Diplomarbeit

Klassifizierung dynamischer Gesten im Kontext multimodaler Infotainmentsysteme

Ausgeführt in der Abteilung ZT-3 für Mensch-Maschine-Interaktion der
BMW Group Forschung und Technik
Technische Universität München
Fakultät für Informatik

Leonhard Walchshäusl

Aufgabensteller : Prof. Gudrun Klinker, Ph.D
Betreuer : Dr. Frank Althoff (BMW)

Abgabetermin : 15.12.2004

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst, und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den _____ Unterschrift: _____

DANKSAGUNG

Besonders möchte ich mich bei meinem Betreuer Dr. Frank Althoff für die von ihm geförderte freie Arbeitsweise und die hilfsbereite und freundliche Unterstützung beim Erstellen dieser Arbeit bedanken. Weiterhin danke ich dem MIAMII-Team, Verena Broy, Rudi Lindl, Stefan Hoch und Frank Althoff für die fruchtbaren Diskussionen zum Thema MMI, sowie allen Mitarbeitern der Abteilung ZT-3, die sich für die Gestenaufnahmen zur Verfügung gestellt haben.

Fürs Korrekturlesen und die liebevolle Fürsorge danke ich Miriam Häusler. Nicht zuletzt danke ich meinen Eltern. Ohne ihren uneingeschränkten Rückhalt wäre mein Studium und somit auch diese Diplomarbeit nicht durchführbar gewesen.

München, im Dezember 2004

INHALTSVERZEICHNIS

KAPITEL 1

EINLEITUNG

1.1. Motivation

Menschen treten unter Zuhilfenahme verschiedener Modalitäten, wie Sprache, Mimik und Gestik in Kontakt und tauschen Informationen aus. Seit Anbruch des Maschinenzeitalters wird untersucht, inwieweit sich diese elementare zwischenmenschliche Form der Kommunikation auf die Interaktion von Menschen mit Maschinen übertragen lässt. Erklärtes Ziel der Forschung ist eine größtmögliche Entlastung des Anwenders hinsichtlich Fragestellungen zur Bedienung. Der Nutzer könnte sich somit vorrangig auf die maschinelle Lösung der Aufgabe konzentrieren. Neben der technischen Umsetzung liegt der Schlüssel zur Verwirklichung dieser Vision in multimodalen Erkennersystemen: Nur die Fusion von visuellen, auditiven und taktilen Benutzerinformationen sowie deren Interpretation im Kontext der aktuellen Situation ermöglichen eine ganzheitliche Analyse.

Gegenwärtig dominieren vornehmlich taktile Eingabegeräte wie Tastatur und Maus die Modalitätenlandschaft. Sie stellen nicht nur eine artifizielle Form der Kommunikation dar, sondern müssen auch vom Anwender durch aufwendiges Training erlernt werden. Wesentlich natürlicher ist eine verbale Steuerung. Stand der Technik sind befehlsorientierte „push to talk“-Systeme, die bereits erfolgreich in kommerziellen Produkten, wie beispielsweise im Fahrzeug, genutzt werden. Im Gegensatz zu natürlichsprachlichen Systemen wird dabei die meist kommandoartige Spracheingabe durch einen Knopfdruck initiiert werden.

Gestik als wichtiger Bestandteil eines multimodalen Dialogs rückte erst in den letzten Jahren in das Blickfeld der Forschung. Zuvor waren die verfügbaren Rechensysteme nicht leistungsfähig genug, die hohe Informationsdichte des visuellen Kanals in Echtzeit auszuwerten. Vergleichbar zur Sprache stellt Gestik eine natürliche, intuitive und effiziente Eingabeform dar.

Vergleichsweise wenig Beachtung fand bisher ein möglicher Einsatz von Gestik im automotiven Umfeld, da der Schritt vom Labor in das Fahrzeug besondere Ansprüche an die Technik stellt. Neben mobilen und zugleich kostengünstigen Hardwarekomponenten, die erst seit kurzem mit ausreichender Leistung verfügbar sind, muss ein automotives Erkennersystem insbesondere mit dynamischen Umweltbedingungen zurecht kommen.

Zur Steuerung von Infotainment-Systemen kommt speziell im Automobil das große Potential einer gestenbasierten Eingabe zum Tragen. Infolge des enormen Zuwachses an elektronischen Zusatzeinrichtungen im Kraftfahrzeug, wird deren Beherrschung mittels Knöpfe und Schieberegler zunehmend komplexer. Ferner ist der verfügbare Bauraum für haptische Bedienelemente stark begrenzt. Abhilfe schafft die Steuerung über Sprach-, aber auch über Gestenkommandos. Wegbereiter für einen multimodalen Dialog wäre die Fusion dieser beiden Eingabemedien. Gesten könnten dabei zum einen Verbales verdeutlichen und die Erkennung durch das System somit robuster gestalten – analog zu Konversationen zwischen Menschen, in welchen Körpersprache einen wesentlichen Teil zur Semantik beiträgt. Zum anderen könnte Gestik als „Fallback“- bzw. alternative Modalität fungieren, wenn Spracheingabe wegen aktueller Umgebungsbedingungen, wie beispielsweise Nebengeräusche bei einer Cabriofahrt, nicht einsetzbar ist.
Während viele Anwender Präferenzen setzen hinsichtlich der Art und Weise, wie sie mit dem Fahrzeug interagieren *möchten*, gibt es Menschen, die aufgrund von körperlichen Einschränkungen manche Eingabeformen nicht in Anspruch nehmen *können*. Gesten erweitern somit das mögliche Interaktionsspektrum um eine weitere Eingabemodalität.
Es ist den Automobilherstellern ein Anliegen von oberster Priorität die Fahrersicherheit zu erhöhen. Da Gesten weitgehendst ohne optische Vergewisserung ausführbar sind, kann während der Interaktion mit den automotiven Infotainment-Systemen der Blick des Fahrers weiter auf dem Verkehrsgeschehen ruhen. Untersuchungen [Gei03] haben des Weiteren gezeigt, dass gestische im Vergleich zur taktilen Bedienung eine geringere mentale Ablenkung des Fahrers verursacht, sofern das System explizit auf eine Interaktion mittels Gestik zugeschnitten ist.

1.2. Zielsetzung

Ziel dieser Arbeit ist die Konzeption und Entwicklung eines Softwaresystems zur bildbasierten, maschinellen Erkennung dynamischer Gesten. In erster Linie werden dabei Handgesten untersucht, wobei ebenfalls diskutiert wird, inwieweit sich die betrachteten Techniken zur Erkennung von Kopfgesten eignen würden. Besonders in der Fahrzeugdomäne bergen anfallende Sensordaten möglicherweise Potential zur Steigerung der Erkennungsraten. Weiteres Ziel ist es diese Zusatzinformation sowie externe Wissensquellen in den Erkennungsprozess zu integrieren und auf ihren Einfluss hin zu überprüfen. Sämtliche Erkennungsverfahren werden evaluiert und über eine Parametrisierung optimiert.

Auf dem Weg zur eigentlichen Intention dieser Arbeit – einer gestengestützten Interaktion im Automobil – wird das System zunächst in der Desktopdomäne entwickelt. Dies beinhaltet den Entwurf einer geeigneten Softwarearchitektur, die einen flexiblen Austausch einzelner

Systemkomponenten erlauben und die Teamarbeit fördern soll. Um Erkenntnisse für eine spätere Portierung des Systems in das automotive Umfeld zu sammeln, werden verschiedene Bildverarbeitungs- und Erkennungstechnologien prototypisch implementiert und gegenübergestellt. Zusätzlich wird ein mechanisches Gestell für den Kamera- und Lichtaufbau konstruiert, damit in der Laborumgebung ähnliche geometrische Bedingungen wie im Fahrzeug simuliert werden können.
Ein Demonstrator mit grafischer Benutzeroberfläche wird eine netzwerkgestützte Konfigurierbarkeit des Systems zur Laufzeit erlauben. Um den Gestenerkenner erlebbar zu machen, wird er in einem letzten Schritt an die BMW-interne Interaktionsplattform MIAMII [LMA+01] angebunden.

1.3. Randbedingungen

Zu Beginn der Diplomarbeit wurden verschiedene Vorbedingungen vereinbart, die den technischen und organisatorischen Rahmen der Arbeit abstecken.

1.3.1. Technische Randbedingungen

Dynamische Gesten sind anders als statische Gesten stets mit Bewegung verbunden. Um eine Bewegung noch als solche identifizieren zu können, sollte die Bildwiederholrate nicht unter 15 Bilder in der Sekunde fallen. Das zu entwickelnde System muss somit *Echtzeitansprüchen* genügen.
Im Gegensatz zu intrusiven Verfahren, die eine Lokalisierung der Hand durch Marker, Datenhandschuhe oder andere Positionierungssensoren erleichtern, erfolgt die Objektsuche berührungslos über den *visuellen Kanal*. Vorgabe ist das „Come as you are“-Paradigma[1] [Yon98] vieler Automobilhersteller.
Des Weiteren beschränkt sich die Hardwareausstattung auf *Standardkomponenten*. Als Bilderfassungssystem kommen handelsübliche Webcams zum Einsatz um deren Potential im Fahrzeug auszuloten. Die nötige Rechenleistung wird von aktuellen Desktop-PCs zur Verfügung gestellt. Dadurch wird zum einen der finanzielle Rahmen der Arbeit gewahrt und zum anderen die verfügbaren Speicher- und Rechenressourcen auf die potentielle Leistung von Bordcomputern in zukünftigen Fahrzeugen begrenzt.

1.3.2. Organisatorische Randbedingungen

Die vorliegende Arbeit ist Teil einer Projektarbeit bei der BMW Forschung und Technik GmbH, die aus zwei komplementären Diplomarbeiten besteht. Erklärtes Projektziel ist die Entwicklung eines Gesamtsystems für eine gestenbasierte Interaktion im automotiven Umfeld. Die

[1] Dem Anwender ist nicht zuzumuten, vor Fahrtantritt, Sensoren oder andere technische Vorrichtungen an seinem Körper zu fixieren.

Diplomarbeit von Rudi Lindl [Lin04] beschäftigt sich vorwiegend mit dem Spotting[2] dynamischer Gesten, wohingegen diese Diplomarbeit vor allem die Klassifizierung[3] dynamischer Gesten thematisiert. Um für das Gesamtsystem ein möglichst großes Spektrum an verschiedenartigen Segmentierungsverfahren zur Verfügung zu stellen, werden in beiden Arbeiten unterschiedliche Teilgebiete der räumlichen Segmentierung behandelt. Diese Arbeit betrachtet verstärkt Ansätze im dreidimensionalen Raum, wohingegen in der Arbeit von Rudi Lindl Möglichkeiten einer Segmentierung im nahen Infrarot Spektrum diskutiert werden. Die Implementierung des Gesamtsystems, die zugrundeliegende Softwarearchitektur und die Infrastruktur zum Transport von Kontextwissen sind im Rahmen des Projektes in Zusammenarbeit abgestimmt und umgesetzt worden.

1.4. Übersicht über die Arbeit

Kapitel 2 geht neben für die Arbeit wichtigen Begriffsklärungen und Definitionen auf den klassischen Ablauf eines Mustererkennungsprozesses ein. Anschließend wird anhand ausgewählter Systeme im kommerziellen und automotiven Umfeld der aktuelle Stand der Technik zum Thema Gestenerkennung dargestellt. Kapitel 3 betrachtet konkrete algorithmische Umsetzungen zur Lokalisierung von Hand- und Kopfregionen in Videobildern. In Kapitel 4 werden verschiedene Berechnungsverfahren zur regel- bzw. wahrscheinlichkeitsbasierten Erkennung dynamischer Gesten diskutiert. Sämtliche vorgestellten Verfahren aus Kapitel 3 und 4 werden auf Leistungsfähigkeit, Robustheit und Implementierungsaufwand hin untersucht. Ein abschließendes Fazit identifiziert die potentialträchtigsten unter ihnen. Kapitel 5 behandelt das realisierte Gestensystem. Nach der zugrundeliegenden Softwarearchitektur werden Implementierungsdetails und Erweiterungen der in den Kapiteln 3 und 4 gewählten Verfahren vorgestellt, sowie die Integration von Wissensquellen in das Erkennersystem abgehandelt. Bilder und Skizzen zu den Versuchsaufbauten sowie ein Abschnitt zur Anbindung des Erkennersystems an die BMW-interne MMI Versuchsplattform MIAMII beschließen das Systemkapitel. Die Leistungsfähigkeit des Gestenerkennersystems wird in Kapitel 6 evaluiert. Dabei werden die Erkennungs- und Fehlerraten des Systems sowohl im Labor als auch im Fahrzeug jeweils mit und ohne Einfluss von Wissensquellen verglichen. In Kapitel 7 werden die wichtigsten Ergebnisse zusammengefasst und ein Ausblick hinsichtlich möglicher Erweiterungen und Verbesserungen gegeben. Im Anhang finden sich zusätzliche Diagramme zur Evaluierung und eine Beschreibung des Gestendemonstrators.

[2] Das Herausfiltern von bedeutungstragenden Handlungen aus einer Folge von beliebigen Bewegungen.
[3] Vorgang oder Methode zur Einteilung von Objekten in Klassen oder Kategorien.

KAPITEL 2

GRUNDLAGEN

In diesem Kapitel werden alle grundlegenden Begriffe und Verfahren erläutert, die in dieser Diplomarbeit von elementarer Bedeutung sind. Auf viele der hier präsentierten Definitionen und Erkenntnisse wird in den nachfolgenden Kapiteln Bezug genommen und dann als bekannt vorausgesetzt.
Da Spotting[1] und Klassifizierung[2] dynamischer Gesten nahezu identische Grundlagen besitzen, wurde nachfolgendes Kapitel gemeinsam mit Rudi Lindl [Lin04] verfasst.

2.1. Gestik

In vielen Bereichen lassen sich Erkenntnisse aus der Grundlagenforschung bzw. Vorbilder aus der Natur direkt für eine maschinelle Umsetzung verwenden. Deshalb ist es auch bei der Thematik dieser Arbeit wichtig, die Definitionen sowie die grundlegenden Eigenschaften der Gestik genauer zu betrachten. Im Folgenden kann jedoch nur ein grober Überblick gegeben werden. Dem interessierten Leser sei [Ken89, Ken96], [VPH97] und [Ekm95] als weiterführende Literatur empfohlen.

2.1.1. Definition

In der Literatur existieren, je nach Situation und Fachrichtung, viele unterschiedliche Definitionen von „Gestik“. Eine Grundaussage scheint zu sein, dass eine Veränderung im Erscheinungsbild einer Person eine Wirkung bei einer anderen Person hervorruft [Ekm95]. Aus

1 Das Herausfiltern von bedeutungstragenden Handlungen aus einer Folge von beliebigen Bewegungen.
2 Vorgang oder Methode zur Einteilung von Objekten in Klassen oder Kategorien.

dieser allgemeinen Definition lässt sich, außer der Informationsübertragung mittels des visuellen Kanals, keinerlei Verfahren für das maschinelle Erkennen und Spotten von Gesten ableiten. Kendon [Ken89, Ken96] hat sich in seinen Arbeiten und Experimenten intensiv damit beschäftigt, wie sich Gesten definieren lassen und vom Menschen erkannt werden. Folgende Aspekte seiner Definition[3] sind hierbei von Interesse:

1. Gesten entsprechen einer sichtbaren Handlung bzw. Bewegungen des Körpers oder von Körperteilen.
2. Die Erkennung und das Spotting von Gesten wird intuitiv und routinemäßig vom Menschen durchgeführt.
3. Gesten sind immer von kommunikativer Natur.

2.1.2. Charakteristische Eigenschaften

Besonders Punkt zwei aus dem vorherigen Abschnitt, der das Spotting bzw. die Klassifizierung beschreibt, ist noch sehr allgemein gehalten. An dieser Stelle liefert Kendon [Ken89, Ken96] in mehreren Experimenten genauere Erkenntnisse. Er entdeckte, dass Menschen mit großer Sicherheit Bewegungen als Gesten identifizieren können, obwohl ihnen weder die Semantik, noch die Ausführung der gezeigten Gesten bekannt ist. Ausgehend von diesem Experiment wurden folgende Punkte als Entscheidungskriterien für das Spotting bzw. als grundlegende Eigenschaften von Gesten ermittelt:

1. Kopfbewegungen (Nick- bzw. Rotationsbewegungen) müssen schnell oder wiederholt ausgeführt werden und wieder in der Ausgangsposition enden, um als Gesten erkannt zu werden. Findet gleichzeitig eine Bewegung der Augen statt, so stellt dies hingegen eine Kontraindikation dar.
2. Bewegen sich Gliedmaßen abrupt vom Körper fort und kehren anschließend wieder in die Ausgangslage zurück, so wird eine Geste wahrgenommen.
3. Ganzkörperbewegungen werden nur dann als Gesten erkannt, wenn sie wieder in die Ausgangsposition zurückkehren und nicht zu einer geänderten Körperhaltung bzw. -position führen.
4. Bewegungen, die als Gesten erkannt werden, weisen eine Art Symmetrie auf. Probanden konnten rückwärts abgespielte Gesten immer noch als solche erkennen.

Zusammenfassend lässt sich konstatieren, dass Gesten zum einen an einer schnellen und abrupten Bewegung und zum anderen an einer spezifischen Trajektorie[4] erkannt werden können. Zusätzlich weisen Gesten einen gewissen symmetrischen Charakter auf.

[3] Original Definition: *The word 'gesture' serves as a label for that domain of visible action that participants routinely separate out and treat as governed by an openly acknowledged communicative intent.*

[4] Die Endposition der Geste entspricht ungefähr der Anfangsposition.

2.1.3. Taxonomie

In der Literatur wurden mehrere Möglichkeiten zur Einteilung von Gesten vorgestellt, die psychologische Aspekte berücksichtigen. Kendon unterscheidet hier zwischen von der Sprache unabhängigen „autonomen Gesten“ und „Gestikulation“, die in Verbindung mit der Sprache auftritt. Eine für die Mensch-Maschine-Kommunikation weitaus sinnvollere Einteilung ist in Abbildung 2.1 zu sehen und wurde von Quek [Que94] entwickelt und von Pavlovic [VPH97] erweitert.

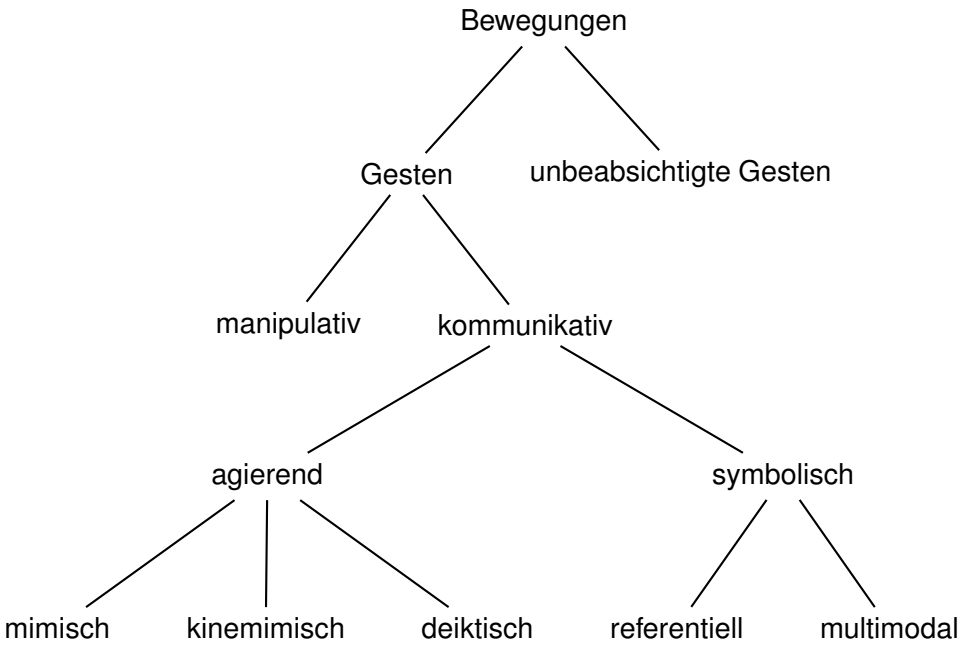


Abbildung 2.1.: Verschiedene Klassen von Gesten [VPH97].

Bewegungen des Körpers, speziell der Hand bzw. des Armes, können grob in *Gesten* bzw. *unbeabsichtigte Bewegungen* unterteilt werden. Gesten setzen sich wiederum aus *manipulativen* und *kommunikativen* Gesten zusammen. Manipulative Gesten sind unmittelbar bzw. haptisch mit einem Objekt oder Gegenstand verbunden und verändern hauptsächlich dessen Position oder Orientierung. Im Fahrzeug treten sehr häufig derartige Gesten, wie zum Beispiel Lautstärkenänderung oder Gangwechsel, auf. In einer gestenbasierten MMI sind jedoch nur kommunikative Gesten von Relevanz, da sie eine Intention übermitteln. Sie können noch weiter in *agierende* und *symbolische* Gesten untergliedert werden. Bei den agierenden steht vor allem die Interpretation der Bewegung an sich im Vordergrund; so imitieren die *mimischen* Gesten wohlbekannte Handlungsabläufe wie beispielsweise Greifen oder das Abnehmen eines Telefonhörers. *Deiktische* Gesten repräsentieren hingegen im Allgemeinen relative Zeigegesten, wohingegen *kinemimische* Gesten hauptsächlich Richtungsgesten darstellen. Im Gegensatz zu *agierenden*- spielen *symbolische* Gesten in der Linguistik eine große Rolle. Entweder werden sie *multimodal* verwendet, um beispielsweise eine sprachliche Aussage zu

untermauern oder dienen als symbolische *Referenz* auf eine andere Handlung, einen Gegenstand oder eine Eigenschaft, wie zum Beispiel die Geldgeste oder „einen Vogel zeigen“. Der Gestenkatalog, der in dieser Arbeit verwendet wird, setzt sich aus *kinemimischen*, *mimischen* und *referentiellen* Gesten zusammen (siehe auch Abschnitt 5.6 auf Seite 74).

2.2. Spotting

Spotting bzw. zeitliches Segmentieren bezeichnet das Auffinden interessanter Informationen bzw. Bereiche in einem kontinuierlichen Datenstrom. Im Zusammenhang mit der Gestenerkennung repräsentieren die Anfangs- bzw. Endpunkte von Gesten die relevanten Informationen und die Videosequenz den Datenstrom (siehe Abbildung 2.2). Ausgehend von dem Start- und Endpunkt ist es Aufgabe eines nachgeschalteten Klassifikators die aufgetretene Geste zu bestimmen (siehe Abschnitt 2.3). Derartige Ansätze, in der Spotting eine Vorstufe der Klassifizierung darstellt, werden „2-Stage-Spotter“ genannt, wohingegen integrale Vorgehensweisen, die Spotting und Klassifizierung kombinieren und in einem Schritt sowohl zeitliche Position als auch das Klassifizierungsergebnis ausgeben, „1-Stage-Spotter“ genannt werden.

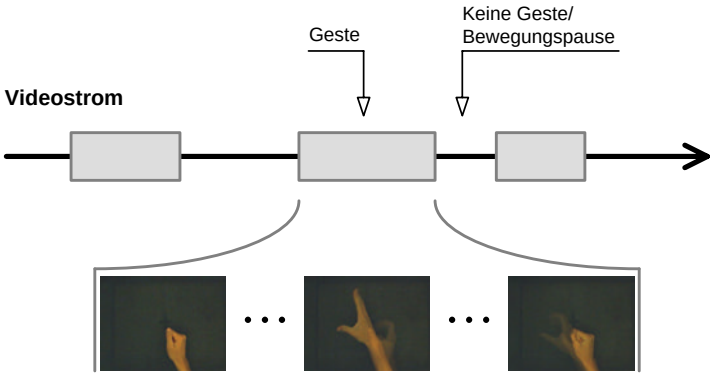


Abbildung 2.2.: Spotting von Gesten in einem kontinuierlichen Strom an Bildern [ML98].

2.3. Klassifizierung

Klassifizierung im Kontext der Mustererkennung bezeichnet den Vorgang, verschiedene Objekte oder Prozesse anhand ihrer Attribute bzw. Eigenschaften, die in dem Merkmalsvektor $\vec{m}$ zusammengefasst werden, in Klassen einzuordnen. Ein maschinelles Verfahren, das diese Einteilung automatisiert, wird als Klassifikator bezeichnet.

Es lässt sich eine grobe Unterteilung in dynamische und statische Klassifizierung treffen. Erstere ermöglicht die Einordnung von temporalen Prozessen der Länge T, modelliert durch eine Sequenz von Merkmalsvektoren $\{\vec{m}_1, ..., \vec{m}_T\}$. Typisches Anwendungsgebiet in Form von

HMM's[5] oder DTW[6] ist die Spracherkennung. Statische Klassifizierung hingegen unterscheidet Objekte anhand eines einzelnen Merkmalsvektors ($T = 1$) und findet beispielsweise in der Personenidentifikation in Form von SVM[7] oder Neuronalen Netzen Anwendung [WF01]. Die zugrundeliegenden Algorithmen der Klassifikatoren können je nach Art der Entscheidungsfindung in regel- und wahrscheinlichkeitsbasierte Vorgehensweisen unterteilt werden.
Dynamische Gesten erzeugen Folgen von Merkmalsvektoren und verlangen demnach eine dynamische Klassifizierung zur Erkennung. Sowohl probabilistische als auch regelbasierte Verfahren werden in dieser Arbeit thematisiert.

2.4. Bildverarbeitungspipeline

Die Erkennung dynamischer Gesten lässt sich zu einem Mustererkennungsprozess abstrahieren, der im Allgemeinen nochmals in mehrere Teilschritte untergliedert wird. Im speziellen Fall einer bildbasierten Mustererkennung wird diese Unterteilung als Bildverarbeitungspipeline bezeichnet. Vergleichbar mit dem Paradigma des ISO/OSI Schichtenmodells lässt sich durch die künstliche Unterteilung oftmals eine größere Flexibilität und Modularität, bei deutlicher Reduktion der Gesamtkomplexität, erreichen. Je nach Anwendungsgebiet kann es vorkommen, dass Schichten kombiniert, gelöscht oder hinzugefügt werden.
Im Folgenden werden die Aufgaben der einzelnen Schritte der Bildverarbeitungspipeline, wie sie in Abbildung 2.3 zu sehen ist, kurz erläutert.

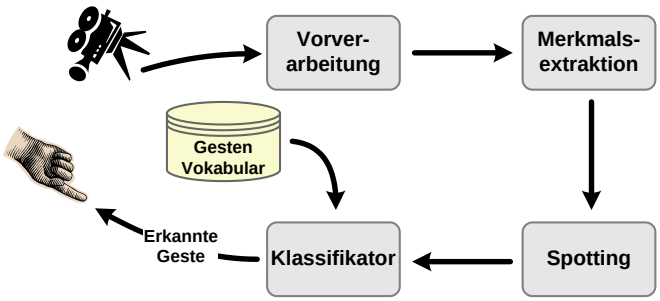


Abbildung 2.3.: Bildverarbeitungspipeline für das Erkennen und Spotten dynamischer Gesten.

In der Vorverarbeitung werden die analogen Bildsignale digitalisiert und gegebenenfalls durch Anwendung von Filtermaßnahmen wie zum Beispiel Rauschunterdrückung, Farbraumkonvertierung oder Kontrastverbesserung weiter aufbereitet.

Anschließend unterteilt ein Segmentierungsverfahren (für weitere Details zu den verschiedenen – in dieser Arbeit untersuchten – Verfahren sei auf Kapitel 3 verwiesen) das Bild in

[5] Hidden Markov Model.
[6] Dynamic Time Warping.
[7] Support Vector Machine.

Vorder- und Hintergrund, welche die Grundlagen für den nächsten Schritt in der Bildverarbeitungspipeline – der Merkmalsextraktion – darstellen.
Hier wird versucht, durch Kombination von charakteristischen Merkmalen des Vordergrundes, diesen in einer möglichst einfachen und kompakten Darstellung zu komprimieren und dadurch eine Datenreduktion zu erreichen. Auf Kosten der Approximationsgenauigkeit werden im Weiteren die Merkmale $\{m_1, ..., m_T\}$ verwendet, die für eine spätere Klassifizierung von Bedeutung sind. Diese Auswahl ist anwendungsspezifisch und kann zum Beispiel durch den Transinformationsgehalt [Hag97] ermittelt werden. Die Attribute $\{m_1, ..., m_N\}$ lassen sich zu dem Merkmalsvektor $\vec{m}$ zusammenfassen, der selbst Teil einer zeitlichen Sequenz von Merkmalsvektoren ist.

Bei einem kontinuierlichen Mustererkennungsprozess stellt sich das Problem der zeitlichen Segmentierung. Hierbei muss aus dem kontinuierlichen Strom an Merkmalsvektoren $\{\vec{m}_0, \ldots, \vec{m}_t\}$ der Startpunkt $\vec{m}_b$ sowie die Endposition $\vec{m}_e$ des zu klassifizierenden Objektes bestimmt werden. Im klassischen Fall stellen die einzelnen Attribute die Grundlage für diese Entscheidung dar, wobei die verwendeten Merkmale für das Spotting und die Klassifizierung nicht zwingend identisch sein müssen.
Ausgehend von dem Spottingergebnis muss in der letzten Stufe des Mustererkennungsprozesses der Klassifikator entscheiden, zu welcher vordefinierten Klasse $k \in K$ die Eingangssequenz $\vec{s} = \{\vec{m}_b, \ldots, \vec{m}_e\}$ gehört. Hierbei enthält K die unterscheidbaren Muster inklusive einer Garbage Klasse, als Repräsentant für Merkmalssequenzen, die nicht eingeordnet werden können.

$$\vec{s} \longmapsto k \in K \tag{2.1}$$

Die verschiedenen Ansätze zum Spotting dynamischer Gesten werden in dieser Arbeit nur am Rande thematisiert. Eine detaillierte Betrachtung und Evaluierung verschiedener Spotter findet sich in der Diplomarbeit von Rudi Lindl [Lin04].

2.5. Kontextwissen

Speziell in der Fahrzeugdomäne fallen eine Vielzahl von Sensorinformationen an, deren Interpretation Kontextwissen bildet (siehe Abbildung 2.4). In dieser Arbeit werden die Begriffe Wissensquelle und Kontextwissen synonym verwendet und lassen sich im automotiven Umfeld wie folgt unterteilen [LMA+01]:

- Der *Systemkontext* liefert Informationen über die Verfügbarkeit einzelner Module bzw. den aktuellen Systemzustand wie beispielsweise die Menüebene, Dialogstruktur, Geschwindigkeit, etc.

- Der *Umgebungskontext* vereinigt alle Informationen, die direkt aus der Umgebung abgeleitet werden können (Lichtverhältnisse, Geräuschkulisse, etc.).

- Informationen, die vom Nutzer deduziert werden können bzw. direkt mit ihm in Zusammenhang stehen, werden unter dem Begriff *Benutzerkontext* zusammengefasst. Zum Beispiel Vorlieben des Anwenders, Historie der letzten Benutzung, Geschlecht, Alter, etc.

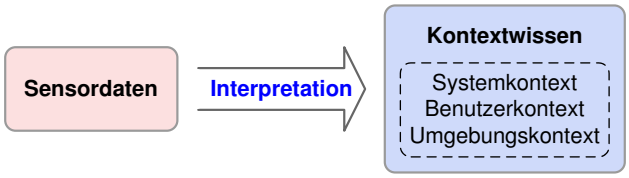


Abbildung 2.4.: Kontextwissen wird durch die Interpretation von Sensordaten erzeugt.

Die Integration von Kontextwissen in Module Spotting und Klassifzierung birgt möglicherweise Potential zur Steigerung der Erkennungsleistung bzw. Fehlerresistenz. Hierfür wird die Bildverarbeitungspipeline aus dem vorherigen Abschnitt um ein weiteres Modul erweitert, das sowohl den Spotter als auch den Klassifikator mit externem Kontextwissen versorgt (siehe Abbildung 2.5).

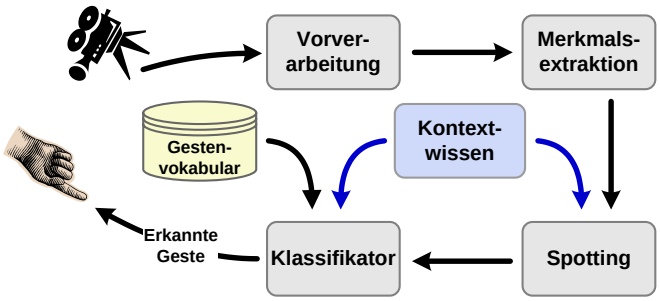


Abbildung 2.5.: Durch Kontextwissen erweiterte Bildverarbeitungspipeline

2.6. Stand der Technik Gestenerkennung

Viele wissenschaftliche Arbeiten thematisieren eine Verbesserung der Mensch-Maschine-Kommunikation durch Einbeziehung zusätzlicher Modalitäten, wie Sprache oder Gestik. Dabei sind sowohl die zugrundeliegenden Anwendungsgebiete als auch die möglichen *Usecases* weit gefächert. Im Folgenden soll ein kurzer Abriss über schon existierende Verfahren aus dem Teilbereich der allgemeinen Gestenerkennung zum einen und über Systeme im speziellen automotiven Umfeld zum anderen gegeben werden.

Abbildung 2.6.: Sony's EyeToy zur gestenbasierten Bedienung von Computerspielen [Son].

2.6.1. Allgemeine Systeme

Gestenerkennung findet in sehr vielfältigen Gebieten Anwendung, die sich seit einiger Zeit nicht mehr nur auf wissenschaftliche Bereiche beschränken. Sony's EyeToy [Son] ist zum Beispiel das erste Gestenerkennersystem, das erfolgreich kommerziell vermarktet und vertrieben wird. Durch eine einfache Webcam, die als Erweiterung für die Play Station 2 erhältlich ist, wird dem Anwender eine zusätzliche Modalität zur Interaktion mit Computerspielen eröffnet. Der Spieler kann mittels einfacher Gesten[8] bzw. Körperbewegungen[9] direkt in das Spielgeschehen eingreifen. Es existieren eine Vielzahl an unterschiedlichen Spielen und Anwendungen, obwohl lediglich eine triviale Bewegungsdetektion eingesetzt wird.

Neben dem Konsumerbereich wird Gestenerkennung häufig für Simultanübersetzungen von Zeichensprachen verwendet. So ist es mit dem von Zieren et al. [JZA02] präsentierten System möglich, mittels einer Frontalaufnahme von Personen, 152 unterschiedliche Gesten der deutschen Gebärdensprache zu detektieren. Hauptgesichtspunkt dieser Arbeit war es ein robustes Segmentieren bzw. Tracken bei unterschiedlichen Hintergründen bzw. zahlreichen Okklusionen der Hand, zu gewährleisten. Dieses Ziel wurde durch die Verwendung eines Kalman-Filters und Mean-Shift-Trackers erreicht, wobei das Gesamtsystem Detektionsraten von 81% erreicht.

Ein weiteres großes Anwendungsgebiet, in dem Gesten als Mittel der MMI verwendet werden, bildet die Augmented Reality [AMLSA]. Zum Beispiel verwendet Morguet [ML97] manipulative und kinemimische Handgesten, um künstliche Objekte zu modifizieren bzw. in einer virtuellen Welt zu navigieren. Eine mehr wissenschaftliche Anwendung in der erweiterten Realität stellt die Simulation der 3D Struktur von sehr großen Molekülen dar, die von Zeller

[8] Zum Beispiel Wink- bzw. Greifgesten.
[9] Beispielsweise Boxen.

et al. [ZPD+97] thematisiert wird. Handgesten dienen in diesem System zur Steuerung und Eingabe für die virtuelle Umgebung.

2.6.2. Anwendungsgebiete im Fahrzeug

Systeme, die eine gestenbasierte MMI mit dem Fahrzeug thematisieren, sind weitaus seltener anzutreffen. Unterscheiden lassen sich diese vornehmlich in der verwendeten Erfassungstechnik.

Abbildung 2.7.: Studie eines visionären MMI Konzept im Fahrzeug mit Head Up Display von Alpern et al. [AM02]. Als Erkennungstechnologie wurde ein „Wizard of Oz“ verwendet.

In dem von Cairnie et al. [NCM00] vorgestellten System können mittels eines „Six Degree of Freedom“-Sensor[10] deiktische Gesten auf einem eingeblendeten virtuellen Bedienfeld zur Steuerung sekundärer[11], Fahrzeugfunktionen verwendet werden. Als Spotting findet ein zusätzlicher Knopf am Lenkrad Verwendung, der die jeweilige Geste initiiert. Audiosteuerung, Klimaeinstellungen, sowie externe Fahrzeuggeräte wie Schweibenwischer bzw. Fensterheber zählen hierbei zu den möglichen Interaktionsmöglichkeiten. Ein realer Systemeinsatz ist jedoch nicht praktikabel, da die intrusive Sensorik den Fahrer zu stark beim Fahren beeinträchtigen würde.

Das an der Technischen Hochschule Aachen [SA00] entwickelte nichtintrusive „iGest“ System zur Steuerung von Verkehrsfunknachrichten und E-Mail Funktionen verwendet zum einen eine Fusion aus Maximum Likelihood und Korrelationsklassifikatoren für sechs unterschiedliche statische Gesten und zum anderen HMMs für die Erkennung 16 dynamischer Gesten, wobei ein manuelles Spotting zum Einsatz kommt. Die Bilderfassung erfolgt über eine Infrarotkamera. Aufgrund unzureichender Rechenleistung und aufwendiger Algorithmik können

[10] Elektromagnetischer Transmitter in Verbindung mit einem am Zeigefinger befestigten Sensor.
[11] Scheibenwischer bzw. Fensterheber betätigen, Klimaeinstellungen, Audiosteuerung.

lediglich statische Gesten im Echtzeitbetrieb erkannt werden. Darüber hinaus sind die verwendeten Hardwarekomponenten zur Bilderfassung und Beleuchtung verhältnismäßig kostspielig.

Geiger [Gei03] wählt in seiner Arbeit keinen videobasierten Ansatz, sondern nutzt zur berührungslosen Erfassung der Hand ein Array von Infrarot Abstandsensoren. Neben reinen Handgesten können durch zusätzliche Sensoren am Sitz des Fahrers auch Kopfgesten klassifiziert werden. Der verwendete Gestenkatalog besteht hauptsächlich aus Richtungsgesten zur Navigation innerhalb der Menüstruktur bzw. zur Bedienung der Musikwiedergabe. Mimische Gesten finden als Shortcut Anwendung und mittels Regelungsgesten ist eine Einstellung der Lautstärke bzw. ein Scrollen in Listen möglich. Bei einer DTW-basierten[12] Klassifizierung von 12 unterschiedlichen Handgesten wurde eine personenunabhängige Erkennungsrate von 89% erreicht. Das verwendete Sensorfeld erreicht nicht die hohe Auflösungsgenauigkeit einer videobasierten Bildanalyse und limitiert so die Anzahl der differenzierbaren Gesten. Typische Probleme der bildbasierten räumlichen Segmentierung werden dadurch hingegen umgangen.

[12] Dynamic Time Warping.

KAPITEL 3

BETRACHTETE ANSÄTZE DER RÄUMLICHEN SEGMENTIERUNG

Unter *räumlicher Segmentierung* versteht man die vollständige Partitionierung eines Bildes in voneinander disjunkte[1] Bereiche [PNR93]. Ergebnis dieser Unterteilung sind Hintergrund- und Vordergrundobjekte. Letztere sind relevant für das weitere Vorgehen innerhalb der Bildverarbeitungspipeline [MG91, Pra91]. In der Handgestenerkennung im Besonderen zählen die Hände, definiert ab dem Handgelenk über die Handfläche bis zu den Fingerkuppen, zum Vordergrund. Alle übrigen Szenenobjekte, insbesondere andere Körperteile des Nutzers, Tische, Tastatur oder die Mittelkonsole im Fahrzeug, bilden den Hintergrund.

Ein maschineller Algorithmus, der das Segmentierungsproblem löst, ist bis heute nicht bekannt [Pra91, PNR93]. Lediglich für Fälle mit vordefinierten Einschränkungen (z. B. konstante Beleuchtung, statischer Bildhintergrund) können ansprechende Ergebnisse erzielt werden. In der Fahrzeugdomäne sind viele dieser Randbedingungen nicht gegeben und machen somit die räumliche Segmentierung zu einem anspruchsvollen Unterfangen [Yon98].

Es lässt sich eine grobe Unterteilung der Segmentierungstechniken in farb-, lage- und formbasierte Verfahrensklassen treffen. Folgende Abschnitte widmen sich Ansätzen aus jeder dieser Kategorien und deren Bewertung im Kontext der Gestenerkennung (mit dem Schwerpunkt auf Handgestik), sowohl in der Labor- als auch in der Fahrzeugumgebung.

3.1. Schwellwertsegmentierung

Die *Schwellwertsegmentierung* ist ein farbbasiertes, pixelorientiertes Segmentierungsverfahren. Pixelorientiert in dem Sinne, dass jeder Bildpunkt für sich betrachtet und isoliert von

[1] nicht überlappend.

seiner Umgebung segmentiert wird. Bilder im RGB-Farbraum beinhalten drei Farbkanäle und können mathematisch über eine diskrete Funktion $f_{rgb}[x,y] = (f_r[x,y], f_g[x,y], f_b[x,y])$ repräsentiert werden. Diese bildet Bildpunkte $[x,y]$ auf Farbtupel mittels der Funktionen $f_r[x,y]$, $f_g[x,y]$ und $f_b[x,y]$ ab, die für Pixel innerhalb des roten (r), grünen (g) bzw. blauen (b) Farbkanals Werte zwischen 0 und 255 annehmen. Da sich folgende Betrachtungen auf Grauwertbilder mit nur einem Farbkanal beziehen, muß das RGB-Farbbild entsprechend in ein Einkanalbild $g[x,y]$ konvertiert werden, was beispielsweise über die Extraktion eines der Farbkanäle ($g[x,y] = f_\Lambda[x,y]$ mit $\Lambda \in \{r,g,b\}$) erfolgen kann.
Liegt das Einkanalbild $g[x,y]$ vor, entscheidet ein Schwellwert s über die Segmentzugehörigkeit der jeweiligen Funktionswerte. Ergebnis ist eine neue binarisierte Bildfunktion $b[x,y]$, wobei ein Bildpunkt mit dem Wert 0 dem Hintergrund- und mit dem Wert 1 dem Vordergrundsegment angehörig ist. Diese beiden Segmente werden im Folgenden mit K_H respektive K_V bezeichnet.

$$b[x,y] = \begin{cases} 1 \text{ für } g[x,y] > s \\ 0 \text{ sonst} \end{cases} \tag{3.1}$$

Grundsätzlich ist auch eine Unterteilung in mehrere Segmente denkbar, wobei die Zahl der verwendeten Schwellwerte dementsprechend angepasst werden muss. Im Hinblick auf die Aufgabenstellung ist die Segmentierung in zwei Bereiche (Vordergrund- K_V und Hintergrundsegment K_H) jedoch ausreichend. Bilder mit mehr als einem Farbkanal ermöglichen eine multidimensionale Schwellwertbildung über *Verbundhistogramme*. Verfahren in diesem Bereich, insbesondere zur Segmentierung von hautfarbenen Regionen, werden in der Diplomarbeit von Rudi Lindl [Lin04] thematisiert.

(a) (b) (c)

Abbildung 3.1.: Schwellwertsegmentierung bei schwarzem Hintergrund, Eingabebild als 8 bit Graustufenbild (a), segmentiertes Bild mit einem Schwellwert von 80 (b), Histogramm des Eingabebildes in logarithmischer Skala (c).

3.1.1. Wahl des Schwellwertes

Eine geeignete Wahl des Schwellwertes ist der wesentliche Punkt für die Güte des Verfahrens. Optimal ist dabei ein Wert s, der eine möglichst exakte Trennung der Bildfunktion in die

beiden Segmente K_V und K_H realisiert. Um diese Schranke zu finden, existieren mehrere Möglichkeiten [Jäh93, GW87].

- Statischer Schwellwert

  Ein konstanter Schwellwert ist sinnvoll, falls a priori Annahmen über die Grauwertverteilung sowohl des Vorder- als auch des Hintergrundobjekts der zu segmentierenden Bildsequenzen getroffen werden können und diese über die gesamte Bildreihe unveränderlich sind.

- Histogrammanalyse

  Oftmals lassen sich die einzelnen Segmente anhand ihrer Grauwertverteilung im zugehörigen Histogramm[2] des Grauwertbildes identifizieren. Dabei ist dieses im Idealfall bimodal[3] (siehe auch Abbildung 3.1(c)) und erlaubt dadurch nach Funktionsglättung und Minimabestimmung eine Ermittlung des idealen Trennschwellwertes für Vorder- und Hintergrundsegment. Dieses Verfahren wird in der Diplomarbeit von Rudi Lindl [Lin04] detailliert betrachtet und evaluiert.

- Verfahren von Otsu [Ots79, Leh02]

  Die Idee für das Verfahren von Otsu beruht auf der Betrachtung von *Grauwertvarianzen*. Durch Variierung des Schwellwertes wird versucht, die Varianz zwischen den Segmenten K_V und K_H zu maximieren und innerhalb der Segmente zu minimieren.

  Ausgangslage bilden zwei Segmente $K_0(\hat{s})$ und $K_1(\hat{s})$ die durch den Schwellwert $\hat{s}$ getrennt werden. $p(g)$ sei die Auftrittswahrscheinlichkeit des Grauwertes $0 < g < G$, wobei G der maximale im Bild auftretende Grauwert ist. Die Wahrscheinlichkeiten P_0 und P_1 für das Auftreten von Bildpunkten in den beiden Segmenten K_0 und K_1 ergeben sich nun abhängig von $\hat{s}$ zu

$$P_0(\hat{s}) = \sum_{g=0}^{\hat{s}} p(g) \tag{3.2}$$

$$P_1(\hat{s}) = \sum_{g=\hat{s}+1}^{G} p(g) \tag{3.3}$$

  Dabei gilt $P_1 = 1 - P_0(\hat{s})$. Sei $\overline{g}$ der Mittelwert aller im Bild vorkommenden Grauwerte und $\overline{g_0}$ bzw. $\overline{g_1}$ die Mittelwerte innerhalb beider Segmente. Dann ergeben sich die Varianzen der beiden Klassen zu

$$\sigma_0^2(\hat{s}) = \sum_{g=0}^{\hat{s}} (g - \overline{g_0})^2 p(g) \tag{3.4}$$

[2] Häufigkeitsverteilung von Grauwerten.

[3] Im Gegensatz zu unimodalen Verteilungen zerfällt das Histogramm in zwei deutlich unterscheidbare Maxima, wobei ein Maxima die Grauwertverteilung des Hintergrunds und das andere die des Vordergrunds modelliert.

$$\sigma_1^2(\hat{s}) = \sum_{g=\hat{s}+1}^{G} (g - \overline{g_1})^2 p(g) \tag{3.5}$$

Die Varianz innerhalb beider Klassen wird durch die Summe der beiden Einzelvarianzen (3.4) und (3.5) gebildet.

$$\sigma_{in}^2(\hat{s}) = P_0(\hat{s}) \cdot \sigma_0^2(\hat{s}) + P_1(\hat{s}) \cdot \sigma_1^2(\hat{s}) \tag{3.6}$$

Die sogenannte Zwischenvarianz beider Klassen ergibt sich durch die Differenz von der Gesamtvarianz σ^2 und (3.6).

$$\sigma_{zw}^2(\hat{s}) = \sigma^2 - \sigma_{in}^2(\hat{s}) = P_0(\hat{s}) \cdot (\overline{g_0} - \overline{g})^2 + P_1(\hat{s}) \cdot (\overline{g_1} - \overline{g})^2 \tag{3.7}$$

Ziel ist es die Varianz σ_{in} innerhalb der Segmente zu minimieren und σ_{zw} zwischen den Segmenten zu maximieren. Dafür wird der Quotient $Q(\hat{s})$ aus (3.6) und (3.7) gebildet:

$$Q(\hat{s}) = \frac{\sigma_{zw}^2(\hat{s})}{\sigma_{in}^2(\hat{s})} \tag{3.8}$$

Der Wert $\hat{s}$, der den Quotienten (3.8) maximiert, bildet den idealen Schwellwert s hinsichtlich der Varianz beider Klassen. Dazu müssen alle möglichen Schwellwerte $0 < \hat{s} < G$ durchiteriert werden, was eine kubische Berechnungskomplexität zur Folge hat.

3.1.2. Nachbearbeitung

Bildrauschen, bedingt durch Störungen im Kamerasystem oder in der Übertragungsleitung bzw. durch Schattenwurf und stellenweise dunkle Pigmentierung der Haut, macht sich durch Lücken und raue Ränder in den Regionen bemerkbar. Um auf die Gestalt von Regionen Einfluss zu nehmen (siehe Abbildung 3.2), sind verschiedene Operatoren definiert, die als *morphologische*[4] *Operationen* bezeichnet werden. Darunter fallen auch die Operatoren Opening und Closing (siehe [Jäh93] für eine ausführliche Abhandlung über morphologische Operationen).

Operationen zur Bildbereinigung

- Schließen von Löchern

  Im Komplementbild werden alle Zusammenhangskomponenten entfernt, die an den Bildrand grenzen. Eine anschließende Oder-Verknüpfung des Komplementbildes mit dem Eingabebild hat ein Schließen aller Löcher zur Folge.

[4] Morphologie ≙ Lehre von den Formen.

- Opening

  Die Opening-Operation setzt sich aus den Basisoperationen Dilatation und Erosion zusammen und wird verwendet, um Einbuchtungen in Regionen zu öffnen und über einen schmalen Steg verbundene Regionen voneinander zu trennen.

- Closing

  Die Closing-Operation setzt sich zusammen aus einer Erosion gefolgt von einer Dilatation. Resultat sind geschlossene Einbuchtungen am Rand von Regionen und verschmolzene benachbarte Zusammenhangskomponenten.

(a) (b)

Abbildung 3.2.: Ungefiltertes Bild (a), Bereinigtes Bild (Geschlossene Löcher mit anschließendem Closing) (b).

Allein die Schwellwertsegmentierung stellt in der Regel nicht sicher, dass die gewonnenen Regionen ausschließlich die gewünschten Objekte beinhalten. Beim Segmentieren der Hand fällt meist der Unterarm, sofern dieser nicht durch ein dunkles Kleidungsstück bedeckt ist, in einen ähnlichen Grauwertbereich wie die Hand und wird somit in einer gemeinsamen Zusammenhangskomponente (ZK) abgelegt.

Geometrische Operationen

Zur Filterung des Handbereichs aus dem Gesamtbereich sind folgende Schritte notwendig: In Abbildung 3.3(a) repräsentiert Punkt C den Schwerpunkt der gefundenen Komponente. Durch diesen wird der Vektor $\vec{d}$ gelegt, dessen Richtung mittels der Orientierung des Bereichs vorgegeben wird. Berechnungsvorschriften der Merkmale Schwerpunkt und Orientierung werden in den Abschnitten 4.1.1 und 4.1.2 thematisiert. Ebenfalls durch den Schwerpunkt C werden die beiden Vektoren $\vec{g}$ und $\vec{h}$ gelegt, wobei die Orientierung durch den Winkel ω zwischen $\vec{d}$ und $\vec{g}$ bzw. $\vec{d}$ und $\vec{h}$ fest vorgegeben ist. Die Punkte G und H errechnen sich aus dem Schnitt der Vektoren $\vec{g}$ respektive $\vec{h}$ mit der Kontur der Gesamtregion.

Durch die beiden Schnittpunkte G und H wird eine Ellipse gelegt, wobei diese durch den Mittelpunkt C, die oBdA. lange Achse $\overline{CG}$ und die kurze Achse $\overline{CH}$ eindeutig definiert ist. Der Rand des Ellipsensektors $\theta(\vec{CG},\vec{CH})$ bildet nun die Schnittstelle zwischen Unterarm und Hand.

Wird der Arm noch weiter in den Szenenausschnitt geschoben ergibt sich eine verstärkte Translation des Schwerpunktes C in den Unterarmabschnitt der Gesamtkomponente und macht es erforderlich, die Filterung wiederholt zu durchlaufen. Abbruchkriterium für die Rekursion ist dabei das Erreichen eines empirisch gewählten Seitenverhältnisses ρ des flächenkleinsten, die gefilterte Komponente umschließenden Rechtecks [U. 95].

(a) (b) (c)

Abbildung 3.3.: Skizze zur Veranschaulichung der Filteroperation (a), Binarisiertes Bild vor der Unterarmfilterung (b), Segmentiertes Bild nach der Unterarmfilterung (c).

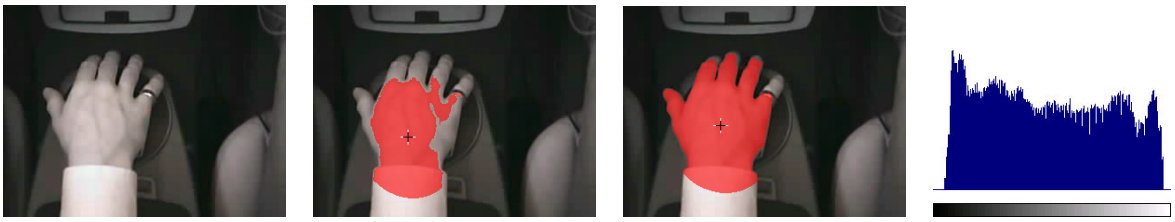
Bewertung

Inhärente Problematik aller vorgestellten Schwellwertverfahren ist das gelegentliche Fehlen einer globalen Schranke zur korrekten Trennung des Vordergrunds vom Hintergrund. Grund dafür ist, dass die Hand- bzw. Kopfregion oftmals nicht das einzige helle Objekt im Szenenausschnitt darstellt. Dies hat zur Folge, dass Bereiche des Hintergrunds fälschlicherweise mit in das Segmentierungsergebnis aufgenommen werden (siehe Abbildung 3.4).

(a) (b) (c) (d)

Abbildung 3.4.: Kein korrekter Trennschwellwert vorhanden, Kamerabild (a), Segmentiertes Bild (Schwellwert 168 ermittelt durch histogrammbasiertes Verfahren) (b), Segmentiertes Bild (Schwellwert 119 ermittelt durch Verfahren von Otsu) (c), Histogramm des Eingabebildes in logarithmischer Skala (d).

Dennoch lassen sich in der Desktopumgebung, selbst mit der statischen Schwellwertfindung, exzellente Ergebnisse erzielen, sofern sich die Grauwertverteilung der Vordergrundregion entsprechend stark vom Hintergrund abhebt und eine konstante Beleuchtung der Szene garantiert ist. In der Fahrzeugdomäne jedoch ist die Szenerie starken Schwankungen in der Lichtintensität unterworfen, was eine Adaption des Schwellwertes unumgänglich macht. In dieser Domäne hat sich gezeigt, dass seitens der dynamischen Verfahren der Algorithmus von Otsu in der Regel bessere Segmentierungsergebnisse liefert als das histogrammbasierte Verfahren (siehe Abbildung 3.5), insbesondere wenn keine strikt bimodale Grauwertverteilung vorliegt, was eine korrekte Minimasuche im Histogramm möglicherweise verhindert (vergleiche auch das Histogramm in Abbildung 3.1(c) mit 3.4(d)). Eine Verwendung zur räumlichen Segmentierung dynamischer Gesten scheitert dennoch wegen der zeitaufwendigen Berechnung (Komplexität von $O(n^3)$), die trotz Optimierungen, nicht mit der kontinuierlichen Verarbeitung der Bildfolgen in Echtzeit schritthalten kann.



(a) (b) (c) (d)

Abbildung 3.5.: Vergleich zwischen Verfahren von Otsu und histogrammbasiertes Verfahren, Kamerabild (a), Segmentiertes Bild (Schwellwert 156 ermittelt durch histogrammbasiertes Verfahren) (b), Segmentiertes Bild (Schwellwert 88 ermittelt durch Verfahren von Otsu) (c), Histogramm des Eingabebildes in logarithmischer Skala (d).

Aussichtsreichster Kandidat der Schwellwertalgorithmen für einen Einsatz im Fahrzeug bildet schlussendlich der histogrammbasierte Ansatz zur Schwellwertfindung, da er einen guten Kompromiss zwischen Segmentierungsqualität und Performanz darstellt.

3.2. Mustervergleich

Beim Mustervergleich (oder auch Template-Matching) wird das Objekt, welches zu finden ist, durch seine Grauwertmatrix beschrieben (d.h. diese bildet das *Muster* bzw. *Template*). Um nun das Muster zu lokalisieren, wird das Template über alle möglichen Positionen des Bildes geschoben und ein weiter unten noch zu definierendes *Ähnlichkeitsmaß* berechnet. Überschreitet der Wert des berechneten Ähnlichkeitsmaßes einen gewissen Schwellwert, so wird das Muster als erkannt angenommen [WLS02].

3.2.1. Summe der absoluten Grauwertdifferenzen

Sei $t[x,y]$ ein Templatebild, mit der Breite w und der Höhe h und $g[x,y]$ das Suchbild, dann erzeugt die Summe über die *Absolutbeträge der Grauwertdifferenzen*

$$a[x,y] = \frac{1}{w \cdot h} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} |t[u,v] - g[x+u, y+v]| \tag{3.9}$$

ein übliches Ähnlichkeitsmaß. Intuitiv ist klar, dass die Übereinstimmung zwischen dem gegebenen Bild und dem Template umso besser ist, je kleiner dieses Ähnlichkeitsmaß ist.

Daher gilt folgendes Vorgehen um ein Muster in einem Bild aufzufinden. Das Ähnlichkeitsmaß wird über das gesamte Bild berechnet und der im letzten Abschnitt erwähnte Schwellwert auf das Ähnlichkeitsbild angewendet. Um die exakten Positionen zu bestimmen, werden darauf lokale Minima bestimmt.

3.2.2. Grauwertkorrelation

Die *Grauwertkorrelation* (3.10) nimmt, im Gegensatz zum Ähnlichkeitsmaß der Grauwertdifferenzen, einen hohen Wert bei guter Übereinstimmung zwischen Template- und Suchbild an. Problematisch ist die große Abhängigkeit vom Kontrast des Bildes. So ist es möglich, dass ein Objekt, welches nicht matchen sollte, im Bild eine größere Korrelation hat als ein optisch gut passendes, da dessen Kontrast entsprechend kleiner ist. Ebenso wie die Summe der absoluten Grauwertdifferenzen ist die Grauwertkorrelation nicht beleuchtungsinvariant.

$$c[x,y] = \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} t[u,v] g[x+u, y+v] \tag{3.10}$$

Werden die Grauwerte jedoch auf den mittleren Grauwert bezogen und mit ihrer Standardabweichung normiert, können sowohl multiplikative als auch additive Beleuchtungsveränderungen ausgeglichen werden.

So ergibt sich die normierte Grauwertkorrelation zu

$$\check{c}[x,y] = \frac{1}{w \cdot h} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} \frac{t[u,v] - m_t}{\sqrt{\sigma_t^2}} \cdot \frac{g[x+u, y+v] - m_g(x,y)}{\sqrt{\sigma_g^2(x,y)}} \tag{3.11}$$

wobei m_t der mittlere Grauwert des Templates ist, $m_g(x,y)$ der mittlere Grauwert des Bildes in der Maske um den Bildpunkt $[x,y]$ und $\sqrt{\sigma_t^2}$ beziehungsweise $\sqrt{\sigma_g^2(x,y)}$ die entsprechenden Standardabweichungen sind. $\check{c}[x,y]$ hat den Wertebereich $[-1,1]$. Aus $\check{c}[x,y] = 1$ folgt $g(x+u, y+v) = l_1 \cdot t(u,v) + l_2$ mit $(l_1 > 0)$ ebenso für $\check{c}[x,y] = -1$ folgt $g(x+u, y+v) = l_1 \cdot t(u,v) +$

l_2 mit $(l_1 < 0)$. Die Übereinstimmung des Templates mit dem Ausschnitt des Suchbildes ist also bis auf eine multiplikative (l_1) und additive (l_2) Komponente gegeben.

(a) (b) (c)

(d)

Abbildung 3.6.: Template-Matching mit normierter Grauwertkorrelation als Ähnlichkeitsmaß. Suchbild (a), Korrelationsbild (je besser die Übereinstimmung des Templates desto heller der Grauwert) (b), Schwellwert von Korrelationsbild (rot) über abgeschwächtem Korrelationsbild visualisiert (c), Verwendetes Muster (d).

Bewertung

Im direkten Vergleich zwischen der Summe der Betragsdifferenzen und der Grauwertkorrelation als Ähnlichkeitsmaß ist der Vorteil der Summe der Betragsdifferenzen die schnelle Berechnung. Darüber hinaus lässt sich ein einfaches Abbruchkriterium finden, wenn der geforderte Schwellwert nicht mehr unterschritten werden kann. Dagegen ist der Berechnungsaufwand für die normierte Grauwertkorrelation hoch, jedoch ist sie im Gegensatz zur Summe der Betragsdifferenzen auch gegenüber multiplikativen Beleuchtungsänderungen invariant. Es lassen sich zwar auch Kriterien für den Abbruch der Berechnung finden wenn der geforderte Schwellwert nicht mehr überschritten werden kann, jedoch müssen dafür entweder die Standardabweichung oder die Korrelation (die nicht normierte Korrelation) über das gesamte Bild berechnet werden.

Im Gegensatz zur Segmentierung von starren Objekten (siehe Abbildung 3.6), ist die Segmentierung der Hand durch das Template-Matching-Verfahren nur bedingt geeignet. Die planar projizierte Ansicht kann stark variieren, da die Hand viele Bewegungsmöglichkeiten aufweist. Dies würde eine sehr grosse Anzahl verschiedener Muster erfordern um alle möglichen Handformen abzudecken, was zum Einen den Verwaltungsaufwand und zum Anderen die Wahrscheinlichkeit zufälliger Hintergrundübereinstimmungen drastisch erhöhen würde. Eine Seg-

mentierung durch den vorgestellten Mustervergleich wäre nur praktikabel, wenn von einer konstanten Handform (z. B. Zeigegeste) ausgegangen werden kann.

3.3. Stereo Segmentierung

Eine Segmentierung in drei Dimensionen basiert auf Kenntnis von Tiefeninformationen der Szene. Lokalisieren lässt sich das zu segmentierende Objekt dann anhand der Position im Raum. In der visuellen Bildverarbeitung sind unterschiedlichste Verfahren bekannt, die in der Lage sind ein Tiefenbild zu berechnen. Einteilen lassen sich diese in *aktive* und *passive Methoden*.

Aktive Methoden interagieren mit der Szenerie, beispielsweise durch Lichtprojektion. Zusätzlich zur Kamera ist demnach meist teure und technisch aufwendige Zusatzhardware erforderlich. Jedoch ist eine einzelne monokulare[5] Kameraaufnahme dann ausreichend um die räumliche Struktur zu rekonstruieren [Kap03].

Passive Methoden hingegen benötigen mehrere, entweder örtlich oder zeitlich versetzte, Aufnahmen der Szene. Sie erzielen nicht die hohe Präzision aktiver Verfahren, sind rechenintensiv und haben Schwierigkeiten mit der Verarbeitung schwach oder periodisch texturierter Bereiche. Für die grobe Segmentierung von Hand bzw. Kopf ist die Genauigkeit dennoch ausreichend und aktuelle PC-Systeme stellen die erforderliche Rechenleistung zur Verfügung. Sie lassen sich, insbesondere innerhalb des begrenzten Budget dieser Arbeit, ohne grössere Mehrkosten an Hardware prototypisch implementieren.

In den folgenden Abschnitten werden beide Methoden vorgestellt, wobei der Schwerpunkt auf den passiven Verfahren liegt.

3.3.1. Aktive Methoden

Stroboskop Differenzverfahren

Eine auf Softwareebene sehr einfach zu realisierende Methode stellt das *Stroboskop Differenzverfahren* dar (siehe auch [AAM03]). Dafür synchronisiert man eine schnell schaltende Lichtquelle mit dem Shutter einer Kamera. Abwechselnd erfolgt eine beleuchtete und eine unbeleuchtete Aufnahme der Szene. Vordergrundobjekte werden durch die Nähe zur Lichtquelle wesentlich stärker beleuchtet als Hintergrundobjekte. Somit können durch eine einfache Subtraktion zweier aufeinanderfolgender Bilder und anschließender Schwellwertbildung (siehe auch Abschnitt 3.1) im Binärbild alle Objekte, deren Beleuchtungsdifferenz stark genug war, ausgemacht werden (siehe Abbildung 3.7). Die Auflösung der resultierenden Tiefeninformation ist auf zwei mögliche Werte beschränkt, respektive nahes bzw. entferntes Objekt von der Lichtquelle.

[5] Systeme mit einer Kamera.

Abbildung 3.7.: Segmentierung mit Stroboskopverfahren (Bild 1: Unbeleuchtete Szene, Bild 2: Beleuchtete Szene, Bild 3: Differenzbild zwischen Bild 1 und Bild 2, Bild 3: Binarisiertes Bild, Bild 4: Segmentierungsergebnis) [AAM03].

Structured Light

Beim Structured Light Ansatz wird auf die, idealerweise abgedunkelte Szenerie, ein Muster, meist in Form von parallelen Linien, projiziert (siehe auch Bild links oben in Abbildung 3.8). Dies kann durch einen Projektor oder einen schnell oszillierenden Laser erfolgen. Projektoren erlauben meist nur ein geringes Sichtfeld und sind nicht so kompakt bzw. flexibel wie ein Lasersystem. Letzteres lässt sich auch im NIR[6] Bereich betreiben und macht die Projektion somit für den Nutzer unsichtbar. Im Gegensatz zu Projektoren müssen Lasersysteme mit der Kamera synchronisiert werden, damit die Abtastung nur erfolgt, wenn der Shutter der Kamera geöffnet ist. Falls zusätzlich zur Tiefeninformation noch die Textur des Objekts von Belang ist, muß in der Regel eine zweite Aufnahme ohne Laserprojektion durchgeführt werden (siehe auch Bild links unten in Abbildung 3.8). Liegt das Projektionsbild vor, erfolgt ein Mustererkennungsprozess, der das durch die Szenerie deformierte Projektionsmuster extrahiert (siehe auch Abbildung 3.8 (c)). Mit den bekannten Positionen von Kamera und Lichtquelle kann jetzt durch Triangulierung eine Punktewolke im $\mathbb{R}^3$ ermittelt werden. Lücken zwischen diesen Punkten werden durch Interpolation geschlossen (siehe auch Abbildung 3.8(d,e)) [Kap03].

Ergebnis ist eine überaus exakte und verlässliche Tiefeninformation der Szenerie. Selbst feine Strukturen wie die Lage einzelner Finger der Hand bzw. Lage der Augen und der Nase im Gesicht könnten damit ausgemacht werden und gewinnbringend für die Gestenerkennung eingesetzt werden.

3.3.2. Passive Methoden

Betrachtet man eine Szene aus mindestens zwei verschiedenen Perspektiven, lässt sich aus den planaren Projektionen durch *Korrespondenzfindung* und *Triangulierung* die grobe räumliche Struktur rekonstruieren. Wie man in Abbildung 3.9 sieht, wird bei einer Stereoaufnahme ein Punkt P_w im Weltkoordinatensystem über die jeweiligen Sehstrahlen auf ein Punktepaar (P_1, P_2) in die Projektionsebenen abgebildet. Über den Versatz und die Kamerageometrie (siehe Abschnitt zur Kalibrierung) berechnet sich somit die Lage des Punktes P_w. Sind alle Punktepaare bekannt, ermöglicht dies eine komplette Rekonstruktion der Tiefeninformation.

6 Nahes Infrarot (zwischen 800 und 1200 nm).

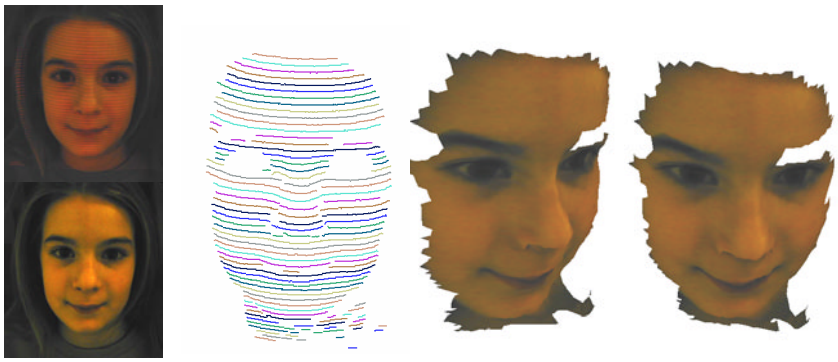


Abbildung 3.8.: Bild mit projiziertem Laserlicht (links oben), Bild ohne Structured Light (links unten), Extrahierte Linien (c), Texturiertes und errechnetes 3D Bild (d,e) [VS95].

In dieser Arbeit werden lediglich stereoskope Verfahren betrachtet. Polykulare[7] Systeme erzielen zwar oftmals robustere Tiefendaten, verursachen aber auch ein Vielfaches an Rechenleistung und Kosten.

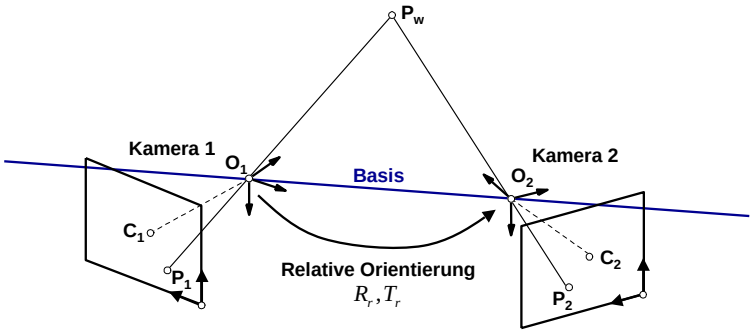


Abbildung 3.9.: Schematischer Kameraaufbau für ein Stereosystem [Ste02]

Übliches Vorgehen aktueller Algorithmen ist das Minimieren einer Kostenfunktion zur Bestimmung der Punktepaare. Eine globale Funktionsanalyse liefert exaktere Ergebnisse, benötigt aber entschieden mehr an Rechenzeit ($>$ 3 Sekunden pro errechnetes Tiefenbild). Für den Echtzeitbetrieb und somit für das Segmentieren dynamischer Gesten sind mit heutiger Standardhardware nur Verfahren verwendbar, die eine lokale Berechnung von Korrespondenzpaaren durchführen.

Kalibrierung des Stereoaufbaus

Ein Punkt P_w im Weltkoordinatensystem wird über die Abbildung π in einen Punkt P_c in das Kamerakoordinatensystem O_1 bzw. O_2 transformiert. Verschiedene Parameter p_i legen die

[7] System mit mehreren Kameras.

Abbildung π fest:

$$P_c = \pi(P_w, p_1, \ldots, p_n) \tag{3.12}$$

Aufgabe der *Kamerakalibrierung* ist es die Parameter p_i, festgelegt durch Objektiv und Kamera, innerhalb der Abbildung π zu bestimmen.

Aufteilen lassen sich die Werte $p_1, \ldots, p_n$ in äussere (extrinsische) und innere (intrinsische) Kameraparameter.

- Die starre Abbildung $P_c = (x\ y\ z)^T = RP_w + \vec{t}$ überführt durch die Rotationsmatrix R und den Translationsvektor $\vec{t}$ den Punkt P_w in den Punkt P_c mit

$$\vec{t} = (t_x\ t_y\ t_z)^T$$

$$R = R(\alpha)R(\beta)R(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  Das Tupel $(\alpha, \beta, \gamma, t_x, t_y, t_z)$ bildet somit die *äussere Orientierung* der Kamera in Bezug auf das Weltkoordinatensystem, wobei α, β und γ die Rotationswinkel um die x, y, bzw. z-Achse beschreiben und t_x, t_y und t_z die Translation in Richtung der jeweiligen Achse. Verwendet man homogene Koordinaten kann man den Rotationsvektor R und den Transformationsvektor $\vec{t}$ zur Matrix

$$M = \begin{pmatrix} R & \vec{t} \\ 0 & 1 \end{pmatrix}$$

  zusammenfassen. Die starre Abbildung vereinfacht sich nun zu $P_c = MP_w$.

- Parameter bezüglich der *inneren Orientierung* beschreiben die Brennweite ϕ der Lochbildkamera[8], radiale Verzeichnungen des Objektivs und Skalierungen der Bildebene s_x und s_y verursacht durch den Abstand der lichtempfindlichen Elemente auf dem CCD-Sensor[9].

  ϕ beschreibt die perspektivischen Verzerrungen einer Lochbildkamera. Es gilt die Projektion

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{\phi}{z} \begin{pmatrix} x \\ y \end{pmatrix}$$

  Radiale Verzeichnungen des Objektivs können als Transformationen in der Bildebene betrachtet werden und lassen sich durch die Formel

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{1 + \kappa(\hat{u}^2 + \hat{v}^2)} \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix}$$

  gut modellieren.

[8] Im Gegensatz zur telezentrischen Kamera gibt es ein Projektionszentrum, durch das alle Sehstrahlen laufen.

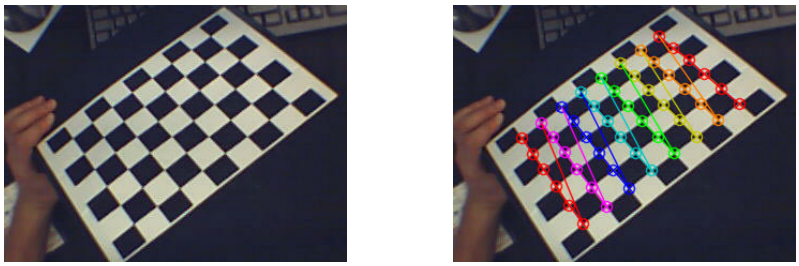
[9] charge coupled device.

Das Tupel $(\phi, \kappa, s_x, s_y, c_x, c_y)$ bildet damit die innere Orientierung einer Lochbildkamera.

Um die Kalibrierung durchführen zu können müssen hinreichend viele Punkte in Weltkoordinaten und ihre korrespondierenden Abbildungen in der Bildebene bekannt sein. Zur Automatisierung dieser Messung wird ein Kalibrierkörper verwendet, (siehe Abbildung 3.10) dessen Abmessungen bekannt sind. Dieser kann über eine einfache Schwellwertsegmentierung (siehe auch Abschnitt 3.1) extrahiert werden. Sei m_i der Positionsvektor eines Markers (visualisiert durch ein von einem Kreis umschlossenes Kreuz in Abbildung 3.10(b)) in Welt- und $\hat{m}_i$ in den korrespondierenden Bildkoordinaten, so gilt die Abbildung $\hat{m}_i = \pi(m_i, \vec{p})$. Die Parameter der inneren und äusseren Orientierung $\vec{p}$ werden nun durch Lösung des folgenden nichtlinearen Minimierungsproblems

$$d(\vec{p}) = \sum_{i=1}^{n} \|\hat{m}_i - \pi(m_i, \vec{p})\|^2 \quad \rightarrow \min \tag{3.13}$$

bestimmt mit $\vec{p} = (\phi, \kappa, s_x, s_y, c_x, c_y, \alpha, \beta, \gamma, t_x, t_y, t_z)$. Die Lösung erfolgt durch das *Marquardt-Levenberg-Verfahren* [Lev44, Mar63].



(a) (b)

Abbildung 3.10.: Linkes Kamerabild mit Kalibrierkörper (a), Extrahierter Kalibrierkörper (b).

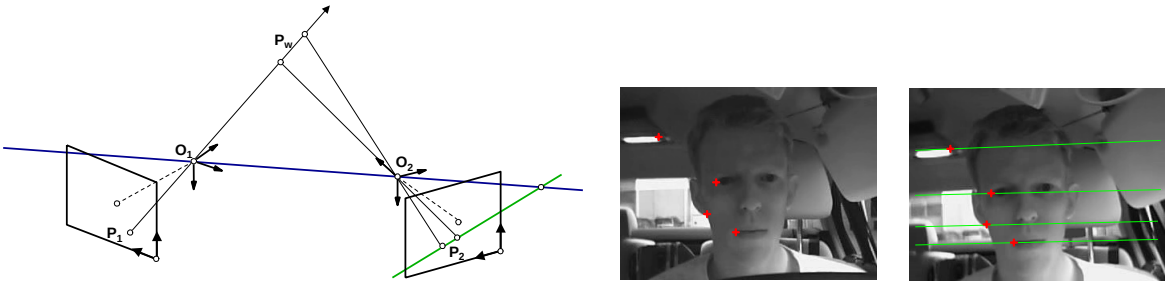
Sollen zwei Kameras in einem Stereoaufbau kalibriert werden, kann die relative Orientierung dieser aus den Einzelkalibrierungen ermittelt werden, sofern der Kalibrierkörper in beiden Bildern zeitgleich sichtbar ist. Dadurch wird aber keine eindeutige Bestimmung der extrinsischen Parameter gewährleistet, da die äusseren Orientierungen unabhängig voneinander errechnet wurden. Ein Erzwingen der Eindeutigkeit kann durch die Minimierung von

$$d(\vec{p}) = \sum_{i=1}^{n} \|\hat{m}_{i,1} - \pi(m_i, \vec{p})\|^2 + \|\hat{m}_{i,2} - \pi(m_i, \vec{p})\|^2 \quad \rightarrow \min \tag{3.14}$$

erreicht werden, wobei $\hat{m}_{i,1}$ die Positionen der extrahierten Marker im linken und $\hat{m}_{i,2}$ die Marker im rechten Kamerabild bezeichnen (siehe auch [Ste02]).

Epipolargeometrie

Die Suche nach korrespondierenden Bildpunkten muss nicht im gesamten Bild erfolgen, sondern lediglich entlang sogenannter Epipolargeraden. Passender wäre die Bezeichnung Epipolarlinie, da wegen radialer Bildverzerrungen, verursacht durch die Kameralinse, die Epipolargeraden in der Regel gekrümmt sind. Zusammen mit den Projektionszentren O_1 und O_2 spannt ein in der linken Bildebene gewählter Punkt P_1 eine Ebene auf. Da der zu rekonstruierende Punkt P_w im Weltkoordinatensystem auf dieser Epipolarebene liegen muss, kann sich die Suche nach dem korrespondierende Punkt P_2 auf die Epipolargerade, definiert durch den Schnitt von Epipolar- und Bildebene, beschränken (siehe Abbildung 3.11).



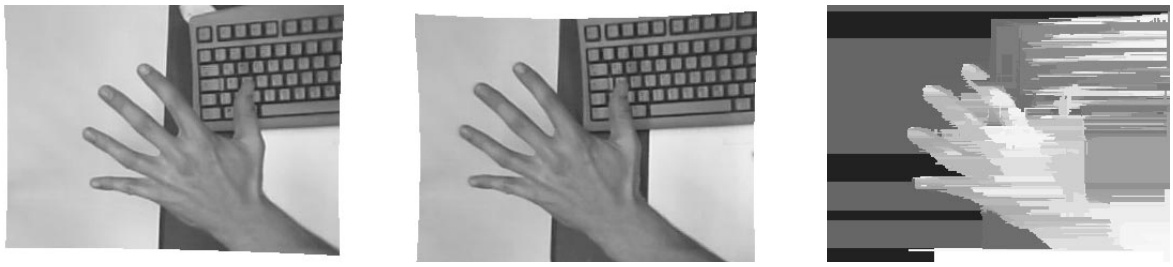
(a) (b) (c)

Abbildung 3.11.: Skizze zur Epipolargeometrie [Ste02] (a), korrespondierende Epipolarlinien der markierten Punkte von linkem (b) und rechtem Stereobild (c).

Aufwendig und rechenintensiv gestaltet sich die Konstruktion von Epipolargeraden, da sie für jeden Punkt von Neuem durchgeführt werden muss. Es existiert jedoch für beliebige Epipolarkonfigurationen eine sogenannte *Standardform*, in der die Bildebenen parallel zueinander und zur Basis sind, einen identischen Abstand zur Basis haben und keinerlei radiale Verzerrungen aufweisen. In dieser epipolaren Standardform sind ebenfalls die Epipolargeraden parallel zueinander und verlaufen in gleicher Höhe. Eine Korrespondenzfindung ist dann besonders effizient, da der Verlauf der Epipolargeraden, der Speicherorganisation heutiger Rechner entspricht (siehe [HZ04] für Details zur Epipolargeometrie). Das Reprojizieren beider Bildebenen in die epipolare Standardform ist nach der Kamerakalibrierung möglich und wird als *Rektifizieren* bezeichnet.

Stereoalgorithmus von Birchfield und Tomasi

Birchfield und Tomasi [BT99] stellen einen *Zweiphasen Stereoalgorithmus* vor. In einem ersten Matchingschritt werden grobe Disparitäten zwischen den Epipolargeraden der beiden Eingabebilder gesucht. Verglichen werden dabei allein die Grauwertintensitäten auf Pixelebene innerhalb einer Bildzeile und durch dynamisches Programmieren wird eine lokale Kostenfunktion minimiert. Ein zweiter Matchingschritt versucht nun die groben Disparitäten durch Verknüpfung der einzelnen Bildzeilen zu verfeinern (siehe Abbildungen 3.12 bzw. 3.14 für Beispiele).



(a) (b) (c)

Abbildung 3.12.: Linkes und rechtes Eingabebild in Standardform (a,b), errechnetes Tiefenbild mit dem Birchfield Algorithmus (c).

SMP-Stereoalgorithmus

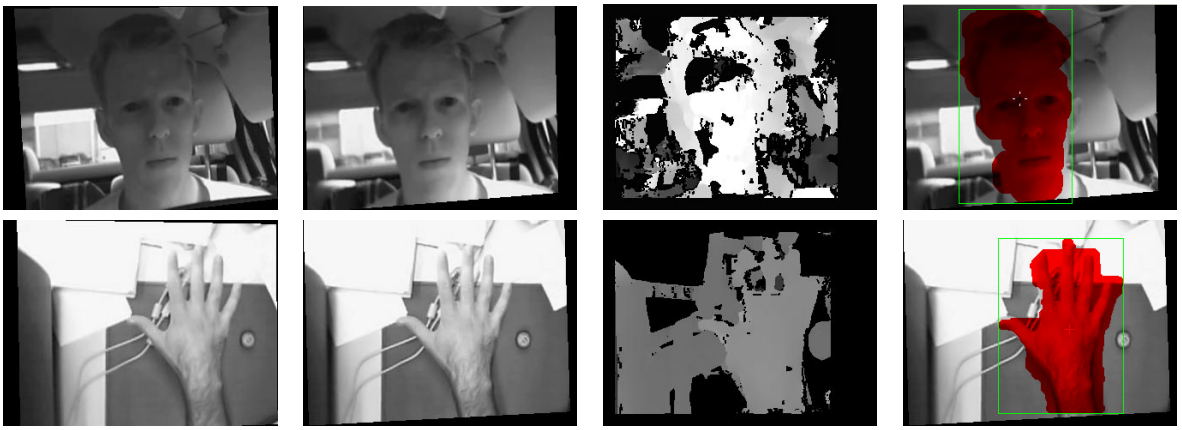
Di Stefano, Marchionni und Mattoccia präsentieren in [DSMM04] einen echtzeitfähigen Algorithmus, der im Gegensatz zu anderen Stereoalgorithmen lediglich eine *einzelne Matchingphase* (SMP ≙ single matching phase) benötigt. In einem ersten Vorverarbeitungsschritt werden die in Standardform vorliegenden Bilder normalisiert. Dafür wird von jedem Bildpunkt der Mittelwert der Grauwertintensitäten subtrahiert, der in einem kleinen, den aktuellen Punkt als Zentrum enthaltendem Fenster, berechnet wird. Dies erlaubt einen Ausgleich differierender Helligkeit in den Stereobildern, die durch unterschiedliche Kameraeigenschaften verursacht werden. Des Weiteren wird in den kleinen Fenstern die Varianz der Grauwertintensitäten errechnet und in einer LUT[10] zwischengespeichert, um in einem späteren Schritt schwachtexturierter Bereiche detektieren zu können. Als lokale Kostenfunktion für die eigentliche Matchingphase fungiert die Summe der absoluten Grauwertdifferenzen (siehe Unterkapitel 3.2.1 für Formelschreibweise), die in einem kleinen Korrelationsfenster der Größe $\rho_P \cdot \rho_P$ berechnet wird. Vorausbedingung ist die Annahme, dass sich Tiefeninformation überwiegend kontinuierlich ändert, und sich demzufolge in den Umgebungen korrespondierender Punkte ähnliche Grauwertintensitäten finden. Die Korrespondenzsuche wird auf die maximal mögliche Disparität d_P eingeschränkt. Anschließend werden, anhand der im Vorverarbeitungsschritt ermittelten Varianzen, Korrespondenzen in schwach texturierten Bereichen verworfen. Eine subpixelgenaue Interpolation der Tiefenwerte beendet die Berechnungen.

Um einen Echtzeitbetrieb zu ermöglichen wird der zeitkritische Teil der Korrespondenzfindung stark optimiert. Redundante Grauwertadditionen werden vermieden und häufig durchlaufene Instruktionen auf den MMX Befehlssatz aktueller Prozessoren abgebildet, was eine semiparallele Ausführung ermöglicht.

Abbildung 3.13 zeigt das Ergebnis von räumlichen Segmentierungen mit dem SMP-Algorithmus. Die Eingabe setzt sich aus Bildern in Standardform von der linken respektive rechten Kamera zusammen. Ausgabe des Berechnungsverfahrens ist ein in Grauwertintensitäten kodiertes Tiefenbild, wobei weiss die maximale Nähe zur Kamera repräsentiert. Schwarze Bereiche

[10] look up table.

wurden durch den Vorverarbeitungsschritt als schwachtexturierte Flächen von der Korrespondenzfindung ausgeschlossen. Eine Schwellwertsegmentierung, gefolgt von morphologischen Operationen, erzeugt das Ergebnisbild.



(a) (b) (c) (d)

Abbildung 3.13.: Linkes und rechtes Kamerabild in Standardform (a,b), In Grauwerten kodierte Tiefeninformation (c), Segmentiertes Bild (d).

Bewertung

Stereoverfahren als Segmentierer sind sehr attraktiv, da sie weitgehend invariant sind gegenüber der Farbe, Textur, Helligkeitsänderung und Form des zu segmentierenden Objekts. Darüber hinaus liefern sie einem nachfolgenden Klassifkator zusätzliche Merkmale, die den Erkennungsprozess verbessern können. Voraussetzung für eine erfolgreiche Segmentierung ist ein korrektes Tiefenbild der Szene und das Vorwissen in welcher Entfernung das Objekt von Interesse vom Kamerastandort entfernt ist. In dieser Arbeit wird vorausgesetzt, dass sowohl die Hand als auch der Kopf des Nutzers das der Kamera am Nächsten positionierte Objekt ist und der Hintergrund ab einer bestimmten Tiefenschwelle beginnt. Probleme treten auf, wenn Hintergrundobjekte diese Entfernungsgrenze unterschreiten, oder der Stereoalgorithmus fehlerhafte Tiefenergebnisse liefert.

Aktive Verfahren konnten im Rahmen dieser Arbeit aus finanziellen Gründen nicht evaluiert werden. Ohnehin sind sie wegen zu hoher Hardwarekosten für einen Einsatz in der Fahrzeugdomäne nur bedingt attraktiv.
Passive Verfahren sind zwar in der Regel preisgünstiger zu implementieren, weisen aber auch eine Reihe von Problemen auf. Da sich das zu segmentierende Objekt im Sichtfeld beider Kameras befinden muss, reduziert sich der mögliche Aktionsradius für Gesten auf den Überlappungsbereich beider Sichtkegel (erfahrungsgemäß ergibt sich eine Sichtfeldeinschränkung von 10% bei einem Kameraabstand von 5cm). Des Weiteren ist ein gesteigerter Rechenbedarf für die Rektifizierung und Korrespondenzberechnung notwendig.

Von den passiven Verfahren ermöglichte die verwendete OpenCV [Int03] Implementierung des Algorithmus von Birchfield und Tomasi [BT99] keinen Echtzeitbetrieb (Bildraten von maximal 5 Bilder pro Sekunde). Zusätzlich bereiten dem Algorithmus vor allem periodische Texturen (siehe Tastaturbereich in Abbildung 3.12(a)) und schwach texturierte Bereiche Probleme. Infolgedessen beinhalten die errechneten Tiefenbilder zahlreiche Artefakte in Form von über die Konturen hinausgehende Linien, was die abschließende Objektsegmentierung durch Schwellwertbildung erschwert (siehe Abbildung 3.14(b)).

(a) (b) (c)

Abbildung 3.14.: Vergleich Birchfield mit SMP-Stereoalgorithmus, linkes Kamerabild der Szene in Standardform (a), Ergebnis Birchfield (b), Ergebnis SMP (c).

Als besonders robust und schnell hat sich der SMP-Stereoalgorithmus [DSMM04] erwiesen. Die Korrespondenzfindung erfolgt durch Mustervergleich eines Fensterausschnitts (siehe auch Unterkapitel 3.2), was zwar eine gröbere Tiefenauflösung zur Folge hat, das Verfahren aber wesentlich weniger anfällig gegenüber periodischen Wiederholungen innerhalb von Texturen macht. Durch einen Vorverarbeitungsschritt werden schwach texturierte Bereiche frühzeitig identifiziert und von der Matchingphase ausgeschlossen. Wegen der zahlreichen Optimierungen zur Vermeidung redundanter Berechnungen erreicht der Algorithmus auf dem Testsystem Bildraten von bis zu 20 Bildern pro Sekunde. Eine feinere Tiefenauflösung ist zwar vorteilhaft, aber nicht essentiell für eine erfolgreiche Segmentierung. Somit stellt der SMP-Stereoalgorithmus innerhalb der passiven Stereoverfahren eine gute Wahl zur Segmentierung dar.

3.4. Zusammenfassung und Fazit

Abhängig vom Umfeld, sind die Ansprüche an ein Segmentierungsverfahren sehr unterschiedlich. Kann man in der Desktopumgebung von weitgehend statischen Umweltbedingungen und somit relativ konstanten Beleuchtungsverhältnissen ausgehen, so müssen in der Fahrzeugdomäne mit starken, rapiden Schwankungen und bei Nacht sogar mit einem völligem Fehlen von Beleuchtung gerechnet werden.
Um diesen gegensätzlichen Anforderungen gerecht zu werden, muss sowohl die verwendete Aufnahmetechnik als auch das Segmentierungsverfahren an die jeweilige Domäne adaptiert werden.

Wichtige Randbedingungen, die sowohl für die Desktop- als auch Fahrzeugumgebung gelten, sind ein Betrieb des Gestenerkenners in Echtzeit ohne den Einsatz von Spezialhardware und eine zufriedenstellende Segmentierungsleistung um eine robuste Erkennung sicherzustellen. Wegen zu hoher Prozessorauslastung erfüllen deshalb das Schwellwertverfahren von Otsu sowie der Stereoalgorithmus von Birchfield und Tomasi nicht die Echtzeitbedingung. Von einer näheren Betrachtung ausschließen lassen sich ebenso die betrachteten Mustererkennungsverfahren zur Handsegmentierung, da sie nur schwer in der Lage sind alle möglichen Handformen zu modellieren. Eine Verwendung ist allenfalls bei starren Handformen oder im Bereich der Kopfsegmentierung denkbar.

Aktive Stereoverfahren würden zwar eine ansprechende Segmentierung erzielen, sind aber meist mit einem zusätzlichen Hardwareaufwand gekoppelt. Da eines der Ziele dieser Arbeit ist, ein System im Hinblick auf den späteren Einsatz im Fahrzeug mit möglichst kostengünstigen Hardwarekomponenten aufzubauen, sind die betrachteten aktiven Stereosysteme nicht realisierbar.

Eines der einfachsten Schwellwertverfahren, das den Grenzwert statisch festlegt, ist sehr gut für die Desktopdomäne geeignet, sofern Annahmen über Hintergrund und Szenenbeleuchtung getroffen werden können. Im späteren System wurde der gesamte Trainingskorpus (siehe Abschnitt 6.1) für das stochastische Modell des Klassifizierers durch einen statischen Schwellwert gewonnen. Der hautfarbenbasierte Segmentierungsansatz (siehe auch [WLS02, Lin04]) erzielt, sogar bei variablem Hintergrund, gute Ergebnisse im Desktopbereich und auch in der Fahrzeugdomäne, sofern eine suffiziente Beleuchtung garantiert ist.

Fehlt im Fahrzeug Licht, ist es unabdingbar für einen visuellen Bildverarbeitungsansatz, die Szene künstlich zu erhellen um somit auch bei Dunkelheit brauchbare Bilder zu erhalten. Um den Fahrer möglichst nicht von seiner Fahraufgabe abzulenken, ist eine Bilderfassung und Ausleuchtung der Szenerie im nichtsichtbaren Infrarotbereich denkbar. Trotzdem ist die Szene Beleuchtungsschwankungen unterworfen, da auch im Sonnenlicht ein erheblicher Anteil an Infrarotlicht enthalten ist. Segmentierungsverfahren, die Verbundhistogramme zur Bildpartitionierung nutzen, sind in diesem Fall nicht mehr einsetzbar, da im Infrarotbereich nur noch ein Farbkanal zur Verfügung steht. Eine künstliche Erweiterung auf mehrere Farbkanäle im nichtsichtbaren Bereich scheitert an den hohen Kosten (siehe auch Spektralanalyse in [Lin04]).

Von den vorgestellten Verfahren kommen deshalb für eine Segmentierung im Fahrzeug nur der histogrammbasierte Schwellwert und der SMP-Stereoalgorithmus infrage. Beide Verfahren kommen gut mit Helligkeitsschwankungen zurecht, laufen in Echtzeit und benötigen als Eingabe ein, respektive zwei Graustufenbilder, die durch Infrarotkameras geliefert werden können.

KAPITEL 4

MERKMALSEXTRAKTION UND BETRACHTETE KLASSIFIKATOREN

Liegen nach der räumlichen Segmentierung die extrahierten Handregionen einer dynamischen Geste vor, werden die Attribute bzw. zeitlichen Änderungen dieser durch eine Merkmalsextraktion auf die wesentliche Information reduziert. Aufgabe des Klassifikators ist es, anhand der Merkmalswerte und deren zeitlichen Änderung eine Einordnung der Bewegung in eine vordefinierte Gestenklasse zu treffen, oder diese als nicht erkannt abzulehnen.

In diesem Kapitel wird zunächst auf verschiedene Berechnungsvorschriften für Merkmale eingegangen, die in den Klassifizierungsverfahren dieser Arbeit von Bedeutung sind. Ferner wird ein wahrscheinlichkeitsbasierter und zwei regelbasierte Ansätze zur Erkennung dynamischer Gesten vorgestellt und beurteilt.
Abschließend erfolgt ein Vergleich der betrachteten Verfahren hinsichtlich des Implementierungsaufwandes und der Erkennungsleistung, sowie der erwarteten Robustheit bei fehlerbehafteten Eingangsdaten.

4.1. Merkmalsextraktion

Die Attributierung von segmentierten Regionen ist ein wesentlicher Bestandteil der Bildverarbeitungspipeline. Dadurch wird die Datenmenge der Regionen aus dem Segmentierungsschritt auf eine Sequenz von besonders informationshaltigen Merkmalsvektoren reduziert (siehe auch das Grundlagenkapitel 2 für das generelle Vorgehen).

Entscheidend für die abschließende Klassifizierung ist eine Auswahl von Merkmalen, die in Bezug auf die Aufgabenstellung aussagekräftig sind. In den folgenden Abschnitten werden

Berechnungsvorschriften üblicher Attribute für die Charakterisierung von Formen und Bewegungen vorgestellt. Unterteilen lassen sie sich in gestalt- (Momente, Orienterung, Exzentrität) und trajektoriebeschreibende (Schwerpunktänderung, Fläche) Merkmale.

4.1.1. Momente

Um die Gestalt von Regionen prägnant zu beschreiben, bilden *Momente* die Basis für kompliziertere, zusammengesetzte Attribute. Über der kontinuierlichen Bildfunktion $f(x,y)$ sind die *allgemeinen Momente* der Ordnung $p+q$ defniert als

$$m_{p,q} = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} x^p y^q f(x,y) dxdy \tag{4.1}$$

Da in dieser Arbeit die Segmentierungsphase Regionen zurückliefert, die in der *binären* und *diskreten* Bildfunktion $b[x,y]$ kodiert sind und auf die beiden Werte 0 (Hintergrundsegment K_H) bzw. 1 (Vordergrundsegment K_V) beschränkt sind, läßt sich die Formel 4.1 wie folgt vereinfachen:

$$m_{p,q} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} x^p y^q = \sum\sum_{(x,y)\in K_V} x^p y^q \tag{4.2}$$

Fläche und Schwerpunkt

Besondere Bedeutung kommt dem Moment der nullten Ordnung $m_{0,0}$, und den Momenten der ersten Ordnung $m_{0,1}$ und $m_{1,0}$ zu. Ersteres ist die *Fläche* $A = m_{0,0}$ der Region und letztere liefern mit $x_C = \frac{m_{1,0}}{m_{0,0}}$ bzw. $y_C = \frac{m_{0,1}}{m_{0,0}}$ den *Schwerpunkt* $C = (x_C, y_C)$ der Region zurück.

Zentralmomente

Bezieht man die Berechnung der Momente auf den Schwerpunkt C, erhält man die *Zentralmomente*

$$\mu_{p,q} = \sum\sum_{(x,y)\in K_V} (x - x_C)^p (y - y_C)^q \tag{4.3}$$

Diese lassen sich ebenso direkt aus den allgemeinen Momenten berechnen:

$$\mu_{p,q} = \frac{m_{p,q}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}}\right)^p \cdot \left(\frac{m_{0,1}}{m_{0,0}}\right)^q \tag{4.4}$$

Die Zentralmomente sind im Gegensatz zu den allgemeinen Momenten invariant hinsichtlich Translationen der Region. Um zugleich noch eine Größeninvarianz zu erzielen, werden die Zentralmomente mit einem Skalierungsfaktor versehen. Meist ist das die Fläche A. Teilt man die Zentralmomente durch Potenzen von A ergeben sich somit die *normierten Zentralmomente*

$$\upsilon_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\frac{p+q}{2}+1}} \tag{4.5}$$

die translations- und skalierungsinvariant sind. Die normierten Zentralmomente nullter und erster Ordnung sind trivial und für eine Objektattributierung nicht verwendbar: $\upsilon_{0,0} = 1, \upsilon_{1,0} = \upsilon_{0,1} = 0$.

Hu-Moment-Invarianten

Invarianz hinsichtlich der Orientierung einer Region lässt sich über eine aufwendige Transformationsvorschrift der normierten Zentralmomente [Tea80] erzielen. Die sogenannten *Hu-Moment-Invarianten* (HMI) besitzen jedoch allein durch ihre Bildungsvorschrift eine immanente Rotationsinvarianz. Sie werden durch Verknüpfung der diskreten Zentralmomente erzeugt [Hu62]:

$$\begin{aligned} H_1 &= \mu_{0,2} + \mu_{2,0} \\ H_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \\ H_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \\ H_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2 \end{aligned}$$

Durch ein systematisches Bildungsgesetz lassen sich Hu-Momente ab der 5. Ordnung mit beliebig hoher Ordnungszahl generieren (siehe [Hu62] für Berechnungsvorschrift). Der wesentliche Informationsgehalt über die Form der Region konzentriert sich auf die ersten sieben HMI, wobei alle sieben HMI translations-, rotations- und skaleninvariant sind. Zusätzlich sind die HMI bis zur Ordnung sechs reflektionsinvariant.

4.1.2. Orientierung und Exzentrität

Sowohl die Orientierung als auch die Exzentrität einer Region lassen sich aus den diskreten Zentralmomenten zweiter Ordnung berechnen. Die Orientierung einer Region ist dabei die Achse, die eine Rotation des Objekts mit der geringsten Massenträgheit zulässt. Sie ist als Winkel ρ zwischen Rotationsachse und x-Achse definiert und berechnet sich wie folgt:

$$\rho = \frac{1}{2} \arctan \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \tag{4.6}$$

Die Exzentrität ε vermittelt den Grad der Rundheit einer Region. Sie hat einen Wertebereich zwischen 0 und 1, wobei ein Wert von 0 eine Kreisform und ein Wert von 1 eine linienartige Gestalt der Region widerspiegelt.

$$\varepsilon = \frac{(\mu_{2,0} - \mu_{0,2})^2 - 4\mu_{1,1}^2}{(\mu_{2,0} + \mu_{0,2})^2} \tag{4.7}$$

4.1.3. Trajektorie

Die Trajektorie eines bewegten Objekts definiert den Kurvenverlauf, den es auf seinem Weg vollzieht. Im Tupel (x_C, y_C, z_C) wird dabei die genaue Position festgehalten, wobei für die ersten beiden Komponenten (x_C, y_C) meist der Schwerpunkt des Objekts verwendet wird (siehe auch Abschnitt 4.1.1). Die dritte Komponente z_C läßt sich bei einer bildbasierten Segmentierung entweder indirekt über die Flächenänderung ΔA des Objekts, oder über ein Stereoverfahren ermitteln (siehe auch Abschnitte 3.3.2 und 5.2.2).

Um die Trajektorieninformation translationsinvariant zu gestalten, wird sie auf Relativwerte abgebildet. Eine Merkmalsvektorsequenz $(\vec{m}_1, \ldots, \vec{m}_T)$ der Länge T als Repräsentant für eine Trajektorie im $\mathbb{R}^3$ ist dann wie folgt aufgebaut:

$$\vec{m}_t = \begin{pmatrix} x_C(t) - x_C(t-1) \\ y_C(t) - y_C(t-1) \\ z_C(t) - z_C(t-1) \end{pmatrix} = \begin{pmatrix} \Delta x_C(t) \\ \Delta y_C(t) \\ \Delta z_C(t) \end{pmatrix} \quad \text{mit } 1 \leq t \leq T \tag{4.8}$$

4.2. Regelbasierte Klassifizierung

4.2.1. Zweiphasen Modell

James P. Mammen beschreibt in [MCA02] ein regelbasiertes, *trajektorienorientiertes* Vorgehen zur Klassifizierung dynamischer Handgesten, das den Erkennungsprozess in zwei Phasen untergliedert. In einem ersten Schritt werden Gesten, entsprechend ihrer *dominanten Trajektorie*, einem reduzierten Gestenkatalog zugeordnet. Eine anschließende Funktionsanalyse ermittelt im zweiten Schritt die finale Ausgabe innerhalb des gewählten Katalogs.

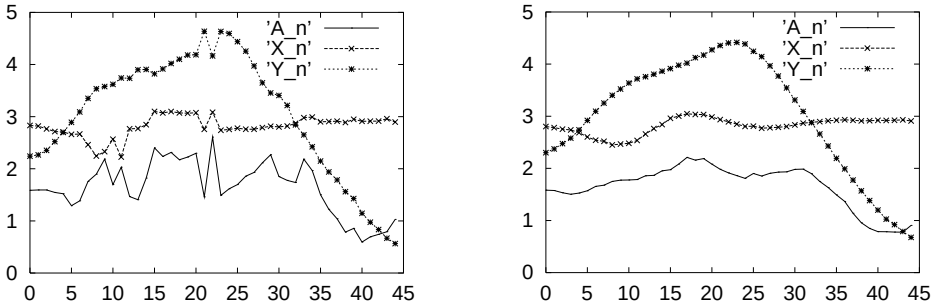
Phase 1: Sondierung dominanter Trajektorien

Die Koordinaten des Schwerpunktes $C = (x_C, y_C)$ und die Fläche A der Handregion werden durch den vorausgehenden Segmentierungs- bzw. Trackingprozess bestimmt. Um eine gewisse Invarianz gegenüber Kameraposition und Nutzer zu erreichen, wird eine Normierung der Fläche auf den Startwert $m_{0,0}(0)$ und der Schwerpunktkoordinaten auf die Breite der Handregion vorgenommen. Ergebnis ist der Merkmalsvektor

$$\vec{m}_t = \begin{pmatrix} X(t) \\ Y(t) \\ A(t) \end{pmatrix} = \begin{pmatrix} m_{1,0}(t)/m_{0,1}(0) \\ m_{0,1}(t)/m_{0,1}(0) \\ m_{0,0}(t)/m_{0,0}(0) \end{pmatrix} \quad \text{mit } 0 \leq t \leq T \tag{4.9}$$

wobei $m_{p,q}(t)$ das Moment der Ordnung $p+q$ der Handregion im Bild zum Zeitpunkt t der Bildsequenz darstellt (siehe auch Abschnitt 4.1.1 für Details über Momente).

Die in der Regel verrauschten Eingangsdaten werden durch einen Mittelwertsfilter geglättet (vgl. auch Abbildung 4.1(a) mit Abbildung 4.1(b)) und anschließend auf ihren Informationsgehalt hin untersucht. Trajektorien mit hohem Informationsgehalt zeichnen sich durch eine große Funktionsamplitude aus und werden als *dominante Trajektorien* bezeichnet. Anhand derer wird m_t drei Mengen S_X, S_Y und S_A zugewiesen, wobei die Indizes die jeweils dominante Trajektorie bezeichnen. James P. Mammen unterscheidet mit seinem Ansatz fünf reduzierte Gestenkataloge, die sich aus Differenz- und Schnittbildung dieser Mengen ergeben (siehe auch *Venndiagramm* in Abbildung 4.2). Die (LEFT) Geste in Abbildung 4.1(b) beispielsweise ist Element des reduzierten Katalogs $S_Y \setminus (S_X \cup S_A)$, da der Y-Verlauf die einzige dominante Trajektorie bildet. Bei den Kreisgesten „Clockwise“ (CW) und „Counterclockwise“ (CCW) ist sowohl die X als auch die Y Trajektorie dominant, was sie zu Elementen des Katalogs $(S_X \cap S_Y) \setminus S_A$ macht.



(a) (b)

Abbildung 4.1.: Dominante Trajektorie $Y(t)$ für die nach links (LEFT) Geste [MCA02]. Ungefilterte Trajektorien (a), mittelwertgefilterte Trajektorien (b)

In mathematischer Schreibweise wird nach der Funktionsglättung durch den Mittelwertfilter die maximale Amplitude

$$\delta_I = \max(I(t)) - \min(I(t)) \quad \text{mit } I \in \{X, Y, A\} \tag{4.10}$$

aller Trajektorien ermittelt und anhand dessen die Einteilung in die fünf Gestenkataloge vorgenommen. Dazu wird eine Entscheidungssequenz durchlaufen, die im Folgenden exemplarisch für zwei Kataloge angegeben ist. f_I steht dabei für $\arg_I \max \delta_I$ und θ_1, θ_2 bezeichnen zwei Mindestschranken.

Katalog 1 (UP, DOWN): Dieser Katalog beinhaltet die nach oben und nach unten Gesten und wird gewählt, falls die Amplitude der X-Trajektorie groß und die der Y- sowie A-Trajektorie klein ist:

$$(f_I = X) \wedge (\delta_X \geq \theta_1) \wedge (\delta_Y < \frac{\delta_X}{2}) \wedge (\delta_A < \frac{\delta_X}{2})$$

Katalog 3 (CW, CCW): Dieser Katalog beinhaltet zwei Kreisgesten (im und gegen den Uhrzeigersinn) und wird zurückgeliefert, falls die Amplituden von $X(t)$ und $Y(t)$ hinrei-

chend groß sind und die Amplitude von $A(t)$ maximal halb so groß ist:

$$\left[(f_I = X) \wedge (\delta_X \geq \theta_2) \wedge (\delta_Y > \frac{\delta_X}{2}) \wedge (\delta_A < \frac{\delta_X}{2})\right] \quad \vee$$
$$\left[(f_i = Y) \wedge (\delta_Y \geq \theta_2) \wedge (\delta_X > \frac{\delta_Y}{2}) \wedge (\delta_A < \frac{\delta_Y}{2})\right]$$

Fällt eine Geste in keinen der fünf Kataloge oder erfüllt innerhalb dieser die Funktionsanalyse nicht, wird sie als nicht erkannt klassifiziert und einer Garbage-Klasse zugewiesen.

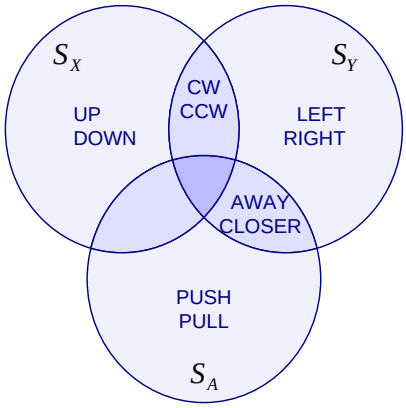


Abbildung 4.2.: Venndiagram der Gestenkategorien nach dominanter Trajektorie [MCA02].

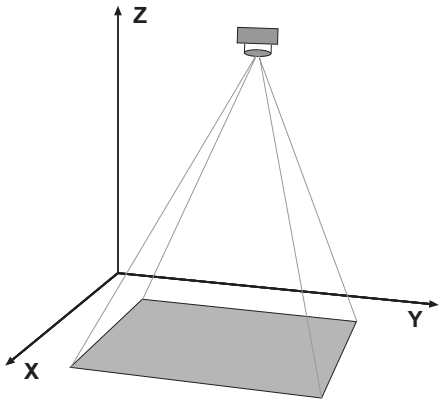


Abbildung 4.3.: Achsenbezug der Gesten in diesem Abschnitt.

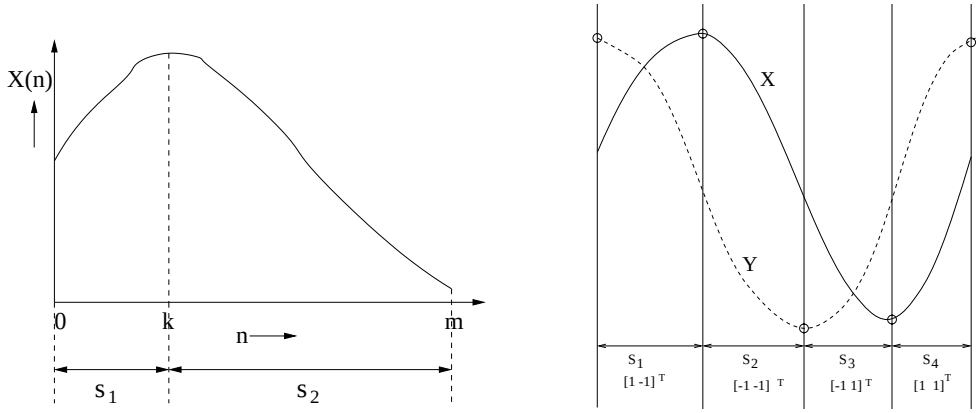
Phase 2: Funktionsanalyse

Ist die Aufteilung in reduzierte Gestenkataloge durch Phase 1 beendet, wird mittels einer genauen Analyse der dominanten Trajektorien die Erkennung abgeschlossen. Dazu werden die Funktionen an ihren Extremstellen in einzelne Abschnitte partitioniert und der Kurvenverlauf innerhalb dieser Bereiche untersucht. Die genaue Art der Analyse differiert dabei zwischen den Katalogen und wird im Folgenden exemplarisch für die Gesten UP, DOWN, CW und CCW erläutert.

Katalog 1 (UP, DOWN): X ist die einzige dominante Trajektorie in diesem Katalog und wird auf ihren Verlauf hin genauer untersucht. Die UP Geste kennzeichnet eine kleine Ausholbewegung der Hand und im Anschluss daran eine kontinuierliche Bewegung gegen die Richtung der X-Achse (siehe auch Achsenbezüge in Abbildung 4.3). Bis zum Umkehrpunkt k, am Ende der Ausholbewegung, steigt somit die korrespondierende und dominante Trajektorie stetig an, um danach bis zum zeitlichen Ende m der Bewegung kontinuierlich abzufallen. Im Idealfall zerfällt die Geste somit in die beiden Zustände

$s_1 = X(k) - X(0)$ und $s_2 = X(m) - X(k)$. Gilt $s_1 + s_2 < 0$ handelt es sich um eine UP ansonsten um eine DOWN Geste (siehe auch Abbildung 4.4(a)).

Katalog 3 (CW, CCW): Innerhalb diesem Katalog sind sowohl X als auch Y dominante Trajektorien und werden auf ihre Verläufe hin untersucht. Die Hand vollführt einen Kreis entweder im, oder gegen den Uhrzeigersinn, was in zwei sinusartigen, gegeneinander verschobenen Funktionsverläufen resultiert (siehe auch Abbildung 4.4(b)). Nach dem Aufsplitten an den Extremwertstellen beider Funktionen werden zwischen den Maxima und Minima verschiedene Zustände anhand des Vorzeichens der beiden Steigungen gebildet. Mit 1 wird eine zunehmende und mit -1 eine abnehmende Steigung gekennzeichnet. Die resultierende Zustandsmenge $Z = \{s_1, s_2, s_3, s_4\}$, mit $s_1 = [1\ \text{-}1]^T$, $s_2 = [\text{-}1\ \text{-}1]^T$, $s_3 = [\text{-}1\ 1]^T$ und $s_4 = [1\ 1]^T$ bezeichnet alle Steigungskombinationen beider Trajektorien. Diese Vektoren werden über die Funktion $g: Z \rightarrow \{0, 1, 2, 3\}$ auf Skalare abgebildet: $g\left([1\ \text{-}1]^T\right) = 0$, $g\left([\text{-}1\ \text{-}1]^T\right) = 1$, $g\left([\text{-}1\ 1]^T\right) = 2$, $g\left([1\ 1]^T\right) = 3$. Für eine Kreisgeste im Uhrzeigersinn (CW) bildet die resultierende Zustandsfolge $g(\hat{s}_1), \ldots, g(\hat{s}_n)$, mit $\hat{s}_i \in Z$ eine Teilsequenz der periodischen Folge $0, 1, 2, 3, 0, 1, 2, \ldots$ Kreise gegen den Uhrzeigersinn (CCW) durchlaufen die periodische Folge rückwärts.



(a) Skizze zur UP-Geste. (b) Skizze zur CW-Geste.

Abbildung 4.4.: Skizzen zum Trajektorienverlauf der Gesten UP und CW [MCA02].

Bewertung

Die Attraktivität des vorgestellten Verfahrens liegt in dessen Einfachheit, was eine rasche Implementierung ermöglicht. Trotz der überschaubaren Komplexität ist das hierarchische Vorgehen robust und erreicht nach den Ergebnissen von [MCA02], Erkennungsraten von über 80% bei 10 unterschiedlichen Gesten, die von fünf Probanden durchgeführt wurden. Die Ausführungsdauer ist auf gängigen Rechensystemen zu vernachlässigen und ein weiteres Plus ist der vollständige Verzicht auf ein möglicherweise sehr aufwendiges Training, da das gesamte Gestenvokabular schon fest im Quellcode verankert ist.

Diese feste „Verdrahtung“ von Gestendefinitionen schränkt die Flexibilität und Skalierbarkeit des Verfahrens ein. Werden neue Gesten in den Gesamtkatalog aufgenommen, müssen erst passende Regeln geschaffen werden und in die zweistufige Hierarchie integriert werden. Darüber hinaus sind Gesten denkbar, die sich nur schwer in das vorgegebene Schema pressen lassen, oder bei einer Integration starke Kongruenz mit schon vorhandenen Gesten aufweisen, wenn lediglich die Trajektorien betrachtet werden. Verbessern ließe sich diese Problematik, wenn in Ergänzung zu den Bewegungskurven, als zusätzliches Attribut die Gestalt der Hand Betrachtung fände.

4.2.2. Erkennung mit Deterministischen Endlichen Automaten (DEA)

P. Hong, M. Turk und T. Huang [HHT00a, HHT00b] schlagen ein *zustandsbasiertes Vorgehen* zum Klassifizieren von Gesten, mittels *Deterministischen Endlichen Automaten* (DEA) vor. Für jede Geste wird in einer Lernphase, basierend auf Trainingsmaterial, ein korrespondierender DEA semiautomatisch erzeugt. Während der Erkennungsphase laufen alle DEA's zeitgleich mit dem Merkmalsstrom. Eine Geste wird als erkannt angenommen, wenn ein DEA in einen Endzustand gelangt. Wie schon im vorherigen Verfahren werden allein Trajektorieninformationen im $\mathbb{R}^2$ als Merkmale verwendet. Ein Merkmalsvektor $\vec{m} = (x\ y)^T$ setzt sich somit aus einer x und einer y Komponente zusammen, die den Schwerpunkt der Handregion bezeichnen.

Gestenmodellierung

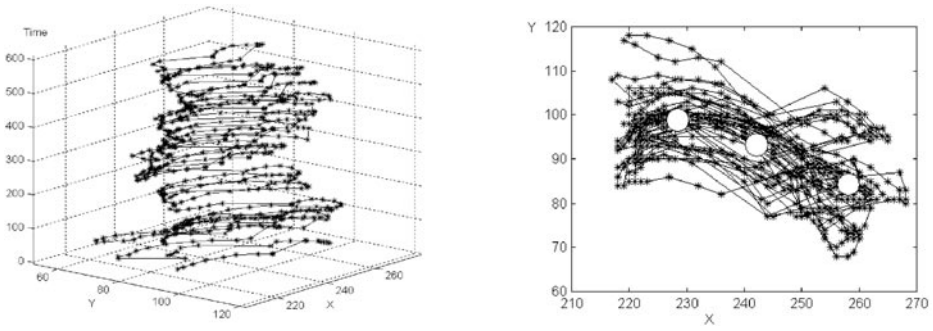
Eine Geste wird als Zustandsfolge innerhalb eines Automaten repräsentiert. Ein Zustand z wird zur räumlichen und zeitlichen Einordnung von Merkmalsvektoren durch das Tupel

$$T_z = < \vec{\mu}_z, C_z, d_z, t_z^{min}, t_z^{max} >$$

charakterisiert, wobei $\vec{\mu}_z$ das räumliche Zentrum und C_z die Kovarianzmatrix der von z modellierten Verteilung angibt. Der maximale Entfernungsschwellwert vom Zentrum wird durch d_z gekennzeichnet. Das Intervall $[t_z^{min}, t_z^{max}]$ stellt die minimale bzw. maximale Verweildauer im Zustand z dar. Über diese Tupel und die Nachbarzustände wird die mögliche Richtung und Geschwindigkeit der Bewegung innerhalb gewisser Varianzen determiniert. Ziel ist es, die Topologie des DEA und die Tupelwerte T_z der einzelnen Zustände für das gesamte Gestenvokabular möglichst automatisiert zu ermitteln.

Es werden für jede Geste Trainingsdaten ermittelt, die zur Modellbestimmung lückenlos aneinandergereiht werden. Wegen der unterschiedlichen Ausführungszeiten der Gesten erweist es sich als schwierig, die zeitlichen und räumlichen Parameter einer Bewegung in einem Schritt zu errechnen. Deshalb wird in einer ersten Analysephase (räumliche *Clusterbildung*[1])

[1] Bildung von Häufungspunkte



(a) (b)

Abbildung 4.5.: Trainingsdaten einer „Wave Hand"-Geste [HHT00b]. Trajektoriedaten in räumlicher und zeitlicher Ansicht (a), Daten in räumlicher Ansicht ohne zeitliche Komponente (b)

die räumliche von der zeitlichen Komponente entkoppelt und lediglich die Datenverteilung betrachtet (siehe auch Abbildung 4.5). Jeder Zustand z korrespondiert dann zu einer Region bestehend aus Datenpunkten $\vec{x} = (x,y)$ im $\mathbb{R}^2$, die durch den Erwartungswert $\vec{\mu}_z = \mathbb{E}(\vec{x})$ und die Kovarianzmatrix $C_z = \mathbb{E}((\vec{x}-\vec{\mu})^T(\vec{x}-\vec{\mu}))$ repräsentiert wird. Als Abstandsmaß eines Datenpunktes $\vec{x}$ zu den Datenpunkten modelliert von z wird die *Mahalanobisdistanz*

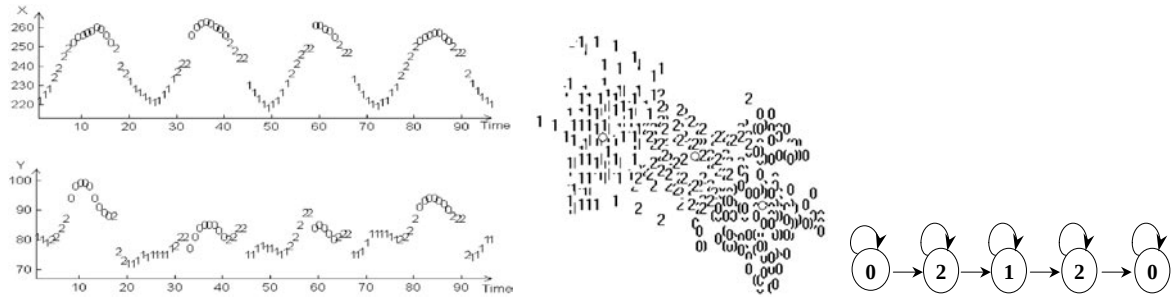
$$D(\vec{x},z) = \sqrt{(\vec{x}-\vec{\mu}_z)^T C_z^{-1}(\vec{x}-\vec{\mu}_z)} \tag{4.11}$$

gewählt. Die Zentren μ_z der Cluster werden durch einen modifizierten k-means-Algorithmus ermittelt. In Abbildung 4.5(b) sind die Zentren durch weisse Kreise markiert.

Anschließend wird jeder Datenpunkt mit einer Markierung versehen anhand derer die Zustandszugehörigkeit sichtbar wird. Zeitlich adjazente Datenpunkte werden gruppiert, sofern sie eine identische Markierung aufweisen und folglich zum gleichen Zustand gehören. Ein Beispiel für den Verlauf markierter Datenpunkte ist die Samplefolge [1 1 1 2 2 2 2 0 0 0 0 2 2 2 1 1]. Sie ist aus der Zustandsfolge [1 2 0 2 1] aufgebaut. Die Sequenz der Zustandsgruppen erlaubt eine manuelle Bestimmung der Topologie des endlichen Automaten. Für einen Zyklus der „Wave Hand"-Geste zum Beispiel ist in Abbildung 4.6(a) die Zustandsfolge $[0,2,1,2,0]$ erkennbar, die zur Bildung des Automaten in Abbildung 4.6(c) führt. Um die Verweildauer $[t_z^{min}, t_z^{max}]$ zu berechnen, wird für alle Zustände die minimale (t_z^{min}) und maximale (t_z^{max}) Anzahl von Datenpunkten je Zustand ermittelt. Da zwischen den Gesten eine beliebig lange Ruhephase erlaubt ist, wird $t_0^{max} = \infty$ gesetzt. Zuletzt wird für die Punktemenge, die zu einem Zustand z gehört, der Mittelwert $\overline{m}$ und die Varianz σ^2 der Entfernungen zum Zentrum $\vec{\mu}_z$ berechnet. Der Schwellwert d_z ergibt sich aus der Summe $\overline{m} + \sigma$.

Erkennung

Gesten werden in Echtzeit und mit dem Merkmalsstrom schritthaltend klassifiziert. Dazu werden die vorliegenden Automaten mit dem aktuellen Merkmalsvektor $\vec{m}$ beaufschlagt. Eine



(a) (b) (c)

Abbildung 4.6.: Automatenkonstruktion für die „Wave Hand"-Geste [HHT00b]. Markierte Daten in räumlicher und zeitlicher Ansicht (a), Räumliche Ansicht (b), Resultierender Automat (c)

Geste gilt als erkannt, wenn der korrespondierende Automat seine gesamte Zustandssequenz durchlaufen hat und den Endzustand erreicht. Der momentane Status eines Automaten wird im Tupel $s = [z_k, \Delta t]$ festgehalten, wobei z_k den aktuellen Zustand und Δt die Verweildauer in diesem kennzeichnet. Ein Zustandsübergang erfolgt ausschließlich, wenn der Schwellwert d_z für den nächsten Zustand unterschritten und die Verweildauer im aktuellen Zustand eingehalten wird. Es muß somit eine der folgenden Regeln erfüllt sein:

1. $(D(\vec{m}, z_{k+1}) \leq d_{k+1}) \wedge (\Delta t > t_k^{max})$
2. $(D(\vec{m}, z_{k+1}) \leq d_{k+1}) \wedge (D(\vec{m}, z_{k+1}) \leq D(\vec{m}, z_k)) \wedge (\Delta t \geq t_k^{min})$
3. $(D(\vec{m}, z_{k+1}) \leq d_{k+1}) \wedge (D(\vec{m}, z_k) \geq d_k)$

Kann durch $\vec{x}$ kein Zustandsübergang erfolgen und wird ebenso der Schwellwert d_z bzw. die maximale Verweildauer des aktuellen Zustands z_k verletzt, wird dieser Automat von der laufenden Erkennung ausgeschlossen und auf den Startzustand zurückgesetzt. Erreichen mehr als ein Automat zeitgleich den Endzustand, liegt eine Mehrdeutigkeit vor. Um diese aufzulösen, wird diejenige Geste zurückgeliefert, deren Trajektorie im Mittel den geringsten Abstand von den Zustandszentren μ_z aufweist.

$$Geste = \arg_g \min \left(\frac{1}{n_g} \sum_{i=1}^{n_g} D(\vec{m}_i, z_{gi}) \right) \tag{4.12}$$

z_{gi} bezeichnet dabei den Zustand innerhalb des Automaten g, zu dessen Verteilung der Punkt $\vec{m}_i$ gehört und n_g beziffert die Anzahl der von g zum aktuellen Zeitpunkt abgearbeiteten Datenwerte.

Bewertung

Nach [HHT00b] unterscheidet das vorgestellte Verfahren lediglich drei verschiedene Handgesten (Kreisbewegung, Winken und Bewegung in Form einer Acht) und erreicht dabei Er-

kennungsraten von über 90%. Diese Ergebnisse basieren auf einer hautfarbenbasierten Segmentierung in der Laborumgebung. Wie sich die Erkennungsquote des Verfahrens bei einem größeren Gestenschatz unter schlechteren Segmentierungsbedingungen verhält, ist allein aus [HHT00a, HHT00b] nicht ersichtlich. Vermutlich verhindert jedoch das strikte Durchlaufen aller Zustände der Automaten eine flexible Reaktion auf Rauschen in der Bildvorverarbeitung. Das Training der Automaten kann zwar mit einer kleinen Menge von Gesten erfolgen, erfordert jedoch einen manuellen Nachbearbeitungsschritt, was den Aufwand der Modellgenerierung erhöht.

4.3. Stochastische Klassifizierung mittels Hidden Markov Modellen

Hidden Markov Modelle (HMM) haben ihren Ursprung in der Spracherkennung. Sie sind aber nicht auf diesen Bereich beschränkt, sondern finden in letzter Zeit auch verstärkt in anderen Domänen Anwendung. Ein HMM ist ähnlich einem Automaten aus Zuständen und Übergängen zwischen diesen aufgebaut, wobei die Zustandsübergänge mit einer bestimmten Wahrscheinlichkeit erfolgen. Zusätzlich sind Zustände in der Lage mit einer gewissen *Emissionswahrscheinlichkeit* Symbole zu emittieren, die von einem externen Beobachter als Symbolsequenz wahrgenommen werden. Die vom Modell durchlaufene Zustandsfolge bleibt dem Beobachter dabei verborgen, woher auch der Name „Hidden“ Markov Modell rührt. Um HMM’s als Klassifikatoren verwenden zu können, muss für jede Klasse ein eigenes Modell trainiert werden. Training bedeutet dabei die Schätzung aller freien Parameter wie z. B. Übergangs- und Einsprungswahrscheinlichkeiten. Eine zu klassifizierende Merkmalsfolge wird dann derjenigen Gestenklasse zugeordnet, dessen HMM die höchste Erzeugungswahrscheinlichkeitsdichte liefert. Eine generelle Einführung für Hidden Markov Modelle findet sich unter [Rab89].

In neuerer Literatur finden sich vermehrt Systeme, die mittels HMM sowohl Hand- als auch Kopfgesten klassifizieren. Starner und Pentland [Sta94, SP95] unterscheiden 40 verschiedene Handzeichen der amerikanischen Zeichensprache (ASL) über diskrete HMM mit einer Erkennungrate von über 90%. Kapoor und Picard [KP01] verwenden HMM zur Klassifizierung von drei verschiedenen Kopfgesten über Infrarotbilder. Sie erzielen Erkennungsraten von 78%. Wilson und Bobick [WB99] erkennen Zeigegesten über parametrische HMM.

Ein System mit modifzierten HMM zur Erkennung dynamischer Handgesten wird von Morguet [Mor00] vorgestellt. Als Merkmale verwendet er relative Trajektorieninformation zur Ableitung des Bewegungsverlaufs und Hu-Moment-Invarianten (siehe auch Abschnitt 4.1.1) für die Kodierung der Handform. Sein Vorgehen wird im Folgenden genauer untersucht.

Semikontinuierliche HMM

Im Gegensatz zu *diskreten HMM* erlauben *kontinuierliche HMM* eine unendliche Anzahl möglicher Emissionswerte. Das begünstigt zwar eine genauere Modellierung des zeitlichen Prozesses, verlangt aber wegen der großen Zahl freier Parameter einen sehr großen Trainingskorpus [Rab89]. Als Kompromiss schlägt Morguet deshalb die Verwendung von *semikontinuierlichen HMM* (sHMM) vor, die das Problem etwas entschärfen. Auf Kosten der Modellierungsgenauigkeit ist bei sHMM die Anzahl der freien Parameter reduziert, wobei sie immer noch in der Lage sind kontinuierliche Merkmale zu modellieren.

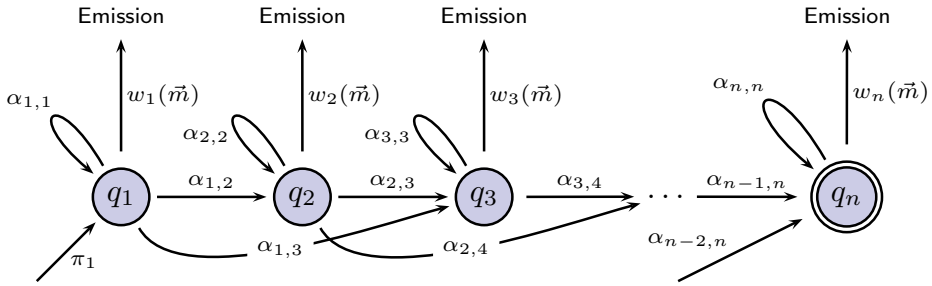


Abbildung 4.7.: Struktur und Parameter eines semikontinuierlichen Hidden Markov Modells

Die Topologie und Parameter eines sHMM λ_g sind in Abbildung 4.7 zu sehen. $Q = \{q_1, \ldots, q_n\}$ bezeichnet dabei die Zustandsmenge des Modells mit der Kardinalität n. Die Übergangswahrscheinlichkeiten von Zustand q_i nach Zustand q_j sind mit $\alpha_{i,j}$ und die Einsprungswahrscheinlichkeit in das Modell mit π_i gekennzeichnet. Jeder dieser Zustände enthält eine *Wahrscheinlichkeitsdichtefunktion* (WDF) $w_i(\vec{m})$, die angibt mit welcher Wahrscheinlichkeit ein D-dimensionaler Merkmalsvektor $\vec{m}$ im Zustand q_i emittiert wird. $w_i(\vec{m})$ setzt sich dabei wiederum aus WDFs in Form von D-dimensionalen Gaußverteilungen zusammen, die durch Mixturkoeffizienten $c_{k,i}$ gewichtet werden.

$$\sum_{k=1}^{L} c_{k,i} \cdot \frac{1}{\sqrt[D]{2\pi}\sqrt{|\Sigma_k|}} \exp\left[-\frac{1}{2}(\vec{m}-\vec{\mu}_k)^T \Sigma^{-1}(\vec{m}-\vec{\mu}_k)\right] \tag{4.13}$$

Mittels eines modifizierten *k-means*-Algorithmus wird die Partitionierung aller Trainingsdaten in L Cluster vorgenommen. Der Mittelwertsvektor $\vec{\mu}_k$ und die Kovarianzmatrix Σ_k beschreiben dann die Verteilung des Clusters $1 \leq k \leq L$.

Die Struktur des HMM liegt implizit durch Nullsetzen diverser Parameter fest. In Abbildung 4.7 handelt es sich um ein *Links-Rechts-Modell*, das durch $\pi_l = 0, (l > 1)$ und $\alpha_{i,j} = 0, (j < i) \vee (j > i+2)$ charakterisiert ist. Das heißt der Einsprung in das Modell erfolgt nur in den ersten Zustand und Übergänge müssen in den selben Zustand oder in einen Zustand weiter rechts im Modell führen, wobei maximal ein Zustand übersprungen werden darf. Links-Rechts-Modelle eignen sich im Gegensatz zu *ergodischen Modellen*[2] besonders gut zur Nachbildung zeitlicher Prozesse [Rab89].

[2] Jeder Zustand des Modells kann von einem anderen Zustand aus in endlich vielen Schritten erreicht werden.

Sowohl für das Training der HMM als auch die Erkennung wird alternativ zum *Baum-Welch-Algorithmus* [Kim02] der *Viterbi-Algorithmus* [LA98] verwendet. Als Gründe nennt Morguet die schnellere Ausführungszeit und die einfachere Überführung der Berechnungsvorschriften in eine logarithmische Darstellung. Diese Überführung ist notwendig um die große numerische Dynamik der WDF beherrschen zu können.

Training

Da das Training lediglich ein lokales Optimum ermitteln kann, ist es von großer Wichtigkeit, alle zu schätzenden Parameter mit geeigneten Initialwerten vorzubelegen.

- Die Einsprungswahrscheinlichkeiten werden fest auf $\vec{\pi} = (1,0,0,\ldots,0)$ initialisiert und müssen nicht nachtrainiert werden. Einsprünge in die Modelle dürfen damit nur in den ersten Zustand erfolgen.
- Die Selbstübergangswahrscheinlichkeiten $\alpha_{i,i}$ werden auf $1 - n \cdot \overline{T}^{-1}$ gesetzt, wobei n die Anzahl der Zustände im Modell und $\overline{T}$ die mittlere Länge der Merkmalssequenzen bezeichnet. Damit wird die mittlere Verweildauer auf alle Zustände gleichverteilt.
- Die Restwahrscheinlichkeit $1 - \alpha_{i,i}$ wird nach Morguet auf die beiden Übergangswahrscheinlichkeiten $\alpha_{i,i+1}$ und $\alpha_{i,i+2}$ im Verhältnis $\frac{1}{599}$ aufgeteilt. Das Überspringen eines Zustands wird dadurch erschwert, aber nicht unmöglich.

Nach der Clusterung des gesamten Trainingsmaterials über den k-means-Algorithmus, werden ausgehend von den gesetzten Startwerten die Parameter iterativ nachgeschätzt. Das Vorgehen zur Schätzung der Parameter führt hier zu weit, weshalb an dieser Stelle dem interessierten Leser [Mor00] als weiterführende Literatur empfohlen wird.

Erkennung

Um eine Geste zu klassifizieren, wird für die Merkmalssequenz der zu erkennenden Bewegung, über den Viterbi-Algorithmus der optimale Pfad durch alle Modelle berechnet. Dieser Pfad approximiert die Erzeugungswahrscheinlichkeit für die beobachtete Merkmalsfolge. Als Ergebnis wird die Geste zurückgeliefert, dessen Modell die höchste Erzeugungswahrscheinlichkeit liefert. Liegen alle Wahrscheinlichkeiten unter einer minimalen Schranke, oder ist der numerische Abstand zwischen den führenden zwei Erzeugungswahrscheinlichkeiten zu gering, ist das Ergebnis die Garbage Geste.

Bewertung

Morguet [Mor00] unterscheidet in seiner Arbeit 40 verschiedene Gesten und erreicht dabei personenabhängige Erkennungsraten von über 95%. Die personenunabhängigen Leistungen sind mit ca. 60% relativ schlecht, was Morguet auf den limitierten Trainingsdatensatz

zurückführt. Die Klassifizierung in seiner Arbeit basiert auf einer guten räumlichen Segmentierung. Im Desktopbereich verspricht das Verfahren somit ansprechende Ergebnisse, sofern ein ausreichend großer Trainingskorpus erstellt wird. Wie sich das System auf fehlerbehafteten Eingabedaten verhält, welche bei der räumlichen Segmentierung im Fahrzeugbereich vermutlich verstärkt auftreten, wird nicht untersucht.

4.4. Zusammenfassung und Fazit

Die Erkennungsrate der drei vorgestellten Verfahren bewegt sich in ähnlichen Bereichen. Einzig der wahrscheinlichkeitsbasierte Ansatz büßt bei der personenunabhängigen Erkennung an Leistung ein, was vermutlich an dem zu kleinen Trainingskorpus liegt. Fraglich ist, ob das Vergleichskriterium Erkennungsquote Aussagekraft hat, da den errechneten Leistungen unterschiedliche Gestentypen, Gestenmengen und Testdaten zugrunde liegen. Besser ist vermutlich in diesem Fall der Vergleich nach anderen Gesichtspunkten, wie Implementierungs- und Trainingsaufwand, Robustheit und Skalierbarkeit.

Durch die Entkopplung von räumlicher und zeitlicher Information der Trajektoriedaten können die Endlichen Automaten mit einer geringen Menge von Trainingsdaten modelliert werden, was sie primär gegenüber dem wahrscheinlichkeitsbasierten Ansatz mit sHMM attraktiv erscheinen lässt.[3] Der Zweiphasen Algorithmus kommt im Gegensatz zu den anderen beiden Verfahren gänzlich ohne Trainingsmaterial aus, erfordert aber das Implementieren von Regeln für den gesamten Gestenschatz. Resultat ist eine schlechtere Skalierbarkeit und Flexibilität, vor allem wenn das Gestenvokabular entsprechend umfangreich wird. Für die 17 verschiedenen Gesten innerhalb dieser Arbeit ist eine Realisierung des Verfahrens jedoch problemlos.

Vorteil des Zweiphasen Algorithmus ist die hierarchische Struktur, welche das Verfahren robuster gestaltet. Kleinere Ausreißer in den Merkmalsdaten können durch die Mittelwertsfilterung korrigiert werden. Größere Fehler, wie Aussetzer der räumlichen Segmentierung, erlauben zumindest eine grobe Einordnung der Gesten in reduzierte Kataloge anhand ihrer dominanten Trajektorien. Somit kann sich das Verfahren dynamisch der Segmentierungsqualität anpassen und dementsprechend die Granularität der Erkennungsgenauigkeit variieren. Das Klassifikatorverhalten des HMM-basierten Verfahrens hingegen weist eine relativ geringe Transparenz auf. Sind die Eingabedaten fehlerbehaftet, erzeugen möglicherweise andere HMM's mit einer größeren Wahrscheinlichkeit die Beobachtungsfolge, als das trainierte Modell. Ebenfalls anfällig gegenüber Fehler in der Vorverarbeitung scheinen die DEA zu sein. Ein einzelner falsch extrahierter Merkmalsvektor reinitialisiert den Automaten und verhindert eine korrekte Klassifizierung.

Als Grundlage für die Implementierung wurden schließlich der Zweiphasen Algorithmus und der HMM-basierte Ansatz gewählt, da der Zweiphasen Algorithmus eine höhere Fehlertoleranz als die DEA verspricht, und für den HMM-basierten Ansatz freie Software-Bibliotheken verfügbar sind, die eine Implementierung wesentlich beschleunigen.

[3] 10-15 Trainingsdaten pro Geste bei DEA gegenüber mindestens 100 bei HMM.

Grundsätzlich eignen sich beide Verfahren ebenso zur Erkennung von Kopfgesten. Für den Zweiphasen Algorithmus müssten lediglich Schwellwerte adaptiert und für das wahrscheinlichkeitsbasierte Verfahren die sHMM's entsprechend trainiert werden. Zuvor ist es erforderlich die Merkmalsvektoren auf die Bewegungsinformation zu beschneiden, da die Kopfform bei der Erkennung von Kopfgesten eine untergeordnete Rolle spielt.

KAPITEL 5

SYSTEMDESIGN UND IMPLEMENTIERUNG

Nachdem in den vorangehenden Kapiteln die theoretischen Grundlagen und möglichen Verfahren zur Segmentierung und Klassifizierung erläutert wurden, wird in diesem Kapitel deren technische Umsetzung und Integration in das Gesamtsystem behandelt. Zudem werden die relevanten Aspekte und Bereiche dargelegt, die im Zusammenhang mit dem Gesamtsystem und dessen Implementierung stehen.

Als softwaretechnischer Ausgangspunkt der Realisierung wird als erstes die zu Grunde liegende Softwarearchitektur vorgestellt. In den nächsten beiden Abschnitten werden Implementierungsdetails der Segmentierung und Klassifizierung behandelt. Ferner werden Probleme thematisiert, die bei der Umsetzung der verschiedenen Verfahren aufgetreten sind. Nach einer Übersicht der prototypisch realisierten Infrastruktur zur Kommunikation von Kontextwissen, behandelt der folgende Abschnitt die Integration von externen Wissensquellen in die Klassifikatoren. Im Folgenden Abschnitt wird der Versuchsaufbau in der Fahrzeug- bzw. Desktopumgebung beschrieben. Den Abschluss dieses Kapitels bildet die Anbindung des Systems an die BMW-interne MMI Versuchsplattform MIAMII.
Da die programmiertechnische Umsetzung des Gestenerkenners zusammen mit Rudi Lindl erfolgte, wurden die Abschnitte zur Systemarchitektur, zur Kontextübersicht, zum Versuchsaufbau und zur Anbindung an den MIAMII-Demonstrator in Gemeinschaftsarbeit erstellt.

5.1. Systemarchitektur

Überschreiten Projekte in der Softwareentwicklung eine gewisse Größe oder Komplexität, birgt eine der Implementierung zugrundeliegenden Softwarearchitektur beträchtlichen Nutzen. Diese definiert eine sinnvolle Unterteilung des Gesamtsystems in einzelne Komponenten und regelt darüber hinaus das Zusammenspiel der Module durch einheitliche Schnittstellen.

Eine gute Architektur schafft mehr Übersichtlichkeit, reduziert die Komplexität und verringert die Fehlerwahrscheinlichkeit. Ferner vereinfacht sie den Austausch von Modulen, fördert die Wiederverwendbarkeit von Quellcode und erleichtert das Programmieren in Teams.

Vor der eigentlichen Implementierungsphase des Gestenerkennersystems wurde deshalb eine geeignete Systemarchitektur konzipiert. Abgesehen von einer reduzierten Fehlerrate profitiert das System vor allem von der Modularisierung. Die Austauschbarkeit erleichtert die Systemevaluierung und erlaubt den dynamischen Wechsel unterschiedlicher Spotting-, Klassifizierungs- und Segmentierungstechniken zur Laufzeit.
Da das System von zwei Entwicklern erstellt wurde, ermöglichten einheitliche Schnittstellendefinitionen das zeitgleiche, unabhängige Programmieren und vereinfachten den Integrationsprozess. Zudem können künftige Module und Programmergänzungen mit geringem Arbeitsaufwand in das System eingegliedert werden. Die Wiederverwendbarkeit des Quellcodes wird somit gewährleistet.

5.1.1. Grobarchitektur

Das Gesamtsystem läßt sich im Wesentlichen in die drei Komponenten *Datenvorverarbeitung*, *Algorithmik* und *Kommunikation* unterteilen (siehe auch Abbildung 5.1).

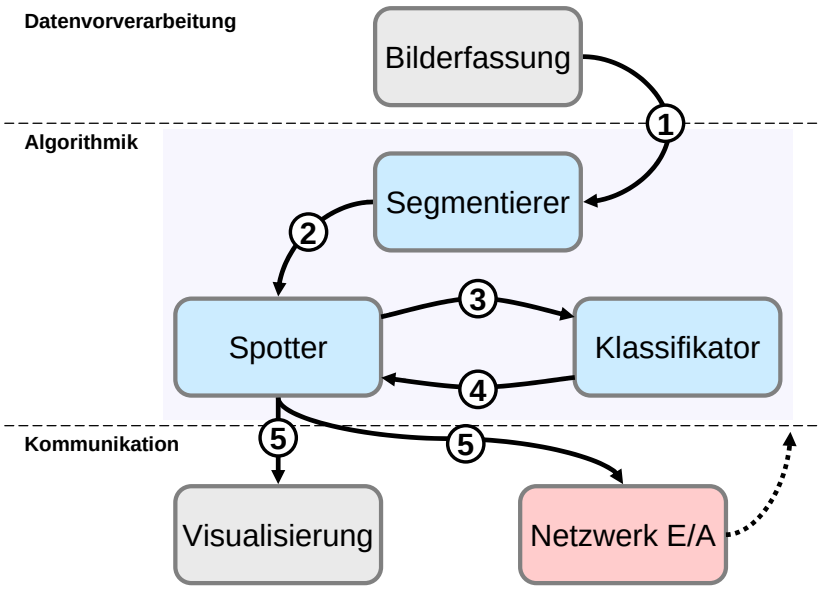


Abbildung 5.1.: Übersicht der Systemarchitektur. Übermittlung der Bilddaten (①), Strom von Merkmalssequenzen (②), Zeitlich segmentierter Strom von Merkmalssequenzen (③), Erkannte Geste (④ und ⑤).

Datenvorverarbeitung: Diese Komponente abstrahiert das Einlesen von diskreten Bildsequenzen. Der genaue Ursprung der Bilddaten ist für die anderen Komponenten transparent. Mögliche Quellen sind Bilder von einer Kamera oder Sequenzen in komprimierter oder unkomprimierter Form auf dem Festspeicher.

Algorithmik: Den eigentlichen Systemkern bildet die Algorithmik-Komponente. Sie ist verantwortlich für das Segmentieren, Spotten und Klassifizieren von Gesten. Die Eingabe bilden unkomprimierte Bildsequenzen von der Datenerfassungskomponente. Als Ausgabe werden die Ergebnisse der Gestenerkennung an die Kommunikationskomponente weitergereicht.

Kommunikation: Die Kommunnikationskomponente nimmt die Ergebnisse der Erkennung in Empfang und visualiert diese entweder direkt auf dem aktuellen Rechensystem oder versendet sie über das Netzwerk. Falls erforderlich werden die Rohdaten transformiert und für das Empfängersystem aufbereitet. Desweiteren erlaubt die Komponente Konfigurationsprogrammen per Fernwartung Einstellungen am System, insbesondere der Algorithmik, vorzunehmen.

5.1.2. Detaillierte Systemarchitektur

Die drei großen Systemkomponenten untergliedern sich in Module, die sich im Allgemeinen auf Programmklassen abbilden lassen. Folgender Abschnitt beschreibt die wichtigsten Module und Klassenabhängigkeiten sowie eine strukturelle Übersicht des Systems in textueller und in struktureller Darstellung (siehe Anhang B für UML-Diagramm). Eine detaillierte Beschreibung des Systems findet sich in einer dem Quellcode beiliegenden Doxygen-Dokumentation [Dim04].

Module der Datenvorverarbeitung

***VideoReader*:** Diese abstrakte Klasse bildet die Basisklasse für alle Bilderfasser. Die Unterklassen sind *WincamReader*, *IrCamReader*, *StereocamReader*, *BmpReader* und *AviReader*.

Die beiden Unterklassen *WincamReader* und *IrCamReader* binden unterschiedliche Kamerasysteme an den Gestenerkenner an. Eine zeitgleiche Bilderfassung von zwei Kameras ermöglicht die Klasse *StereocamReader*. Sie regelt die softwareseitige Bildsynchronisation über eine ins Betriebssystem integrierte Callback-Funktion. Das Einlesen von Einzelbildern, die im BMP- oder JPEG-Dateiformat auf dem Festspeicher vorliegen müssen, bewerkstelligt die Unterklasse *BmpReader*. Komprimierte Bildsequenzen die logisch in einer Datei zusammengefasst sind, können über die Klasse *AviReader* eingelesen werden. Der Zugriff auf die Bilddaten erfolgt bei sämtlichen Unterklassen über die Funktion nextImage, die, mit Ausnahme der Klasse *StereocamReader*, ein einzelnes Bild zurückliefert.

Module der Algorithmik

***Segmentation*:** Die Unterklassen der abstrakten Basisklasse *Segmentation* sind *NirHandSegmentation*, *OtsuThreshSegmentation*, *SimpleThreshSegmentation*, *SkinHandSegmentation* und *StereoSegmentation*. Sie beinhalten die Algorithmen sämtlicher prototypisch implementierter Verfahren zur räumlichen Segmentierung.

***Classifier*:** Eine abstrakte Basisklasse für alle Klassifizierungsverfahren bildet die Klasse *Classifier*. Die Unterklassen sind *RulebasedClassifier*, *RuleMlpClassifier* und *HmmClassifier*.

***Spotter*:** Die abstrakte Basisklasse für alle Spottingverfahren ist die Klasse *Spotter*. Davon abgeleitet sind die Klassen *OneStageSpotter*, *TwoStageSpotter* und *ManualSpotter*.

***Context*:** Die Klasse *Context* ist ein Container für das lokal gespeicherte Kontextwissen.

Alle diese Basisklassen implementieren zu Konfigurationszwecken die abstrakte Klasse *InputHandlerInterface*. Zusätzlich implementieren die Spottingklassen die Klasse *OutputHandler* um Systemausgaben durchführen zu können.
Die Klassen *SimpleThreshSegmentation* (Statisches Schwellwertverfahren), *OtsuThreshSegmentation* (Schwellwertbestimmung nach Otsu) und *NirHandSegmentation* (histogrammbasiertes Segmentieren) erwarten über die $\ll$-Operatorfunktion ein Bild im Graustufenformat und liefern über die Funktion getRegion das segmentierte Objekt, inklusive berechneter Merkmale zurück (siehe auch Abschnitt 3.1.1). Die Klasse *SkinHandSegmentation* (hautfarbenbasierte Segmentierung (siehe auch [WLS02, Lin04])) arbeitet auf RGB-Farbbildern.
Den Spottingmodulen *OneStageSpotter* (Einstufiger, integraler Spottingansatz), *TwoStageSpotter* (Zweistufiger, bewegungsorientierter Spottingansatz) bzw. *ManualSpotter* (manuelles Spotting) werden die Merkmale der Region mittels einer $\ll$-Operatorfunktion weitergereicht. Dem im Spottingkonstruktor übergebenen Klassifikator wird bei erfolgtem Spotting durch die Funktion addSymbolVector eine Sequenz von Merkmalsvektoren überreicht.
Mit getMostProbable liefern die Klassifikatoren *RulebasedClassifier* (Zweistufiger regelbasierter Ansatz), *RuleMlpClassifier* (Dreistufiger regelbasierter Ansatz mit Neuronalen Netz (siehe auch 3-Phasen-Klassifikator in Abschnitt 5.3.1)) bzw. *HmmClassifier* (sHMM-Klassifikator (siehe auch Abschnitt 5.3.2)) die erkannte Geste dem jeweiligem Spottingmodul zurück.
Die *Context*-Klasse dient als lokaler Zwischenspeicher für das aktuelle Kontextwissen (siehe auch Abschnitte 2.5 und 5.4). Dieses wird in einer Hashtabelle gespeichert, die Gestenbezeichnungen Wahrscheinlichkeiten zuordnet. Der Zugriff wird Fremdklassen über die Routine getParameterMap gewährt.

Module der Kommunikation

***OutputHandler*:** Von der abstrakten Basisklasse *OutputHandler* sind die Klassen *ImageOutputHandler*, *ConsoleOutputHandler* und *SocketOutputHandler* abgeleitet.

***InputHandler*:** Der Aufgabenbereich der abstrakten Basisklasse *InputHandler* umfasst Systemeingaben. Einzige Unterklasse bildet die Klasse *SocketInputHandler*.

Die Klassen *SocketOutputHandler* (Systemausgabe über Socketverbindung), *ImageOutputHandler* (Grafische Textdarstellung im Bild) und *ConsoleOutputHandler* (Textdarstellung auf der Konsole) werden bei Klassen, die das Interface *OutputHandlerInterface* implementieren müssen, mit dem Aufruf registerOutputHandler registriert. Mittels der Funktion output erfolgt innerhalb dieser Klassen die Ausgabe – entweder als eine in Zeichen kodierte Geste oder als eine beliebige Textnachricht.
Die Klasse *SocketInputHandler* öffnet einen Kommunikationsendpunkt und übergibt ankommende Netzwerknachrichten an alle registrierten Objekte (Listeners) weiter. Die Registrierung am *InputHandler* erfolgt mittels der Funktion registerListener. Sämtliche Listeners müssen das abstrakte Interface *SocketInputHandlerInterface* implementieren. Im System sind die Klassen *Context*, *Segmentation*, *Classifier* und *Spotter* registriert und erhalten über diesen Weg Konfigurationsnachrichten bzw. aktuelles Kontextwissen.

5.2. Segmentierung und Merkmalsextraktion

Die Segmentierungskomponente des Gestenerkennersystems umfasst vier verschiedene Verfahren zur Handlokalisation. Für die statische Schwellwertsegmentierung und den SMP-Stereoalgorithmus werden im Folgenden die Implementierungsdetails betrachtet. Da die theoretischen Grundlagen für das prinzipielle Vorgehen bereits in Kapitel 3 gelegt wurden, wird hier im Wesentlichen auf Erweiterungen und den konkreten Ablauf der Algorithmen eingegangen.
Ansätze zur histogramm- und hautfarbenbasierten Segmentierung werden im Implementierungsabschnitt der Arbeit von Rudi Lindl betrachtet. Da sich diese beiden Vorgehensweisen auch im realen Einsatz am Besten bewährt haben, wird die hautfarbenbasierte Segmentierung in der Laborumgebung und der histogrammbasierte Ansatz in der Fahrzeugdomäne eingesetzt. Für das Training der stochastischen Modelle und die Evaluierung der Klassifikatoren wird das statische Schwellwertverfahren verwendet, da es die höchste Segmentierungsqualität liefert.

Viele Basisfunktionalitäten wie grundlegende Filteroperationen auf Bildern, Einlese- und Darstellungsroutinen sowie ein Modul zur Kamera-Kalibrierung sind in der frei verfügbaren Bildverarbeitungsbibliothek OpenCV von Intel [Int03] enthalten. Die Methode des SMP-Stereoalgorithmus wurde freundlicherweise von L. Di Stefano [DSMM04] zur Verfügung gestellt.

5.2.1. Statische Schwellwertsegmentierung der Hand

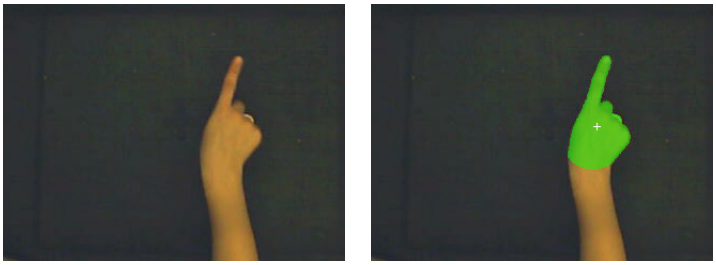
Die statische Schwellwertsegmentierung erfolgt analog zu den Ausführungen im Abschnitt 3.1. Durchgeführt wird das Verfahren dabei auf dem roten Farbkanal, da dieser aufgrund des Haut-

farbenspektrums einen Großteil der Farbinformation enthält (vgl. auch Abbildung 5.2). Der Verfahrenszyklus ist wie folgt strukturiert (siehe auch Abbildung 5.3):

1. Aus dem Bild im RGB-Format $f_{RGB}[x,y]$ wird der rote Farbkanal $f_R[x,y]$ extrahiert und durch den statischen Schwellwert $\vartheta_S = 83$ binarisiert.
2. Von allen Zusammenhangskomponenten im binarisierten Bild, wird die flächengrößte gewählt und markiert, sofern der Flächeninhalt mindestens $\chi = 100$ beträgt.
3. Falls erforderlich wird – nachdem alle Löcher in der Komponente geschlossen sind – durch das geometrische Verfahren aus dem Abschnitt 3.1.2 der Unterarm von der markierten Region getrennt.

(a) (b) (c)

Abbildung 5.2.: Aufgetrennte Farbkanäle des Kamerabildes aus Abbildung 5.3. Roter Farbkanal (a), Grüner Farbkanal (b), Blauer Farbkanal (c).



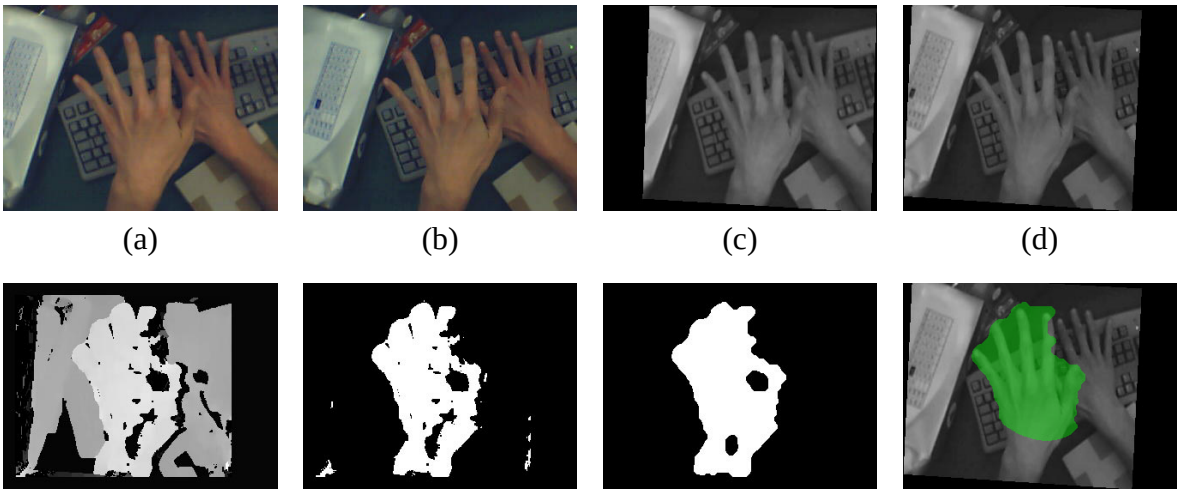
(a) (b)

Abbildung 5.3.: Statische Schwellwertsegmentierung vor schwarzem Hintergrund. Kamerabild im RGB-Format (a), Segmentiertes Bild über statische Schwellwertbildung auf dem roten Farbkanal (b).

5.2.2. 3D Segmentierung

Bevor die Segmentierung durch den SMP-Stereoalgorithmus möglich ist, müssen beide Kamerabilder in Standardform vorliegen. Um die erforderlichen Transformationparameter zu gewinnen, wird eine Offline-Kalibrierung des Stereosystems durchgeführt (siehe auch Abschnitt 3.3.2). Die ermittelten Kameraparameter werden in einer Datei zwischengespeichert

und vor dem Programmlauf ausgelesen. Eine erneute Kalibrierung ist lediglich nötig, wenn sich die Geometrie des Stereosystems ändert.



(a) (b) (c) (d)

(e) (f) (g) (h)

Abbildung 5.4.: Segmentierung durch SMP-Stereoalgorithmus im Desktopbereich. Linkes und rechtes Kamerabild im RGB-Format (a,b), Rektifizierte Kamerabilder in Grauwertdarstellung (c,d), Errechnetes Tiefenbild (e), Schwellwert über das Tiefenbild (f), Morphologische Operationen und Auswahl der flächengrößten Region (g), Nach Unterarmfilterung und Schließen von Löchern finale Ausgabe der Segmentierung (h).

Berechnungszyklus

1. Mittels eines Betriebssystemaufrufs und einer *Callback*-Funktion werden das linke und rechte Kamerabild im RGB-Format $f_{RGB}^{links}[x,y]$ und $f_{RGB}^{rechts}[x,y]$ über die USB-Schnittstelle zeitgleich eingelesen. Von Softwareseite wird dadurch eine bestmögliche Synchronisation der Stereokameras erreicht.

2. Beide RGB-Bilder werden durch die Formel

$$g[x,y] = 0.299 * f_R[x,y] + 0.587 * f_G[x,y] + 0.114 * f_B[x,y]$$

   in ihre Graustufendarstellungen $g^{links}[x,y]$ bzw. $g^{rechts}[x,y]$ überführt. Diese Konvertierung erfolgt auch für Aufnahmen von Infrarotkameras, da die Bilder treiberbedingt stets im RGB Format vorliegen.

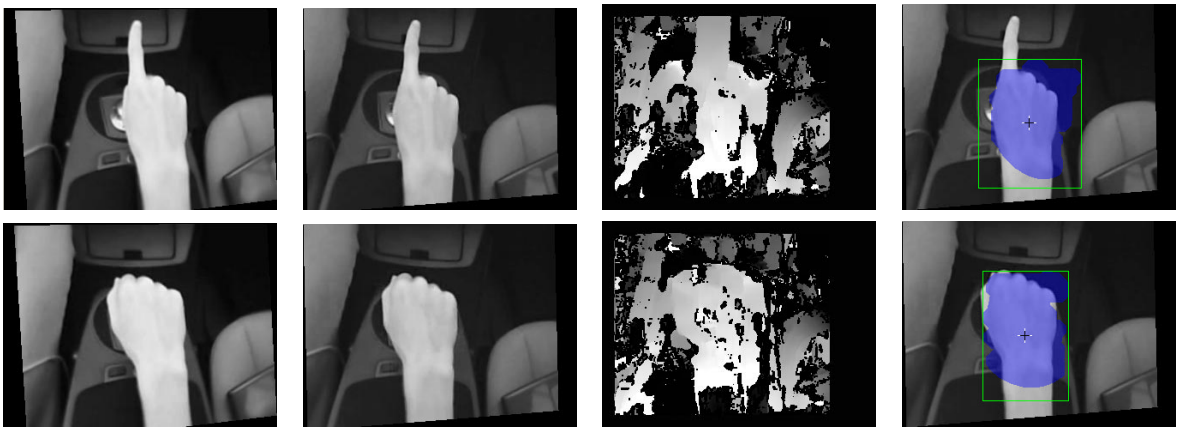
3. Über die von der Kalibrierung ermittelten intrinsischen und extrinsischen Kameraparameter werden beide Graustufenbilder rektifiziert.

4. Liegen die Bilder in Standardform vor, berechnet hieraus der SMP-Algorithmus ein Tiefenbild. Die Entfernungen sind dabei in der Helligkeit der einzelnen Grauwerte kodiert.

Eine statische Schwellwertoperation mit der Schranke ϑ_P blendet den Hintergrund ab einer bestimmten Entfernung aus.

5. Über morphologische Operationen werden die enstandenen Regionen geglättet und verschmolzen. Diese sind im einzelnen ein Closing gefolgt von einem Opening mit einem kreisförmigen strukturierenden Element mit Durchmesser a_P und eine abschließende Dilatation mit einem kreisförmigen Element mit Durchmesser b_P.

6. Die flächengrößte Zusammenhangskomponente wird als Ergebnis zurückgeliefert (siehe auch Abbildungen 5.4 und 5.5). Eine Armfilterung, analog zur statischen Schwellwertsegmentierung erfolgt optional, damit das Verfahren auch zur Segmentierung von Kopfregionen (siehe auch Abbildung 3.13 auf Seite 31) eingesetzt werden kann.



(a) (b) (c) (d)

Abbildung 5.5.: Segmentierung durch SMP-Stereoalgorithmus in der Fahrzeugdomäne. Rektifizierte Infrarot-Kamerabilder (a,b), Errechnetes Tiefenbild (c), Nach Unterarmfilterung und Schließen von Löchern finale Ausgabe der Segmentierung (d).

Problematik

Um die Kamerabilder exakt zeitgleich aufzunehmen, ist eine softwareseitige Synchronisation nicht ausreichend. Aufgrund unterschiedlicher Verschlusszeiten der Kameras und nichtdeterministischen Verzögerungen des USB-Bussystems, treffen die Bilder nicht nur zeitlich versetzt ein, sondern weisen insbesondere bei Bewegungen im Bild verschiedenartige Motionblur Effekte auf. Nachdem der zeitliche Versatz der Bilder die Tiefeninformation des bewegten Objekts verfälscht und die Bewegungsunschärfe eine Korrespondenzfindung stark erschwert (siehe auch Abbildung C.1 auf Seite 105) bleibt dem SMP-Stereoalgorithmus ein realer Systemeinsatz verwehrt.

Eine in Hardware realisierte Synchronisationsvorrichtung und niedrigere Verschlusszeiten würden das Problem lösen. Für die verwendeten USB-Kameras [Cre04] ist jedoch weder eine hardwareseitige Synchronisation vorgesehen noch eine Modifikation der Verschlusszeiten. Alternative Stereokamerasysteme (z. B. STH-MDCS von Videre Design [Vid04] oder Digiclops von Point Grey Research Inc. [Poi04]) kamen wegen der initial postulierten Rahmenbedingungen nicht in Frage (siehe auch Randbedingungen der Diplomarbeit in Abschnitt 1.3).

5.2.3. Merkmalsgewinnung

Neben den Merkmalen Schwerpunkt, Fläche und Hu-Moment-Invarianten (siehe auch Abschnitt 4.1 zur Merkmalsextraktion) wird als weiteres Attribut die Entfernung der Hand zur Kamera ermittelt.

Zur Bestimmung der Tiefeninformation wird dabei zum einen die Flächenänderung der Hand als Entfernungskriterium verwendet (je größer die Fläche desto geringer die Entfernung). Dieses Maß korreliert jedoch nicht eineindeutig mit dem Abstand der Hand von der Kamera, da eine Änderung in der Fläche ebenso durch eine Veränderung in der Gestalt oder Orientierung der Hand herbeigeführt werden kann.
Alternativ dazu wird deshalb zum anderen, analog zu den passiven Stereomethoden in Abschnitt 3.3.2, eine Korrespondenzfindung in den Stereoaufnahmen durchgeführt:

1. In beiden Kamerabildern wird über ein räumliches Segmentierungsverfahren (z. B. Schwellwertsegmentierung, hautfarbenbasierte Segmentierung) die Handregion lokalisiert.

2. In den beiden Handregionen werden die Schwerpunkte berechnet.

3. Über den horizontalen Versatz der beiden Schwerpunkte lässt sich über Triangulierung und Kamerageometrie die Entfernung zur Hand ermitteln.

(a) (b) (c)

Abbildung 5.6.: Beispiel 1 für alternative Bestimmung von Tiefeninformation (Die Hand ist weit entfernt von der Kamera. Der Horizontaler Versatz der Schwerpunkte beträgt 13 Pixel). Segmentierte Hand im linken Kamerabild (a), Segmentierte Hand im rechten Kamerabild (b), Überlagerte Konturen (c).

Eine Triangulierung wird nicht durchgeführt, da die qualitative Entfernungsinformation über den horizontalen Versatz der Regionen in Pixel suffizient ist (vgl. auch Abbildungen 5.6 und 5.7 für Beispiele).

Durch eine Kombination von Flächenänderung und Versatzbestimmung zur Berechnung von Tiefeninformation lässt sich der Einfluss der Fehlerquellen (z. B. ungenügende Kamerasynchronisation, Fehler in der räumlichen Segmentierung) abmindern (vgl. auch Abbildung C.5 im Anhang C.1 für aufgetretene Probleme).

(a) (b) (c)

Abbildung 5.7.: Beispiel 2 für alternative Bestimmung von Tiefeninformation (Die Hand befindet sich nah an der Stereokamera. Der Horizontaler Versatz der Schwerpunkte beträgt 4 Pixel). segmentierte Hand im linken Kamerabild (a), Segmentierte Hand im rechten Kamerabild (b), Überlagerte Konturen (c).

5.3. Klassifizierung

Die folgenden beiden Abschnitte behandeln Implementierungsdetails und Erweiterungen der beiden in Kapitel 4 ausgewählten Verfahren zur Klassifizierung von Gesten. Sowohl der regel- als auch der wahrscheinlichkeitsbasierte Klassifikator werden adaptiert und ergänzt um das aus 17 verschiedenen Gesten bestehende Vokabular dieser Arbeit abdecken zu können. Eine Übersicht, Beschreibung und Nomenklatur der verwendeten Gesten findet sich in Abschnitt 5.6 auf Seite 74. Dynamische Gesten aus diesem Katalog G werden mit g bezeichnet wobei g_f die vom Klassifikator erkannte Geste repräsentiert. Konkrete Zahlenwerte zu den verwendeten Parametern finden sich gesondert im Anhang A.

5.3.1. Erweiterter regelbasierter Klassifikator

Basis für die Implementierung des Klassifikators bildet das Verfahren von Mammen [MCA02] (siehe auch Kapitel 4.2.1). Das zweistufige hierarchische Vorgehen wird in der Implementierung auf weitere reduzierte Kataloge und eine dreistufige Hierarchie erweitert, um das auf

17 Gesten angewachsene Vokabular dieser Arbeit abdecken zu können (siehe auch Abbildung 5.8 und Gestenkatalog im Abschnitt 5.6 auf Seite 74). Der erweiterte Algorithmus wird im Folgenden mit *3-Phasen-Klassifikator* bezeichnet.

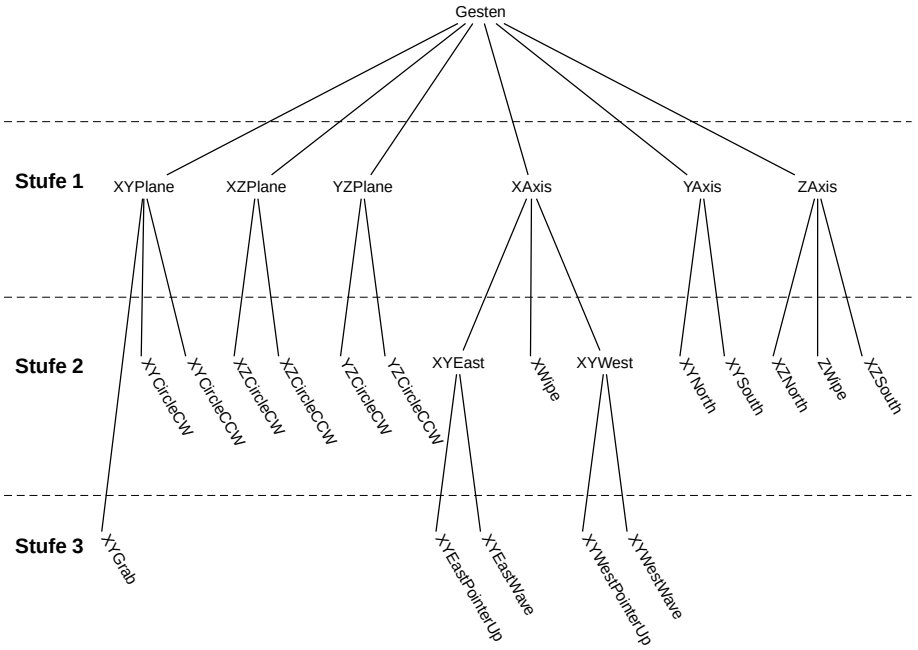


Abbildung 5.8.: Erkennungshierarchie des 3-Phasen-Klassifikators (die Gesten befinden sich in den Blättern)

Verwendete Merkmale

Die Merkmale des 3-Phasen-Klassifikators lassen sich in bewegungs- und gestaltbeschreibende Attribute unterteilen. Erstere dienen den Hierarchiestufen 1 und 2 zur groben Einordnung der Geste und letztere ermöglichen in der Hierarchiestufe 3 eine Verfeinerung der Erkennung über die Handform. Zur einfacheren Schreibweise werden im Folgenden die beiden Bezeichnungen $\vec{m}_t^{regel}$ und $\vec{m}$ für Merkmalsvektorsequenz synonym verwendet.

	Merkmal	Bezeichnung
Bewegungsattribute	• X-Komponente des Schwerpunktes • Y-Komponente des Schwerpunktes • Z-Komponente (vgl. Abschnitt 5.2.3) • Fläche der Region	• $X_C(t)$ • $Y_C(t)$ • $Z_C(t)$ • $A(t)$
Gestaltbeschreibende Attribute	• Hu-Moment 1 bis Hu-Moment 6	• $H_1(t), \ldots, H_6(t)$

Tabelle 5.1.: Verwendete Merkmale des 3-Phasen-Klassifikators zum Zeitpunkt t. Sie werden zum Merkmalsvektor $\vec{m}_t^{regel}$ zusammengefasst.

Vorverarbeitung

Die Rohdaten der Trajektorien $X_C(t)$, $Y_C(t)$, $Z_C(t)$ und $A(t)$ für $0 \leq t \leq T$ werden zu Beginn des Verfahrens normiert, um einen Vergleich der Kurven zu ermöglichen. $X_C(t)$, $Y_C(t)$ und $Z_C(t)$ werden dazu angehoben bzw. abgesenkt, so dass der erste Funktionswert 0 ergibt:

$$\begin{aligned} X_C^{norm}(t) &= X_C(t) - X_C(0) \\ Y_C^{norm}(t) &= Y_C(t) - Y_C(0) \\ Z_C^{norm}(t) &= [Z_C(t) - Z_C(0)] \cdot \xi_Z \end{aligned}$$

Zusätzlich werden die Funktionswerte von $Z_C(t)$ mit dem Faktor $\xi_Z = 4$ multipliziert, um die gröbere Granularität der Tiefeninformation zu kompensieren.

Die Flächentrajektorie $A(t)$ wird zuerst auf den Wertebereich zwischen 0 und 1 skaliert, mit dem Faktor ξ_A = -100 multipliziert und schließlich analog zu den anderen Trajektorien auf den Startwert 0 abgesenkt:

$$\begin{aligned} A'(t) &= \frac{A(t)}{\max_{0 \leq t \leq T} A(t)} \cdot \xi_A \\ A^{norm}(t) &= A'(t) - A'(0) \end{aligned}$$

Die normierten Kurven werden erst mit einem Medianfilter (Kernelgröße $\eta_M = 3$) und danach mit einem Gaußfilter (Kernelgröße $\eta_G = 5$) geglättet. Ergebnis sind die normierten und geglätteten Kurvenverläufe $X_C^{gl}(t)$, $Y_C^{gl}(t)$, $Z_C^{gl}(t)$ und $A^{gl}(t)$ (vgl. auch Abbildung 5.10(a) mit 5.10(b)).
Die kodierte Tiefeninformation in den Trajektorien $A^{gl}(t)$ und $Z_C^{gl}(t)$ ist oftmals fehlerbehaftet und wird deshalb vor der Trajektoriensondierung zur Kurve $\tilde{Z}_C^{gl}(t) = \min[A^{gl}(t), Z_C^{gl}(t)]$ fusioniert (siehe auch Abschnitt 5.2.3).

Hierarchiestufe 1: Sondierung der dominanten Trajektorien

Anschließend werden analog zu Abschnitt 4.2 von $X_C^{gl}(t)$, $Y_C^{gl}(t)$ und $\tilde{Z}_C^{gl}$ die maximalen Amplituden δ_X, δ_Y und δ_Z ermittelt und anhand dessen die Merkmalsvektorsequenz $\vec{m}$ durch folgenden Entscheidungsgraph den drei Mengen S_X, S_Y und S_Z zugewiesen. Der Wert f_I steht dabei für $\arg_I \max \delta_I$ mit $I \in \{X, Y, Z\}$ und θ bezeichnet je nach Indizes unterschiedliche Schranken.

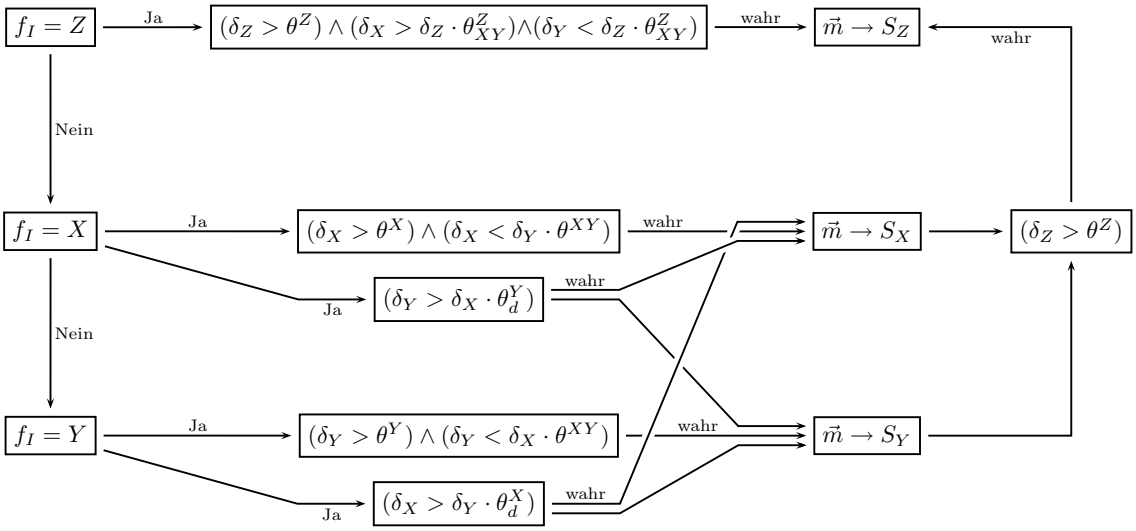


Abbildung 5.9.: Entscheidungsgraph des 3-Phasen-Klassifikators

Die erste Stufe ist mit der Einteilung von $\vec{m}^{regel}$, über die dominanten Trajektorien, in folgende reduzierte Kataloge beendet.

XAxis:	$\vec{m} \in S_X \wedge \vec{m} \notin (S_Y \cup S_Z)$	XYPlane:	$\vec{m} \in S_X \wedge \vec{m} \in S_Y \wedge \vec{m} \notin S_Z$
ZAxis:	$\vec{m} \in S_Z \wedge \vec{m} \notin (S_X \cup S_Y)$	XZPlane:	$\vec{m} \in S_X \wedge \vec{m} \in S_Z \wedge \vec{m} \notin S_Y$
YAxis:	$\vec{m} \in S_Y \wedge \vec{m} \notin (S_X \cup S_Z)$	YZPlane:	$\vec{m} \in S_Y \wedge \vec{m} \in S_Z \wedge \vec{m} \notin S_X$
Garbage:	sonst		

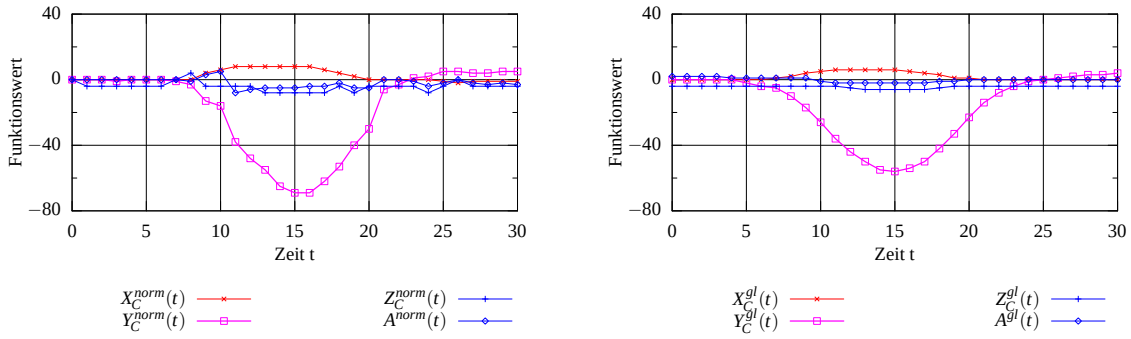
Besonderer Bedeutung kommt der Klasse Garbage zu: Kann die aktuelle Merkmalsvektorsequenz keinem reduzierten Katalog zugeordnet werden, wird der Algorithmus abgebrochen und als Ergebnis „nicht erkannt“ bzw. Garbage ausgegeben.

Hierarchiestufe 2: Funktionsanalyse

Wurde die Einteilung in reduzierte Kataloge vorgenommen, werden die Kurvenverläufe der im jeweiligen Katalog dominanten Trajektorien untersucht. In diesem Abschnitt wird lediglich auf die Analyse der Zeigegesten XYEast bzw. XYWest und der Wischgesten XWipe bzw. ZWipe eingegangen. Für die Gesten XZNorth bzw. XZSouth und XYNorth bzw. XYSouth ist das Vorgehen analog zu den Zeigegesten XYEast und XYWest. Die Analyse von Kreisgesten wurde bereits in Abschnitt 4.2.1 behandelt (vgl. auch Abbildung 5.12). Weitere Diagramme mit Funktionsverläufen zu sämtlichen Gesten finden sich im Anhang E.

Zeigegesten Bei allen Zeigegesten muss die Startposition näherungsweise der Endposition der Ausführung entsprechen. Es gilt also $X_C^{gl}(0) \approx X_C^{gl}(T)$. Über das Vorzeichen der Summe $P = \sum_{i=0}^{T} X_C^{gl}(i)$ lässt sich somit auf die Richtung der Bewegung schließen. Ist $P < 0$ wird im reduzierten Katalog XAxis die Geste XYWest erkannt, ansonsten XYEast. Werden die Zeigegesten mit einer zu starken Ausholbewegung ausgeführt, kann

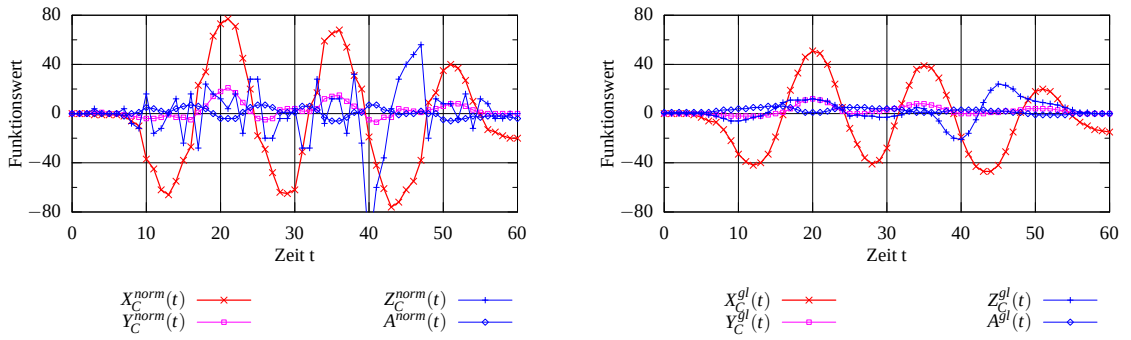
das Vorzeichen der Summe wechseln und fälschlicherweise die komplementäre Geste erkannt werden.
Die Erkennung für die Gesten XYNorth, XYSouth bzw. XZNorth, XZSouth erfolgt analog dazu im reduzierten Katalog YAxis bzw. ZAxis (siehe auch Abbildung 5.10 für die Funktionsverläufe einer XYNorth-Geste).



(a) (b)

Abbildung 5.10.: Trajektorien einer XYNorth-Geste. $Y_C(t)$ ist dominant. Normierte Trajektorien (a), Geglättete, normierte Trajektorien (b).

Wischgesten Für Gesten mit einer einzelnen dominanten Trajektorie wird die Anzahl e der Extremwerte im dominanten Funktionsverlauf gezählt. Überschreitet e die Schranke θ_e wird innerhalb des reduzierten Kataloges XAxis die Geste XWipe und innerhalb von ZAxis die Geste ZWipe erkannt. Trotz Median- und Gaußglättung der Kurven können sich minimale Kurvenschwankungen ergeben, die als Extremwert detektiert werden. Um eine Fehlklassifizierung zu vermeiden müssen deshalb Extremwerte einen Mindestabstand von anderen Extremwerten und eine Mindesthöhe aufweisen (siehe Abbildung 5.11 für die Funktionsverläufe einer XWipe-Geste).



(a) (b)

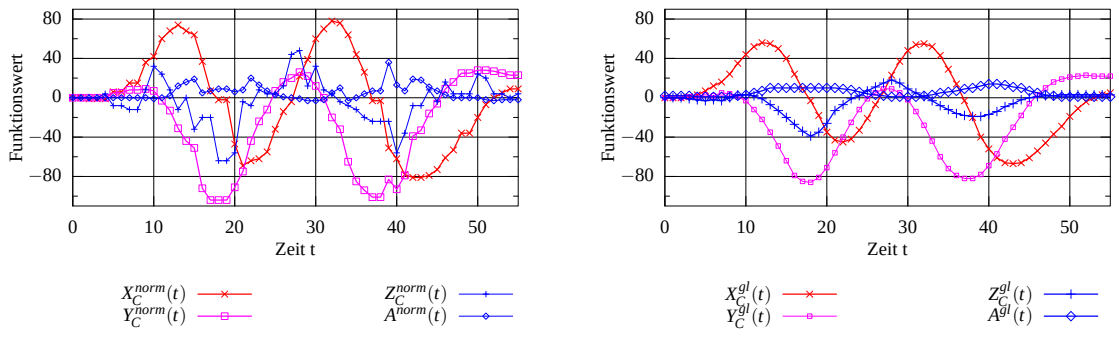
Abbildung 5.11.: Trajektorien einer XWipe-Geste. $X_C(t)$ ist dominant. Normierte Trajektorien (a), Geglättete, normierte Trajektorien (b).

Hierarchiestufe 3: Gestaltanalyse

Im letzten optionalen Schritt des Klassifikators wird die Form der Hand überprüft. Diese ist in den sechs Hu-Momenten $H_1, \ldots, H_6$ kodiert und wird mittels eines neuronalen Netzes klassifiziert. Dazu wurden mit Hilfe der TORCH-Bibliothek [CBM03] über einen Trainingssatz von je 20 statischen Gesten für die Handformen „Faust“ (HandFist), „Hand in Zeigeform“ (HandPointer) und „Hand in Greifform“ (HandGrab) drei NN's erstellt. Die Handkante als statische Geste konnte nicht modelliert werden, da einerseits die benutzerspezifische Varianz der Ausführung zu groß ist und andererseits die geometrische Filteroperation der Unterarmkomponente Schwierigkeiten mit zu schmalen Regionen aufwies (siehe auch Abbildung C.6 im Anhang).
Als Eingabe erwartet das Neuronale Netz sechs gestaltbeschreibenden Hu-Momente $H_1, \ldots, H_6$. Für den übergebenen Merkmalsvektor, bestehend aus den Hu-Momenten, werden die Erkennungswahrscheinlichkeiten $P(\text{HandPointer})$, $P(\text{HandFist})$ und $P(\text{HandGrab})$ ausgegeben. Ergebnis der Teilklassifizierung ist die statische Geste

$$g_f^s = \arg_{g^s} \max[P(\text{HandPointer}), P(\text{HandFist}), P(\text{HandGrab})] \tag{5.1}$$

Während einer Bewegung wird das NN zu den Zeitpunkten $t = 0$ (Start der dynamischen Geste), $t = \lceil \frac{T}{2} \rceil$ (Mitte der Geste) und $t = T$ (Ende der Geste) mit den jeweils vorliegenden Hu-Momenten $H_1(t), \ldots, H_6(t)$ beaufschlagt und ausgewertet.
Sind die statischen Gesten beispielsweise drei Mal übereinstimmend HandPointer und die erkannte dynamische Geste nach der 2. Hierarchiestufe XYEast, wird als Ergebnis XYEast-PointerUp zurückgeliefert. Analog werden die dynamischen Gesten XYWestPointerUp, XYWestWave und XYEastWave erkannt.
Für die XYGrab-Geste, muss die Abfolge der statischen Gesten HandFist, HandGrab HandFist auftreten und nach der 1. Hierarchiestufe der reduzierte Katalog XYPlane gewählt werden (siehe auch Gestenkatalog in Abschnitt 5.6 für Achsenbezüge).



(a) (b)

Abbildung 5.12.: Trajektorien einer XYCircleCW-Geste. $X_C(t)$ und $Y_C(t)$ sind dominant. Normierte Trajektorien (a), Geglättete, normierte Trajektorien (b).

5.3.2. sHMM-Klassifikator

Die theoretischen Grundlagen für die HMM-basierte Klassifizierung zeitlicher Prozesse wurden in Kapitel 4.3 gelegt. In diesem Abschnitt werden die Implementierung und die verwendeten Parameter konkretisiert. Softwaretechnische Basis des Klassifikators bilden sHMM der frei verfügbare TORCH-Bibliothek [CBM03], die zahlreiche Machine-Learning-Algorithmen in sich vereint, sowie Werkzeuge zum Training von Hidden Markov Modellen bereitstellt.

Verwendete Merkmale

Im Desktopbereich setzt sich der Merkmalsvektor $\vec{m}_t^{hmm\_d}$ aus Relativdaten von Trajektorien-, sowie Gestaltinformationen zusammen. Diese Komposition hat sich in der Fahrzeugdomäne jedoch nicht bewährt, da aufgrund der schlechteren Segmentierungsqualität insbesondere die Handform nicht korrekt extrahiert wird. Verringerte Erkennungsraten sind die Folge, da die Modelle im Desktopbereich unter optimalen Segmentierungsbedingungen trainiert wurden. Durch die Beschränkung des Merkmalsvektors $\vec{m}_t^{hmm\_f}$ auf die relative x- und y-Komponente der Handregion profitiert die Erkennungsleistung etwas (siehe auch Evaluierungskapitel 6). Der Merkmalsvektor für den sHMM-basierten Klassifikator wird zur vereinfachten Schreibweise im Folgenden mit $\vec{m}$ bezeichnet.

	Merkmal	Bezeichnung
Bewegungsattribute	• relative X-Komponente des Schwerpunktes • relative Y-Komponente des Schwerpunktes • relative Z-Komponente (vgl. Abschnitt 5.2.3) • relative Fläche der Region	• $\Delta X_C(t)$ • $\Delta Y_C(t)$ • $\Delta Z_C(t)$ • $\Delta A(t)$
Gestaltbeschreibende Attribute	• Hu-Moment 1 • Hu-Moment 2 • zeitliche Änderung von Hu-Moment 1 • zeitliche Änderung von Hu-Moment 2	• $H_1(t)$ • $H_2(t)$ • $\Delta H_1(t)$ • $\Delta H_2(t)$

Tabelle 5.2.: Verwendete Merkmale des sHMM-basierten Klassifikators zum Zeitpunkt t in der Desktopdomäne. Werden zum Merkmalsvektor $\vec{m}_t^{hmm\_d}$ zusammengefasst.

	Merkmal	Bezeichnung
Bewegungsattribute	• relative X-Komponente des Schwerpunktes • relative Y-Komponente des Schwerpunktes	• $\Delta X_C(t)$ • $\Delta Y_C(t)$

Tabelle 5.3.: Verwendete Merkmale des sHMM-basierten Klassifikators zum Zeitpunkt t in der Fahrzeugdomäne. Werden zum Merkmalsvektor $\vec{m}_t^{hmm\_f}$ zusammengefasst.

Training

Die 17 Modelle wurden mit je 90 Gesteninstanzen trainiert, die von 6 verschiedenen Versuchspersonen unter optimalen Aufnahmebedingungen getätigt wurden. Eine Mittelwertsbereinigung der Trainingsdaten hat zu einer robusteren Erkennung beigetragen. Für alle Merkmalsvektoren jeder Geste werden dazu die Mittelwertsvektoren gebildet und anschließend von den Trainingsdaten subtrahiert. Eine Varianzbereinigung hatte hingegen keine Auswirkungen auf die Erkennungsleistung.
Vor dem Training werden die von Morguet vorgeschlagenen Parameterinitialisierungen vorgenommen (siehe auch Abschnitt 4.3). Darüber hinaus hat sich eine variable Zustandsanzahl $|Z_g|$ der Modelle als vorteilhaft erwiesen. Die Datenverteilung wird mit 200 Gaußmixturen angenähert. Eine größere Zahl an Mixturen trägt zwar zur besseren Erkennung bei, resultiert aber auch in wesentlich längeren Berechnungszeiten (siehe auch Tabelle 5.4 und Parametrisierung in Abschnitt 6.2).

sHMM	Gaußmixturen	$\|Z_g\|$	$\overline{T}_g$	Varianz
λ_{XWipe}	200	15	30.85	8.64
$\lambda_{\text{XYCircleCCW}}$	200	25	49.96	15.31
$\lambda_{\text{XYCircleCW}}$	200	23	48.37	15.41
$\lambda_{\text{XYEastPointerUp}}$	200	10	19.72	6.30
$\lambda_{\text{XYEastWave}}$	200	8	16.50	4.14
λ_{XYGrab}	200	10	21.33	5.10
λ_{XYNorth}	200	10	20.22	6.75
λ_{XYSouth}	200	9	19.27	5.16
$\lambda_{\text{XYWestPointerUp}}$	200	10	21.03	8.28
$\lambda_{\text{XYWestWave}}$	200	8	16.91	5.48
$\lambda_{\text{XZCircleCCW}}$	200	19	40.69	10.08
$\lambda_{\text{XZCircleCW}}$	200	21	40.71	13.68
λ_{XZNorth}	200	9	20.04	6.68
λ_{XZSouth}	200	9	18.40	5.04
$\lambda_{\text{YZCircleCCW}}$	200	19	39.38	8.92
$\lambda_{\text{YZCircleCW}}$	200	22	44.28	10.55
λ_{ZWipe}	200	17	33.94	9.78

Tabelle 5.4.: Statistik zur Gestenausführung und gewählte Parameter der sHMM. ($|Z_g|$: Gewählte Zustandsanzahl, $\overline{T}_g$: Mittlere Ausführungsdauer der Geste in Bildern mit Varianz).

Erkennungsvorgang

Um einen Merkmalsvektor $\vec{m}$ einer Geste zuzuordnen, wird für alle Modelle über den Viterbi-Algorithmus die Erzeugungswahrscheinlichkeit berechnet. Ergebnis ist die Geste g_f des Modells mit dem höchsten Ausgangs-Score.

$$g_f = \arg_g \max_{\forall g \in G} \lambda_g(\vec{m}) \tag{5.2}$$

Probleme treten auf, wenn alle Erzeugungswahrscheinlichkeiten sehr kleine Werte aufweisen, weil beispielsweise eine Bewegung durchgeführt wurde, die außerhalb des Gestenvokabulars liegt. Mit obigem Vorgehen würde nach der Maximabestimmung in jedem Fall ein Ergebnis innerhalb des Gestenkatalogs G vorliegen. Um dies zu verhindern, werden vor der Maximabestimmung der Scores alle Modelle, die unter einem Schwellwert τ liegen von der Erkennung ausgeschlossen. Die übrigen Modelle werden in der Menge Λ_τ zusammengefasst.

$$\Lambda_\tau = \{\lambda_g | \lambda_g(\vec{m}) > \tau\} \tag{5.3}$$

Fallen alle Scores unter den Schwellwert, wird als Ergebnis die Garbage-Klasse zurückgeliefert. Die Formel (5.2) wird somit zu

$$g_f = \begin{cases} \arg_g \left[\max_{\forall \lambda_g \in \Lambda_\tau} \lambda_g(\vec{m})\right] & \text{für } |\Lambda_\tau| \neq 0 \\ \text{Garbage} & \text{für } |\Lambda_\tau| = 0 \end{cases} \tag{5.4}$$

erweitert. Indiz für eine falsche Klassifizierung sind oftmals geringe Score-Differenzen k zwischen den Erzeugungswahrscheinlichkeiten der beiden führenden sHMM λ_{g_f} und λ_{g_2}. In Erkennungsvorgängen, bei denen $k = \left|\lambda_{g_f}(\vec{m}) - \lambda_{g_2}(\vec{m})\right|$ unter eine Schranke υ fällt, wird deshalb die Garbage-Klasse als Ergebnis geliefert.

5.4. Kontextwissen

In heutigen Fahrzeugen fallen eine Vielzahl von Zustandsinformationen an, die als zusätzliche Wissensquellen interpretiert werden können und im Folgenden mit Kontextwissen bezeichnet werden. In diesem Zusammenhang wird zwischen System-, Umgebungs- und Benutzerkontext unterschieden (siehe auch Kapitel 2.5 und [Alt04]). Die Auswertung und Nutzung verspricht nicht allein für die Klassifizierung und das Spotting dynamischer Gesten einen Gewinn an Effektivität und Robustheit (siehe Tabelle 5.5 für Beispiele) sondern auch für andere Eingabemodalitäten, wie beispielsweise Sprach- und Emotionserkennung.

5.4.1. Infrastruktur

Um Kontextwissen für das Erkennersystem nutzbar zu machen, wurde eine Infrastruktur zur Verwaltung und Kommunikation von Kontextwissen konzipiert und ein vertikaler Durchstich

Beispiel	Beschreibung
Schaltvorgang	Führt der Nutzer im Fahrzeug einen Schaltvorgang durch, befindet sich seine Hand im Gesteninteraktionsbereich. Die Bewegung der Hand während des Schaltens könnte folglich fälschlicherweise als Geste interpretiert werden. Eine entsprechende Auswertung des Systemkontextes durch das Spottingmodul könnte diese Bewegung als nicht relevant einstufen.
Listennavigation	Navigiert der Nutzer innerhalb einer Desktopumgebung durch eine Liste, steht ihm ein vermindertes Kommunikationvokabular (nächstes Element, vorheriges Element, Auswahl) zur Verfügung. Bei entsprechender Auswertung des Systemkontextes durch das Klassifizierungsmodul könnte deshalb der Gestenraum eingeschränkt und somit die Erkennungsleistung gesteigert werden.
Systemdialog	Ebenfalls eine Einschränkung des Gestenvokabulars ist bei sehr einfachen Systemdialogen denkbar. Kann der Nutzer beispielsweise, bei einem eingehendem Anruf nur zwischen Anruf annehmen und Anruf ablehnen wählen, sind eine bejahende (zum Beispiel Kopfnicken) und eine verneinende Geste (zum Beispiel Kopfschütteln) ausreichend.

Tabelle 5.5.: Beispiele für die Nutzung von Kontextwissen in der Gestenerkennung.

implementiert (siehe Abbildung 5.13). Den Kern bildet der Kontextserver, dem die Aufgabe der Interpretation und Aufbereitung von Kontextwissen und der Fusionierung von Erkennungsergebnissen zukommt. Formal erfolgt die Aufbereitung über komplexe Entscheidungsbäume, deren Implementierung nicht Teil dieser Arbeit war. Ergebnis der Interpretation ist der Modalitätenkontext, der in dieser Arbeit als $\vec{K}$ repräsentiert wird. Dieser Vektor gibt hierbei für die einzelnen Gesten bzw. Modalitäten die *Kontextwahrscheinlichkeit* an, dass sie zum aktuellen Zeitpunkt auftreten können.

$$\vec{K} = \begin{pmatrix} P_K(Geste_1) \\ P_K(Geste_2) \\ \vdots \\ P_K(Geste_n) \end{pmatrix} = \begin{pmatrix} P_K(\mathsf{XWipe}) \\ P_K(\mathsf{XYCircleCW}) \\ \vdots \\ P_K(\mathsf{ZWipe}) \end{pmatrix} \tag{5.5}$$

Angebunden an die zentrale Instanz Kontextserver, sind zum einen die Kontexterzeuger und zum anderen die Kontextverwerter bestehend aus den Interaktionsmodulen. Unter Kontexterzeuger wird in diesem Zusammenhang *Umgebungskontext*, *Benutzerkontext*, *Systemkontext* und zu Evaluierungszwecken Debugkontext (siehe auch Kontextsimulator im Anhang D.2) verstanden. Dabei ist eine strikte Trennung in Kontexterzeuger und -verwerter nicht möglich, da letztere über einen Rückkanal Informationen (*Feedback-Kontext*) wie zum Beispiel Dynamik der Gestenausführung oder biometrische Daten zurückpropagieren können. Für die Art und Weise der Nutzung von übermittelten Kontextwissen sind die jeweiligen Erkennungsmodule verantwortlich. Die Kommunikation zwischen den Komponenten erfolgt über TCP/IP

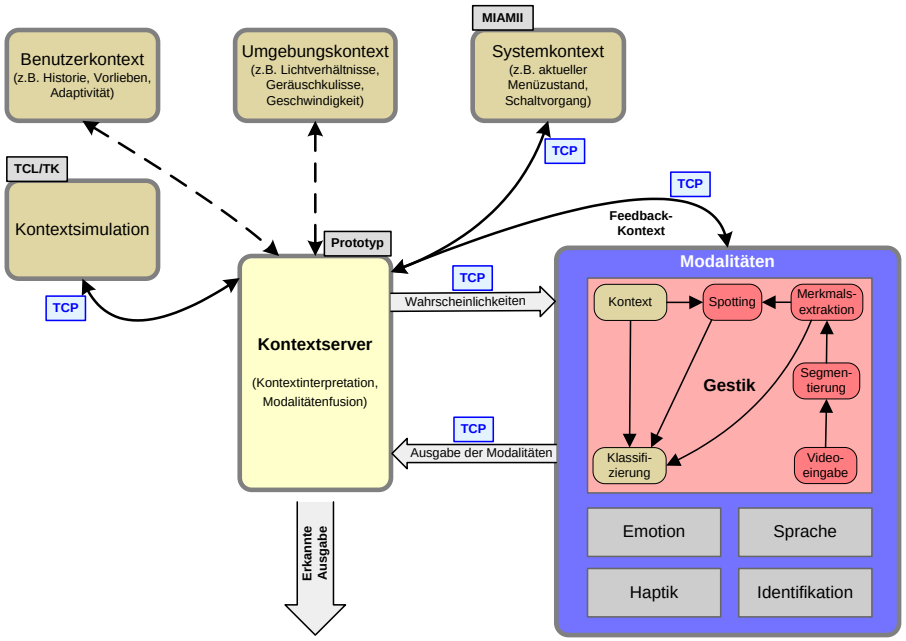


Abbildung 5.13.: Infrastruktur zur Nutzung von Kontextwissen im Fahrzeug.

Socketverbindungen. Im Rahmen dieser Arbeit wurde der Systemkontext mit Hilfe des MIAMII Frameworks simuliert und durch den prototypisch implementierten Kontextserver auf die *Kontextwahrscheinlichkeiten* $\vec{K}$ abgebildet. Verwerter sind die angebundenen Module Klassifizierung und Spotting.
Je nach verwendetem Klassifikator gestaltet sich die Verarbeitung und Integration von Kontextwissen unterschiedlich. Deshalb wird im Folgenden die Integration für jedes Erkennungsverfahren gesondert behandelt. Die Kontextintegration in die Spotting Module wird in der Diplomarbeit von Rudi Lindl [Lin04] thematisiert.

5.4.2. Integration von Kontextwissen in den 3-Phasen-Klassifikator

Fehlentscheidungen des 3-Phasen-Klassifikators werden in erster Linie durch verfälschte Trajektoriendaten herbeigeführt. Die Ursachen dafür liegen meist in einer ungenügenden Segmentierungsqualität, aber auch in der nutzerspezifischen Ausführung der Gesten begründet. Werden gezielt einzelne Kurvenverläufe in der Vorverarbeitungsphase kontextabhängig verstärkt bzw. abgeschwächt, wird die Angriffsfläche potentieller Fehler verringert. Zudem trägt die kontextabhängige Revalidierung der erkannten Geste zu einer Verringerung der Falschakzeptanzrate bei. Die konkrete Verwirklichung der Kontextintegration wird im Folgenden dargelegt.

Skalierung der Trajektorien

Die Kontextwahrscheinlichkeiten $\vec{K}$, die nur für einzelne Gesten existieren, werden umgewandelt in die Wahrscheinlichkeiten $P(X_C)$, $P(Y_C)$, $P(Z_C)$ und $P(A)$ für das Auftreten der vier normierten und geglätteten Trajektorien $X_C^{gl}(t)$, $Y_C^{gl}(t)$, $Z_C^{gl}(t)$ und $A^{gl}(t)$. Dafür werden ausgehend von den Blättern der Gestenhierarchie (siehe Abbildung 5.8) die Kontextwahrscheinlichkeit in Richtung Wurzel des Baums propagiert. Sei n ein Knoten des Baums und n_g dessen semantische Bedeutung (unterliegende Geste bzw. reduzierter Katalog). Enthalte die Menge B alle Blattknoten des Baums und die Menge K_n alle Kindknoten des Knotens n. Dann berechnen sich die Wahrscheinlichkeiten rekursiv wie folgt.

$$\begin{aligned} \forall n \in B \quad P(n) &= P_K(n_g) \\ \forall n \in \overline{B} \quad P(n) &= \max_{\forall \hat{n} \in K_n} [P(\hat{n})] \end{aligned} \tag{5.6}$$

Die Auftrittswahrscheinlichkeiten für die dominanten Trajektorien setzen sich wie folgt aus den jeweiligen Wahrscheinlichkeiten der reduzierten Kataloge zusammen, wobei $P(n_g) = P(n)$ gilt:

$$\begin{aligned} P(X_C) &= \max\left[P(\mathsf{XAxis}), P(\mathsf{XYPlane}), P(\mathsf{XZPlane})\right] \\ P(Y_C) &= \max\left[P(\mathsf{YAxis}), P(\mathsf{XYPlane}), P(\mathsf{YZPlane})\right] \\ P(Z_C) &= \max\left[P(\mathsf{ZAxis}), P(\mathsf{XZPlane}), P(\mathsf{YZPlane})\right] \\ P(A) &= P(Z_C) \end{aligned} \tag{5.7}$$

Im letzten Schritt der Vorverarbeitung werden die Trajektorien mit ihren entsprechenden Wahrscheinlichkeiten skaliert:

$$\begin{aligned} \tilde{X_C}(t) &= X_C(t) \cdot P(X_C) \\ \tilde{Y_C}(t) &= Y_C(t) \cdot P(Y_C) \\ \tilde{Z_C}(t) &= Z_C(t) \cdot P(Z_C) \\ \tilde{A}(t) &= A(t) \cdot P(A) \end{aligned} \tag{5.8}$$

Werden zum Beispiel durch das Kontextwissen alle Gesten mit dominanter Bewegung auf der Z-Achse ausgeblendet, also $P(\mathsf{XZNorth}) = 0$, $P(\mathsf{XZSouth}) = 0$, $P(\mathsf{ZWipe}) = 0$, $P(\mathsf{XZCircleCW}) = 0$, $P(\mathsf{XZCircleCCW}) = 0$, $P(\mathsf{YZCircleCW}) = 0$ und $P(\mathsf{YZCircleCCW}) = 0$, dann berechnen sich nach Formel (5.6) die Wahrscheinlichkeiten der reduzierten Kataloge ZAxis, XZPlane und YZPlane zu 0. Wegen Formel (5.7) werden die Faktoren $P(Z_C)$ und $P(A)$ ebenso 0 und skalieren über Formel (5.8) die Trajektorien $Z_C(t)$ und $A(t)$ auf die Nullinie. Der Gestenraum wird dadurch auf die verbleibenden Gesten in der XY-Ebene reduziert, da $Z_C(t)$ und $A(t)$ nicht mehr dominant werden können. Etwaige Fehler in der Tiefeninformation haben damit keinen Einfluss auf das Ergebnis.

Revalidierung der erkannten Geste

Wurde die Erkennungshierarchie des regelbasierten Klassifikators komplett durchlaufen, wird nachträglich überprüft, ob das Ergebnis mit dem aktuellen Kontextwissen vereinbar ist. Dazu wird die erkannte Geste g_f mit ihrer korrespondierenden Kontextwahrscheinlichkeit $P_K(g_f)$ revalidiert. Ist $P_K(g_f) < d$, so wird das Ergebnis verworfen und als Resultat die Geste oder der reduzierte Katalog eine Hierarchiestufe (siehe Abbildung 5.8) höher zurückgeliefert, sofern diese Kontextwahrscheinlichkeit größer als d ist. Wird die Wurzel der Gestenhierarchie erreicht ist das Ergebnis die Garbage-Klasse.

5.4.3. Integration von Kontextwissen in den sHMM-Klassifikator

Im Gegensatz zum 3-Phasen-Klassifikator erfolgt beim sHMM-basierten Vorgehen die Klassifizierung nicht hierarchisch. Ausschlaggebend für das Ergebnis sind lediglich die Erzeugungswahrscheinlichkeiten der einzelnen sHMM. Diese bilden einen geeigneten Angriffspunkt für die Manipulation durch Kontextwissen. Durch eine gezielte Modifikation lassen sich kontextabhängig Gesten für die Erkennung bevorzugen bzw. benachteiligen.

Der Wertebereich der Erzeugungswahrscheinlichkeiten liegt wegen der logarithmischen Numerik im Intervall $]-\infty;0]$. Da bei kontinuierlichen HMM die Emissionswahrscheinlichkeiten der Zustände durch Überlagerung von Gaußfunktionen approximiert (Gaußmixturen) werden und diese Wahrscheinlichkeitsdichten auch Werte ≥ 0 annehmen, liegen die Erzeugungswahrscheinlichkeiten in der Praxis erfahrungsgemäß im Intervall $[-30000;280]$.
Eine einfache Multiplikation der Kontextwahrscheinlichkeiten $\vec{K}$ auf die Erzeugungswahrscheinlichkeiten liefert somit nicht den gewünschten Effekt, da letztere vorzeichenabhängig im Wert angehoben oder abgesenkt würden. Zudem wäre die Skalierung nicht linear. Werte um 0 würden marginal, und Werte nahe der linken Intervallgrenze übermäßig stark von der Skalierung beeinflusst werden.

Skalierung der Erzeugungswahrscheinlichkeiten

Um dennoch eine Kontextintegration zu ermöglichen werden vor der Multipliation mit $\vec{K}$ alle Erzeugungswahrscheinlichkeiten der sHMM für den Merkmalsvektor $\vec{m}$ auf Werte größer 0 angehoben, indem der kleinste Score subtrahiert wird. Es werden analog zur Formel (5.3) nur sHMM betrachtet deren Ausgangsscore über einer gewissen Schranke τ liegt.

$$(\forall \lambda \in \Lambda_\tau) \quad \tilde{\lambda}_g(\vec{m}) = \left[\lambda_g(\vec{m}) - \min_{(\forall \lambda \in \Lambda_\tau)} \lambda_g(\vec{m})\right] \cdot P_K(g) \tag{5.9}$$

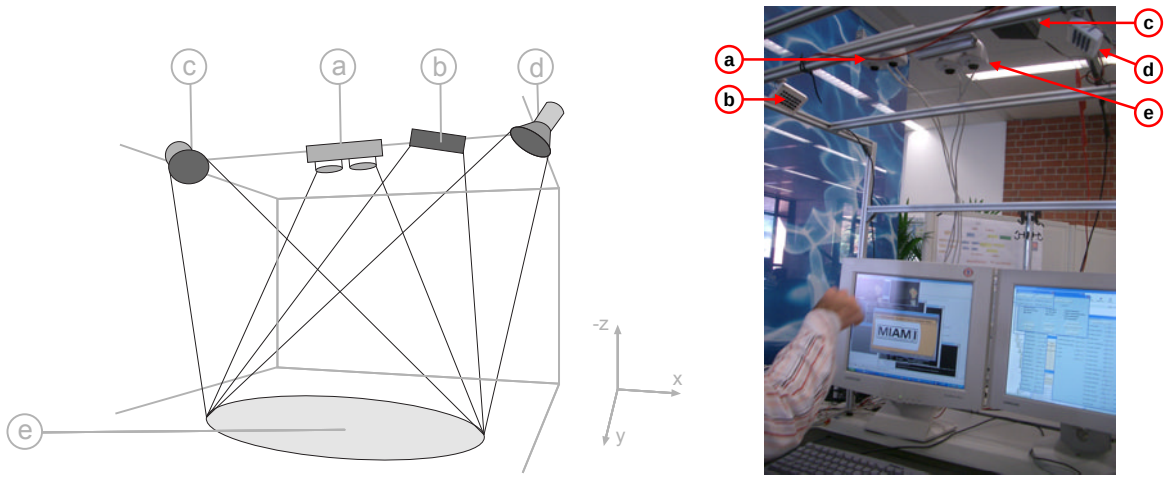
Die abschließende Erkennung erfolgt durch Maximabestimmung analog zur Formel (5.2).

5.5. Versuchsaufbau

Im Folgenden wird der Aufbau und die erforderliche Hardware für das Gesamtsystem im Fahrzeug sowie in der Desktopumgebung beschrieben. Distanzmaße finden sich im Anhang A.

5.5.1. Desktopumgebung

An einem experimentellen Versuchsaufbau können bereits im Vorfeld, bevor das endgültige Gesamtsystem im Fahrzeug realisiert wird, ohne großen Aufwand verschiedene Segmentierungsarten, Beleuchtungs- und Kamerasysteme evaluiert werden. Darüber hinaus ist ein grundlegender Vergleich von verschiedenen Klassifizierungs- und Spottingmethoden nur unter perfekten Umgebungsbedingungen sinnvoll, die im Fahrzeug im Allgemeinen nicht gegeben sind.



(*I*) (*II*)

Abbildung 5.14.: Schematischer Aufbau (I): Stereo Farb- und NIR-Kamera CREATIVE Webcam NX [Cre04] (a), 15W IR-Strahler (b), Conrad IR-Strahler [Con04] (c) und (d), Grundfläche des Gesteninteraktionsbereiches (e).
Realer Aufbau (II): Stereo Farbkamera CREATIVE Webcam NX (a), Conrad IR-Strahler (b) und (d), 15W IR-Strahler (c), Umgebaute Stereo NIR-Kamera CREATIVE Webcam NX (e).

Für die Experimente in der Desktopumgebung wurde ein Gestell konstruiert und angefertigt, das ungefähr den Höhenverhältnissen im Fahrzeug entspricht. Dort wurden zwei 3D Stereosysteme angebracht, die jeweils aus zwei CREATIVE Webcam NX USB Kameras aufgebaut sind. Eines dieser Systeme ist durch die Montage eines Tageslichtfilters in der Art modifiziert worden, dass Aufnahmen im IR Spektrum möglich sind. Das zweite Paar wird für farbbasierte und Stereo-Segmentierungstechniken verwendet.
Da die 15W IR-Lichtquelle, die sich direkt neben der Kamera befindet, bei senkrechter Strah-

lung auf die Bildebene einen starken Glanzpunkt auf dem Tisch hervorruft, wurde der Einstrahlwinkel verändert und somit die Szene schräg von der Seite beleuchtet. Zwei zusätzliche schwache IR-Strahler, die von beiden Seiten die Szene beleuchten, mindern den Effekt der ungleichmäßigen Ausleuchtung und der daraus resultierenden vermehrten Schattenbildung. Der gesamte Versuchsaufbau mit allen Komponenten ist in Abbildung 5.14 zu sehen.

5.5.2. Fahrzeug

Das Gesamtsystem für den Einsatz im Fahrzeug wurde so konzipiert, dass sich der Gesteninteraktionsbereich über der Mittelkonsole befindet und somit bequem von Fahrer und Beifahrer erreicht werden kann. Um eine gleichmäßige Beleuchtung zu gewährleisten, wurde über der Mittelkonsole in den Himmel des Fahrzeuges, ein IR-Strahler integriert, wie in Abbildung 5.15 zu sehen. Direkt daneben befindet sich das 3D IR-Stereokamerasystem. Aufgrund der großen Bauhöhe der IR-Beleuchtung konnte keine seitliche Beleuchtung realisiert werden.

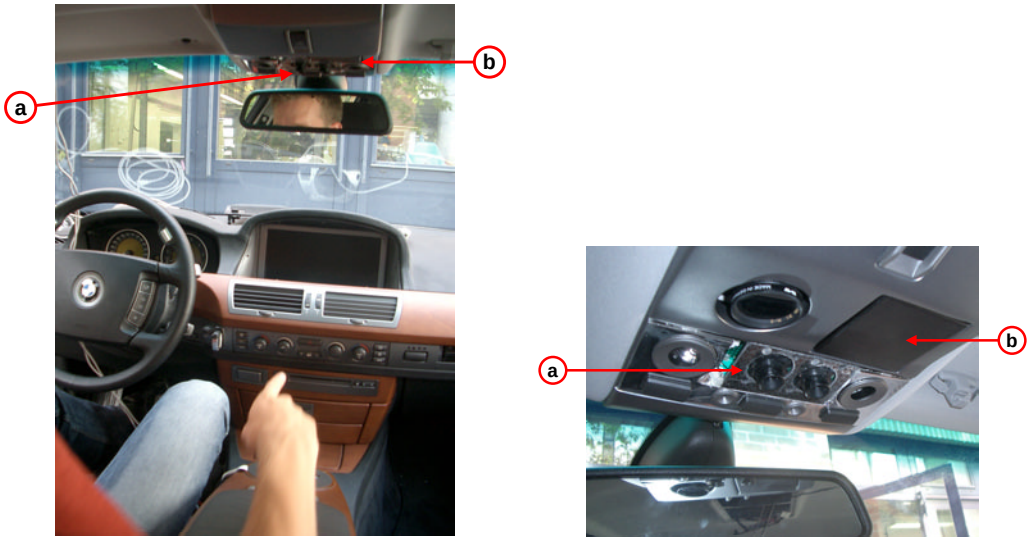


Abbildung 5.15.: (a) Stereo NIR-Kamera (CREATIVE Webcam NX). (b) 15W IR-Strahler.

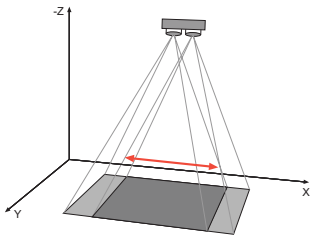
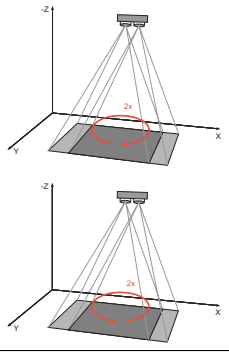
5.6. Gestenvokabular

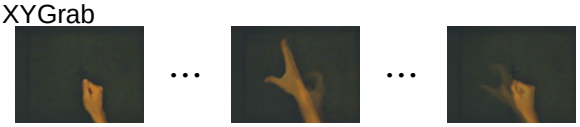
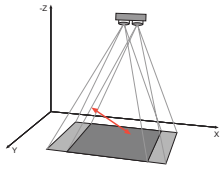
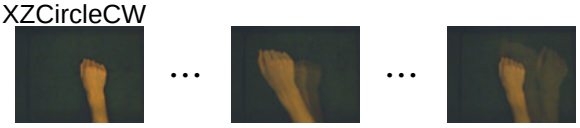
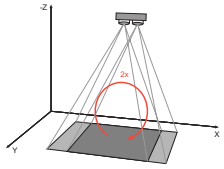
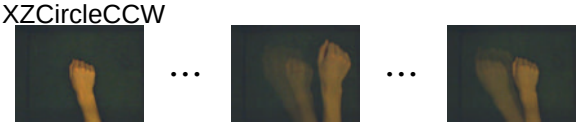
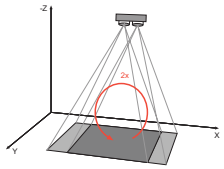
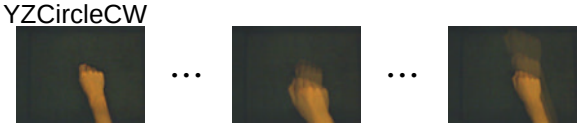
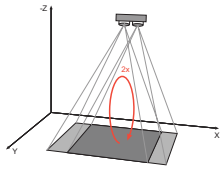
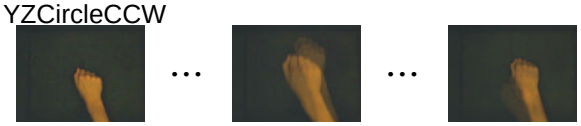
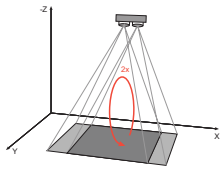
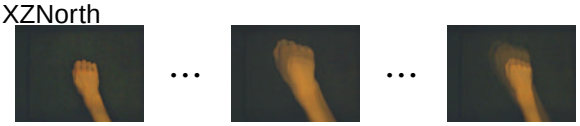
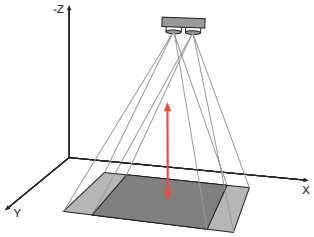
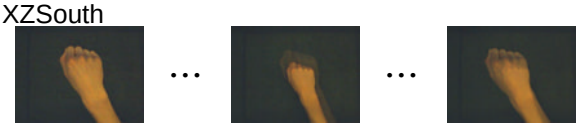
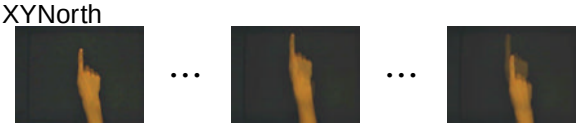
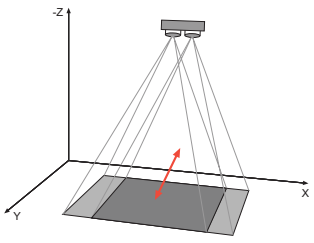
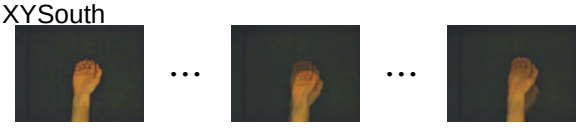
Inhalt dieses Abschnittes ist das Gestenvokabular des Erkennersystems, welches sich aus referentiellen, mimischen und kinemimischen Handgesten zusammensetzt (siehe auch Grundlagenkapitel 2). Zum einen enthält der Gestenschatz Instanzen wie beispielsweise Richtungs- und Zeigegesten, die für eine Mensch-Maschine-Interaktion im automotiven Umfeld sinnvoll erscheinen. Zum anderen wurde die Zusammenstellung bewusst gewählt, um das System vor

gewisse Herausforderungen bei der Erkennung zu stellen und dabei zu erproben. So beinhaltet das Gestenvokabular

- komplexe und in der Art der Ausführung stark benutzerabhängige Gesten wie die Greifgeste.
- Gesten mit Bewegungen zur Kamera wie z. B. ZWipe.
- Gesten mit ähnlichem Bewegungsverlauf wie beispielsweise XYWestPointerUp und XYWestWave.
- sowohl Gesten mit kleiner (z. B. XYNorth) als auch mit großer Bewegungsamplitude (z. B. XYCircleCW).

Tabelle 5.6 klärt die Nomenklatur des Gestenvokabulars und veranschaulicht anhand von Momentaufnahmen der Bewegung und Verlaufskizzen die Charakteristika sämtlicher Handgesten. In Tabelle 5.7 finden sich die detaillierten Ablaufbeschreibungen und die Einordnung in die Gestentaxonomie aus Abschnitt 2.1.3.

Gestenbezeichnung und Verlauf	Achsenbezug
XYEastWave XYWestWave XYEastPointerUp XYWestPointerUp	
XYCircleCW XYCircleCCW	

Gestenbezeichnung und Verlauf	Achsenbezug
XYGrab 	
XZCircleCW 	
XZCircleCCW 	
YZCircleCW 	
YZCircleCCW 	
XZNorth 	
XZSouth 	
XYNorth 	
XYSouth 	

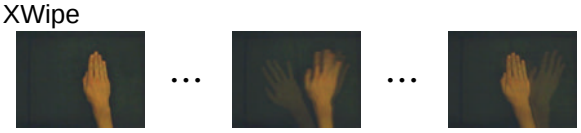
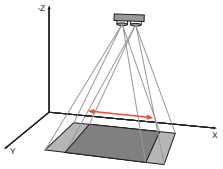
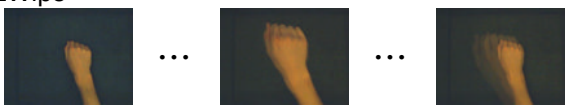
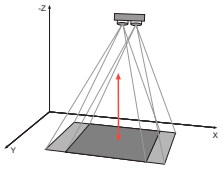
Gestenbezeichnung und Verlauf	Achsenbezug
XWipe 	
ZWipe 	

Tabelle 5.6.: Gestenvokabular (Momentaufnahmen mit Verlaufsandeutung und Achsenbezug).

Geste	Beschreibung	Taxonomie
XYEastPointerUp	Handform ist in Zeigegeste, schlägt kurz in Richtung der X-Achse aus und kehrt in die Ausgangslage zurück.	kinemimische Gesten
XYWestPointerUp	Handform ist in Zeigegeste, schlägt kurz entgegen der X-Achse aus und kehrt in die Ausgangslage zurück.	
XYEastWave	Handform ist in Winkgeste, schlägt kurz in Richtung der X-Achse aus und kehrt in die Ausgangslage zurück.	
XYWestWave	Handform ist in Winkgeste, schlägt kurz entgegen der X-Achse aus und kehrt in die Ausgangslage zurück.	
XZNorth	Hand zur Faust geformt bewegt sich zur Kamera (entgegen der Z-Achse) und kehrt zurück zur Ausgangsposition.	
XZSouth	Hand zur Faust geformt bewegt sich von Kamera weg (in Richtung der Z-Achse) und kehrt zurück zur Ausgangsposition.	
XYNorth	Handform ist in Zeigegeste, bewegt sich entgegen der Y-Achse und kehrt in die Ausgangslage zurück.	
XYCircleCW	Hand ist in Zeigegeste und beschreibt zwei Kreise im Uhrzeigersinn in der XY-Ebene.	referentielle Gesten
XYCircleCCW	Hand ist in Zeigegeste und beschreibt zwei Kreise gegen den Uhrzeigersinn in der XY-Ebene.	

Geste	Beschreibung	Taxonomie
XZCircleCW	Hand zur Faust geformt beschreibt zwei Kreise im Uhrzeigersinn in der XZ-Ebene.	
XZCircleCCW	Hand zur Faust geformt beschreibt zwei Kreise gegen den Uhrzeigersinn in der XZ-Ebene.	
YZCircleCW	Hand zur Faust geformt beschreibt zwei Kreise im Uhrzeigersinn in der YZ-Ebene.	
YZCircleCCW	Hand zur Faust geformt beschreibt zwei Kreise gegen den Uhrzeigersinn in der YZ-Ebene.	
XWipe	Offene Hand bewegt sich mehrmals auf der X-Achse hin und her.	
ZWipe	Hand zur Faust geformt bewegt sich mehrmals auf der Z-Achse hin und her.	
XYGrab	Hand greift nach einem virtuellen Gegenstand. Die Bewegung erfolgt diagonal in der XY-Ebene.	mimische Gesten
XYSouth	Handform ist in Ziehgeste, bewegt sich in Richtung der Y-Achse und kehrt in die Ausgangslage zurück.	

Tabelle 5.7.: Gestenvokabular (Ablaufbeschreibung und Taxonomie).

5.7. MIAMII Demonstrator

In dieser Diplomarbeit wurde konsequent eine Anbindung an eine reale MMI-Demonstrationsplattform verfolgt, um eine gestenbasierte Interaktion im Rahmen automotiver Infotainmentsysteme realisieren zu können. Für diese Aufgabe wurde das MIAMII-Framework ausgewählt, das in diesem Abschnitt behandelt werden soll.

MIAMII (Multimodal Individual AutoMotive Infotainment Interface) ist ein BMW-internes MMI-Projekt. Das MIAMII System baut grundsätzlich auf Vorarbeiten und Ergebnissen aus dem universitären Kooperationsprojekt FERMUS [LMA+01] auf. Wesentliche Grundlage des überwiegend in TCL/TK [Act04] implementierten Systems ist eine TCP/IP Client-Server-Architektur, die eine Anbindung neuer modularer Komponenten erlaubt. Das *Integrator-Modul* agiert hierbei als zentraler Server, der eingehende Nachrichten zu abstrakten Steuermeldungen verarbeitet, die anschließend an alle Clients propagiert werden. In dieser globalen Steuereinheit ist die komplette Logik der Menüstruktur sowie der Modalitätenintegration bzw. -fusion implementiert.

(a) Hauptauswahl. (b) Unterebene Musik.

Abbildung 5.16.: MIAMII Demonstrator.

Im Rahmen dieser Arbeit wurde der *Integrator* um die Möglichkeit zur Erzeugung von Systemkontextwissen erweitert. Abhängig vom aktuellen Menüzustand werden Steuerinformationen an den angeschlossenen Kontextserver geschickt, der diese Informationen zu Kontextwahrscheinlichkeiten konvertiert (siehe Abschnitt 5.4). Diese Sequenz an Wahrscheinlichkeiten wird an das Klassifikator- bzw. Spottingmodul zur Kontextintegration mittels TCP/IP Nachrichten verschickt. Nachdem der Spotter das Ende einer Geste erkannt hat, wird der Klassifikator aufgerufen und dessen Ergebnis als Steuernachricht direkt über die Kommunikationskomponente an den *Integrator* gesendet, der die Abbildung von Geste zu Interaktion durchführt und gegebenenfalls den Menüzustand wechselt.

KAPITEL 6

EVALUIERUNG

Dieses Kapitel beinhaltet zahlengestützte Ergebnisse hinsichtlich der Erkennungsleistung der Klassifikatoren und des Gesamtsystems. Zunächst wird auf die Erzeugung der Testdaten eingegangen und auf die verwendeten Vergleichskriterien. Mit Hilfe dieser wird in einem ersten Schritt die bestmögliche Parameterkonfiguration der Klassifikatoren ermittelt. Erst dann werden die Erkenner sowohl in der Labor- als auch in der Fahrzeugumgebung gegenübergestellt und auf ihre Leistungsfähigkeit hin verglichen. Desweiteren wird in einem praxisrelevanten Szenario untersucht inwieweit sich die Integration von Kontextwissen auf die Erkennungsleistung auswirkt. Im letzten Schritt wird die Kombination von Spotting [Lin04] und isolierter Klassifizierung im Gesamtsystem evaluiert.

6.1. Datenerfassung und Evaluierungskriterien

Datenerfassung

Zum Training des wahrscheinlichkeitsbasierten Klassifikators und zur Ermittlung der Erkennungsleistung wurden drei Gestenkorpora erstellt. Die zugrundeliegenden Videoaufnahmen unterscheiden sich hinsichtlich Gestenumfang, Anzahl beteiligter Probanden und der Umgebung (siehe Tabelle 6.1).

Die Versuchspersonen mussten die Gesten in einer fest vorgegebenen Abfolge tätigen. Um eine zu „mechanische“ Ausführung der Gesten zu vermeiden, wurde die Reihenfolge so gewählt, dass gleiche Gestentypen nicht direkt hintereinander folgen.
Nach der Aufnahme wurden sämtliche Videos manuell mit Zeitstempeln versehen (*tagging*), in denen Startzeit, Endzeit und der jeweilige Gestentyp festgehalten ist. Dadurch werden Fehlerquellen, wie sie durch ein möglicherweise ungenaues maschinelles Spotting entstehen,

ausgeschlossen. Somit kann unabhängig von der zeitlichen Segmentierung allein die isolierte Erkennung der Gesten bewertet werden.

	Zusammensetzung	Umgebung
Gestenkorpus 1	17 Gestentypen von einer Person mit jeweils 15 Wiederholungen (insgesamt 255 Gesten)	Desktopumgebung mit statischer Schwellwertsegmentierung
Gestenkorpus 2	17 Gestentypen von 6 Probanden mit je 15 Wiederholungen (insgesamt 1530 Gesten)	Desktopumgebung mit statischer Schwellwertsegmentierung
Gestenkorpus 3	17 Gestentypen von 2 Probanden mit je 5 Wiederholugen (insgesamt 170 Gesten)	Fahrzeugumgebung mit histogrammbasierter Schwellwertsegmentierung (Infrarotaufnahme)

Tabelle 6.1.: Verschiedene Gestenkorpora.

Evaluierungskriterien

Folgende Kriterien zur Evaluierung liefern Aussagen über die Leistungsfähigkeit eines Verfahrens. Der Wertebereich der Evaluierungskriterien Erkennungsrate, Falschakzeptanzrate und Erkennungssicherheit liegt im Intervall $[0,1]$. Für die ersten beiden Kriterien werden in diesem Kapitel die Angaben oft auch in % angegeben, wobei die Werte dann entsprechend auf das Intervall $[0\%,100\%]$ abgebildet werden. Das Maß „Fehler pro Minute“ nimmt ganzzahlige Werte zwischen 0 und ∞ an.

Erkennungsrate (R_e): Als wichtigstes Leistungskriterium bezeichnet die Erkennungsrate das Verhältnis zwischen der Anzahl korrekt erkannter Gesten G^k innerhalb einer Testmenge G^{Test}:

$$R_e = \frac{|G^k|}{|G^{Test}|} \tag{6.1}$$

Falschakzeptanzrate (R_f): Die Falschakzeptanzrate, oder auch Fehler 2. Art, ist ein Maß für falsch positive Klassifizierungen. Das heißt Bewegungen, die in Bezug auf das Gestenvokabular keine Geste darstellen, werden fälschlicherweise vom Klassifikator als Instanz des Gestenvokabulars erkannt. Berechnet wird die Falschakzeptanzrate aus dem Verhältnis von der Anzahl der falsch positiven Klassifizierungen B^f zu der Anzahl der Bewegungen B, die in Bezug auf das Gestenvokabular keine Geste darstellen:

$$R_f = \frac{|B^f|}{|B|} \tag{6.2}$$

Erkennungssicherheit (R_s): Allein anhand der Erkennungsrate unterscheidet sich die Leistung der probabilistischen Klassifizierung bei verschiedenen Merkmalsvektorkompositionen gelegentlich nur marginal. Um dennoch einen Vergleich ziehen zu können, gibt bei gleichwertigen Erkennungsraten die Erkennungssicherheit ein Maß zur Güte der Klassifizierung. Nach [Mor00] wird in dieser Arbeit davon ausgegangen, dass eine wahrscheinlichkeitsbasierte Erkennung umso sicherer ist, je signifikanter der durchschnittliche Abstand der besten Erzeugungswahrscheinlichkeit $\lambda_g(\vec{m})$ von der zweitbesten Erzeugungswahrscheinlichkeit $\lambda_2(\vec{m})$ ist:

$$R_s = \frac{1}{\left|G^k\right|} \sum_{\forall g \in G^k} \exp\left[\frac{-10}{\lambda_g(\vec{m}) - \lambda_2(\vec{m})}\right] \tag{6.3}$$

Fehler pro Minute ($\tilde{R}_f$): Werden vom Spotter die zeitlichen Grenzen einer Geste falsch gesetzt, wird der Fehler pro Minute inkrementiert. Der Fehler pro Minute bezeichnet somit die Falschakzeptanzrate der zeitlichen Segmentierung (siehe auch [Lin04]). Der nachgeschaltete Klassifikator kann $\tilde{R}_f$ weiter verbessern, sofern er die fälschlich gespottete Bewegung als „nicht erkannt“ ablehnt.

6.2. Parametrisierung

Es hat sich gezeigt, dass die Wahl der klassifikatorinternen Parameter die Leistung des Systems signifikant beeinflussen. Dabei sind je nach verwendeter Parameterkonfiguration Schwankungen in den Erkennungsraten von bis zu 50% zu beobachten. Ziel dieses Abschnittes ist es, eine möglichst optimale Voreinstellung jeweils für den 3-Phasen- und den sHMM-basierten Klassifikator zu finden.

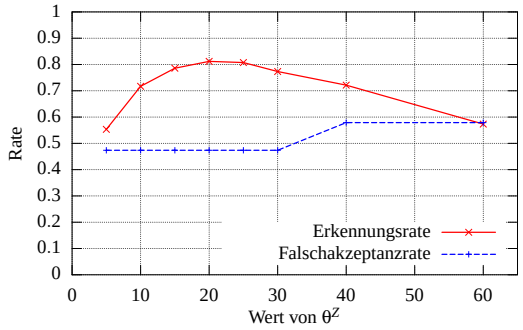
6.2.1. 3-Phasen-Klassifikator

Der regelbasierte Klassifikator wird über 16 verschiedene Parameter konfiguriert. Von diesen haben insbesondere die sechs Schwellwerte θ^X, θ^Y, θ^Z, θ^{XY}, θ_d^X und θ_d^Y einen unmittelbaren Einfluss auf die Erkennungsleistung. Die restlichen Parameter, wie beispielsweise die Fenstergröße des Medianfilters oder der Skalierungsfaktor der Flächentrajektorie, dienen im wesentlichen der Datenvorverarbeitung. Sie wurden empirsch ermittelt und beeinflussen die Erkennung lediglich indirekt (siehe auch Anhang A.1 für eine vollständige Auflistung und Beschreibung aller Parameter).

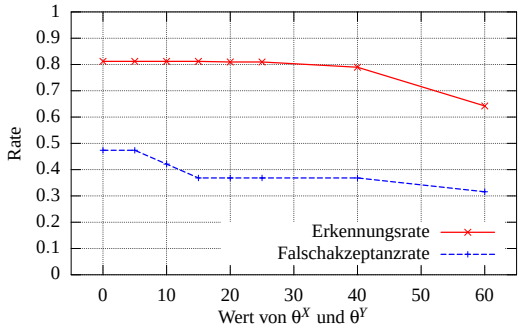
Eine Parametrisierung aller 16 Einflussgrößen ist ein kompliziertes und aufwendiges Optimierungsproblem weshalb die zu betrachtende Auswahl auf die wichtigsten Parameter eingeschränkt wird. Eine erschöpfende Analyse der oben genannten sechs Schwellwerte würde gleichwohl an der nötigen Rechenzeit scheitern. Um dennoch eine bestmögliche Konfiguration abschätzen zu können, wird von einer idealen Parameterunabhängigkeit ausgegangen und

eine iterative Einschätzung vorgenommen. Grundlage für die folgenden Erläuterungen ist der Entscheidungsgraph in Abbildung 5.3.1 auf Seite 63.

Die absoluten Schwellwerte θ^X, θ^Y, θ^Z regeln die Schranke für den minimal zulässigen Amplitudenunterschied der jeweiligen Trajektorie. Änderungen von θ^Z wirken sich besonders auf Gesten mit dominanter Z-Trajektorie aus. Ein zu großer Wert (≥ 60) verhindert die Erkennung von Gesten in Z-Richtung, erleichtert jedoch die Klassifizierung der übrigen Gesten. Ein zu kleiner Wert (≤ 10) erlaubt zwar eine bessere Erkennung von Gesten in Z-Richtung, erschwert aber wesentlich eine Erkennung von Gesten mit dominanter X- oder Y-Trajektorie, da diese bei marginalen Schwankungen der Z-Achse aus ihrer eigentlichen Klasse fallen. Ein Kompromiss zwischen diesen beiden Extremwerten liegt bei einem Wert von ca. 20 (siehe Abbildung 6.1(a)).



(a) Parametrisierung des Schwellwertes θ^Z ($\theta^X = 5,\ \theta^Y = 5$)



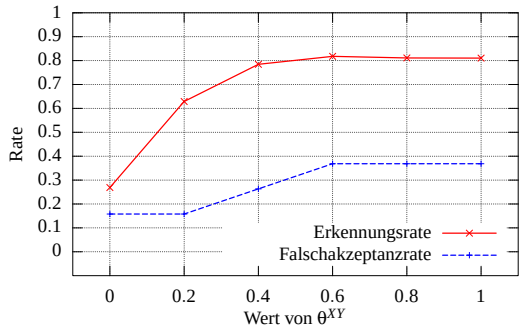
(b) Parametrisierung der Schwellwerte θ^X und θ^Y ($\theta^Z = 20$)

Abbildung 6.1.: Parametrisierung der absoluten Schwellwerte θ^X, θ^Y und θ^Z

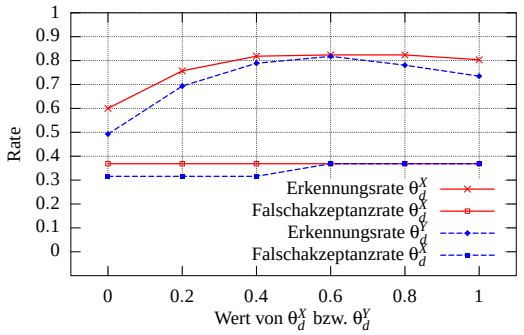
Die Parameter θ^X und θ^Y werden wegen Symmetrieeigenschaften kollektiv betrachtet. Im Gegensatz zu θ^Z bricht die Erkennungsrate beim Minimalwert nicht ein, da aufgrund des Entscheidungsgraphes in Abbildung 5.3.1 Z-dominante Gesten von θ^X und θ^Y nicht beeinflusst werden. Die Falschakzeptanzrate steigt jedoch signifikant an, da die Einschränkungen für Garbage-Gesten entsprechend gelockert werden. Erhöht man die beiden Schwellwerte sinken die Erkennungsraten erwartungsgemäß, wobei ausschließlich Gesten in der XY-Ebene betroffen sind. Gleichzeitig lässt sich ein Absinken der Falschakzeptanzrate beobachten, da die Art möglicher Bewegungen eingeschränkt wird. Eine Konfiguration von $\theta^X = 15$ und $\theta^Y = 15$ bietet eine geringe Falschakzeptanzrate bei akzeptablen Erkennungsraten (siehe auch Abbildung 6.1(b)).

Die relativen Schwellwerte θ^Z_{XY}, θ^{XY}, θ^X_d und θ^Y_d entscheiden anhand der Verhältnisse zwischen den Trajektorien die Klassenzugehörigkeit. Die Einflussgröße θ^{XY} regelt das maximal zulässige Amplitudenverhältnis zwischen X- und Y-Trajektorie bei einer alleinigen X- bzw. Y-Dominanz (siehe auch Abbildung 6.2(a)). Geht der Wert des Parameters gegen 0, wird keinerlei Abweichung der anderen Trajektorie toleriert. Da die Gesten immer einen entsprechenden Fremdtrajektorienanteil aufweisen, werden Gesten die eine X- oder Y-Trajektorie

als dominantes Merkmal enthalten nicht mehr erkannt. Die durchschnittliche Erkennungsrate sinkt somit gleichzeitig mit der Falschakzeptanzrate ab. Bei einer Lockerung von θ^{XY} steigt die Erkennungsrate bis zu einem Wert von 0.4 kontinuierlich an. Einen ähnlichen Kurvenverlauf beschreibt die Falschakzeptanzrate, nur dass die Steigungsphase erst mit einem Wert von 0.6 endet. Ein Wert von 0.4 bildet einen Kompromiss zwischen optimaler Falschakzeptanzrate und möglichst hoher Erkennungsrate.



(a) Parametrisierung des Schwellwertes θ^{XY} ($\theta^X = 15,\ \theta^Y = 15,\ \theta^Z = 20$)



(b) Parametrisierung der Schwellwerte θ_d^X und θ_d^Y ($\theta^X = 15,\ \theta^Y = 15,\ \theta^Z = 20,\ \theta^{XY} = 0.9$)

Abbildung 6.2.: Parametrisierung der relativen Schwellwerte θ^{XY}, θ_d^X und θ_d^Y

Die relativen Einflussgrößen θ_d^X und θ_d^Y entscheiden bei dominanter Y bzw. X-Trajektorie, ob nachträglich beide Trajektorien dominant werden. Abhängig von diesen Parametern sind somit sämtliche Gesten im reduzierten Katalog XYPlane (XYCircleCW, XYCircleCCW und XYGrab). Werden die Werte der Parameter zu gering gewählt, bricht die Erkennungsrate für Gesten mit nur einer dominanten X- oder Y-Trajektorie ein, da wegen dem geringen Fehleranteil der nichtdominanten Trajektorie immer beide Trajektorien dominant werden. Zu hohe Werte verhindern dafür die Klassifizierung von Gesten in der XY-Ebene. Ein signifikanter Abfall in der Erkennungsrate ist nur bei θ_d^Y zu beobachten, da sämtliche Kreisgesten in der XY-Ebene meist eine größere X als Y-Amplitude aufweisen und somit in den meisten Fällen X die dominante Trajektorie ist. Da die Falschakzeptanzrate weitgehend konstant bleibt, ergeben die Werte $\theta_d^Y = 0.6$ und $\theta_d^X = 0.8$ eine optimale Konfiguration hinsichtlich der Erkennungsrate (siehe Abbildung 6.2(b)).

6.2.2. sHMM-Klassifikator

Um den Aufwand der Trainingsphase zu begrenzen, wird auf Kosten der Genauigkeit keine aufwendige *Kreuzvalidierung* (siehe auch Abschnitt 6.3.1) durchgeführt und die verwendeten Trainings- und Testdaten von nur einem Probanden aus Gestenkorpus 1 (17 Gestentypen von einer Person mit jeweils 15 Wiederholungen (insgesamt 255 Gesten)) verwendet. Somit spiegeln die Erkennungsraten in diesem Abschnitt keine repräsentative benutzerunabhängige Klassifikatorleistung wider. Sie geben lediglich eine qualitative und vergleichende Auskunft

über das Potential verschiedener Parameterkonfigurationen für den sHMM-basierten Klassifikator.

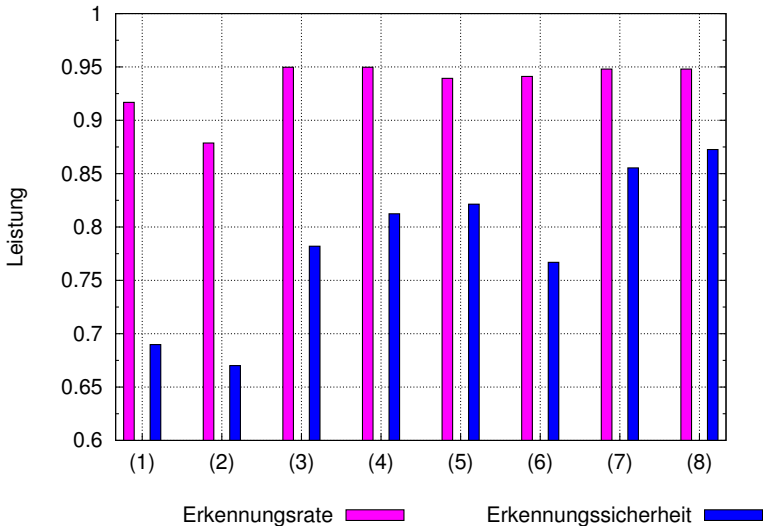
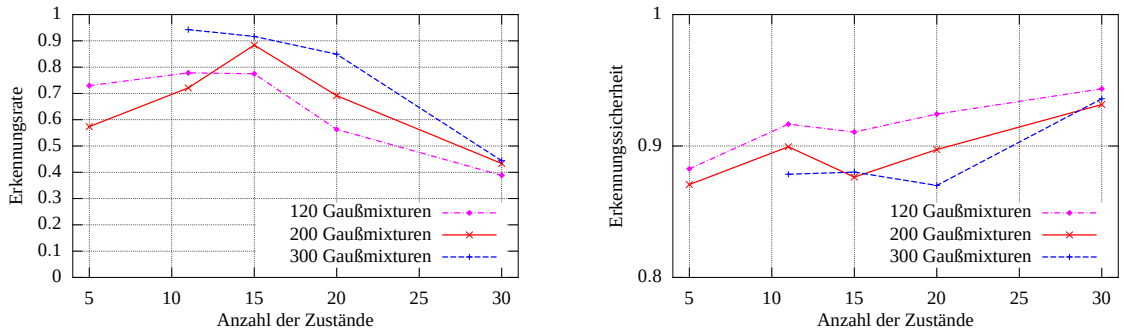


Abbildung 6.3.: Vergleich verschiedener Merkmalsvektorkompositionen bei sHMM-basierter Erkennung. (1) $x'y'$, (2) $x'y'z'$, (3) $x'y'A'$, (4) $x'y'x''y''$, (5) $x'y'z'A'x''y''z''A''$, (6) $x'y'z'A'$, (7) $x'y'z'A'H_1H_2H_3H_4$, (8) $x'y'z'A'H_1H_2H_1'H_2'$

Abbildung 6.3 zeigt einen Vergleich der Erkennungsraten und Erkennungssicherheiten der sHMM mit verschiedenen Merkmalsvektorkompositionen (MVK) im Desktopbereich. Offensichtlich genügt allein der Verlauf der x- und y-Komponente des Schwerpunktes (MVK (1)), um Erkennungsraten über 0.9 zu erzielen. Selbst Gesten in Z-Richtung verfügen somit über einen markanten charakteristischen Verlauf in der XY-Ebene. Wird zusätzlich die errechnete Tiefeninformation Z hinzugenommen (MVK (2)) sind die Erkennungsraten sogar rückläufig. Zurückführen lässt sich das auf das nichtdeterministische Fehlverhalten der Z-Komponente (siehe auch letzten Absatz von Abschnitt 5.2.2). Eine weitere Steigerung sowohl der Erkennungsrate als auch der Erkennungssicherheit ergibt sich durch die Hinzunahme der Flächenverhältnisse A' (MVK (3)) bzw. der 2. Ableitungen (MVK (3)). Die beste Modellierung lässt sich durch die Hinzunahme von Formmerkmalen erzielen, die in den Hu-Momenten kodiert sind (MVK (8) und (9)). MVK (9), der zusätzlich eine zeitliche Veränderung der ersten beiden Hu-Momente enthält, modelliert das Gestenvokabular am Besten und wird deshalb im folgenden für Klassifizierungen im Desktopbereich verwendet. Im Fahrzeugbereich wurde der MVK (1) verwendet, um negative Einflüsse falscher Tiefeninformation zu minimieren (siehe auch Abschnitt 6.3.2).

Abbildung 6.4 veranschaulicht den Verlauf der Erkennungsraten bzw. -sicherheiten abhängig von den global verwendeten Zustands- und Gaußmixturanzahlen. Diese Parameter besitzen einen wesentlichen Einfluss auf die Güte der sHMM. Grundsätzlich erlaubt eine höhere Anzahl von Gaußmixturen eine bessere Datenmodellierung, was sich auch in der Erkennungsrate



(a) Verhältnis von Erkennungsrate und Anzahl der Zustände

(b) Verhältnis von Erkennungssicherheit und Anzahl der Zustände

Abbildung 6.4.: Erkennungsraten der sHMM bei einer variablen Anzahl von Gaußmixturen und Zuständen

entsprechend niederschlägt. Gleichzeitig steigt jedoch auch der nötige Rechen- und Speicheraufwand, so dass eine Gaußmixturenzahl von 250 ungefähr die Grenze zur Echtzeitfähigkeit markiert.
Unabhängig von den Gaußmixturen zeigt sich bei einer globalen Zustandsanzahl von 15 ein deutliches Maxima in der Erkennungsrate. Dieser Wert korreliert mit der durchschnittlichen Ausführungslänge sämtlicher Gesten im verwendeten Gestenkorpus. Werden die Zustände entsprechend zu klein gewählt, sprechen insbesondere sHMM längerer Gesten wie XWipe auf kürzere inkludierte Gesten wie XYEast oder XYWest mit einer hohen Ausgangswahrscheinlichkeit an. Bei einer zu großen Zustandsanzahl erreichen die Merkmalsvektorsequenzen kürzerer Gesten nicht die Endzustände der sHMM und können somit nicht mehr erkannt werden.

Aufgrunddessen wurden die Zustandslängen aller sHMM auf die durchschnittliche Merkmalsanzahl $\overline{T}$ ihrer korrespondierenden Geste adaptiert (siehe auch Tabelle 5.4). Ergebnis ist ein signifikant schnellerer Anstieg der Erkennungsrate (siehe Abbildung 6.5) im Vergleich zu einer statischen Zustandsanzahl von 15. Die Sättigung tritt schon bei einer Gaußmixturenanzahl von 160 auf und liegt durchgehend über dem Niveau von statischen Zustandsanzahlen aus Abbildung 6.4(a).

6.3. Erkennungsleistung der Klassifizierung

Folgender Abschnitt evaluiert die im realen System eingesetzten Klassifikatoren basierend auf den Gestenkorpora aus dem Abschnitt 6.1 mittels der Evaluierungskriterien Erkennungsrate, Falschakzeptanzrate und Fehler pro Minute. Beide vorgestellten Verfahren werden mit dem optimierten Parametersatz konfiguriert (siehe auch Abschnitt 6.2).

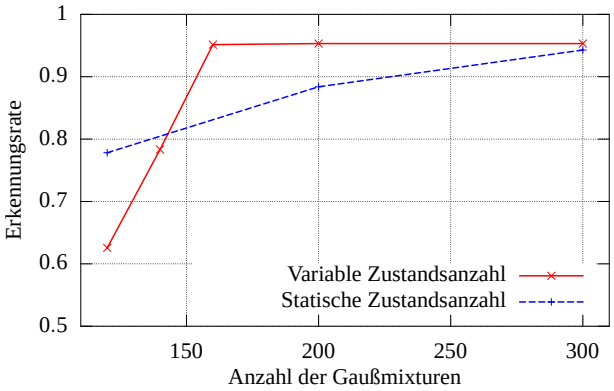


Abbildung 6.5.: Erkennungsraten der sHMM bei einer variablen Anzahl von Gaußmixturen und individueller Zustandsadaption.

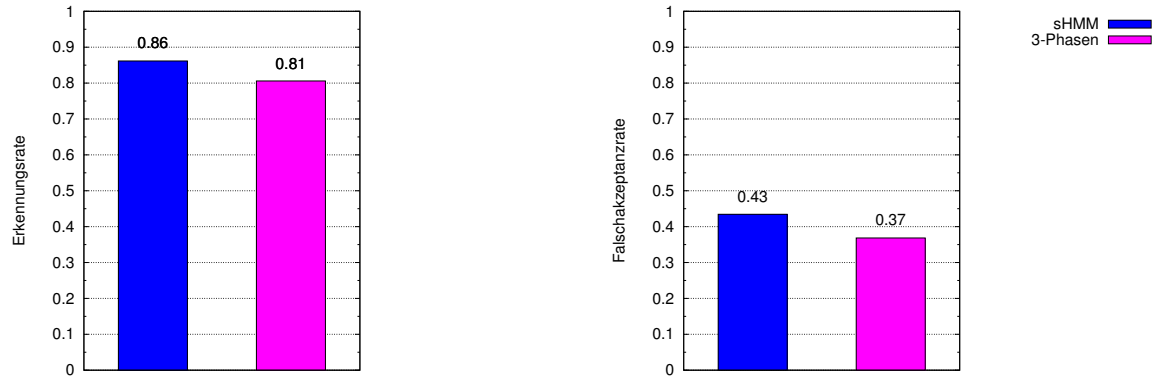
6.3.1. Desktopumgebung

Als Grundlage für den Vergleich der beiden Klassifikatoren in der Desktopumgebung diente der Gestenkorpus 2 (6 Probanden mit 17 unterschiedlichen Gestentypen mit je 15 Wiederholungen). Die Auswertungen erfolgten dabei benutzerunabhängig. Dazu stammt für die durchgeführte 6-fache *Kreuzvalidierung*[1] (siehe auch Anhang C.2) des sHMM-basierten Klassifikators das Gestenmaterial der Trainingsdaten von unterschiedlichen Probanden wie das der Testdaten. Für den regelbasierten Ansatz mussten keine speziellen Vorkehrungen getroffen werden um die Nutzerunabhängigkeit bei der Evaluierung zu erzielen. Lediglich die Parameteroptimierung wurde auf gesondertem Datenmaterial (Gestenkorpus 1) vorgenommen.

Im direkten Vergleich erzielen die beiden Klassifikatoren ähnlich gute Erkennungsraten von über 80%, wobei der wahrscheinlichkeitsbasierte Ansatz mit 86% geringfügig besser abschneidet als der regelbasierte Ansatz mit 81% (siehe Abbildung 6.6(a)). Bei den Falschakzeptanzraten hingegen liegt der regelbasierte Ansatz leicht vorne (siehe Abbildung 6.6(b)). Eine genaue Leistungsanalyse ermöglicht erst die Aufschlüsselung der Erkennungsrate in die Einzelraten für sämtliche Gesten (siehe Abbildung 6.7).

Starke Leistungseinbußen sind für den *3-Phasen-Klassifikator* bei den Gesten XWipe und XY-Grab zu erkennen. Da die Wischgeste XWipe oftmals sehr schnell ausgeführt wird, verursacht die immanente Synchronisationsproblematik bei den Bewegungsphasen der Gestik einen starken zeitlichen Versatz der Stereobilder und somit eine falsche Tiefeninformation (siehe auch Abschnitt 5.2.3). Ändert sich zugleich, durch unterschiedliche Rekursionstiefen der Armfilterung (siehe auch Abschnitt 3.1.2), sprunghaft die Fläche der Handregion (siehe auch Abbildung C.4 auf Seite 106), wird fälschlicherweise eine Bewegung in Z-Richtung registriert.

[1] Die n-fache Kreuzvalidierung ist ein Schätzverfahren zur Approximation der Erkennungsleistung eines Klassifikators. Dazu werden sämtliche Daten in n Partitionen unterteilt. Eine Partition wird zum Testen und die restlichen $n-1$ Partitionen zum Training des Klassifikators verwendet. Wird dieser Vorgang n-mal mit jeweils anderen Trainings- und Testpartitionen wiederholt, ergibt die Mittelung der Raten eine gute Abschätzung der Erkennungsrate [WF01].



(a) Erkennungsraten des 3-Phasen- und des sHMM-Klassifikators in der Desktopumgebung

(b) Falschakzeptanzraten des 3-Phasen- und des sHMM-Klassifikators in der Desktopumgebung

Abbildung 6.6.: Vergleich der Fehler- und Erkennungsraten von 3-Phasen und sHMM-basierten Algorithmus in der Desktopumgebung

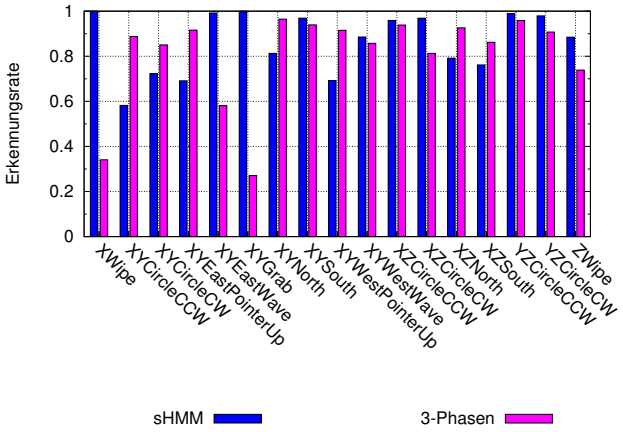


Abbildung 6.7.: Detaillierter Vergleich der Erkennungsraten von 3-Phasen und sHMM-basierten Algorithmus in der Desktopumgebung.

Geste	3-Phasen	sHMM
XWipe	XZPlane	–
XYCircleCCW	Garbage	XYCircleCW
XYCircleCW	Garbage	ZWipe
XYEastPointerUp	XZPlane	XYEastWave
XYEastWave	XZPlane	–
XYGrab	XYPlane	–
XYNorth	XYSouth	XYSouth
XYSouth	XYNorth	XYGrab
XYWestPointerUp	XZPlane	XYWestWave
XYWestWave	XZPlane	XYEastWave
XZCircleCCW	XZPlane	XZCircleCW
XZCircleCW	XZPlane	XZCircleCCW
XZNorth	XZPlane	ZWipe
XZSouth	XZPlane	ZWipe
YZCircleCCW	YZPlane	–
YZCircleCW	YZPlane	–
ZWipe	YZCircleCCW	YZCircleCW

Tabelle 6.2.: Häufigste Fehlklassifikationen in der Desktopumgebung.

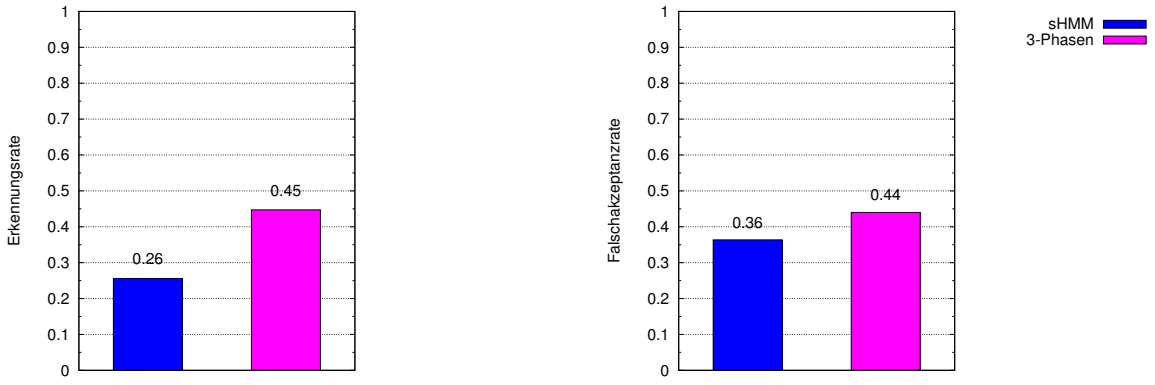
Häufigste Fehlerkennungen (siehe auch Tabelle 6.2) für die XWipe-Geste sind XZCircleCW bzw. XZCircleCCW im reduzierten Katalog XZPlane, verursacht durch den inkorrekten Verlauf der Z-Trajektorie. Ein leichter Einbruch der Erkennungsrate ist bei der XYEastWave-Geste zu beobachten, da diese von vielen Probanden schnell aus dem Handgelenk ausgeführt wird, was wiederum einen Einfluss der Z-Trajektorie verursacht. Für die komplexe Greifgeste XYGrab wird zwar im Regelfall der richtige reduzierte Katalog XYPlane erkannt. In der 3. Hierarchiestufe scheitert eine richtige Erkennung jedoch häufig an der falschen Abfolge der statischen Teilgesten. Ursache ist die relativ triviale Suche nach der statischen Handform in der Mitte der dynamischen Geste, welche je nach benutzerspezifischer Ausführung nicht zwingend die Hand in Greifform darstellt.
Verfälschte Tiefeninformationen bereiten dem *sHMM-basierten Klassifikator* keine Schwierigkeiten, da die Fehler beim Training mitmodelliert wurden. Probleme treten erst auf, sobald sich der Einfluss des Fehlers stark verändert, wie beispielsweise durch eine modifizierte Bildwiederholrate. Schlechtere Erkennungsraten treten vor allem bei den Kreisgesten XYCircleCW und XYCircleCCW und bei den Richtungsgesten XYEastPointerUp und XYWestPointerUp auf. Insbesondere Winkgesten und Kreisgesten in der XY-Ebene werden von den Probanden mit hoher personenspezifischer Varianz ausgeführt. Zusätzlich weisen Zeige- und Winkgesten sehr ähnliche Trajektorienverläufe auf, was Verwechslungen begünstigt (siehe auch Tabelle 6.2).

Um das System auf Spiegelungsinvarianz zu überprüfen, wurden 170 Gesten von einem Linkshänder klassifiziert. Erwartungsgemäß konnte keine signifikante Änderung in den Erkennungsraten festgestellt werden, da alle Gesten mit Ausnahme der Greifgeste spiegelungsinvariant sind.
Eine Evaluierung des 3-Phasen-Klassifikators ohne Fusion der Flächen- und Z-Komponente – die Tiefeninformation wurde lediglich aus der Flächenänderung bezogen – hatte einen Einbruch der Erkennungsrate auf 68% zur Folge. Die Falschakzeptanzrate stieg von 37% auf 58% an. Die zweite Kamera ist also gerechtfertigt, sofern Gesten mit dominanter Z-Achse detektiert werden sollen.

6.3.2. Fahrzeugdomäne

Da der Aufbau im Fahrzeug erst gegen Ende der Diplomarbeit fertiggestellt wurde, konnte im Vergleich zum Desktopbereich nur ein relativ begrenztes Set an Gesten gewonnen werden (Gestenkorpus 3: 2 Probanden mit insgesamt 85 Gesten). Um dennoch über eine ausreichende Menge an Trainingsmaterial verfügen zu können, wurden für den wahrscheinlichkeitsbasierten Ansatz deshalb die gleichen Trainingsdaten (Gestenkorpus 2) wie für den Desktopbereich verwendet. Diese suboptimale Ausgangsbedingung schlägt sich entsprechend signifikant in der Erkennungsrate des *sHMM-Klassifikators* nieder. Würden die meist reproduzierbaren Segmentierungsfehler, – analog zur Desktopumgebung – durch das Training modelliert werden, könnte eine höhere Erkennugsleistung erzielt werden. Die Erkennungsraten des 3-Phasen-Klassifikators sind mit 45% nahezu doppelt so hoch wie die des sHMM-Klassifikators aber

auf deutlich niedrigerem Niveau als in der Desktopumgebung (siehe Abbildung 6.8(a)). Die weiteren Untersuchungen beschränken sich somit auf den *3-Phasen-Klassifikator*.



(a) Erkennungsraten des 3-Phasen- und des sHMM-basierten Klassifikators in der Fahrzeugdomäne

(b) Falschakzeptanzraten des 3-Phasen- und des sHMM-basierten Klassifikators in der Fahrzeugdomäne

Abbildung 6.8.: Vergleich der Fehler- und Erkennungsraten von 3-Phasen und sHMM-basierten Algorithmus in der Fahrzeugdomäne.

Laut der detaillierten Erkennungsansicht in Abbildung 6.9 erreichen insbesondere die Kreisgesten nur sehr niedrige Raten. Das liegt zum einen daran, dass weitläufige Bewegungen oftmals den Gesteninteraktionsbereich kurzzeitig verlassen, was einen kompletten Verlust oder zumindest einen falschen Flächenanteil der Handkomponente zur Folge hat. Zum anderen ist die Ausleuchtung der Szenerie durch die eingeschränkte Lichtquellenpositionierung im Fahrzeug (siehe auch Versuchsaufbau im Abschnitt 5.5.2) inhomogen. Vor allem an der Peripherie des Gesteninteraktionsbereichs nimmt infolgedessen die Segmentierungsqualität stark ab (vgl. auch Abbildung C.3 auf Seite 106).

Die schlechten Werte bei den Richtungsgesten XYEastWave, XYEastPointerUp, XYWestWave und XYWestPointerUp sind auf die ungenaue Formextraktion der Handregion zurückzuführen. Speziell die räumliche Segmentierung der Winkgesten ist sehr fehlerbehaftet (siehe auch Abbildungen C.3 und C.6 im Anhang C.1). Die Gesten XYWestPointerUp und XYEastPointerUp werden zumindest in den korrekten reduzierten Katalog XYWest bzw. XYEast eingeordnet (siehe Tabelle 6.3).

Um eine gestenbasierte Interaktion im Fahrzeug robuster zu gestalten und höhere Erkennungsraten zu erzielen, wird durch permanente Kontextwahrscheinlichkeiten $\vec{K}$ (siehe auch Abschnitt 5.4.1) das Gestenvokabular entsprechend eingeschränkt auf die ausführungskurzen Gesten XYWest, XYEast, XYNorth, XYSouth, XZNorth, XZSouth und auf die Wischgesten XWipe und ZWipe. Es wird dabei keine Unterscheidung der Handform getroffen (XYEast statt XYEastWave bzw. XYEastPointerUp und XYWest statt XYWestWave bzw. XYWestPointerUp). Die Erkennungsleistung des Klassifikators auf den reduzierten Gestenkatalog wird in Verbindung mit dem Spottingmodul in Abschnitt 6.4 durchgeführt.

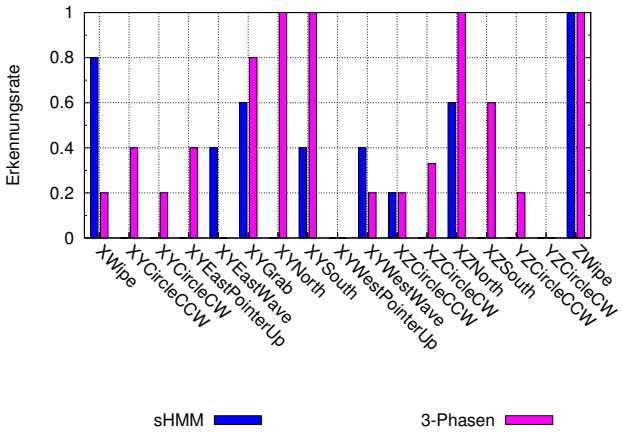


Abbildung 6.9.: Detaillierter Vergleich der Erkennungsraten von 3-Phasen und sHMM-basierten Algorithmus in der Fahrzeugumgebung.

Geste	3-Phasen	sHMM
XWipe	XZPlane	XYNorth
XYCircleCCW	Garbage	XWipe
XYCircleCW	Garbage	XWipe
XYEastPointerUp	XYEast	XYEastWave
XYEastWave	XZPlane	XWipe
XYGrab	XYPlane	XWipe
XYNorth	–	XYGrab
XYSouth	–	XZNorth
XYWestPointerUp	XYWest	XYWestWave
XYWestWave	XZPlane	XYNorth
XZCircleCCW	XZPlane	XWipe
XZCircleCW	XZPlane	ZWipe
XZNorth	–	ZWipe
XZSouth	XZPlane	ZWipe
YZCircleCCW	YZPlane	XWipe
YZCircleCW	YZPlane	ZWipe
ZWipe	–	–

Tabelle 6.3.: Häufigste Fehlklassifikationen in der Fahrzeugumgebung.

6.3.3. Einfluss von Kontextwissen

Um den Einfluss von externen Wissensquellen auf die Erkennungsleistung der Klassifikatoren qualitativ und praxisbezogen zu ermitteln, wurde ein typischer Bedienvorgang simuliert. Bei diesem Szenario steht, abhängig vom aktuellen Systemzustand, lediglich ein eingeschränktes Vokabular zur gestischen Interaktion zur Verfügung. Insgesamt gibt es vier Zustände auf die das gesamte Gestenvokabular wie folgt verteilt wird:

Zustand 1: XYWestWave, XYWestPointerUp, XYEastWave, XYEastPointerUp, XYNorth, XYSouth

Zustand 2: XYCircleCW, XYCircleCCW, XWipe

Zustand 3: XZNorth, XZSouth, XZCircleCW, XZCircleCCW

Zustand 4: YZCircleCW, YZCircleCCW, ZWipe

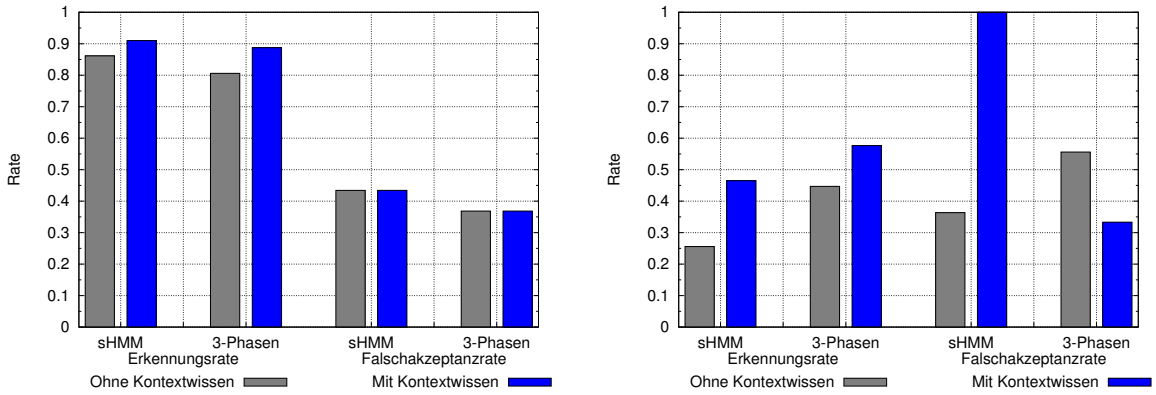
Für die Bestimmung der kontextabhängigen Erkennungsrate wird für jede Geste des Testdatensatzes der Systemzustand abhängig von der aktuellen Geste neu gesetzt. Dabei erfolgt der Wechsel immer in einen Zustand, der eine Interaktionsmöglichkeit mit der aktuellen Geste erlaubt. Es wird angenommen, dass der Anwender nur Gesten ausführt, die im aktuellen Menüzustand eine Operation zur Folge haben.

Formal werden die *Kontextwahrscheinlichkeiten* $\vec{K}$ (siehe auch Abschnitt 5.4.1) abhängig von der aktuellen Geste und somit vom aktuellen Zustand neu gesetzt. Ist beispielsweise im Testdatensatz die nächste zu erkennende Geste g ein XZSouth, wird in den Zustand 3 gewechselt und alle Gesten bis auf XZNorth, XZSouth XZCircleCW und XZCircleCCW ausgeblendet:

$$P_K(g) = \begin{cases} 1 \text{ für } g \in \{\text{XZNorth}, \text{XZSouth}, \text{XZCircleCW}, \text{XZCircleCCW}\} \\ 0 \text{ sonst} \end{cases} \tag{6.4}$$

Indem Bewegungen außerhalb des Gestenvokabulars innerhalb der vier Zustände klassifiziert werden – also mit reduziertem Gestenvokabular – wird analog zu Abschnitt 6.4 die Falschakzeptanzrate bestimmt (siehe auch Evaluierungskriterien im Abschnitt 6.1).

Sowohl in der Desktop- als auch in der Fahrzeugdomäne werden über die zusätzliche Wissensquelle, nach obigem Bedienszenario, verbesserte Erkennungsraten erzielt (siehe Abbildung 6.10(a)). In der Desktopumgebung lässt sich ein 10-prozentiger Zuwachs der Erkennungsraten des 3-Phasen-Klassifikators auf 89% und ein 5-prozentiger Zuwachs der Raten des sHMM-basierten-Klassifikators auf 90% beobachten. Die Falschakzeptanzraten sind dabei auf unverändertem Niveau.
Noch deutlicher fallen die Ergebnisse im Fahrzeug aus. Dort werden 20-prozentige Zunahmen auf 46% bei dem probabilistischen Verfahren und 10-prozentige Steigerungen auf 59% bei dem regelbasierten Ansatz erreicht. Die Falschakzeptanzrate kann bei dem 3-Phasen-Klassifikator von 55% auf 32% gedrückt werden. Auf die Falschakzeptanzrate des sHMM-basierten Klassifikators hat die zusätzliche Kontextquelle einen kontraproduktiven Effekt (siehe auch Abbildung 6.10(b)).



(a) Einfluss von Kontextwissen im Desktopbereich

(b) Einfluss von Kontextwissen in der Fahrzeugdomäne

Abbildung 6.10.: Einfluss von Kontextwissen in den 3-Phasen- und in den sHMM-basierten Klassifikator.

Zusammenfassend lässt sich sagen, dass externe Wissensquellen den Erkennungsprozess positiv beeinflussen können. Eine temporäre Reduktion des Gestenkatalogs erleichtert die Ent-

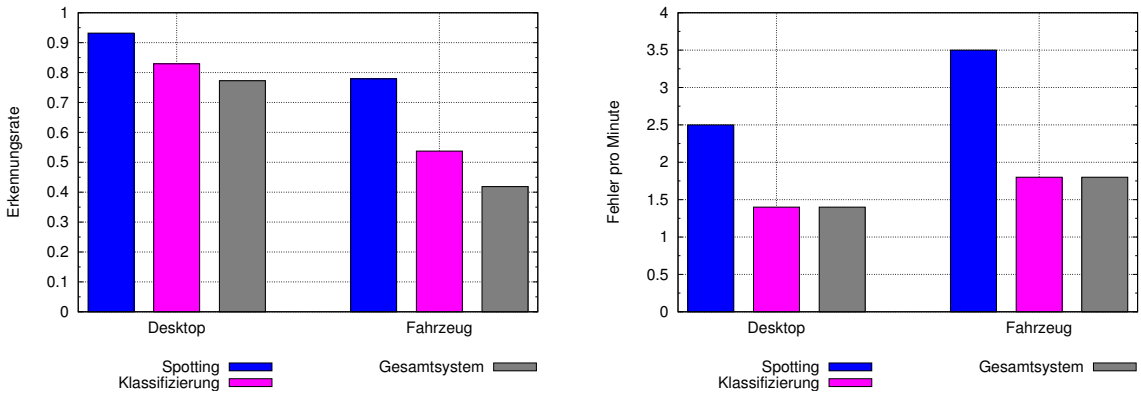
scheidungsfindung des Klassifikators. Dabei können Fehler im räumlichen Segmentierungsprozess und eine unsaubere Gestenausführung kompensiert werden.

6.4. Erkennungsleistung des Gesamtsystems

Die Ergebnisse dieser Diplomarbeit wurden mit den Erkenntnissen der Diplomarbeit von Rudi Lindl [Lin04] zu einem abgeschlossenen Gestenerkennersystem kombiniert, dessen Leistungsfähigkeit im Weiteren untersucht wird. Für das Gesamtsystem wurden die in Tabelle 6.4 beschriebenen Technologien bzw. Testdaten verwendet.

Phase	Verwendete Technologien	
	Desktop	Fahrzeug
Bilderfassung	RGB Farbbild	NIR Graustufenbild
Segmentierung	statischer Schwellwert vor schwarzem Hintergrund	Histogrammbasierter Schwellwert
Spotting	Regelbasiert	Regelbasiert
Klassifizierung	3-Phasen-Klassifikator	3-Phasen-Klassifikator
Testdaten	22 Minuten Testvideo: Gesamter Gestenkatalog	$2 \cdot 9$ Minuten Testvideo: Gesamter Gestenkatalog und ein auf 8 Gesten reduziertes Vokabular

Tabelle 6.4.: Verwendete Technologien im Gesamtsystem



(a) Erkennungsraten

(b) Fehler pro Minute

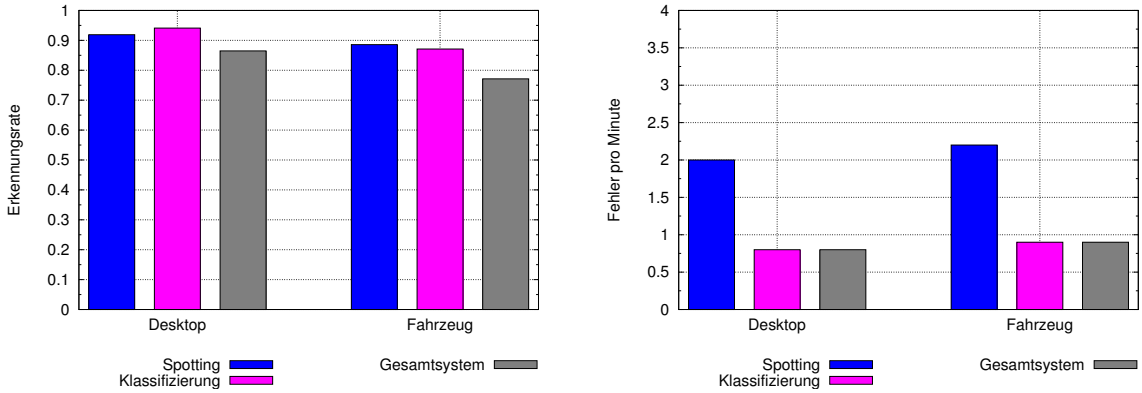
Abbildung 6.11.: Erkennungsleistung des Gesamtsystems (Klassifikator und Spotter) bei vollem Gestenkatalog in der Fahrzeug- und Desktopdomäne.

In der Desktopdomäne werden sehr gute Erkennungsraten von fast 80% erreicht. Das Spottingmodul bestimmt hierbei die obere Grenze der Systemleistung mit 93%. Zusätzlich ist der Klassifikator direkt abhängig von der Genauigkeit des Spottings. Werden Gesten vom Spotter

zu früh beendet oder zu spät begonnen, so fehlen wichtige Informationen für die anschließende Klassifizierung. Dieser Effekt kann jedoch vernachlässigt werden, da die automatisch ermittelten Start- bzw. Endpunkte durchschnittlich weniger als drei Frames von den manuell bestimmten Werten abweichen.

Besonders im Vergleich der Desktopsystemleistung mit den Erkennungsraten im Fahrzeug zeigt sich, dass das Gesamtsystem sehr abhängig von der Qualität der räumlichen Segmentierung ist. Insbesondere die Klassifizierung besitzt in dieser Domäne eine deutlich geringere Erkennungsrate. Das Spotting erreicht unter diesen Randbedingungen Erkennungsraten von 78%. Auch hier ist eine deutliche Korrelation zur Qualität der Segmentierung gegeben, die allerdings keinen Einfluss auf die Genauigkeit des Spottings ausübt (siehe auch Abbildung 6.11(a)). Eine detaillierte Evaluierung der Spottingverfahren findet sich in der Diplomarbeit von Rudi Lindl [Lin04].
Die prinzipbedingten hohen Fehlerraten des regelbasierten Spottings (2,5 Fehler pro Minute) werden durch die Klassifizierung auf 1,4 Fehler pro Minute halbiert (siehe Abbildung 6.11(b)). In Anbetracht der Tatsache, dass die verwendeten Testvideos fast ausschließlich aus Bewegungen bestehen, stellt dies einen sehr guten Wert dar. Im Fahrzeug besitzt der Fehler zweiter Art mit 1,8 Fehlern pro Minute ein zur Desktopumgebung identisches Verhalten.

Die niedrige Gesamtsystemleistung von 42% in der Fahrzeugdomäne führte zu einer Reduktion des Gestenwortschatzes. Komplexe Gesten, wie zum Beispiel die Kreis- oder Greifgesten, wurden aus dem Katalog entfernt, da sie aufgrund der schlechten Segmentierung schwierig zu klassifizieren sind (siehe auch Abschnitt C.1). Der resultierende reduzierte Wortschatz enthält die acht folgenden Gesten: XYWest, XYEast, XYNorth, XYSouth, XWipe, XZNorth, XZSouth und ZWipe.



(a) Erkennungsraten (b) Fehler pro Minute

Abbildung 6.12.: Erkennungsleistung des Gesamtsystems (Klassifikator und Spotter) bei einem auf 8 Gesten reduzierten Gestenkatalog in der Fahrzeug- und Desktopdomäne.

Für das Spotting lässt sich durch die Reduktion des Kataloges nur eine marginale Verbesserung der Erkennungsrate um 5% erzielen. Die Erkennungsrate des Klassifikators steigt hingegen signifikant um 33% auf insgesamt 88%. Somit wird eine Gesamtsystemleistung im

Fahrzeug von 76% bei einer Fehlerrate von 0,9 Fehlern pro Minute erreicht (siehe Abbildung 6.12).

KAPITEL 7

RÉSUMÉ

7.1. Zusammenfassung

Mit dieser Arbeit konnte demonstriert werden, dass eine videobasierte maschinelle Erkennung dynamischer Gesten im automotiven Umfeld grundsätzlich möglich ist. Neben dem Vergleich klassischer Bildverarbeitungsansätze zur räumlichen Segmentierung von Hand- und Kopfregionen war die Implementierung und Evaluierung dynamischer Klassifikatoren ein Schwerpunkt dieser Arbeit. Weiterhin wurde untersucht inwiefern externe Wissensquellen Potential zur Steigerung der Detektionsleistung beinhalten und wie sie in den Erkennungsprozess integriert werden können.
Zur Klassifizierung von 17 dynamischen Handgesten wurde ein regelbasiertes hierarchisches Verfahren von Mammen [MCA02] und ein probabilistischer Ansatz basierend auf HMM's von Morguet [Mor00] gewählt und implementiert. Die Algorithmen wurden erweitert, um das Gestenvokabular dieser Arbeit abzudecken. Zudem wurden beide Verfahren in einer Art und Weise modifiziert, die eine Einflussnahme externer Wissensquellen ermöglicht.
Auf dem Weg zum finalen System wurden zu Erprobungszwecken unterschiedlichste Segmentierungsverfahren zunächst in der Desktopdomäne prototypisch umgesetzt. Verstärkt wurden dabei visuelle Ansätze im dreidimensionalen Raum thematisiert. Ein Systemeinsatz scheiterte jedoch an immanenten Synchronisationsproblemen der Kamerakomponenten.
Um zum einen das Potential von „low-cost" Hardware zu untersuchen und zum anderen die verfügbaren Leistungsfaktoren zukünftiger Fahrzeuge zu simulieren, wurden bewusst handelsübliche Webcams und aktuelle PCs als Basishardware für das Erkennersystem eingesetzt. Unter Einhaltung dieser Randbedingungen wurde ein System realisiert, das Handgesten unabhängig vom Benutzer im Echtzeitbetrieb erkennt. Im Desktopbereich wurden mit dem kompletten Gestenschatz Erkennungsraten von 86% erzielt. In der Fahrzeugdomäne betrug die Erkennungsleistung 88%, wobei das verfügbare Gestenvokabular auf acht Instanzen reduziert wurde.

In Kooperation mit Rudi Lindl wurde eine Softwarearchitektur konzipiert. Neben einer Komplexitätsminderung förderte diese durch einheitliche Schnittstellendefinitionen das Arbeiten im Team und vereinfachte den Integrationsprozess. Als Vorbild für die Zergliederung des Gesamtprojektes in Teilkomponenten dienten hierbei die Phasen der konventionellen Bildverarbeitungspipeline. Erst durch diese Modularisierung wurde eine dynamische Substituierbarkeit einzelner Teilkomponenten zur Laufzeit möglich.

Zur maschinellen Lokalisierung der Hand- und Kopfregion des Nutzers im Videobild, wurden in dieser Arbeit vornehmlich passive stereoskope Segmentierungsverfahren untersucht. Im Gegensatz zum Birchfield Algorithmus [BT99], dessen resultierendes Tiefenbild zahlreiche Artefakte aufwies, bewerkstelligte der SMP-Stereoalgorithmus [DSMM04] eine robuste Szenenrekonstruktion in Echtzeit. Ausgehend von den Tiefeninformationen konnten sowohl Kopf- als auch Handregionen weitgehend unabhängig von den vorherrschenden Lichtverhältnissen erfolgreich lokalisiert werden, sofern sich die relevanten Szenenobjekte in Ruhe befanden. Bei auftretender Bewegung war eine räumliche Segmentierung nicht mehr gewährleistet, da die Eingabebilder hardwarebedingt nicht exakt zeitgleich eingelesen wurden. So müssten die eingesetzten USB-Kameras mit einer Triggereinheit synchron geschaltet werden können, was für handelsübliche kostengünstige Modelle zum gegenwärtigen Zeitpunkt nicht vorgesehen war.
Um dennoch den Einfluss von Tiefeninformation auf die Güte der Erkennung prüfen zu können, wurde die 3D-Information zum einen über eine Versatzbestimmung der Hand in einem Zweikamerasystem und zum anderen indirekt über die Flächenänderung der extrahierten Handregion in einem monokularen System gewonnen. Im Vergleich zum Einkamerasystem konnte die Erkennungsrate von 68% gegenüber dem Zweikamerasystem nur geringfügig auf 81% angehoben werden. Lediglich die Falschakzeptanzrate liess sich durch die zusätzliche Tiefeninformation signifikant von 58% auf 37% drücken.

Mit über 95% erreichte das probabilistische Verfahren benutzerabhängig die höchsten Erkennungsraten im Desktopbereich. Selbst wenn die stochastischen Modelle allein mit Informationen zur Trajektorie der Gesten trainiert wurden, erzielte das Verfahren eine Erkennungsleistung von 91%. Somit belegte der HMM-basierte Ansatz eindrucksvoll die gängige Forschermeinung: Dynamische Handgesten sind vorwiegend durch ihren Bewegungsverlauf charakterisiert und weniger durch die Gestalt der Hand. Bei einer benutzerunabhängigen Evaluierung sanken die Raten auf 86%.
Das hierarchische Berechnungsschema des regelbasierten Algorithmus, das den Ergebnisraum schrittweise eingrenzt, erwies sich bei verminderter Segmentierungsqualität als sehr flexibel und robust. Somit konnte das Verfahren insbesondere im Fahrzeug die beste Klassifikatorleistung erzielen. Mit einem auf acht Gesten reduzierten Katalog betrug die Erkennungsrate im automotiven Umfeld 88%.
Mit der Interpretation von Wissensquellen durch die Klassifikatoren liessen sich diese Raten weiter steigern. Dazu wurden abhängig vom aktuellen Kontextwissen gezielt Gesten bei der Erkennung bevorzugt bzw. benachteiligt oder völlig ignoriert. Neben dieser Gestenrevalidierung wurde beim regelbasierten Klassifikator darüber hinaus eine kontextabhängige Vorverarbeitung der Merkmalssequenzen durchgeführt. Untermauert wird der Nutzen dieser Integration von Kontextwissen durch einen 5-prozentigen Zuwachs der Erkennungsraten in

der Desktop- und einen 15-prozentigen Zuwachs in der Fahrzeugdomäne. Desweiteren konnte die Falschakzeptanzrate im automotiven Umfeld von 58% auf 37% beträchtlich reduziert werden. Eine Integration von Wissensquellen in den Erkennungsprozess birgt demnach beträchtliches Potential zur Steigerung der Fehlerrobustheit im Mensch-Maschine-Dialog.

In Zusammenarbeit mit Rudi Lindl [Lin04] wurden in einem letzten Schritt die Spotter- und Klassifikatorverfahren zu einem funktionsfähigen Gesamtsystem kombiniert. Um dem Nutzer eine gestische Bedienung einer realen grafischen Interaktionsplattform zu erlauben, wurde das Erkennersystem gemeinsam an die BMW-interne Interaktionsplattform MIAMII angebunden und somit erlebbar gemacht.

7.2. Ausblick

Abschließend werden einige mögliche Ergänzungen und Verbesserungen des Systems zur Gestenerkennung dargelegt.

Klassifikatoren

HMM-basierte Verfahren profitieren enorm von einem großen Trainingskorpus. Insbesondere für eine benutzerunabhängige Erkennung von Gesten sollte von sehr vielen Probanden Trainingsmaterial vorliegen, um möglichst alle Variationen in der Bewegungsausführung abzudecken. Eine derartig aufwendige Datenerfassung konnte im Rahmen der Arbeit aus Zeitgründen nicht durchgeführt werden.
Wenn nach und nach anwenderspezifische Eigenheiten der Gestenausführung in die stochastischen Modelle aufgenommen würden, könnten diese zur Laufzeit nachtrainiert werden. Eine schrittweise Adaption des Klassifikators an den Anwender könnte somit den Erkennungsprozess robuster gestalten.
Die als Bagging und Boosting bekannte Zusammenführung von Klassifikatoren wertet die Ergebnisse mehrerer Erkenner aus und kombiniert diese, um einen neuen leistungsfähigeren Klassifikator zu kreieren. So könnten beispielsweise regelbasierte und probabilistische Klassifikatoren fusioniert werden, um von den Vorzügen beider Verfahren profitieren zu können.

Räumliche Segmentierung

Schwächstes Glied im Mustererkennungsprozess ist die räumliche Segmentierung. Insbesondere im Fahrzeug macht sich dies durch eine verminderte Leistung des Erkennersystems bemerkbar. Schwierigkeiten bereiten den vorgestellten Verfahren in erster Linie dynamische Beleuchtungsbedingungen, wie Direkteinstrahlung von Sonnenlicht und starker Schattenwurf. Durch die Bilderfassung im nahen Infrarotbereich konnte diese Problematik zwar abgemindert, aber nicht vollkommen behoben werden. Eine nichtintrusive, räumliche Segmentierung

mit alternativen Techniken, wie Abstandssensorik [Gei03] oder Ultraschall [BBNB03] könnte gegebenenfalls diese Probleme mindern und zu einem robusteren System beitragen. Dabei müssen aber gleichermaßen die typischen Randbedingungen im automotiven Umfeld, wie Kostenfaktoren, Echtzeitfähigkeit, Leistungsaufnahme und begrenzter Bauraum beachtet werden.

Use-Cases

Mögliche Anwendungsgebiete für gestische Bediensituationen im Fahrzeug wurden in dieser Arbeit nur am Rande thematisiert. Vielmehr stand die technische Umsetzung des Erkennersystems im Vordergrund. Auf dem Weg zu einem kommerziellen Produkt stehen jedoch die Anwendungsmöglichkeiten im Zentrum des Interesses. Neben einer einfachen gestenbasierten Abhandlung von Ja-Nein Dialogen, Listennavigationen und Abbruch von Systemrückmeldungen („barge-in“) sind weitere Use-Cases denkbar. So könnten insbesondere Regelgesten kontinuierliche Anwendungsfälle, wie Klimaeinstellungen oder Lautstärkenänderungen abdecken. Bei einer personalisierten Interaktion könnte der Anwender selbst Gesten definieren und diese beispielsweise für häufig durchgeführte Bedienschritte als „shortcut“nutzen.

Wissensquellen

Als externe Wissensquelle wird in dieser Arbeit neben einem Kontextsimulator lediglich der aktuelle Systemzustand der MMI-Demonstrationsplattform MIAMII herangezogen. Die Anbindung und Nutzung weiterer externer Informationslieferanten würde das betrachtete Wissensspektrum erweitern und eine umfassendere Kontextmodellierung ermöglichen. Ferner müsste die konzipierte Infrastruktur zur Kontextnutzung weiter ausgebaut und neuartige Integrationsansätze von Kontextwissen in die Klassifikatoren erarbeitet werden.

ANHANG A

ÜBERSICHT WICHTIGER KONSTANTEN

Parameter	Beschreibung	Wert
	Geometrische Maße	
	Horizontaler Abstand zwischen den beiden Stereokameras.	4,5 cm
	Abstand der Stereokameras vom Hintergrund.	80 cm
	Maße der Grundfläche des Gesteninteraktionsbereichs.	50 cm · 38 cm
	SMP-Stereosegmentierung	
w	Breite eines Kamerabildes in Pixel.	320
h	Höhe eines Kamerabildes in Pixel.	240
ρ_P	Seitenlänge des rechteckigen Korrelationsfensters.	7
d_P	Suchlänge in Pixel (maximale Disparität).	40
ϑ_P	Schwellwert für abschließende statische Schwellwertsegmentierung.	140
a_P, b_P	Durchmesser für strukturierende Elemente.	6, 10
	Schwellwertsegmentierung	
ϑ_S	Schwellwert für die statische Segmentierung.	83
ρ	Minimal zulässiges Seitenverhältnis des kleinsten, die Handregion umschließendes Rechtecks (Abbruchkriterium für Armfilteroperation).	0.3
ω	Winkel zwischen Orientierung und Schnittgeraden bei der Armfilterung.	45°
χ	Maximal zulässiger Flächeninhalt einer Handregion.	500

Parameter	Beschreibung	Wert
	3-Phasen-Klassifikator	
η_M	Kernelgröße des Mittelwertsfilters.	3
η_G	Kernelgröße des Gaußfilters.	5
ξ_Z	Faktor zur Skalierung der Z-Trajektorie.	4
ξ_A	Faktor zur Skalierung der A-Trajektorie.	-100
θ_e	Minimale Schranke für die Anzahl von Maxima zur Erkennung von Wischgesten.	3
θ^Z_{XY}	Minimal nötiges Verhältnis zwischen den Amplituden der Z- und X- bzw. der Z- und Y-Trajektorien damit Z dominant wird.	1
θ^{XY}	Maximal zulässiges Verhältnis zwischen den Amplituden der X und Y Trajektorien um eine dominante Trajektorie zu ermitteln.	0.4
θ^X	Minimal zulässige Amplituden für X als dominante Trajektorie.	15
θ^Y	Minimal zulässige Amplituden für Y als dominante Trajektorie.	15
θ^Z	Minimal zulässige Amplituden für Z als dominante Trajektorie.	20
θ^X_d	Verhältnis zwischen der X und Y Amplitude (mit Y als dominanter Trajektorie), ab dem beide Trajektorien dominant werden.	0.8
θ^Y_d	Verhältnis zwischen der Y und X Amplitude (mit X als dominanter Trajektorie), ab dem beide Trajektorien dominant werden.	0.6
d	Minimale mögliche Wahrscheinlichkeit, bevor der Erkenner Garbage zurückliefert.	0.3
	sHMM-Klassifikator	
τ	Minimal zulässiger Ausgangsscore.	-2000
L	Anzahl der Gaußmixturen.	200
υ	Maximal zulässige Scoredifferenz der beiden führenden sHMM.	10

Tabelle A.1.: Systeminterne Parameter.

ANHANG B

KLASSENDIAGRAMM DES GESTENERKENNERSYSTEMS

Auf der nächsten Seite findet sich das Klassendiagramm des Gestenerkennersystems. Um die Übersichtlichkeit zu wahren, wurden die Klassen zur Datenerfassung (*BmpReader*, *AviReader*, *IrCamReader*, *StereocamReader* und *WincamReader*) nicht eingezeichnet. Darüber hinaus sind nur die wichtigsten Methoden der einzelnen Klassen angegeben. Eine ausführliche Klassenübersicht mit einer Beschreibung aller Methoden und Variablen findet sich in der dem Quellcode beiliegenden Doxygen-Dokumentation [Dim04].

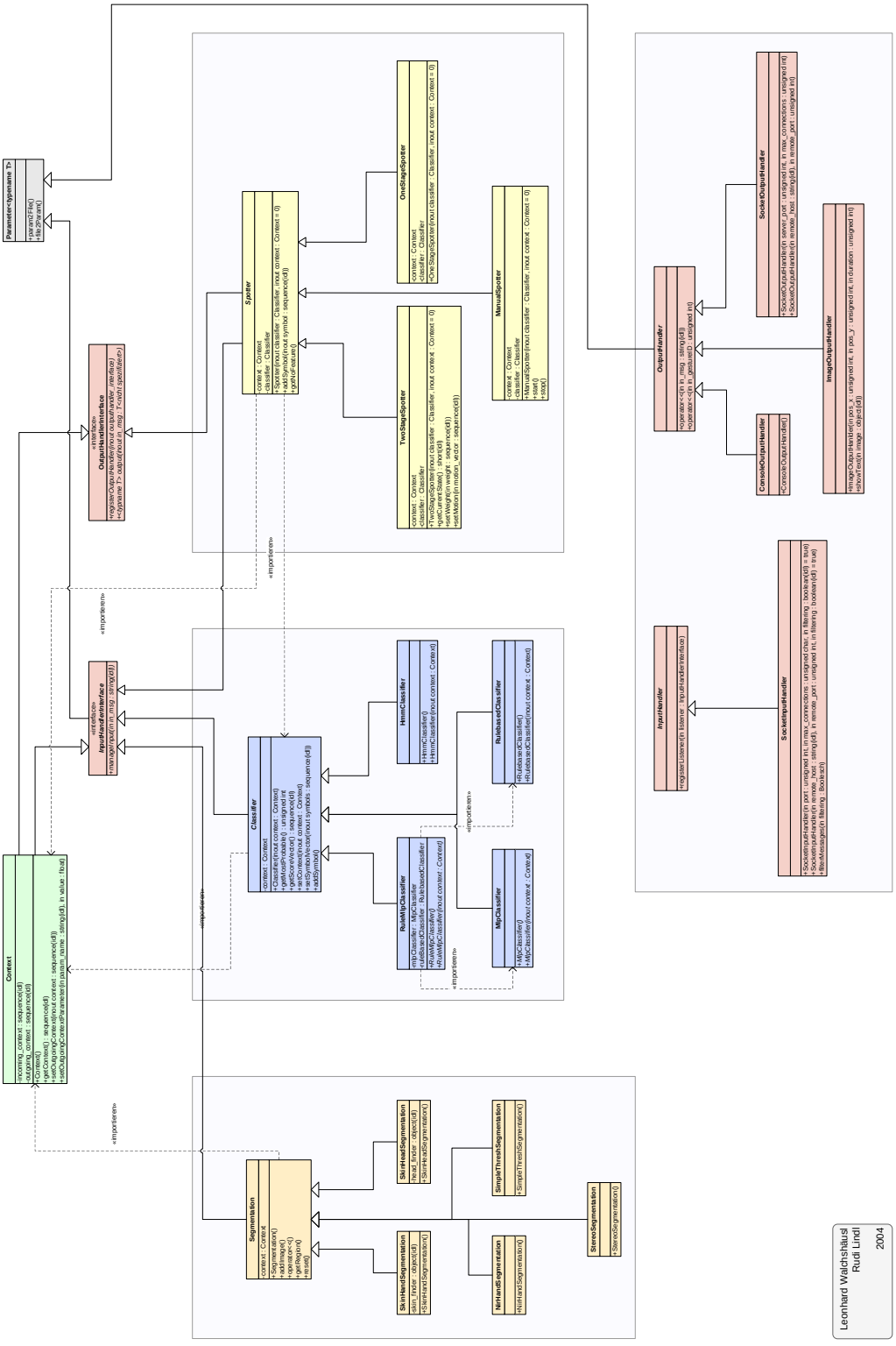


Abbildung B.1.: Klassendiagramm des Gestenerkenners.

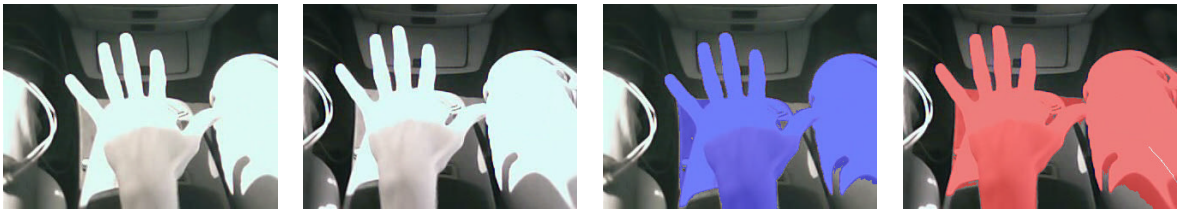
ANHANG C

ZUSÄTZLICHE DIAGRAMME FÜR DIE EVALUIERUNG

C.1. Ausgewählte Segmentierungsprobleme

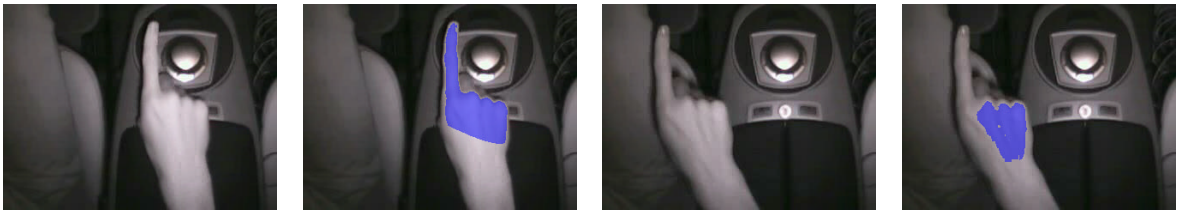
(a) (b) (c) (d)

Abbildung C.1.: Verhalten des SMP-Algorithmus bei Bewegungsunschärfe. Linkes und rechtes Kamerabild (a,b), Errechnetes Tiefenbild und Ergebnis (c,d).



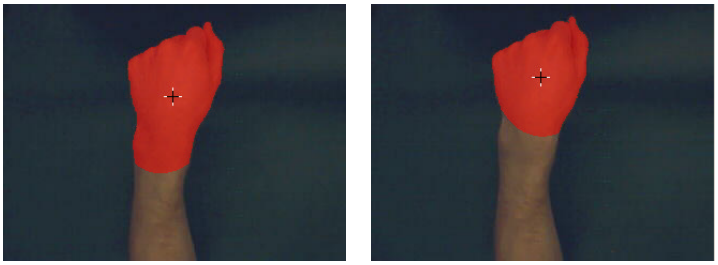
(a) (b) (c) (d)

Abbildung C.2.: Schwellwertsegmentierung im Fahrzeug bei starker Sonneneinstrahlung. Linkes und rechtes Infrarot-Kamerabild (a,b), Segmentierungsergebnis (c,d).



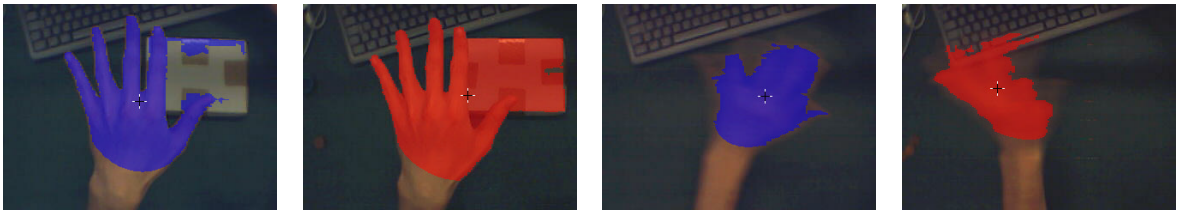
(a) (b) (c) (d)

Abbildung C.3.: Inhomogene Ausleuchtung im Fahrzeug. Vor allem an der Peripherie des Kamerasichtbereichs treten deshalb verstärkt Segmentierungsfehler auf. Bilder von der Infrarotkamera im Fahrzeug (a, c), Ergebnis der histogrammbasierten Schwellwertesegmentierung (b, d).



(a) (b)

Abbildung C.4.: Unstetige Flächenänderungen bei der Armfilteroperation. 1-fache Armfilterung (Rekursionstiefe 1) (a), 2-fache Armfilterung (Rekursionstiefe 2) (b).



(a) (b) (c) (d)

Abbildung C.5.: Fehlerquellen für die Bestimmung der Tiefeninformation über den räumlichen Versatz. Starker Fehler in der räumlichen Segmentierung des rechten Kamerabildes (a,b), Synchronisationsfehler der beiden Kamerabilder (c,d).

(a) (b) (c) (d)

Abbildung C.6.: Ungenügende Armfilterung bei den Winkgesten. Aufgrund der schmalen Handregion wird die Filteroperation zu oft iteriert. Eingabebild und Ergebnis im Desktopbereich (a,b), Eingabebild und Ergebnis im Fahrzeug (c,d).

C.2. Benutzerunabhängige Erkennung mittels sHMM in der Desktopumgebung

Proband	XWipe	XYCircleCCW	XYCircleCW	XYEastPointerUp	XYEastWave	XYGrab	XYNorth	XYSouth	XYWestPointerUp	XYWestWave	XZCircleCCW	XZCircleCW	XZNorth	XZSouth	YZCircleCCW	YZCircleCW	ZWipe	∅
VP1	1	0.2	0.86	0.95	0.95	1	1	1	1	0.95	1	1	0.93	0.85	1	1	1	**0.93**
VP2	1	0.6	1	1	1	1	0.94	0.94	1	1	0.88	1	0.88	0.88	1	1	0.91	**0.94**
VP3	1	0.94	1	0.75	1	1	1	1	0.4	0.63	1	0.94	0.79	0.69	1	1	1	**0.88**
VP4	1	1	0.56	0.88	1	1	1	1	0.88	1	0.93	1	0.35	0.59	0.94	0.88	1	**0.88**
VP5	1	0	0	0	1	1	0	0.88	0	0.8	0.94	0.88	1	0.85	1	1	0.94	**0.66**
VP6	1	0.75	0.92	0.56	1	1	0.94	1	0.88	0.94	1	1	0.85	0.71	1	1	0.46	**0.89**
∅	1	0.58	0.72	0.69	0.99	1	0.81	0.97	0.69	0.89	0.96	0.97	0.79	0.76	0.99	0.98	0.88	**0.86**

Tabelle C.1.: Einzelerkennungsraten für die sechsfache Kreuzvalidierung zur Bestimmung der benutzerunabhängigen Erkennungsrate des sHMM-basierten Klassifikators ohne Einfluss von Kontextwissen.

Proband	XWipe	XYCircleCCW	XYCircleCW	XYEastPointerUp	XYEastWave	XYGrab	XYNorth	XYSouth	XYWestPointerUp	XYWestWave	XZCircleCCW	XZCircleCW	XZNorth	XZSouth	YZCircleCCW	YZCircleCW	ZWipe	∅
VP1	1	0.67	1	1	1	1	1	1	1	0.95	1	1	1	1	1	1	1	**0.98**
VP2	1	0.6	1	1	1	1	0.94	1	1	1	0.94	1	1	1	1	1	0.91	**0.96**
VP3	1	0.94	1	0.75	1	1	1	1	0.4	0.63	1	1	1	0.92	1	1	1	**0.91**
VP4	1	1	0.56	0.88	1	1	1	1	0.88	1	0.93	1	0.85	1	0.94	0.88	1	**0.94**
VP5	1	0	0	0	1	1	0	0.88	0	0.8	1	0.94	1	1	1	1	1	**0.67**
VP6	1	0.92	0.92	0.81	1	1	1	1	1	0.94	1	1	0.92	0.93	1	1	1	**0.97**
∅	1	0.68	0.75	0.74	1	1	0.82	0.98	0.71	0.89	0.98	0.99	0.96	0.98	0.99	0.98	0.99	**0.89**

Tabelle C.2.: Einzelerkennungsraten für die sechsfache Kreuzvalidierung zur Bestimmung der benutzerunabhängigen Erkennungsrate des sHMM-basierten Klassifikators mit Einfluss von Kontextwissen.

ANHANG D

KONFIGURATIONS- UND SIMULATOR GUI

D.1. Systemdemonstrator

Die Demonstratoroberfläche erlaubt eine Konfiguration des Gestenerkennersystems zur Laufzeit. Dazu werden Kommandos über das Netzwerk geschickt, die auf der Empfängerseite als Aktionen interpretiert und ausgeführt werden. Kern der in TCL/TK [Act04] programmierten GUI[1] bildet das Hauptdialogfeld (siehe Abbildung D.1), das sich in zwei große Bereiche unterteilen lässt. Im oberen Dialogabschnitt lassen sich Netzwerkeinstellungen treffen sowie Verbindungen zum Zielrechner aufbauen. Desweiteren können alle Einstellungen aus einer Datei ausgelesen werden. Der untere Bereich ist in die drei Abschnitte „Classifier“, „Spotter“ und „Segmentation“ gegliedert, wobei über „Radiobuttons“ die verschiedenen Verfahren gewählt werden können. Sämtliche Feineinstellungen zu den Verfahren werden in separaten Fenstern getroffen (siehe auch Abbildungen D.2, D.3 und D.4), die sich über die entsprechenden Druckknöpfe im unteren Bereich des Hauptdialogfeldes öffnen bzw. schließen lassen.

Bezeichnung	Parameter	Beschreibung
Size of Medianmask	η_M	Kernelgröße des Mittelwertsfilters.
Size of Gausskernel	η_G	Kernelgröße des Gaußfilters.
Scale of Z Axis	ξ_Z	Faktor zur Skalierung der Z-Trajektorie.
Scale of Area	$\frac{1}{\xi_A}$	Faktor zur Skalierung der A-Trajektorie.
Z Garbage Difference	θ_{XY}^{Z}	Minimal nötiges Verhältnis zwischen den Amplituden der Z- und X- bzw. der Z- und Y-Trajektorien damit Z dominant wird.

[1] graphical user interface.

Bezeichnung	Parameter	Beschreibung
Garbage Difference	θ^{XY}	Maximal zulässiges Verhältnis zwischen den Amplituden der X und Y Trajektorien um eine dominante Trajektorie zu ermitteln.
Garbage Minimum X	θ^{X}	Minimal zulässige Amplituden für X als dominante Trajektorie.
Garbage Minimum Y	θ^{Y}	Minimal zulässige Amplituden für Y als dominante Trajektorie.
Garbage Minimum Z	θ^{Z}	Minimal zulässige Amplituden für Z als dominante Trajektorie.
Extremacount for XWipe	θ^{e}	Mindestanzahl an lokalisierten Extremwerten, bevor eine Geste als XWipe erkannt wird.
Extremacount for YWipe	θ^{e}	Mindestanzahl an lokalisierten Extremwerten, bevor eine Geste als YWipe erkannt wird.
Extremacount for ZWipe	θ^{e}	Mindestanzahl an lokalisierten Extremwerten, bevor eine Geste als ZWipe erkannt wird.
X Dominance Fraction	θ_d^{X}	Verhältnis zwischen der X und Y Amplitude (mit Y als dominanter Trajektorie), ab dem beide Trajektorien dominant werden.
Y Dominance Fraction	θ_d^{Y}	Verhältnis zwischen der Y und X Amplitude (mit X als dominanter Trajektorie), ab dem beide Trajektorien dominant werden.
Minimum Extrema Distance		Mindestabstand zwischen zwei Extremwerten bei der Funktionsanalyse.
Extrema Margin		Extrema, die diesen Mindestabstand vom Anfang bzw. vom Ende der Merkmalsvektorsequenz, unterschreiten werden ignoriert.
Is Input Deviation?		Die möglichen Werte sind true oder false. Wenn true ausgewählt wird erwartet der Klassifikator den Merkmalsvektor in relativer Darstellung im Gegensatz zu einer absoluten Darstellung.
Take only Area as Z?		Die möglichen Werte sind true oder false. Wenn true ausgewählt wird, verwendet der Klassifikator nur die Flächenänderung der Handregion als Indikator für die Tiefeninformation und nicht die Fusion von Fläche und Versatz-Z-Wert (siehe auch Abschnitt 5.2.3).

Tabelle D.1.: Einstellungsmöglichkeiten für den 3-Phasen-Klassifikator.

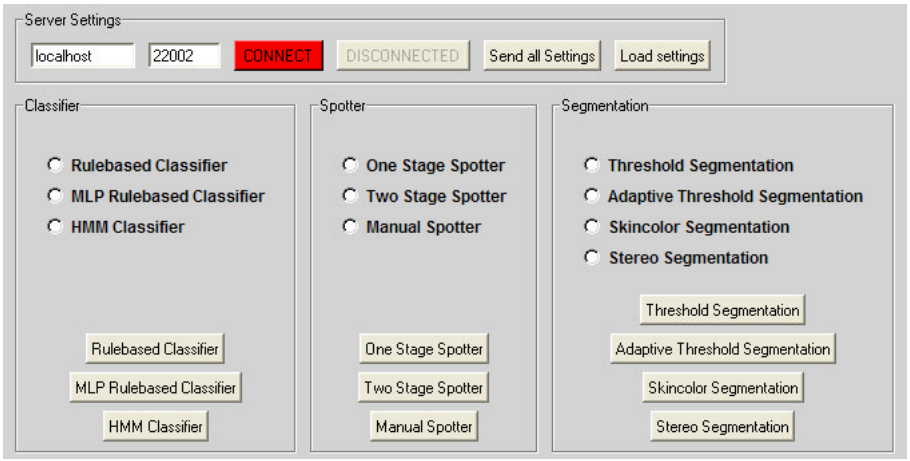


Abbildung D.1.: Hauptdialogfeld für den Systemdemonstrator.

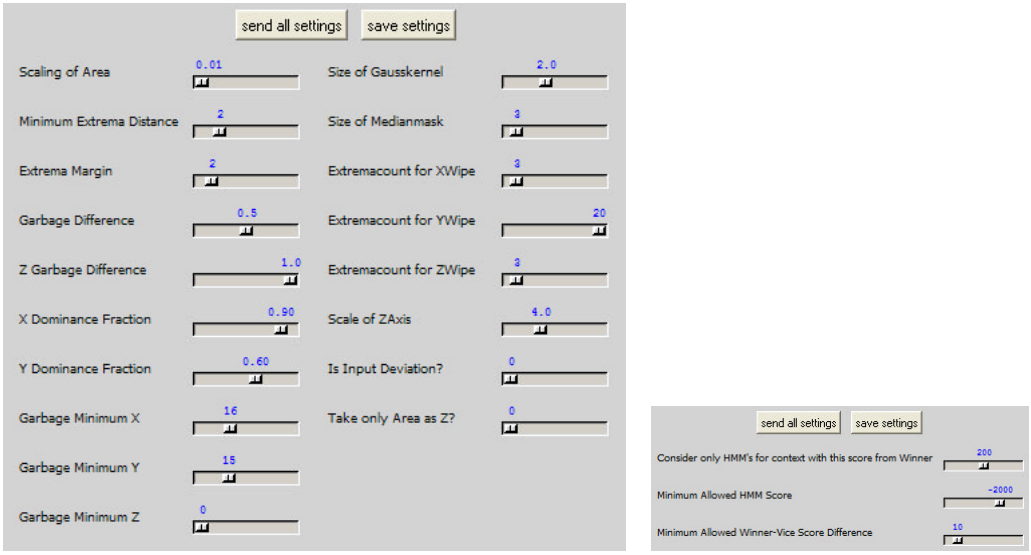


Abbildung D.2.: „MLP Rulebased Classifier“ Einstellungen zum 3-Phasen Klassifikator (links), „HMM Classifier“ Einstellungen zum sHMM-basierten Klassifikator (rechts).

Bezeichnung	Parameter	Beschreibung
Consider only sHMM's with this score from Winner		Nötiger Scoreabstand vor Kontextzuweisung zum führenden HMM.
Minimum Allowed HMM Score	τ	Minimaler Score, bevor das sHMM von der Erkennung ausgeschlossen wird.
Minimum Allowed Winner-Vice Score Difference	υ	Faktor zur Skalierung der Z-Trajektorie.

Tabelle D.2.: Einstellungsmöglichkeiten für den sHMM-basierten Klassifikator.

Abbildung D.3.: „Skincolor Segmentation" Einstellungen für das hautfarbenbasierte Segmentierungsverfahren.

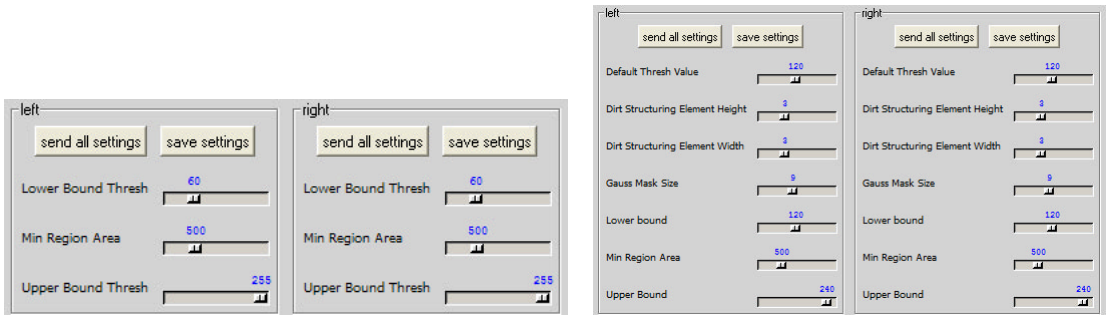


Abbildung D.4.: „Threshold Segmentation" Einstellungsmöglichkeiten für das statische Schwellwertverfahren (links) „Adaptive Threshold Segmentation" Einstellungsmöglichkeiten für das histogrammbasierte Schwellwertverfahren (rechts).

D.2. Kontextsimulator

Über den in TCL/TK [Act04] programmierten Kontextsimulator können beliebige Kontextwahrscheinlichkeiten (siehe auch Abschnitt 5.4.1) für das gesamte Gestenvokabular (siehe auch Gestenkatalog in Abschnitt 5.6 auf Seite 74 und Gestenübersicht in Abbildung 5.8) vorgegeben werden. Dazu wird über das Netzwerk eine Verbindung mit dem Kontextserver aufgebaut. Mit Hilfe von Schiebereglern kann nun für jede Geste bzw. jeden reduzierten Katalog

eine Auftrittswahrscheinlichkeit zwischen 0 und 1 vorgegeben werden (siehe Abbildung D.5). Sämtliche Änderungen am Kontextsimulator fließen unmittelbar in den aktuellen Erkennungsprozess mit ein.

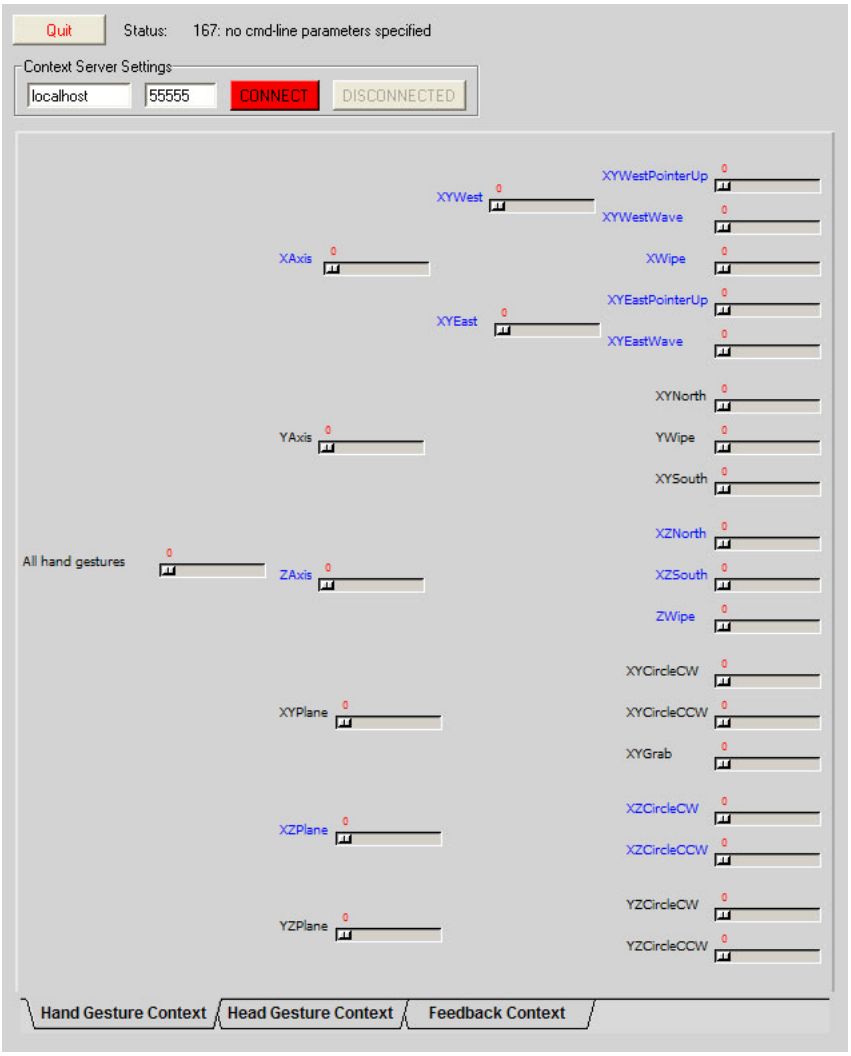


Abbildung D.5.: Einstellungsmöglichkeiten zum simulierten Kontextwissen.

Der sogenannte „Feedback-Kontext“ wurde prototypisch implementiert (siehe auch Abschnitt 5.4.1). Dieser setzt sich zusammen aus neuem Wissen vom Gestenerkennersystem. Gegenwärtig umfasst dieser die durchschnittliche Beschleunigung der Gestenausführung („Gesture Acceleration“), die Sicherheit mit welcher die Geste erkannt wurde („Gesture Assurance“), die Dauer der Ausführung („Gesture Duration“) und die ermittelte Segmentierungsqualität. Letztere definiert sich aus dem durchschnittlichen Anteil der Bilder, in denen erfolgreich eine Hand extrahiert werden konnte. Alle diese Werte werden während der laufenden Erkennung im Fenster in Abbildung D.6 mittels Netzwerknachrichten aktualisiert.

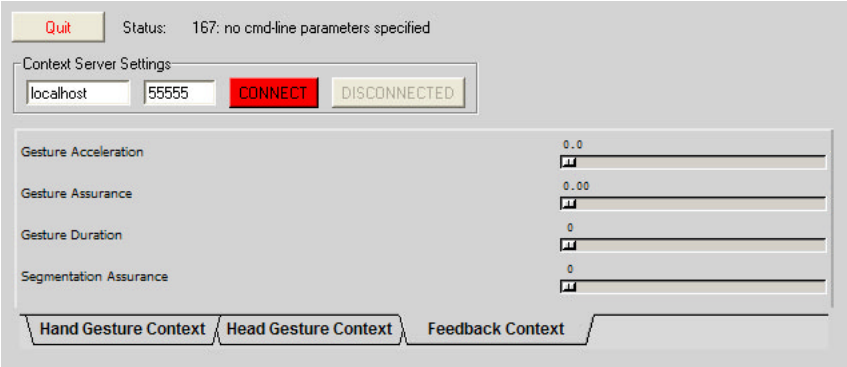


Abbildung D.6.: Darstellung für den „Feedback“-Kontext.

ANHANG E

TRAJEKTORIENVERLÄUFE ALLER GESTEN

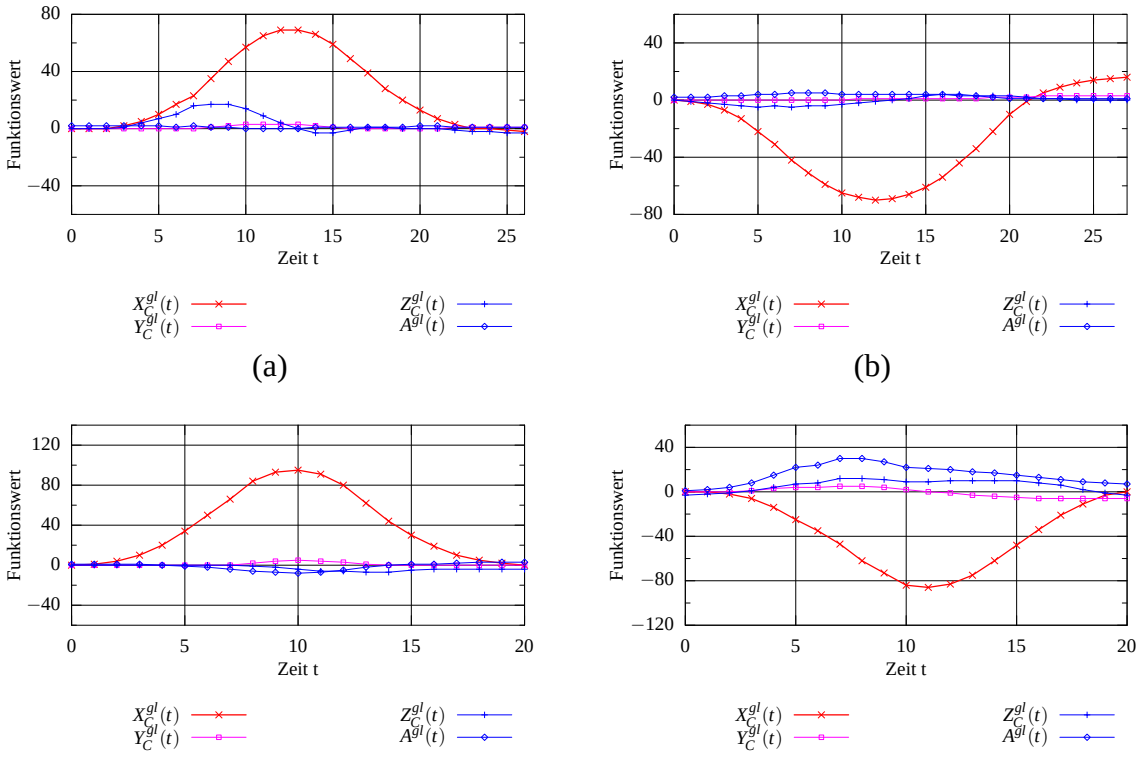
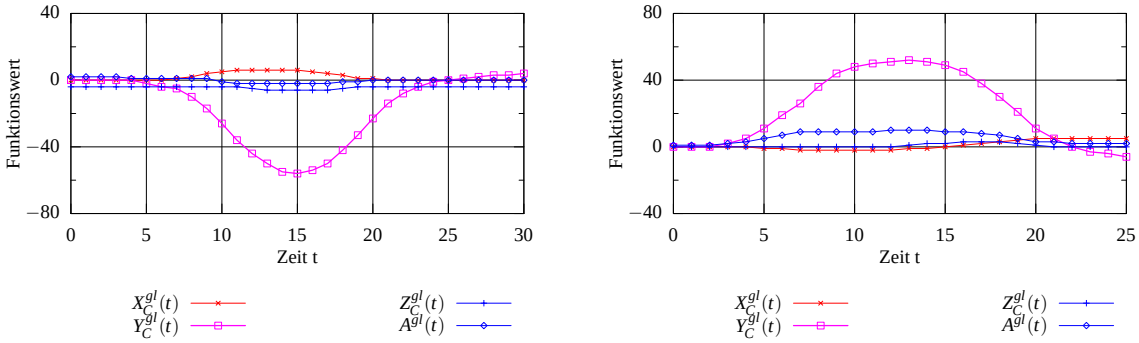
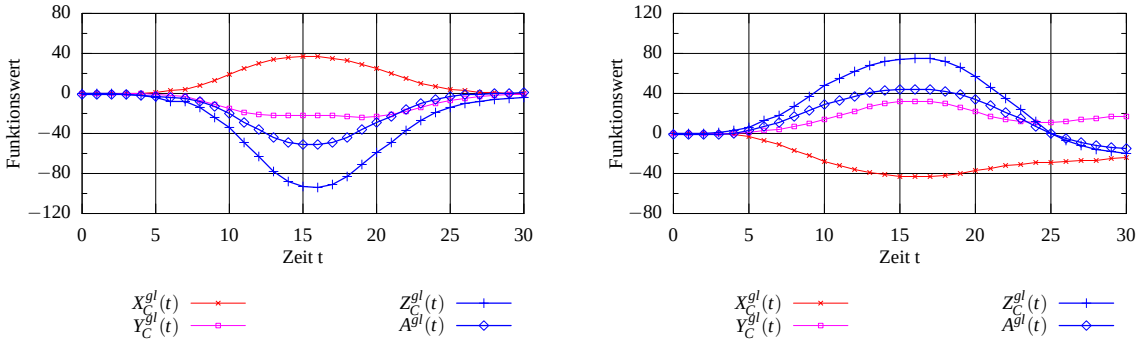


Abbildung E.1.: XYEastPointerUp-Geste (a), XYWestPointerUp-Geste (b), XYEastWave-Geste (c), XYWestWave-Geste (d). Bei allen vier Gesten ist $X_C(t)$ dominant.



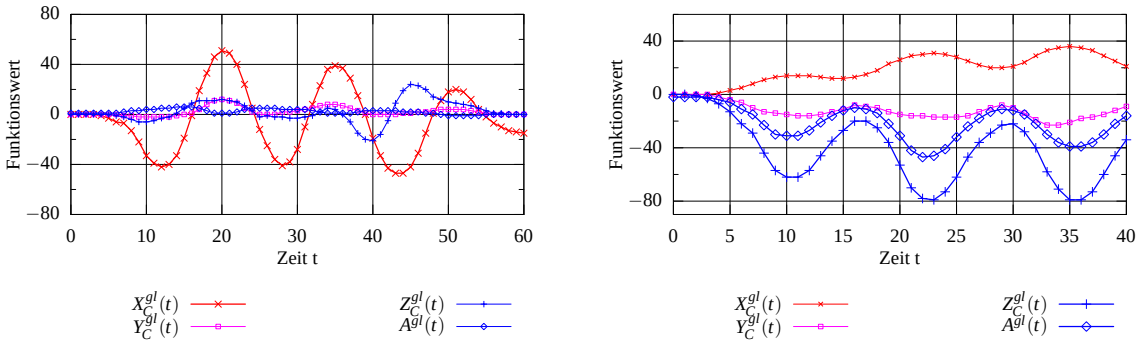
(a) (b)

Abbildung E.2.: XYNorth-Geste. $Y_C(t)$ ist dominant (a), XYSouth-Geste. $Y_C(t)$ ist dominant (b).



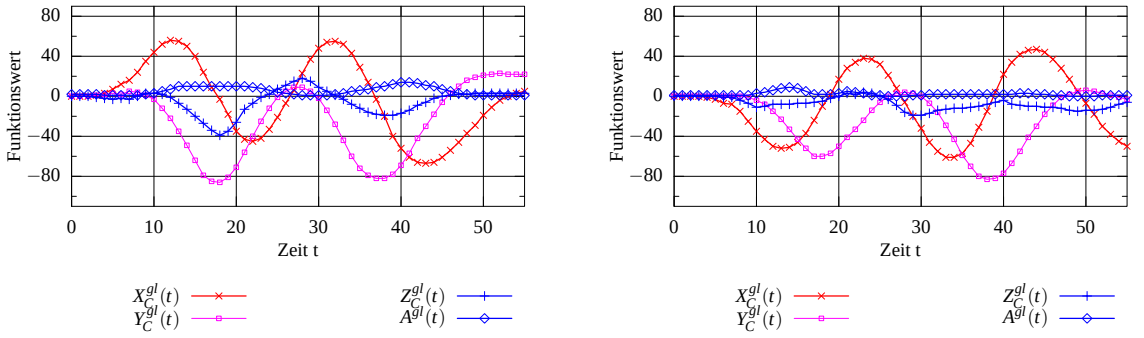
(a) (b)

Abbildung E.3.: XZNorth-Geste. $Z_C(t)$ ist dominant (a), XZSouth-Geste. $Z_C(t)$ ist dominant (b).



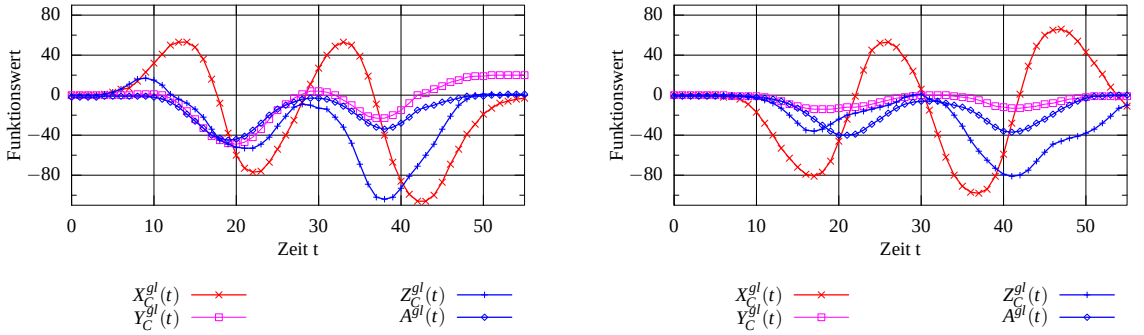
(a) (b)

Abbildung E.4.: XWipe-Geste. $X_C(t)$ ist dominant (a), ZWipe-Geste. $Z_C(t)$ ist dominant (b).



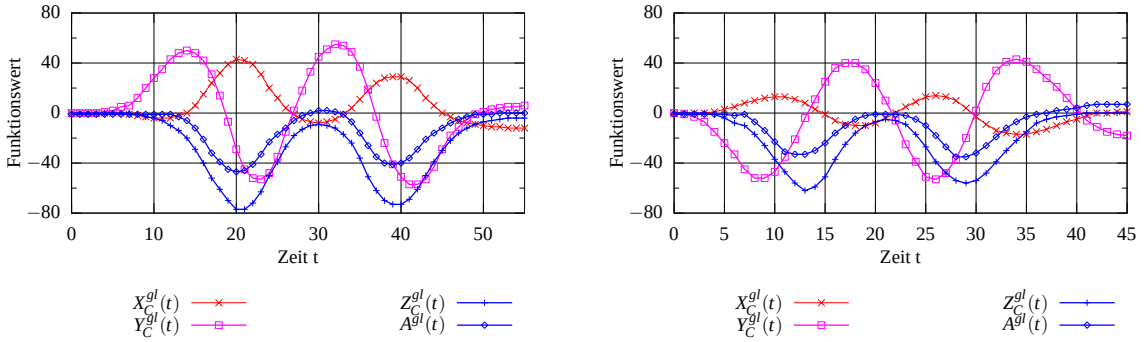
(a) (b)

Abbildung E.5.: XYCircleCW-Geste. $X_C(t)$ und $Y_C(t)$ sind dominant (a), XYCircleCCW-Geste. $X_C(t)$ und $Y_C(t)$ sind dominant (b).



(a) (b)

Abbildung E.6.: XZCircleCW-Geste. $X_C(t)$ und $Z_C(t)$ sind dominant (a), XZCircleCCW-Geste. $X_C(t)$ und $Z_C(t)$ sind dominant (b).



(a) (b)

Abbildung E.7.: YZCircleCW-Geste. $Y_C(t)$ und $Z_C(t)$ sind dominant (a), YZCircleCCW-Geste. $Y_C(t)$ und $Z_C(t)$ sind dominant (b).

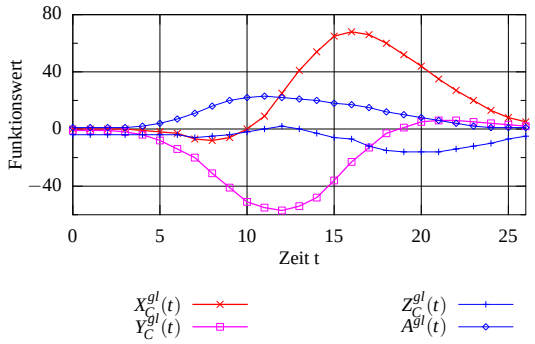


Abbildung E.8.: XYGrab-Geste: $X_C(t)$ und $Y_C(t)$ sind dominant.

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

LITERATURVERZEICHNIS

[AAM03] A. ARSENIO, P. FITZPATRICK, C. KEMP und G. METTA: *The Whole World in Your Hand: Active and Interactive Segmentation*. In: *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Seiten 49–56, Boston, MA, USA, 2003.

[Act04] ACTIVE STATE: *TCL/TK 8.0*. URL: http://www.tcl.tk/, 2004.

[Alt04] ALTHOFF, FRANK: *Ein generischer Ansatz zur Integration multimodaler Benutzereingaben*. Doktorarbeit, TU-München, Lehrstuhl für Mensch-Maschine-Kommunikation, 2004.

[AM02] ALPERN, M. und K. MINARDO: *Developing a Car Gesture Interface For Use as a Secondary Task*. Technischer Bericht, Human-Computer Interaction Institute (HCII), Carnegie Mellon University, Pittsburgh, PA 15213-3891 USA, 2002.

[AMLSA] ALTHOFF, FRANK, GREGOR MCGLAUN und MANFRED LANG: *Using Multimodal Interaction to Navigate in Arbitrary Virtual VRML Worlds*. In: *Proc. of Workshop on Perceptive User Interfaces (PUI 2001)*. ACM Digital Library: www.acm.org/uist/uist2001, November 2001, Orlando, USA. ISBN 1-58113-448-7.

[BBNB03] BOUKERROUI, DJAMAL, ATILLA BASKURT, J. ALISON NOBLE und OLIVIER BASSET: *Segmentation of ultrasound images: multiresolution 2D and 3D algorithm based on global and local statistics*. Pattern Recogn. Lett., 24(4-5):779–790, 2003.

[BT99] BIRCHFIELD, STAN und CARLO TOMASI: *Depth Discontinuities by Pixel-to-Pixel Stereo*. Int. J. Comput. Vision, 35(3):269–293, 1999.

[CBM03] COLLOBERT, R., S. BENGIO und J. MARIÉTHOZ: *Torch: A modular machine learning software library*. URL: http://www.torch.ch, 2003.

[Con04] CONRAD ELECTRONIC GMBH: *5 Watt IR-Strahler*. URL: http://www.conrad.com/, 2004.

[Cre04] CREATIVE TECHNOLOGY LTD.: *Creative Webcam NX*. URL: http://www.creative.com/, 2004.

[Dim04] DIMITRI VAN HEESCH: *Doxygen Documentation System for Programming Language*. URL: http://www.doxygen.org/, 2004.

[DSMM04] DI STEFANO, L., MARCHIONNI. M und S. MATTOCCIA: *A Fast Area-Based Stereo Matching Algorithm*. Image and Vision Computing, 22(12):983–1005, Oct 2004.

[Ekm95] EKMAN, P.: *Essential Behavioral Science of the Face and Gesture that Computer Scientists Need to Know*. In: *1st Int. Workshop on Automatic Faceand Gesture-Recognition*, Seiten 7–11, Zürich, Switzerland, 1995.

[Gei03] GEIGER, M.: *Berührungslose Bedienung von Infotainment-Systemen im Fahrzeug*. Doktorarbeit, TU-München, 2003.

[GW87] GONZALEZ, R. C. und P. WINTZ: *Digital image processing*. Addison Wesley, 1987.

[Hag97] HAGENAUER, J.: *Informationstheorie und Quellencodierung*. Vorlesungsmanuskript, Lehrstuhl für Nachrichtentechnik, Technische Universität München, 1997.

[HHT00a] HONG, PENGYU, THOMAS S. HUANG und MATTHEW TURK: *Constructing Finite State Machines for Fast Gesture Recognition*. In: *ICPR*, Seiten 3695–3698, 2000.

[HHT00b] HONG, PENGYU, THOMAS S. HUANG und MATTHEW TURK: *Gesture Modeling and Recognition Using Finite State Machines*. In: *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, Seite 410. IEEE Computer Society, 2000.

[Hu62] HU, M. K.: *Visual Pattern Recognition by Moment Invariants*. IRE Transactions in Information Theory, IT-8:179–187, 1962.

[HZ04] HARTLEY, R. I. und A. ZISSERMAN: *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, Second Auflage, 2004.

[Int03] INTEL CORP.: *OpenCV 0.9.4*. URL: http://www.intel.com/research/mrl/research/opencv/, 2003.

[Jäh93] JÄHNE, B.: *Digitale Bildverarbeitung*, Band 3. Springer Verlage, 1993.

[JZA02] J. ZIEREN, N. UNGER und AKYOL: *Hands Tracking from Frontal View for Vision-Based Gesture Recognition*. Technischer Bericht, Aachen University (RWTH), 2002.

[Kap03] KAPUSTA, ROMAN: *Scene reconstruction using structured light*. URL: http://www.cg.tuwien.ac.at/studentwork/CESCG/CESCG-2003/RKapusta/paper.pdf, 04 2003.

[Ken89] KENDON, A.: *Current Issues in the Study of Gesture*. Anthropological Study of Human Movement, 5(3):101–134, 1989.

[Ken96] KENDON, A.: *An Agenda for Gesture Studies*. Semiotic Review of Books, 7(3):8–12, 1996.

[Kim02] KIMPAN, ATTILA F.: *HMMs der Baum/Welch Algorithmus und Metastabilität in der Konformationsdynamik von Molekülen*. URL: http://www.math.fu-berlin.de/~biocomp/Lehre/SemMLMK_SS02/Vortraege/BaumWelch.pdf, 6 2002.

[KP01] KAPOOR, ASHISH und ROSALIND W. PICARD: *A real-time head nod and shake detector*. In: *Proceedings of the 2001 workshop on Percetive user interfaces*, Seiten 1–5. ACM Press, 2001.

[LA98] LUZ ABRIL, TORRES MÉNDEZ: *The Viterbi Algorithm in Text Recognition*. URL: http://www.cim.mcgill.ca/~latorres/Viterbi/va_alg.htm, 1998.

[Leh02] LEHMANN, TH. ET AL: *Bildverarbeitung für die Medizin*. Springer, 2002.

[Lev44] LEVENBERG, K.: *A Method for the Solution of Certain Problems in Least Squares*. Appl. Math., 2 Auflage, 1944.

[Lin04] LINDL, RUDI: *Spotting dynamischer Gesten im Kontext multimodaler Infotainmentsysteme*. Diplomarbeit, TU-München, 2004.

[LMA+01] LANG, MANFRED, GREGOR MCGLAUN, FRANK ALTHOFF, BJÖRN SCHULLER und KARLA GEISS: *FERMUS Phase I (Fehlerrobuste multimodale Sprachdialoge)*. Technischer Bericht, Lehrstuhl für Mensch-Maschine-Kommunikation, Juli 2001. Auftragsforschung für die BMW AG, die DaimlerChrysler AG und die Siemens-VDO Automotive AG.

[Mar63] MARQUARDT, D.: *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. Appl. Math., 11 Auflage, 1963.

[MCA02] MAMMEN, JAMES P., SUBHASIS CHAUDHURI und TUSHAR AGARWAL: *A Two Stage Scheme for Dynamic Hand Gesture Recognition*. In: *Proceedings of the National Conference on Communication (NCC 2002)*, Seiten 35–39, 2002.

[MG91] M., HARALICK R. und SHAPIRO L. G.: *Glossary of Computer Vision Terms*. Pattern Recognition, 24(1):69–93, 1991.

[ML97] MORGUET, PETER und MANFRED LANG: *A Universal HMM-Based Approach to Image Sequence Classification*. In: *Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 3*, Seite 146. IEEE Computer Society, 1997.

[ML98] MORGUET, P. und M. LANG: *Spotting Dynamic Hand Gestures in Video Image Sequences Using Hidden Markov Models*. In: *International Conference on Image Processing*. IEEE, October 1998.

[Mor00] MORGUET, P.: *Stochastische Modellierung von Bildsequenzen zur Segmentierung und Erkennung dynamischer Gesten*. Doktorarbeit, TU-München, 2000.

[NCM00] N. CARINIE, I. RICKETTS, S. MCKENNA und G. MCALLISTER: *Using finger-pointing to operate secondary controls in automobiles*. In: *IEEE Inteligent Vehicle Symposium*, Dearborn (MI), USA, October 2000. IEEE.

[Ots79] OTSU, N.: *Review on Image Segmentation Techniques*. IEEE Trans. System Man and Cybernetics, 9, 1979.

[PNR93] PAL N. R., PAL S. K.: *A Review on Image Segmentation Techniques*. Pattern Recognition, 26(9):1277–1294, 1993.

[Poi04] POINT GREY RESEARCH INC.: *Digiclops® Stereo Vision Camera System*. URL: http://www.ptgrey.com/, 2004.

[Pra91] PRATT, W. K.: *Digital Image Processing*. John Wiley and Sons, New York, 1991. 2nd edtiton.

[Que94] QUEK, F.: *Toward a vision-based hand gesture interface*. In: *Virtual reality software and technology*, Seiten 17–31, Singapore, Singapore, 1994.

[Rab89] RABINER, LAWRENCE R.: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, 77:257–286, 1989.

[SA00] S. AKYOL, U. CANZLER: *GeKomm - Gestenbasierte Mensch-Maschine Kommunikation im Fahrzeug*. Doktorarbeit, Rheinisch-Westfälische Technische Hochschule Aachen, 01 2000.

[Son] SONY: *Eye Toy (Sony Computer Entertainment Inc.)*. URL: http://www.us.playstation.com/content/ogs/scus-97319/site/.

[SP95] STARNER, T. und A. PENTLAND: *Real-Time American Sign Language Recognition From Video Using Hidden Markov Models*. In: *SCV95*, Seite 5B Systems and Applications, 1995.

[Sta94] STARNER, THAD: *Visual recognition of American Sign Language using Hidden Markov Models*. Technischer Bericht Master's Thesis, MIT, Feb 1995, Program in Media Arts & Sciences, MIT Media Laboratory, 1994.

[Ste02] STEGER, C.: *Folien zur Vorlesung Industrielle Bildverarbeitung*. URL: http://wwwradig.in.tum.de/vorlesungen, 2002.

[Tea80] TEAGUE, M. R.: *Image Analysis via the General Theory of Moments*. Journal of the Optical Society of America, 1980.

[U. 95] U. BRÖCKL-FOX: *Untersuchung neuer, gestenbasierter Methoden für die 3D Interaktion*. Doktorarbeit, Rheinisch-Westfälische Technische Hochschule Aachen, 1995.

[Vid04] VIDERE DESIGN: *STH-MDCS Stereo Megapixel Camera*. URL: http://www.videredesign.com/, 2004.

[VPH97] V. PAVLOVIC, R. SHARMA und T. HUANG: *Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):677–695, 1997.

[VS95] V. STRAT, M. OLIVEIRA: *A Point-and-Shoot Color 3D Camera*. In: *1st Int. Workshop on Automatic Faceand Gesture-Recognition*, Seiten 7–11, Zürich, Switzerland, 1995.

[WB99] WILSON, ANDREW D. und AARON F. BOBICK: *Parametric Hidden Markov Models for Gesture Recognition.* IEEE Trans. Pattern Anal. Mach. Intell., 21(9):884–900, 1999.

[WF01] WITTEN, IAN H. und EIBE FRANK: *Data Mining*. Hanser, 2001.

[WLS02] WALCHSHÄUSL, LEONHARD, RUDI LINDL und JUSTUS SCHWARTZ: *HMM-basierte Erkennung dynamischer Kopfgesten*. Interdisziplinäres Projekt im Bereich Elektrotechnik am Lehrstuhl für Mensch-Maschine-Kommunikation, 2002.

[Yon98] YONGHUA, DONG GUO: *Vision-based hand gesture recognition for human-vehicle interaction*. In: *International Conference on Control, Automation and Computer Vision*, 1998.

[ZPD+97] ZELLER, M., J. C. PHILLIPS, A. DALKE, W. HUMPHREY, K. SCHULTEN, T. S. HUANG, V. I. PAVLOVIC, Y. ZHAO, Z. LO, S. CHU und R. SHARMA: *A Visual Computing Environment for Very Large Scale Biomolecular Modeling*. In: *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures and Processors*, Seite 3. IEEE Computer Society, 1997.