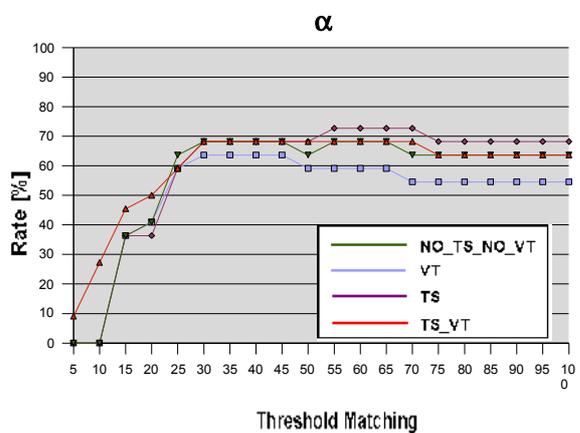


Abbildung 62: Bei Set 4 stellen sich ab Threshold Werten von ca. 60 relativ stabile Endwerte ein. Auffallend ist, dass mithilfe der externen Parameter nicht das beste Resultat erzielt wird!

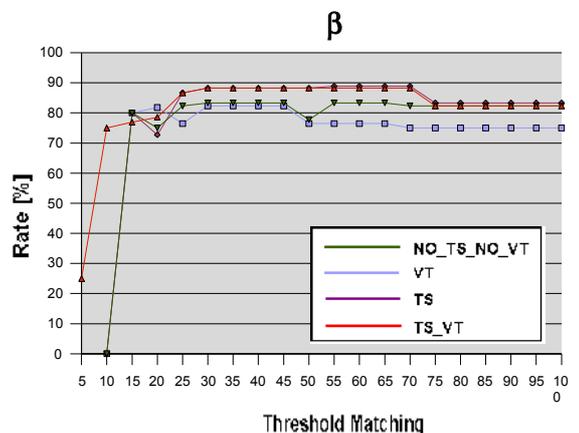


Abbildung 63: Für β von Set 4 werden noch gute Werte von 70% - 90% erreicht.

Für das Set 1 ist das Ergebnis wie erwartet. Für realistische Werte der Feature-Distanz ab ca. 30 ist bald ein stabiler Endwert erreicht, und die Resultate werden mit jedem zusätzlich eingesetzten externen Parameter besser. Letzteres gilt auch für das Set 2, wobei hier ein relativ ausgeprägter optimaler Bereich zwischen ca. 35 und 50 für die Matching-Distanz auffällt. Hier ist vor allem ausschlaggebend, dass wegen der gegenüber Set 1 ähnlicheren Fahrzeuge, die zu erwartende Distanz der Features zweier Fahrzeuge kleiner ist, somit der Threshold auf diesen sensiblen (kleineren) Bereich eingestellt werden muss.

Paradox erscheint, dass bei Set 3 die optimalen Ergebnisse nicht immer unter Verwendung aller externen Informationen erzielt werden. Dies ist jedoch durch das Verfahren erklärbar, und wird im Anhang anhand einer kleinen Fallstudie schrittweise gezeigt.

Es zeigt sich anhand der vorliegenden Daten, dass die Definition einer einfachen (linearen) Diskriminante kaum möglich ist, da durch die sich ändernden Lichtverhältnisse manche false positives bessere Scores erzielen als true positives (der überlappende Bereich der true positives und false positives ist also relativ groß). Erst durch Anwenden des Verifikationsschrittes können die meisten false positives als solche erkannt werden (und damit die akzeptablen bis guten Werte für β erklärt werden.)

Wirkung des Verifikationsschrittes

Ein wichtiger Schritt zur Stabilisierung der Ergebnisse ist der Verifikationsschritt nach dem Matching. Variiert man dabei den zulässigen Maximaldistanz der Moden, so verändert sich die Anzahl der gefundenen false positives. In diesem Abschnitt soll untersucht werden, wie stark sich die Anzahl an false positives durch Berücksichtigung des globalen Farbhistogrammes reduzieren lässt. Da in diesem Schritt die stärksten (diejenigen mit dem größten Support also) 5 Moden aus dem Selektionsschritt herangezogen werden, ist es wichtig, dass die Kernelbandbreite nicht zu klein gewählt wird, da ansonsten zu viele Farbschattierungen der dominanten Farben entstehen, was die Wahl der stärksten Moden nicht sehr sinnvoll macht.

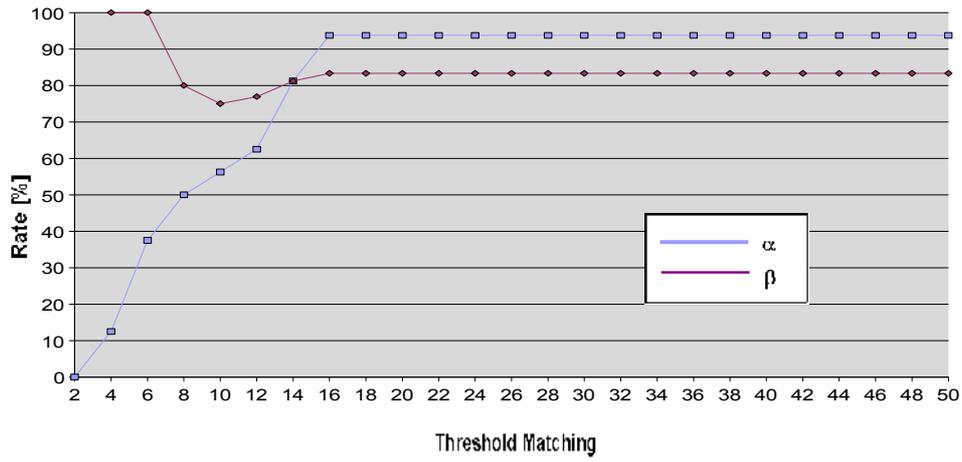


Abbildung 64: Verifikationsschritt bei Set 1.

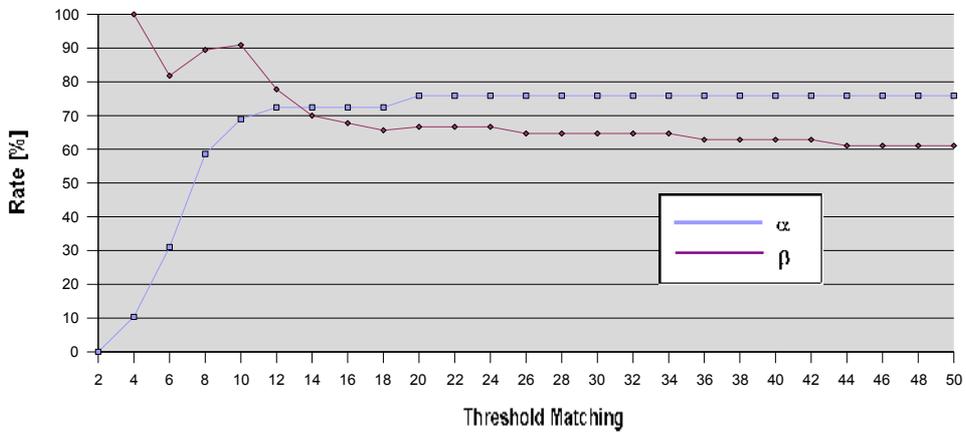


Abbildung 65: Verifikationsschritt bei Set 2

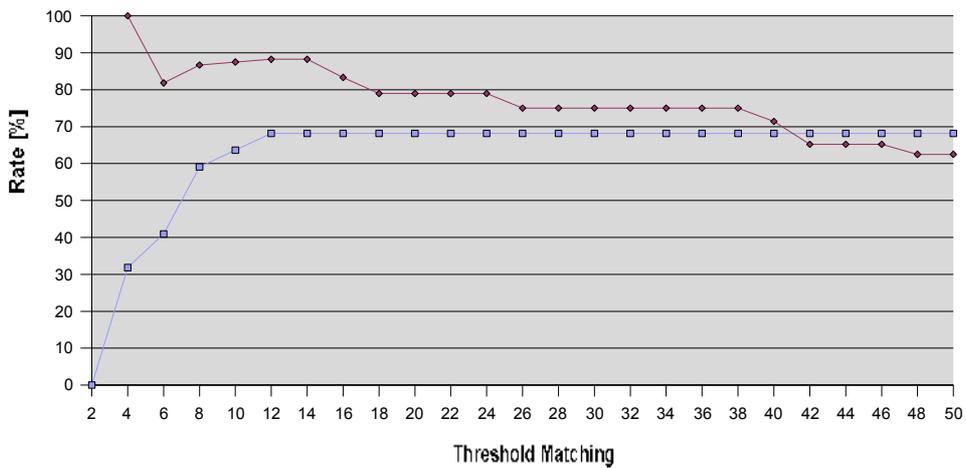


Abbildung 66: Verifikationsschritt bei Set 3

Die Abbildungen 64-66 zeigen, dass – vor allem für α - nach einem gewissen Einstellungsverhalten bei Werten von 12-16 immer eine stabile Lage erreicht wird. Unter diesen Werte werden die meisten Matches aussortiert; je kleiner der Threshold T ist, umso mehr korrekte Matches werden leider auch aussortiert, da z.B. durch Beleuchtungsänderungen zwischen den Aufnahmen immer eine gewisse Toleranz notwendig ist. Daher sind die Werte von α für kleine Thresholds ebenfalls sehr schlecht.

Mit steigender Toleranz werden immer weniger Matches „beanstandet“ (der Grenzfall $T \rightarrow \infty$ ist ja gleichbedeutend mit keiner Verifikation, da dann alle Matches akzeptiert werden), darum ändert sich das Verhalten auch nicht mehr so stark. Man erkennt an den Abbildungen auch gut die in [10] geäußerte Einschätzung, dass für die Optimierung von α und β in der Realität oft nur ein Kompromiss möglich ist. Da man mit der Verifikation vor allem β verbessern möchte, sollte der Wert von T nicht zu groß (typischerweise 10-14) gewählt werden. Abbildungen 67 und 68 zeigen Beispiele für echte false positives und solche, die durch die Verifikation erkannt worden sind.



Abbildung 67: Typische Beispiele für false positives, die nicht durch die Verifikation entdeckt wurden (echte false positives). Man erkennt gut, dass hier die auch die globalen Moden für die Verifikation ähnlich aussehen werden.

Abbildung 68: Beispiele für false positives, die durch die Verifikation entdeckt wurden.

5.7 Zusammenfassung der Ergebnisse für die Datensets

Hier wird zusammenfassend gezeigt, welche Performance man für die drei Sets mit gut gewählten Parametern erreichen kann. Die Ergebnisse sind aus folgendem Grund nicht unbedingt optimal: die Performance wird ja anhand der Parameter α und β bewertet. Da dies aber Verhältniszahlen sind, kann der Fall eintreten, dass optimale Werte von 100 % beim Vergleich von nur jeweils 2 Fahrzeugen auftreten; dies ist aber keine sinnvolle Anzahl von Fahrzeugen für die vorliegende Anwendung. Daher wurde die erreichbare Performance nur im Bereich der sinnvollen Werte für die Anzahl zu vergleichender Fahrzeuge gesucht. Für die Sets 1-3 werden die Ergebnisse in Tabellenform angegeben (Tabelle 5-7).

Für das erste (einfache) Set können sehr gute Ergebnisse erzielt werden; praktisch alle gut geeigneten Objekte werden vom Selektionsschritt erfasst und korrekt gematcht werden. Es sind zwar insgesamt nur wenige gut geeignete Objekte vorhanden, die meisten davon werden durch die Selektion gefunden.

Das zweite Set ist schwieriger, da wesentlich mehr „geeignete“ Fahrzeuge enthalten sind, die jedoch untereinander ähnlichere Features haben. Außerdem machen größere Schwankungen der Helligkeit dem Verfahren Probleme. Daher ist die Performance in diesem Set deutlich schwächer ($\alpha \approx 74\%$). Obwohl hier mehr Objekte nach der Selektion übrig sind, wurden dennoch strengere Bedingungen (vgl. dazu den Wert MIN_SCORE in Tabelle 4) angewendet.

Für das dritte Set sind für die Selektion konservative Einstellungen gewählt, da die Bildqualität durch den Regen durchschnittlich niedriger ist als in Set1 und Set2. Es sind in diesem Set aber fast nur LWK und Busse vorhanden, es steht daher immer noch eine relevante Anzahl an Fahrzeugen für das Matching zur Verfügung. Da diese untereinander

nicht zu ähnlich sind, gelingt auch hier eine gute Performance von ($\alpha = \beta \approx 82\%$)

Die wesentlichen verwendeten Parameter für die jeweiligen Sets sind der Vollständigkeit halber in Tabelle 8 zusammengefasst; es sind nur Parameter angegeben, die in den vorigen Kapiteln variiert wurden.

	<i>Anzahl Objekte gesamt</i>	<i>Anz. Objekte nach Preselection</i>	α	β
Stream 1	554	14	92.3 % (12 von 13)	100 %
Stream 2	461	21		

Tabelle 5: Ergebnisse für das Matching von Set 1

	<i>Anzahl Objekte gesamt</i>	<i>Anz. Objekte nach Preselection</i>	α	β
Stream 1	547	46	ca. 74.07 % (20 von 27)	80 %
Stream 2	390	54		

Tabelle 6: Ergebnisse für das Matching von Set 2

	<i>Anzahl Objekte gesamt</i>	<i>Anz. Objekte nach Preselection</i>	α	β
Stream 1	111	26	82.35 % (14 von 17)	82.35 % (14 von 17)
Stream 2	160	31		

Tabelle 7: Ergebnisse für das Matching von Set 3

	MIN_SIZE	MIN_SCORE	RADIUS	MIN_DIST	TILE_SIZE	THR_SELECT	w	THR_MATCHING	THR_VERIFICATION
Set1	70	15	20	15	16	60	0.75	35	15
	75	15	20	15	16	60	0.75	35	15
Set2	70	50	20	10	16	60	0.75	40	12
	75	60	20	10	16	60	0.75	40	12
Set3	70	15	20	10	16	60	0.75	30	12
	75	15	20	10	16	60	0.75	30	12

Tabelle 8: Die verwendeten Parameter für die Ergebnisse aus Tabelle 1-3. Die einzigen großen Abweichungen sind in MIN_SCORE von Set2 (rot) zu vermerken. Dies ist aus den vielen ähnlichen Fahrzeugen in diesem Set zu erklären, die eine strengere Selektion notwendig machen.

Die obere Zeile jedes Sets gibt die Parameter für Kamera 1, die untere Zeile die Parameter für Kamera 2 an. Diese sind durch die ähnlichen Aufnahmeeinstellungen meistens identisch.

5.8 Ermittelte Durchschnittsgeschwindigkeiten aus den Matches

Für die korrekten Matches wurden aus den Zeitstempeln die Durchschnittsgeschwindigkeiten ermittelt. Um die Werte zu stabilisieren, wurde ein gleitender Mittelwert mit der Mittelungsperiode $t = 3$ angewendet. Die Abbildungen 69 - 74 zeigen den Verlauf der Durchschnittsgeschwindigkeit für die Sets 1-3, wobei die x-Achse den Zeitpunkt des Matches widerspiegelt (also weiter „stromabwärts“) und in Minuten angegeben ist. Die Diagramme auf der linken Seite zeigen den Geschwindigkeitsverlauf wenn alle möglichen Fahrzeugkorrespondenzen aus der Ground Truth verwendet werden. Die rechte Spalte zeigt die Ergebnisse für die gematchten Fahrzeuge. Es ist zu erkennen, dass die Anzahl der gefundenen Matches nicht ausreicht, mit der gegebenen Methode (einfache gleitende Mittelung ohne Berücksichtigung der zeitlichen Distanz zwischen den „Messungen“) eine gute Annäherung an die reale Verkehrsdynamik zu erreichen. Unter Umständen würde hier eine andere Form der Mittelung (z.B. zeitlich exponentiell gewichtet) bessere Resultate liefern.

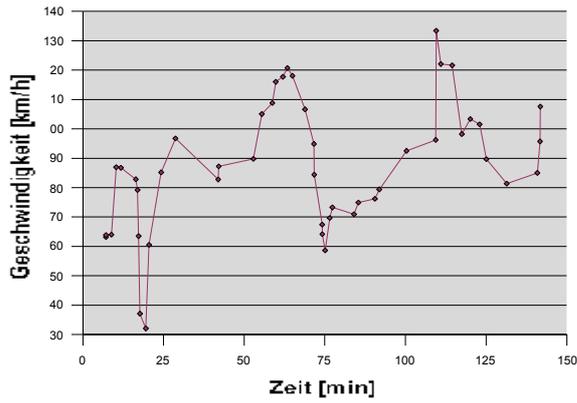


Abbildung 69: Geschwindigkeitsverlauf (Ground Truth) für Set1.

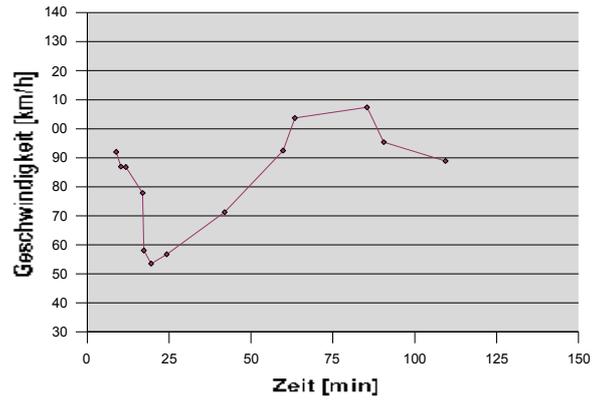


Abbildung 70: Geschwindigkeitsverlauf (laut Matching) für Set1.

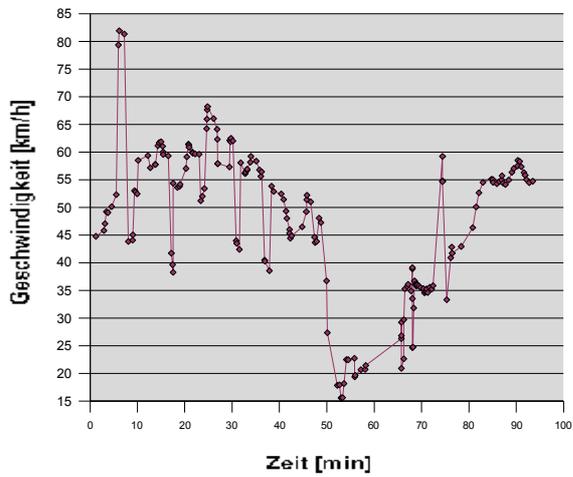


Abbildung 71: Geschwindigkeitsverlauf (Ground Truth) für Set2.

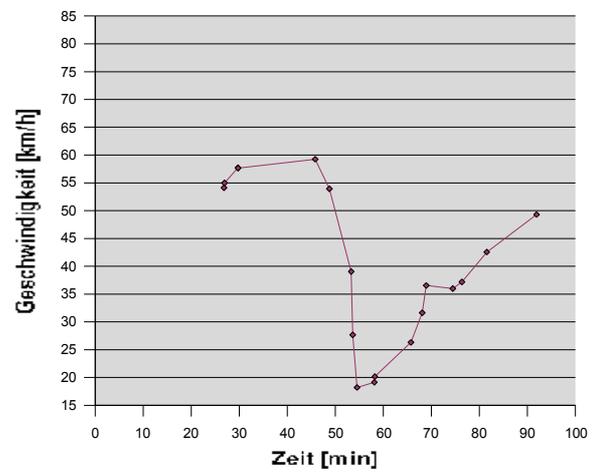


Abbildung 72: Geschwindigkeitsverlauf (laut Matching) für Set2.

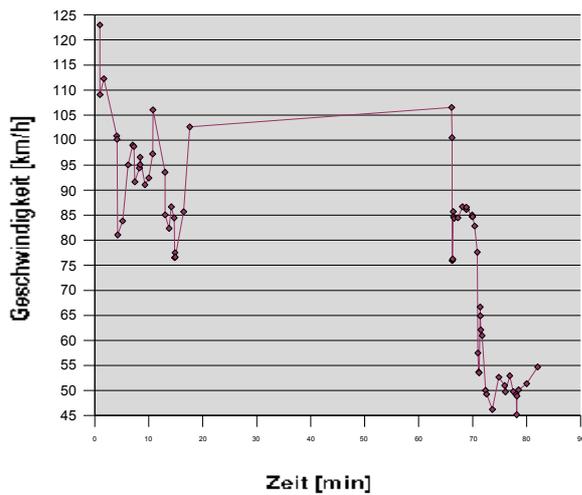


Abbildung 73: Geschwindigkeitsverlauf (Ground Truth) für Set3.

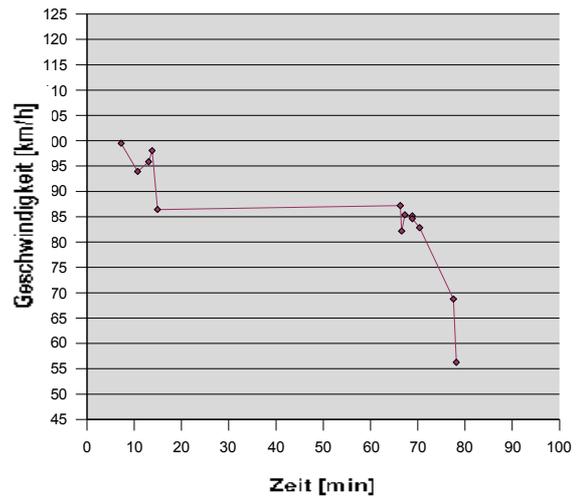


Abbildung 74: Geschwindigkeitsverlauf (laut Matching) für Set3.

5.9 Zusammenfassung

In diesem Kapitel wurden die wesentlichen Eigenschaften des vorgestellten Verfahrens untersucht. Dazu gehört in erster Linie die Auswirkung, die die Selektion auf die Performance des Algorithmus hat. Die Ergebnisse zeigen, dass die Selektion Fahrzeuge mit geeigneten Features erfolgreich herausfiltert; dieser Schritt ist weniger rechenintensiv als die Erzeugung der Features, und erhöht die Erkennungsrate. Allerdings bleiben nur mehr relativ wenige Fahrzeuge nach der Selektion über, wenn eine bessere Erkennungsrate gewünscht ist; die Dynamik der Geschwindigkeitsschätzung ist also gering und für den Einsatz im Feld wahrscheinlich nicht ausreichend. Ein höherer Wert für die Erkennungsrate α wäre hier wünschenswert; dies ist unter anderem bei einer besseren Bildqualität (ohne Komprimierung) zu erhoffen.

Für die Extraktion wurde der Einfluss wichtiger Parameter wie Patchgröße und das Ausmaß des Prunings sowie die Berechnung invarianter Features analysiert. Während die Patchgröße einen sensiblen Parameter darstellt und kein für alle Sets optimaler Wert gefunden werden kann, können für das Pruning und die Helligkeitsinvarianz relativ stabile Werte mit guten Ergebnissen für alle Sets angegeben werden. Die Extraktion komplexerer Features (Triple-Mode Kombinationen gegenüber 2-fachen Kombinationen wie in [38]) hat schließlich zu akzeptablen Resultaten in der Erkennungsrate geführt.

Aus der Untersuchung des Matchings können 2 wesentliche Erkenntnisse gewonnen werden: (1) die Einbeziehung externer Information (wie Zeit-Bedingungen) führt in fast allen Fällen zu besseren Resultaten und (2) ein zusätzlicher Verifikationsschritt vermindert die Anzahl der false positives drastisch; dies wird durch die Untersuchung globaler *und* lokaler Features erreicht.

6 Fazit und Ausblick

6.1 Zusammenfassung

Es wurde eine Methode für möglichst robustes Matchen von Fahrzeugen unter realistischen Bedingungen untersucht. Angelehnt an den Algorithmus der Multimodal Neighborhood Signature, wurden Modifikationen erläutert, um trotz der nicht idealen, aber vorgegebenen Qualität der Bilddaten akzeptable Ergebnisse zu erzielen. Mehrere Einsichten wurden dabei gewonnen:

- die **Verknüpfung** mehrerer einfacher Kriterien (constraints) kann das Matching robuster machen
- die Verwendung weniger, dafür besserer Daten (Objektselektion) ist in Anwendungen wie der vorliegenden (wo Aussagen qualitativer Natur das Ziel sind) von Vorteil. Es können Fahrzeuge, die für ein Matching nicht geeignet sind, ignoriert werden, da nicht alle Fahrzeuge für eine Geschwindigkeitsschätzung betrachtet werden müssen.
- die **Bildqualität** ist ein Schlüsselfaktor bei der Wiedererkennung. Auch wenn die Daten bei flüchtiger Betrachtung durchaus akzeptabel erscheinen, ist der tatsächliche Informationsverlust deutlich.

6.2 Ausblick

In Hinblick auf das übergeordnete Verfahren sind einige Erweiterungen möglich, die in der vorliegenden Arbeit noch nicht berücksichtigt sind und im folgenden diskutiert werden.

Lokalität der Features

Ein Problem bei der vorliegenden Implementierung ist die Tatsache, dass die Lokalität der Features beim Matchen keine Rolle spielt. Somit können

auch Features matchen, die von jeweils ganz anderen Teilen der Objekte stammen. Begünstigt durch die spezielle Situation, dass die Beobachtungsstandpunkte relativ ähnlich sind, und die Fahrzeuge immer in einer vergleichbaren „Pose“ aufgenommen werden, sind auch die korrespondierenden Features oft an ähnlicher Stelle im Bild zu finden. Vor allem, da hauptsächlich Busse und LKW als Objekte in Frage kommen, die einen gut geeigneten geometrischen Aufbau besitzen, könnte ein Miteinbeziehen der Position eines Features im Bild die Ergebnisse verbessern: nicht nur die Distanz zweier Features im Farbraum wird bewertet, sondern auch die räumliche Distanz geht in den Score ein. Allerdings ist diese Variante nicht sehr generalisierungsfähig.

Platoon Matching

Wenn die Distanz zwischen den Kameras nicht zu groß ist, so kann ein einfacher Effekt genutzt werden: Es werden nicht mehr nur jeweils einzelne Fahrzeuge auf beiden Streams betrachtet, sondern die Gruppe an Fahrzeugen, die sich zum selben Zeitpunkt im Blickfeld der Kamera befindet (Fahrzeugverbände oder „Platoons“). Diese Gruppe wird sich zwar mit größerer Entfernung von der Kamera immer mehr verändern (unterschiedliche Geschwindigkeiten innerhalb einer Gruppe), bei relativ kleiner Distanz zwischen den Kameras kann dennoch die Information über die Gruppenzugehörigkeit dazu genutzt werden, Aussagen über die Wahrscheinlichkeit eines Matches zu machen.

Komplexere Verfahren zur Vorselektion

In der vorliegenden Implementierung ist die Beurteilung der Farbigkeit ein relativ einfacher Algorithmus, was ja auch eine Designentscheidung war: die Vorselektion soll im Vergleich zur Feature Extraktion schnell sein. Trotz der guten Performance werden einige Fahrzeuge akzeptiert, bei denen die Buntheit nur scheinbar gegeben ist: sie entsteht durch

Beleuchtungseffekte bei wirklich einfarbigen Objekten. Ein - immer noch einfacher - Algorithmus, der diese Farbstufen von echten Objektfarben zumindest teilweise trennen kann, könnte die Treffsicherheit der Vorselektion zusätzlich erhöhen.

Berücksichtigung relativer Farbhäufigkeiten bei der Vorselektion

Neben den bereits diskutierten Möglichkeiten der Vorselektion wie Bildgröße oder Farbgehalt, könnte auch die relative Häufigkeit der Fahrzeugtextur als Auswahlkriterium herangezogen werden. Fahrzeuge, die eine - innerhalb eines bestimmten Zeitabschnittes - sehr selten oder nur einmal auftretende Farbe haben, könnten ebenso in die Menge der selektierten Objekt aufgenommen werden wie sehr farbenreiche Objekte, da nur sehr wenige oder gar kein Fahrzeug mit diesem „verwechselt“ werden könnten, ausgenommen das ursprüngliche Fahrzeug.

Einsatz statistischer Methoden zur Steigerung der Robustheit

Durch Bestimmen der Durchschnittsgeschwindigkeit nach einer bestimmten Anzahl von durchgeführten Matches, kann das Matching weiter verfeinert werden. Es ist dann leichter möglich, Ausreißer in den Daten zu erkennen (Objekte, deren errechnete Geschwindigkeit infolge eines Matches zu stark von der Durchschnittsgeschwindigkeit abweicht), und statistisch robusterer Aussagen über den Geschwindigkeitsverlauf zu machen. Beispielsweise könnten mehrere Geschwindigkeitsprofile (sinnvollerweise unter Berücksichtigung der Spur auf der die Fahrzeuge unterwegs sind) über Gaußverteilungen parallel modelliert werden. Diese müssten dann ständig adaptiert werden, um die Dynamik der Geschwindigkeitsverlaufs zu berücksichtigen.

Lernen bzw. automatische Bestimmung der Parameter

Viele Verfahren leiden unter dem Problem, das zwar eine Lösung der zugrunde liegenden Aufgabenstellung möglich ist, deren Qualität aber sehr stark mit einem bestimmten, meist empirisch gefundenen Parametersatz verbunden ist. Für veränderte Rahmenbedingungen müssen diese Parameter erneut bestimmt (oder aus vorhandenen Musterparametern geschätzt/extrapoliert) werden – damit ist die praktische Einsetzbarkeit mancher Verfahren stark in Frage gestellt. Während im hier vorgestellten Verfahren nur eine geringfügige Abhängigkeit von den Parametern besteht (und diese nur in einem kleinen Bereich variieren), so ist es für optimale Ergebnisse dennoch notwendig, die Parameter anzupassen. Soll dies automatisch geschehen, muss laufend die Information aus den Bilddaten bewertet werden (z.B. statistisch), und daraus ein Schätzer für eine möglichst optimale Parameterkombination entwickelt werden (z.B. durch Optimierung einer Kostenfunktion). Dies ist jedoch eine komplexe Aufgabe, und wurde in dieser Implementierung nicht berücksichtigt.

6.3 Fazit

Farbe als Feature zur *Objektwiedererkennung* ist eine wertvolle Objekteigenschaft. Farbe ist ein wesentlicher Bestandteil der Objektidentität vieler (singulär betrachteter) Objekte; wenngleich im Bereich der Fahrzeugdetektion auch keine Klassifikation auf Basis von Farbfeatures sinnvoll ist, so ist eine Wiedererkennung dennoch möglich. Es muss allerdings gewährleistet werden, dass alle letztlich zu matchenden Objekte eine Farbstruktur aufweisen, aus der gute Features berechnet werden können. Diese Problematik hat die vorliegende Arbeit mittels Vorselektion gelöst; hier liegt jedoch noch ein großes Potential, das weitere Untersuchungen in diesem Bereich sinnvoll erscheinen lässt.

7 Anhang

7.1 Implementierungsbeschreibung

Da für Implementierung und Test der Wiedererkennungsstufe Ausgangsdaten (also Bilder mit Fahrzeugen) notwendig sind, wurde ein eigenes System für eine Objektdetektion implementiert. Dieses wurde in der eigentlichen Arbeit stillschweigend vorausgesetzt, ist aber der Vollständigkeit halber hier in Form einer Implementierungsbeschreibung dargestellt.

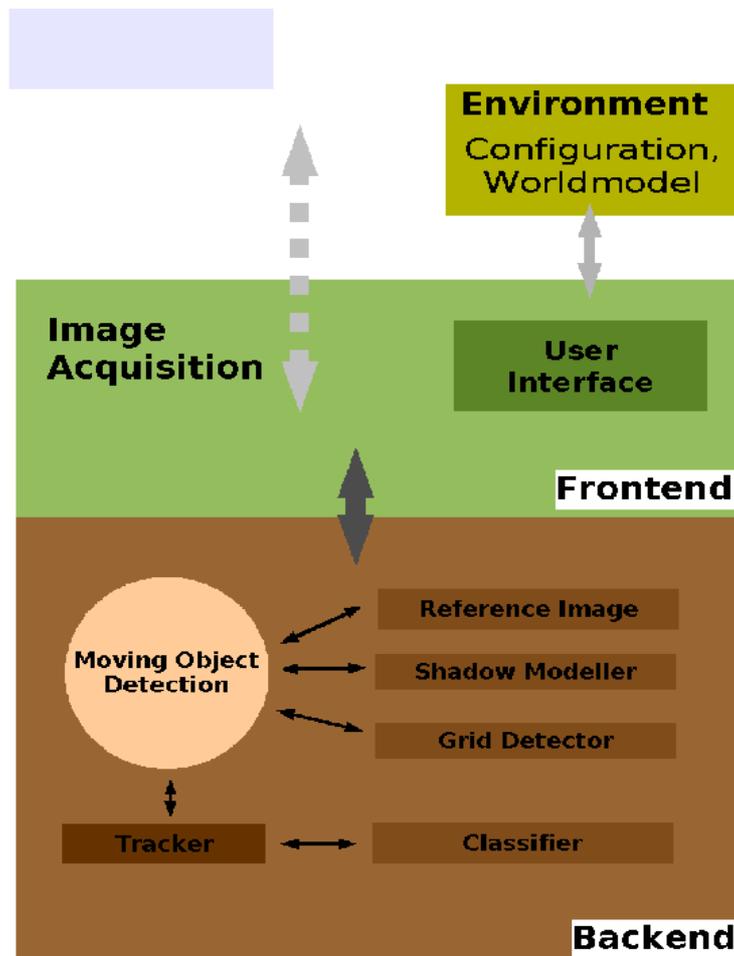


Abbildung 75: Architektur des implementierten Systems. Der zentrale Beitrag dieser Arbeit ist links unten schraffiert zu sehen.

Als Implementierungssprache fiel die Entscheidung auf C++, das sehr effizienten Code ermöglicht und gleichzeitig fortschrittliche Sprachmittel zum Entwickeln komplexer Anwendungen bietet. Weiterer Vorteil sind die vielen Bibliotheken, die für C++ existieren.

Für die Entwicklung des Erkennungsverfahrens wurde schließlich das Mathematikpaket MatLab (R12) verwendet, das für Prototyping deutlich besser geeignet ist als C++. Die verwendeten Tools und Bibliotheken sind im Anhand angeführt. Im folgenden werden die einzelnen Module der Abbildung kurz besprochen.

Environment

Über eine Konfigurationsdatei werden im wesentlichen die folgenden Parameter verwaltet:

- Koordinaten einer Region of Interest und der Fahrbahnsuren
- Netzwerkeinstellungen, um die Bilder vom Kameramodul abzufragen
- Interne Parameter der einzelnen Algorithmen

Image Acquisition

Dieses Modul hat die **Bereitstellung der Bilder** für das System zur Aufgabe. Die analogen Ausgangssignale der Videokameras werden von „intelligenten“ Dekodermodulen (Tricams) der Firma Frequentis in einen JPEG Stream (Auflösung: 320x288 Pixel) konvertiert und über eine eingebaute Ethernet Schnittstelle in das Computernetzwerk verteilt; sie werden dann von der Applikation über ein Netzwerkinterface empfangen.

User Interface

Die Steuerung der Applikation ist entweder über eine grafische Oberfläche (direkt vor Ort) oder über ein Textinterface möglich, sodass auch eine Fernsteuerung z.B. über einen Telnet-Zugang durchführbar ist.

Moving Object Detection (MVOD)

Dieses Modul ist der Kern des Systems. Aus Gründen der Einfachheit und Effizienz wurde eine Moving Object Detection auf Basis von **Hintergrundmodellierung und -subtraktion** implementiert. Als guter Kompromiss zwischen Performance, Speicherbedarf und Güte wurde der approximierte Median implementiert. Dieser bildet einen laufenden Schätzer für den Median, der gegen Wert des wahren Median konvergiert [47]. Der Median aus den letzten L Frames wird dabei angenähert durch ein Funktion, die folgendermaßen definiert ist:

$$B(i, j) = \begin{cases} B(i, j)-1 & \text{für } I(x, y) < B(i, j) \\ B(i, j)+1 & \text{für } I(x, y) > B(i, j) \\ B(i, j) & \text{sonst} \end{cases}$$

mit $B(i, j)$ als Hintergrundmodell und $I(i, j)$ als aktueller Frame.

Der approximierte Median benötigt nur ein Referenzbild und jeweils das aktuelle Bild für das Update, und kommt mit 1-2 Vergleichen pro Pixel aus. Das erreichte Verhalten ist ein guter Kompromiss aus Adaptivität und Toleranz gegen kurze transiente Störungen. Durch unterschiedliche Inkremente kann außerdem die Adaptivität gesteuert werden.

Mit folgender Vorschrift vor wird aus dem RGB-Referenzbild I_{ref}^c und dem aktuellen RGB-Frame I^c ein binäres Differenzbild I_d erzeugt:

$$1. I_{\text{sub}}^c = |I_{\text{ref}}^c - I^c| \text{ für } c=R, G, B$$

$$2. I_d = \begin{cases} 0 & \text{falls } \frac{I_{\text{sub}}^R + I_{\text{sub}}^G + I_{\text{sub}}^B}{3} < T \\ 1 & \text{sonst} \end{cases}$$

Anschließend wird auf das Differenzbild noch die morphologische Sequenz Schließen->Öffnen durchgeführt, um schwach zusammenhängende Objektteile wieder stärker zu binden (Operation **Schließen**), und Bildartefakte mit kleiner Ausdehnung zu eliminieren (Operation **Öffnen**).

Reference Image

Das Reference Image Modul entspricht dem Background Model, über eine Update Funktion wird vom MVO Modul immer der aktuelle Frame in das Hintergrundmodell integriert. Das aktuelle Referenzbild kann von anderen Modulen (nach einer gewissen Vorlaufzeit, bis der Filter in einen stabilen Zustand übergeht) jederzeit abgerufen werden.

Shadow Modeller

Nachdem die Detektionsergebnisse durch Schatten immer wieder zu Fehlsegmentierungen führten (unterschiedliche Objekte „kleben“ an den Schatten zusammen), wurde auch ein Modul zur Schattendetektion eingebaut [48], das mit Farbverhältnissen und -differenzen arbeitet. Dazu werden zuerst das Referenzbild und der aktuellen Frame in den HSV Raum transformiert. Dann wird für jeden Pixel das Verhältnis der V-Komponenten und die Differenz der S- und H-Komponenten zwischen aktuellem Bild und Referenzbild berechnet. Wenn diese Werte bestimmte Schwellwerte übersteigen, so wird der Pixel als „Schattenpixel“ klassifiziert und im aktuellen Bild mit dem Wert aus dem Referenzbild überschrieben. All diese Pixel werden damit beim nächsten Differenzbild nur mehr als Hintergrund behandelt.

Grid Detector

Es wurde ein eigenes Scanline Verfahren entwickelt, dessen Basis die bereits beschriebene Region of Interest (Polygonzug) ist. Die Idee der Segmentierung ist nun, die Scanlines nicht den Koordinatenachsen des Bildes gemäß anzuordnen, sondern ein neues Koordinatensystem zu definieren, dessen 2 Achsen X_1 , X_2 durch die ROI definiert werden.

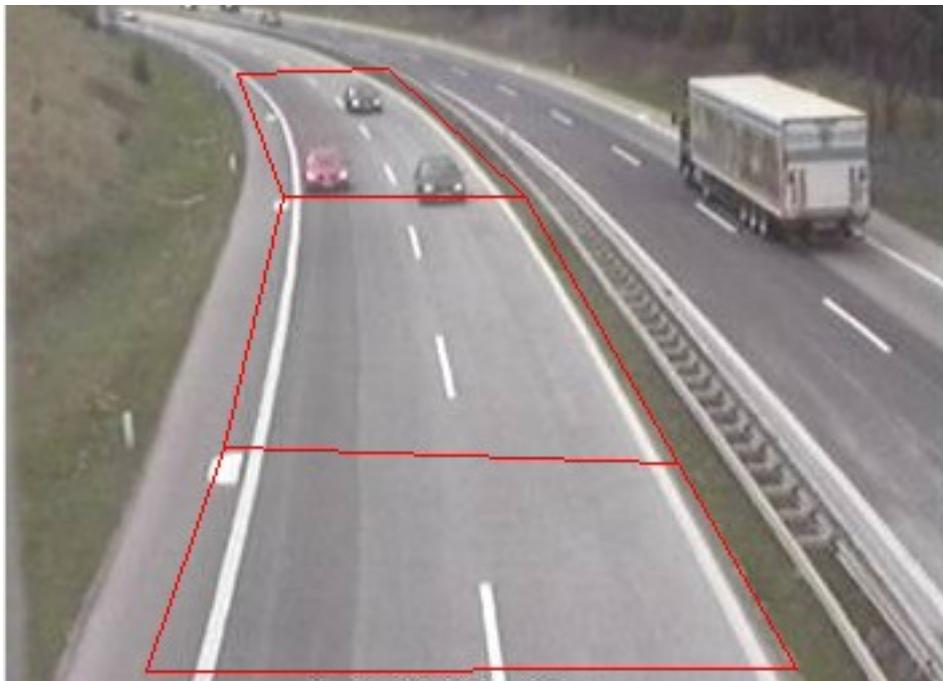


Abbildung 76: Für jede Kameraansicht wurde eine Region of Interest definiert, die jedoch nicht aus einem rechteckigen Bildbereich besteht, sondern auf einem Polygonzug beruht, der den Fahrbahnverlauf nachbildet.

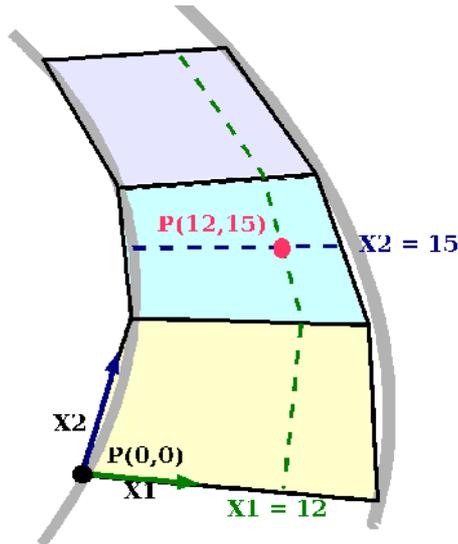


Abbildung 77: Koordinatensystem zur Approximation der Straßengeometrie

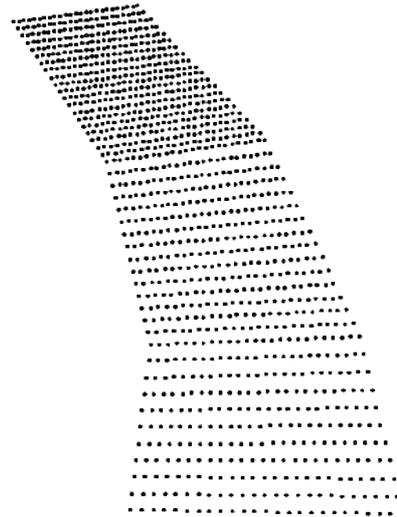
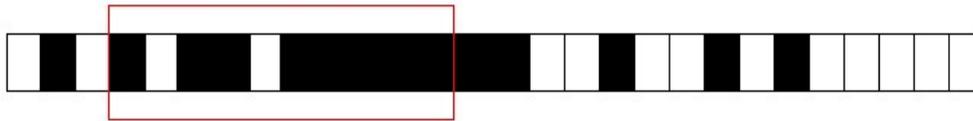


Abbildung 78: Diskretisierung der Straßengeometrie

Die Segmentierung verläuft nur mehr anhand dieses Gitters, welches über das Differenzbild gelegt wird. Für jede Scanline (in den Koordinaten des Gitters) wird dann der Inhalt des „darunter liegenden“ Differenzbildes ausgelesen. Damit wird eine sehr gute Performance erreicht, und die Genauigkeit der Methode kann für jede Szene durch die Anzahl der Unterteilungen fein abgestimmt werden. Das Scanline Verfahren selbst besteht aus 2 Schritten: (1) Durchsuchen des Differenzbildes nach günstigen Startpunkten („Seeds“) für eine Segmentierung, und (2) einer Art vereinfachten Region Growing: aus einem Startpunkt wird ein Rechteck solange nach allen möglichen Seiten hin vergrößert, bis an allen Seiten zu wenige Pixel für eine verbundene Region detektiert werden, was dann als äußere Objektkontur interpretiert wird. Dazu werden Scanlines benutzt. Eine Zeile von Bildpunkten wird von einem Ende bis zum anderen durchlaufen. Falls in dieser Zeile eine durchgehende Pixelsequenz gefunden wird, in der von M Pixeln N gesetzt sind, so wird der Scan als erfolgreich gewertet (s. Abbildung 79 mit $M=10$ und $N=8$).



8 von 10 Pixel
sind gesetzt

Abbildung 79: Beispiel einer Scanline, in der ein gültiger Seed gefunden wurde.

Der Algorithmus einer einzelnen Blobdetektion ist wie in Abbildung 80 illustriert: das Objekt ist grau eingezeichnet, eine Scanline ist exemplarisch blau unterlegt, das Growing als rote Rechtecke angedeutet. Das Ergebnis für den gesamten Frame ist ein Vektor von Blobs; innerhalb der Koordinaten dieser Rechtecke im Bild liegen die Objekte:

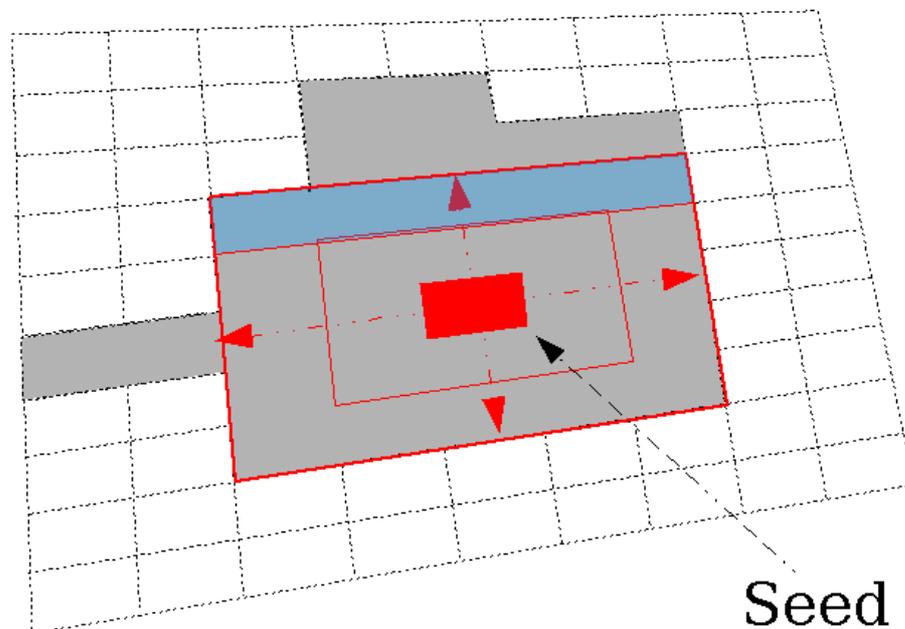


Abbildung 80: Blob Growing (Schema). Ausgangspunkt ist ein Seed (rotes Rechteck in der Mitte), der zuvor mittels Scanlines detektiert wurde.

Tracker & Classifier

Das Tracker-Modul stabilisiert die MVOD. Für jeden gefundenen Blob wird ein Tracker instantiiert; dieser implementiert die Kalman Systemgleichungen mit effizienten Matrixoperationen. Nach der ersten Initialisierung mit der Position, der Größe und einem Schätzwert für die Geschwindigkeit der Blobs, wird später nur mehr in der Nähe der „vorhergesagten“ Koordinaten des Trackers nach dem Blob gesucht. Aus den im Tracker gespeicherten Bildern wird mithilfe des Classifier Moduls eine Klassifikation der Fahrzeuge vorgenommen. Im Wesentlichen wird ein ADA-Boost Algorithmus auf die vom Hintergrund ausgeschnittenen Fahrzeugbilder angewendet. Ein offline trainierter Klassifikator nimmt darauf eine Einteilung in die 4 Fahrzeugtypen Motorrad, LKW Bus und PWK vor. Dieses Verfahren ist Gegenstand einer eigenen Diplomarbeit [46]. Die Ergebnisse werden zum Stabilisieren des Matching verwendet.

7.2 Eingesetzte Tools

Die folgende Tabelle gibt einen Überblick über die in der zuvor beschriebenen Implementierung verwendeten Werkzeuge und Bibliotheken.

Bibliothek bzw. Tool	Beschreibung	Her- steller	Ver- sion	Einsatz
DevIL (Developers Image Library)	Laden, Speichern in diversen Bild- formaten, einfacher Zugriff und Bearbeitung von Bilddaten	Denton Woods	1.6.6	Laden und Speichern von Bildern, Extrahieren von Bildausschnitten
IPP (Intel Performance Primitives) (Bild- verarbeitung)	Performante Implementierung von Funktionen zur Bildverarbeitung (Filter, Farbraum- Konvertierungen, Signalverarbeitung ,...)	Intel	4.1	Filter (Median, Gauss, ...), Bildarithmetik (Hintergrundsubtraktion), Logische Operationen auf Bilder, Morphologische Operationen, Farbraum- Konvertierungen, Subsampling
IPP (Intel Performance Primitives) (Matrizen)	Performante Implementierung von Funktionen zur Manipulation großer Matrizen	Intel	4.1	Kalman Filter (Berechnungen der Systemgleichungen), Berechnung der Homographiegleichungen
Qt	GUI Bibliothek, Elemente zur Darstellung grafischer Oberflächen, zahlreiche Zusatzmodule	Trolltech	3.1.1	Grafischer Wrapper (Konfiguration div. Ein- stellungen, visuelle Ausgabe und Definition von ROI und Punkten für Homo- graphie), Multithreading, XML-Konfiguration
MySQL	Freie SQL Datenbank- verwaltung	MySQL AB	4.x	Speicherung von Ergebnissen (Fahrzeugtyp , Geschwindigkeits- schätzung, benutzte Fahrbahn)

Tabelle 9: Eingesetzte Software Tools

7.3 Nachtrag: Fallbeispiel aus der Evaluierung

An dieser Stelle wird das Fallbeispiel nachgetragen, das aus Übersichtsgründen in der Evaluierung nicht mehr Platz gefunden hat. Das (paradoxe) Problem entsteht dadurch, dass beim Matching unter Zuhilfenahme des Fahrzeugtyps ein schlechteres Resultat entsteht als wenn der Fahrzeugtyp nicht verwendet wird. Als Beispiel werden nun folgende Fahrzeuge gematcht:



Abbildung 81: Die 3 Fahrzeuge aus Set 1. Die Fahrzeugindizes sind 1 (links oben), 2 (rechts oben) und 3 (links unten)

Abbildung 82: Die 3 Fahrzeuge aus Set 2. Die Fahrzeugindizes sind ebenfalls 1 (links oben), 2 (rechts oben) und 3 (links unten)

Die richtigen Matches (1-1 und 3-3) sind offensichtlich; für alle Fahrzeuge im Set 1 sind die Fahrzeugtypen LWK eingetragen. In Set 2 ist nur Fahrzeug 2 (der Wohnwagen) als „Sonstige“ klassifiziert. Die Score-Matrizen sehen nun folgendermaßen aus:

0.6005	0.9480	0.9315
0.9164	0.6727	0.6849
0.8963	0.9053	0.8282

0.6005	1	0.9315
0.9164	1	0.6849
0.8963	1	0.8282

Tabelle 10: Die Score-Matrizen nach dem Matching. Links ist die Matrix für den Fall ohne Fahrzeugtyp, rechts das Ergebnis mit Berücksichtigung des Fahrzeugtyps. Die Zeilen stehen für das erste Set, die Spalten für das zweite Set. Der Threshold für den Score ist 0.85. Grüne Zahlen sind korrekt gefundene Matches (true positives), rote Zahlen entsprechen falschen Matches (false positives). Blau ist ein zwar inkorrekt, aber in der Verifikation als solcher erkannter false positive.

Im ersten Fall (linke Matrix, ohne Fahrzeugtyp) wird zuerst der Match (1,1) gefunden (und damit die erste Zeile und Spalte aus der Matrix „gestrichen“). Dann folgt als nächster Match (2,2), der inkorrekt ist, und Spalte 2 und Zeile 2 wird gestrichen. Als letzter Match bleibt nur das Element (3,3), da es kleiner als der Threshold ist, wird auch dieser - ebenfalls korrekte - Match gefunden. Von 2 möglichen Matches wurden alle 2 gefunden, α ist damit **100%**. Von 3 gefundenen Matches waren 2 korrekt, β ist somit **66.67%**.

Im zweiten Fall (Matrix rechts, mit Fahrzeugtyp) wird ebenfalls zuerst der Match (1,1) gefunden (die erste Zeile und Spalte aus der Matrix wird wieder gestrichen). Nun stellt sich heraus, dass das Fahrzeug 2 aus Set 2 durch den Fahrzeugtyp „Sonstige“ zu keinem Fahrzeug aus Set 1 passt, somit ist Spalte 2 in der rechten Matrix komplett gestrichen (rot hinterlegt). Somit bleiben nur die Felder (3,2) und (3,3) zum Matchen, und hier ist der Wert (3,2) das Minimum (obwohl ein falscher Match). Damit ergibt sich für α nur **50%**, für β jedoch wieder **100%**, da der falsche Match durch die folgende Verifikation erkannt wird (s. blauer Wert in der Tabelle als entsprechenden Hinweis.) Die Interpretation des Beispiels ist, dass eine Verschlechterung mit Zusatzinformationen dadurch möglich ist, dass durch den Ausschluss von bestimmten Matching-Kombinationen – wie im Beispiel durch Ausschluss des false positives (2,2) – eine global schlechtere Matching-Situation entstehen

kann, wenn dadurch z.B. andere true positives beeinflusst werden (im Beispiel (3,2) statt (3,3)).

8 Literaturverzeichnis

1. Journal of Public Transportation. National Center of Transit Research, University of South Florida. Volume 8, No. 5, pp 3-4. 2005
2. EEA Briefing 3/2004 - „Verkehr und Umwelt in Europa“.
http://reports.de.eea.europa.eu/briefing_2004_3/de/DE_Briefing_No_03_web.pdf
3. Helmut Oismüller. „Die neue Verkehrspolitik“.
<http://www.gsv.co.at/doku/verkehrspolitik.pdf>
4. Weißbuch – „Die europäische Verkehrspolitik bis 2010: Weichenstellungen für die Zukunft“. Amt für amtliche Veröffentlichungen der Europäischen Gemeinschaften. Luxemburg. 2001
5. http://www.verkehrsforum.de/publikationen/magazin_archiv/1_98/1_98_4.html
6. Meffert, B.; Blaschek, R.; Knauer, U.; Reulke, R.; Schischmanov, A.; Winkle, F. „Monitoring Traffic by Optical Sensors“. 2nd International Conference on Intelligent Computing and Information Systems, Cairo, Egypt, Proceedings pp. 9-14. March 2005
7. Rosetta Projekt.
http://www.trg.soton.ac.uk/rosetta/workareas/3_mmr/mmr_pr1_present.htm
8. Chatterjee, K.; Hounsell, N.B.; Waterson, B.J.; Firmin, P.E.; Bonsall, P.W. „Evaluation of the London driver information system: results from the CLEOPATRA study“. Road Transport Information and Control, 2000. Tenth International Conference on (Conf. Publ. No. 472). Page(s):22 – 26

9. Projekt Info A10.
http://www.attc.at/presse/Pst_17_08_2006/pst_attc_170806_projekt_info_a10.pdf (2007)
10. Knauer, U.; Reulke, R.; Meffert, B. „Fahrzeugdetektion und -erkennung mittels mehrdimensionaler Farbhistogrammanalyse“. Farbbildverarbeitung 2005, Gesellschaft zur Förderung angewandter Informatik, Berlin, Oktober 2005, Proceedings S. 93-100.
11. Motamed, C.; Wallart, O. „A Cooperative Distributed Vision Algorithm for Wide Area Vehicle Tracking“. Laboratoire LASL, Université du Littoral Côte D'Opale, Bat 2, 50 Rue F. Buisson, 62228 Calais, France. 2003
12. Kogut, G.; Trivedi, M. „A Wide Area Tracking System for Vision Sensor Networks“. Submitted to the 9 th World Congress on Intelligent Transport Systems, Chicago, Illinois, October, 2002 Computer Vision and Robotics Research Laboratory (CVRR)
13. Ploetner, J.; Trivedi, M. „A Multimodal Framework for Vehicle and Traffic Flow Analysis“. *IEEE International Intelligent Transportation Systems Conference*. September 2006
14. Turner, S.; Eisele, W.; Benz, R.; Holdener, D. „Travel Time Data Collection Handbook“. Report FHWA-PL-98-035. U.S. Department of Transportation, Federal Highway Administration. March 1998.
15. Chachich, A.; Pau, A.; Barber, A.; Kennedy, K.; Olejniczak, E.; Hackney, J.; Sun, Q.; Mireles, E. „Traffic Sensor Using a Color Vision Method“, *Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, SPIE Proc. Vol 2902, pp 156-165
16. Takaba, S.; Morita, T; Hada, T. „Estimation and Measurement of Travel Times By Vehicle Detectors and License Plate Readers“, *Vehicle*

navigation and Information Systems Conference Proceedings, Society of Automotive Engineers, 1 257-267. 1991

17. Bertini, R.L.; Lasky, M.; Monsere, C. „Validating Predicted Rural Corridor Travel Times from an Automated License Plate Recognition System: Oregon’s Frontier Project.” 8th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria. September 13-16, 2005.
18. Busch, C.; Dörner, R.; Freytag, C.; Ziegler, H. „Feature Based Recognition Of Traffic Video Streams“. Vehicular Technology Conference, 1998. VTC 98. 48th IEEE Volume 3, Issue , 18-21 May 1998 Page(s):1790 - 1794 vol.3
19. Tanaka, Y. „Travel-Time Data Provision System Using Vehicle License Number Recognition Devices“. Intelligent Vehicles '92 Symposium., Proceedings of the. 29 Jun-1 Jul 1992 Page(s):353 – 358
20. Trivedi, M.; Gandhi, T.; Huang, K. „Distributed Interactive Video Arrays for Event Capture and Enhanced Situational Awareness“. IEEE Intelligent Systems 20(5): 58-66. 2005
21. Antone, M.; Arambel, P.O.; Douglas, J.; Washburn, B. „Feasibility Study of a video-Based Urban Traffic Tracking System“. IEEE International Conference on Systems, Man and Cybernetics. Volume 1, Issue , 5-8 Oct. 2003 Page(s): 709 - 716 vol.1
22. Woesler, R. „Real-time Recognition and Reidentification of Vehicles from Video Data with high Reidentification Rate“. In: Callaos, N. [Hrsg.]: Proceedings of the 8th World Multiconference on Systemics, Cybernetics

and Informatics (SCI 2004), IIS, S. 347 - 352, SCI 2004 - 8th World Multiconference on Systemics, Cybernetics, and Informatics, Orlando (USA), 2004.07.18 - 2004.07.21, ISBN 980-6560-13-2

23. Arr, G.; Sun, C.; Ramachandran, R. „Fusion of wavelet transform and color information features for automatic vehicle reidentification in intelligent transportation, A Pattern Recognition and Feature Fusion Formulation for Vehicle reidentification in ITS“. 2004
24. Zeng, N.; Crisman, J.D. „Vehicle matching using color“. Intelligent Transportation System, 1997. ITSC 97. IEEE Conference on. 9-12 Nov 1997 Page(s):206 - 211
25. Kamijo, S. Kawahara, T. Sakauchi. „Vehicle sequence image matching for travel time measurement between intersections“. Systems, Man and Cybernetics, 2005 IEEE International Conference on. Volume 2, 10-12 Oct. 2005 Page(s): 1359 - 1364 Vol. 2
26. Sun, C.; Arr, G.; Ramachandran, R.P. „An Investigation in the Use of Vehicle Reidentification for Deriving Travel Time and Travel Time Distributions“. 82nd Annual Transportation Research Board Meeting, Washington, DC. January 12-16, 2003
27. Sun, C.; Ritchie, S. G.; Tsai, K.; Jayakrishnan, R. „Use of Vehicle Signature Analysis and Lexicographic Optimization for Vehicle Reidentification on Freeways“. Transportation Research Part C: Emerging Technologies, Volume 7, Number 4, August 1999, pp. 167-185(19)
28. Kogut, G.; Trivedi, M. M. „Maintaining the Identity of Multiple Vehicles as They Travel Through a Video Network“. Submitted to IEEE Int'l Conf. On Intelligent Transportation Systems, Oakland, California. 2001

29. Lowe, D. G. "Object recognition from local scale-invariant features". *International Conference on Computer Vision, Corfu, Greece (September 1999)*, pp. 1150-1157.
30. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. „Robust wide baseline stereo from maximally stable extremal regions“. In Paul L. Rosin and David Marshall, editors, *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, London, UK, September 2002. BMVA.
31. Guo, Y.; Hsu, S.; Sawhney, H.S.; Kumar, R.; Rao, C. „Video Object Fingerprinting for Confirmatory Identification (CID)“. Sarnoff's Contribution to Large Area Multi-Target Tracking (MTT) in Aerial Videos under Phase I of the DARPA VIVID Program
32. Huang, T.; Russel, S. „Object Identification: A Bayesian Analysis with Application to Traffic Surveillance“. *Artificial Intelligence*, Volume 103, Number 1, August 1998, pp. 77-93(17)
33. Pasula, H.; Russell, S.; Ostland, M.; Ritov, Y. „Tracking many objects with many sensors“. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 1160-1171, Stockholm 1999.
34. Koubaroulis, D.; Matas, J.; Kittler, J. „Mns: A novel method for colour based object recognition and image retrieval“. Technical Report VSSP-TR-6/99, University of Surrey. Dec. 1999
35. Comaniciu, D.; Meer, P. „Robust Analysis of Feature Spaces: Color Image Segmentation“. In *IEEE Conf. Computer Vision and pattern Recognition*, (San Juan, Puerto Rico), pp. 750-755. June 1997
36. Hastie, T.; Tibshirani, R.; Friedman, J. „The Elements of Statistical Learning: Data Mining, Inference, and Prediction“. Springer. 2001

37. Comaniciu, D.; Meer, P. „Mean Shift Analysis and Applications“. Proc. IEEE Seventh Int'l Conf. Computer Vision, vol. 2, pp. 1197-1203, Sept. 1999
38. Koubaroulis, D. „The Multimodal Neighbourhood Signature for modelling object colour appearance and applications in computer vision“. PhD thesis, University of Surrey. October 2001
39. De Berg, M.; Schwarzkopf, O.; Van Kreveld, M.; Overmars, M. „Computational Geometry: Algorithms and Applications“. Springer. 2000.
40. Sedgewick, R. „Algorithmen in C“. Addison-Wesley. 1992
41. Finlayson, G.D.; Schiele, B.; Crowley, J.L. „Comprehensive Colour Image Normalization“. Proc. 5th European Conference on Computer Vision. 1998
42. Gevers, T.; Smeulders, A. „Color-based Object Recognition“. PR 32 (3). pp 453-464. 1999
43. Koubaroulis, D.; Matas, J.; Kittler, J. „Illumination Invariant Object Recognition using the Mns Method“. Proceedings of the 10th European Signal Processing Conference, pages 2173-2176, Tampere, Finland, Tampere University of Technology. September 2000
44. Gusfield, D.; Irving, R.W. „The Stable Marriage Problem: Structure and Algorithms“. MIT Press. 1989
45. Chetverikov, D.; Megyesi, Z.; Janko, Z.; Matas, J. "Using Periodic Texture as a Tool for Wide-Baseline Stereo". Proc. 26th Workshop of the Austrian Association for Pattern Recognition, Graz, 2002, pp.37-44
46. Arth, C. „Fahrzeugklassifikation unter Verwendung des ADA-Boost Algorithmus“. Diplomarbeit. Technische Universität Graz, September

2004

47. McFarlane, N. J. B.; Schofield, C. P. „Segmentation and Tracking of Piglets in Images“. *Machine Vision and Applications*, Vol. 8, pp 187-193. 1995

48. Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A. „Detecting Moving Objects, Ghosts, and Shadows in Video Streams“. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(10): 1337-1342. 2003