# Videobasierte Verfahren zur Schätzung des Interaktionsinteresses bei der Mensch-Roboter-Interaktion mittels Analyse durch Synthese

Dissertation
Zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt der
Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von

**Dipl.-Inf. Christian Martin**

geboren am 21.07.1978 in Ilmenau

| | |
|---|---|
| Tag der Einreichung: | 27.04.2012 |
| Tag der wissenschaftlichen Aussprache: | 20.12.2012 |
| Gutachter: | 1.) Univ.-Prof. Dr.-Ing. Horst-Michael Groß |
| | 2.) Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll |
| | 3.) Prof. Dr.-Ing. habil. Hans-Joachim Böhme |

urn:nbn:de:gbv:ilm1-2012000454

## Zusammenfassung

Um den Einsatz von mobilen Servicerobotern im Alltag zu realisieren, ist es notwendig, intelligente und adaptive Dialogsysteme zu entwickeln, die es auch einem nicht-eingewiesenen Benutzer erlauben, einen Serviceroboter intuitiv bedienen und nutzen zu können. Dazu ist es erforderlich, die Stimmung und den Gemütszustand des Benutzers zu erfassen, um entsprechend darauf reagieren zu können. Im Rahmen dieser Dissertation werden Methoden entwickelt und vorgestellt, die als Indikatoren zur Schätzung des Interaktionsinteresses (bzw. der Aufmerksamkeit) eines Benutzers auf einem mobilen Serviceroboter unter Realweltbedingungen verwendet werden können.

Hierfür werden drei Teilsysteme präsentiert, die die Orientierung des Oberkörpers, die Blickrichtung und die Mimik des Benutzers schätzen können. Alle drei Teilsysteme werden mittels *Analysis-by-Synthesis* Verfahren realisiert. Dabei kommen *Active Shape Models* und *Active Appearance Models* zum Einsatz. Zur anschließenden Klassifikation bzw. Schätzung der gesuchten Merkmale werden u.a. Verfahren der linearen Regression, *Multi Layer Perceptrons*, *Support Vector Machines* und *Self-organizing Maps* miteinander verglichen. Es wird gezeigt, dass es mit den drei Teilsystemen möglich ist, die gesuchten Informationen zu bestimmen und damit Indizien für Interesse und Aufmerksamkeit gewonnen werden können. Die Tests wurden dabei jeweils mit bekanntem und unbekanntem Datenmaterial durchgeführt. Zusätzlich wird gezeigt, dass eine Vorauswahl relevanter Parameter auf Basis der *Mutual Information* zu besseren Ergebnissen führt bzw. gleich gute Ergebnisse mittels einfacherer Klassifikatoren erreicht werden können.

Weiterhin wird ein Gesamtsystem vorgestellt, in dem alle drei Teilsysteme miteinander kombiniert werden. Zur Schätzung von Interesse und Aufmerksamkeit kommen dabei Methoden aus der probabilistischen Robotik zum Einsatz. Anhand durchgeführter Experimente mit eingewiesenen Probanden wird gezeigt, dass die Ergebnisse der drei Teilmodule plausibel sind und die Resultate zur Schätzung von Interesse und Aufmerksamkeit verwendet werden können. Das prototypische Gesamtsystem kann daher als Grundlage und Basis für zukünftige sozialwissenschaftliche Untersuchungen zur Bestimmung des Interaktionsinteresses genutzt werden, die nicht Bestandteil dieser Dissertation sind.

## Abstract

To realize the operation of mobile service robots in everyday life, it is necessary to develop intelligent and adaptive dialog systems. Such dialog systems must be designed in a way that allows an easy and intuitive operation even for untrained users. For that purpose, it is necessary to detect the mood and intentions of a user. In this thesis, methods for the detection and estimation of the attention and/or interaction interest of a user of a mobile service robot will be developed and presented.

For this purpose, three subsystems are presented: the estimation of the orientation of the upper body, the estimation of the head pose, and the analysis of the facial expression of a user. Each subsystem is realized by using an *Analysis by Synthesis* approach. More precisely, *Active Shape Models* and *Active Appearance Models* are utilized within the three subsystems. Furthermore, different classification and function approximation systems will be applied to estimate the different features. For that, different methods like linear regression, *Multi Layer Perceptrons*, *Support Vector Machines*, and *Self-organizing Maps* will be compared. This thesis shows that it is possible to estimate the requested features in a sufficient quality and robustness by using the proposed subsystems. Hence it is possible, to estimate the attention and interaction interest by using the upper body orientation, the head pose and the facial expression. Each subsystem was tested with different data sets. Besides own data bases also foreign data sets were utilized to show the robustness and to measure the detection rates of the proposed methods. Additionally, this thesis shows, that a selection of the relevant model parameters leads to better results or at least to equal results, which can be achieved by easier classifiers. For this parameter selection the *Mutal Information* is applied in this thesis.

Furthermore, an overall system, which integrates the results of the different subsystems, is presented in this thesis. The fusion of the results is realized by using methods from the domain of probabilistic robotics. Based on some easy experiments (performed by briefed subjects) it is shown, that all subsystems can deliver feasible results, which can be integrated in an overall estimation of the attention and/or interaction interest of a user. Thus, the work presented in this thesis can be used for further socioscientific experiments, which are not part of this thesis.

**Danksagung**

Ich möchte an dieser Stelle die Gelegenheit nutzen, mich bei all denjenigen zu bedanken, die mich auf dem Weg zu dieser Arbeit begleitet und unterstützt haben.

An erster Stelle möchte ich hier meinem Betreuer und Leiter des Fachgebiets Neuroinformatik und Kognitive Robotik Prof. Dr. Horst-Michael Groß für die Möglichkeit danken, diese Arbeit unter seiner Regie erstellen zu können. Die vielen gemeinsamen fachlichen Diskussionen haben mir geholfen, immer wieder den roten Faden zu finden und das Gesamtwerk trotz vieler Details nicht aus den Augen zu verlieren.

Bedanken möchte ich mich auch bei den Mitarbeitern des Fachgebietes Neuroinformatik und Kognitive Robotik für die gute Arbeitsatmosphäre während meiner Zeit als Mitarbeiter am Fachgebiet. Insbesondere möchte ich mich für die fachlichen Gespräche und Diskussionen bedanken, die mir Wege aus der einen oder anderen Sackgasse aufgezeigt haben. Ein besonderer Dank geht an Ronny Stricker, Erik Schaffernicht und Erik Einhorn.

Mein Dank gilt auch den Mitarbeitern der MetraLabs GmbH, die mir oftmals als willige Testpersonen zur Verfügung gestanden haben. Ohne diese Tests hätte ich meine Ergebnisse nicht so präsentieren können.

Weiterhin möchte ich mich bei allen Studenten bedanken, die durch ihre Unterstützung zum Gelingen dieser Arbeit beigetragen haben, als da insbesondere zu nennen wären: Birthe Babies, Uwe Werner, Nils Einecke, Sebastian Belz, Ronny Stricker, Sebastian Hommel und Johannes Rühle.

Zu guter Letzt sei den wichtigsten Wegbegleitern, meiner Frau, meinen beiden Söhnen und meinen Eltern, gedankt. Sie haben mir in all den Jahren den Rücken frei gehalten und mich in meinem Schaffen bestärkt. Ganz herzlichen Dank dafür. Ein ganz besonderer Dank geht an meine Frau Kristin für das eifrige Korrekturlesen und ihr Verständnis für die vielen Abende, die ich mit der Arbeit an dieser Dissertation verbracht habe. Ein Versprechen geht an meine beiden Söhne: In Zukunft werde ich wieder mehr Zeit mit euch verbringen.

# Inhaltsverzeichnis

# 1 Einleitung

## 1.1 Mobile Serviceroboter werden zum Alltag

Im Laufe der letzten 30 Jahre hat sich das Alltagsleben im Hinblick auf technische Anwendungen und Geräte stark gewandelt. Die „Technisierung des Alltags“ beschleunigt sich zunehmend auf Grund der verfügbaren Technologien und der Leistungsfähigkeit der eingesetzten Hardware. Nachdem sich in den 1980er Jahren zunächst einfache Anwendungen (wie z.B. Geld- oder Fahrkartenautomaten) im Alltagsleben etabliert haben, bestand der nächste Schritt seit Mitte der 1990er Jahre in der Entwicklung und Verbreitung von mobilen Systemen, die das tägliche Leben erleichtern können (z.B. Mobiltelefon, PDA, Navigationssysteme).
Ein wesentliches Hauptproblem dieser Systeme, besteht darin, dass die Interaktion zwischen den technischen Systemen und dem menschlichen Benutzer sehr einseitig ist. Die Bedienung der Geräte ist, je nach Entwicklungsstand, mehr oder weniger intuitiv und erfordert vom Benutzer teilweise erheblichen Lernaufwand, bis die Systeme optimal be-/genutzt werden können. Der größte Teil der heute im Alltag eingesetzten Systeme besitzt eine mehr oder weniger starre Menüführung, die teilweise durch den Benutzer auf dessen Bedürfnisse angepasst werden kann. Ein deutlicher Fortschritt im Hinblick auf die Bedienung wurde durch die Markteinführung moderner Smartphones ab 2006 und der ersten Tablett-PCs mit Multitouch-Display im Jahr 2010 erreicht.
Neben einer intuitiven Bedienung der technischen Systeme, ist eine beidseitige Interaktion zwischen Benutzer und Gerät notwendig, um letztendlich wirklich intelligente Systeme realisieren zu können. Ein optimales System sollte in der Lage sein, sich an die Eingaben eines Benutzers anzupassen und auf Veränderungen der (momentanen) Stimmungslage reagieren zu können. Beispielsweise sollte ein Lernspiel so gestaltet sein, dass der Schwierigkeitsgrad so hoch ist, dass beim Benutzer keine Langeweile aufkommt, aber auch nicht zu hoch, da sonst der Benutzer frustriert aufgeben oder abbrechen wird. Auskunftssysteme könnten Details oder Zusatzinformationen überspringen, wenn der Benutzer offensichtlich in Eile ist.

Eine besondere Stellung im Rahmen der fortschreitenden Technisierung des Alltags haben die sogenannten *Serviceroboter*. Seit vielen Jahren arbeiten verschiedene Forschungsgruppen und Firmen weltweit an der Entwicklung von alltagstauglichen Robotersystemen, die verschiedene Service- und Dienstleistungen für Benutzer erbringen sollen. Vom Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA) wurde 1994 ein *Serviceroboter* wie folgt

definiert:

> „Ein Serviceroboter ist eine frei programmierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet. Dienstleistungen sind dabei Tätigkeiten, die nicht der direkten industriellen Erzeugung von Sachgütern, sondern der Verrichtung von Leistungen für Menschen und Einrichtungen dienen."

Erweitert wurde diese Definition 1997 von der *International Federation of Robotics* (IFR) mit dem Zusatz:

> „They are mobile or manipulative or combinations of both."

Neben einer Vielzahl von Prototypen und Forschungsprojekten gibt es bereits heute auch eine Reihe von ausgewählten Anwendungen, in denen Serviceroboter im tagtäglichen Einsatz sind.

Im Gegensatz zu den oben beispielhaft genannten technischen Systemen des Alltags besitzen Serviceroboter einen eigenen „Körper" und Gestalt und werden daher von Benutzern völlig anders wahrgenommen, als beispielsweise eine Lernsoftware. Weiterhin ermöglicht diese eigene Gestalt auch eine direkte und natürliche Interaktion zwischen Benutzer und Roboter. Doch genau diese direkte Interaktionsmöglichkeit führt zu besonders hohen Anforderungen im Hinblick auf intuitive Bedienbarkeit und Adaption an einen Benutzer, da solche Systeme bedingt durch einschlägige Science-Fiction Literatur, Film und Fernsehen als per se intelligent betrachtet werden.
Neben einiger technischer Hürden, die vor einem verstärkten Einsatz von Servicerobotern im Alltag beseitigt werden müssen, ist es daher notwendig, intelligente und adaptive Dialogsysteme zu entwickeln, die es auch einem nicht-eingewiesenen Benutzer erlauben, einen Serviceroboter intuitiv bedienen und nutzen zu können. Dazu ist es zwingend notwendig, die Stimmung und den Gemütszustand des Benutzers zu erfassen, um entsprechend darauf reagieren zu können. Zu diesem Problemkreis soll diese Dissertation einen Beitrag leisten. Generell sind die vorgestellten Methoden und Verfahren auch auf andere technische Systeme übertragbar.

Im Rahmen dieser Dissertation sollen Methoden entwickelt und vorgestellt werden, die als Indikatoren zur Schätzung des Interaktionsinteresses (bzw. der Aufmerksamkeit) eines Benutzers bei einem mobilen Serviceroboter verwendet werden können. Dabei wurden die untersuchten Methoden nach phänomenologischen Kriterien ausgewählt. Inwieweit und in welchem Maß die eingesetzten Teilsysteme auch aus psychologischer Sicht auf ein Interaktionsinteresse (oder Aufmerksamkeit) schließen lassen, erfordert eine Reihe von sozialwissenschaftlichen Untersuchungen. Diese sind jedoch nicht Bestandteil dieser Arbeit. Stattdessen sollen als Ergebnis dieser Dissertation einige notwendige Methoden und Verfahren zur Verfügung gestellt werden, um solche sozialwissenschaftlichen Experimente und Untersuchungen zukünftig zu ermöglichen. Weiterhin soll der Funktionsnachweis der Teilsysteme unter Realwelt-Bedingungen erbracht werden.

## 1.2 Soziales Verhalten in der Mensch-Roboter-Interaktion

Je mehr mobile Service- und Assistenzrobotersysteme technisch weiterentwickelt und alltagstauglicher werden, desto mehr wächst in der Forschung und Praxis das Interesse an der sozioemotionalen Ebene der Mensch-Roboter-Interaktion. Wie u.a. in [Weiss et al., 2010] [Dautenhahn, 2004] [Dautenhahn, 2007] [Syrdal et al., 2006] und [Salvini et al., 2010] gezeigt wird, ist diese ein wichtiger Faktor für die Akzeptanz von Servicerobotern durch die Nutzer. Diese erwarten im Allgemeinen ein sozialverträgliches Roboterverhalten, das auf der Basis von Erfahrungen der zwischenmenschlichen Interaktion intuitiv verstanden werden kann und gleichzeitig möglichst auch keine negativen Effekte auslöst.

Soziale Verhaltensweisen umfassen in der zwischenmenschlichen Interaktion unter anderem die Erfassung und das Verstehen von Körpersprache und Mimik sowie die entsprechenden Reaktionen darauf (z.B. eine aktive Ansprache oder eine Begrüßung bei erkanntem Interaktionsinteresse). Die Erkennung von Interesse und Aufmerksamkeit (als Teil dieser sozialen Verhaltensweisen) sind ein Teil des Verstehens von Körpersprache.

Bevor eine Interaktion beginnen kann, müssen die Teilnehmer einander Interaktionsbereitschaft signalisieren. Im Rahmen der zwischenmenschlichen Interaktion geschieht dies typischerweise über Blickkontakt [Goffman, 1971] [Argyle and Cook, 1976] [Goffman, 1963] [Goodwin, 1981], sowie über die Kopf- und Körperausrichtung [Yamazaki et al., 2007]. Dies gilt auch für die Mensch-Roboter-Interaktion: So ermutigt der Blickkontakt des Roboters die Nutzer zur Aufnahme und Aufrechterhaltung einer Interaktion [Mutlu et al., 2009] [Holthaus et al., 2010]. Umgekehrt sind der Blickkontakt einer Person und vor allem die Ausrichtung ihres Kopfes und Oberkörpers Indikatoren, anhand derer ein sozialverträglicher mobiler Assistenzroboter frühzeitig das Interaktionsinteresse eines menschlichen Gegenübers einschätzen sollte.

## 1.3 Interesse und Aufmerksamkeit in der Körpersprache

Wie im vorherigen Abschnitt beschrieben wurde, ist die im Rahmen dieser Dissertation anvisierte Schätzung des Interaktionsinteresses bei der Mensch-Roboter-Interaktion eng mit der Körpersprache in der zwischenmenschlichen Kommunikation verbunden.

Körpersprache ist eine wesentliche Komponente des zwischenmenschlichen Verhaltens. Der Begriff Körpersprache umfasst dabei sämtliche Körperbewegungen, Haltungen, Gesten, Mienen, Handlungen sowie die relative Position im Raum zueinander. Im erweiterten Sinne gehört auch der Tonfall in gewisser Weise zur Körpersprache. Als wichtiger Teil der nonverbalen Kommunikation beinhaltet die Körpersprache damit vor allem Informationen auf der Beziehungsebene.

Wie sich durch die Körpersprache, Aufmerksamkeit oder Desinteresse ausdrückt, wird beispielsweise von dem Pantomimen Samy Molcho [Molcho, 1983] und [Molcho, 2006] beschrieben:

> „Der Blick der Augen hinterlässt einen intensiven Eindruck. Wenn wir angeblickt werden, fühlen wir uns beachtet. Blickzuwendung bedeutet Aufmerksamkeit, Zuneigung und Freundlichkeit. Den Blickkontakt zu meiden signalisiert Desinteresse, Gleichgültigkeit oder auch Scham. Zu langes Anstarren hingegen wird meist als aufdringlich und aggressiv empfunden. Die Augenbewegung ist ein wichtiger Bestandteil der so genannten Mimik, dem Begriff für die Ausdrucksbewegungen des Gesichts."

oder:

> „Das Sitzen mit gestreckten Beinen zeigt Entspannung - eine Haltung, die man ungenierter in der Freizeit und beim Lagern auf ebenem Boden einnimmt. Sind dann die Knie hochgezogen und die Hände hinten aufgestützt, so bildet man eine Mauer. Umfangen dagegen die Arme die Knie und ziehen dabei den Oberkörper nach vorne wie an eine Brüstung, so ist das ein Zeichen von Gefaßtsein und konzentrierter Aufmerksamkeit."

Molcho [Molcho, 2006] beschreibt auch die Wichtigkeit und Bedeutung einiger Elemente der Körpersprache im Hinblick auf Interesse und Aufmerksamkeit bei zwischenmenschlichen Dialogen oder Präsentationen:

> „Augen: Sie spielen im Umgang mit anderen eine wesentliche Rolle. Ein direkter Augenkontakt mit offenen Augen signalisiert Aufmerksamkeit, wobei darauf geachtet werden soll, dass man das Gegenüber nicht fixiert. Sonst fühlt es sich kontrolliert. Ein kurzes Abwenden des Blickes mit gesenkten Augen zeigt, dass man über das Gesagte nachdenkt. Der Blick nach links und rechts vermittelt den Eindruck von Desinteresse."
>
> „Mund: Zusammengepresste Lippen zeigen, dass man nichts annehmen oder sagen will. Der verkniffene Mund kann auch Misstrauen ausdrücken. Ein offener Mund signalisiert Überraschung."
>
> „Kopf: Ein leicht zur Seite geneigter Kopf signalisiert Zutrauen. Ein in aufrechter Haltung zur Seite gedrehter Kopf zeigt die Interessensrichtung an: Aufmerksamkeit und Interesse wechseln in eine andere Richtung."
>
> „Schultern: Fühlt sich jemand in Gefahr, zieht er die Schultern hoch. Gerade Schultern signalisieren, dass er keine Last trägt. Wendet man die Schultern dem Gegenüber zu, deutet das auf Ablehnung hin."

Diese von Molcho gemachten Beobachtungen, Feststellungen und Schlußfolgerungen weisen darauf hin, dass Aussagen über Interesse und Aufmerksamkeit aus verschiedenen Elementen der Körpersprache abgeleitet werden können. Im Rahmen dieser Dissertation wurden daher drei Teilsysteme entwickelt, die unterschiedliche Aspekte der Körpersprache erfassen und klassifizieren sollen, um letztendlich eine Aussage über das Interaktionsinteresse eines Benutzers bzw. dessen Aufmerksamkeit ermitteln zu können.

## 1.4 Randbedingungen und Einschränkungen

Im Rahmen dieser Dissertation wurden eine Reihe von Randbedingungen und Einschränkungen festgelegt, unter denen die einzelnen Bestandteile realisiert werden sollen:

- *Einsatz auf einem mobilen Roboter*: Die entwickelten Methoden und Techniken sollen auf einem mobilen Roboter eingesetzt werden können. Damit stehen nur beschränkte Ressourcen (insb. Speicher und Rechenzeit) zur Verfügung und es muss mit teilweise stark veränderlichen Umweltbedingungen gerechnet werden.

- *Echtzeitfähigkeit*: Damit die entwickelten Systeme auch in Realwelt online eingesetzt werden können, müssen die Verfahren echtzeitfähig sein. Dabei ist zu berücksichtigen, dass auf einem mobilen Robotersystem ggf. auch noch andere Teilsysteme (z.B. für Navigation) einen Teil der vorhandenen Ressourcen benötigen.

- *Einheitliches Methodenframework*: Ein weiteres wichtiges Ziel dieser Dissertation war es, ein einheitliches Konzept aus verschiedenen Methoden (ein *Methodenframework*) für die verschiedenen Teilsysteme zu entwickeln und zu verwenden. Es sollte verhindert werden, dass eine mehr oder weniger unübersichtliche Sammlung von Verfahren entwickelt wird. Stattdessen sollen alle Teilsysteme nach einem ähnlichen Prinzip arbeiten.

- *Keine auditiven oder 3D-Informationen*: Zur Schätzung des Interaktionsinteresses können grundsätzlich verschiedene Inputmodalitäten eingesetzt werden. Beispielsweise können Stereokameras, Time-of-Flight-Kameras oder Tiefenkameras verwendet werden, um 3D-Informationen über den Interaktionspartner zu bestimmen und daraus Informationen über die Körperhaltung zu gewinnen [Shotton et al., 2011]. Auch auditive Informationen können benutzt werden, um den Interaktionspartner zu lokalisieren oder beispielsweise dessen Gemütszustand anhand des aktuellen Klangs der Stimme zu ermitteln [Brueckmann et al., 2007]. All diese Methoden und Klassen von Verfahren sollen hier jedoch nicht weiter betrachtet werden.

- *Verwendung von Standardkameras*: Im Rahmen dieser Dissertation sollen ausschließlich Videoinformationen von Standardkameras benutzt werden, um das Interaktionsinteresse zu schätzen. Andere Methoden und Techniken, die beispielsweise seit dem Erscheinen

der Kinect im Dezember 2010 rasant an Bedeutung gewonnen haben, werden in dieser Dissertation explizit nicht weiter berücksichtigt.

## 1.5 Aufbau der Arbeit

Nach dieser Einleitung werden im Kapitel 2 zwei Szenarien vorgestellt, für deren Realisierung die Bestimmung des Interaktionsinteresses beim Mensch-Roboter-Dialog sehr wichtig ist. Dabei handelt es sich um ein öffentliches Robotersystem mit einer Lotsenaufgabe und ein persönliches Robotersystem aus dem häuslichen Umfeld. Anhand dieser Beispiele werden drei relevante Teilsysteme abgeleitet und die Gesamtarchitektur und das methodische Framework beschrieben.

Bei den drei Teilsystemen handelt es sich um Module zur Schätzung der *Oberkörperpose*, der *Blickrichtung* und der *Mimik*. In den Kapiteln 3 bis 5 werden diese drei Teilsysteme näher erläutert. Dazu werden jeweils der State-of-the-Art, die Grundidee, die Funktionsweise, die Struktur des Teilsystems, theoretische Grundlagen, die Umsetzung und Ergebnisse jeweils im Hinblick auf die festgelegten Randbedingungen (siehe Abschnitt 1.4) beschrieben. Alle drei Teilsysteme werden mittels *Analysis-by-Synthesis* Verfahren realisiert. Zusätzlich wird gezeigt, dass eine Vorauswahl relevanter Merkmale auf Basis der *Mutual Information* zu besseren Ergebnissen führt bzw. gleich gute Ergebnisse mittels einfacher Klassifikatoren erreicht werden können.

Es wird gezeigt, dass es mit den drei Teilsystemen möglich ist, die gesuchten Informationen zu bestimmen und damit Indizien für Interesse und Aufmerksamkeit gewonnen werden können. Die Tests wurden dabei jeweils mit bekanntem und unbekanntem Datenmaterial durchgeführt.

Abschließend wird in Kapitel 6 ein prototypisches Gesamtsystem — bestehend aus den drei entwickelten Teilsystemen — vorgestellt. Anhand durchgeführter Experimente mit eingewiesenen Probanden wird gezeigt, dass die Ergebnisse der drei Teilmodule plausibel sind und Informationen zur Schätzung von Interesse und Aufmerksamkeit verwendet werden können. Das prototypische Gesamtsystem kann daher als Grundlage und Basis für zukünftige sozialwissenschaftliche Untersuchungen zur Bestimmung des Interaktionsinteresses genutzt werden.

Der Schwerpunkt dieser Dissertation liegt insbesondere in der Integration der Teilmodule zu einem Gesamtsystem auf einem mobilen Roboter unter Berücksichtigung der Realwelt-Anforderungen. Hierzu werden bekannte Verfahren und Methoden so erweitert, dass diese für das zu realisierende System geeignet sind.

# 2 Szenarien und Gesamtarchitektur

Im folgenden Kapitel werden die im Rahmen dieser Dissertation relevanten Beispielszenarien in den Abschnitten 2.1 und 2.2 vorgestellt. Anschließend findet sich im Abschnitt 2.3 eine Beschreibung der vorgeschlagenen Gesamtarchitektur und im Abschnitt 2.4 eine Beschreibung des eingesetzten methodischen Frameworks.

## 2.1 Shoppingroboter für den Einzelhandel

Seit 1998 hat das Fachgebiet Neuroinformatik und Kognitive Robotik der Technischen Universität Ilmenau an der Konzeption und Entwicklung eines Serviceroboters für den Einsatz im Einzelhandel als Lotsensystem (z.B. in Baumärkten, Supermärkten oder Elektronikfachmärkten) gearbeitet. Nachdem in den ersten Jahren der Forschung und Entwicklung die Grundlagen für ein solches System erarbeitet wurden, sind danach in gemeinsamen Projekten mit der Firma MetraLabs GmbH - Neue Technologien und Systeme [MetraLabs, 2011] diese Servicerobotersysteme so weit weiterentwickelt worden, dass diese seit Mitte 2008 in verschiedenen Einsatzumgebungen (u.a. einige ausgewählte Baumärkte und Elektronikfachmärkte) im Dauereinsatz sind [Böhme et al., 2006, Gross et al., 2008] (siehe auch Abb. 2.1).

Die entwickelten Serviceroboter sollen im Einzelhandel folgende Hauptaufgaben erfüllen:

- Begrüßung von Kunden im Eingangsbereich bzw. Suche nach Kunden durch regelmäßiges Patrouillieren und Anbieten der Lotsen-Dienste,
- Suche nach Artikeln oder Produktgruppen mittels einer Suchsoftware und
- Lotsen des Kunden zum Regal oder Standort mit den gewünschten Produkten.

Um diese Aufgaben erfüllen zu können, sind im Hinblick auf die Interaktion zwischen Mensch/Kunde und Roboter folgende Problemstellungen und Fragen zu lösen:

- Welche Kunden sollen angesprochen werden?
- Wann kommen Kunden nicht mehr mit der Suchsoftware zurecht bzw. wann und wie kann einem Kunden bei der Bedienung des Roboters geholfen werden?
- Folgt der Kunde während der Lotsenfahrt zum Produkt noch dem Roboter?

Zur Lösung dieser Fragestellungen muss das Robotersystem u.a. folgende Fähigkeiten besitzen:

- *PersonTracker*: Detektion und Tracking von Personen: Wo im Umfeld um den Roboter halten sich Personen auf und in welche Richtung bewegen sie sich?
- *BodyPoseTracker*: Bestimmung der Körperhaltung: Wie ist Körperhaltung der Person?
- *HeadPoseTracker*: Bestimmung der Kopfpose: Wohin schaut eine Person bzw. worauf ist die Aufmerksamkeit einer Person gerichtet?
- *MimicDetector*: Schätzung des Gesichtsausdrucks: Wie ist der aktuelle Gemütszustand eines Kommunikationspartners?

Im Abschitt 2.3 werden diese Teilmodule kurz charakterisiert und vorgestellt.

## 2.2 Roboter im häuslichen Umfeld

Seit einigen Jahren wird weltweit an verschiedenen Universitäten und Forschungsinstituten verstärkt an der Entwicklung von Verfahren und Methoden für Robotersysteme im häuslichen Bereich gearbeitet. Damit sollen perspektivisch Lösungsmöglichkeiten bzw. Systeme zur Bewältigung für die Probleme einer zukünftig immer älter werdenden Gesellschaft geschaffen werden.

Seit 2008 wird im Rahmen des Projektes *CompanionAble* [CompanionAble, 2009, Gross et al., 2011] am Fachgebiet Neuroinformatik und Kognitive Robotik an der Entwicklung von Verfahren für ein Robotersystem im häuslichen Bereich zur Unterstützung von älteren hilfsbedürftigen Personen gearbeitet. Wichtig sind hierbei kognitive Trainingsprogramme, die helfen sollen, die geistige Leistungsfähigkeit einer Betreuungsperson länger zu erhalten und somit der Person ein eigenständiges Leben im gewohnten Umfeld zu ermöglichen.

In einem weiteren Projekt *ALIAS* [ALIAS, 2010] wird an der Entwicklung eines häuslichen Robotersystems gearbeitet, dass einerseits durch Spiele und kognitive Trainingsprogramme ältere Personen länger geistig fit halten soll und andererseits gleichzeitig als Kommunikations-Terminal den Kontakt mit dem sozialen Umfeld (Familie und Freunde) aufrecht erhalten soll. Weiterhin sollen moderne Trends des World-Wide-Web (soziale Netzwerke, Cloud-Services, Bilderdienste, ...) durch eine geeignete Integration den Betreuungspersonen nutzbringend zugänglich gemacht werden.

In einem solchen häuslichen Einsatzszenario sind im Hinblick auf die Interaktion zwischen Roboter und der Betreuungsperson u.a. folgende Problemstellungen und Fragen zu lösen:

- Ist der Benutzer an einem Dialog interessiert und ist seine Aufmerksamkeit während einer Interaktion mit dem Roboter noch auf diesen gerichtet?

- Konzentriert sich der Benutzer noch auf die Aufgaben eines kognitiven Trainingsprogrammes?

- Wie ist der Gemütszustand des Benutzers? Kann durch einen gezielten Dialog z.B. Freude oder Spaß erzeugt werden?

Um diese Fragestellungen auf einem mobilen Robotersystem zu lösen, sind prinzipiell die gleichen Erkennungsmodule wie beim Shoppingroboter (siehe Abschnitt 2.1) notwendig. Unterschiedlich ist jedoch die Wichtigkeit der Module: Zur Ermittlung des Interaktionsinteresses sind für die beschriebenen Aufgaben im häuslichen Umfeld vor allem der *HeadPoseTracker* und der *MimicDetector* relevant. Der *PersonTracker* und *BodyPoseTracker* spielen z.B. bei einer kognitiven Trainingsaufgabe nur eine untergeordnete Rolle, da man davon ausgehen kann, dass sich die Betreuungsperson dabei die meiste Zeit unmittelbar vor dem Display des Roboters aufhalten wird.

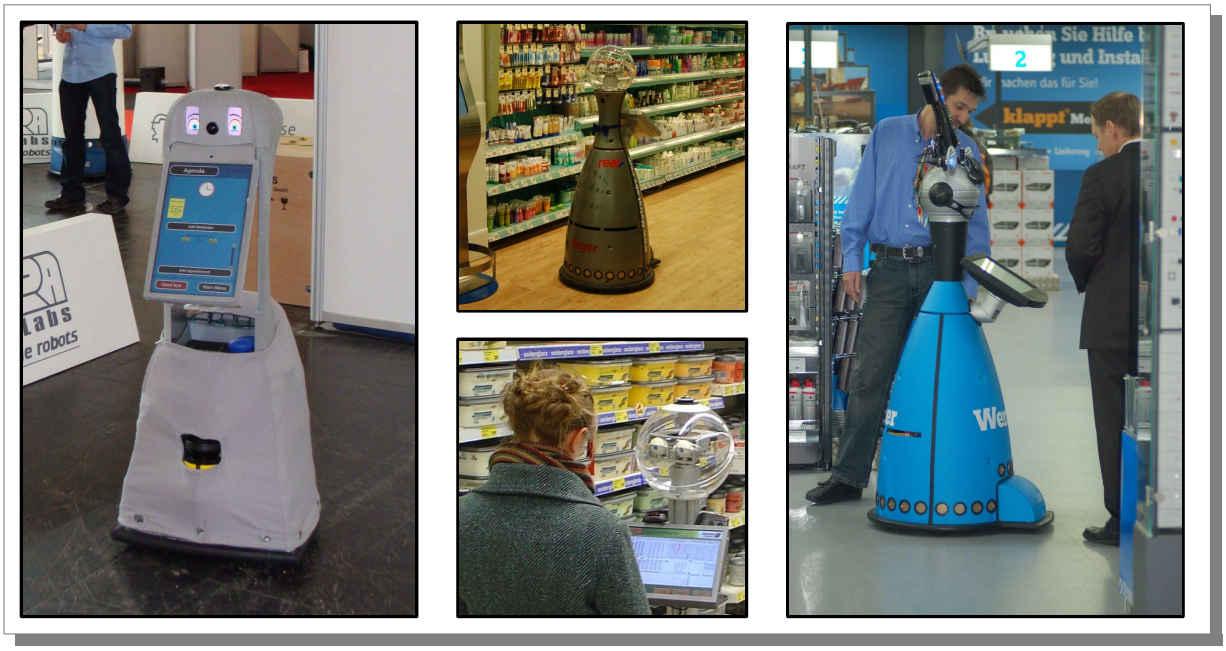


***Abbildung 2.1:*** *Verschiedene Einsatzszenarien eines mobilen Serviceroboters:*
*Links: Prototyp des Projektes CompanionAble,*
*Mitte-oben: Roboter im Supermarkt,*
*Mitte-unten: Roboter in einem Baumarkt,*
*Rechts: Roboter in einem Elektronikfachmarkt.*

## 2.3 Teilmodule und Systemstruktur zur Schätzung von Interaktionsinteresse auf einem mobilen Robotersystem

### Teilmodule zur Schätzung des Interaktionsinteresses

Um die im vorangegangenen Abschnitt benannten Probleme der beiden Beispielanwendungen unter Realweltbedingungen zu lösen, muss ein Robotersystem mit einer Reihe von Erkennungsmodulen ausgestattet sein, die im Folgenden kurz charakterisiert werden sollen:

- *PersonTracker:* Um auf Personen (Kunden, potenzielle Nutzer, Betreuungsperson, ...) reagieren zu können, muss ein Roboter in der Lage sein, die in seiner unmittelbaren Umgebung befindlichen Personen zu detektieren. Neben der bloßen räumlichen Erfassung der Personen (z.B. in Form von Richtung und Entfernung) sollte der PersonTracker auch eine zeitliche Verknüpfung der Detektionen durchführen, um somit eine Aussage über die Bewegung der Personen (z.B. in Form von Bewegungstrajektorien) treffen zu können.
  Zur Realisierung eines solchen *PersonTracker-Moduls* können verschiedene Sensoriken des Robotersystems eingesetzt werden. Hauptsächlich werden Kameras (omnidirektional und/oder frontal) und Entfernungsmesser (Laser, Ultraschall) verwendet [Martin et al., 2005a, Müller et al., 2008].

- *BodyPoseTracker:* In der zwischenmenschlichen Kommunikation spielt die Körpersprache eine sehr wichtige Rolle (siehe Abschnitte 1.2 und 1.3). Beispielsweise drückt man durch das Abwenden des Oberkörpers von seinem Gesprächspartner eine Form von Desinteresse aus. Durch ein Hinwenden zeigt man sich interessiert oder neugierig. Ein Serviceroboter kann somit durch die Erfassung der Orientierung des Oberkörpers einer Person sehr schnell erkennen, ob die Person mit dem Roboter kommunizieren möchte (z.B. bei der Kontaktaufnahme) oder auf Grund eines langwierigen oder komplizierten Dialogs die Kommunikation abbrechen wird (Kontaktabbruch).
  Die Orientierung des Oberkörpers einer Person kann am besten mit einem entsprechend ausgerichteten Weitwinkel-Kamerasystem erfasst werden. Auch die Bilddaten einer omnidirektionalen Kamera können hierzu verwendet werden.

- *HeadPoseTracker:* Neben der Pose des (Ober-)Körpers spielt auch die Pose des Kopfes eine wichtige Rolle in der zwischenmenschlichen Kommunikation (siehe Abschnitte 1.2 und 1.3). Ähnlich wie bei der Oberkörperorientierung können hieraus Informationen über das Interesse an der Kommunikation mit dem Kommunikationspartner gewonnen werden.
  Durch ein regelmäßiges oder dauerhaftes Abwenden des Blickes vom Bildschirm des Roboters kann bei der Durchführung einer kognitiven Trainingsanwendung auf ein Desinteresse beim Benutzer geschlossen werden.
  Prinzipiell muss zwischen der Kopfpose (*head gaze*) und der Blickrichtung (*eye gaze*)

unterschieden werden, da Menschen die Augen unabhängig vom Kopf bewegen können. In der Praxis ist für die Abschätzung von Interesse und Aufmerksamkeit jedoch die Bestimmung der Kopfpose ausreichend, da Menschen bei einem Dialog die Augen nur in einem relativ schmalen Bereich bewegen und bei größeren Blickrichtungsänderungen den Kopf nachführen.
Zur Erkennung der Kopfpose wird typischerweise ein Bild einer Frontalkamera verwendet. Auch hier ist der Einsatz einer omnidirektionalen Kamera denkbar.

- *MimicDetector:* Eine sehr wichtige Komponente zur Abschätzung des Interaktionsinteresses ist der Gesichtsausdruck des Kommunikationspartners. Mit Hilfe des Gesichtsausdrucks kann der emotionale Zustand einer Person sehr gut abgeschätzt werden.
Die Bestimmung des Gesichtsausdrucks einer Person erfordert ein gutes Bild des Kopfes bzw. des Gesichts. Hierzu wird meist eine Kamera eingesetzt, die frontal auf den Nutzer ausgerichtet ist und ein Bild mit entsprechend hoher Auflösung liefern kann.

Im Rahmen dieser vier Teilmodule hat das *PersonTracker-Modul* eine besondere Eigenschaft. Es kann auf zwei verschiedene Varianten eingesetzt werden:

(a) Einsatz als Input: Die anderen drei Teilmodule verwenden den Output des *PersonTrackers* als Input für eine Initialisierung und/oder Startschätzung. Der PersonTracker steht somit am Anfang der Verarbeitungskette.

(b) Einsatz als Integrator: Die Ergebnisse der drei Module *BodyPoseTracker*, *HeadPoseTracker* und *MimicDetector* werden als Input-Modalitäten für den Tracker eingesetzt. Der PersonTracker steht bei dieser Variante am Ende der Verarbeitungskette.

Im Rahmen dieser Dissertation wird der PersonTracker auf Basis der Variante (a) eingesetzt: Der Tracker soll als Input zur Initialisierung der anderen Module genutzt werden können. Im Folgenden wird ein solches *PersonTracker-Modul* als vorhanden angenommen und wird in dieser Arbeit nicht weiter betrachtet werden.

#### Systemstruktur - Integration der Teilmodule

Die drei Teilmodule *BodyPoseTracker*, *HeadPoseTracker* und *MimicDetector* können als Stufen in einem Verfeinerungsprozeß angesehen werden: Solange sich der Benutzer nicht dem Roboter zugewandt hat, macht die Bestimmung der Kopfpose und des Gesichtsausdrucks keinen Sinn bzw. ist teilweise auch nicht möglich. Je nach Situation liefern keines, nur eines oder mehrere Module eine Information über Interesse und Aufmerksamkeit des Benutzers. Auf Grund möglicher Fehldetektionen soll das System nicht streng hierarchisch konzipiert werden, sondern es soll zwischen den einzelnen Stufen es eine gewisse Überlappung geben. Die Ergebnisse der drei Teilmodule können in einem nachgeschalteten Modul zu einer Gesamtaussage fusioniert werden. Als Ergebnis soll es mit dieser möglich sein, die

Dialogansteuerung so anzupassen, dass das eigentliche Ziel der Anwendung (z.B. Kunde ist zufrieden mit der Artikelsuche oder Finden des passenden Schwierigkeitsgrads bei einer kognitiven Trainingsanwendung) erreicht werden kann. Dieser letzte Schritt ist stark anwendungsabhängig und soll im Rahmen dieser Dissertation nicht untersucht werden.

Abbildung 2.2 zeigt die vorgeschlagene Gesamtstruktur der genannten Teilmodule.

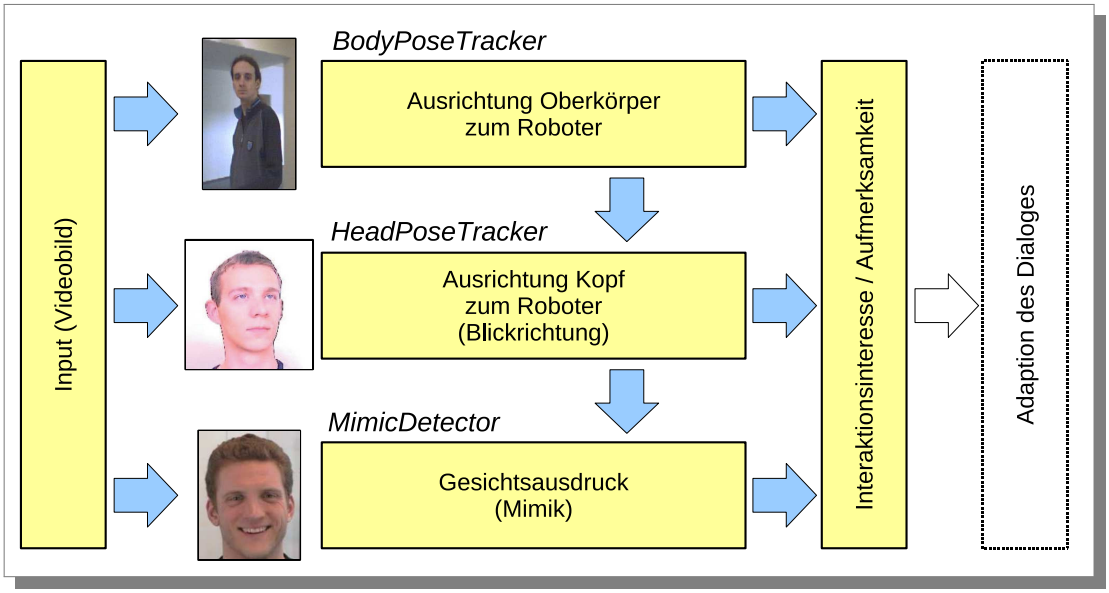



***Abbildung 2.2:*** *Gesamtstruktur der Teilmodule zur Schätzung von Interaktionsinteresse und Aufmerksamkeit beim Mensch-Roboter-Dialog: In einem ersten Teilsystem wird der Oberkörper (insbesondere die Kopf-Schulter-Kontur) des Interaktionspartners und dessen Ausrichtung bestimmt. Wenn der Nutzer in Interaktion mit dem Roboter tritt, kann bei einem Dialog die Blickrichtung (Kopfpose) ermittelt werden. Solange der Nutzer seine Aufmerksamkeit direkt auf den Roboter richtet, kann dessen Gesichtsausdruck (Mimik) bestimmt werden. Alle drei Teilsysteme können einzeln oder gemeinsam benutzt werden, um das Interaktionsinteresse bzw. die Aufmerksamkeit des Nutzers zu ermitteln. In einem optional nachgeschalteten System kann dann die Dialogführung entsprechend den Zielen des Dialogs angepasst werden.*

Die drei Teilmodule haben unterschiedliche Anforderungen an die Bildauflösung. Daher kann nicht jedes der Module bei beliebigem Abstand zwischen Benutzer und Roboter zum Einsatz kommen. Abbildung 2.3 veranschaulicht drei Bereiche:

- Befindet sich der Nutzer unmittelbar vor dem Roboter (Abstand $< 0.5$m), kann ein hochaufgelöstes Bild des Gesichts für den *MimicDetector* und *HeadPoseTracker* auf-

genommen werden. Der *BodyPoseTracker* kann nicht zum Einsatz kommen, da kein geeignetes Bild des gesamten Oberkörpers aufgenommen werden kann.

- Im mittleren Bereich (Abstand 0.5m...1.5m) ist die Auflösung zur Mimikdetektion typischerweise bereits zu klein. Der HeadPoseTracker kann je nach Kameraauflösung noch eingesetzt werden.
- Im Fernbereich (Abstand > 1.5m) ist die Auflösung zur Bestimmung der Mimik und Kopfpose zu klein. Stattdessen kann dann die Oberkörperpose geschätzt werden, da der gesamte Oberkörper im Bild sichtbar ist.

Beim Einsatz von "normalen" Kameras kann, je nach eingesetztem Objektiv, ein sinnvoller Erfassungsbereich von bis zu ca. 60-90°realisiert werden. Bei größeren Winkeln werden die Verzerrungen im Randbereich sehr stark und erschweren die Analyse. Beim Einsatz einer omnidirektionalen Kamera für den *BodyPoseTracker* können dagegen volle 360°abgedeckt werden.

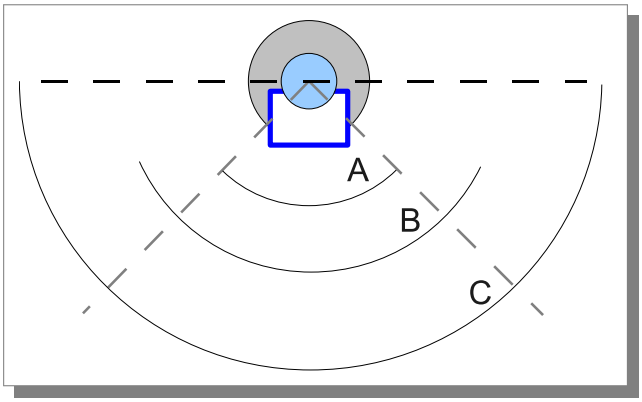



***Abbildung 2.3:*** *Arbeitsbereiche der Teilmodule zur Schätzung von Interaktionsinteresse und Aufmerksamkeit beim Mensch-Roboter-Dialog. Draufsicht auf einen Roboter: Im Nahbereich (A) können der HeadPoseTracker und vor allem der MimicDetector genutzt werden. Im mittleren Bereich (B) kommen der HeadPoseTracker und je nach Auflösung der BodyPoseTracker zum Einsatz. Im Fernbereich (C) wird ausschließlich der BodyPoseTracker benutzt.*

## 2.4 Methodisches Framework

Ein Ziel dieser Dissertation ist es, bei der Realisierung der drei Teilmodule möglichst ähnliche Methoden und Verfahren einzusetzen. Im Bereich der Mustererkennung auf Bildern oder Videodaten lassen sich allgemein vier große Gruppen von Verfahren differenzieren, in die Methoden zur Realisierung der drei Teilmodule eingeordnet werden können:

- **Bildbasierte Verfahren:** Die bildbasierten (*appearance-based*) Verfahren beruhen auf der direkten Verarbeitung der Farb- oder Grauwertinformationen des Bildes. Ein Vorteil dieser Verfahren ist, dass sie keine Initialisierung benötigen, sondern direkt auf einem Input angewendet werden können. Des Weiteren sind sie zeitunabhängig und können oftmals auch auf Bildern mit geringer Auflösung angewendet werden. Nachteilig ist jedoch die typischerweise sehr hohe Dimensionalität der zu verarbeitenden Daten.

- **Merkmalsbasierte Verfahren:** Bei den merkmalsbasierten (*feature-based*) Ansätzen werden Merkmale mit Hilfe entsprechender Detektoren oder Filter aus dem Bild extrahiert. Dadurch erreichen diese Verfahren eine erhebliche Reduzierung der zu verarbeitenden Daten gegenüber den bildbasierten Verfahren. Weiterhin kann, abhängig von den genutzten Daten, eine größere Robustheit gegenüber Beleuchtungsänderungen oder kleineren Änderungen in der Perspektive erreicht werden. Nachteilig ist, dass durch die Informationsreduktion Informationen, z.B. bezüglich der Textur, verloren gehen können.

- **Templatebasierte Verfahren:** Bei den templatebasierten (*template-based*) Ansätzen werden in einer vorgeschalteten Phase der Modellerstellung verschiedene Ansichten der relevanten Objektklasse aufgezeichnet oder generiert. Diese *Templates* werden anschließend in der Anwendungsphase zur Suche und zum Abgleich der Bilddaten auf dem aktuellen Input verwendet. Nachteilig bei diesen Verfahren sind die teilweise großen Datenmengen und oftmals auch eine zeitaufwendige Suche in der Anwendungsphase. Ein Vorteil der templatebasierten Verfahren sind die typischerweise hohen Erkennungsraten.

- **Modellbasierte Verfahren:** Bei den modellbasierten (*model-based*) Ansätzen werden in einer vorgelagerten Phase der Modellerstellung geometrische und/oder texturelle Eigenschaften der relevanten Objektklasse aus einer Reihe von (manuell) gelabelten Daten extrahiert. Diese Informationen werden anschließend in ein parametrisches Modell integriert. In der Anwendungsphase wird eine geschätzte Modellinstanz mit der aktuellen Eingabe verglichen. Durch die Anpassung der Modellparameter wird versucht, eine möglichst gute Übereinstimmung zwischen der Hypothese und der aktuellen Beobachtung zu erzeugen: *"Analysis by Synthesis"* (siehe Abbildung 2.4). Die so ermittelten Modellparameter können in einer weiteren Verarbeitungsstufe anschließend zur Gewinnung von Aussagen über den unbekannten Input genutzt werden. Wie bei den merkmalsbasierten Verfahren, findet hier also auch eine erhebliche Dimensionsreduzierung statt. Nachteilig ist, dass diese Verfahren typischerweise eine grobe Initialisierung benötigen.

Nicht bei allen Ansätzen kann eine exakte Zuordnung zu einer der vier Gruppen vorgenommen werden. Oftmals kombinieren konkrete Ansätze einzelne (Teil-)Verfahren der

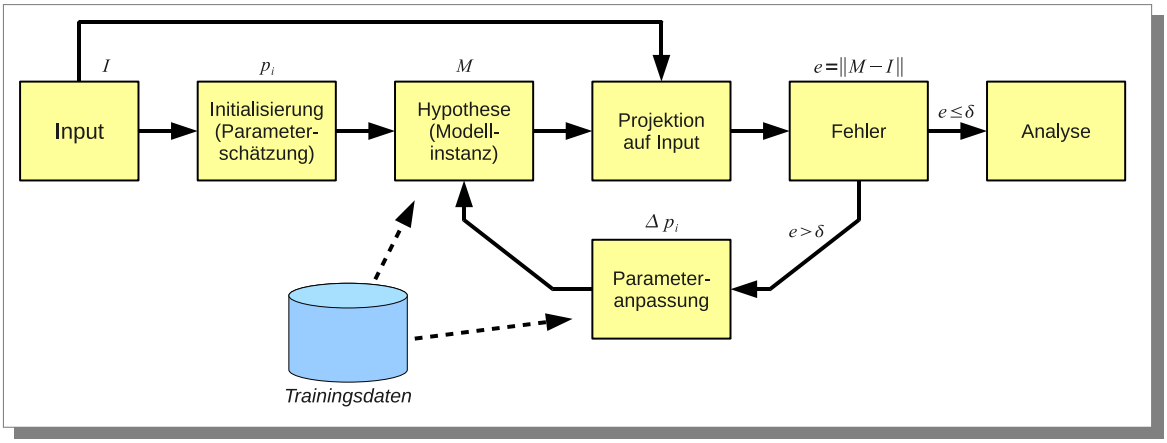



***Abbildung 2.4:*** *Das Prinzip "Analyse durch Synthese": Auf einen unbekannten Input $I$ erfolgt eine Initialisierung und eine erste Schätzung der Modellparameter $p_i$. Die so erzeugte Modellinstanz $M$ wird mit dem aktuellen Input verglichen. Wenn der Fehler $e$ größer als eine festgelegte Schwelle $\delta$ ist, erfolgt eine Parameteranpassung um $\Delta p_i$. Wenn der Fehler klein genug ist, kann die weitere Analyse des Inputs basierend auf den $p_i$ erfolgen.*

verschiedenen Gruppen. Typisch ist beispielsweise die Verwendung von merkmalsbasierten Ansätzen im Rahmen von templatebasierten Verfahren (Merkmale werden zur Beschreibung und Lokalisierung der Templates genutzt) und modellbasierten Verfahren (Merkmale werden zur Erstellung und Anpassung eines Modells genutzt).

In den nachfolgenden Kapiteln, die die drei Teilmodule beschreiben, werden jeweils Methoden zum State-of-the-Art entsprechend dieser Systematisierung vorgestellt.

Ein entscheidender Vorteil der modellbasierten Verfahren ist die Tatsache, dass durch die eingesetzte Dimensionsreduzierung die resultierenden Modelle oft mit vergleichsweise weniger Parametern auskommen und trotzdem in der Lage sind, einen großen Inputraum abzudecken. Gleichzeitig können die ermittelten Modellparameter direkt für eine weitergehende Analyse eines Inputs eingesetzt werden. Der kontinuierliche Parameterraum erlaubt weiterhin eine kontinuierliche Analyse des Input im Gegensatz zu bspw. einer diskreten Anzahl fester Templates mit zugehörigen Eigenschaftswerten.

In der vorliegenden Dissertation werden daher vorrangig modellbasierte Verfahren eingesetzt. Zur Analyse der Oberkörperpose werden *Active Shape Modelle (ASM)* verwendet. Die Bestimmung der Kopfpose und des Gesichtsausdrucks erfolgt auf der Basis von *Active Appearance Modellen (AAM)*, die letztendlich eine Weiterentwicklung der ASM sind. In den Teilmodulen werden diese Verfahren jeweils aufgabenspezifisch eingesetzt und mit entspre-

chenden anderen Methoden zur Vor- und Nachverarbeitung und zur Analyse erweitert.

Als weitere methodische Gemeinsamkeit wird in allen Teilsystemen eine problemorientierte Merkmalsauswahl auf Basis der *Mutual Information* durchgeführt, die zu einer erheblichen Reduktion der relevanten Modellparameter (und damit einer Vereinfachung) für eine anschließende Klassifikation und/oder Funktionsapproximation führt.

Abbildung 2.5 zeigt die vorgeschlagene Gesamtarchitektur auf Basis der "Analyse durch Synthese", die im Rahmen dieser Dissertation realisiert werden soll.

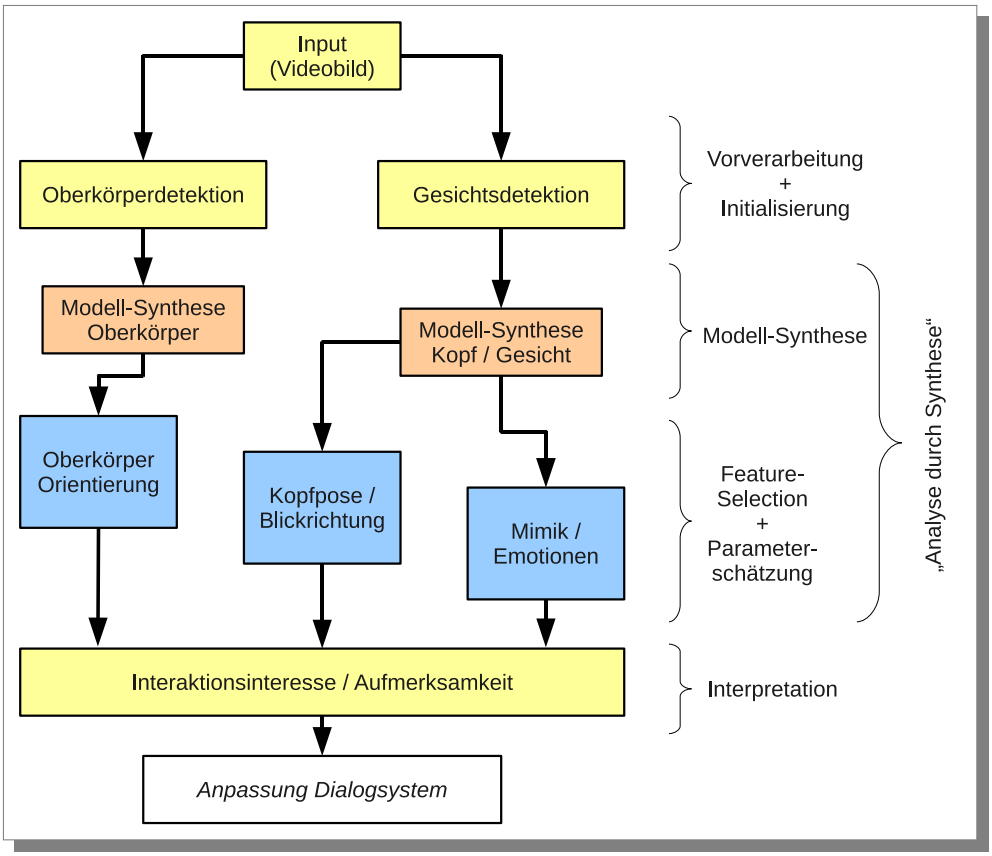
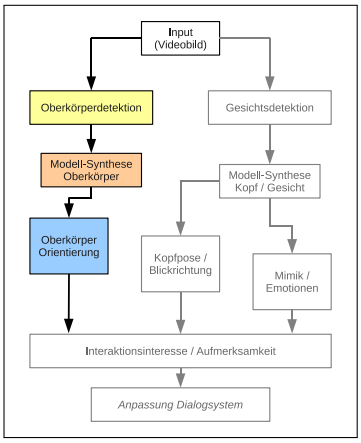



***Abbildung 2.5:*** *Architektur Gesamtsystem: Der Inputdatenstrom wird bei allen drei Teilsystem zunächst einer Vorverarbeitung unterzogen. Darauf basierend erfolgt die iterative Synthese eines Modells, aus dessen Parametern die gesuchten Größen geschätzt werden können. Alle drei Teilergebnisse werden abschließend zu einer Gesamtaussage fusioniert.*

In den nachfolgenden Kapiteln werden die drei auf Basis dieses Frameworks realisierten Teilsysteme *BodyPoseTracker*, *HeadPoseTracker* und *MimicDetector* ausführlich vorgestellt und anhand durchgeführter Realwelt-Experimente bewertet.

# 3 Teilsystem 1: Oberkörperpose

## 3.1 Einleitung




In der zwischenmenschlichen Kommunikation kann die Pose des Oberkörpers einer Person u.a. Aufschluss über deren Aufmerksamkeit, deren Intentionen und die innere Einstellung zu einer anderen Person oder einer Sache geben. Um eine natürliche Interaktion zwischen Mensch und Roboter zu ermöglichen, ist die Erkennung der Oberkörperpose daher unerlässlich. Unter realen Einsatzbedingungen ist diese Erkennungsaufgabe auf Grund der Vielzahl der möglichen Posen, der Kleidung, des Hintergrundes und variabler Beleuchtungsbedingungen keine leicht zu lösende Aufgabe.

Um einen Erstkontakt zu einer Person im Umfeld eines mobilen Robotersystems herzustellen, muss das Robotersystem die Person(en) zunächst detektieren. Hierzu kann ein *PersonTracker* eingesetzt werden, wie er beispielsweise in [Martin et al., 2005a] oder [Müller et al., 2008] vorgestellt wird. Ein solcher liefert eine Menge von Hypothesen über die Positionen (und möglicherweise auch die Entfernung und die Bewegungsrichtung) der Personen im Bild. Auf Basis dieser Hypothesen, könnte der Roboter eine Person dann ansprechen, wenn diese sich eine gewisse Zeitspanne vor dem Roboter aufhält. Da ein reiner *PersonTracker* jedoch typischerweise keine Informationen über die Pose der Person bestimmt, würde der Roboter auch eine Person ansprechen, die mit dem Rücken zum Roboter steht oder eine seitlich stehende Person, die mit einer zweiten ein Gespräch führt. Je nach konkreter Anwendung und Szenario kann dies sinnvoll oder eher kontraproduktiv sein.
Um zu bestimmen, ob eine Person in Interaktion mit dem Roboter treten möchte, ist es notwendig, die Oberkörperpose der Person zu bestimmen: Eine Person, die direkt frontal vor dem Roboter steht, richtet ihre Aufmerksamkeit mit hoher Wahrscheinlichkeit auf den Roboter. Eine Person, die nur in seitlicher Ansicht zu sehen ist, richtet die Aufmerksamkeit eher weniger auf den Roboter.

Das folgende Kapitel beschreibt den im Rahmen dieser Dissertation realisierten *BodyPoseTracker*, der zur Bestimmung der Orientierung des Oberkörpers einer vor dem

Roboter stehenden Person eingesetzt werden soll. Auf Basis dieser Orientierung kann auch eine gewisse Aussage über die Aufmerksamkeit bzw. das Interaktionsinteresse der Person getroffen werden. Inwieweit die Ausrichtung des Oberkörpers zum Roboter auch aus psychologischer Sicht ein Maß für das Interaktionsinteresse ist, erfordert verschiedene sozialwissenschaftliche Experimente, die nicht Bestandteil dieser Dissertation sind.

Im Folgenden werden in diesem Kapitel zunächst einige ausgewählte verwandte Verfahren aus der Literatur vorgestellt und bewertet. Im Anschluss werden die Struktur dieses Teilmoduls und die einzelnen Bestandteile genauer erläutert. Das Kapitel endet mit der Präsentation der erzielten Ergebnisse.

## 3.2 State-of-the-Art

Verfahren zur visuellen Bestimmung oder Schätzung der Pose einer Person können im Wesentlichen in vier verschiedene Gruppen (siehe Abschnitt 2.3) eingeteilt werden:

- *Appearance-based* Verfahren
  Beispiele: [Stiene, 2005], [Hanek and Beetz, 2004], [Panin et al., 2006], [Andriluka et al., 2009]
- *Feature-based* Verfahren
  Beispiele: [Taylor, 2000], [Ferrari et al., 2009]
- *Template-based* Verfahren
  Beispiele: [Wu and Nevatia, 2005], [Gavrila and Munder, 2007], [Li et al., 2010], [Dimitrijevic et al., 2005]
- *Model-based* Verfahren
  Beispiele: [Lee and Cohen, 2004], [Treptow et al., 2005], [Schmidt et al., 2006]

Ein sehr großer Teil der bekannten Arbeiten konzentriert sich dabei auf die Erkennung des ganzen Körpers oder des Oberkörpers und der entsprechenden Gliedmaßen (z.B. zur Erkennung von Zeigegesten [Martin et al., 2010]). Die Erkennung der Oberkörperpose (genauer gesagt: die Orientierung des Oberkörpers) wird in relativ wenigen Publikationen behandelt. Jedoch gerade bei der Mensch-Roboter-Interaktion spielt genau dieses Problem eine entscheidende Rolle. Auf Grund des typischen Erfassungsbereiches einer auf dem Roboter integrierten Kamera und des Abstands zwischen Mensch und Roboter beim Dialog ist in den meisten Fällen auch nur der Oberkörper sichtbar und relevant.

Die meisten vorhandenen Verfahren arbeiten auf Basis von 2D Bildern. Da diese jedoch nur eine Projektion der 3D Welt darstellen, sind hierbei Probleme auf Grund von Mehrdeutigkeiten und Selbstverdeckungen zu erwarten. Ein Ziel dieser Dissertation ist aber gerade

die Realisierung ohne Stereokamerasysteme (siehe Abschnitt 1.4), deshalb werden diese Probleme in Kauf genommen und auf Tiefeninformationen basierende Ansätze nicht näher betrachtet.

Jedes Verfahren zur Oberkörperdetektion benötigt verschiedene Merkmale, anhand derer ein Oberkörper in einem Kamerabild erfasst werden kann. Im Abschnitt 3.2.1 werden daher eine Reihe relevanter Merkmale beschrieben. Anschließend werden im Abschnitt 3.2.2 verschiedene Verfahren zum Konturtracking vorgestellt, mit denen die Detektion eines Oberkörpers möglich ist. Dabei liegt der Schwerpunkt der recherchierten Verfahren und Features auf einer möglichst effizienten Verfolgung und Beschreibung der Kontur in Echtzeit, um das resultierende Konturtrackingsystem auf einem mobilen Roboter mit beschränkter Rechenkapazität einsetzen zu können.

### 3.2.1 Konturbeschreibende Bildmerkmale

Mittels einer, ein beliebiges Objekt beschreibenden Kontur, wird eine Region des Eingangsbildes segmentiert. Allgemein werden Konturen als eine Kette von interessanten Punkten erfasst, die neben den eigentlichen Konturkoordinaten lokale Bildmerkmale beschreiben. Im folgenden Abschnitt werden daher verschiedene Bildmerkmale bezüglich ihrer Konturbeschreibungsfähigkeit verglichen. Dabei wird auf Grund der extrem hohen Anzahl von bildbeschreibenden Merkmalen nur eine Auswahl vorgestellt, welche in aktuellen Veröffentlichungen zum Einsatz kommt. Komplexe und rechenintensive Merkmale wie sie beispielsweise von [Tsoligkas and Xu, 2007] zur Videokompression verwendet werden, stehen nicht im Vordergrund.

Das wohl am häufigsten verwendete Merkmal ist die Beschreibung von Konturen durch Grauwertgradienten, erzeugt durch Faltung des Eingangsbildes mit linearen Operatoren in x- bzw. y-Richtung. So verwenden [Wu and Nevatia, 2005] und [Cootes and Taylor, 2001] Sobelfilter zur Erzeugung des konturbeschreibenden Grauwertgradientenbetrags. Ist die geometrische Ausrichtung des zu beschreibenden Konturobjekts bekannt, so ist zusätzlich die Gradientenorientierung ein interessantes Merkmal, welches beispielsweise mit der Orientierung eines Konturmodells [Treptow et al., 2005] verglichen werden kann. Der größte Nachteil von Grauwertgradienten ist ihre große Abhängigkeit von Beleuchtungsschwankungen und der hohe Bedarf an Rechenzeit für die nötigen Faltungsoperationen bei der Verwendung größerer Operatoren. Zusätzlich weisen Grauwertgradienten eine hohe Unspezifität in Realwelt-Szenarien auf, d.h. ein stark strukturierter Hintergrund liefert eine hohe Anzahl von Gradienten, die eine Personenkontur verrauschen können.

Ein weiteres Merkmal, welches auf Lumineszenzkanten aufbaut, ist das FAST-Feature [Rosten and Drummond, 2005], welches mit einem Harris Operator [Deparis, 2004] ver-

gleichbare Grauwertecken beschreibt. Zusätzlich lässt sich dieses Merkmal mittels ID3-Algorithmus [Quinlan, 1986] optimieren. Dieses Merkmal wird bei der Verfolgung von Konturen mit hohem Anteil lokaler Konturorientierung im Bereich $\in [-180, -90, 0, 90, 180]$ als optimal beschrieben [Panin et al., 2006], ist aber als Merkmal von kreisförmigen oder ellipsoiden Objektkonturen nicht geeignet. Alternativ lassen sich nach [Rosten and Drummond, 2006] Eckenmerkmale auch durch Einsatz eines Multi-Layer Perceptrons (MLP) gewinnen, welches auf Harris-Merkmalen eines Inputbildes trainiert wurde.

| Merkmal | Vorteile | Nachteile |
|---|---|---|
| Gradientenbetrag | • effiziente Berechnung mit Differenzoperator möglich<br>• weite Verbreitung in aktueller Forschung<br>• schnellstes Verfahren | • rauschempfindlich<br>• empfindlich gegen Helligkeitsschwankungen<br>• kleine Filtermaske erfasst nur scharfe Kanten |
| Gradienten-orientierung | • effiziente Berechnung durch Look-Up-Table<br>• mit Orientierung der Kontur vergleichbar | • stark rauschempfindlich<br>• empfindlich gegen Helligkeitsschwankungen |
| Canny-Filter | • effiziente Berechnung möglich<br>• sehr leistungsfähig | • Ergebnis und Rechenzeit abhängig von gewählten Parametern |
| FAST | • sehr schnell<br>• auf Anwendung optimierbarer ID3-Algorithmus | • schlechte Beschreibung von runden Konturabschnitten<br>• empfindlich gegen Helligkeitsschwankungen |
| Harris Operator | • optimale Konturbeschreibung<br>• reduziert Rauscheinfluss | • hohe Rechenzeit |

***Tabelle 3.1:*** *Vor- und Nachteile verschiedener lokaler Konturmerkmale*

Tabelle 3.1 listet die wichtigsten Vor- und Nachteile der vorgestellten Features auf. Die Ergebnisse der Berechnung dieser Features auf einem typischen Beispielbild sind in Abbildung 3.1 für eine Testperson dargestellt. Insbesondere der Canny-Filter und der Harris-Operator liefern sehr gute Ergebnisse auf dem Beispielbild, jedoch ist hier die Rechenzeit höher als bei Gradientenbetrag und -orientierung. Generell kann festgestellt werden, dass eine Rauschunterdrückung in den Filterantworten durch die Anwendung einer Vorverarbeitung (z.B. Gauß-

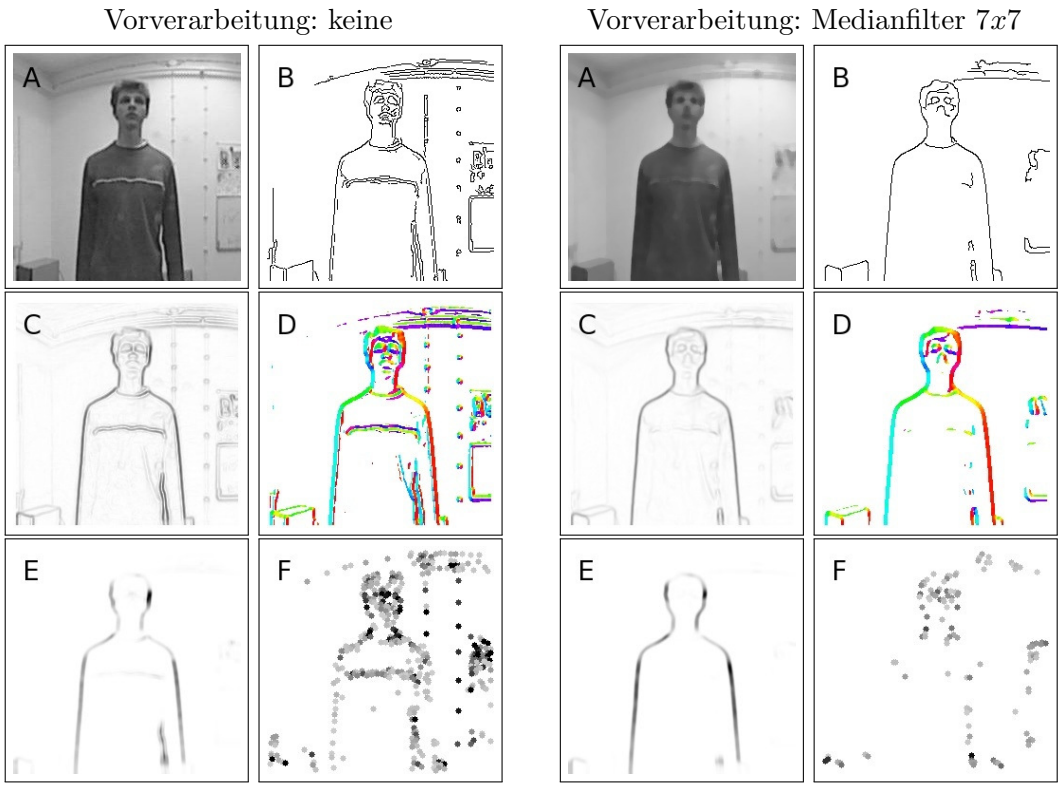



***Abbildung 3.1:*** *Ergebnisse verschiedener konturbeschreibender Features ohne Vorverarbeitung (linke Seite) und mit 7x7 Medianfilter (rechte Seite). Dargestellt sind jeweils das entsprechende Inputbild (A), der Canny-Filter (B), der Gradientenbetrag (C), die Gradientenorientierung (D), der Harris-Operator (E) und FAST-Features (F). Die Vorverarbeitung mittels Medianfilter reduziert das Rauschen, reduziert aber auch die FAST-Features erheblich.*

oder Median-Filter) erreicht werden kann. Hierdurch steigt zwar die benötigte Rechenzeit leicht an, jedoch steigt auch die Qualität der Filterantworten deutlich. Lediglich bei den FAST-Features verschlechtert die Rauschunterdrückung das Ergebnis.

Die Auswahl eines optimalen Features als konturbeschreibendes Merkmal kann jedoch nicht allgemeingültig erfolgen. Abhängig vom Bildinhalt (z.B. Objekte im Hintergrund) und der Bildqualität (z.B. Kontrast) liefern die verschiedenen Features unterschiedlich gute Antworten. Dies wird auch in [Allili and Ziou, 2007] bestätigt: Mittels Diskriminanzanalyse wurde eine Bewertung verschiedener objektbeschreibender Textur- und Konturmerkmale durchgeführt. Dabei konnte jedoch kein für beliebige Objekte optimales Konturmerkmal gefunden werden.

### 3.2.2 Verfahren zur Detektion und zum Tracken von Konturen

Neben den konturbeschreibenden Bildmerkmalen sind Verfahren notwendig, die die eigentliche Detektion und das Tracken einer Kontur im Bild ermöglichen. Aus den vier verschiedenen Gruppen von Algorithmen, werden im Folgenden einige typische Vertreter vorgestellt. Einige beinhalten eine vollständige Trackinglösung, andere nur ein Verfahren zur Detektion einer Kontur im Bild. Bei reinen Detektionsverfahren kann zusätzlich ein Tracker beispielsweise auf Basis eines *Particle Filters* realisiert werden.

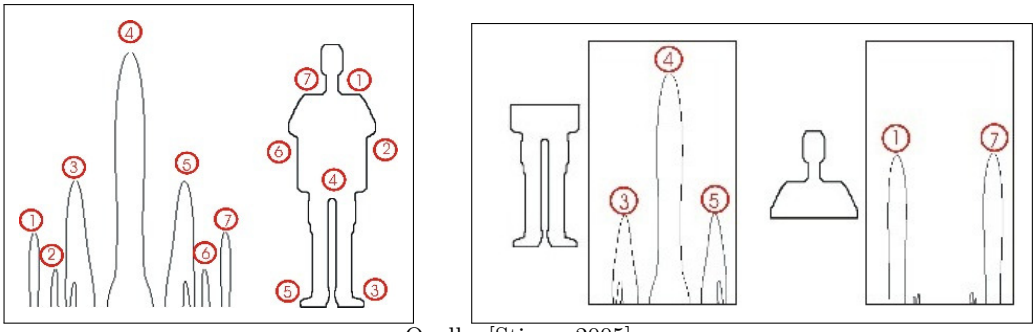
#### Appearance-based Verfahren zum Konturtracking

Der *Contracting Curve Density* (CCD) Algorithmus [Hanek and Beetz, 2004], [Panin et al., 2006] verwendet eine komplexe Energiefunktion, mit Hilfe derer eine vorgegebene Kontur an ein Inputbild angepasst werden kann. Mit Hilfe von lokalen Grauwertstatistiken wird dabei die Anpassung der Kontur an das Inputbild und eine Trennung der Kontur vom Hintergrund erreicht. Das Tracking selbst erfolgt dabei über einen linearen Kalmanfilter oder direkt durch den CCD-Algorithmus, wenn sich die Hypothese bei ausreichend hoher Bildrate noch im Bereich der letzten Beobachtung befindet. Dieser Ansatz liefert bei Daten mit geringem Rauschanteil eine stark deformierbare Kontur, ist aber zwingend auf eine gut passende und schnelle Initialsuche angewiesen, um die Konvergenz des Verfahrens zu gewährleisten. Abbildung 3.2 zeigt ein Beispiel des CCD-Algorithmus zum Tracken eines Oberkörpers.

Quelle: [Hanek and Beetz, 2004]

***Abbildung 3.2:*** *Beispiel aus [Hanek and Beetz, 2004] zum Tracken einer Person mittels CCD-Algorithmus. Gezeigt sind die Initialisierung (links) und das Ergebnis nach 2 (mitte) und 4 (rechts) Iterationsschritten.*

Ein weiterer *Appearance-based* Ansatz, der eine Konturbeobachtung in einem Unterraum abbildet, wurde von [Stiene, 2005] vorgestellt. Dabei wird eine auf der *Curvature Scale Space* (CSS) [Mokhtarian and Suomela, 1998] Theorie aufbauende Darstellung der aktuellen Beobachtung erzeugt und erneut als Abbildungsproblem formuliert. Die CSS-Repräsentation stellt dabei eine Faltung einer pfadparametrisierten Kurve (z.B. der aktuellen Beobachtung) mit einer Gaußfunktion dar. Durch die Faltung wird die Kurve geglättet. Durch mehrfache



Quelle: [Stiene, 2005]

***Abbildung 3.3:*** *Links: Darstellung eines Menschen im CSS.*
*Rechts: CSS-Darstellung zwei beispielhafter Teilkonturen.*

Anwendung dieser Faltung mit wachsender Standardabweichung $\sigma$ werden die Teile der Kontur ermittelt, die hauptsächlich zum Aussehen beitragen. Der große Vorteil des Verfahrens ist die Möglichkeit, Objektkonturen auch bei teilweiser Verdeckung, durch die Aufteilung der CSS-Darstellung auf menschliche Teilkonturen wie Kopf oder Beine (siehe Abb. 3.3), zu erfassen. Um eine Konvergenz des Verfahrens zu ermöglichen, ist jedoch auch bei diesem Ansatz eine gute Initialisierung notwendig.

#### Features-based Verfahren zum Konturtracking

Bei den *Feature-based* Verfahren werden spezifische (Bild-)Merkmale mit Hilfe entsprechender Detektoren oder Filter aus dem Bild extrahiert, die in einem nachfolgenden Schritt zum Tracking verwendet werden können.

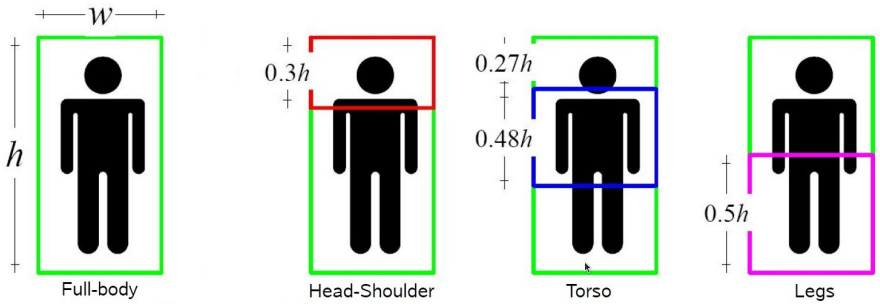
In [Ferrari et al., 2009] wird ein System zur Erkennung der Oberkörperpose basierend auf *Histograms of Oriented Gradients (HOG)* vorgestellt. Die grundlegende Idee der HOGs wird im Abschnitt D.5 näher beschrieben. Eine ausführliche Beschreibung ist in [Dalal and Triggs, 2005] zu finden.

In [Ferrari et al., 2009] werden auf einem Bild nach einer initialen Oberkörperdetektion diese HOGs eingesetzt, um verschiedene *pose descriptors* zu ermitteln. Diese werden anschließend genutzt, um ähnliche Posen in einer Vergleichsdatenbank (*retrieval database*) zu finden (*pose retrieval*). Auf diese Art und Weise können Oberkörperposen aus unbekannten Bildern klassifiziert werden. In den in [Ferrari et al., 2009] vorgestellten Experimenten werden dabei drei Klassen (Arm an die Hüften gestützt, Arm normal hängend und Arme vor dem Oberkörper verschränkt) unterschieden.

### Template-based Verfahren zum Konturtracking

Auf *Template-Matching* aufbauende Verfahren verwenden ein Vergleichsmuster von meist offline erzeugten Konturfeaturemasken (Templates). Diese werden mit Merkmalen des Eingangsbildes verglichen. Letztlich wird eine gute Korrelation der verwendeten Templates mit den Merkmalen des Inputbildes gesucht.

Eine Variante, die zum Tracking von mehreren Personen mit einer stationären Kamera geeignet ist, stellen die *Edgelet Part Detectors* [Wu and Nevatia, 2005] dar, die mittels *AdaBoost* [Freund and Schapire, 1997] trainierte Konturklassifikatoren zum Tracken von mehreren Personen verwenden. Ein *Edgelet* repräsentiert dabei eine Filtermaske als Kombination von einfachen Konturliniensegmenten für Beinpaare, Kopf-Schulter und Torso, welche mit den Grauwertgradienten des Inputs gefaltet werden. Der große Nachteil dieses Ansatzes ist der sehr aufwendige Trainingsprozess zur Erstellung der *Edgelet Part Detectors*.




Quelle: [Wu and Nevatia, 2005]

***Abbildung 3.4:*** *Zusammensetzung einer menschlichen Kontur aus Edgelet-Teilklassifikatoren nach [Wu and Nevatia, 2005]: Die Gesamtkontur (links) kann zusammengesetzt werden aus einer Kopf-Schulter-Kontur (rot), einem Torso (blau) und den Beinen (violett). Jeder der Teilklassifikatoren wird als ein Strong Classifier im AdaBoost-Algorithmus realisiert.*

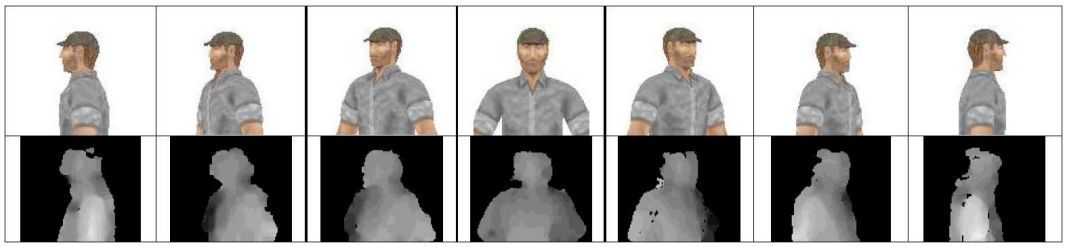
Ein texturbasierter Ansatz zur Personenklassifikation wurde von [Gavrila and Munder, 2007] in einer Anwendung in der Automobilindustrie vorgestellt. Dabei werden interessante Bildregionen mittels hierarchischer Korrelation von Shape-Templates gesucht und in einem zweiten Schritt die Texturinformationen der Region zur Klassifikation der Kontur der Person verwendet. Als Abstandsmaß kommt hier die *Champfer*-Distanz zum Einsatz. Es wurden gelabelte Paare von Textur- und Shapetemplates aus einer Testdatenbank verwendet. Genauere Angaben über den verwendeten Trackingansatz wurden nicht veröffentlicht. Abbildung 3.5 zeigt ein typisches Beispiel beim Einsatz dieses Systems.

In [Li et al., 2010] wird ein Template-basierter Ansatz zur Schätzung der Oberkörperorien-

Quelle: [Gavrila and Munder, 2007]

***Abbildung 3.5:*** *Beispiel aus [Gavrila and Munder, 2007]. Positive Hypothesen sind Rot und Fehldetektionen Grün dargestellt.*

tierung vorgestellt. Mit Hilfe des Tiefenbildes einer Stereokamera werden zunächst zusammenhängende Bildregionen identifiziert. Diese werden anschließend mit Hilfe eines Template-Matchings auf sieben mögliche Oberkörperposen überprüft. Als Templates stehen sieben Ansichten von aufrecht stehenden Personen in den Winkeln ±90°, ±60°, ±30° und 0° zur Verfügung, die mit Hilfe eines modifizierten AdaBoost-Ansatzes trainiert wurden. Abbildung 3.6 zeigt synthetische Beispielbilder der sieben verwendeten Ansichten und korrespondierende Tiefenbilder echter Daten.



Quelle: [Li et al., 2010]

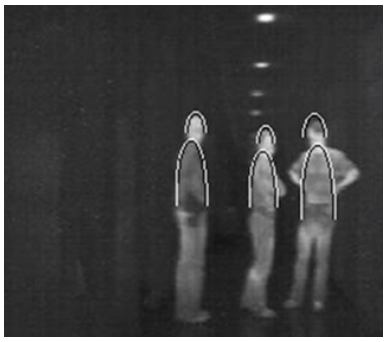
***Abbildung 3.6:*** *Verwendete Ansichten und Tiefenbilder zur Schätzung der Oberkörperorientierung mittels Templates aus [Li et al., 2010].*

Auf einer Testdatenbank mit 430 Tiefenbildern, die in etwa gleichverteilt über die sieben Kategorien ist, wurde eine Treffergenauigkeit von 90.7% erreicht. Dabei wurden für die Winkelklassen 0° und ±90° Erkennungsraten von nahezu 100% erzielt. Bei ±30° und ±60° lag die Quote bei ca. 80%. In [Li et al., 2010] wird keine Aussage gemacht, ob das Verfahren mit gleicher Güte auch ohne Tiefeninformationen funktionieren würde.

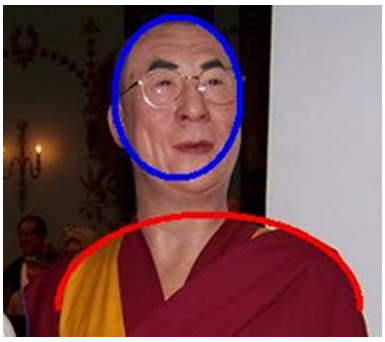
### Model-based Verfahren zum Konturtracking

In [Treptow et al., 2005] wird ein Verfahren vorgestellt, in dem ein parametrisches Modell zum Konturtracking in Bildern einer Infrarotkamera eingesetzt wird. Das parametrische Modell besteht aus einem neun-dimensionalen Vektor, der die Geometrie des menschlichen

Oberkörpers (insbesondere der Kopf-Schulter-Kontur) bestehend aus mehreren Ellipsenteilen beschreibt. Auf Grund des Einsatzes der Infrarotkamera können einfache Grauwertgradienten als Konturmerkmale verwendet werden. Von [Einecke, 2006][1] wird dieser Ansatz um die Verwendung von Gradientenorientierung erweitert, wodurch ein Einsatz auf herkömmlichen Bildern möglich ist. Beide Varianten verwenden zum eigentlichen Tracken einen *Particle Filter* [Isard and Blake, 1998], mit dessen Hilfe die Modellparameter unter Berücksichtigung eines Beobachtungs- und eines Bewegungsmodells bestimmt werden. Abbildung 3.7 zeigt ein Beispielergebnis beider Verfahren.



Quelle: [Treptow et al., 2005]



Quelle: [Einecke, 2006]

***Abbildung 3.7:*** *Beispiele zum Konturtracking eines menschlichen Oberkörpers auf Basis von elliptischen Modellen. Gezeigt sind die Verfahren aus [Treptow et al., 2005] und [Einecke, 2006] jeweils mit Bild und Detektionsergebnissen.*

Ein weiterer Ansatz wird von [Schmidt et al., 2006] vorgestellt, der ein dreidimensionales Modell an mehrere Features des Eingangsbildes anpasst. Dabei werden neben der Erfassung der Grauwertkanten auch Informationen der von Rechtecken umschlossenen Flächen wie z.B. der mittlere Farbwert oder die Ähnlichkeit mit Hautfarbe verwendet. Für das Tracking der aggregierten Features wird ein *Kernel Particle Filter* [Chang et al., 2005a] verwendet.

Eine weitere Klasse von modellbasierten Verfahren stellen die *Active Shape* und *Active Appearance Models* dar, die beide eine distanzminimierende Abbildung der aktuellen Beobachtung in einen mittels Hauptkomponentenanalyse dimensionsreduzierten Unterraum verwenden. Die bereits älteren Active Shape Models (ASM) wurden von [Cootes et al., 1995] vorgestellt. Dabei wird ein parametrisches Modell (eine pfadparametrische Kurve der Objektkontur) im dimensionsreduzierten Unterraum iterativ so angepasst, dass dieses möglichst gut auf eine aktuelle Beobachtung im Bildraum passt. Als Bildmerkmale werden hier Grauwertgradienten verwendet. Weitere Veröffentlichungen basieren auf den Active Appearance Models (AAM) [Cootes and Taylor, 2001], welche neben der Gradientenin-

[1] Diese Arbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

formation auch die Grauwerte innerhalb einer Konturfläche betrachten. Der Nachteil der AAM-Modelle besteht in der Notwendigkeit, ein (Grauwert-)Modell für die eingeschlossene Fläche der Kontur zu erstellen. Im Hinblick auf das Ziel des Trackens des menschlichen Oberkörpers, kann man leicht erkennen, dass auf Grund der Vielfalt von Bekleidung kaum ein geeignetes Modell gefunden werden kann. Weiterhin ist der Berechnungsaufwand für die Anpassung eines Actives Appearance Models deutlich höher als bei einem Active Shape Model. Beide Varianten erfordern eine gute Initialisierung, damit eine Konvergenz der Anpassung erreicht werden kann.

Basierend auf den *Active Shape Models* wurde von [Belz, 2008][2] ein prototypisches Trackingsystem für den Oberkörper einer Person vorgestellt, das im Rahmen dieser Dissertation unter den gewählten Randbedingungen und Einschränkungen weiterentwickelt wurde.

### 3.2.3 Zusammenfassung

Allgemein kann festgestellt werden, dass es kein eindeutiges Merkmal gibt, dass allein zum robusten Tracken des menschlichen Oberkörpers in beliebigen Umgebungen optimal geeignet ist. Je nach Bildinhalt und Bildqualität können unterschiedliche Features unterschiedlich gut geeignet sein. Im Allgemeinen können Grauwertgradienten (Betrag und Orientierung) jedoch als ein guter Kompromiss angesehen werden. Der Berechnungsaufwand ist relativ niedrig, und die Robustheit ist in vielen Konstellationen ausreichend gut.

Im Hinblick auf die eigentlichen Trackingverfahren kann man feststellen, dass fast alle Verfahren eine hinreichend gute Initialisierung erfordern. Ohne eine Initialisierung oder mit nur einer sehr schlechten, konvergieren die genannten Verfahren typischerweise kaum bis überhaupt nicht.
Bezüglich der Echtzeitfähigkeit unterscheiden sich die Verfahren der vier Kategorien teilweise recht erheblich: Algorithmen basierend auf dem *Curvature Scale Space* (CSS) von [Stiene, 2005] oder Verfahren auf HOG-Basis [Ferrari et al., 2009] benötigen im Allgemeinen viel Rechenzeit. Dagegen arbeiten die *Edgelet Part Detectors* von [Wu and Nevatia, 2005] oder die *Active Shape Models* von [Cootes et al., 1995] deutlich schneller.

In der Literatur wurden nur sehr wenige Verfahren (wie z.B. [Li et al., 2010]) gefunden, die eine Schätzung der Oberkörperorientierung realisieren. Die meisten Verfahren konzentrieren sich nur auf die Erfassung der Kontur des Oberkörpers. Eine anschließende Verwertung der Kontur zur Schätzung der Oberkörperpose wird fast nie beschrieben. Es finden sich teilweise einige qualitative Aussagen, aber kaum quantitative Auswertungen.

[2] Diese Diplomarbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

Da im Rahmen dieser Dissertation vorranging modellbasierte "Analyse durch Synthese"-Verfahren zum Einsatz kommen sollen, erfolgt die hier vorgestellte Schätzung der Oberkörperorientierung auf Basis von Active Shape Models. Diese stellen einen sehr guten Kompromiss zwischen Anpassungsfähigkeit, Echtzeitfähigkeit und Güte der Detektionsergebnisse dar.

## 3.3 Systembeschreibung

Zur Bestimmung der Oberkörperpose ist es zunächst notwendig, den Oberkörper des potentiellen Benutzers im Kamerabild zu detektieren und möglichst genau zu erfassen. Nach einer erfolgreichen Erfassung des Oberkörpers kann dann die eigentliche Bestimmung der Orientierung vorgenommen werden. Dieser gesamte Erkennungsprozess wird im Rahmen dieser Dissertation in drei Teilschritte untergliedert:

1. Grobdetektion des Oberkörpers im Kamerabild:
   Im ersten Schnitt soll zunächst nur eine grobe Detektion des Oberkörpers erfolgen. Ziel ist es hierbei aus einem Inputbild nur die wirklich relevanten Bereiche *(ROI=Region-of-Interest)* für eine weitere Analyse herauszufiltern.
   Idealerweise sollte in dieser ersten Stufe ein Algorithmus eingesetzt werden, der möglichst keine Personen im Bild übersieht (also hohe Detektionsrate, *true-positive*) und auch möglichst wenig Fehldetektionen *(false-positive)* generiert. Da das System auf einem mobilen Roboter in Echtzeit (oder zumindestens annähernd) arbeiten soll, muss das Verfahren zur Grobdetektion online auf Videodaten arbeiten können.
   Als Ergebnis dieser ersten Stufe stehen eine Reihe von Bildregionen (z.B. Rechtecke) zur Verfügung, die die Person(en) im Sichtbereich der Kamera enthalten. Die Größe und welche Körperteile der Person sichtbar sind, hängt stark von der verwendenten Kamera und der Entfernung der Person zum Roboter ab.
   Weitere Details zu dieser ersten Verarbeitungsstufe finden sich im Abschnitt 3.4.

2. Feinanpassung mit Oberkörpermodell:
   Nachdem in der ersten Stufe der Oberkörper als grobe Bildregion ermittelt wurde, soll in der zweiten Stufe die Feinanpassung erfolgen. Hierzu soll ein modellbasierter Algorithmus verwendet werden. Dabei wird ein parametrisches Modell innerhalb eines ausgewählten Bildausschnittes so angepasst, dass die erzeugte Instanz des Modells den Inhalt des Bildausschnitts (hier also der Oberkörper) möglichst gut beschreibt.
   Da wie im ersten Teilschritt auf Videodaten online gearbeitet werden soll, muss die Feinanpassung ebenfalls in Echtzeit arbeiten können.

   Im Rahmen dieser Dissertation wird für diesen Schritt ein *Active Shape Model* (basierend auf den Vorarbeiten von [Belz, 2008][3] eingesetzt. Ausführliche Details hierzu

[3] Diese Diplomarbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

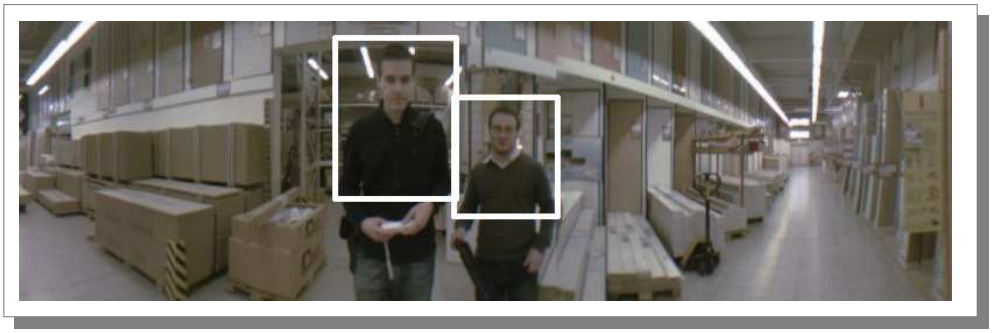


***Abbildung 3.8:*** *Schematische Darstellung zur Auswahl von relevanten Bildregionen bei der Oberkörperdetektion. Das Bild zeigt einen Ausschnitt von 180° einer omnidirektionalen Kamera. Die Personen befinden sich hier ca. 1m vom Roboter entfernt. Die weiß-markierten Bildbereich können in einer nachgeschalteten Stufe weiter untersucht werden.*

finden sich im Abschnitt 3.5.

Als Ergebnis diesen zweiten Schrittes steht ein an die Person angepasstes Oberkörpermodell in parametrischer Form zur Verfügung.

3. Schätzung der Oberkörperpose:
   In der letzten Stufe dieses Teilsystems erfolgt die eigentliche Schätzung der Oberkörperpose. Da aus dem vorangegangenen Schritt ein parametrisches Modell vorliegt, kann die Schätzung anhand der gefundenen Parameter erfolgen.
   Als Ergebnis liegt letztendlich eine Winkelschätzung vor, die die Ausrichtung des Oberkörpers einer Person im Kamerabild beschreibt.

## 3.4 Oberkörperdetektion

Im Rahmen des ersten Verarbeitungsschrittes sollen die Bildregionen ermittelt werden, die die Personen im Blickfeld der Kamera enthalten. Hierzu wurde im Rahmen der Dissertation ein zweistufiger Ansatz untersucht: In der ersten Stufe wird mittels eines adaptiven Hintergrundmodells eine Trennung von Person(en) und Hintergrund vorgenommen. In der zweiten Stufe erfolgt die eigentliche Detektion der Person(en).

### 3.4.1 Trennung von Vorder- und Hintergrund

Die meisten Verfahren, die in einem Bild den Hintergrund vom Vordergrund trennen, basieren auf derselben Annahme: Der Hintergrund eines Bildes wird als unveränderlich/statisch angesehen, während ein Objekt des Vordergrundes sich bewegt. Die Verfahren definieren ein Objekt des Vordergrundes also nicht durch seine Distanz zur Kamera, sondern dadurch, dass

es sich bewegt. Eine Ausnahme bilden hier nur Verfahren, die mit Stereo-Kameras arbeiten, die aber im Weiteren hier nicht betrachtet werden. Da die Definition des Vordergrundes über eine Bewegung im Bild erfolgt ist, ist eine Voraussetzung für die Anwendung eines solchen Verfahrens, dass sich die Kamera selbst nicht bewegt, da sonst im gesamten Bild die Bewegung der Kamera deutlich wird und keine Segmentierung mehr möglich ist.

Aufbauend auf der Annahme, dass der Hintergrund statisch ist und sich Objekte des Vordergrundes bewegen, gibt es eine Vielzahl von Verfahren, die versuchen, die sich bewegenden Objekte aus dem Bild zu extrahieren. Erschwerend dabei ist der Umstand, dass sich Objekte des Hintergrundes nicht immer absolut statisch verhalten. Ein Beispiel dafür ist ein Baum, der zwar eigentlich statisch ist, dessen Blätter und Äste sich aber im Wind leicht bewegen können. Auch veränderte Beleuchtungsbedingungen können dafür sorgen, dass eigentlich statische Objekte im Bild mit der Zeit ein verändertes Aussehen durch Schattenwurf oder eine andere Farbe durch veränderte Helligkeit haben. Einige Verfahren, die unterschiedlich stark auf diese Aspekte eingehen, werden im Folgenden genannt:

- Differenzbildverfahren
- running Gaussian average
- temporal median filter
- mixture of Gaussians
- kernel density estimation
- sequential kernel density approximation
- co-occurence of image variations
- Eigen-Backgrounds

Ein Überblick über diese Verfahren findet sich in [Piccardi, 2004]. In [Wimmer, 2005] wird ein Einblick in die Unterschiede bei der Anwendung der verschiedenen Modelle gegeben.

Im Rahmen dieser Dissertation wurde ein einfaches Differenzbildverfahren mit Nachbehandlung des Bildes durch einen *Closing*- und einen *Connected-Regions*-Algorithmus verwendet, da diese Kombination mit geringem Rechenaufwand in Echtzeit realisiert werden kann und bereits eine sehr gute Segmentierung erreicht wird, insofern der Hintergrund weitestgehend statisch oder bekannt ist. Weitere Details finden sich im Anhang D.3.

Die Adaptivität des Verfahrens wird durch eine kontinuierliche Anpassung des statischen Hintergrundes erreicht. Hierzu findet eine zeitliche Tiefpass-Filterung des Bildes statt. Da ein solcher Tiefpass jedoch dazu führen würde, dass still-stehende Personen nach einer

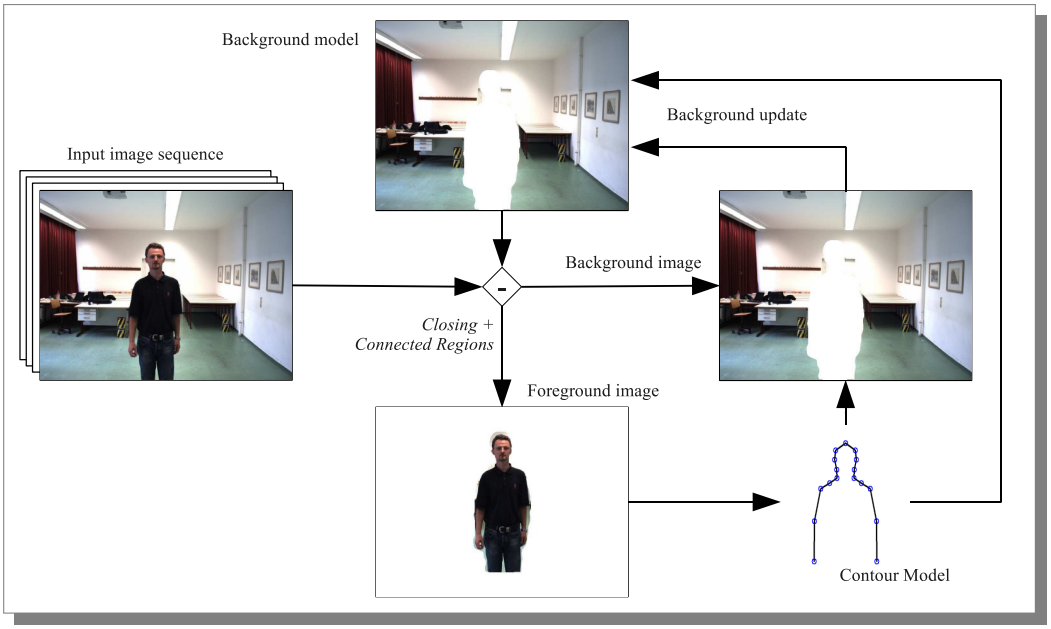



***Abbildung 3.9:*** *Prinzip Segmentierung Vorder-/Hintergrund: Die Segmentierung in Vorder- und Hintergrund wird durch ein Differenzbild des Inputs mit dem aktuellen Hintergrundmodell erreicht. Das Hintergrundmodell wird bei jedem Update auf Basis des aktuellen Hintergrundes und des Tracker-Outputs adaptiert.*

entsprechenden Zeit teilweise oder vollständig mit dem Hintergrund "verschmelzen", werden die Regionen des Bildes von der Tiefpassfilterung ausgeschlossen, in denen bereits Personen detektiert wurden. Dies wird durch eine Rückkopplung der Ergebnisse des Trackers erreicht (siehe Abbildung 3.9).

Die Trennung von Vorder- und Hintergrund, basierend auf dem beschriebenen Differenzbildverfahren, funktioniert nur, wenn sich der Roboter bzw. die Kamera nicht bewegt. Dies trifft zu, wenn der Roboter an einer Ruhe- oder Warteposition auf einen neuen Benutzer wartet. Wenn ein Shopping-Roboter jedoch auf der Suche nach einem neuen Kunden ist, kann diese Annahme nicht mehr getroffen werden und das Differenzbildverfahren kann nicht mehr eingesetzt werden. Sobald der Roboter jedoch wieder zum Stillstand kommt, kann das Verfahren wieder aktiviert werden.

### 3.4.2 Oberkörperdetektion mittels HOG

Basierend auf den *Histograms of Oriented Gradients (HOG)* wurde in [Dalal and Triggs, 2005] ein Verfahren zur Detektion von Personen vorgestellt, dass deutlich bessere Detektionsergebnisse als andere Verfahren erreicht hat.

Der Grundgedanke des Verfahrens ist, dass die lokale Form und das Aussehen eines Objektes sehr gut durch eine Verteilung von lokalen Gradientenorientierungen beschrieben werden können, ohne die genaue Position der Gradienten kennen zu müssen. Dazu erfolgt die Bestimmung eines hochdimensionalen HOG-Featurevektors, indem ein Inputbild in Zellen zerlegt wird und für jede ein Histogramm der Gradientenorientierungen innerhalb der Zelle bestimmt wird. Die eigentliche Detektion erfolgt mittels einer herkömmlichen *Support-Vector-Machine*, welche die HOG-Featurevektoren eines über das Bild geschobenen Suchfensters klassifiziert.

Weitere Details zur Funktionsweise eines HOG-Detektors und zum Training eines geeigneten HOG-Detektors für Oberkörper sind im Anhang D.5 beschrieben.

Da dieses Verfahren u.a. in der Lage ist, Personen auch vor strukturiertem Hintergrund zu detektieren, wurde es in Kombination mit dem Differenzbildverfahren im Rahmen dieser Dissertation zur Oberkörperdetektion eingesetzt.

### 3.4.3 Kombination von Hintergrundmodell und HOG

Im Abschnitt 3.4.1 wurde ein Verfahren zur Trennung von Vorder- und Hintergrund basierend auf einem Differenzbild vorgestellt. Bei stehendem Roboter wird auf dem Ergebnisbild dieser Vorverarbeitung der HOG-Detektor angewandt, um eine Grobdetektion zu realisieren. Dies ist notwendig, um die Person von anderen sich bewegende Bildregionen (z.B. einem Schatten) zu unterscheiden. Gleichzeitig werden personenähnliche Strukturen im Hintergrund durch das Differenzbildverfahren unterdrückt.

Wenn sich das Robotersystem bewegt, kann nur noch der HOG-Detektor zur Grobdetektion der Person eingesetzt werden. Im Regelfall liefert dieser allein jedoch immer noch sehr gute Ergebnisse.

## 3.5 Active Shape Models

Die *Active Shape Models (ASMs)* wurden von [Cootes et al., 1995] vorgestellt. Im folgenden Abschnitt werden die Grundidee, die Modellerstellung und die Modellanpassung näher beschrieben. Im Rahmen dieser Dissertation werden die ASMs eingesetzt, um den grob detektierten Oberkörper einer Person genauer zu erfassen und zu beschreiben, so dass auf dieser Grundlage eine Schätzung der Oberkörperpose möglich wird.

### 3.5.1 Grundidee

Das Ziel dieses Verfahrens ist die genaue Detektion oder Erfassung von Formen einer bestimmten Klasse (z.B. eine einfache geometrische Form, eine menschliche Hand oder die Außenkontur des Oberkörpers) in einem gegebenen Bild. Hierzu wird ein im Vorfeld errechnetes parametrisches Modell der entsprechenden Formenklasse eingesetzt, das iterativ an das Inputbild angepasst wird. Als Ergebnis entsteht ein entsprechend angepasstes Modell und der dazugehörige Parametersatz. Für weitere nachgeschaltete Analyseschritte reicht dann der ermittelte Parametersatz aus, da dieser das Modell vollständig beschreibt.

Die Bestimmung des Parametersatzes eines Active Shape Models stellt einen klassischen *Analysis-by-synthesis* Ansatz dar. Hierbei wird nicht versucht, die zu detektierende Form direkt aus dem Bild zu extrahieren. Stattdessen wird in einem iterativen Prozess das Modell ausgehend von einer Startschätzung solange verfeinert, bis es der gesuchten Form möglichst ähnlich ist (siehe Abschnitt 2.4).

Aus dem synthetisierten Modell und einer aktuellen Beobachtung, ergibt sich ein Fehler zwischen Modell und Beobachtung. Basierend auf diesem Fehler wird durch einen Anpassungsalgorithmus versucht, die Modellparameter so anzupassen, dass sich im nächsten Iterationsschritt ein kleinerer Fehler ergibt (also die Parameter das gesuchte Objekt besser beschreiben). Dieser Iterationsprozess wird solange wiederholt, bis eine definierte Fehlerschwelle unterschritten, eine bestimmte Anzahl von Schritten überschritten oder ein anderes Abbruchkriterium erreicht wird.

### 3.5.2 Modellerstellung

Zur Erstellung eines Active Shape Models bestehend aus $n$ Stützstellen $p_j = (x_j, y_j)$ der Kontur müssen zunächst eine Reihe von Trainingsbildern manuell gelabelt werden. Konkret müssen die einzelnen Stützstellen der Kontur möglichst exakt erfasst werden. Dies erfolgt typischerweise manuell. Die Stützstellen $p_j$ sollten dabei sowohl an markanten Stellen (z.B. Ecken oder Wendepunkten) der Kontur, als auch entlang längerer gerader Abschnitte gleichmäßig verteilt werden. Für jedes Bild $i$ einer Datenbasis mit $N$ Bildern ergibt sich somit ein $2n$-dimensionaler Labelvektor $\mathbf{z_i}$:

$$\mathbf{z_i} = \left(x_1^{(i)}, y_1^{(i)}, ..., x_n^{(i)}, y_n^{(i)}\right)^T \qquad i = 1, ..., N \tag{3.1}$$

Diese $N$ Labelvektoren dienen als Grundlage für die Erstellung des Active Shape Models. In einem Vorverarbeitungsschritt werden diese von globalen Transformationen wie Translation, Skalierung und Rotation befreit, da das Modell lediglich die lokale Form beschreiben soll. Hierzu wird typischerweise der *Procustes-Algorithmus* (siehe Anhang D.2) verwendet. Als Ergebnis entstehen eine Reihe von Labelvektoren $\mathbf{z'_i}$, deren Mittelwert (also die Durch-

schnittsform) wie folgt bestimmt werden kann:

$$\overline{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{z_i'} \tag{3.2}$$

Die Menge der erstellten Vektoren $\mathbf{z_i'}$ kann als Punktewolke eines $2n$-dimensionalen Ellipsoiden aufgefasst werden. Durch die Anwendung einer Hauptkomponentenanalyse (PCA) auf dieser Punktewolke, können die Hauptachsen dieses Ellipsoids bestimmt werden. Dazu wird für jede Form $\mathbf{z_i'}$ zunächst die Abweichung $d\mathbf{z_i}$ von der mittleren Form bestimmt:

$$d\mathbf{z_i} = \mathbf{z_i'} - \overline{\mathbf{z}} \tag{3.3}$$

Mit Hilfe der $d\mathbf{z_i}$ kann eine Kovarianzmatrix $\mathbf{S}$ berechnet werden:

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} d\mathbf{z_i} d\mathbf{z_i}^T \tag{3.4}$$

Die Hauptachsen des Ellipsoids können durch die Eigenvektoren $\mathbf{s_i}$ von $\mathbf{S}$ beschrieben werden:

$$\mathbf{S}\mathbf{s_i} = \lambda_i \mathbf{s_i} \qquad i = 1, ..., 2n \tag{3.5}$$

wobei $\lambda_i$ der $i$-te Eigenwert von $\mathbf{S}$ ist mit $\lambda_i \geq \lambda_{i+1}$ und $\mathbf{s_i}^T\mathbf{s_i} = 1$. Es kann gezeigt werden, dass die Eigenvektoren $\mathbf{s_i}$ von $\mathbf{S}$, die zu den größten Eigenvektoren $\lambda_i$ gehören, auch die größten Hauptachsen des Ellipsoids und somit auch die größte Verformungskomponente (Variation innerhalb der Daten) der Datenmenge beschreiben.

Auf Basis dieser Eigenvektoren $\mathbf{s_i}$ kann jede Form $\mathbf{s} \in \{\mathbf{z_i'}, i = 1..N\}$ exakt über eine Linearkombination über die $\mathbf{s_i}$ wie folgt beschrieben werden:

$$\mathbf{s} = \mathbf{s_0} + \sum_{i=1}^{N} p_i \mathbf{s_i} \qquad \text{mit} \qquad \mathbf{s_0} = \overline{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{z_i'} \tag{3.6}$$

wobei $\mathbf{s_0} = \overline{\mathbf{z}}$ den Mittelwert aller vorverarbeiteten Formen $\mathbf{z_i'}$ darstellt und die Eigenvektoren $\mathbf{s_i}$ jeweils mit einem Parameter $p_i$ skaliert werden. Durch die Variation der einzelnen Parameter $p_i$ können alle Formen $\mathbf{z_1'}$ bis $\mathbf{z_N'}$ sowie auch Linearkombinationen zwischen diesen Formen approximiert werden. Der Parameter $p_i$ sollte dabei maximal im Intervall $\left[-2\sqrt{\lambda_i}...+2\sqrt{\lambda_i}\right]$ variiert werden, wobei $\lambda_i$ der zur Komponente $\mathbf{s_i}$ korrespondierende Eigenwert ist. Damit kann sichergestellt werden, dass die resultierende Modelländerung über die Statistik der Labeldaten abgedeckt wird und nicht entartet oder deformiert ist [Cootes et al., 1995].

Abbildung 3.10 zeigt ein Beispiel für die ersten Komponenten eines Active Shape Models

einer menschlichen Oberkörperkontur.

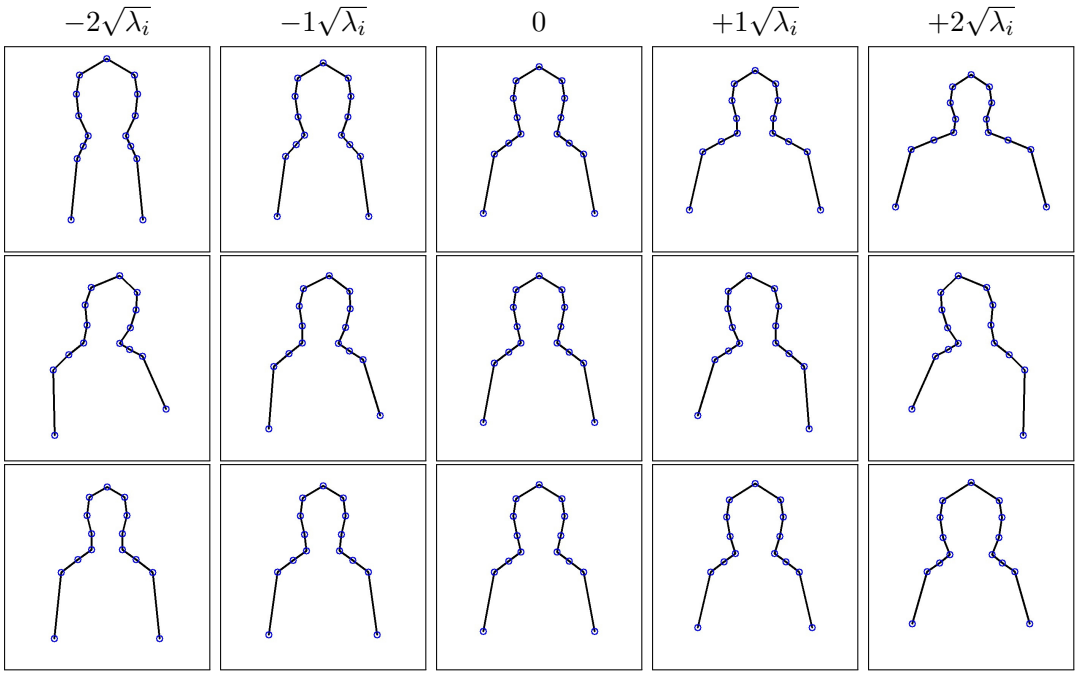



***Abbildung 3.10:*** *Beispiel eines Active Shape Models für eine menschliche Oberkörperkontur. Dargestellt sind die ersten drei Komponenten $s_i$ des Modells mit jeweils deren Ausprägung im Intervall $\left[-2\sqrt{\lambda_i}, 2\sqrt{\lambda_i}\right]$. Die mittlere Spalte zeigt jeweils die mittlere Form $\mathbf{s_0} = \overline{\mathbf{z}}$.*

Weiterhin kann durch das Entfernen von Eigenvektoren $\mathbf{s_i}$ mit geringem Eigenwert $\lambda_i$ eine erhebliche Dimensionsreduktion erreicht werden. Dies ist möglich, da diese Eigenvektoren nur kleinere Änderungen (z.B. einzelne Ausreißer oder Rauschen) in den Daten beschreiben und daher nicht weiter relevant sind. Das reduzierte Modell kann dann wie folgt beschrieben werden:

$$\mathbf{s} \approx \mathbf{s_0} + \sum_{i=1}^{N'} p_i \mathbf{s_i} \qquad \text{mit} \qquad N' < N \tag{3.7}$$

Dabei wird $N'$ typischerweise so gewählt, dass mit dem reduzierten Modell noch 90% oder 95% der Trainingsdaten abgedeckt werden können. Alternativ kann $N'$ auch manuell bestimmt werden.

Als Ergebnis der Modellerstellung liegt somit ein Active Shape Model bestehend aus einem Mittelwertvektor $\mathbf{s_0}$ und einer Reihe von Eigenvektoren $\mathbf{s_i}$ mit korrespondierenden Eigenwerten $\lambda_i$ vor. Dieses Modell beschreibt die lokalen Formveränderungen der Formenklasse

aus den gelabelten Trainingsbildern innerhalb eines festen (Einheits-)Koordinatensystems, da die Formvektoren eingangs durch Anwendung des *Procustes-Algorithmus* von sämtlichen globalen Transformationen befreit wurden. Zur Anpassung des Modells auf einem beliebigen Inputbild muss dieses daher zusätzlich noch global (im Bild) transformiert werden.

Die globale Transformation innerhalb eines Inputbildes wird durch eine Translation, eine Skalierung und eine Rotation beschrieben. Die Integration dieser drei Transformationen in das Active Shape Model kann auf zwei verschiedenen Wegen erfolgen: Die Transformationen können als separater Berechnungschritt oder als zusätzliche synthetische Formkomponenten integriert werden.

#### Globale Transformation als separater Berechnungsschritt

Eine globale Modellinstanz $\mathbf{S_M}$ innerhalb eines Inputbildes kann wie folgt dargestellt werden:

$$\mathbf{S_M} = M(s,\phi)[\mathbf{x}] + \mathbf{X_c} \qquad \text{mit} \qquad \mathbf{X_c} = (x_c, y_c, x_c, y_c, ..., x_c, y_c)^T \tag{3.8}$$

wobei $\mathbf{x}$ das lokale Formmodell gemäß (3.7) ist und $M(s,\phi)[\cdot]$ eine Skalierung um den Faktor $s$ und eine Rotation um den Winkel $\phi$ darstellt. Die Translation wird als eine Verschiebung in den Mittelpunkt $(x_c, y_c)$ realisiert.

Bei dieser Art der Integration der globalen Transformation in das Active Shape Model können die Skalierung $s$, der Drehwinkel $\phi$ und die Position im Inputbild $(x_c, y_c)$ direkt "abgelesen" werden. Jedoch müssen diese Parameter bei der Anpassung eines Modells an ein Inputbild auch in einem separaten Berechnungsschritt ermittelt werden.

Mit Hilfe von synthetischen Formkomponenten kann die separate Berechnung der Translation, Skalierung und Rotation vermieden werden.

#### Globale Transformation mittels synthetischer Formkomponenten

In [Matthews and Baker, 2004] wurde eine Variante zur Integration der globalen Transformation in den Anpassungsalgorithmus eines *Active Appearance Model* vorgestellt, die auch für ein *Active Shape Model* eingesetzt werden kann.

Ausgehend von der Grundform $\mathbf{s_0} = (x_1^0, y_1^0, ..., x_n^0, y_n^0)^T$ können vier zusätzliche Formkomponenten $\mathbf{s}_{tx}, \mathbf{s}_{ty}, \mathbf{s}_{scale}, \mathbf{s}_{rot}$ zur Beschreibung von Translation, Skalierung und Rotation erstellt

werden. Dazu werden zunächst folgende Hilfskomponenten erstellt:

$$\begin{aligned} \mathbf{s}^*_{scale} &= \left(x^0_1, y^0_1, ..., x^0_n, y^0_n\right)^T \\ \mathbf{s}^*_{rot} &= \left(-y^0_1, x^0_1, ..., -y^0_n, x^0_n\right)^T \\ \mathbf{s}^*_{tx} &= (1, 0, ..., 1, 0)^T \\ \mathbf{s}^*_{ty} &= (0, 1, ..., 0, 1)^T \end{aligned} \tag{3.9}$$

Die Komponenten $\mathbf{s}^*_{tx}$ und $\mathbf{s}^*_{ty}$ beschreiben eine Translation in x- und y-Richtung. Die Komponente $\mathbf{s}^*_{scale}$ realisiert eine Skalierung um den Mittelpunkt der lokalen Form. Eine Rotation senkrecht zur Bildebene wird durch die Komponente $\mathbf{s}^*_{rot}$ approximiert. Für kleine Rotationswinkel ist diese Approximation hinreichend genau.

Die Hilfskomponenten werden anschließend nach dem *Gram-Schmidt'schen Verfahren* (siehe Anhang D.4) orthogonalisiert: Die beiden Komponenten für die Verschiebung in x- und y-Richtung $\mathbf{s}^*_{tx}$ und $\mathbf{s}^*_{ty}$ sind bereits orthogonal zueinander und müssen daher nur noch normiert werden:

$$\mathbf{s}_{tx} = \frac{\mathbf{s}^*_{tx}}{\|\mathbf{s}^*_{tx}\|} \quad \text{und} \quad \mathbf{s}_{ty} = \frac{\mathbf{s}^*_{ty}}{\left\|\mathbf{s}^*_{ty}\right\|} \tag{3.10}$$

Anschließend kann als erstes die Komponente $\mathbf{s}_{scale}$ zur Beschreibung der Skalierung bestimmt werden. Diese entspricht dem auf den Formkomponenten $\mathbf{s}_{tx}$ und $\mathbf{s}_{ty}$ orthogonalen Anteil von $\mathbf{s}^*_{scale}$, wobei zusätzlich eine Normierung erfolgt:

$$\begin{aligned} \tilde{\mathbf{s}}_{scale} &= \mathbf{s}^*_{scale} - \frac{\mathbf{s}^T_{tx}\mathbf{s}^*_{scale}}{\mathbf{s}^T_{tx}\mathbf{s}_{tx}} \cdot \mathbf{s}_{tx} - \frac{\mathbf{s}^T_{ty}\mathbf{s}^*_{scale}}{\mathbf{s}^T_{ty}\mathbf{s}_{scale}} \cdot \mathbf{s}_{ty} \\ \mathbf{s}_{scale} &= \frac{\tilde{\mathbf{s}}_{scale}}{\|\tilde{\mathbf{s}}_{scale}\|} \end{aligned} \tag{3.11}$$

Analog wird die Formkomponente für die Rotation $\mathbf{s}_{rot}$ ermittelt. Dabei wird $\mathbf{s}^*_{rot}$ zu den restlichen drei Komponenten orthogonalisiert und normiert:

$$\begin{aligned} \tilde{\mathbf{s}}_{rot} &= \mathbf{s}^*_{rot} - \frac{\mathbf{s}^T_{scale}\mathbf{s}^*_{rot}}{\mathbf{s}^T_{scale}\mathbf{s}_{scale}} \cdot \mathbf{s}_{scale} - \frac{\mathbf{s}^T_{tx}\mathbf{s}^*_{rot}}{\mathbf{s}^T_{tx}\mathbf{s}_{tx}} \cdot \mathbf{s}_{tx} - \frac{\mathbf{s}^T_{ty}\mathbf{s}^*_{rot}}{\mathbf{s}^T_{ty}\mathbf{s}_{ty}} \cdot \mathbf{s}_{ty} \\ \mathbf{s}_{rot} &= \frac{\tilde{\mathbf{s}}_{rot}}{\|\tilde{\mathbf{s}}_{rot}\|} \end{aligned} \tag{3.12}$$

Als Ergebnis dieses Schritts stehen insgesamt vier synthetisch erzeugte Komponenten zur Verfügung. Eine Modellinstanz $\mathbf{S_M}$ innerhalb eines Inputbildes kann dann wie folgt darge-

stellt werden:

$$\mathbf{s} \approx \mathbf{s_0} + (p_{tx}\mathbf{s}_{tx} + p_{ty}\mathbf{s}_{ty} + p_{scale}\mathbf{s}_{scale} + p_{rot}\mathbf{s}_{rot}) + \sum_{i=1}^{N'} p_i\mathbf{s_i} + = \mathbf{s_0} + \mathbf{P} \cdot \mathbf{S}$$

mit (3.13)

$$\mathbf{P} = (p_{tx}, p_{ty}, p_{scale}, p_{rot}, p_1, ..., p_i)$$
$$\mathbf{S} = (\mathbf{s}_{tx}, \mathbf{s}_{ty}, \mathbf{s}_{scale}, \mathbf{s}_{rot}, \mathbf{s_1}, ..., \mathbf{s_i})$$

Diese von [Matthews and Baker, 2004] vorgeschlagene Variante der Realisierung der globalen Transformation des lokalen Formmodells in ein Inputbild kann also vollständig in die vorhandene Berechnungsvorschrift des Active Shape Models integriert werden. Die Modifikation des Anpassungsalgorithmus, wie im vorherigen Abschnitt, ist bei dieser Variante also nicht erforderlich. Daher wird im Rahmen dieser Dissertation diese Variante verwendet.

### 3.5.3 Modellanpassung

Als Basis für die Modellanpassung wird von einem vorhandenen Active Shape Model ausgegangen. Dieses besteht aus:

| | | |
|---|---|---|
| $\mathbf{s_0}$ | : | Mittelwert der normierten Formen |
| $\mathbf{s_i}$ | : | Eigenvektoren $i$ des Formmodells |
| $\mathbf{s}_{tx}$ | : | Komponente zur Beschreibung der Translation in x-Richtung |
| $\mathbf{s}_{ty}$ | : | Komponente zur Beschreibung der Translation in y-Richtung |
| $\mathbf{s}_{scale}$ | : | Komponente zur Beschreibung der Skalierung |
| $\mathbf{s}_{rot}$ | : | Komponente zur Beschreibung der Rotation |

O.B.d.A. seien im Folgenden die Komponenten $\mathbf{s}_{tx}$, $\mathbf{s}_{ty}$, $\mathbf{s}_{scale}$ und $\mathbf{s}_{rot}$ Bestandteil der $\mathbf{s_i}$.

Weiterhin sei die aktuelle Beobachtung $\mathbf{S_B}$ bekannt:

$$\mathbf{S_B} = \left(x_1^{(B)}, y_1^{(B)}, ..., x_n^{(B)}, y_n^{(B)}\right)^T \tag{3.14}$$

Der Fehler zwischen der aktuellen Modellkontur $\mathbf{S_M}$ und der Beobachtung $\mathbf{S_B}$ kann durch die folgende distanzminimierende Energiefunktion $E$ beschrieben werden:

$$\begin{aligned} E &= (\mathbf{S_B} - \mathbf{S_M})^T \mathbf{W} (\mathbf{S_B} - \mathbf{S_M}) \\ &= \left(\mathbf{S_B} - \mathbf{s_0} - \sum_{i=1}^{h} p_i\mathbf{s_i}\right)^T \mathbf{W} \left(\mathbf{S_B} - \mathbf{s_0} - \sum_{i=1}^{h} p_i\mathbf{s_i}\right) \end{aligned} \tag{3.15}$$

Dabei beschreibt $\mathbf{W}$ eine optionale Gewichtung der Stützstellen der Kontur.

Wird nun $E$ partiell nach $p_i$ abgeleitet und nach $p_i$ aufgelöst, lässt sich die Bestimmung der

optimalen Modellparameter $p_i$ in einem stark überbestimmten linearen Gleichungssystem der Form $\mathbf{Ax} = \mathbf{b}$ zusammenfassen:

$$\begin{aligned} \frac{\partial E}{\partial p_i} &= -2\mathbf{s_i}\mathbf{W}\left(\mathbf{S_B}^i - \mathbf{s_0}^i\right) \\ p_i &= -\frac{1}{\mathbf{s_i}}\left(\mathbf{S_B}^i - \mathbf{s_0}^i\right) \end{aligned} \tag{3.16}$$

Nun können die einzelnen $p_i$ in einem weiteren Gleichungssystem zusammengefasst werden:

$$\begin{pmatrix} \mathbf{s_1}^1 & \cdots & \mathbf{s_h}^1 \\ \vdots & \ddots & \vdots \\ \mathbf{s_1}^{2n} & \cdots & \mathbf{s_h}^{2n} \end{pmatrix} \begin{pmatrix} p_1 \\ \vdots \\ p_h \end{pmatrix} = \begin{pmatrix} \mathbf{S_{B}}1 - \mathbf{s_0}_1 \\ \vdots \\ \mathbf{S_{B}}2n - \mathbf{s_0}_{2n} \end{pmatrix} \tag{3.17}$$

Wobei der Parameter $h$ die Anzahl der Formkomponenten einschließlich der Komponenten zur Beschreibung der globalen Transformation ist. Das entstandene Gleichungssystem entspricht der Form $\mathbf{Ax} = \mathbf{b}$. Somit gilt

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$$

mit

$$\mathbf{A} = \begin{pmatrix} \mathbf{s_1}^1 & \cdots & \mathbf{s_h}^1 \\ \vdots & \ddots & \vdots \\ \mathbf{s_1}^{2n} & \cdots & \mathbf{s_h}^{2n} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{S_{B}}1 - \mathbf{s_0}_1 \\ \vdots \\ \mathbf{S_{B}}2n - \mathbf{s_0}_{2n} \end{pmatrix} \tag{3.18}$$

wobei $\mathbf{W}$ auf Grund der bestehenden Gleichgewichtung der Konturpunkte als Einheitsmatrix angenommen und entfernt wurde.

Mit Hilfe der aktuellen Beobachtung $\mathbf{S_B}$ können gemäß (3.18) somit die Modellparameter $p_i$ ermittelt werden. Wie im Abschnitt 3.5.2 erwähnt, erfolgt nach Berechnung der $p_i$ anschließend eine zusätzliche Plausibilitätsprüfung. Insofern der Parameter $p_i$ (mit zugehörigem Eigenwert $\lambda_i$) sich im Intervall $\left[-2\sqrt{\lambda_i}...+2\sqrt{\lambda_i}\right]$ befindet, kann von einem plausiblen/korrekten Wert ausgegangen werden [Cootes et al., 1995]. Überschreitet $p_i$ die Intervallgrenzen, ist das Modell sehr wahrscheinlich entartet oder deformiert. In diesem Fall wird $p_i$ auf die entsprechende Intervallgrenze gesetzt. Falls mehrere $p_i$ außerhalb der entsprechenden Grenzen liegen, kann man von einer falschen Beobachtung ausgehen. In diesem Fall sollte die gesamte aktuelle Schätzung verworfen und mit einer neuen Detektion begonnen werden.

Voraussetzung für die Anpassung ist die Kenntnis von $\mathbf{S_B}$. Im folgenden Abschnitt wird beschrieben, wie diese Beobachtung ermittelt werden kann.

### 3.5.4 Bestimmung der aktuellen Beobachtung

Basierend auf der aktuellen (Modell-)Schätzung soll eine neue Beobachtung im Bild ermittelt werden, die das Modell besser an den aktuellen Input anpasst. Dazu soll für jede Stützstelle der Kontur ein lokaler Bewegungsvektor bestimmt werden, der letztendlich zu einer verbesserten Anpassung des Modells an den aktuellen Input führt. Diese einzelnen Bewegungsvektoren werden in (3.18) durch den Vektor $\mathbf{b}$ zusammengefasst.

Im Rahmen dieser Dissertation werden zwei Verfahren aus der Literatur zur Bestimmung der aktuellen Beobachtung kombiniert: An jeder Stützstelle wird eine Grandientensuche entlang der Normalen bezüglich der lokalen Modellkontur [Cootes et al., 1995] mit einem Beobachtungsmodell aus dem *Contracting Curve Density* Algorithmus [Hanek and Beetz, 2004] kombiniert.

#### Grandientensuche entlang der Normalen

Mit der Vorstellung der Active Shape Models haben [Cootes et al., 1995] auch eine einfache aber effektive Variante zur Bestimmung der aktuellen Beobachtung präsentiert. Die Idee besteht darin, an jeder Stützstelle $\mathbf{x}_i$ der Modellinstanz zunächst die Normale $\mathbf{n}_i$ zu bestimmen und entlang dieser nach dem größten Gradientenbetrag $|g_{max}(\mathbf{n}_i)|$ im Bild zu suchen (siehe Abbildung 3.11). Die Bestimmung der Normalen $\mathbf{n}_i$ kann beispielsweise mittels einer B-Spline-Interpolation durch die benachbarten Stützstellen $\mathbf{x}_{i-1}$ und $\mathbf{x}_{i+1}$ erfolgen.

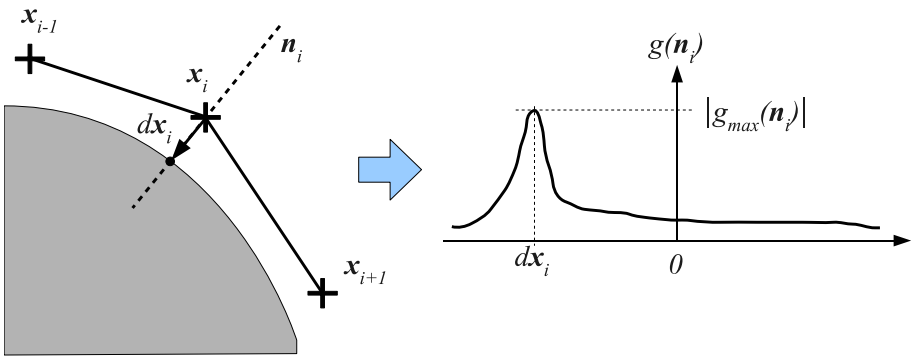



***Abbildung 3.11:*** *Entlang der Normalen $\mathbf{n}_i$ an der Stützstelle $\mathbf{x}_i$ wird nach dem größten Gradientenbetrag $|g_{max}(\mathbf{n}_i)|$ im Bild gesucht. Als Ergebnis entsteht ein Verschiebungsvektor $d\mathbf{x}_i$.*

Mit dieser einfachen Vorgehensweise können bereits sehr gute Ergebnisse erreicht werden. Bei stark strukturiertem Hintergrund und nicht perfekter Trennung von Vorder- und Hindergrund besteht jedoch die Gefahr, dass im Hintergrund stärkere Gradienten gefunden werden, als an der eigentlich gesuchten Objektkante. Je größer der Suchbereich entlang der Normalen $\mathbf{n}_i$ ist, desto höher ist jedoch auch die Gefahr von Fehldetektionen. Für einen großen

"Fangbereich" des Modells, ist ein großer Suchbereich entlang der Normalen aber notwendig. Dieses Problem kann abgemildert werden, indem bei der Bewertung auch die Gradientenrichtung betrachtet wird: Der gesuchte Gradient sollte im Idealfall senkrecht zur Normalen stehen. Diese Gewichtung kann z.B. mittels dem Skalarprodukt zwischen dem Einheitsvektor in Gradientenrichtung und der Normale erfolgen.

### CCD: Minimierung der Varianzen

Beim CCD-Algorithmus [Hanek and Beetz, 2004] kommt ein Verfahren zum Einsatz, dass anhand von Mittelwert und Streuung der Grauwertverteilung entlang einer Geraden den bestmöglichen Punkt zur Trennung von zwei (Teil-)Flächen gefunden werden soll. Dazu wird jeweils links und rechts eines Punktes $x$ auf der Geraden der Mittelwert $m(x)$ und die Streuung $\sigma(x)$ bezüglich des Grauwertes oder der Intensität $I(x)$ betrachtet. Dabei wird der Punkt $x$ gesucht, an dem die Energiefunktion

$$E(x) = \sqrt{(\sigma_l^2(x) + \sigma_r^2(x))} \tag{3.19}$$

minimal wird. Abbildung 3.12 veranschaulicht dies an einem Beispiel.

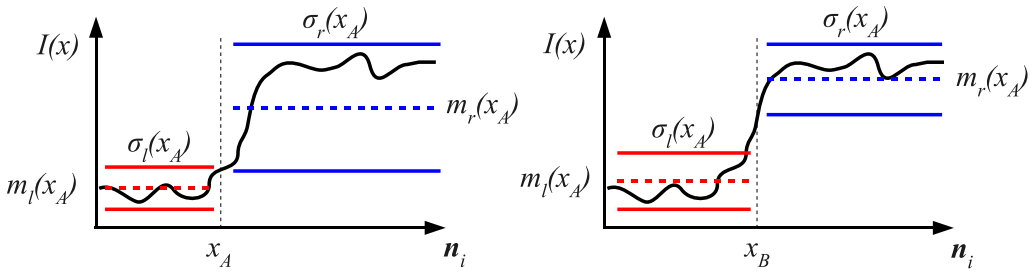



***Abbildung 3.12:*** *Entlang der Normalen $\mathbf{n}_i$ wird nach dem Punkt $x$ gesucht, bei dem auf beiden Seiten die Streuung $\sigma_l(x)$ und $\sigma_r(x)$ der Intensität $I(x)$ möglichst klein sind. Im linken Teil der Grafik ist $\sigma_l(x_A)$ zwar sehr klein, aber $\sigma_r(x_A)$ sehr groß. Im rechten Teil sind sowohl $\sigma_l(x_A)$ als auch $\sigma_l(x_B)$ relativ klein. Der Punkt $x_B$ beschreibt also den Gradientenübergang besser als der Punkt $x_A$.*

Der Nachteil dieses Verfahrens (gegenüber der einfachen Gradientensuche) ist der deutlich höhere Berechnungsaufwand, da für jede Normale der optimale Trennpunkt gefunden werden muss. Daher wurden im Rahmen dieser Dissertation beide Verfahren miteinander, wie im Folgenden beschrieben, kombiniert.

#### Kombination beider Varianten

Die Gradientensuche entlang der Normalen kann sehr schnell berechnet werden und liefert gute Ergebnisse, die jedoch stark von der Strukturierung der Flächen innerhalb und außerhalb der Kontur abhängig sein können. Die Suche des optimalen Trennpunktes basierend auf dem CCD-Algorithmus ist robuster gegenüber der Strukturierung der Teilflächen, verursacht jedoch einen deutlich höheren Berechnungsaufwand. Daher sollen die Vorteile beider Verfahren kombiniert werden, um die gesuchte Objektkante im Bild möglichst schnell und robust gegenüber Störungen im Hintergrund zu finden.

Hierfür werden zuerst eine kleine Anzahl von Punkten $x_j$ auf der Normalen als Kandidaten ausgewählt und diese anschließend bewertet:

- Die Auswahl der Punkte $x_j$ erfolgt anhand des Gradientenbetrages. Der Gradientenbetrag $g(x_j)$ wird für alle Punkte der Normalen mit Hilfe eines Sobel-Filters mit einer 5x5- oder 7x7-Maske berechnet.

- Für alle Punkte mit einer hohen Filterantwort ($g(x_j) > g_{threshold}$) werden anschließend gemäß dem Verfahren aus dem CCD-Algorithmus die Werte der Energiefunktion $E(x_j)$ berechnet. Der Wert $g_{threshold}$ kann dabei fest eingestellt sein oder variabel so adaptiert werden, dass z.B. immer 10% der Punkte näher betrachtet werden.

Durch die Vorauswahl der Kandidatenpunkte kann der Berechnungauswand für die CCD-Funktion erheblich reduziert werden. Als Ergebnis der Suche wird letztendlich der Punkt $x_j$ mit minimalem $E(x_j)$ ausgewählt.

## 3.6 Ergebnisse

Im folgenden Abschnitt werden die erzielten Ergebnisse bei der Schätzung der Oberkörperpose beschrieben. Dazu erfolgt zunächst eine Beschreibung des verwendeten Koordinatensystems, der Datensätze und anschließend die Erläuterung der verschiedenen Ergebnisse mit verschiedenen Klassifikatoren.

### 3.6.1 Koordinatensystem

Für die nachfolgenden Erläuterungen der Ergebnisse zur Schätzung der Oberkörperpose wird der Drehwinkel $\phi$ der Person in Bezug auf die Kamera in Draufsicht in mathematisch positiver Richtung betrachtet. Ein Winkel von $\phi = 0°$ bedeutet, dass die Person direkt in die Kamera blickt, $\phi = -90°$ für eine Person, die nach links schaut und $\phi = 90°$ für eine Person, die nach rechts schaut. Ein Winkel von $\phi = 180° \widehat{=} -180°$ bedeutet, dass die Person mit dem Rücken zur Kamera steht (siehe Abbildung 3.13).

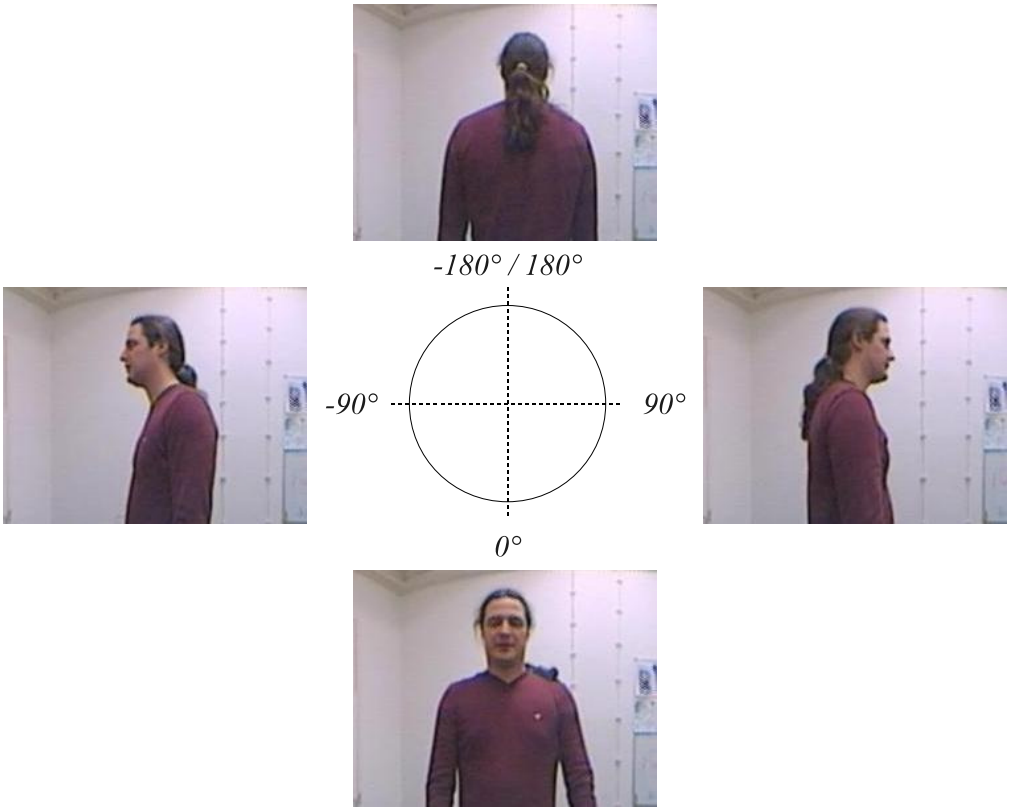



***Abbildung 3.13:*** *Das Koordinatensystem für die Erkennung der Oberkörperpose: Ein Winkel von $\phi = 0°$ bedeutet, dass die Person direkt in die Kamera schaut. Der Winkel ist mathematisch positiv orientiert.*

Bei der Betrachtung der Oberkörperorientierung einer Person in zwei Richtungen, die symmetrisch zur $-90°...90°$-Achse liegen, ist eine eindeutige Bestimmung anhand der Silhouette oder der Kontur nur sehr schwierig. Der Grund hierfür sind teilweise nur minimale Unterschiede in der Silhouette bzw. der Kontur, die durch perspektivische Verzerrung verursacht werden. Abbildung 3.14 zeigt von zwei Personen je zwei symmetrische Beispiele.

Da hier bildlich eine räumliche Spiegelung an der $-90°...90°$-Achse stattfindet, wird im Folgenden in einen *Vorderen Winkelbereich* (Intervall von $[-90°...+90°]$, Gesicht der Person zeigt in den Halbraum in Richtung Kamera) und in einen *Hinteren Winkelbereich* (Gesicht der Person zeigt von der Kamera weg) unterschieden.

Da durch die nur minimalen Unterschiede in der Silhouette die Trennung von vorderem und hinterem Winkelbereich sehr schwierig ist, werden bei der Durchführung der Experimente zur Schätzung der Oberkörperorientierung zwei Koordinatensysteme verwendet: einerseits ein vollständiges 360°-System und andererseits ein 180°-System:

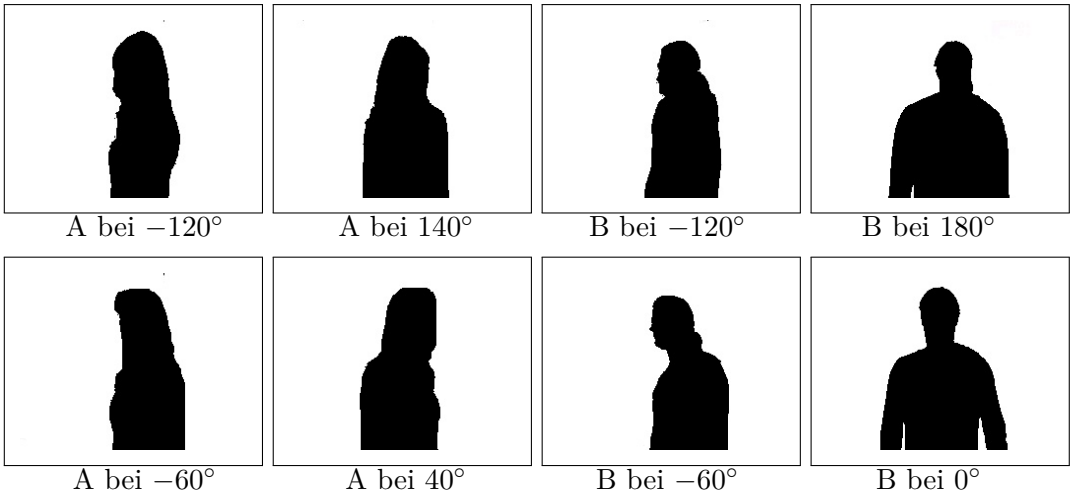



***Abbildung 3.14:*** *Beispiele von Silhouetten bei Oberkörperorientierung symmetrisch zur $-90°...90°$-Achse: In den ersten drei Spalten ist eine Unterscheidung bzw. Erkennung als menschlicher Betrachter gerade noch möglich. In der letzten Spalte nicht.*

- Im ersten Fall ist das Ziel die Schätzung des Winkels im Intervall $[-180°... + 180°]$. Hierbei muss der eingesetzte Klassifikator mit den minimalen Unterschieden in der Silhouette bzw. der Kontur umgehen.

- Im zweiten Fall erfolgt die Schätzung lediglich im vorderen Winkelbereich im Intervall $[-90°... + 90°]$. Daten aus dem hinteren Winkelbereich werden in den vorderen gespiegelt. In einem separaten Schritt erfolgt zusätzlich eine Erkennung, ob die Kontur im vorderen oder hinteren Winkelbereich liegt.

  Für die Unterscheidung, ob eine Person eher mit dem Gesicht oder eher mit dem Rücken zum Roboter steht, kann insbesondere der Bereich des Gesichts verwendet werden. Dieser ist auf Basis der geschätzten Kontur leicht zu bestimmen. Als Merkmal kann beispielsweise der Anteil an Hautfarbe im Bereich des Gesichts verwendet werden. Die Hautfarbe kann mit einem geeigneten Modell im zweidimensionalen *r-g-Farbraum* wie in [Wilhelm et al., 2003] einfach und ausreichend robust klassifiziert werden.

  Als Alternative könnte auch ein poseninvarianter Gesichtsdetektor eingesetzt werden. Im Rahmen dieser Dissertation wird aus Gründen der Performance und Robustheit jedoch die Hautfarbe als Entscheidungskriterium eingesetzt.

### 3.6.2 Daten zur Modellerstellung und Auswertung

#### Modellerstellung

Für die Modellerstellung wurde eine Datenbank bestehend aus 250 Bildern verwendet. Die Datenbank enthält Bilder von 10 verschiedenen Personen (6 männlich, 4 weiblich) mit jeweils 25 Aufnahmen. Die Bilder zeigen die Personen in Alltagsbekleidung vor strukturiertem Hintergrund mit verschiedener Orientierung des Oberkörpers. Abbildung 3.15 zeigt einige Beispiele.

***Abbildung 3.15:*** *Einige Beispielbilder aus der Datenbank zur Modellerstellung.*

In den Bildern dieser Datenbank wurde manuell jeweils die Kontur des Oberkörpers mit einem 17-Punkte-Modell gelabelt. Da das Modell und das zugrunde liegende Problem symmetrisch sind, konnte aus jedem Datensatz ein weiterer vertikal gespiegelter Datensatz erzeugt werden. Für die Modellerstellung standen somit 500 Datensätze zur Verfügung. Der Aufbau des Modells erfolgte gemäß dem in Abschnitt 3.5.2 vorstelltem Verfahren.

Bei einer gewünschten Modellabdeckung von 95% bezüglich der Trainingsdaten entstand ein Modell mit lediglich sechs lokalen Formparametern. Abbildung 3.16 zeigt diese Komponenten.

#### Testdaten

Für die anschließenden Tests wurde eine weitere Datenbank verwendet. Die Datenbank enthält Bilder von 15 Personen (5 männlich, 10 weiblich), die in Winkelschritten von 20° vor der Kamera stehen. Für jede Person enthält die Datenbank zwei vollständige Umdrehungen. Also insgesamt 36 Bilder pro Person. Bei drei Personen wurde ein Satz Bilder in Schritten von 10° aufgenommen. Abbildung 3.13 zeigt ausgewählte Beispielbilder einer Person.

Für alle Bilder der Datenbank wurde das erstellte Active Shape Modell gemäß Kapitel 3.5.3 angepasst. Anschließend wurden manuell die Daten herausgefiltert, bei denen die Anpassung zu ungenau war. Als Ergebnis standen 530 Datensätze zur Auswertung für die verschiedenen Klassifikatoren und Funktionsapproximatoren zur Verfügung.


***Abbildung 3.16:*** *Resultierende Komponenten des Active Shape Models bei 95% Abdeckung der Trainingsdatenbank. Dargestellt sind alle sechs Komponenten $s_i$ des Modells mit jeweils deren Auswirkung auf die Grundform $\mathbf{s_0}$ im Intervall $p_i \in \left[-2\sqrt{\lambda_i}, 2\sqrt{\lambda_i}\right]$. Die mittlere Spalte zeigt jeweils die mittlere Form $\mathbf{s_0} = \overline{\mathbf{z}}$.*

### 3.6.3 FeatureSelection mittels *Mutual Information*

Wie bereits in Abbildung 3.16 zu erkennen ist, lassen sich aus den verschiedenen Formkomponenten des Modells mehr oder weniger Aussagen über die gesuchte Oberkörperorientierung entnehmen. Einige Komponenten beschreiben z.B. lediglich die allgemeine Statur der Person und enthalten keine Information über die Oberkörperorientierung. Um einen nachgeschalteten Klassifikator oder Funktionsapproximator nur mit relevanten Daten zu verwenden und nicht mit irrelevanten Daten zu "belasten", erfolgt zunächst die Auswahl der relevanten Parameter (*Feature Selection*). Dabei wird im Rahmen dieser Dissertation die *Mutual Information for Feature Selection (MIFS)* benutzt. Bei der MIFS handelt es sich um ein Verfahren aus der Informationstheorie zur Bestimmung der Relevanz von Parametern eines Klassifikations- oder Funktionsapproximationsproblems. Weitere Details zur MIFS finden sich in Anhang D.1.

Als Merkmal für die MIFS wurde der aus den Trainingsdaten bekannte Winkel der Oberkörperpose verwendet. Als Indikator, ob ein MIFS-Wert eines bestimmten Kanals ein "guter" Wert ist, wird zusätzlich die MIFS für einen Kanal mit Zufallsdaten berechnet. Alle Parameter, die einen kleineren MIFS-Wert als dieser Rauschkanal haben, enthalten keinerlei Information zum gesuchten Problem und können bei der anschließenden Klassifikation ignoriert werden. Für das 180°- und 360°-System wurden folgende Werte für die MIFS bestimmt:

| 180°-System | | | 360°-System | |
|---|---|---|---|---|
| Parameter | MIFS | | Parameter | MIFS |
| $p_0$ | 0.551347 | | $p_3$ | 0.840869 |
| $p_3$ | 0.466893 | | $p_0$ | 0.733290 |
| $p_1$ | 0.322254 | | $p_1$ | 0.434766 |
| $p_4$ | 0.066021 | | $p_4$ | 0.128036 |
| | | | $p_5$ | 0.063656 |

***Tabelle 3.2:*** *Werte der MIFS der einzelnen Modellparameter $p_i$ für das 180°-System und 360°-System. Nicht aufgelistete Parameter haben weniger Relevanz/Signifikanz als ein künstlich hinzugefügter Rauschkanal.*

Bei beiden Varianten haben die Parameter $p_0$, $p_1$ und $p_3$ die größte Relevanz. Lediglich die Reihenfolge und Höhe der Gewichtung unterscheiden sich. Die Parameter $p_4$ und $p_5$ sind deutlich geringer bewertet und können daher im weiteren Verlauf der Ergebnisbetrachtung vernachlässigt werden. Die nicht genannten Parameter enthalten keinerlei weitere relevante Informationen über die Orientierung der Person und entfallen ebenso. Das ursprünglich 6-dimensionale Problem konnte somit auf ein 3-dimensionales Problem reduziert werden.

Abbildung 3.17 und 3.18 zeigen den Verlauf der ermittelten signifikanten Parameter in Bezug auf die Orientierung des Oberkörpers. Es ist ersichtlich, dass die Parameter $p_0$, $p_1$ und $p_3$

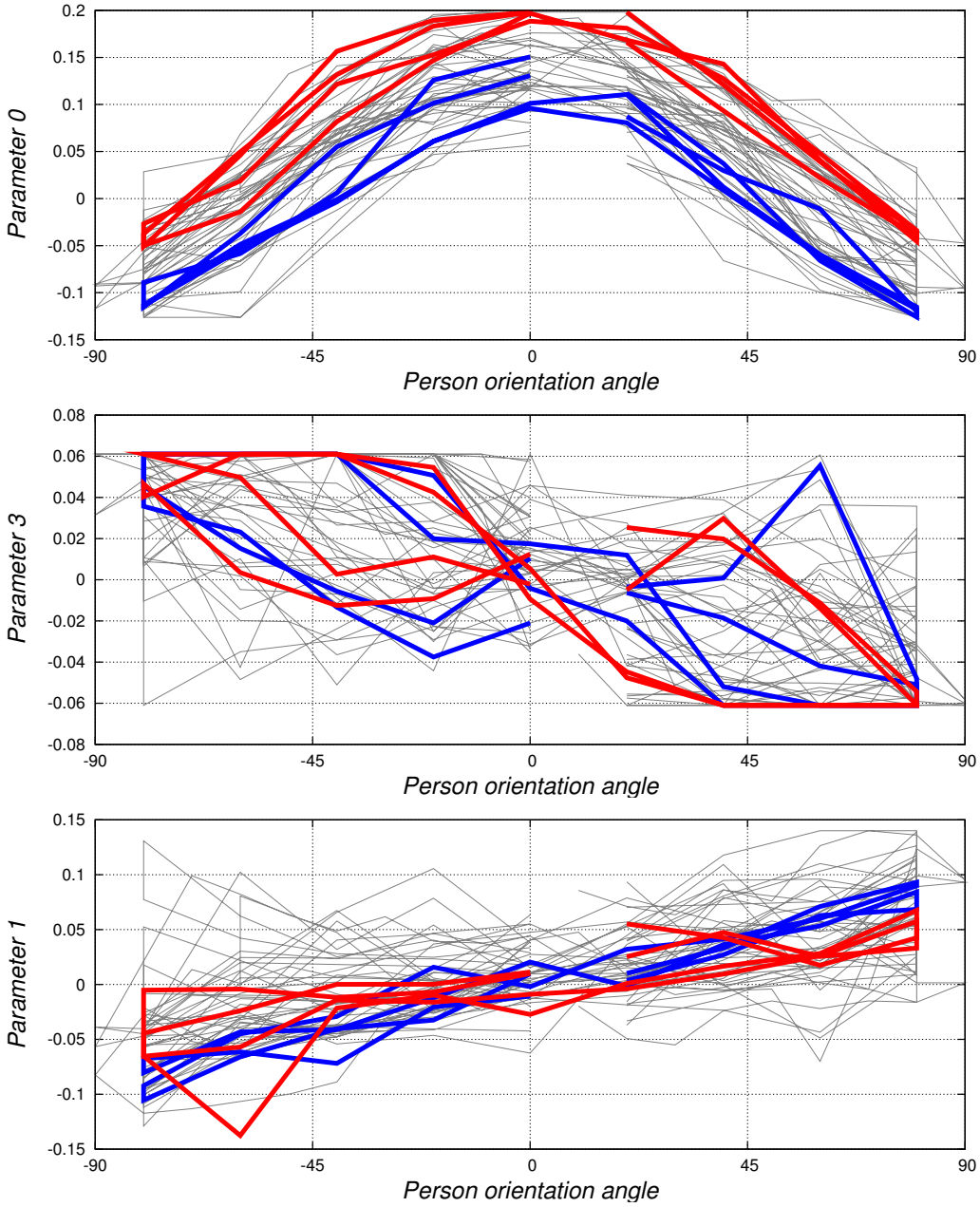



***Abbildung 3.17:*** *Darstellung der gemäß MIFS relevanten Parameter beim* 180°*-System. Abgebildet sind jeweils der Werteverlauf des Parameters $p_i$ in Bezug auf die Orientierung des Oberkörpers. Farblich hervorgehoben sind zwei verschiedene Testpersonen. Neben einem prinzipiell recht deutlichen funktionalen Zusammenhang zwischen der Orientierung und den Parametern $p_i$ sind auch klare Ausreißer zu erkennen.*

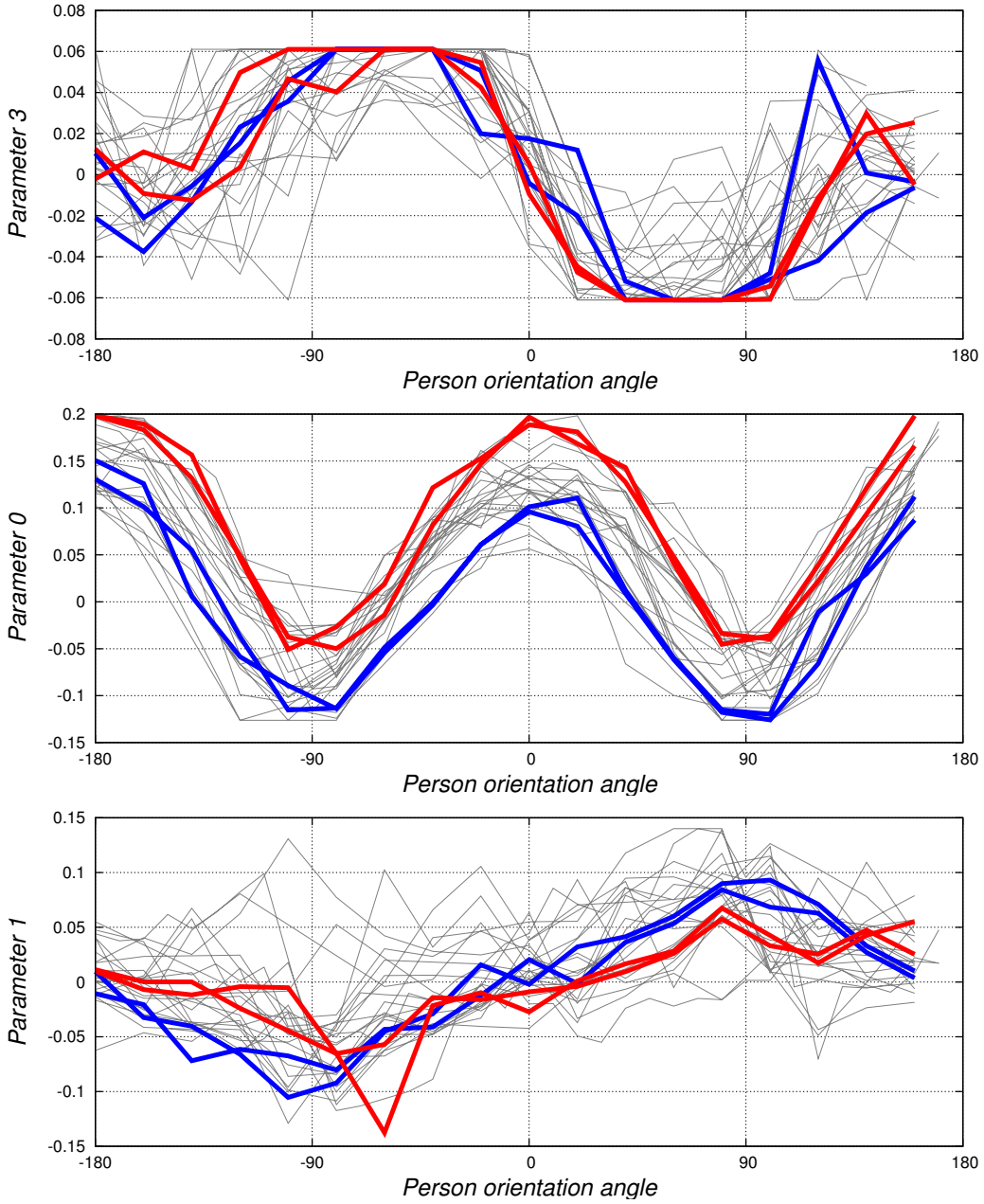



***Abbildung 3.18:*** *Darstellung der gemäß MIFS relevanten Parameter beim 360°-System. Abgebildet sind jeweils der Werteverlauf des Parameters $p_i$ in Bezug auf die Orientierung des Oberkörpers. Farblich hervorgehoben sind zwei verschiedene Testpersonen. Neben einem prinzipiell recht deutlichen funktionalen Zusammenhang zwischen der Orientierung und den Parametern $p_i$ sind auch klare Ausreißer zu erkennen.*

auch sinnvoll interpretiert werden können (siehe dazu auch Abbildung 3.16):

- $p_0$: "Breite" der Person (körperliche Statur und Orientierung der Person zur Kamera)
- $p_1$: Kopfneigung in Kombination mit der Armhaltung
- $p_3$: Kopfrichtung (insbesondere Kinn und Hinterkopf)

Für die folgenden Auswertungen wurde nur noch der reduzierte Parametervektor $\mathbf{p} = (p_0, p_1, p_3)$ verwendet. Die restlichen Parameter werden im Rahmen der Modellanpassung zwar bestimmt, bei der Schätzung der Oberkörperorientierung dann aber ignoriert, da sie keine relevanten Informationen enthalten.

### 3.6.4 Schätzung mittels *linearer Regression*

Als Vergleichsgrundlage für die Schätzung der Oberkörperorientierung wurde als erstes eine *lineare Regression* auf dem reduzierten Parametervektor $\mathbf{p} = (p_0, p_1, p_3)$ durchgeführt.

Für das 180°- und 360°-System wurden dazu jeweils die Koeffizienten $c_0$, $c_1$ und $c_3$ so bestimmt, dass:

$$\sum_{j=1}^{N} \left| \phi^{(j)} - \left( c_0 \cdot p_0^{(j)} + c_1 \cdot p_1^{(j)} + c_3 \cdot p_3^{(j)} + c \right) \right|^2 \rightarrow min \tag{3.20}$$

wobei $N$ die Anzahl der Datensätze, $\phi^{(j)}$ die bekannte Orientierung aus der Trainingsdatenbank und $p_i^{(j)}$ die Parameter des Datensatzes $j$ sind.

Anschließend wurde eine Schätzung der Oberkörperorientierung mit Hilfe der berechneten Koeffizienten durchgeführt und für jeden Datensatz die Abweichung $\Delta\phi$ bestimmt. Die Verteilung der Abweichungen $\Delta\phi$ für das 180°- und 360°-System ist in Abbildung 3.19 dargestellt. Es ist deutlich zu erkennen, dass der größte Teil der Fehler beim 180°-System kleiner gleich 40° ist. Bezogen auf die Testdatenbank (die in 20° Schritten aufgenommen wurde) bedeutet dies, dass der Winkel oft korrekt bzw. nur ein Intervall daneben geschätzt wurde. Größere Fehler kommen deutlich seltener bis gar nicht vor.

Diese Ergebnisse zeigen, dass eine gute Schätzung der Oberkörperorientierung im 180°-System bereits durch eine einfache lineare Approximation möglich ist. Beim 360°-System sind dagegen noch deutlich größere Abweichungen zu verzeichnen.

### 3.6.5 Schätzung mittels *k-Nearest-Neighbour*

Als nächstes Verfahren zur Schätzung der Oberkörperpose anhand des reduzierten Parametervektors $\mathbf{p} = (p_0, p_1, p_3)$ wurde ein modifizierter *k-Nearest-Neighbour*-Algorithmus

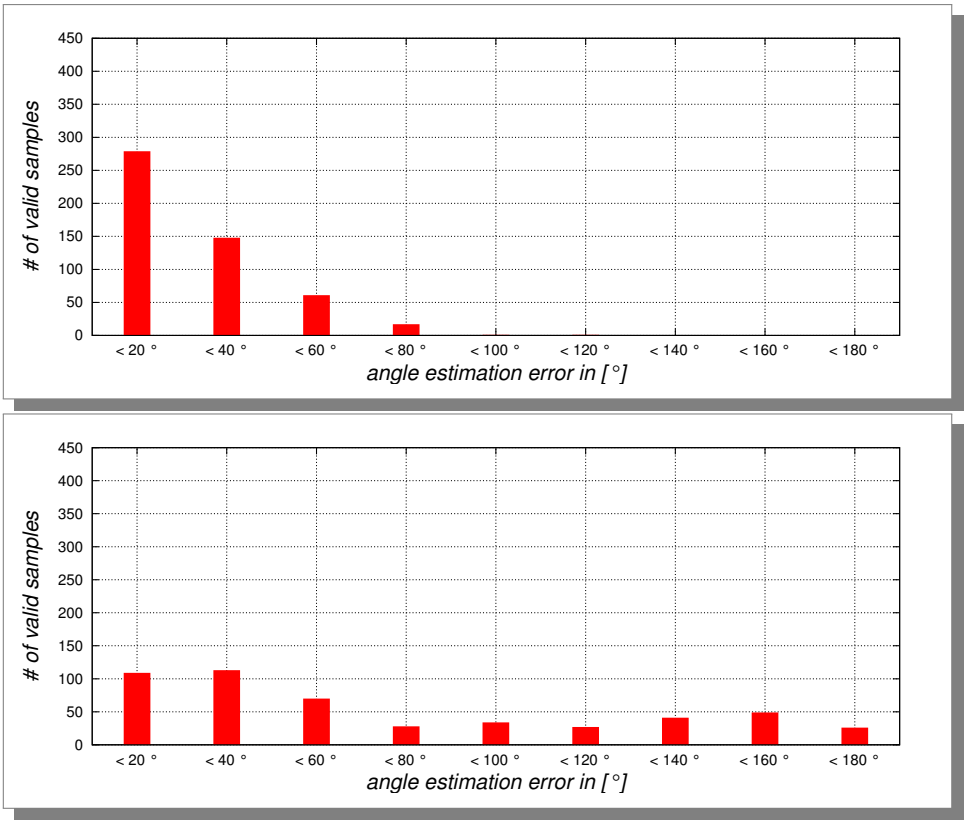



***Abbildung 3.19:*** *Fehlerhistogramme im* 180°*-System (oben) und* 360°*-System (unten) bei Schätzung der Oberkörperorientierung mit Hilfe linearer Regression. Auf Grund der Symmetrie ist beim* 360°*-System für Fehler* $> 90°$ *ein Anstieg der Fehlerhäufigkeit zu verzeichnen. (Das* 20°*-Intervall ist bedingt durch die zugrunde liegenden Testdaten.)*

eingesetzt. Anstatt einer reinen Klassifikation wird eine Funktionsapproximation realisiert, in dem das Ergebnis als gewichteter Mittelwert der Klassen der $k$ nächsten Nachbarn gebildet wird. Als Gewichtung wird hierbei $1/d_i$ verwendet, wobei $d_i$ der euklidische Abstand des Nachbarn $i$ zum aktuellen Input $\mathbf{p}$ ist. Weitere Details hierzu finden sich in Anhang A.1.

Da eine $k$-Nearest-Neighbour-Klassifikation nur sinnvoll funktioniert, wenn benachbarte Klassen bzw. Funktionswerte (hier: Winkel) auch im Parameterraum benachbart sind, wurde zunächst der Parameterraum näher untersucht. Dazu wurden jeweils die Mittelwerte $\mathbf{m}_i$ aller Parametervektoren $\mathbf{p}_j$ der Testdatensätze $\mathbf{t}_j$ ermittelt, die zu einem konkreten Funkti-

onswert $\phi(\mathbf{t}_j)$ (also einer Winkelklasse) gehören:

$$\mathbf{m}_i = \frac{1}{|\mathbf{C}_i|} \sum_{j \in \mathbf{C}_i} \mathbf{p}_j \qquad \mathbf{C}_i = \{k \,|\phi(\mathbf{t}_k) = i\} \qquad i = ..., -20°, 0°, 20°, ... \tag{3.21}$$

Anschließend wurden die Abstände von $\mathbf{m}_i$ zu allen $\mathbf{m}_j$ mit $j \neq i$ bestimmt. Somit ergibt sich eine Abstands-Matrix, die für beide Systeme in Abbildung 3.20 dargestellt ist. Es ist deutlich zu erkennen, dass in beiden Varianten benachbarte Winkelklassen auch ähnliche Parametersätze (also geringe Abstände im Parameterraum) haben. Weiterhin treten im 180°-System kaum Mehrdeutigkeiten auf. Die einzelnen Winkelklassen können hier recht gut voneinander unterschieden werden. Dagegen sind im 360°-System die Mehrdeutigkeiten auf Grund der Symmetrie zur $-90°...90°$-Achse deutlich zu erkennen.

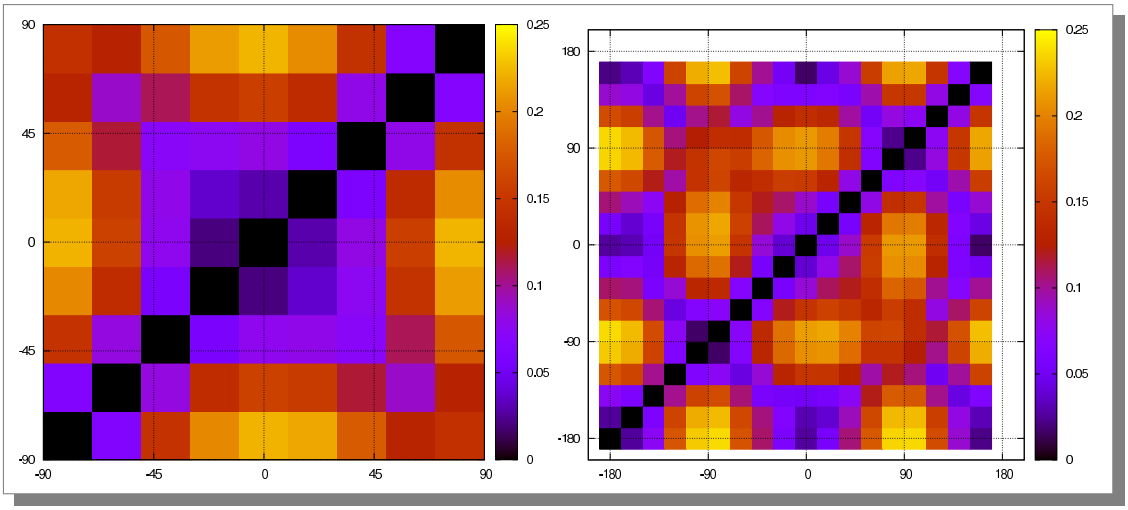



***Abbildung 3.20:*** *Darstellung der Abstände der Klassenzentren* $\mathbf{m_i}$ *der Trainingsdaten in* 20° *Schritten beim* 180°*-System (oben) und* 360°*-System (unten). Im unteren Fall treten deutlich mehr Ähnlichkeiten bzw. Mehrdeutigkeiten als im oberen Bild auf.*

Anschließend wurde eine $k$-Nearest-Neighbour Klassifikation auf sämtlichen Datensätzen der Testdatenbank (siehe Abschnitt 3.6.2) durchgeführt. Da die Testdatenbank lediglich aus 530 Datensätzen besteht, wurden als bekannte Stützstellen im Parameterraum lediglich die nach (3.21) ermittelten Klassenmittelpunkte $\mathbf{m}_i$ verwendet.

Für jede Person in der Testdatenbank wurde nacheinander für jedes Bild einer vollen Umdrehung vor der Kamera die $k$-Nearest-Neighbour-Klassifikation durchgeführt und somit die Schätzung für die Oberkörperorientierung $\phi$ ermittelt. Betrachtet man dies als zeitliche Abfolge, kann somit die geschätzte Oberkörperorientierung bezüglich der Zeit bzw. der Umdrehung der Person vor der Kamera grafisch dargestellt werden.

Abbildung 3.21 zeigt die erzielten Ergebnisse (für $k = 2$) für eine ausgewählte Testperson (weitere Ergebnisse finden sich im Anhang E.1). Im Idealfall würden alle Ergebnispunkte der Schätzung (rote Kreuze) auf der gestrichelten Linie (*Ground Truth*) liegen. Bei beiden Winkelsystemen ist ersichtlich, dass ein solch einfacher $k$-Nearest-Neighbour-Klassifikator bereits prinzipiell in der Lage ist, die Oberkörperorientierung basierend auf den Formparametern $p_0$, $p_1$ und $p_3$ grob zu schätzen. Es gibt jedoch in beiden Systemen auch deutliche Ausreißer und Fehlschätzungen. Auf Grund der Symmetrie zur $-90°...90°$-Achse sind beim 360°-System mehr Fehler zu finden: Beispielsweise werden viele Winkel nahe 0° auf Werte bei $-180°$ geschätzt. Im 180°-System treten solche Fehler nicht auf.

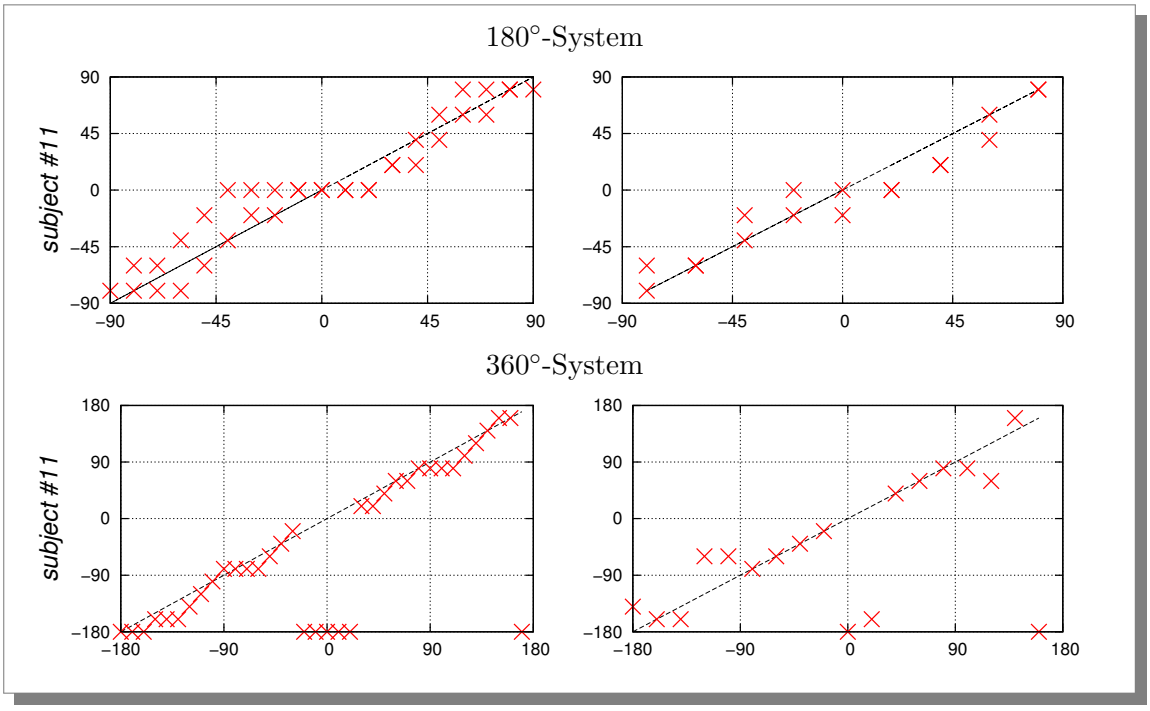



***Abbildung 3.21:*** *Ergebnisse der Oberkörperschätzung mittels Nearest Neighbour im* 180°*-System (oben) und im* 360°*-System (unten) für eine ausgewählte Testperson.*

Weiterhin wurde untersucht, wie sich die Auswahl der benutzten Parameter $p_i$ auf die Winkelschätzung auswirkt. Tabelle 3.3 zeigt die vollständigen Ergebnisse für beide Systeme in Abhängigkeit des maximal erlaubten Schätzfehlers $\Delta\phi_{max}$ und der benutzten Parameter. Für praktische Anwendungen ist ein maximaler Fehler von 30° sinnvoll. Für eine noch gröbere Schätzung kann auch eine maximale Abweichung von 60° benutzt werden.

| | 180°-System | | | 360°-System | |
|---|---|---|---|---|---|
| Parameter | $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ | | $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ |
| $p_0$ | 60.3% | 71.1% | | 35.6% | 45.7% |
| $p_1$ | 56.7% | 86.4% | | 37.7% | 61.7% |
| $p_3$ | 54.0% | 88.3% | | 40.5% | 68.4% |
| $p_0 + p_1$ | 80.4% | 89.1% | | 54.0% | 65.4% |
| $p_0 + p_3$ | 83.0% | 91.3% | | 65.4% | 73.5% |
| $p_1 + p_3$ | 69.2% | 96.2% | | 59.5% | 77.9% |
| $p_0 + p_1 + p_3$ | 87.7% | 95.5% | | 72.3% | 79.6% |

***Tabelle 3.3:*** *Ergebnisse des k-Nearest-Neighbour Klassifikators für das* 180°- *und* 360°-*System in Abhängigkeit eines maximalen Schätzfehlers* $\Delta\phi_{max}$ *und der benutzten Parameter.*

Die Ergebnisse zeigen, dass die Merkmalsauswahl mittels MIFS die relevanten Parameter ermittelt hat (siehe dazu auch Tabelle 3.2). Weiterhin ist ersichtlich, dass basierend auf einem einzelnen der drei Parameter allein keine Winkelschätzung mit akzeptablem Fehler erreicht werden kann. Mit Hilfe von zwei oder allen drei Parametern können signifikant bessere Ergebnisse erreicht werden.

Die Verteilung der Abweichungen $\Delta\phi$ für beide Varianten bei Verwendung aller drei Parameter ist in Abbildung 3.22 dargestellt. Es ist deutlich zu erkennen, dass der größte Teil der Fehler kleiner gleich 20° ist. Größere Fehler kommen deutlich seltener vor. Auf Grund der Symmetrie ist beim 360°-System bei $\Delta\phi > 90°$ ein Anstieg der Fehlerhäufigkeit zu verzeichnen. Beim 180°-System nimmt die Anzahl der Fehler mit steigender Abweichung $\Delta\phi$ kontinuierlich ab.

Im Vergleich zur linearen Regression sind die Ergebnisse im 180°-System fast identisch. Im 360°-System konnte jedoch eine deutliche Verbesserung erzielt werden.

### 3.6.6 Schätzung mittels *Multi Layer Perceptron*

Bei einem *Multi Layer Perceptron (MLP)* handelt es sich um ein mehrschichtiges, einfach verkoppeltes, vorwärts-gerichtetes, neuronales Netzwerk, dass für Klassifkationsaufgaben oder zur Funktionsapproximation verwendet werden kann. Bei einem MLP kommt ein überwachtes Lernverfahren (z.B. Back-Propagation) zum Einsatz. Weitere Details zum MLP finden sich im Anhang A.2.

Zur Vermeidung einer Überspezialisierung wurde beim Training das bekannte Verfahren der *Cross Validation* eingesetzt. Der vorhandene Datensatz wurde dazu in drei Teile zerlegt: Trainingsdatensatz (z.B. $\approx$ 70%), Validierungsdatensatz (z.B. $\approx$ 15%) und Testdatensatz

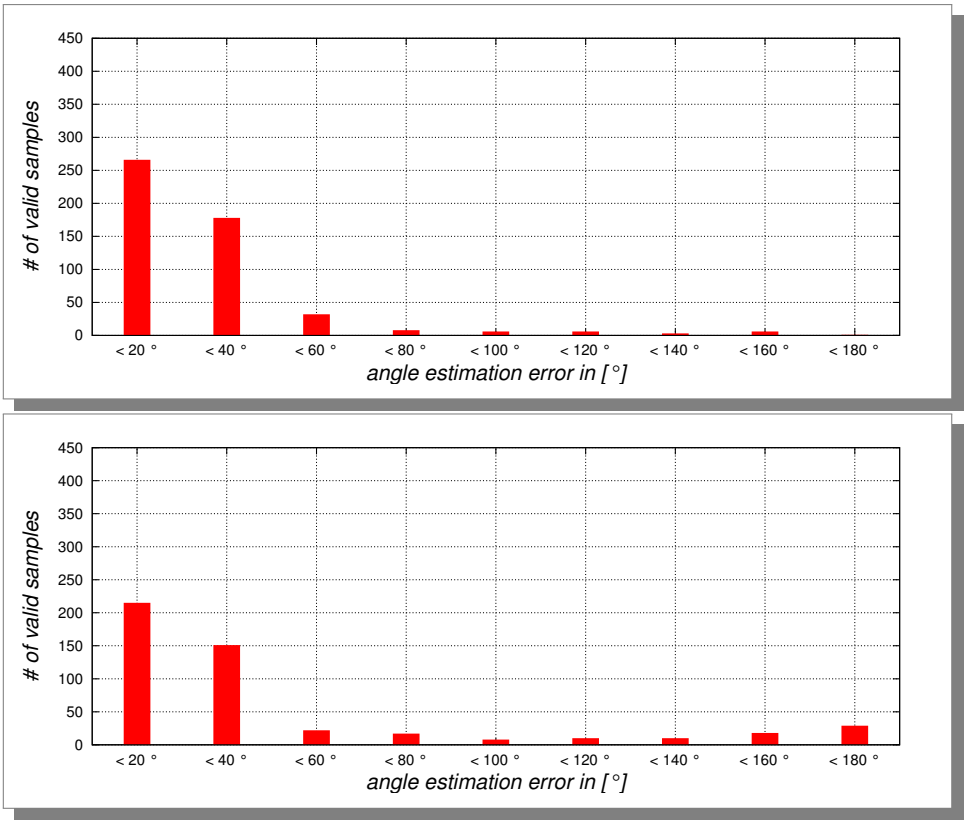



***Abbildung 3.22:*** *Fehlerhistogramme im 180°-System (oben) und 360°-System (unten) bei Verwendung eines k-Nearest-Neighbour Klassifikator. Auf Grund der Symmetrie ist beim 360°-System für Fehler > 90° ein Anstieg der Fehlerhäufigkeit zu verzeichnen. (Das 20°-Intervall ist bedingt durch die zugrunde liegenden Testdaten.)*

(z.B. $\approx 15\%$). Als Abbruchkriterium des Trainings wurde ein steigender Netzwerkfehler auf dem Validierungsdatensatz verwendet.

Zur Schätzung der Oberkörperorientierung wird als Input der reduzierte Parametervektor $\mathbf{p} = (p_0, p_1, p_3)$ verwendet. Dabei werden die drei Parameter $p_0$, $p_1$ und $p_3$ jeweils noch auf das Intervall $[-1...1]$ normiert.

In der Ausgabeschicht des MLP wurde eine "Fuzzy-artige"-verteilte Ausgabekodierung wie z.B. in [Pomerleau, 1989] verwendet. Dabei wird ein skalarer Wert $y$ mit Hilfe von $k$ Bins der Breite $w = \frac{1}{k} \cdot (y_{max} - y_{min})$ dargestellt. Jeder Bin kann dabei Werte im Bereich von 0 bis 1 annehmen. Bei der Erstellung der Teacher-Vektoren $\mathbf{t}_i$ kann eine gaußförmige Verteilung über die Bins gelegt werden, wobei das Maximum beim Bin $\lfloor y/w \rfloor$ liegt. Praktisch ist es jedoch schon ausreichend, beim Bin $\lfloor y/w \rfloor$ den Wert 1.0 und jeweils links und rechts

davon den Wert 0.5 einzutragen. Für die Bestimmung eines Wertes $y$ aus einer gegebenen Verteilung über den Bins gibt es verschiedene Vorgehensweisen. Beispielsweise könnte eine Gaußverteilung eingepasst werden. Der gesuchte Wert $y$ ergibt sich dann aus der Position des Maximums der Verteilung. In praktischen Fällen genügt es jedoch oft schon, den Bin $b_j$ mit maximalem Wert zu finden. Anhand des Wertes von $b_j$ und der benachbarten Bins $b_{j+1}$ und $b_{j-1}$ kann dann eine Approximation von $y$ erfolgen.

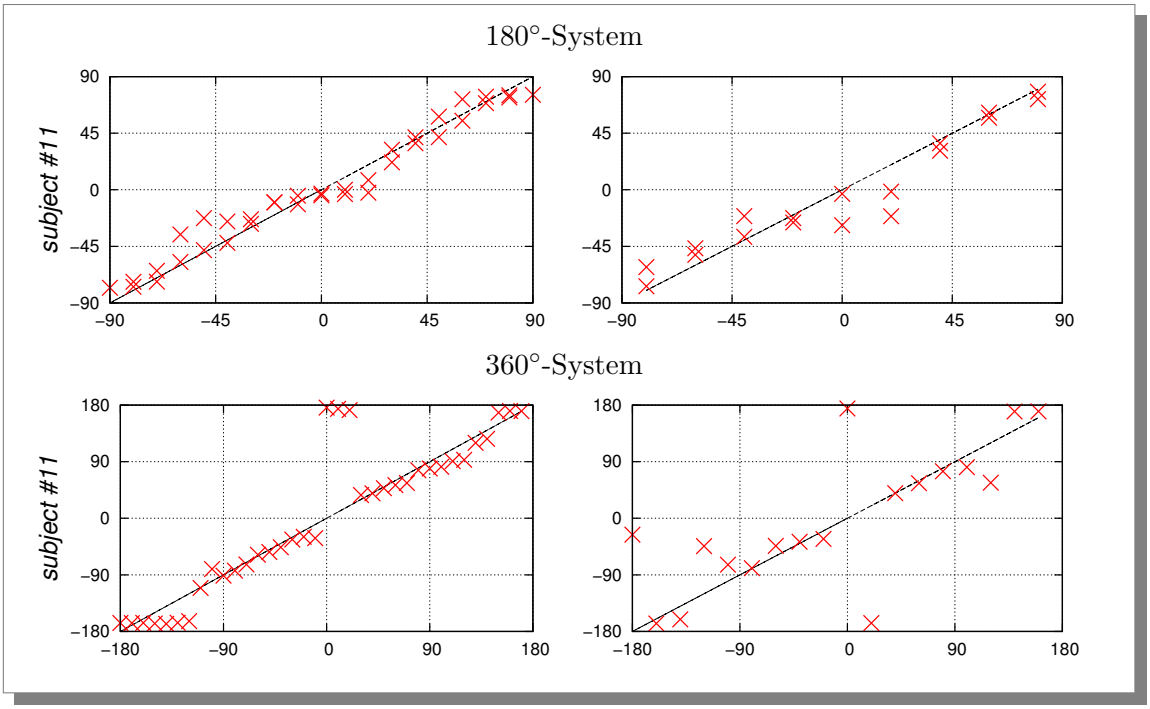



***Abbildung 3.23:*** *Ergebnisse der Oberkörperschätzung mittels MLP im 180°-System (oben) und im 360°-System (unten) für eine ausgewählte Testperson.*

Sowohl beim 180°-System, als auch beim 360°-System besteht die Inputschicht aus drei Neuronen. Als Aktivierungsfunktion wurde die symmetrische Sigmoid-Funktion ausgewählt.

Beim 180°-System wurde eine Ausgabeschicht mit Bins von 40° verwendet. Da es sich beim 180°-System nicht um einen geschlossenen Kreis handelt, wurde jeweils am Rand ein weiterer Bin hinzugefügt. Somit waren insgesamt 7 Neuronen in der Ausgabeschicht notwendig. Beim 360°-System wurde eine Ausgabeschicht aus neun Bins mit einer Breite von jeweils 40° eingesetzt. Da es sich hier um einen geschlossenen Kreis handelt, sind weitere Neuronen nicht notwendig.

Abbildung 3.23 zeigt die erzielten die Ergebnisse bei einer 3-6-7 bzw. 3-6-9-Netzwerk-

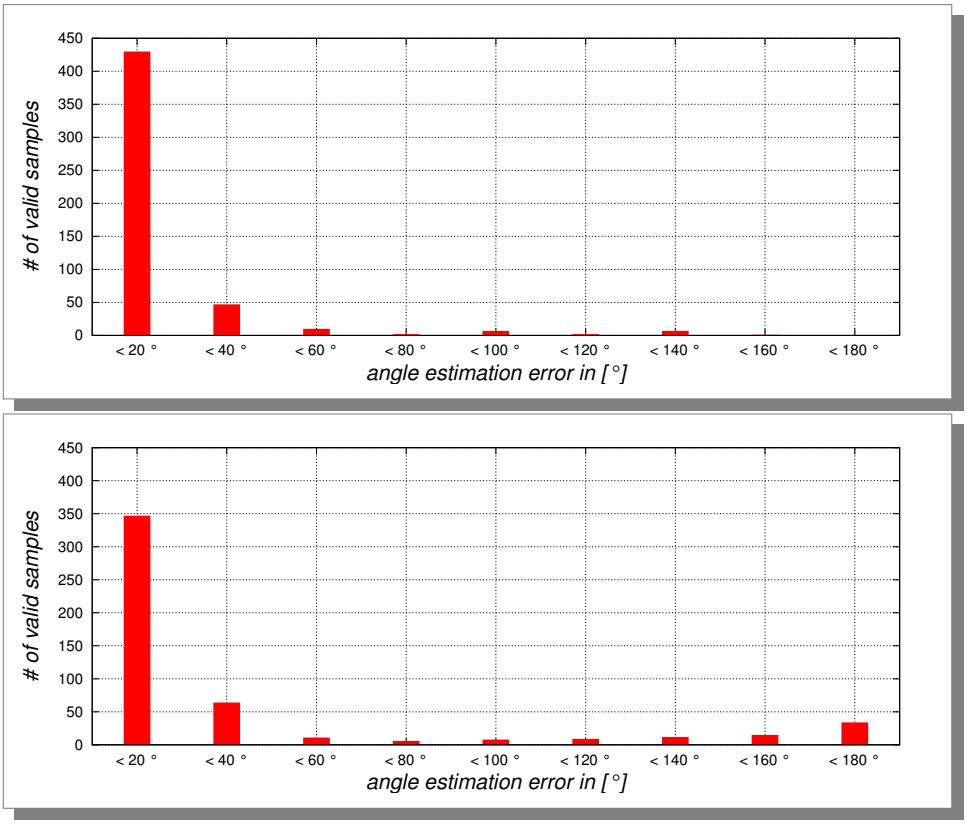



***Abbildung 3.24:*** *Fehlerhistogramme im* 180°*-System (oben) und* 360°*-System (unten) bei Verwendung eines Multi Layer Perceptrons. Auf Grund der Symmetrie ist beim* 360°*-System für Fehler* $> 90°$ *ein Anstieg der Fehlerhäufigkeit zu verzeichnen. (Das* 20°*-Intervall ist bedingt durch die zugrunde liegenden Testdaten.)*

Architektur für eine ausgewählte Testperson (weitere Ergebnisse finden sich im Anhang E.1). Im Idealfall würden alle Ergebnispunkte der Schätzung (rote Kreuze) wieder auf der gestrichelten Linie (*Ground Truth*) liegen. Bei beiden Systemen ist ersichtlich, dass mit dem MLP – wie erwartet auf Grund der Kodierungsmächtigkeit zweischichtiger MLPs – bessere Ergebnisse als mit einer linearen Regression oder einem einfachen $k$-Nearest-Neighbour-Klassifikator erzielt werden können. Fehlschätzungen und größere Abweichungen sind vorhanden, kommen aber nur in geringer Anzahl vor.

Die Verteilung der Abweichungen $\Delta\phi$ für beide Varianten ist in Abbildung 3.24 dargestellt. Es ist deutlich zu erkennen, dass wie beim $k$-Nearest-Neighbour-Klassifikator der größte Teil der Fehler kleiner gleich 20° ist. Auf Grund der Symmetrie ist beim 360°-System bei $\Delta\phi > 90°$ auch hier ein Anstieg der Fehlerhäufigkeit zu verzeichnen. Beim 180°-System nimmt die Anzahl der Fehler mit steigender Abweichung $\Delta\phi$ kontinuierlich ab.

Weiterhin wurde untersucht, ob die Architektur des MLP noch weiter vereinfacht und die Anzahl der Gewichte weiter reduziert werden kann. Ausgehend von dem eingesetzten 3-6-7 bzw. dem 3-6-9 Netzwerk wurde die Hiddenschicht jeweils weiter reduziert. Die entstehenden Netze wurden jeweils 5-mal trainiert und in einer Kann-Phase bezüglich des maximalen Schätzfehlers $\Delta\phi_{max}$ von 30° und 60° getestet. Die erzielten gemittelten Ergebnisse sind in Tabelle 3.4 zusammengefasst. Es ist ersichtlich, dass die Ergebnisse auch bei geringerer Anzahl von Neuronen in der Hiddenschicht nur wenig schlechter werden. Erst beim Entfernen der Hiddenschicht, nehmen die Fehler deutlich zu. Dies kann damit begründet werden, dass die Abbildungkomplexität eines einfachen Perceptrons im Wesentlichen der einer linearen Regression entspricht, und damit offenbar nicht für eine entsprechend genaue Schätzung der Oberkörperorientierung geeignet ist.

| 180°-System | | | 360°-System | | |
|---|---|---|---|---|---|
| Architektur | $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ | Architektur | $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ |
| $3-6-7$ | 92.1% | 96.8% | $3-6-9$ | 77.1% | 84.0% |
| $3-4-7$ | 91.5% | 96.0% | $3-4-9$ | 76.9% | 84.4% |
| $3-3-7$ | 89.7% | 95.7% | $3-3-9$ | 73.9% | 83.2% |
| $3-7$ | 78.1% | 88.1% | $3-9$ | 51.3% | 62.3% |

***Tabelle 3.4:*** *Ergebnisse verschiedener MLP-Architekturen im 180°- und 360°-System für maximale Schätzfehler $\Delta\phi_{max}$ von 30° und 60°.*

Im Rahmen dieser Dissertation wurde für das Training und Testen der MLPs die Implementierung der *Fast Artifical Neural Network Library (FANN)* [FANN, 2010] verwendet. Dabei wurden sowohl der *Standard Backpropagation* Algorithmus, das *Quickprop*-Verfahren und *Resilient Propagation (RProp)* untersucht. Die besten und stabilsten Trainings-Ergebnisse wurden mittels des RProp-Ansatz erzielt. Weitere ausführliche Details zu den verschiedenen Lernverfahren finden sich beispielsweise in [Zell, 1994].

### 3.6.7 Schätzung mittels *Support Vector Machine*

Als weiterer Klassifikator wurde eine *Support Vector Machine (SVM)* untersucht. Dabei wurde die SVM als binärer Klassifikator entsprechend dem *one-versus-one* Verfahren [Knerr et al., 1990,Kreßel, 1999] eingesetzt. Im Rahmen dieser Dissertation wurde die Implementierung aus der *LIBSVM - A Library for Support Vector Machines* [libSVM, 2010] verwendet. Weitere Details zu SVMs finden sich im Anhang A.3.

Als Input für die SVM kam wie bei den anderen Verfahren der reduzierte Parametervektor $\mathbf{p} = (p_0, p_1, p_3)$ zum Einsatz. Es wurden ein linearer Kernel, ein polynomieller Kernel und ein RBF-Kernel untersucht. Die besten Ergebnisse wurden mit einem RBF-Kernel erreicht.

Als Ergebnis der Schätzung der Oberkörperorientierung mit einer SVM wird direkt eine der Winkelklassen ausgegeben. Zwischenwerte sind nicht möglich.

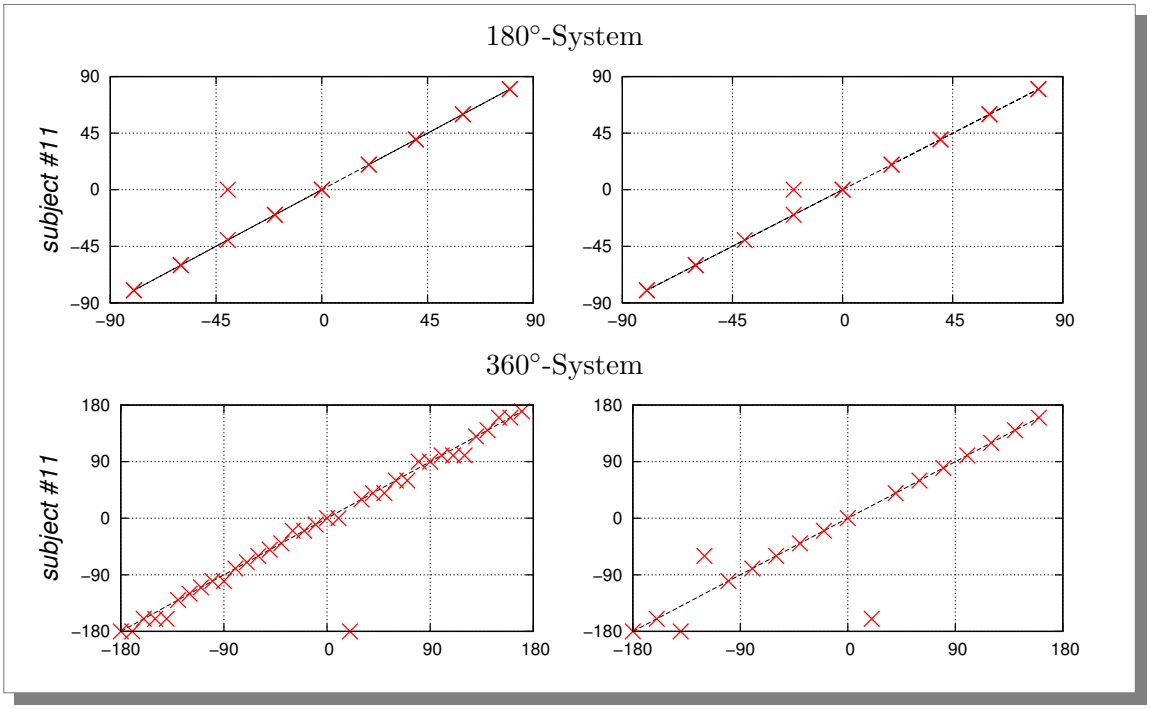



***Abbildung 3.25:*** *Ergebnisse der Oberkörperschätzung mittels SVM im 180°-System (oben) und im 360°-System (unten) für eine ausgewählte Testperson.*

Abbildung 3.25 zeigt die erzielten Ergebnisse der SVM-Multiklassen-Klassifikation für eine Testperson (weitere Ergebnisse finden sich im Anhang E.1). Im Idealfall würden wieder alle Ergebnispunkte der Schätzung (rote Kreuze) auf der gestrichelten Linie (*Ground Truth*) liegen. Im Vergleich zum MLP entstehen Ergebnisse die qualitativ sehr ähnlich sind. Fehlklassifikationen und größere Abweichungen kommen in geringer Anzahl vor, sind aber weiterhin vorhanden.

Die Verteilung der Abweichungen $\Delta\phi$ für beide Varianten ist in Abbildung 3.26 dargestellt. Auch ist zu sehen, dass die SVM sehr ähnlich zum MLP abschneidet. Der größte Teil der Fehler ist auch hier kleiner gleich 20°. Beim 180°-System ist die Anzahl der Fehler größer 60° sehr gering. Es bei $\Delta\phi_{max} > 140°$ steigt die Anzahl der Fehler auf Grund der Symmetrie wieder leicht an. Beim 180°-System nimmt die Anzahl der Fehler mit steigender Abweichung $\Delta\phi$ kontinuierlich ab. Fehler größer 60° kommen praktisch überhaupt nicht vor.

Die Ergebnisse der SVM-Klassifikation bezüglich des maximalen Schätzfehlers $\Delta\phi_{max}$ von

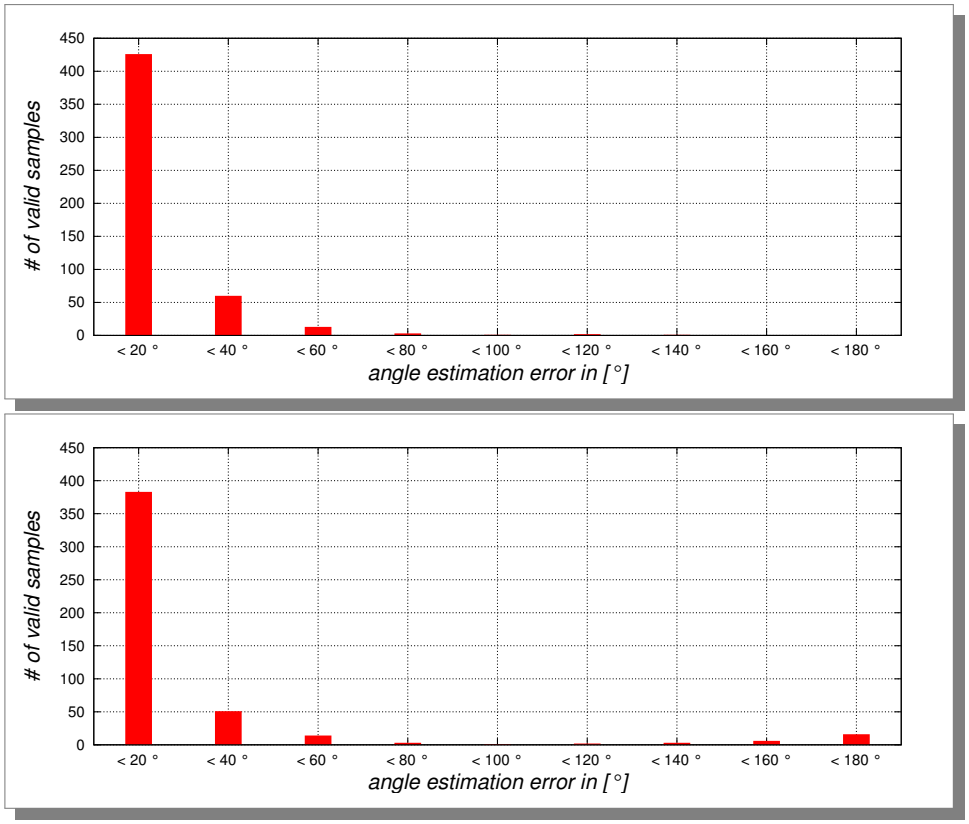



***Abbildung 3.26:*** *Fehlerhistogramme im 180°-System (oben) und 360°-System (unten) bei Verwendung einer Support Vector Machine. Auf Grund der Symmetrie ist beim 360°-System für Fehler $> 90°$ ein Anstieg der Fehlerhäufigkeit zu verzeichnen. (Das 20°-Intervall ist bedingt durch die zugrunde liegenden Testdaten.)*

30° und 60° sind in Tabelle 3.5 dargestellt. Auch hier ist zu erkennen, dass die erzielten Ergebnisse etwas besser sind als beim MLP. Dies lässt sich mit der Tatsache begründen, dass

| 180°-System | | | 360°-System | |
|---|---|---|---|---|
| $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ | | $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ |
| 96.0% | 99.2% | | 85.1% | 89.1% |

***Tabelle 3.5:*** *Ergebnisse der SVM-Klassifikation im 180°- und 360°-System für maximale Schätzfehler $\Delta\phi_{max}$ von 30° und 60°.*

eine SVM eine optimaler Klassifikator ist und bedingt durch die Transformation in einen höherdimensionalen Merkmalsraum eine bessere Trennung der Klassen ermöglicht.

### 3.6.8 Vergleich der Klassifikatoren

In den vier vorangegangenen Abschnitten wurden vier Verfahren zur Schätzung der Oberkörperorientierung auf Basis des reduzierten Parametervektors $\mathbf{p} = (p_0, p_1, p_3)$ untersucht. Alle vier Verfahren zeigen, dass eine Schätzung gut möglich ist, jedoch auch mit Ausreißern zu rechnen ist. Als Bewertung wurde eine maximale Abweichung des Schätzfehlers $\Delta\phi_{max}$ von 30° und 60° ausgewählt. Die Tabelle 3.6 zeigt zusammengefasst die erzielten Ergebnisse.

| Typ | 180°-System | | | 360°-System | |
|---|---|---|---|---|---|
| | $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ | | $\Delta\phi_{max} = 30°$ | $\Delta\phi_{max} = 60°$ |
| Regression | 70.1% | 96.3% | | 34.7% | 57.6% |
| k-NN | 87.7% | 95.5% | | 72.3% | 79.3% |
| MLP | 92.3% | 96.8% | | 78.5% | 83.8% |
| SVM | 96.0% | 99.2% | | 85.1% | 89.1% |

***Tabelle 3.6:*** *Vergleich der Ergebnisse von linearer Regression, k-NN, MLP und SVM in Bezug auf einen maximalen Schätzfehler $\Delta\phi_{max}$ von 30° und 60°.*

Die lineare Regression reicht für eine einfache Schätzung bereits aus, kann jedoch auf Grund der geringen Abbildungskomplexität nicht mit den anderen drei Verfahren mithalten. Es ist ersichtlich, dass k-NN, MLP und SVM sehr ähnlich abschneiden. Keines der Verfahren zeichnet sich hinsichtlich der Ergebnisse besonders gegenüber den Anderen aus. Generell ist das k-NN immer etwas schlecht als MLP und SVM, die beide sehr ähnliche Ergebnisse erzielen. Nur beim 360°-System ist die SVM-basierte Schätzung etwas besser.

Die erzielten Ergebnisse sind auch vergleichbar mit denen, die mittels des Template-basierten Verfahren von [Li et al., 2010] (siehe Abschnitt 3.2.2) auf der entsprechenden Testdatenbank erreicht wurden.

In Bezug auf die Komplexität und damit verbunden der Aufwand zum Erstellen des gewünschten Klassifikators und die benötigte Rechenzeit, hat das k-NN Vorteile gegenüber den anderen beiden Verfahren. Das Training der SVMs ist am aufwendigsten. Der Aufwand zur Berechnung eines MLP in der Kann-Phase ist auch noch relativ gering. Bei den SVMs ist der Rechenaufwand am größten.

Deshalb wurde im Folgenden das MLP-Verfahren für die weiteren Untersuchungen ausgewählt, da dieses das beste Aufwand-Nutzen-Verhältnis besitzt.

### 3.6.9 Schätzung von Trajektorien

In den vorangegangenen Abschnitten wurde die Leistungsfähigkeit verschiedener Klassifikatoren zur Schätzung der Ausrichtung des Oberkörpers auf Basis von Einzelbildern betrachtet. In der Praxis ist jedoch eine Schätzung der Oberkörperpose auf einem Videodatenstrom notwendig, um beispielsweise eine näher kommende Person in Bezug auf ein mögliches Interaktionsinteresse zu klassifizieren. Als Ergebnis steht hierzu dann der zeitliche Verlauf des geschätzten Winkels zur Verfügung. (In Kombination mit einem *PersonTracker* kann hiermit auch eine Bewegungstrajektorie der Person ermittelt werden.)

Im Rahmen der Dissertation wurden in verschiedenen Szenarien Experimente mit einem realen Robotersystem durchgeführt. Da in Realwelt-Anwendungen mit beliebigen Personen keine *Ground-Truth* Informationen vorhanden sind, konnte hier nur eine subjektive Bewertung der Ergebnisse vorgenommen werden. Zusätzlich wurden in einem Testszenario *Ground-Truth* Informationen aufgenommen und anschließend eine vergleichende Auswertung durchgeführt.

#### Tests in Realwelt-Szenarien ohne Ground-Truth Daten

In diesen Tests wurde ein Shopping-Robotersystem in einem Baumarkt eingesetzt. Das System wurde an verschiedenen, geeigneten Stellen im Markt plaziert und Daten wurden mit einer omnidirektionalen Kamera aufgenommen. Die Daten wurden nachträglich zur Auswertung in einzelne Sequenzen zerlegt.

Abbildung 3.27 zeigt eine erste Sequenz aus dem Baumarkt: In diesem Beispiel stand der Roboter in einem Hauptgang und eine Person hat sich frontal auf den Roboter zubewegt. Die Sequenz beginnt bei einer Entfernung von ca. 7m. Bei diesem Abstand ist die Person noch zu klein im Bild und kann nicht getrackt werden. Bei Frame 40 beträgt die Entfernung noch etwa 4m. Ab diesem Punkt ist ein kontinuierliches Tracken des Oberkörpers möglich. Bei Frame 70 ist die Person noch ca. 1.5m vom Roboter entfernt und wird seitlich rechts am Roboter vorbeigehen. Auf Grund der Kamerageometrie ist die Person in den weiteren Frames nicht mehr vollständig sichtbar und kann nicht mehr weiter getrackt werden.

Im Kurvenverlauf der Winkelschätzung ist zu sehen, dass die Kurve anfangs um die 0°-Achse pendelt. Dies ist durch die deutlichen Armbewegungen der Person beim Laufen bedingt. Zum Ende der Sequenz geht die Person seitlich rechts am Roboter vorbei. Die ist durch einen kontinuierlich positiven Winkel ab etwa Frame 70 ersichtlich.

Abbildung 3.28 zeigt eine weitere Sequenz: In diesem Fall wurde der Roboter im Eingangsbereich plaziert. Zunächst bewegt sich eine Person von vorn kommend nach rechts am Roboter vorbei. Die Winkelschätzung ist korrekt positiv bis etwa Frame 55. Ab diesem Zeitpunkt ist die Person nicht mehr vollständig im Bild, und es kommt zu einer Fehlschätzung, bevor

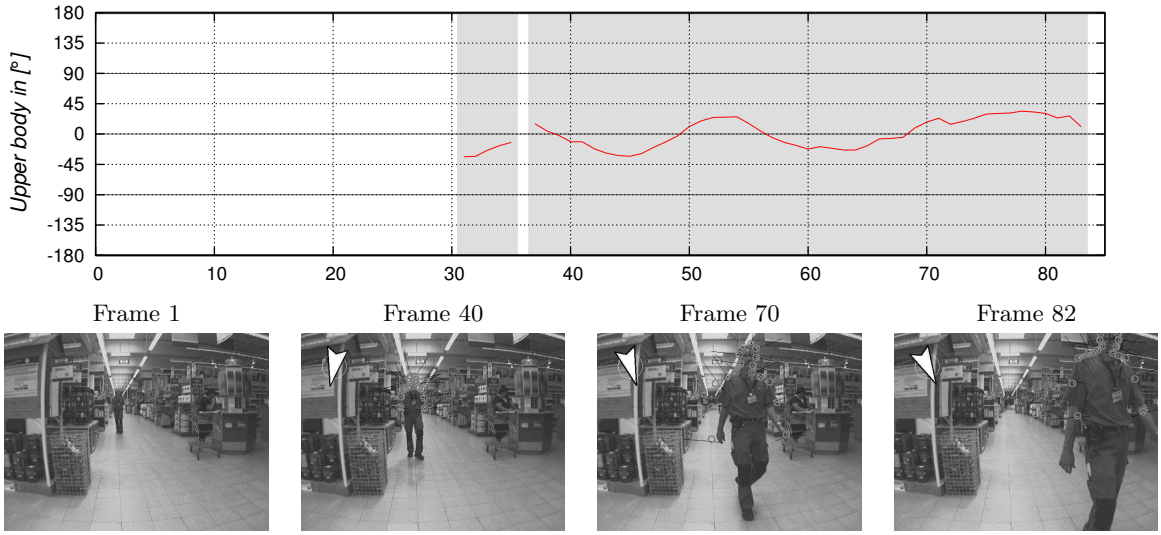



***Abbildung 3.27:*** *Beispielsequenz 130403 aus Baumarkt: Der obere Teil der Grafik zeigt den Verlauf des geschätzten Winkels des Oberkörpers. Im grau hinterlegten Bereich (Abstand zur Person: ca. 4m bis 0.7m) war eine gültige Schätzung möglich. Im unteren Teil sind vier Beispielbilder der Sequenz dargestellt.*

die Kontur endgültig ungültig wird. Als diese Person aus dem Sichtbereich verschwindet, kommen zwei weitere Personen von links und bewegen sich nach vorn rechts vom Roboter weg. Von Frame 65-80 wurde eine falsche Kontur zwischen den beiden Personen geschätzt. In diesem Fall ist auch die Winkelschätzung falsch. Danach springt die Kontur nach einer automatischen Re-Initialisierung auf die rechte Person und der Winkel zeigt nun korrekt, dass sich die Person vom Roboter wegbewegt.

Diese beiden Beispiele zeigen, dass eine monokulare Schätzung der Oberkörperorientierung in Realwelt-Szenarien möglich ist. Eine quantitative Auswertung ist auf Grund der fehlenden *Ground-Truth* Daten nicht möglich.

### Test-Szenario mit Ground-Truth Daten

Da in den untersuchten Realwelt-Szenarien keine Ground-Truth-Daten verfügbar sind, wurden weitere Tests in einem separaten Testszenario durchgeführt. Während der Datenaufnahmen wurden zusätzlich Informationen über die sich im Bild bewegende Person mit Hilfe eines separaten Trackingsystems aufgenommen. Dabei handelt es sich um den in [Schenk et al., 2011] vorgestellten Ansatz, bei dem mehrere Laserscanner die Szene beobachten und mit Hilfe der detektierten Beinpaare die Position der Person bestimmt werden kann. Dabei kommt ein *Particle Filter* zum Einsatz, der sowohl Position als auch Geschwindigkeit

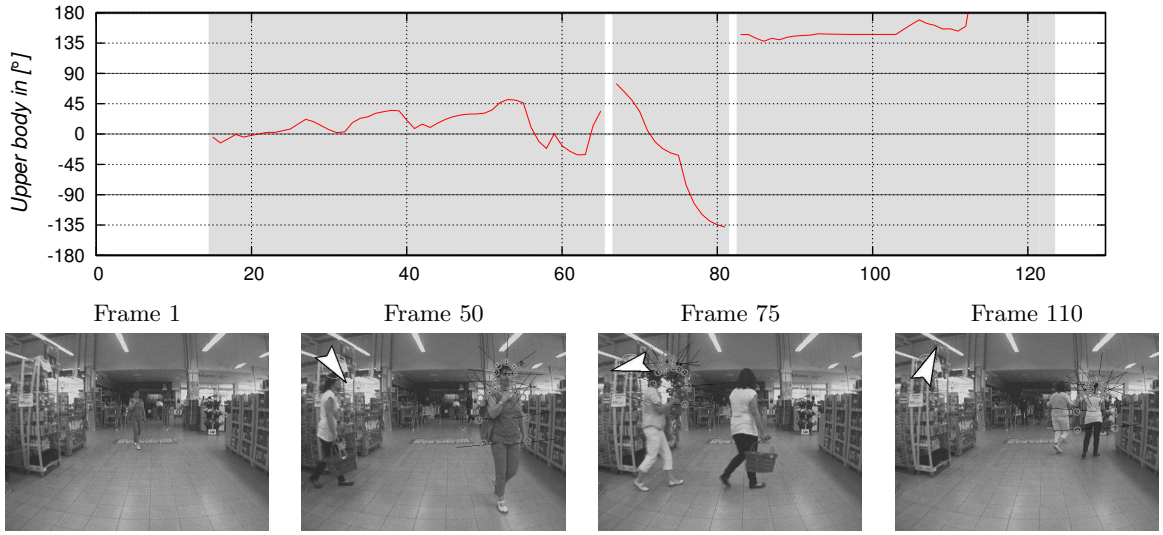



***Abbildung 3.28:*** *Beispielsequenz 133232 aus Baumarkt: Der obere Teil der Grafik zeigt den Verlauf des geschätzten Winkels des Oberkörpers. Im grau hinterlegten Bereich war eine gültige Schätzung möglich. Im unteren Teil sind vier Beispielbilder der Sequenz dargestellt.*

und Bewegungsrichtung der Person schätzt. Diese dient im Folgenden als Ground-Truth-Information. Problematisch ist hierbei, dass dieses System die Bewegungsrichtung nur schätzen kann, wenn sich die Person auch bewegt. Die Orientierung einer stehenden Person (oder einer sich auf der Stelle drehenden Person) ist damit nicht bestimmbar. Daher muss beim Vergleich der Daten die Geschwindigkeit mit berücksichtigt werden. Bei niedrigen Geschwindigkeiten sind die Ground-Truth-Daten nicht sicher.

Abbildung 3.29 zeigt einen ersten Testlauf, bei dem eine Person von links ins Bild ($t = 0s$) kommt, sich dem Roboter nähert ($t = 0..5s$), dann nach rechts abdreht ($t = 6s$), einen Kreis läuft ($t = 6..10s$) und dann nach links am Roboter vorbei ($t = 14s$) sich entfernt.

Zum Beginn der Testsequenz konnte die Person nicht getrackt (und somit auch keine Oberkörperpose bestimmt) werden, da sich die Person kaum vom Hintergrund abhebt (dunkles T-Shirt und dunkelroter Vorhang). Später konnte die Person gut getrackt und eine Orientierungsschätzung vorgenommen werden. Der Verlauf der Winkelschätzung stimmt dabei gut mit den Ground-Truth-Informationen überein. Zum Ende der Sequenz ist die Person nicht mehr im Bild, und daher kann keine Orientierung mehr geschätzt werden.

Das Ergebnis einer weiteren Testsequenz ist in Abbildung 3.30 zu sehen. Hierbei kam die Person von links ins Bild ($t = 1s$). ist geradeaus nach rechts bis zur Wand gelaufen ($t = 5s$)

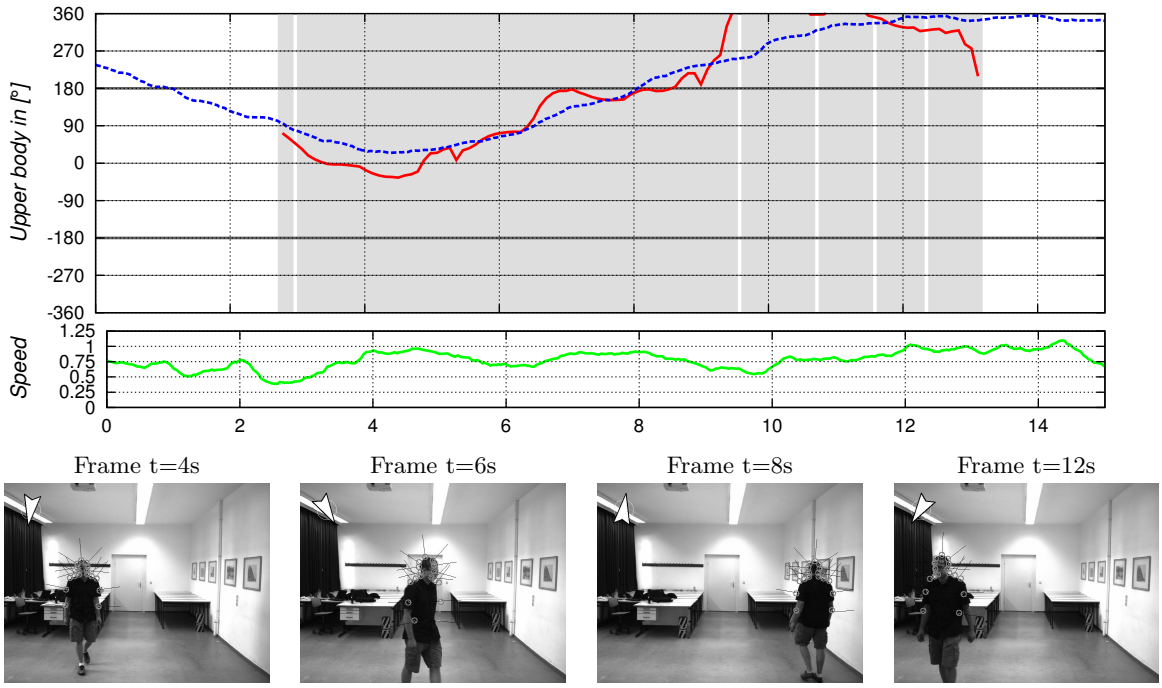



***Abbildung 3.29:*** *Beispielsequenz A4: Der obere Teil der Grafik zeigt in Rot den Verlauf des geschätzten Winkels des Oberkörpers. Im grau hinterlegten Bereich war eine gültige Schätzung möglich. Die Ground-Truth-Daten des Lasertrackers sind gestrichelt in Blau dargestellt. Das untere Diagramm zeigt in Grün die geschätzte Geschwindigkeit der Person. Im unteren Teil sind vier Beispielbilder der Sequenz dargestellt.*

und von dort direkt zum Roboter. Von $t = 8s$ bis $t = 12s$ stand die Person vor dem Roboter und ist dann nach rechts abgedreht ($t = 14s$) und hat sich in einem Kreisbogen laufend wieder nach rechts aus dem Bild entfernt.

Wie in der ersten Testsequenz konnte die Testperson nicht vor dem dunklen Vorhang getrackt werden. Der Verlauf der geschätzten Oberkörperorientierung und die Daten des Lasertrackers passen gut zusammen. Von $t = 8s$ bis $t = 20s$ vollzieht die Person eine vollständige 360° Drehung (im Kreis gelaufen), die sich als durchlaufender Winkel von $0 - 360°$ auch im Diagramm zeigt. Zwischen $t = 11s$ und ca. $t = 14s$ steht die Person einige Sekunden vor dem Roboter und schaut ein wenig umher. Dies äußert sich durch schwankende Werte der geschätzten Oberkörperorientierung. Die Ground-Truth-Daten bleiben hier dagegen relativ konstant, da bezogen auf die Beine keine Veränderung beobachtet werden können.

In beiden Beispielen zeigt sich ein gewisses Rauschen der Orientierung im Vergleich zum Lasertracker. Dieses Rauschen wird durch kleinere Sprünge bei der Anpassung der Kontur (ins-

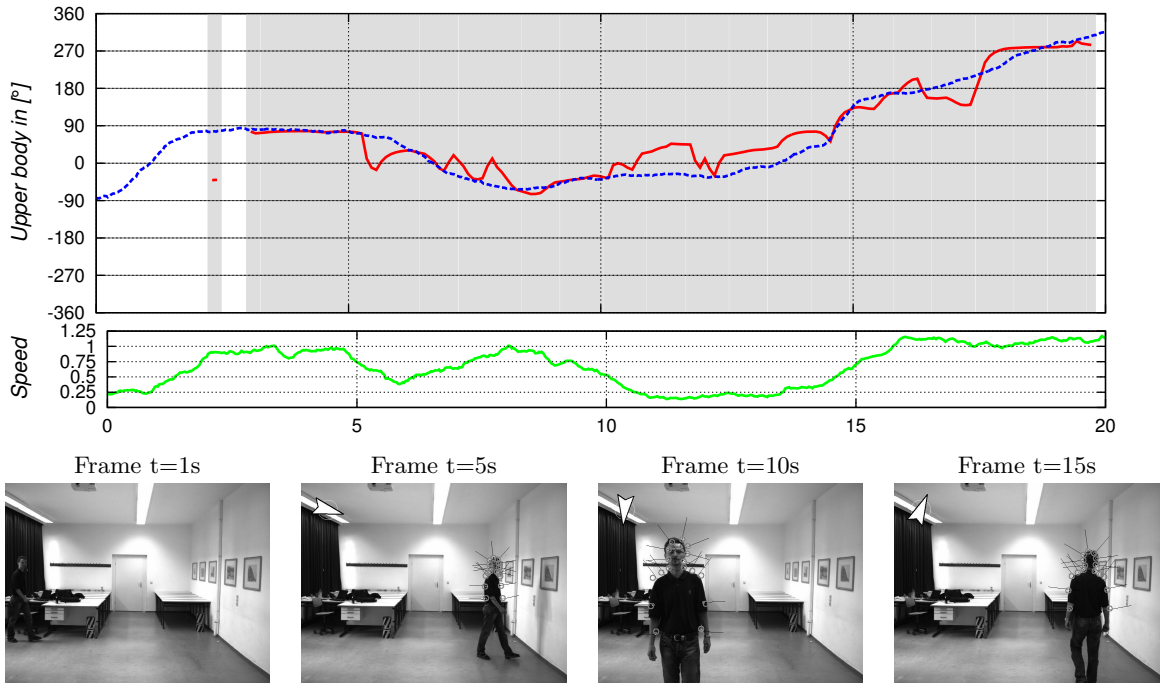



***Abbildung 3.30:*** *Beispielsequenz C4: Der obere Teil der Grafik zeigt in Rot den Verlauf des geschätzten Winkels des Oberkörpers. Im grau hinterlegten Bereich war eine gültige Schätzung möglich. Die Ground-Truth-Daten des Lasertrackers sind gestrichelt in Blau dargestellt. Das untere Diagramm zeigt in grün die geschätzte Geschwindigkeit der Person. Im unteren Teil sind vier Beispielbilder der Sequenz dargestellt.*

besondere bei schnellen Bewegungen) verursacht. Mittels einer zeitlichen Tiefpass-Filterung kann hier jedoch eine gute Glättung erreicht werden.

## 3.7 Zusammenfassung

Um eine natürliche und intuitive Interaktion zwischen einem mobilen Robotersystem und einem Menschen realisieren zu können, ist es notwendig, dass ein Robotersystem in die Lage versetzt wird, einen Kommunikationspartner im Hinblick auf Interaktionsinteresse und Aufmerksamkeit einschätzen zu können. Ein erster Anhaltspunkt hierfür ist die Orientierung des Oberkörpers.

In den vorangegangenen Abschnitten wurde hierzu ein erstes Teilsystem vorgestellt, dass in der Lage ist, zunächst eine Person im Bild zu detektieren, anschließend den Oberkörper mit Hilfe eines *"analysis by synthesis"* Systems genau zu erfassen und auf Basis der bestimmten Modellparameter die Orientierung in Form eines Winkels zu schätzen.

Für die Initialdetektion wurde ein *HOG*-Detektor gewählt, da dieser bei entsprechendem Training auch nicht frontal stehende Personen gut detektieren kann. Die detailierte Erfassung des Oberkörpers erfolgt auf Basis der äußeren Kontur der Silhouette mit Hilfe eines *Active-Shape-Models*. Anschließend erfolgte eine Reduktion der Modellparameter durch Bestimmung der *Mutual Information*. Als Ergebnis steht am Ende der Detektion ein niedrigdimensionaler Merkmalsvektor zur Verfügung, der zur Schätzung der Oberkörperpose weiter verwendet werden kann.

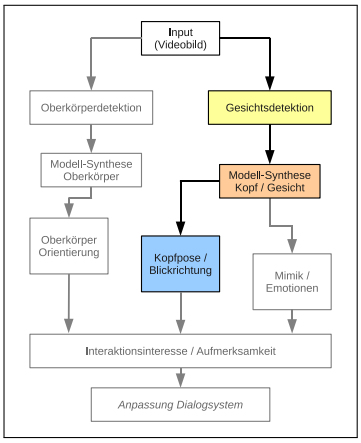
Auf Grundlage einer erstellten Datenbank wurden vier verschiedene Funktionsapproximatoren bzw. Klassifikatoren (*lineare Regression*, *k-Nearest-Neighbour*, *Multi Layer Perceptron* und *Support Vector Machine*) trainiert und miteinander verglichen. Die besten Ergebnisse wurden dabei mit einem SVM erreicht. Das beste Aufwand-Nutzen-Verhältnis wurde mit dem MLP erzielt.

Das Gesamtsystem wurde auf verschiedenen Videodatenströmen getestet. Es wurden Testläufe in Realwelt-Szenarien durchgeführt, die auf Grund nicht vorhandener Ground-Truth Daten jedoch nur qualitativ bewertet werden können. Zusätzlich wurden Tests auf Videodaten mit bekannten Ground-Truth Informationen durchgeführt. Hier zeigte sich, dass das vorgestellte System in der Lage ist, den Winkel korrekt zu schätzen. Probleme treten auf, sobald die Kontur auf Grund von Verdeckungen oder zu geringer Kontrastunterschiede zum Hintergrund nicht mehr korrekt erfasst werden kann.

Zusammenfassend kann festgestellt werden, dass das vorgestellte System zur Bestimmung der Orientierung des Oberkörpers geeignet ist und somit als Grundlage zur Schätzung der Aufmerksamkeit oder des Interaktionsinteresses eines Benutzers herangezogen werden kann.

# 4 Teilsystem 2: Blickrichtung

## 4.1 Einleitung




Im vorangegangenen Kapitel wurde beschrieben, wie die Orientierung des Oberkörpers bei der Interaktion zwischen Mensch und Roboter basierend auf einem Kamerabild geschätzt werden kann. Diese Information ist sowohl wichtig für die Anbahnung der Interaktion zwischen Benutzer und Roboter (Person kommt auf den Roboter zugelaufen) als auch während des Dialogs (Person steht unmittelbar vor dem Roboter oder wendet sich ab). Das Abwenden des Kommunikationspartners bedeutet im Normalfall auch das Ende oder den Abbruch des geführten Dialogs (z.B. weil dieser langweilig oder nicht weiter von Interesse war).

Vor einem Abwenden des Dialogpartners zeigt sich ein Desinteresse oder eine Ablenkung aber typischerweise bereits durch Änderung der Blickrichtung. Wenn der Dialog langweilig wird, beginnen Benutzer oftmals in der Umgebung umherzuschauen (siehe Abschnitte 1.2 und 1.3). Durch die Bestimmung der Blickrichtung kann bereits frühzeitig erkannt werden, ob der Benutzer seine Aufmerksamkeit noch auf den Roboter richtet und somit auch noch Interesse am Dialog hat. Bei der Blickrichtung ist dabei die Bewegung des Kopfes (*Head Gaze*) und die Bewegung der Augen (*Eye Gaze*) zu unterscheiden.

Im Rahmen dieser Dissertation soll mittels des *HeadPoseTrackers* lediglich die Kopfbewegung bzw. die Kopfpose als weiterer möglicher Input zur Schätzung des Interaktionsinteresses genutzt werden. Die Bewegung der Augen spielt nur eine untergeordnete Rolle. Die Ausrichtung des Kopfes kann durch drei Winkel um die drei Raumachsen vollständig beschrieben werden (siehe Abbildung 4.1). In der Literatur werden sowohl die Winkel *yaw - pitch - roll* als auch *pan - tilt - roll* verwendet. Beide Möglichkeiten beschreiben jedoch die gleichen Winkel. Im Rahmen dieser Dissertation wurde die letztere Variante gewählt.

Zur Schätzung des Interaktionsinteresses sind vor allem Winkel um die $x$- und $y$-Achse wichtig. Der *roll*-Winkel liefert keine Aussagen zur Blickrichtung, ist jedoch zur Beschreibung der Lage des Kopfes trotzdem wichtig. Im Rahmen dieser Dissertation spielt er jedoch nur

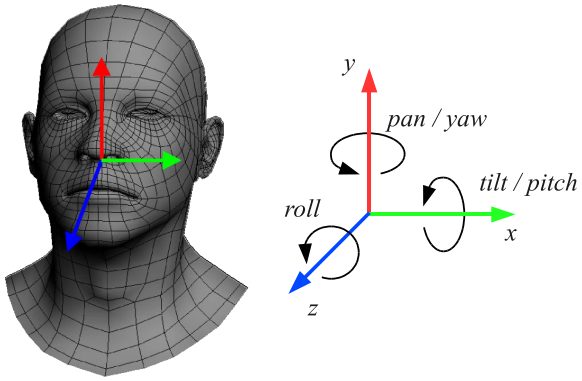



***Abbildung 4.1:*** *Der Kopf befindet sich im dreidimensionalen Raum. Dementsprechend kann die Ausrichtung bzw. Lage des Kopfes mit Hilfe von drei Winkeln, die die Rotation um die drei Raumachsen kennzeichnen, beschrieben werden.*

eine untergeordnete Rolle.

Im folgenden Abschnitt 4.2 werden einige bekannte Verfahren aus der Literatur zur Schätzung der Blickrichtung vorgestellt. Danach werden in den Abschnitten 4.3 und 4.4 die vorgeschlagene Architektur des *HeadPoseTrackers* und die einzelnen Bestandteile im Detail erläutert. Am Ende des Kapitels befindet sich in Abschnitt 4.5 die Beschreibung der Ergebnisse.

## 4.2 State-of-the-Art

Beim Begriff der Blickrichtung ist einerseits die Pose (also die räumliche Lage im 3D-Raum) des Kopfes (auch *Head Gaze* genannt) und andererseits die Ausrichtung der Augen (genannt *Eye Gaze*) zu unterscheiden.

Beim Menschen zeigen in der “Ruhelage” die Kopfpose und der Blick der Augen in die gleiche Richtung. Für kleine Winkel und schnelle Bewegungen werden typischerweise nur die Augen (z.B. beim Lesen) verwendet. Mit einem gewissen zeitlichen Versatz bewegt der Mensch dann auch den Kopf hinterher und bringt dabei die Augen wieder in die Nulllage. Die Kopfpose kann auch als zeitliche Tiefpassfilterung der Augenbewegung betrachtet werden.

Als Pose bezeichnet man allgemein die Position und die Orientierung eines Objektes. Bei der Bestimmung der Kopfpose in Videodatenströmen wird die Position des Kopfes im Bild und die Ermittlung der drei Raumwinkel (siehe Abbildung 4.1) betrachtet. Die Bestimmung der Kopfpose spielt in vielfältigen Szenarien eine Rolle. Es geht um die Klärung grundlegender

Fragen wie: *„In welche Richtung blickt die Person?“*, *„Auf was blickt die Person?“* und *„Schaut die Person in die Kamera?“*. Anwendungen finden die Informationen der Kopfpose z.B. als Bestandteil von Fahrerassistenzsystemen, mit Hilfe derer überprüft wird, ob sich der Fahrer auf den Verkehr konzentriert oder abgelenkt ist [Trefflich, 2010]. Daneben finden die Daten auch in der Robotik ihre Anwendung. So lassen sich mit Hilfe der Kopfpose Aussagen darüber treffen, ob sich der Nutzer gerade auf den zur Interaktion bereitstehenden Roboter konzentriert oder ob andere Umweltelemente seine Aufmerksamkeit auf sich ziehen.

Für die Schätzung der Aufmerksamkeit und des Interaktionsinteresses im Mensch-Roboter-Dialog ist die Bestimmung der Kopfpose ausreichend. Die exakte Blickrichtung der Augen spielt nur eine untergeordnete Rolle.

Eine Gemeinsamkeit der meisten Verfahren ist ein zweistufiger Prozess: Zuerst wird die Position des Kopfes oder Gesichts im Bild ermittelt und anschließend die Schätzung der Pose oder Blickrichtung vorgenommen. Die meisten Verfahren können in die Kategorien *Appearance-based*, *Feature-based* oder *Model-based* (siehe Abschnitt 2.3) eingeordnet werden. *Template-based* Verfahren sind dagegen wenig zu finden.

Im Folgenden werden bekannte Verfahren zur Gesichtsdetektion (Abschnitt 4.2.1) und zur Bestimmung der Kopfpose (Abschnitt 4.2.2) vorgestellt. Verfahrens aus dem Bereich *Eye Gaze* sind für diese Dissertation nicht relevant und werden daher nicht weiter betrachtet.

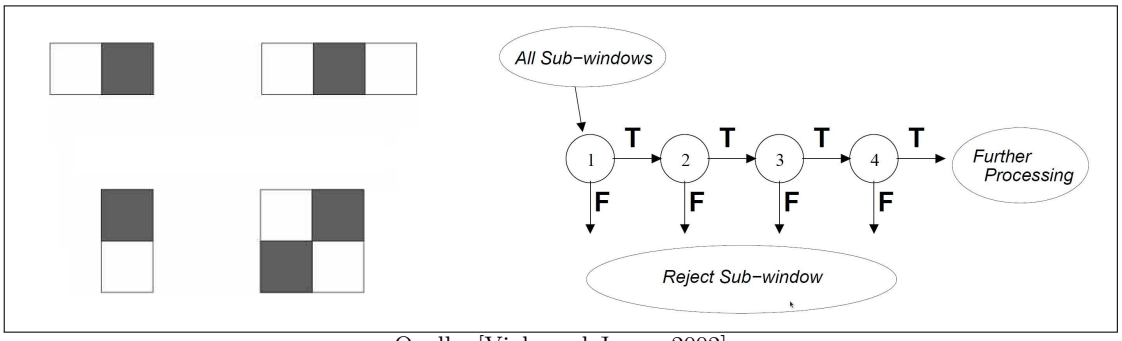
### 4.2.1 Gesichtsdetektion nach Viola und Jones

Bevor die Kopfpose oder Gesichtsmerkmale aus einem Eingabebild extrahiert werden können, ist eine exakte Lokalisierung des Gesichts im Bild notwendig. In den letzten Jahren hat sich der nach den Autoren benannte *Viola-Jones-FaceDetector* [Viola and Jones, 2001, Viola and Jones, 2002] als de facto Standard etabliert. Ein kurzer Überblick über alternative Verfahren findet sich im Anhang B.
Der Grundgedanke des sehr leistungsfähigen Ansatzes von Viola und Jones besteht in der stufenbasierten Kombination von schwachen Klassifikatoren zu einem einzigen, leistungsfähigen Klassifikator:

- Als *schwache Klassifikatoren* werden dabei einfache Rechteck-Filter (siehe Abbildung 4.2 links) eingesetzt, die sich sehr effizient auf einem Integralbild berechnen lassen. Durch Anwendung eines passenden Schwellwertes entsteht ein binärer Klassifikator.

- Mehrere solcher schwachen Klassifikatoren können zu einem binären *komplexen Klassifikator* zusammengesetzt werden. Dabei werden die schwachen Klassifikatoren in verschiedenen Größen und an verschiedenen Positionen in einem Detektorfenster von 24x24 Pixeln plaziert.

- In einem dritten Schritt werden mehrere komplexe Klassifikatoren in einer sog. *Klassifikatorkaskade* hintereinander geschaltet (siehe Abbildung 4.2 rechts). In den ersten Stufen dieser Kaskade werden dabei die sehr häufig auftretenden Klassen von Nicht-Gesichtern (z.B. homogene Flächen) heraus gefiltert. In den weiteren Stufen werden dann weitere immer seltener auftretende Klasse eliminiert, bis in der letzten Stufe nur noch die Klasse der Gesichter übrig bleibt.




Quelle: [Viola and Jones, 2002]

***Abbildung 4.2:*** *Links: Einfache Rechteck-Filter der schwachen Klassifikatoren. Der Filterwert ergibt sich aus der Summe der Grauwerte der weißen Fläche abzüglich der Summe der Grauwerte der schwarzen Fläche.*
*Rechts: Prinzip der Klassifikatorkaskade. In jeder Stufe werden mehr falsche Hypothesen verworfen. Die Anzahl der zu untersuchenden Hypothesen nimmt immer mehr ab.*

Die konkrete Auswahl der Zusammensetzung (Größe und Position) der schwachen und komplexen Klassifikatoren in der Klassifikatorkaskade aus der Vielzahl von möglichen Kombinationen erfolgt dabei mit einem *AdaBoost*-Lernverfahren. In der Kann-Phase wird die erstellte Klassifikatorkaskade dann über das Input-Bild geschoben und an jeder Position nach Gesichtern gesucht. Da die schwachen Klassifikatoren sehr schnell arbeiten, können falsche Hypothesen frühzeitig ausgeschlossen werden, wodurch eine hohe Geschwindigkeit erreicht wird. Die Suche in verschiedenen Größen kann mit Hilfe des Integralbildes ebenfalls sehr effizient durchgeführt werden. Als Resultat ermittelt der Algorithmus die Regionen des Bildes, in denen sich mit hoher Wahrscheinlichkeit Gesichter befinden.

### 4.2.2 Verfahren zur Schätzung der *Head Gaze*

Verfahren, die unter die Kategorie *Head Gaze* fallen, versuchen die Blickrichtung nicht aus den Augen zu extrahieren, sondern aus der Ausrichtung des Kopfes. Dies ist in vielen Fällen, in denen die Personen beispielsweise weiter von der Kamera entfernt sind, die einzige Möglichkeit, da die Augenpartie nicht mehr hoch genug aufgelöst werden kann.

Betrachtet eine Person einen Gegenstand, der sich in seinem mittleren Blickwinkelbereich befindet, also nicht ganz rechts oder ganz links von ihm steht, stimmt die Blickrichtung der Augen im Regelfall mit der Ausrichtung des Kopfes überein. Somit ist ein Verfahren, das lediglich die Ausrichtung des Kopfes berücksichtigt unter vielen Rahmenbedingungen geeignet, die Blickrichtung zu ermitteln.

In den folgenden Unterabschnitten werden einige ausgewählte Verfahren vorgestellt. Eine umfassende Übersicht über verschiedene Verfahren, erzielte Ergebnisse und verwendete Datenbanken zur *Head Pose Estimation* findet sich in [Murphy-Chutorian and Trivedi, 2009].

#### Appearance-based Verfahren zur Schätzung der Kopfpose

Die bildbasierten (*appearance-based*) Verfahren beruhen im Allgemeinen auf der Verarbeitung der Farb- oder Grauwertinformationen des Bildes. Bei der Verwendung von Stereokameras kommen noch Tiefeninformationen dazu. Die Verarbeitung erfolgt in der Regel mittels neuronaler Netze.

Ein Vorteil dieser Verfahren ist, dass sie keine Initialisierung benötigen. Des Weiteren sind sie zeitunabhängig und können auch auf Bilder mit geringer Auflösung angewandt werden.

In [Stiefelhagen et al., 2002] wurde ein Verfahren vorgestellt, das während Videokonferenzen den jeweils redenden Teilnehmer bestimmt. Dies geschieht indirekt über die Blickrichtung der Teilnehmer. Dabei geht man davon aus, dass die redende Person angeschaut wird. Als Input dient das Bild einer omnidirektionalen Kamera (siehe Abbildung 4.3 links). In diesem Kamerabild werden zunächst Gesichter gesucht. Aus den gefundenen Gesichtern wird anschließend mit zwei MLPs die jeweilige Blickrichtung bestimmt. Ein MLP dient zur Schätzung des Neigungswinkels (vertikale Achse) und eines für den Schwenkwinkel (horizontale Achse). Jedes Netz besteht aus einem Ausgabeneuron, 20 bis 80 Hiddenneuronen und 20x90 Eingabeneuronen. Als Eingabe wird ein auf die Größe 20x30 Pixel normalisiertes Bild des Gesichts gewählt. Zusätzlich wird ein vertikales und horizontales Kantenbild des Gesichts mit gleicher Auflösung (siehe Abbildung 4.3 rechts) verwendet.
In der Trainingsdatenbank befinden sich 15 Personen. Von jeder Person gibt es zwei Serien à 93 Bilder. Eine Serie wird für das Training verwendet, die andere als Testdatensatz. Ausgegeben wird direkt der vom Netzwerk geschätzte Blickwinkel. Dieses System erreicht nach dem Training für unbekannte Personen eine maximale Auflösung von 9.5° für den Schwenk- und 9.7° für den Neigungswinkel. Auf diesen Arbeiten basierend wurden weitere Verfahren z.B. in [Voit et al., 2006] und [Voit and Stiefelhagen, 2008] präsentiert, die ähnliche Ergebnisse in anderen Szenarien erreichen.

In [Zhao et al., 2002] wurde ein sehr ähnlicher Ansatz verfolgt: Nach einer Vorverarbeitung


Quelle: [Stiefelhagen et al., 2002]

***Abbildung 4.3:*** *Links: Aufnahme der Omnikamera aus einer Konferenz. Die Gesichter wurden bereits detektiert und markiert. Anschließend wird die Blickrichtung der Personen ermittelt, um den gerade sprechenden Konferenzteilnehmer zu ermitteln. Dieser wird von den übrigen Personen angeschaut. Rechts: Eingabedaten für das neuronale Netz (extrahiertes Gesicht plus zugehörige horizontale und vertikale Kantenbilder).*

werden die Grauwertbilder auf eine Größe von 48x48 Pixel normiert. Diese "Retina" wird als Input für ein zweischichtiges MLP verwendet. Auf einer Testdatenbank mit 15 Personen wurde ein mittlerer Fehler von etwa 10° für Pan und Tilt erreicht.

#### Feature-based Verfahren zur Schätzung der Kopfpose

Bei den merkmalsbasierten (*feature-based*) Ansätzen werden Merkmale der Gesichter z.B. über Gaborwavelets oder Graph Matching extrahiert. Diese Verfahren zeigen im Allgemeinen eine größere Toleranz gegenüber Änderungen der Person oder der Beleuchtung als die *appearance-based* Verfahren. Vertreter der *feature-based* Verfahren sind u.a.:

- Gaborwavelets
  [Wu and Toyama, 2000], [Kalliomäki and Lampinen, 2003]
- Elastic Graph Matching
  [Krüger et al., 1997]
- Gabor Wavelet Networks
  [Bruske et al., 1998], [Krüger and Sommer, 2002]

In [Bruske et al., 1998] wurde ein Verfahren vorgestellt, dass ein gefundenes Gesicht nach Ausschneiden und Normierung mittels Gaborfilter weiter analysiert. Insgesamt kommen 64 Gaborfilter in einem 4x4 Gitter zum Einsatz. An jedem Punkt des Gitters befinden sich vier Filter mit den Orientierungen 0, $\pi/4$, $\pi/2$, $3\pi/4$. Die 64 Filterantworten dienen als Input für ein neuronales Netz. Allerdings werden die Daten vorher noch auf einen Subraum reduziert, der mittels OTPM (*Optimally Topology Preserving Maps*) und der PCA erzeugt wird [Bruske and Sommer, 1997] [Bruske and Sommer, 1998]. Diese Vorverarbeitung dient im Wesentlichen dazu, den Eingaberaum zu reduzieren und einem Problem der im folgenden verwendeten LLMs (*Local Linear Maps*) [Ritter et al., 1991] vorzubeugen. Diese sind anfällig für Eingabestörungen. Mit Hilfe der Subraumkonstruktion aus den eigenwertgrößten

Eigenvektoren (mittels PCA) können solche Störungen, die orthogonal zum konstruierten Subraum stehen, jedoch weitgehend vermieden werden.
Mit dieser Variante wurde auf einem Datensatz von 500 Bildern des Kopfes einer Puppe ein maximaler Fehler von 1.88° erreicht (minimal 0.64°). Allerdings weist der Autor darauf hin, dass für einen produktiven Einsatz die Vorverarbeitung optimiert werden muss und das System auch nicht echtzeitfähig ist.
In einer Weiterentwicklung dieses Verfahrens von [Krüger and Sommer, 2002] wurden das Verfahren auf realen Daten eingesetzt. Das System erzielt dabei eine Genauigkeit von 2° bei der Schätzung des Kopfwinkels, allerdings auf Grund der aufwändigen Gaborfilterung nur mit einer Framerate von 1 Hz. Erlaubt sind dabei Verdrehungen des Kopfes um maximal 30°.

Aktuellere Verfahren zur Anwendung von des *Elastic Graph Matching* oder der *Gabor Wavelet Networks* zur Kopfposenschätzung sind in der Literatur nicht zu finden, obwohl die von [Bruske et al., 1998] und [Krüger and Sommer, 2002] erzielten Ergebnisse recht vielversprechend waren. Ein Grund hierfür könnte der hohe Berechnungsaufwand sein, der für die Berechnung der Antworten der Gaborfilter notwendig ist.

#### Model-based Verfahren zur Schätzung der Kopfpose

In dieser Gruppe werden sowohl Gesichtsmerkmale als auch geometrische Eigenschaften verwendet. Diese Informationen werden anschließend in ein gemeinsames Modell integriert:

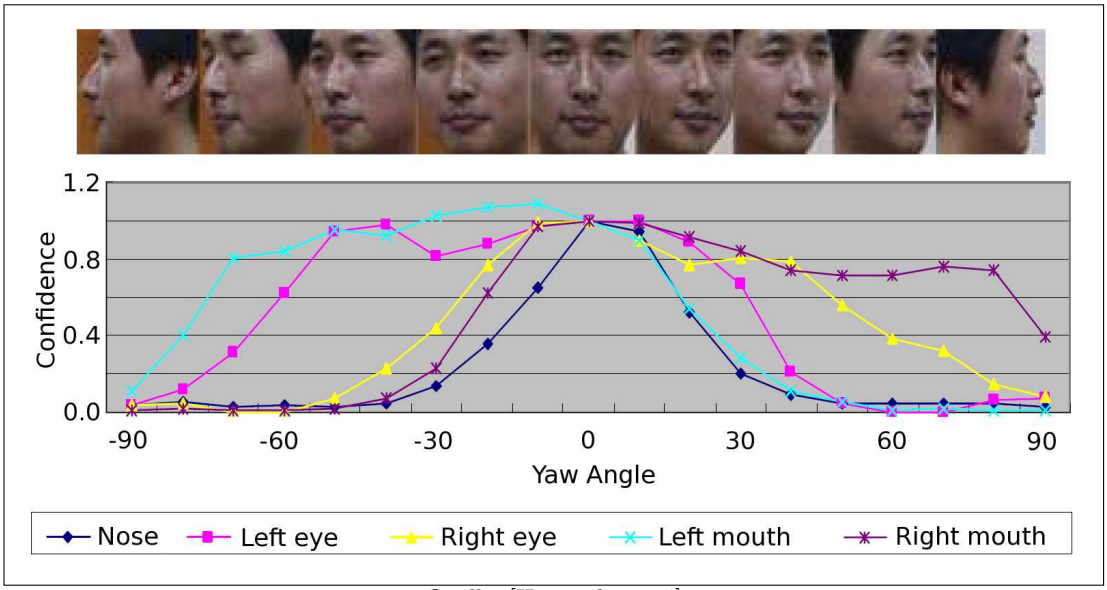
- Feature models - basierend auf detektierten Gesichtern in Kombination mit Merkmalsdetektoren
  [Hu et al., 2004], [Oka et al., 2005]

- Texture models - Texturieren eines gegebenen geometrischen Modells
  [Zivkovic and van der Heijden, 2001], [Dornaika and Ahlberg, 2004], [Xiao et al., 2002]

- Flexible Models, Active Shape Modells, Active Appearance Modells
  [Lanitis et al., 1997], [Cootes et al., 2000], [Baker et al., 2004]

Diese Verfahren können auch im dreidimensionalen Raum angewandt werden. Ihre Vorteile liegen in der guten Personenunabhängigkeit und ihrer Güte. Positiv ist auch, dass für das Training typischerweise keine vollständig gelabelten Daten benötigt werden.
In [Hu et al., 2004] wird ein Verfahren zur Schätzung der Kopfpose auf Basis eines sog. *coarse-to-fine frameworks* vorgestellt. Dabei wird in einem ersten Schritt zunächst eine Grobschätzung der Kopfpose vorgenommen. Diese erfolgt auf Basis getrackter Gesichtsmerkmale der Mund-Nase-Region. Zum Tracken wird hierbei ein modifizierter *sum-of-squared differences* (SSD) Tracker aus [Hager and Belhumeur, 1998] eingesetzt. Abbildung 4.4 zeigt den Verlauf verschiedener Merkmale in Abhängigkeit des *yaw/pan*-Winkels. Auf Basis der gefundenen

Merkmale erfolgt eine erste Schätzung der Kopfpose. In einer zweiten Stufe erfolgt die Feinabstimmung mit Hilfe der Abbildung eines 3D-Modells auf die Features im Inputbild. Hierbei kommen eine Reihe von geometrischen Berechnungen im 2D- und 3D-Raum zum Einsatz. Bei den von [Hu et al., 2004] durchgeführten Experimente lag der maximale Fehler immer kleiner 10°. Beim *pan*-Winkel lag der durchschnittliche Fehler bei 6.5° und beim *tilt*-Winkel bei 5.3°. Das System erreicht auf einem P4 1.4GHz bei 320x240 Pixel eine Geschwindigkeit von 15fps.

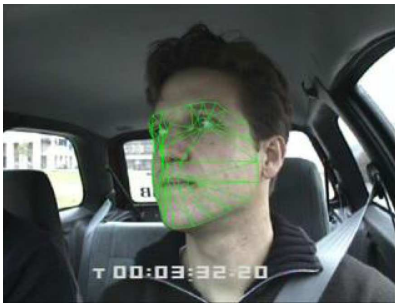



Quelle: [Hu et al., 2004]

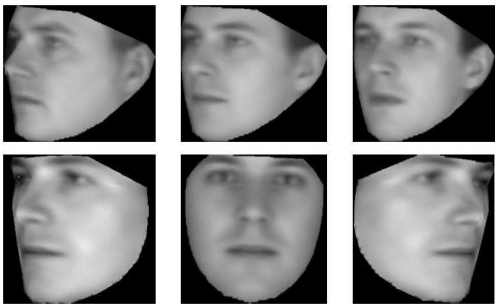
***Abbildung 4.4:*** *Abbildung verschiedener Gesichtsmerkmale aus der Mund-Nase-Region auf den yaw-Winkel der Kopfpose.*

Eine andere große wichtige Gruppe von Verfahren zur Schätzung der Kopfpose sind die *Flexible Models*, *Active Shape Modells (ASM)* und *Active Appearance Modells (AAM)*. Bei diesen Methoden handelt es sich um *"Analysis by Synthesis"* Verfahren (siehe Abschnitt 2.4), bei denen ein parametrisches Modell auf Basis von entsprechend gelabelten Daten aufgebaut wird. In der Kann-Phase wird das Modell auf unbekannte Daten angepasst. Durch die Anpassung der Modellparameter wird versucht, eine möglichst gute Übereinstimmung zwischen der Hypothese und der aktuellen Beobachtung zu erzeugen. Mit Hilfe der so ermittelten Modellparameter kann anschließend mit entsprechenden Klassifikatoren oder Funktionsapproximatoren eine Schätzung der Kopfpose vorgenommen werden.
Bereits in [Lanitis et al., 1997] wurde gezeigt, dass mit Hilfe der eigenwertgrößten Parameter des Modells im Bereich von ±20° eine gute Schätzung der Kopfpose durchgeführt werden kann. Genaue Angaben zum Fehler sind nicht zu finden.
Durch die Verwendung von verschiedenen AAMs für verschiedene Winkelklassen, konnte in [Cootes et al., 2000] der Arbeitsbereich für eine AAM-basierte horizontale Kopfposenschät-

zung auf $\pm 100°$ deutlich vergrößert werden. Der mittlere Fehler der Schätzung basierend auf einer linearen Regression lag bei 10°.
In [Baker et al., 2004] wurden AAMs als Fahrerassistenzsystem zur Schätzung der Aufmerksamkeit des Fahrers (basierend auf der Kopfpose) verwendet. Angaben zur erzielten Genauigkeiten finden sich leider nicht.




Quelle: [Baker et al., 2004]



Quelle: [Cootes et al., 2000]

***Abbildung 4.5:*** *Beispielbilder zur Kopfposenschätzung mittels AAMs. Links: Ein Bild eines Fahrer mit angepassten Modell aus [Baker et al., 2004]. Rechts: Abdeckungsbereich zweier Modelle aus [Cootes et al., 2000].*

### 4.2.3 Zusammenfassung

Im letzten Abschnitt wurden eine Reihe von Verfahren zur Schätzung der Blickrichtung aus der Literatur vorgestellt. Nahezu alle Verfahren verwenden einen Algorithmus zur Gesichtsdetektion zwecks Initialisierung der Schätzung der Blickrichtung.

Im Themenfeld der Blickrichtung muss in die beiden Varianten *Head Gaze* (Kopfpose) und *Eye Gaze* (Blickrichtung der Augen) unterschieden werden. In Bezug auf die exakte Blickrichtung einer Person sind *Eye Gaze* Verfahren im Allgemeinen wesentlich genauer als *Head Gaze* Varianten. Grund hierfür ist die Tatsache, dass diese Verfahren in der Lage sind, wirklich die Blickrichtung der Person zu erfassen und nicht nur die Ausrichtung des Kopfes. Dafür ist aber eine sehr hochauflösende Kamera notwendig, die die notwendigen hochaufgelösten Gesichtsausschnitte auch bei größerer Entfernung ($>\approx 0.5m$) liefern kann. Im Rahmen der angestrebten Anwendungsszenarien kann diese jedoch nicht erreicht werden. Die *Head Gaze* Verfahren sind daher die einzig mögliche Alternative zur Schätzung der Blickrichtung bei geringer Auflösung der Kamera und/oder größerem Abstand zur Kamera.

Da die Kopfpose auch als zeitliche Tiefpassfilterung der Augenbewegung betrachtet werden kann und somit Rückschluss auf den visuellen Aufmerksamkeitsfokus der betrachteten Person erlaubt, soll im Rahmen dieser Dissertation die Schätzung der Blickrichtung (bzw.

des Aufmerksamkeitsfokus) auf Basis der Kopfpose erfolgen.

Typische Vertreter aus dem Bereich der *Head Gaze*-Verfahren erreichen im Mittel eine Genauigkeit von kleiner als 5° in Bezug auf den *pan*- und *tilt*-Winkel der Kopfpose.

## 4.3 Systembeschreibung

Zur Bestimmung der Kopfpose ist es zunächst notwendig, den Kopf im Bild zu detektieren. Nach einer Grobdetektion muss die Kopfpose möglichst genau erfasst werden. Nach einer erfolgreichen Erfassung des Kopfes kann dann die eigentliche Bestimmung der Blickrichtung/Kopfpose vorgenommen werden. Dieser gesamte Erkennungsprozess soll im Rahmen dieser Dissertation somit in drei Teilschritte untergliedert werden:

1. Grobdetektion des Kopfes im Kamerabild:
   Im ersten Schnitt soll zunächst nur eine grobe Detektion des Kopfes im Bild erfolgen. Hierzu wird im Rahmen dieser Dissertation ein Standard-Gesichtsdetektor eingesetzt. Dieser sollte idealerweise eine möglichst gute Erkennungsrate aufweisen und gleichzeitig möglichst wenig Fehldetektionen erzeugen. Da auf Videodaten online gearbeitet werden soll, muss das Verfahren in der Lage sein, in Echtzeit zu arbeiten. Der in [Viola and Jones, 2001,Viola and Jones, 2002] vorgestellte Algorithmus erfüllt diese Anforderungen und wird daher im Rahmen dieser Dissertation eingesetzt.
   Als Ergebnis dieser ersten Stufe stehen eine Reihe von Bildregionen (z.B. Rechtecke) zur Verfügung, die die Gesichter der Personen im Sichtbereich der Kamera enthalten.

2. Feinanpassung Gesichtsmodell:
   Nachdem in der ersten Verarbeitungsstufe das Gesicht im Bild detektiert wurde, soll in der zweiten Stufe eine Feinanpassung eines parametrischen Gesichtsmodells an den Input erfolgen. Dieser Schritt der Modellanpassung muss einerseits in Echtzeit erfolgen können, um auf Videodaten online arbeiten zu können und andererseits aber auch genau genug sein, um die wirkliche Kopfpose zu erfassen.
   Im Rahmen dieser Dissertation wird für diesen Schritt ein *Active Appearance Model* (basierend auf den Vorarbeiten von [Wilhelm, 2005], [Werner, 2007][1] und [Stricker, 2008][1] eingesetzt. Ausführliche Details hierzu finden sich im Abschnitt 4.4. Als Ergebnis dieser zweiten Stufe steht ein an die Person angepasstes Gesichts-/Kopfmodell in parametrischer Form zur Verfügung.

3. Schätzung der Blickrichtung:
   In der letzten Stufe dieses Teilsystems erfolgt die eigentliche Schätzung der Kopfpose bzw. des Aufmerksamkeitsfokus auf Basis der im vorangegangenen Schritt ermittelten

[1] Diese Diplomarbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

Parameter des parametrischen Modells. Als Ergebnis liegt letztendlich eine Winkelschätzung vor, die die Kopfpose in Bezug auf das Kamerabild beschreibt.

Abbildung 4.6 veranschaulicht den Systemaufbau:

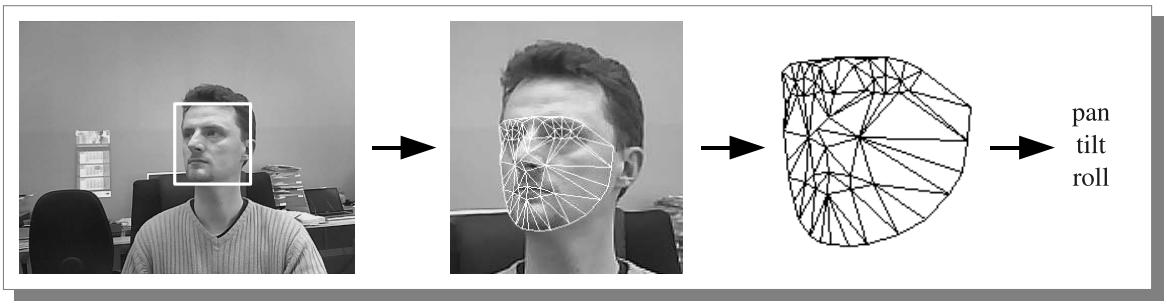



***Abbildung 4.6:*** *Systemaufbau zur Kopfposenschätzung: Im Inputbild wird ein Gesicht mit Hilfe eines Gesichtsdetektors gesucht. In der relevanten ROI wird das parametrische Gesichtsmodell angepasst. Als Ergebnis entsteht ein Modell, aus dessen Parametern die Kopfpose bestimmt werden kann.*

## 4.4 Active Appearance Models

*Active Appearance Models* (AAMs) wurden von [Cootes et al., 1998] vorgestellt. Sie wurden mit dem Ziel entwickelt, bestimmte Klassen von elastischen Objekten in einem beliebigen/unbekannten Bild zu detektieren und in eine parametrisierbare Form zu überführen. Erste Anwendungen von Active Appearance Modellen finden sich im Bereich der medizinischen Bildverarbeitung beispielweise zur automatischen Objektsegmentierung in Röntgen- oder Tomographenbildern. Mittlerweile werden Active Appearance Modelle jedoch in einer Vielzahl von anderen Anwendungsbereichen der Bildverarbeitung eingesetzt. Insbesondere wurde die Entwicklung der AAM-Verfahren durch den Einsatz zur Gesichtsdetektion und Gesichtsanalyse weiter vorangetrieben.

Bei der Anpassung eines AAMs an ein gegebenes Inputbild ist es das Ziel, einen Parametersatz zu finden, der sowohl die äußere und innere Form (*shape*) als auch das Aussehen (*appearance*) des Objektes beschreibt. In der weiteren Verarbeitung des Inputbildes in einem gesamten Bildverarbeitungsprozess können diese Parameter dann benutzt werden, um Aussagen über das detektierte Objekt zu treffen. Dies können beispielsweise Aussagen zur Größe des Objektes, über dessen Ausrichtung (z.B. Blickrichtung) oder ganzheitliche Eigenschaften (z.B. Geschlecht oder Mimik) sein.

Die Bestimmung des Parametersatzes eines Active Appearance Models stellt einen klassischen *Analysis-by-synthesis* Ansatz dar (siehe auch Abschnitt 2.4). Hierbei wird nicht direkt

versucht, dass zu detektierende Objekt aus dem Bild zu extrahieren. Stattdessen wird in einem iterativen Prozess ein Modell, ausgehend von einer Startschätzung, solange verfeinert, bis es möglichst ähnlich dem gesuchtem Objekt ist. Hierzu wird ein geschätztes Modell des zu suchenden Objektes erstellt und dieses in das Bild projiziert. Aus der Überlagerung des Bildes und des synthetisierten Modells ergibt sich ein Differenzbild, das den Fehler zwischen Modell und Bild enthält. Basierend auf diesem Fehler wird durch einen Anpassungsalgorithmus versucht, die Parameter so anzupassen, dass sich im nächsten Iterationsschritt ein kleinerer Fehler ergibt (also die Parameter das gesuchte Objekt besser beschreiben). Dieser Iterationsprozess wird solange wiederholt, bis eine definierte Fehlerschwelle unterschritten, eine bestimmte Anzahl von Schritten überschritten oder ein anderes Abbruchkriterium erreicht wird. Abbildung 4.7 veranschaulicht diesen Iterationsprozess.

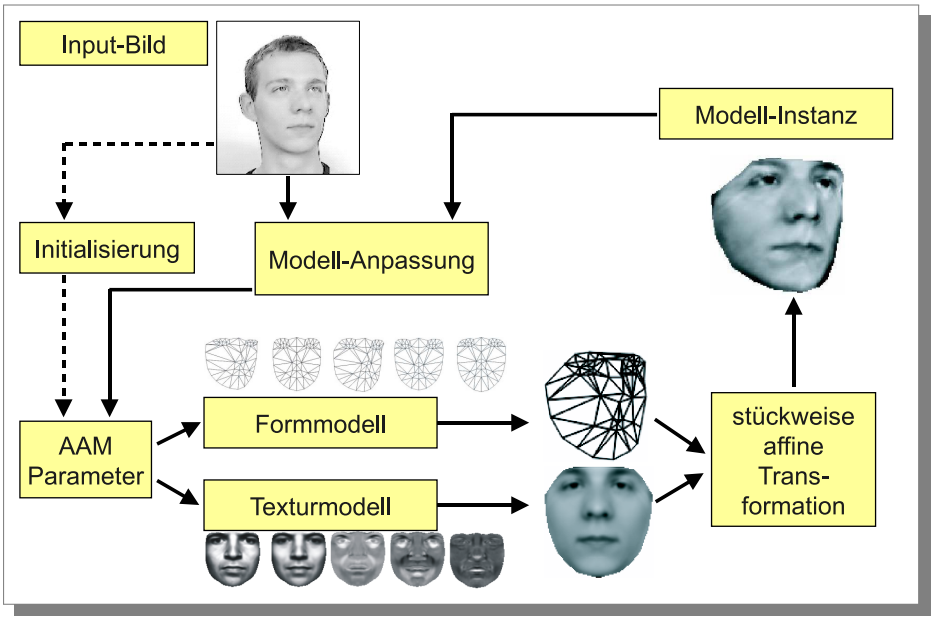



***Abbildung 4.7:*** *Iterationsprozess eines Active Appearance Model: Aus dem Inputbild wird mit Hilfe einer Initialisierung ein Parametersatz gewonnen. Auf Basis des Form- und Texturmodells wird eine Modellinstanz gebildet. Der Anpassungsalgorithmus vergleicht den Input mit der Modellinstanz und versucht, einen besseren Parametersatz mit kleinerem Fehler zu finden.*

Ein Active Appearance Model besteht somit aus zwei wesentlichen Komponenten: einem Modell, welches das Erscheinungsbild (Form und Aussehen) der Objektklasse beschreibt, und einem Anpassungsalgorithmus, der die Anpassung an ein unbekanntes Bild ermöglicht. Diese beiden Komponenten werden in den folgenden Abschnitten näher erläutert.

(Im Folgenden wird davon ausgegangen, dass Active Appearance Modelle zur Gesichtsde-

tektion und Gesichtanalyse eingesetzt werden.)

Da die Active Appearance Models ein wichtiger Bestandteil der vorliegenden Dissertation sind, wird der Aufbau und die Verwendung dieser Modelle in den folgenden Abschnitten näher erläutert. Die Inhalte sind dabei sinngemäß entnommen aus [Cootes et al., 1998], [Cootes et al., 2001], [Matthews and Baker, 2004], [Baker et al., 2003a], [Baker et al., 2003b] und [Baker and Matthews, 2004]. Weitere relevante Quellen sind an den entsprechenden Stellen angegeben.

### 4.4.1 Komponenten eines Appearance Modell

Das Appearance Modell dient zur Beschreibung von Gesichtern und muss daher deren Vielfältigkeit abbilden können. Hierzu sind zwei Arten von Variabilitäten zu unterscheiden: Einerseits in Bezug auf die Form (Gesichter unterschiedlicher Personen, unterschiedliche Gesichtsausdrücke oder Ausrichtung des Gesichts im Bild) und andererseits in Bezug auf das Aussehen (Hautfarbe, Bartwuchs, Brille, etc.). Die Formkomponente eines Active Appearance Modells ist hierbei eine Weiterentwicklung der in Abschnitt 3.5.2 vorgestellten Active Shape Models.

Die Erzeugung der beiden Modellkomponenten erfolgt auf Basis einer Reihe von Gesichtsbildern. Diese Gesichtsbilder müssen an ausgewählten markanten Stellen mit Labelpunkten $(x_i, y_i)$ versehen sein. Für jedes Bild der Datenbasis ergibt sich somit ein Labelvektor $\mathbf{l}$:

$$\mathbf{l} = (x_1, y_1, ..., x_n, y_n)^T. \tag{4.1}$$

Diese Labelvektoren und die dazugehörigen Bilder dienen als Grundlage für die Erstellung des Form- und Texturmodells. Abbildung 4.8 zeigt ein Bild mit den dazugehörigen Labelpunkten.

#### Formmodell

Das Formmodell dient der Beschreibung der geometrischen Eigenschaften des Objektes. Die Varianzen der Form werden auf Basis der Labelvektoren $\mathbf{l_1}$ bis $\mathbf{l_n}$ der Bilddatenbank bestimmt. In einem Vorverarbeitungsschritt werden diese von globalen Transformationen wie Translation, Skalierungen und Rotation befreit, da das Formmodell lediglich die lokale Form beschreiben soll. Hierzu wird typischerweise der *Procustes-Algorithmus* (siehe Anhang D.2) Algorithmus verwendet. Anschließend werden die so erstellten Vektoren $\mathbf{l'_i}$ einer Hauptkomponentenanalyse (PCA) unterzogen, um lineare Abhängigkeiten zwischen den Labelvektoren

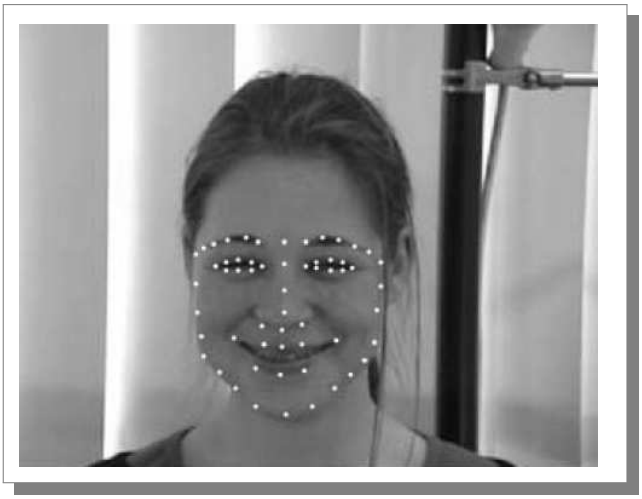


***Abbildung 4.8:*** *Beispielbild eines Gesichts nach dem Labeln zur Erstellung eines Active Appearance Models. Sichtbar sind 58 Punkte $(x_i, y_i)$, die an den relevanten/markanten Stellen im Gesicht plaziert wurden.*

zu beseitigen. Das vollständige Formmodell kann wie folgt beschrieben werden:

$$\mathbf{s} = \mathbf{s_0} + \sum_{i=1}^{n} p_i \mathbf{s_i} \qquad \text{mit} \qquad \mathbf{s_0} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{l'_i} \tag{4.2}$$

wobei $\mathbf{s_0}$ den Mittelwert aller vorverarbeiteten Gesichter $\mathbf{l'_i}$ und $\mathbf{s_i}$ den i-ten Eigenvektor der Hauptkomponentenanalyse darstellt, der jeweils mit einem Parameter $p_i$ skaliert werden kann. Durch die Variation der einzelnen Parameter $p_i$ können alle Formen $\mathbf{l_1}$ bis $\mathbf{l_n}$ sowie Linearkombinationen zwischen diesen Formen approximiert werden. Der Parameter $p_i$ sollte dabei maximal im Intervall $\left[-2\sqrt{\lambda_i} ... + 2\sqrt{\lambda_i}\right]$ variiert werden, wobei $\lambda_i$ der zur Komponente $\mathbf{s_i}$ korrespondierende Eigenwert ist. Damit kann sichergestellt werden, dass die resultierende Modelländerung über die Statistik der Labeldaten abgedeckt wird und nicht entartet oder deformiert ist [Cootes et al., 1995].

Abbildung 4.9 zeigt die Grundform $\mathbf{s_0}$ und vier ausgewählte Formkomponenten.

Weiterhin kann durch das Entfernen von Eigenvektoren $\mathbf{s_i}$ mit geringem Eigenwert eine erhebliche Dimensionsreduktion erreicht werden. Das reduzierte Formmodell kann wie folgt beschrieben werden:

$$\mathbf{s} \approx \mathbf{s_0} + \sum_{i=1}^{n'} p_i \mathbf{s_i} \qquad \text{mit} \qquad n' < n \tag{4.3}$$

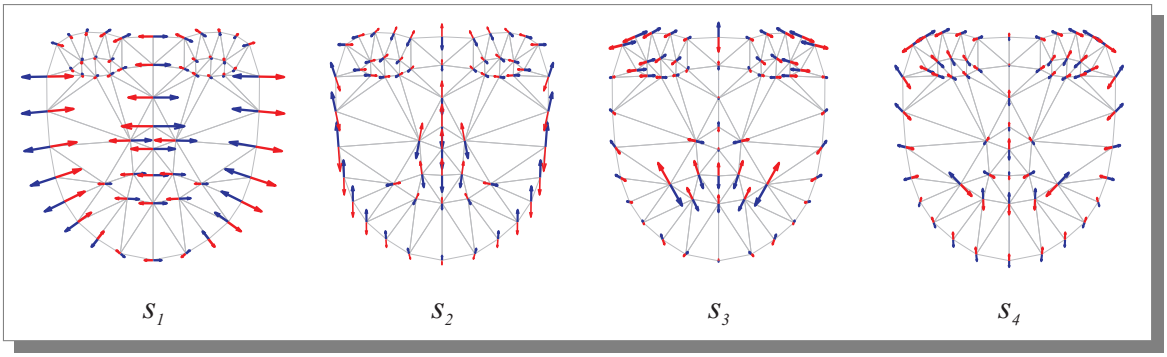



***Abbildung 4.9:*** *Darstellung der Formeigenvektoren $s_1$ bis $s_4$ in Abhängigkeit der Grundform $s_0$. Positive Änderungen der Skalierungsfaktoren $p_i$ sind durch rote Vektoren codiert. Negative Skalierungsfaktoren sind blau dargestellt. Siehe auch* (4.2)*. Quelle: [Werner, 2007]*

Dabei wird $n'$ meist so gewählt, dass mit dem reduzierten Modell typischerweise noch 90% oder 95% der Trainingsdaten abgedeckt werden können. Die Dimensionsreduktion ist erforderlich, um im nachgeschalteten Anpassungs- oder in einem Klassifizierungsprozess die Komplexität des Problems deutlich zu verringern. Anderenfalls würden sich hochdimensionale Parameterräume ergeben, die typischerweise stark zerklüftete Fehlergebirge aufweisen und eine Suche nach einem globalen Minimum schwierig oder unmöglich machen.

Ein resultierender Parametervektor einer konkreten Modellinstanz sei im Folgenden als $\mathbf{p} = (p_0, p_1, ..., p_{n'})$ definiert.

Als erste Komponente des Active Appearance Models liegt somit das Formmodell, bestehend aus einer Grundform $\mathbf{s_0}$ und einer Reihe von Formkomponenten $\mathbf{s_i}$ mit korrespondierenden Eigenwerten $\lambda_i$, vor. Eine konkrete Ausprägung des Modells ist über den Parametervektor $\mathbf{p}$ beschrieben. Dieses Modell beschreibt die lokalen Veränderungen der Gesichter aus der zugrunde liegenden Bilddatenbank. Neben dieser lokalen Anpassung ist jedoch auch noch eine globale Transformation notwendig, um ein unbekanntes Gesicht in einem beliebigen Inputbild an beliebiger Position beschreiben zu können.

Die globale Transformation innerhalb eines Inputbildes kann durch eine Translation, eine Skalierung und eine Rotation beschrieben werden. Ähnlich wie bei den Active Shape Models können diese Transformationen als zusätzliche synthetische Formkomponenten in das Modell integriert werden.

### Globale Transformation des Formmodells

Um globale Transformationen, die durch den *Procustes-Algorithmus* entfernt wurden, explizit zu modellieren, wird eine Transformation $N(\mathbf{x};\mathbf{q})$ definiert, welche auf einer Form $\mathbf{x}$ die durch den Vektor $\mathbf{q}$ parametrisierten globalen Transformationen durchführt. Die globale Transformation selbst kann als ein spezieller Satz an Formvektoren betrachtet werden und lässt sich, wie das Formmodell auch, über eine lineare Gleichung beschreiben:

$$N(\mathbf{x};\mathbf{q}) = \mathbf{x} + \sum_{i=1}^{4} q_i s_i^* \tag{4.4}$$

Ist die Grundform $\mathbf{s_0}$ durch $\mathbf{s_0} = (x_1^0, y_1^0, ..., x_n^0, y_n^0)^T$ gegeben, dann können die vier synthetischen Formkomponenten $\mathbf{s}_i^*$ wie bei den Active Shape Models (siehe Abschnitt 3.5.2) beschrieben werden durch:

$$\begin{aligned} \mathbf{s}_{tx}^* &= (1,0,...,1,0)^T & \mathbf{s}_{ty}^* &= (0,1,...,0,1)^T \\ \mathbf{s}_{scale}^* &= \mathbf{s}_0 = (x_1^0, y_1^0, ..., x_n^0, y_n^0)^T & \mathbf{s}_{rot}^* &= (-y_1^0, x_1^0, ..., -y_n^0, x_n^0)^T \end{aligned} \tag{4.5}$$

Der Vektor $\mathbf{s}_{scale}^*$ beschreibt die Skalierung der Grundform, $\mathbf{s}_{rot}^*$ ermöglicht näherungsweise die Rotation und die Vektoren $\mathbf{s}_{tx}^*$ und $\mathbf{s}_{ty}^*$ realisieren die Translation in x- und y-Richtung. Wie in Gleichung (3.9) bis (3.12) beschrieben, ist auch hier noch eine Normierung und Orthogonalisierung dieser Vektoren notwendig.

Als Ergebnis der Integration der globalen Transformation kann das Formmodell mit Hilfe der beiden Parametervektoren $\mathbf{p}$ und $\mathbf{q}$ vollständig beschrieben werden.

### Triangulation der Grundform

Nach der bisherigen Definition besteht das Formmodell lediglich aus einer Menge von Knotenpunkten $\mathbf{s}$, welche sich über die beiden Parametervektoren $\mathbf{p}$ und $\mathbf{q}$ modifizieren lassen. Dies ist jedoch noch keine hinreichende Beschreibung für ein Gesicht (also eine Oberfläche). Es ist daher notwendig eine Triangulation durchzuführen, um eine gesamtheitliche Beschreibung der Oberfläche zu erhalten. Dazu kommt typischerweise eine *Delaunay-Triangulation* (z.B. der *Flip*-Algorithmus [Shewchuk, 1996]) zum Einsatz. Dabei werden die Knotenpunkte durch Dreiecke verbunden, die anschließend eine geschlossene Oberfläche bilden (Abb. 4.10).

### Warp des Formmodells

Sowohl für die Erzeugung des Texturmodells als auch für die Anpassung des Active Appearance Modells besteht die Notwendigkeit, einen beliebigen Punkt innerhalb der Form $\mathbf{s_s}$ auf eine Zielform $\mathbf{s_d}$ überführen zu können. Eine solche Überführung wird über eine stückweise

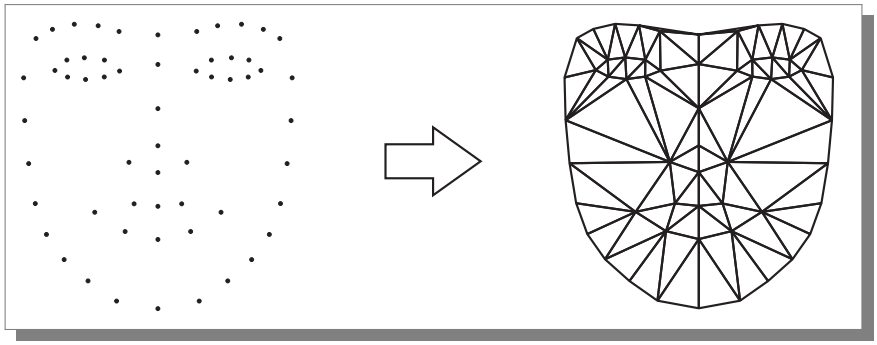


***Abbildung 4.10:*** *Durch eine Triangulation werden die einzelnen Knotenpunkte in eine geschlossene Oberfläche überführt. Links: Knotenpunkte* $\mathbf{s}$ *des Formmodells. Rechts: Delaunay-Triangulation der Fläche. Quelle: [Stricker, 2008]*

affine Transformation definiert. Das Prinzip dieser Transformation wird in Abbildung 4.11 verdeutlicht. Für jedes der, durch die Triangulation entstandenen, Dreiecke wird eine affine Abbildung definiert. Diese Abbildung ist durch die Knotenpunkte der jeweiligen Dreiecke in der Quellform $\mathbf{s_s}$ und in der Zielform $\mathbf{s_d}$ festgelegt und wird auch als *Warp* bezeichnet.

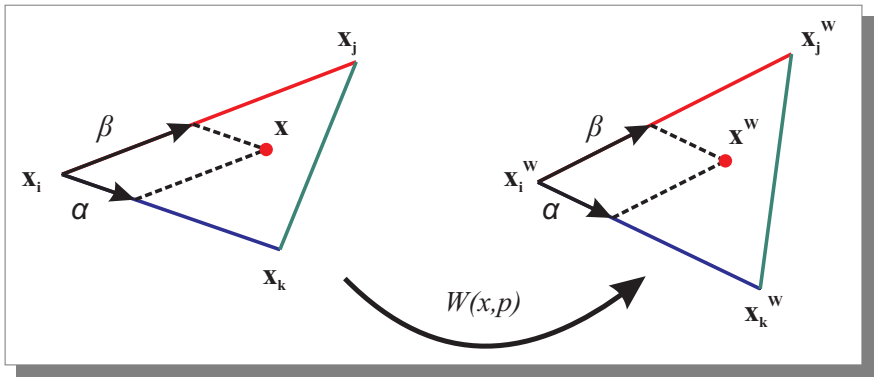



***Abbildung 4.11:*** *Überführung des Punktes* $x$ *von der Quellform in die Zielform über eine affine Transformation mit* $W(\mathbf{x}, \mathbf{p})$*. Quelle: [Stricker, 2008]*

Um jeden Punkt der Quellform $\mathbf{x}$ überführen zu können, wird für jedes Dreieck - und somit stückweise - die affine Transformation mit $W(\mathbf{x}, \mathbf{p})$ durchgeführt. Wobei die Zielform dieses Warps über den Parametervektor $\mathbf{p}$ des Formmodells bestimmt wird. Weitere Details zur Berechnung des Warps befinden sich im Anhang C.1.

### Texturmodell

Neben dem Formmodell besteht ein Active Appearance Model zusätzlich noch aus einem Texturmodell. Dieses dient zur Modellierung der Textur des Gesichtes, wie sie etwa durch

verschiedene Hauttypen oder den Bartwuchs einer Person entsteht. Da die Form des Gesichts bereits durch das Formmodell vollständig beschrieben ist, sollte das Texturmodell so gestaltet werden, dass dies möglichst unabhängig von der Form ist. Somit kann eine redundante Beschreibung der Form vermieden werden. Diese Unabhängigkeit kann erreicht werden, indem das Texturmodell auf die Grundform $\mathbf{s_0}$ transformiert wird (siehe Abbildung 4.12). Dazu kann der im letzten Abschnitt beschriebene Warp genutzt werden:

$$A_{S_0}(\mathbf{x}) = I(W(\mathbf{x}, p)) \ , \quad \forall \mathbf{x} \in A \tag{4.6}$$

Wobei $A_{S_0}$ das Bildkoordinatensystem der Grundform $\mathbf{s_0}$ beschreibt, welches im Folgenden auch als Templatekoordinatensystem bezeichnet wird.

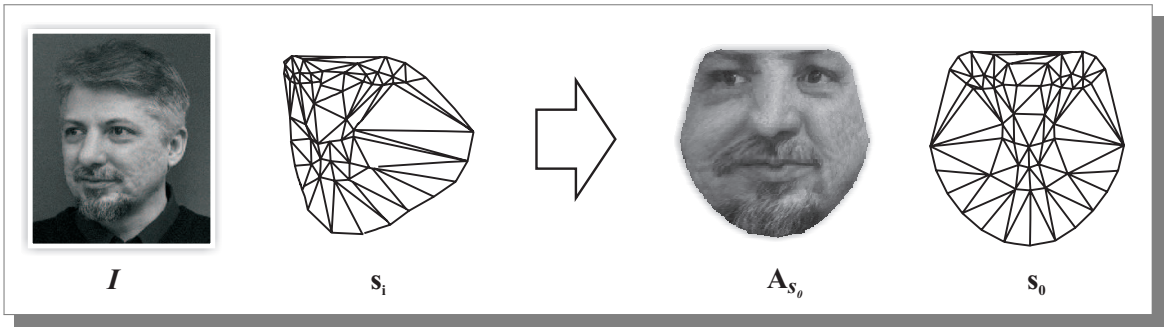



***Abbildung 4.12:*** *Überführung (Warp) eines Inputbildes I mit der Form* $\mathbf{s_i}$ *in das Templatekoordinatensystem* $A_{S_0}$ *der Grundform* $\mathbf{s_0}$*. Quelle: [Stricker, 2008]*

Vor der Erstellung des Texturmodells werden alle Bilder der zugrunde liegenden Datenbank auf die Grundform $\mathbf{s_0}$ transformiert. Die Pixel des Bildkoordinatensystems der Grundform können jeweils in einen Texturvektor $\mathbf{A}$ umgeformt werden, in dem sämtliche Grauwerte der Grundform zeilenweise in einen Spaltenvektor umgeordnet werden. Anschließend erfolgt die Erstellung des Texturmodells. Dies erfolgt vollkommen identisch der Erstellung des Formmodells über eine Hauptkomponenanalyse. Das resultierende Modell kann wie folgt beschrieben werden:

$$A = \mathbf{A_0} + \sum_{i=1}^{m} \lambda_i \mathbf{A_i} \tag{4.7}$$

Dabei beschreibt $\mathbf{A_i}$ den i-ten Eigenvektor der Hauptkomponentenanalyse und $\lambda_i$ den dazugehörigen Texturparameter. Abbildung 4.13 zeigt ein Mittelwertgesicht und die ersten Komponenten eines typischen Texturmodells. Dieses Modell ist sehr ähnlich zu der als *Eigenfaces* bekannten Repräsentation. Der wesentliche Unterschied zwischen den beiden Repräsentationsformen liegt in der Formnormierung des Texturmodells.

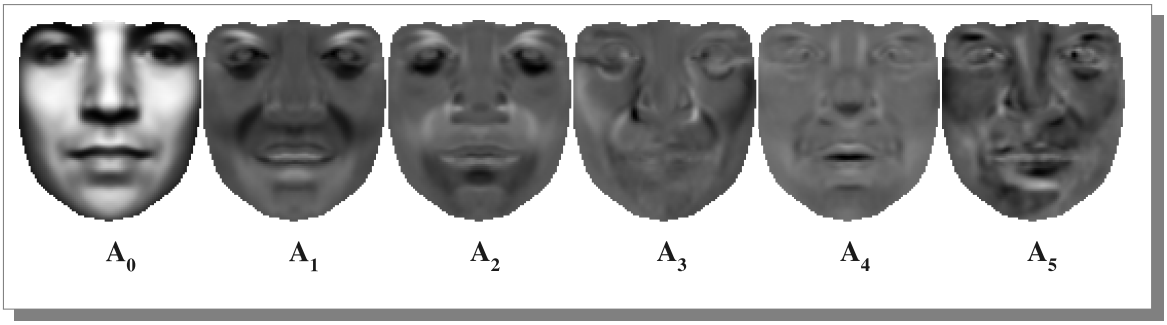



***Abbildung 4.13:*** *Darstellung des Mittelwertgesichts* $\mathbf{A_0}$ *und der ersten fünf Komponenten* $\mathbf{A_i}$ *eines typischen Texturmodells. Quelle: [Werner, 2007]*

### Kombination von Form und Textur

In den vorangegangenen Abschnitten wurden das Form- und Texturmodell und die dazugehörigen Komponenten vorgestellt. Die Mächtigkeit eines Active Appearance Modells wird jedoch erst durch die Kombination beider Teilmodelle erreicht. Durch die Kombination kann eine große Varianz an Formen und Texturen abgebildet werden.
Eine konkrete Ausprägung eines Active Appearance Modells wird auch als *Modellinstanz M* bezeichnet. Diese wird über die drei Parametervektoren $\mathbf{p}$, $\mathbf{q}$ und $\lambda$ beschrieben. Die Generierung einer Modellinstanz erfolgt in drei Schritten: Im ersten Schritt wird das Texturmodell mit Hilfe des Parametervektors $\lambda$ auf der Grundform $\mathbf{s_0}$ erzeugt. Anschließend wird die aktuelle Form der Modellinstanz mit Hilfe der lokalen und globalen Formparameter $\mathbf{p}$ und $\mathbf{q}$ beschrieben. Dies erfolgt durch einen Warp der durch $\mathbf{p}$ definierten lokalen Form auf die durch $\mathbf{q}$ bestimmte globale Form. Die Zielform des Warps wird in diesem Fall nicht durch $\mathbf{s_0}$ bestimmt, sondern durch die Form, welche die globale Transformation beinhaltet. Im letzten Schritt wird das auf der Grundform definierte Texturmodell mit Hilfe eines weiteren Warps auf die Form $\mathbf{s}$ projiziert:

$$M(x, \lambda, \mathbf{p}, \mathbf{q}) = A_0(W_q(x, p)) + \sum_{i=1}^{m} \lambda_i A_i(W_q(x, p)) \ , \quad \forall x \in M \tag{4.8}$$

Das resultierende Modell verfügt über separate Möglichkeiten der Repräsentation von Form und Textur über die Parametervektoren $\mathbf{p}$, $\mathbf{q}$ und $\lambda$. Ein solches Modell wird auch als *Independend Active Appearance Model* bezeichnet. Abbildung 4.14 zeigt beispielhaft die Synthese eines Gesichts aus Form- und Texturmodell.

Ein Möglichkeit um weitere lineare Abhängigkeiten zwischen den Parametervektoren $\mathbf{p}$ und $\lambda$ zu beseitigen wurde in [Cootes et al., 1998] vorgestellt. Hierbei handelt es sich um die sog. *Combined Active Appearance Models*. Bei diesem Ansatz werden lineare Abhängigkeiten zwischen den Form- und Texturparametern durch eine weitere Anwendung

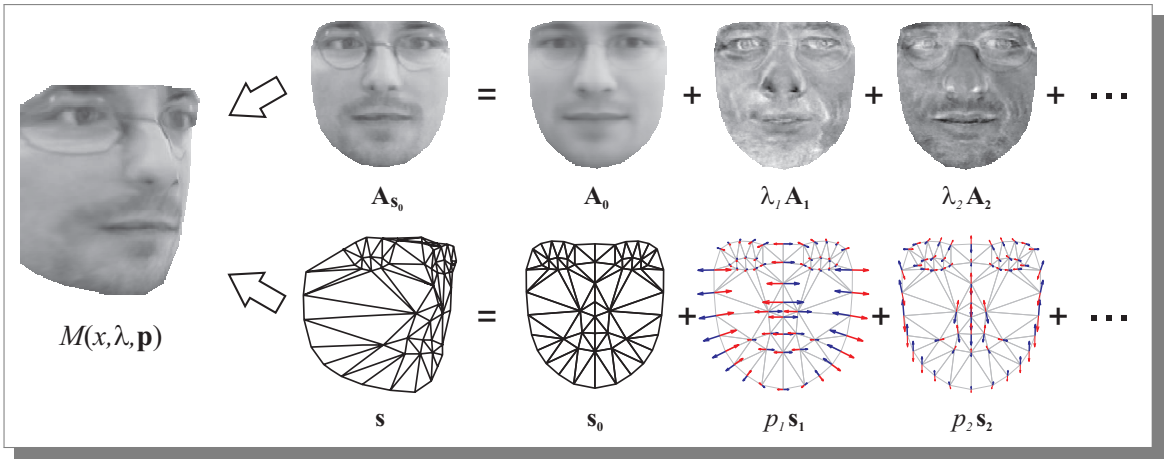



***Abbildung 4.14:*** *Synthese eines beispielhaften Gesichts durch Kombination des Form- und Texturmodells. Quelle: [Stricker, 2008]*

einer Hauptkomponentenanalyse über einen kombinierten Vektor $\mathbf{b} = f(\mathbf{p}, \lambda)$ entfernt. Ein Nachteil dieser Verfahrensweise ist, dass der sehr leistungsfähige *Project-Out-Algorithmus* (siehe Abschnitt C.3) für diese Modelle nicht mehr eingesetzt werden kann, da diesem eine getrennte Behandlung der Form- und Texturparameter zugrunde liegt. Daher werden die *Combined Active Appearance Models* im Rahmen dieser Dissertation nicht weiter betrachtet.

Zwecks Vereinfachung wird im Folgenden der Parametervektor der globalen Transformation $\mathbf{q}$ nicht weiter betrachtet, da dieser analog zum lokalen Parametervektor $\mathbf{p}$ behandelt werden kann.

### 4.4.2 Anpassungsalgorithmen

Im folgenden Abschnitt wird ein kurzer Überblick über verschiedene Algorithmen zur Anpassung eines Active Appearance Modells gegeben. Der Inhalt ist entnommen aus [Cootes et al., 1998, Cootes et al., 2001, Baker and Matthews, 2001, Baker and Matthews, 2004, Matthews and Baker, 2004]. Details zu den Verfahren finden sich in C.3.

Bei der Anwendung eines Active Appearance Modells besteht die typische Aufgabe darin, zu einem gegebenen Inputbild $I(\mathbf{x})$ die optimalen Parametervektoren für das gegebene Form- und Texturmodell zu bestimmen. Dabei müssen $\lambda$ und $\mathbf{p}$ so bestimmt werden, dass die Modellinstanz $M(x, \lambda, \mathbf{p})$ gemäß (4.8) und $I(\mathbf{x})$ möglichst gut übereinstimmen. Betrachtet man diese Aufgabe im Koordinatensystem des Inputbildes $I(\mathbf{x})$ ergibt sich folgendes Optimie-

rungsproblem:

$$\arg\min_{\mathbf{p},\lambda} \sum_{\mathbf{x}\in M} \left[ A_0(W(x,p)) + \sum_{i=1}^{m} \lambda_i A_i(W(x,p)) - I(\mathbf{x}) \right]^2 \tag{4.9}$$

Für einen effizienten Anpassungsalgorithmus ist es jedoch günstiger, das Problem im Koordinatensystem der Grundform $\mathbf{s_o}$ zu betrachten, da dies eine feste Größe besitzt und je nach eingesetztem Algorithmus hierdurch eine Reihe von aufwendigen Berechnungen vorab durchgeführt werden können. Für jedes Pixel $\mathbf{x}$ in der Grundform $\mathbf{s_0}$ liegt hierbei das korrespondierende Pixel im Inputbild bei $W(\mathbf{x},\mathbf{p})$. Der zu diesem Pixel gehörige Grauwert im Eingabebild ist $I(W(\mathbf{x},\mathbf{p}))$. Damit kann man ein einfaches Fehlermaß mit Hilfe der Summe der quadratischen Fehler der Grauwertdifferenzen definieren:

$$\arg\min_{\mathbf{p},\lambda} \sum_{\mathbf{x}\in \mathbf{s}_0} \left[ A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(W(\mathbf{x},\mathbf{p})) \right]^2 \tag{4.10}$$

Bei diesem Ansatz sind die Parametervektoren $\mathbf{p}$ und $\lambda$ über die Summe der quadratischen Fehler simultan zu minimieren. Dabei ist die Optimierung über $\mathbf{p}$ im Allgemeinen nichtlinear, die Optimierung über $\lambda$ linear. Die Pixelintensitätsdifferenzen im Koordinatensystem des AAMs werden auch als Fehlerbild $E(\mathbf{x})$ bezeichnet und wie folgt berechnet:

$$E(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(W(\mathbf{x},\mathbf{p})) \tag{4.11}$$

In der Regel wird zunächst $I(W(\mathbf{x},\mathbf{p}))$ berechnet, wobei für jedes Pixel der Grundform $\mathbf{s}_0$ das korrespondierende Pixel des Eingabebildes $I(\mathbf{x})$ mit der Transformationsfunktion $W(\mathbf{x},\mathbf{p})$ ermittelt wird. Dabei ist es sinnvoll, den Grauwert zwischen ganzzahligen Koordinaten zu interpolieren. Dieser Wert wird dann von dem aktuellen Grauwertbild der Modellinstanz $A(\mathbf{x})$ subtrahiert. Mit anderen Worten wird das Eingabebild auf die Grundform transformiert und von dem aktuellen AAM Grauwertbild subtrahiert.

Abbildung 4.15 zeigt die Entwicklung verschiedener AAM-Anpassungsalgorithmen, die diese beiden Varianten der Minimierung nach (4.9) oder (4.10) vornehmen. Die Algorithmen sollen im Folgenden jeweils kurz skizziert werden.

Ausgangspunkt für die Anpassung eines Active Appearance Models bildet der von Lucas und Kanade in [Lucas and Kanade, 1981] vorgestellte und nach den Autoren benannte Algorithmus. Das Hauptproblem dieses Algorithmus ist, dass dieser sehr rechenaufwendig ist und keinerlei Vorabberechnungen vorgenommen werden können. Sowohl die Jacobi- als auch die Hesse-Matrix müssen in jedem Iterationsschritt neu berechnet werden. Daher erreicht dieser Algorithmus nur eine sehr schlechte Performance und hat deshalb praktisch

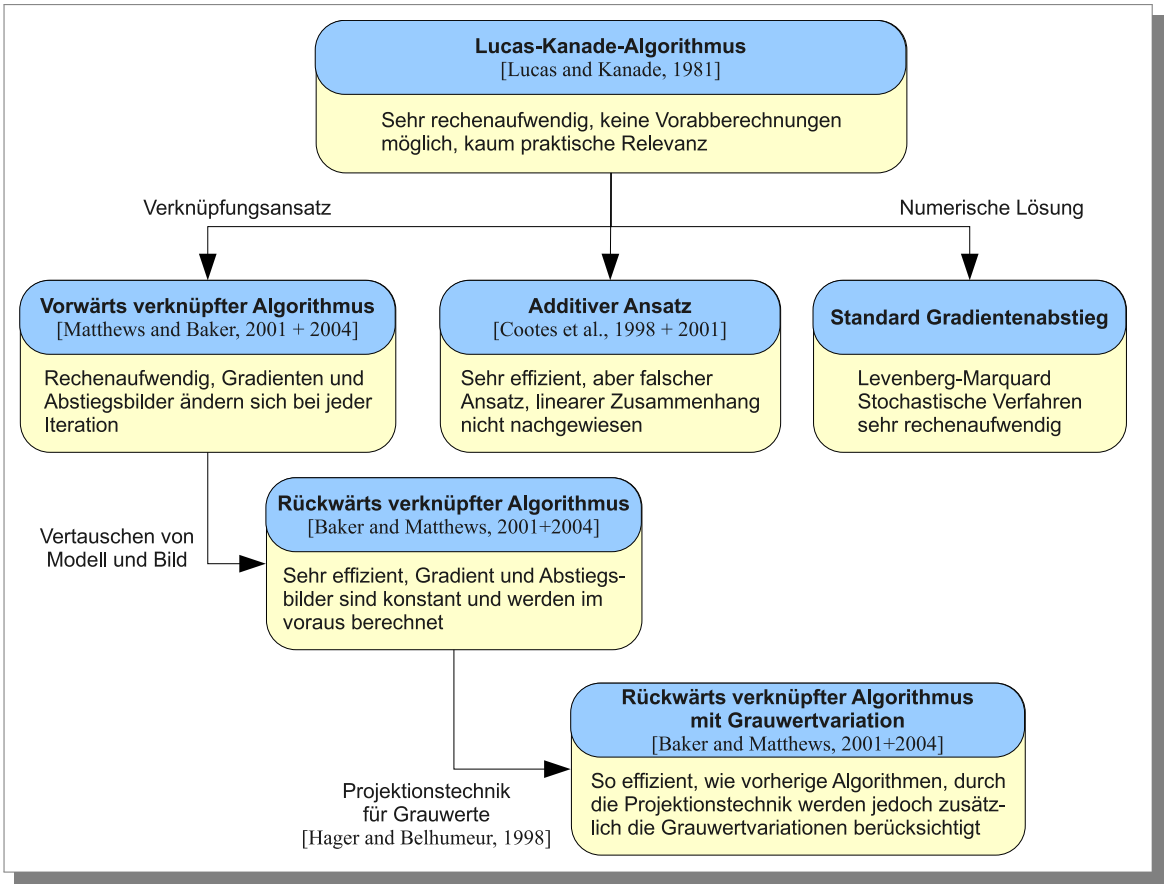



***Abbildung 4.15:*** *Entwicklung der AAM Anpassungsalgorithmen: Der Ausgangspunkt ist der Lucas-Kanade-Algorithmus. Über verschiedene Zwischenstufen, wurde der sehr leistungsfähige rückwärts verknüpfte Algorithmus mit Grauwertvariantion entwickelt.*

kaum Relevanz erreicht.

Ein weiterer Ansatz um den Term (4.10) zu lösen, ist die Verwendung eines Standard Gradientenabstiegsverfahren. Dazu gibt es zahlreiche Vorschläge, wie zum Beispiel mit der Levenberg-Marquard-Optimierung in [Sclaroff and Isidoro, 2003] oder mit stochastischen Verfahren, wie in [Blanz and Vetter, 1999]. Der Vorteil dieser Algorithmen ist, dass sie analytisch sind und ihre Konvergenzeigenschaften weithin bekannt sind. Ihr größter Nachteil liegt jedoch im Rechenaufwand. Die partiellen Ableitungen, die Hessematrizen und Gradientenrichtungen müssen in jeder Iteration wie beim Lucas-Kanade-Algorithmus neu berechnet werden. Daher haben diese Verfahren auch kaum eine praktische Relevanz.

Eine erste Verbesserung konnte mit dem *Additiven Ansatz* von [Cootes et al., 1998] [Cootes

et al., 2001] erreicht werden. Hierbei wird die Annahme getroffen, dass es einen linearen Zusammenhang zwischen dem Fehlerbild $E(\mathbf{x})$ und einem additiven Parameterinkrement für die Form und das Grauwertbild gibt. Auf Grund dieser Annahme können große Teile der Matrizen, die für eine Bildanpassung benötigt werden, vorberechnet werden. Daher ist die Algorithmus weit effizienter.

In [Matthews and Baker, 2004] wird aber gezeigt, dass der lineare Zusammenhang nur in einem sehr kleinen Bereich gilt. Daher wird von Matthews und Baker vorgeschlagen, anstatt eines linearen Parameterinkrements eine *Warpkomposition* einzusetzen. Dies führt zum *Vorwärts verknüpften Algorithmus ohne Grauwertvariation*. Ein Nachteil dieses Algorithmus ist aber, dass die benötigten Gradientenabstiegsbilder in jedem Interationsschritt neu berechnet werden müssen und daher das Verfahren nicht echtzeitfähig ist.

Die entscheidende Verbesserung der Performance konnte durch das Vertauschen der Rollen des Modells und des Eingangsbildes im sog. *Rückwärts verknüpften Algorithmus ohne Grauwertvariation* [Baker and Matthews, 2004] erreicht werden. Anstatt das Modell auf das Eingangsbild anzupassen, wird das Eingangsbild zurück auf die Grundform $\mathbf{s}_0$ transformiert. Dies wird als sogenannte *Warpinversion* bezeichnet. Als Folge davon, können die Jacobi- und Hessematrix vorberechnet werden. Damit ist dieser Algorithmus sehr effizient und ermöglicht eine Modellanpassung in Echtzeit.

Der Algorithmus mit der geringsten benötigten Rechenzeit pro Iterationsschritt ergibt sich aus der Kombination der Warpkomposition und der Warpinversion. Der entstehende Algorithmus kann weiterhin mit in [Hager and Belhumeur, 1998] vorgestellten Projektionstechniken zur Variation des Texturmodells ergänzt werden. Mit diesem Schritt werden die Texturvariationen aus den Gradientenabstiegsbildern „herausprojiziert“. Der entstehende Algorithmus wird daher auch als *Project-Out-Algorithmus* bezeichnet.

Aufgrund der effizienten Art der Anpassung wird dieser Algorithmus immer dann gewählt, wenn die Echtzeitfähigkeit der Modellanpassung gewährleistet werden soll. Dieser Algorithmus bildet daher auch die Grundlage dieser Dissertation. Als Implementierung des *Project-Out-Algorithmus* kommen dabei die Arbeiten von [Stricker, 2008] zum Einsatz. Diese wurden unter dem Gesichtspunkt der Echtzeitfähigkeit und der Tauglichkeit in realen Anwendungsszenarien entwickelt. Die Implementierung enthält zusätzlich die Möglichkeit verschiedene Vorverarbeitungsschritte (z.B. Glättung des Bildes oder Durchführung eines Histogrammausgleichs) im Rahmen der Modellanpassung durchzuführen.

Ein typischer Anpassungsverlauf eines Active Appearance Models auf einem Inputbild ist in Abbildung 4.16 dargestellt.

***Abbildung 4.16:*** *Iterationen des AAM Anpassungsalgorithmus. Während das erste Bild das Eingabebild zeigt, ist im zweiten zusätzlich die Modellinstanz mit der entsprechenden Initialisierung zu sehen. In den weiteren Bildern (Iteration 5, 15, 20 und 30) sind die Anpassungsschritte bis zur Konvergenz des Modells zu sehen. Quelle: [Werner, 2007]*

### 4.4.3 Verbesserung der Robustheit

Unter Realweltbedingungen zeigen alle im vorherigen Abschnitt vorgestellten Anpassungsalgorithmen eine Reihe von Schwächen. Einerseits ergeben sich Probleme aus unkontrollierbaren Umgebungsbedingungen der Realwelt (z.B. wechselnde Beleuchtungen oder Verdeckungen) und zum Anderen existieren aber auch eine Reihe von Problemen, die sich aus dem Aufbau der Modelle und deren Anpassung über einen Gradientenabstieg (z.B. lokale Minima) ergeben. Im Folgenden werden die typischen Probleme kurz skizziert und anschließend Lösungsmöglichkeiten aufgezeigt.

#### Probleme der Active Appearance Models

Unter den systembedingten Problemen spielen folgende Punkte die größte Rolle:

- **Wahl des Formmodells:**
  Die Auswahl und genaue Positionierung der Knotenpunkte beim Aufbau der Modelle beeinflusst die spätere Anpassung. Wird ein Modell mit ungünstig gewählten Knotenpunkten erstellt (z.B. Knotenpunkte an sehr unspezifischer Position) oder weisen die gesetzten Knotenpunkte eine unzureichende Genauigkeit auf, wird die Anpassung deutlich negativ beeinflusst.

- **Abhängigkeit vom Trainingsdatensatz:**
  Während der Modellerstellung werden sowohl das Form- als auch das Texturmodell basierend auf der Statistik der vorliegenden Trainingsdaten erstellt. Die Trainingsdaten definieren somit auch den Raum, in dem das resultierende Modell später sinnvoll eingesetzt werden kann. Der Datensatz sollte daher immer so gewählt werden, dass möglichst alle Variationen von Form und Textur, die für den späteren Einsatzzweck relevant sind, auch im Datensatz entsprechend repräsentiert sind. Dazu gehören z.B. auch die Kopfausrichtung, der Gesichtsausdruck, die Beleuchtungsverhältnisse und auch verschiedene Personen. Hierbei ist auch zu beachten, dass Modelle mit einer sehr hohen Generalisierungsfähigkeit und Vielfalt von Details typischerweise auch viele Parameter besitzen. Dies kann sich aber auch negativ auf den Gradientenabstieg auswirken (siehe nächster Punkt).

- **Lokale Minima beim Gradientenabstieg:**
  Die Anpassung eines Active Appearance Models stellt ein hochdimensionales Optimierungsproblem dar. Typischerweise ergibt sich dabei ein komplexes Fehlergebirge, das eine Vielzahl lokaler Minima aufweist. Je nach Güte der Initialisierung und Form des Fehlergebirges kann eine gute Optimierung stattfinden, aber auch ein Abdriften in ein lokales Minimum ist möglich. In [De la Torre et al., 2007] wird gezeigt, wie bereits kleine Abweichungen bei der Initialisierung zu Problemen führen können. Ein guter Anpassungsalgorithmus muss in der Lage sein, mit lokalen Minima umzugehen oder die Existenz von lokalen Minima zu reduzieren. Ein Ansatz zur Lösung des Problems besteht in der Nutzung einer Hierarchie von Modellen mit jeweils ansteigender Komplexität. Problematisch hierbei ist die Frage, zu welchem Zeitpunkt oder nach welchen Kriterien die Umschaltung zu einem Modell mit höherer oder niedrigerer Komplexität erfolgen soll.

- **Oszillation der Parameter:**
  Die von den Anpassungsalgorithmen durchgeführten Schritte sind eine lineare Approximation und stellen somit nur eine Näherung dar. Bei Algorithmen ohne Möglichkeit die Schrittweite der Parameter zu beeinflussen, besteht die Gefahr, dass um das gesuchte Minimum hin und her gesprungen wird. In der Praxis wurden Situationen beobachtet, bei denen in aufeinander folgenden Anpassungsschritten die gleiche Parameteränderung nur mit unterschiedlichem Vorzeichen ausgeführt wurde.

- **Spezielle Probleme des Project-Out-Algorithmus:**
  Ein wesentliches Problem beim Project-Out-Algorithmus ist auf die Art und Weise der Anpassung zurückzuführen: Die eigentlich notwendige gleichzeitige Minimierung von Form- und Texturparametern erfolgt sequentiell. Die Formparameter werden zuerst in einem texturfreien Raum berechnet. Die Texturparameter werden anschließend separat ermittelt. Dies setzt voraus, dass sich in dem texturfreien Raum überhaupt eine Lösung

finden lässt. In [Baker et al., 2003b] wird gezeigt, dass dies nicht immer der Fall sein muss.
Weiterhin spielt das Mittelwertbild des Texturmodells eine wichtige Rolle, da dies über die darin enthaltenen Gradienten direkt in die Berechnung der Abstiegsbilder eingeht. Wenn keine (oder nur sehr schwache Gradienten) vorhanden sind, gehen die Abstiegsbilder gegen 0 und eine Parameteranpassung ist nicht mehr möglich.
Somit spielt das Texturmodell für den Project-Out-Algorithmus eine wichtige Rolle. Die praktische Relevanz wird in [Gross et al., 2005] gezeigt: Es wird gezeigt, dass ein Algorithmus der gleichzeitig Form- und Texturparameter optimiert (*Simultaneous Inverse Compositional Algorithm*) bessere Anpassungsraten (nahezu 100%) erreicht, der Project-Out-Algorithmus auf gleichen Daten jedoch nur $50 - 70\%$.

Bezogen auf die geforderte Realwelttauglichkeit auf einem mobilen Robotersystem ergeben sich folgende Aspekte, die die Anpassung des Modells beeinflussen:

- **Beleuchtungsverhältnisse** Der größte Störfaktor beim Einsatz unter Realweltbedingungen sind die wechselnden Beleuchtungsverhältnisse. Künstliche Lichtquellen oder seitlich in einen Raum einfallendes Licht können zu starken Schlagschatten und einer damit verbundenen sehr ungleichmäßigen Ausleuchtung des Gesichts führen. Da Active Appearance Models typischerweise auf den Intensitätswerten des Kamerabildes arbeiten, ist eine Berücksichtigung unterschiedlicher Beleuchtungsverhältnisse unumgänglich. Beim Einsatz unterschiedlicher Kamerasysteme sind weiterhin die spezifischen Eigenschaften wie z.B. die Konstrastverhältnisse oder die Verschiebung der Helligkeitswerte zu berücksichtigen.

- **Verdeckungen** Die vorgestellten Algorithmen gehen davon aus, dass das Gesicht immer vollständig im Bild zu sehen ist. Das Fehlerbild wird immer ganzheitlich und in ungewichteter Form betrachtet. Durch Verdeckungen (z.B. Bart oder Brille) wird das Fehlerbild jedoch so beeinflusst, dass beim Gradientenabstieg (basierend auf den Jacobimatrizen und der Hessematrix) falsche Parameter entstehen können.

- **Kontinuierliches Tracken** Im praktischen Einsatz soll typischerweise nicht nur ein einzelnes Bild eines Gesichts ausgewertet werden, sondern ein Gesicht soll über eine Bildsequenz getrackt und das Modell kontinuierlich angepasst werden. Eine globale Initialisierung in jedem neuen Bild und anschließend eine lokale Modellanpassung durchzuführen, ist einerseits wenig praktisch und auf der anderen Seite nur begrenzt echtzeitfähig, da eine globale Initialisierung meist sehr rechenintensiv ist. Im Rahmen einer Bildsequenz kann das Ergebnis des vorherigen Bildes als aktuelle Initialisierung verwendet werden. Bei relativ kleinen Bewegungen des Gesichts im Bild entstehen sehr gute Ergebnisse. Bei größeren Bewegungen besteht aber auch die Gefahr, dass die Anpassung in einem lokalen Minimum endet.

### Lösung der Probleme zur Erhöhung der Robustheit

Da eine möglichst gute Modellanpassung an ein gegebenes Inputbild Grundvoraussetzung ist, um basierend auf den Modellparametern eine Schätzung der Kopfpose vornehmen zu können, wurden in [Stricker et al., 2009] und [Martin and Gross, 2008] eine Reihe von Möglichkeiten zur Verbesserung der Robustheit des Anpassungsprozesses vorgestellt, die im Folgenden kurz zusammengefasst werden.

**Robustheit gegen wechselnde Beleuchtungsverhältnisse** In [Zou et al., 2007] wurde der Einfluss verschiedener Beleuchtungen in Bezug auf die Gesichtserkennung untersucht. Eine vollständige Modellierung der vorliegenden Beleuchtungsverhältnisse würde optimale Ergebnisse liefern, erfordert jedoch exakte Kenntnisse über die vorherrschenden Bedingungen und ist auch nur bedingt in Echtzeit möglich. Als eine Alternative wurde ein biologisch motivierter Ansatz in [Jobson et al., 1997b] präsentiert. Hierbei handelt es sich um den sog. *Retinex Filter*, der die Signalverarbeitung des menschlichen Auges nachbildet. Dazu wird jedes Pixel an Position $(x, y)$ in Bezug auf seine lokale Nachbarschaft betrachtet. Mathematisch wird diese Faltung wie folgt beschrieben:

$$R(x,y) = log\ I(x,y) - log\,|F(x,y) * I(x,y)| \tag{4.12}$$

Dabei ist $I$ das Inputbild und $F$ steht für eine Funktion, die die Pixel in der Nachbarschaft von $(x, y)$ beschreibt. Die Größe der Nachbarschaft muss dabei problemspezifisch gewählt werden. Eine ungünstige Wahl kann dabei zu sog. "Geisterbildern", einem Verlust von Details oder einer schlechten Normalisierung der Beleuchtung führen.

Basierend auf den Arbeiten von [Jobson et al., 1997a] und [Wang et al., 2004] wurde in [Stricker et al., 2009] der *Adaptive Retinex Filter* vorgestellt. Dieser kombiniert die Vorteile des *Multiscale Retinex* aus [Jobson et al., 1997a] mit einer lokalen Dynamikfunktion (vgl. [Wang et al., 2004]). Dazu werden zwei Retinex Filter $R_1(x, y)$ und $R_2(x, y)$ mit unterschiedlich großer Nachbarschaft und ein Kantendetektor $E(x, y)$ verwendet:

$$S(x,y) = \begin{cases} R_1(x,y), & E(x,y) < l_{lower} \\ R_2(x,y), & E(x,y) > l_{upper} \\ \frac{E(x,y)}{l_{upper}} R_1(x,y) + \left(1 - \frac{E(x,y)}{l_{upper}}\right) R_2(x,y), & l_{lower} \leq E(x,y) \leq l_{upper} \end{cases} \tag{4.13}$$

Hierbei definieren $l_{lower}$ und $l_{upper}$ die Schwellwerte für die Verwendung der beiden Retinex-Filter $R_1(x, y)$ und $R_2(x, y)$ in Bezug auf die lokalen Gradientenstärke $E(x, y)$. Zwischen $l_{lower}$ und $l_{upper}$ erfolgt eine lineare Interpolation zwischen beiden Filterantworten. Im Gegensatz zu [Wang et al., 2004] kann diese Variante auch deutlich schneller berechnet werden. Abbildung 4.17 veranschaulicht die Wirkung des Adaptive Retinex Filters.

Eine weitere Möglichkeit zur Erhöhung der Robustheit unter verschiedenen Beleuchtungsbe-

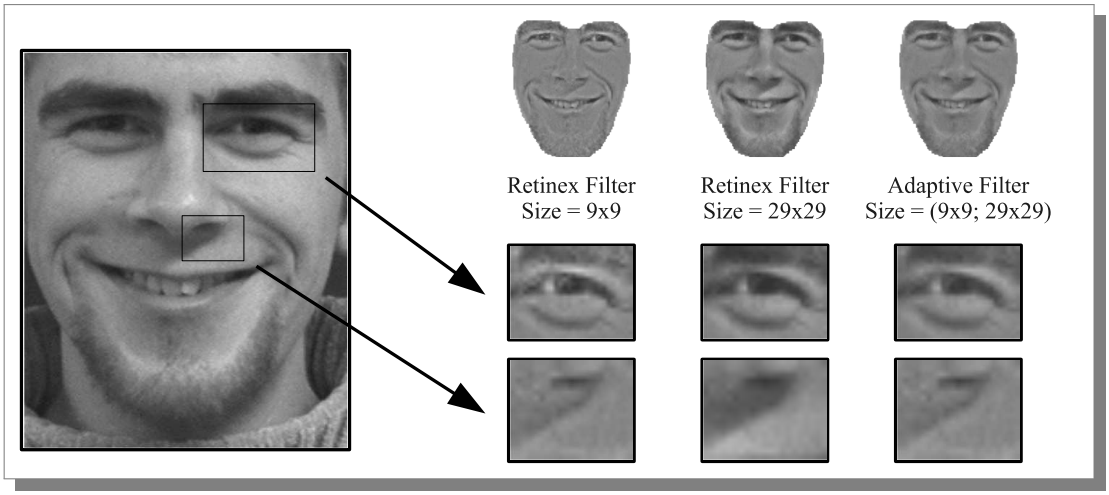



***Abbildung 4.17:*** *Beispiel zum Adaptive Retinex Filter: Der Standard Retinex Filter mit einer Umgebungsgröße von 9x9 Pixeln erreicht im Bereich des Mundes eine gute Beleuchtungskorrektur. Beim Auge treten jedoch kleine Störungen auf. Bei einer Umgebungsgröße von 29x29 Pixeln sind die Störungen im Bereich des Auges reduziert, jedoch verschlechtert sich die Beleuchtungskorrektur (Schatten unter der Nase). Der Adaptive Retinex Filter kombiniert die Vorteile beider Varianten durch die selektive Auswahl der Filtergröße und Interpolation. Quelle: [Stricker, 2008]*

dingungen wurde in [Martin and Gross, 2008] durch die Verwendung eines Kantenmodells vorgestellt. Dabei wird das gesamte Texturmodell nicht auf einem Grauwertbild, sondern auf einem Gradientenbild betrachtet. Das normierte Inputbild $I(W(\mathbf{x}, p))$ wird dazu mit zwei Kantenfiltern $G_x$ und $G_y$ gefaltet:

$$S_x = I(W(\mathbf{x}, p)) * G_x \quad , \quad S_y = I(W(\mathbf{x}, p)) * G_y \tag{4.14}$$

Für die nachfolgenden Berechnungen im gesamten Anpassungalgorithmus wird statt des Grauwertes die Gradientenstärke $S$ verwendet:

$$S = \sqrt{S_x^2 + S_y^2} \tag{4.15}$$

Abbildung 4.18 zeigt die Anwendung des Gradientenfilters auf ein Inputbild.

In [Martin and Gross, 2008] wurde gezeigt, dass diese Art der Vorverarbeitung die Modellanpassung verbessert. Auf einem Referenzdatensatz erreichte ein normales Grauwertmodell mit einer Gauß-Glättung und einem Histogrammausgleich als Vorverarbeitung eine Anpassungsrate von 73%. Mit Hilfe des Kantenmodells konnte eine Quote von 84% erreicht werden. Weiterhin konnte gezeigt werden, dass durch die Anwendung des Kantenmodells auch die Klassifikationsergebnisse in Bezug auf die Mimikerkennung verbessert werden können.

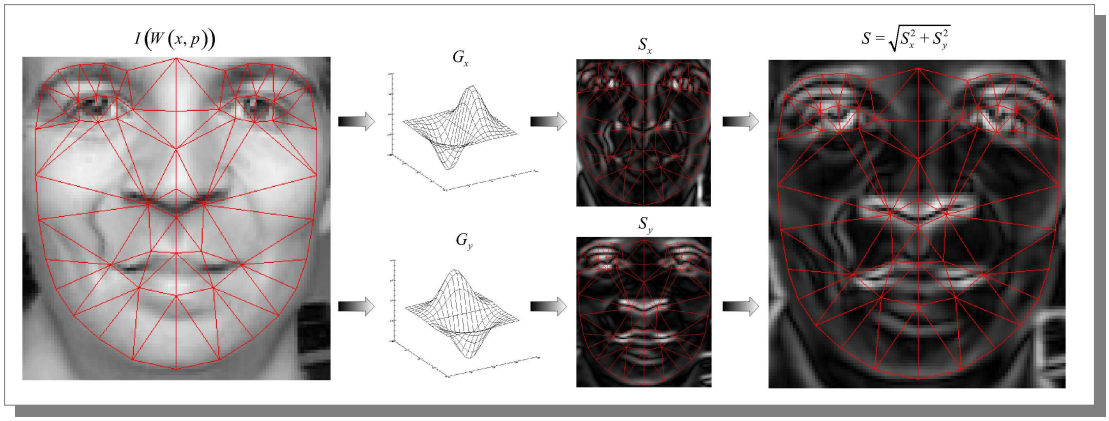



***Abbildung 4.18:*** *Anwendung des Gradientenfilters auf ein Inputbild: Zunächst wird das Inputbild mit $G_x$ und $G_y$ gefaltet. Das Gradientenbild ergibt sich aus der Berechnung der Gradientenstärke $S$.*

**Adaptive Parameter Fitting** Der *Project-Out-Algorithmus* arbeitet auf Basis eines Gradientenabstiegs. Wie im vorangegangenen Abschnitt beschrieben, besteht hierbei das Problem, in ein lokales Minimum zu gelangen. Eine Möglichkeit um dies zu vermeiden, ist die Verwendung eines hierarchischen Modells, wie in [Martin and Gross, 2008] beschrieben wurde. Problematisch hierbei ist jedoch die Bestimmung einer entsprechenden Umschaltbedingung zwischen verschiedenen Modellhierarchien.

In [Stricker et al., 2009] wird eine Variante zur adaptiven Parameteranpassung vorgestellt, die auf Basis der Sortierung der Eigenvektoren nach der PCA arbeitet. Im Rahmen der PCA werden die Eigenvektoren nach Größe der zugehörigen Eigenwerte sortiert. Somit repräsentieren die ersten (größten) Eigenvektoren die Komponenten, die die größte Varianz in den Daten verursachen.

Im Standard Project-Out-Algorithmus werden die Parameter aller Eigenvektoren gleichzeitig angepasst. Dies kann dazu führen, das die Komponenten mit kleineren Eigenwerten divergieren, da zu Beginn der Anpassung die Komponenten mit größeren Eigenwerten noch nicht optimal angepasst sind und daher die anderen Komponenten möglicherweise in eine falsche Richtung adaptiert werden.

Im Rahmen des *Adaptive Parameter Fittings* [Stricker, 2008] werden die Modellparameter in zwei Gruppen aufgeteilt: In der ersten Gruppe werden die sog. *primären* Parameter zusammengefasst, die die größten Varianzen in den Trainingsdaten beschreiben. Dazu gehören typischerweise vor allem die Parameter, die Lage, Pose und prinzipielle Form des Gesichts beschreiben. In der zweiten Gruppe werden die sog. *sekundären* Parameter zusammengefasst, die die innere Struktur des Gesichts beschreiben.

Während der Modellanpassung werden die sekundären Parameter in Abhängigkeit der Änderung der primären Parameter angepasst. Hierzu wird die Energie der primären Parameteränderungen $(\Delta p_1, ..., \Delta p_n)$ als Summe der quadrierten und normierten Parameteränderungen $E_p$ betrachtet:

$$E_p = \sum_{i=1}^{n} \left( \frac{\Delta p_i}{EV\,(p_i)} \right)^2 \tag{4.16}$$

Die Parameteränderung der sekundären Parameter $(\Delta p_{n+1}, ..., \Delta p_s)$ wird berechnet als:

$$\Delta p_i = \Delta p_i \cdot scale \tag{4.17}$$

wobei der Faktor *scale* zwischen den Schranken $l_l$ und $l_u$ logarithmisch angepasst wird:

$$scale = \begin{cases} 0 & l_u < E_p \\ log_{(l_u/l_l)} \left( \frac{E_p}{l_l} \right) & l_u < E_p < l_l \\ 1 & E_p < l_l \end{cases} \tag{4.18}$$

Dabei beschreibt $l_l$ die untere und $l_u$ die obere Anpassungsgrenze. Überschreitet $E_p$ einen Wert von $l_u$, findet entsprechend keine Anpassung der sekundären Parameter statt, da sich die primären Parameter noch nicht stabilisiert haben. Unterschreitet die Energie von $E_p$ hingegen die untere Grenze $l_l$, erfolgt die Anpassung der sekundären Parameter in vollem Umfang. Somit werden falsche oder ungünstige Anpassungen der sekundären Parameter vermieden. Abbildung 4.19 zeigt die Wirkung der adaptiven Anpassung am Beispiel zweier Parameter.

**Oszillation der Parameter** Ein weiteres für den *Project-Out-Algorithmus* typisches Problem ist die bereits beschriebenen Parameteroszillation: Ohne Anpassung der Schrittweite der Parameteränderung, besteht die Gefahr, dass die Parameter um ein Minimum herum springen. Weisen die Parameteränderungen in aufeinander folgenden Anpassungsschritten ein anderes Vorzeichen auf, ist zu vermuten, dass ein Minimum im Fehlergebirge übersprungen wurde. Durch die fehlende Skalierung der Parameteränderungen ist es dem Algorithmus in der Standardvariante jedoch nicht möglich, dieses Minimum zu erreichen.

Ein sehr einfacher und dennoch effektiver Lösungsansatz besteht in der Verringerung der Schrittweite des betreffenden Parameters, sobald dieser einen wiederholten Vorzeichenwechsel aufweist. Diese Modifikation kann relativ leicht und ohne Verlust der Performance in den Project-Out-Algorithmus integriert werden. Abbildung 4.20 zeigt die beispielhafte Auswirkung dieser Modifikation während einer Modellanpassung. Durch die Skalierung weisen die Parameter einen deutlich stabileren Verlauf auf. Das Oszillieren der Parameter, welches besonders stark ab der 10. Iteration auftritt, kann durch die Einführung des Skalierungsfaktors

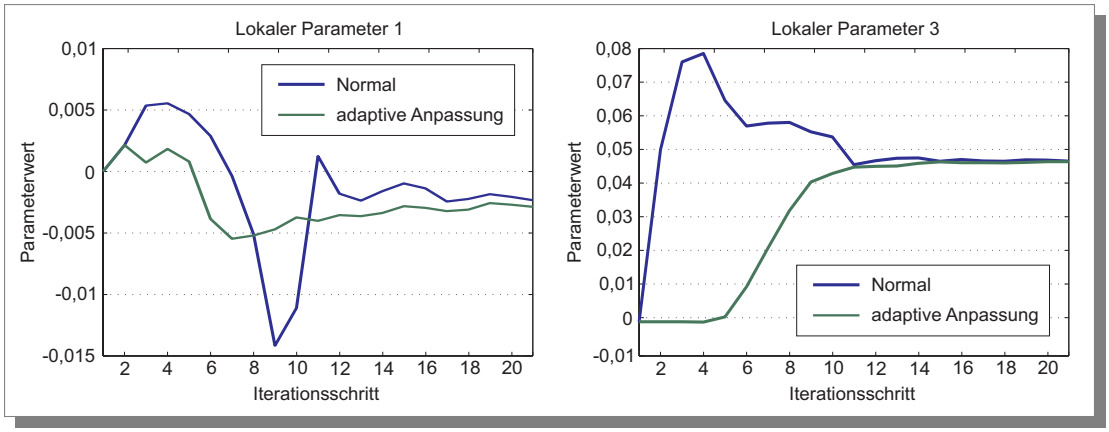



***Abbildung 4.19:*** *Vermeiden des Schwingens lokaler Parameter: Durch die priorisierte Anpassung der primären Parameter weisen die sekundären Parameter einen zielgerichteteren Verlauf auf. Die Anpassung des sekundären lokalen Parameters 3 erfolgt erst, nachdem sich die primären Parameter 1 und 2 (nicht im Bild) im Anpassungsschritt t=5 stabilisiert haben.*

deutlich reduziert werden.

**Verdeckungen** Bei Verdeckungen sind zwei Fälle zu betrachten: *Objektverdeckungen* und *Selbstverdeckungen*. Im ersten Fall verdeckt ein Objekt (z.B. eine Brille oder eine Hand) einen Teil des Gesichts. Im zweiten Fall wird ein Teil des Gesichts beispielsweise durch eine seitliche Kopfdrehung unsichtbar. Abbildung 4.21 zeigt einige Beispiele hierzu.

Im Rahmen der Modellanpassung wird zur Behandlung von Verdeckungen der Begriff des *Outliers/Ausreißers* eingeführt [Huber, 1981]. Darunter werden einzelne Pixel oder Bildbereiche des Inputbildes $I(x)$ verstanden, die trotz Modellanpassung nicht durch das Modell abgebildet werden können und im Fehlerbild $E(x)$ hierdurch negativen Einfluss auf die Güte der Modellanpassung haben. Ziel ist es also, solche Ausreißer zu detektieren und bei der Modellanpassung separat zu behandeln bzw. entsprechend zu ignorieren.

Im Rahmen dieser Dissertation werden Verdeckungen bei der AAM-Anpassung nur am Rande betrachtet. Daher soll hier nur auf weiterführende Literatur zu diesem Thema verwiesen werden: [Cootes et al., 2000], [Baker et al., 2003a], [Theobald et al., 2006], [Gross et al., 2006] und [Rühle, 2009][2].

In [Stricker et al., 2009] durchgeführte Tests auf der IMM-Datenbank [Nordstrøm et al., 2004] haben ergeben, dass durch die Anwendung der in den letzten Abschnitten vorgestellten Modifikationen der Modellanpassung eine etwa 10%-15% bessere Anpassung möglich ist.

[2]Diese Bachelorarbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

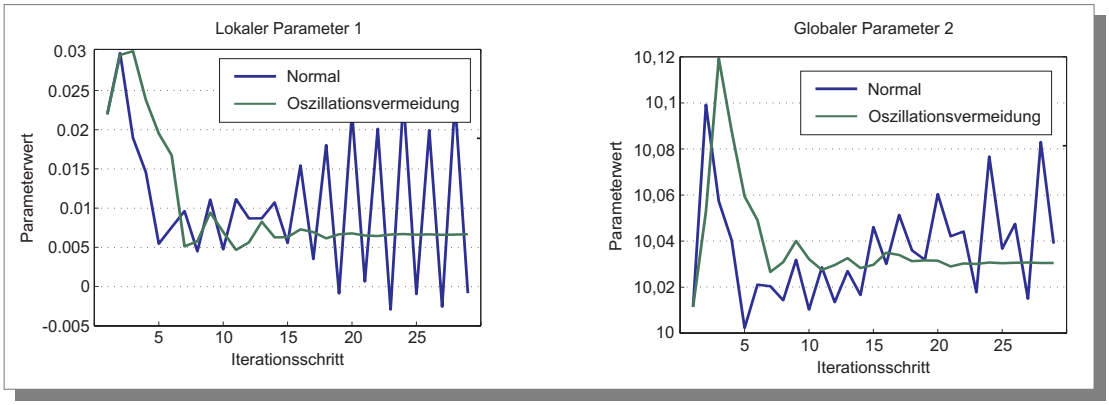



***Abbildung 4.20:*** *Vermeiden des Oszillierens der Parameter: Durch die Skalierung der Parameteränderung kann ein Oszillieren von beiden Parametern besonders ab der 10. Iteration vermieden werden. (Blau: Parameterverlauf ohne Oszillationsvermeidung, Rot: Parameterverlauf mit aktiver Oszillationsvermeidung.)*

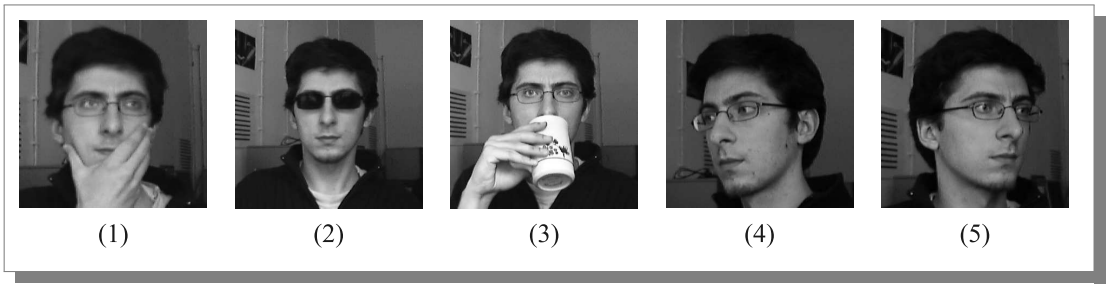



***Abbildung 4.21:*** *Beispiele für Verdeckungen im Gesicht: Die Bilder (1)-(3) zeigen Objektverdeckungen durch Hand, Brille und eine Tasse. Eine Selbstverdeckung ist in (4) und (5) dargestellt. Quelle: [Rühle, 2009]*

## 4.5 Ergebnisse

Im folgenden Abschnitt werden die erzielten Ergebnisse bei der Schätzung der Blickrichtung auf Basis der Active Appearance Modelle beschrieben. Dazu wird zunächst die Datenbank vorgestellt, die zur Untersuchung der Stabilität und Qualität der Modellanpassung verwendet wurde. Anschließend wird erläutert, wie die Kopfposen-relevanten Modellparameter bestimmt werden können und welche möglichen Verfahren zur Kopfposenschätzung geeignet sind. Die Kopfposenschätzung selbst wird anschließend auf einer Reihe von Videos mit und ohne vorhandenen *Ground-Truth-Informationen* ausgewertet.

### 4.5.1 Beschreibung der Datenbank

Für eine Reihe von Untersuchungen wurde am Fachgebiet Neuroinformatik und Kognitive Robotik eine eigene Datenbank mit Bildern von 28 Personen verschiedener Altersklassen und Geschlecht erstellt [Babies, 2007][3]. Hierzu wurden Videosequenzen von den Personen aufgenommen, während diese den Kopf in natürlicher Art und Weise bewegten. Hiermit konnten Daten (Einzelbilder und Videosequenzen) für das Training und das Testen der Kopfposenschätzung gewonnen werden, die eine Grundlage zur Schätzung des Interaktionsinteresses bildet. Um eine möglichst große Abdeckung bei den Kopfposen zu erreichen, wurde ein Winkelbereich von $-45°$ bis $+45°$ in horizontaler Richtung und $-20°$ bis $+45°$ in vertikaler Richtung festgelegt. Das seitliche Kippen des Kopfes (*roll*-Winkel) wurde vernachlässigt. Diese Bereiche wurden in jeweils elf Klassen aufgeteilt, so dass insgesamt 121 Winkelklassen entstanden sind. Bei den Videoaufnahmen wurde von jedem Teilnehmer mindestens ein Bild aus jeder Winkelklasse aufgenommen. Für die Bestimmung der aktuellen Kopfpose wurde ein *Flock of Bird*[4] genutzt. Aus diesen Videosequenzen wurden pro Person je 25 Bilder verschiedener Posen gelabelt. Die Daten wurden anschließend gespiegelt, so das insgesamt 1400 gelabelte Gesichtsbilder existieren. Abbildung 4.22 zeigt einige Beispielbilder aus der Datenbank.

***Abbildung 4.22:*** *Die Abbildungen zeigen verschiedene Kopfposen, die die maximalen Winkelauslenkungen von ± 45° in horizontaler Richtung und −20° bis +25° in vertikaler Ausrichtung zeigen. Dieser Bereich wurde für die Erzeugung der AAMs genutzt. Auf dem Kopf der Probanden ist der Flock of Bird Sensors zu erkennen.*

Für die Modellerstellung und die Durchführung der Experimente fand eine Unterteilung der 1400 gelabelten Personendaten in zwei personendisjunkte Gruppen statt, d.h. Personen deren Gesichtsbilder in der ersten Datenbank enthalten sind, sind nicht in der zweiten Datenbank

[3] Diese Diplomarbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

[4] Beim *Flock of Bird* handelt es sich um einen auf dem Kopf tragbaren Sensor, der mit Hilfe eines Magnetfeldes und einer entsprechenden Basisstation die drei Winkelauslenkungen der Kopfpose messen kann.

enthalten. Die erste Datenbank umfasst 700 Bilder und wurde zur Erstellung der AAMs genutzt. Die zweite Datenbank umfasst ebenfalls 700 Bilddaten und wurde für die Tests und Auswertungen verwendet.

### 4.5.2 Stabilität und Qualität der Anpassung

Zunächst wurde untersucht, wie gut es möglich ist, AAM-Modelle, die auf Basis der Trainingsdatenbank erstellt wurden, auf die Bilder der Trainings- und der Testdatenbank anzuwenden bzw. anzupassen. Für diese Untersuchung wurde der *Project-Out-Algorithmus* (siehe Abschnitt C.3) verwendet. Zur Bestimmung der Anpassungsgüte wurde ein einfaches Abstandsmaß eingesetzt: Ein Gesicht wird als korrekt beschrieben angenommen, wenn die Summe der absoluten Differenzen zwischen der geschätzten Form und den bekannten Labeldaten einen festgelegten Schwellwert nicht überschreitet:

$$E_L = \frac{1}{|\mathbf{x}|} \sum_{j=1}^{n} |\mathbf{x}_j - \mathbf{x}_{j,\ approx}| \tag{4.19}$$

Als weiteres Maß bei der Anpassung wurde die Grauwertdifferenz

$$E_M = \sum_{\mathbf{x} \in \mathbf{s}_0} |A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))| \tag{4.20}$$

zwischen dem Mittelwertgesicht $A_0(\mathbf{x})$ und dem mit Hilfe der Schätzung formnormierten Eingangsgesichtes $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ genutzt. Dieses Kriterium wird als Abbruchkriterium für den Anpassungsalgorithmus genutzt, da es relativ schnell zu berechnen ist und keine Informationen aus der Datenbank benötigt.

Bezogen auf das Grauwertmodell kann auch die Summe der Differenzen zwischen den geschätzten Grauwerten und den Grauwerten des formnormierten Eingabegesichtes genutzt werden:

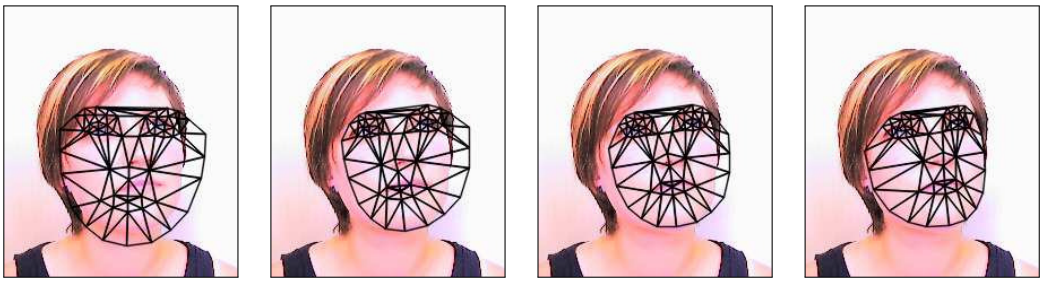
$$E_G = \sum_{\mathbf{x} \in \mathbf{s}_0} \left| A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right| \tag{4.21}$$

Auch diese Differenz benötigt kein zusätzliches Wissen aus der Datenbank. Dadurch lässt sie sich ebenfalls gut als Abbruchkriterium für die Anpassung nutzen. Die Berechnung ist im Vergleich zu (4.20) aufwendiger, da in jedem Iterationsschritt zusätzlich die Berechnung der Grauwertparameter notwendig wird.

Der Wert von $E_M$ als auch bei $E_G$ wird auch durch die Größe des formnormierten Gesichtes beeinflusst, also der Skalierung der mittleren Form $s_0$. Im Rahmen der durchgeführten Experimente wurde eine Größe von 70x70 Pixeln verwendet. Die Nutzung der Grauwertdif-

ferenz besitzt den Vorteil, dass anders als bei der Nutzung der Differenz der Formen, diese direkt aus dem geschätzten Modell errechnet werden kann und kein Vorwissen, wie z.B. die manuell gesetzten Labelpunkte, benötigt.

Abbildung 4.23 zeigt verschiedene Anpassungsstufen eines Modells auf ein Eingabebild mit den zugehörigen Werten für $E_L$, $E_M$ und $E_G$. Basierend auf einer Reihe von Beobachtungen wurde der Wert $E_L \leq 2.6$ als Schwellwert für ein korrekt angepasstes Modell festgelegt. Abhängig von dem zugrunde liegenden Datensatz und dem verwendeten Modell wird dieser Wert verschieden sein.



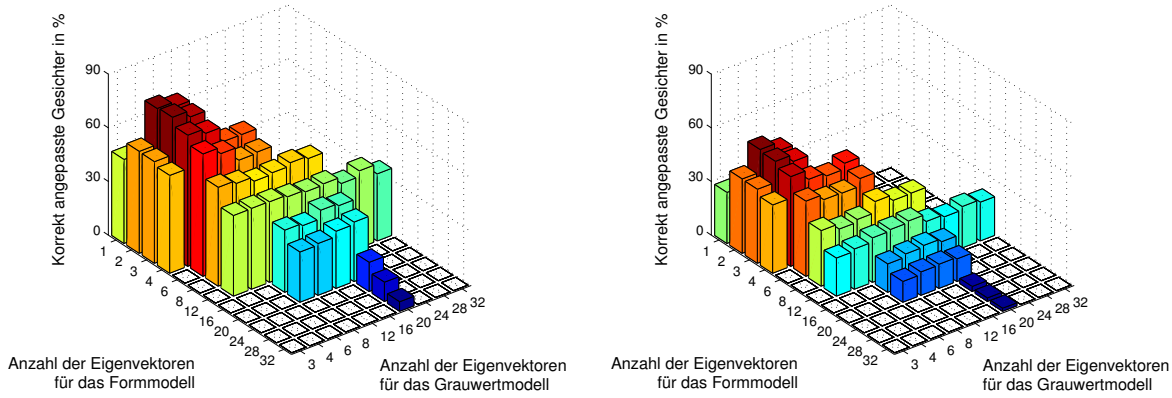
| | | | | |
|---|---|---|---|---|
| $E_L$ | 7.56 | 4.41 | 2.87 | 1.63 |
| $E_M$ | 45.11 | 34.15 | 29.08 | 26.22 |
| $E_G$ | 35.47 | 29.71 | 24.84 | 18.77 |

***Abbildung 4.23:*** *Die Abbildung zeigt verschiedene Formschätzungen und den dazugehörigen Formfehler $E_L$ bezüglich der optimalen Form, den Grauwertfehler $E_M$ bezüglich des mittleren Gesichtes und den Grauwertfehler $E_G$ bezüglich der geschätzten Grauwertparameter.*

#### Anpassungsquote in Abhängigkeit von der Modellkomplexität

Zunächst wurde die Anpassungsfähigkeit verschiedener AAMs auf der Testdatenbank quantitativ untersucht. Dabei wurden Modelle mit unterschiedlicher Anzahl von Form- und Grauwertparametern auf der Trainingsdatenbank erstellt und anschließend auf den Bildern des Testdatensatzes angewendet. Als Referenz wurde zusätzlich die Anpassung auf dem Trainingsdatensatz selbst durchgeführt.

Abbildung 4.24 zeigt die erreichten Anpassungsraten. Es ist deutlich zu sehen, dass die Fähigkeit zur Anpassung der Modelle mit der Erhöhung der Anzahl der genutzten Eigenvektoren abnimmt. Einige Gründe für diesen Effekt wurden im Abschnitt 4.4.3 erläutert. Es ist jedoch auch ersichtlich, dass eine gewisse Mindestanzahl von Parametern benötigt wird, da sonst auch keine korrekte Anpassung möglich ist. Die besten Ergebnisse wurden bei jeweils 3-4 Form- und Grauwertparametern erreicht. Auf den bekannten Gesichtern der Trainingsdatenbank konnten ca. 60-70% wieder korrekt angepasst werden. Bei den




Trainingsdatenbank Testdatenbank

***Abbildung 4.24:*** *Die beiden Grafiken zeigen den jeweils prozentualen Anteil der korrekt angepassten Gesichter in Abhängigkeit von der Komplexität des Modells auf der Trainings- und Testdatenbank. Bei allen Versuchen ist deutlich erkennbar, dass die Fähigkeit zur Anpassung mit der Erhöhung der Anzahl genutzter Eigenvektoren abnimmt.*

unbekannten Gesichtern der Testdatenbank liegt die Quote bei etwa knapp 50%.

Es gibt zwei Ursachen für diese relativ schlechten Ergebnisse: Einerseits wurden die Tests auf zufällig ausgewählten Bildern mit beliebiger Kopfpose durchgeführt. Dabei hat sich gezeigt, dass die Anpassung auf ein Bild mit großer Auslenkung des Kopfes (ohne Kenntnis der Historie) deutlich schlechter funktioniert, als auf ein frontales Gesicht (siehe nachfolgende Seiten). Ein weiterer Grund ist die relativ hohe Schwelle, die für eine als korrekt gezählte Anpassung gewählt wurde.

### Anpassungsgüte in Abhängigkeit von der Modellkomplexität

Neben der Anpassungsrate wurde auch die Qualität der Anpassung verschiedener AAMs auf die Trainings- und Testdatenbank untersucht. Wie bei der Untersuchung der Anpassungsrate wurden auch hier verschiedene Modelle mit unterschiedlicher Anzahl von Form- und Grauwertparametern auf der Trainingsdatenbank erstellt und anschließend auf den Bildern des Testdatensatzes angepasst. Als Referenz diente auch hier die Anpassung der Modelle auf die Bilder der Trainingsdatenbank.

Abbildung 4.25 zeigt den mittleren Form- und Grauwertfehler auf Trainings- und Testdatenbank in Abhängigkeit von der Modellkomplexität.

Die durchgeführten Experimente haben gezeigt, dass sich ein Gesicht umso genauer synthe-

tisieren lässt, umso höher die Anzahl an Parametern sowohl für das Formmodell als auch für das Grauwertmodell ist. Durch die hohe Anzahl an freien Parametern können auch kleine Details in den Bilder synthetisiert werden.
Dies steht jedoch in Konflikt mit der Anpassungsrate, die ihr Optimum bei einer möglichst geringen Anzahl an Parametern besitzt (siehe vorheriger Abschnitt). Diese gegensätzlichen Anforderungen an die Parametrisierung zeigen sich im Verhalten der Synthesequalität. Die Güte der Anpassung der Form steigt anfänglich wie erwartet mit der Parameterzahl. Ab etwa sechs Formparametern beginnt der Fehler aber wieder zu steigen.
Ein weiterer Grund für die Verschlechterung der Schätzung liegt auch hier im konkurrierenden Verhalten einiger Eigenvektoren, was in ein lokales Minimum führt. Die Güte bezüglich der Grauwerte steigt wie erwartet mit der Anzahl der Grauwertparameter.

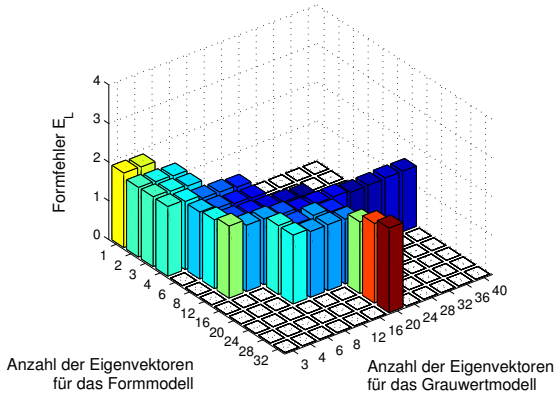
#### Anpassungsgüte in Abhängigkeit von der Kopfpose

Bei den durchgeführten Experimenten zeigte sich, dass die Anpassung am besten auf frontal aufgenommenen Gesichtern vorgenommen werden kann. Bei der Untersuchung wurde der genutzte Winkelbereich eingeschränkt auf Winkelauslenkungen von $\pm$ 25° in horizontaler und vertikaler Richtung. Dieser Winkelbereich ist für die geplante Verwendung zur Schätzung der Kopfpose zur Bestimmung des Interaktionsinteresses gut ausreichend. Mit diesem Erfassungsbereich kann das typische Verhaltens eines Benutzers, der sich gerade in Interaktion mit einem mobilen Robotersystem befindet, gut abgedeckt werden.
Jede Richtung wurden in 5 Winkelklassen aufgeteilt. Für jede der entstandenen 25 Klassen wurde die Anzahl der korrekt angepassten Bilder (auf Basis des Formfehlers $E_L$) bestimmt. Dazu wurde im Rahmen der durchgeführten Tests auf jedem Testbild zunächst eine Initialisierung mit Hilfe eines Gesichtsdetektors und anschließend eine Anpassung des Modells mit dem *Project-Out-Algorithmus* vorgenommen. Das eingesetzte Modell enthielt drei Form- und vier Grauwertkomponenten. Die Ergebnisse sind in Tabelle 4.1 zusammengefasst.

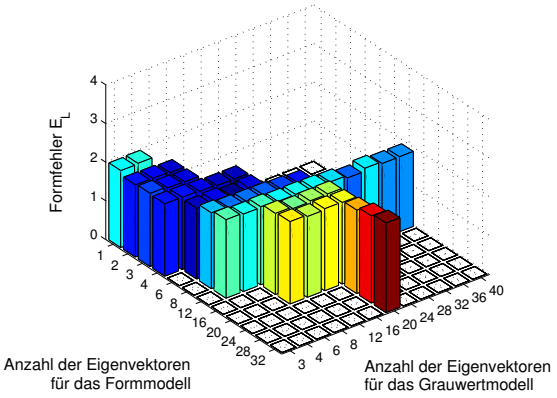
| Winkel | horizontal | | | | |
|---|---|---|---|---|---|
| vertikal | -25° ... -15° | -15° ... -5° | -5° ... 5° | 5° ... 15° | 15° ... 25° |
| -25° ... -15° | 50 | 57 | 61 | 46 | 43 |
| -15° ... -5° | 36 | 71 | 75 | 64 | 39 |
| -5° ... 5° | 54 | 68 | 93 | 64 | 46 |
| 5° ... 15° | 36 | 61 | 79 | 61 | 32 |
| 15° ... 25° | 18 | 32 | 50 | 36 | 25 |

***Tabelle 4.1:*** *Anpassungsrate in Abhängigkeit von der Kopfpose in Prozent: Frontal ausgerichtete Gesichter können am besten angepasst werden. Mit steigender Auslenkung in horizontaler oder vertikaler Richtung nimmt die Anpassungsfähigkeit des Modell bei Einzelbildversuchen ab.*

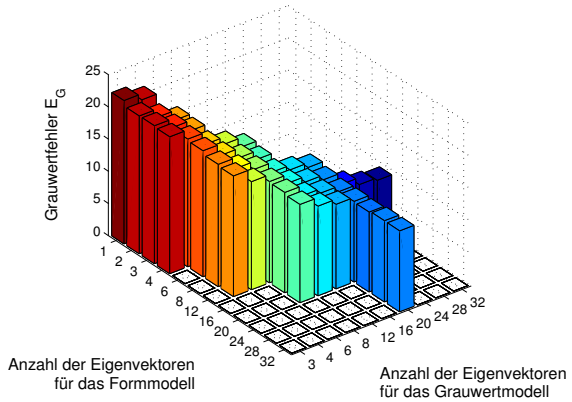
Betrachtet man den prozentualen Anteil der korrekt gefitteten Modelle, so zeigt sich deutlich,



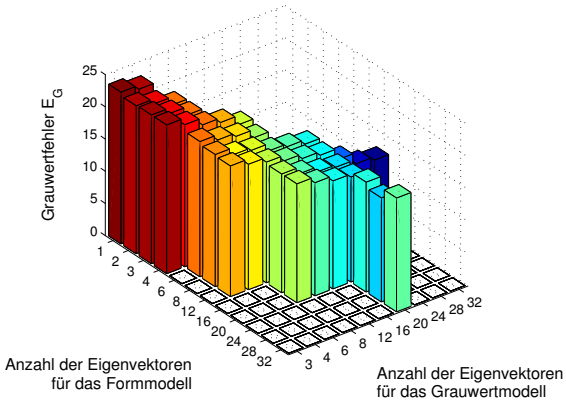

Formfehler $E_L$ Trainingsdatenbank




Formfehler $E_L$ Testdatenbank




Grauwertfehler $E_G$ Trainingsdatenbank




Grauwertfehler $E_G$ Testdatenbank

***Abbildung 4.25:*** *Die vier Grafiken zeigen den durchschnittlichen Formfehler $E_L$ und Grauwertfehler $E_G$ für verschiedene AAMs in Abhängigkeit von der Komplexität des Modells auf der Trainings- und Testdatenbank. Es ist zu erkennen, dass die Anzahl der genutzten Eigenvektoren Einfluss auf die Qualität der Anpassung hat. Bei einer zu großen Anzahl sinkt die Güte der Anpassung.*

dass der Anpassungsalgorithmus mit steigender Auslenkung zunehmend schlechter wird. Dies entspricht den Erwartungen, da die benötigten Anpassungen bei frontal ausgerichteten Gesichtern wesentlich geringer sind und somit die Wahrscheinlichkeit bei der Anpassung in ein lokales Minimum zu geraten, weniger hoch ist als bei starken Auslenkungen des Kopfes.

#### Verbesserung bei der Anpassung auf Bildsequenzen

In den vorangegangen Abschnitten wurde die Güte und Anpassungsfähigkeit der erstellten Active Appearance Modelle, jeweils auf ein Einzelbild bezogen, betrachtet. Die so erreichten Ergebnisse und Aussagen sind im Hinblick auf die Zielstellung der kontinuierlichen Bestimmung der Kopfpose des Benutzers teilweise kritisch zu werten. Bei einer Einzelbildbetrachtung ist für jedes Bild eine neue Initialisierung (z.B. auf Basis der Gesichtsdetektion) und eine Reihe von Iterationen notwendig, um ein Modell gut an das gegebene Bild anzupassen. Bei der fortlaufenden Anpassung eines AAMs auf Bilder einer Videosequenz liegen grundsätzlich andere Rahmenbedingungen vor: Das Modell muss nicht auf jedem Bild neu initialisiert, sondern kann mit Hilfe der auf dem vorhergehenden Bild geschätzten Parameter initialisiert werden. Dadurch wird erreicht, dass typischerweise nur geringere Parameteränderungen durchgeführt werden müssen und somit das Risiko, in ein lokales Minima zu gelangen, verringert wird. Neben der verbesserten Anpassung, führt diese Art der Initialisierung auch zu einer Verringerung der benötigten Iterationsschritte und somit zu einem geringeren Rechenaufwand.

Da keine umfangreichen Labeldaten für komplette Bildsequenzen existieren, konnten diese Untersuchungen nicht automatisiert werden. Einzelne herausgegriffene Beispiele zeigen aber, dass sich die Anpassung auf Kopfposen mit größeren Auslenkungen verbessert, wenn sie als Teil einer Sequenz betrachtet werden. Abbildung 4.26 zeigt die Schätzung der Formparameter und die zugehörigen Grauwertfehler $E_G$ (siehe (4.21)) auf Basis von Einzelinitialisierungen und innerhalb einer Sequenz. Es ist deutlich zu sehen, dass eine Initialisierung auf Basis der Schätzung des letzten Bildes zu einer deutlich besseren Anpassung führt, als die Modellanpassung auf Einzelbildern.

### 4.5.3 Relevante Parameter für die Kopfpose

Während der Erstellung der Active Appearance Modells hat sich gezeigt, dass die Kopfpose mit Hilfe der beiden ersten Formparameter beschrieben werden kann. Der Grund hierfür ist die PCA im Rahmen der Modellerstellung: Dabei werden die Eigenvektoren nach Größe der zugehörigen Eigenwerte sortiert. Somit repräsentieren die ersten (größten) Eigenvektoren die Komponenten, die die größte Varianz in den Daten verursachen. Die beiden größten Varianzen in den Daten stellen in der erstellten Datenbank die horizontale und die vertikale

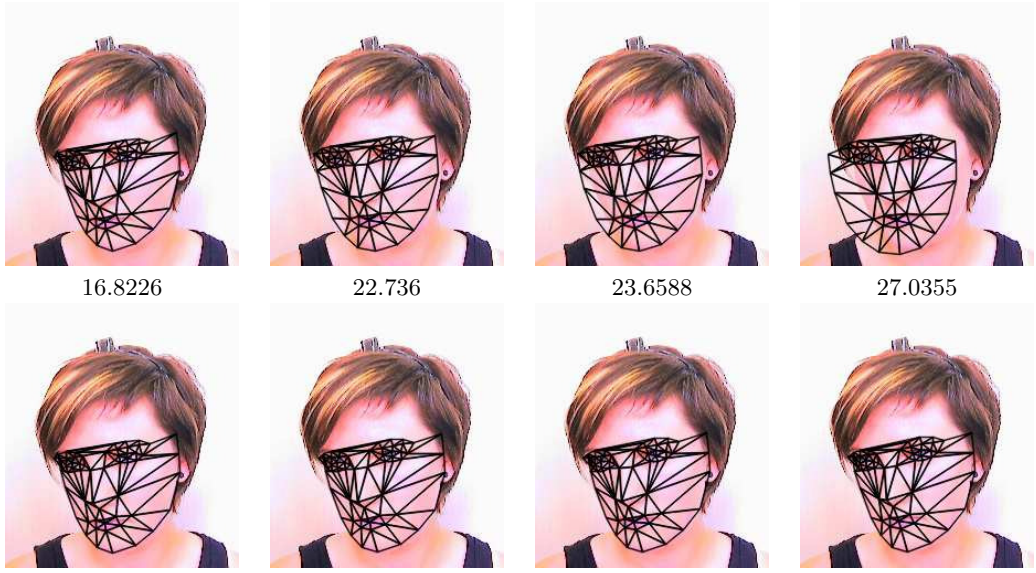



***Abbildung 4.26:*** *Die erste Reihe zeigt die erreichte Anpassung der Form und die zugehörigen Grauwertfehler $E_G$ für Einzelbilder einer Sequenz. Die zweite Reihe zeigt die Entwicklung der Form und die Grauwertdifferenz $E_G$ unter der Voraussetzung, dass für jedes Bild die auf dem jeweils vorhergehenden Bild berechneten Parameter zur Initialisierung genutzt werden.*

Auslenkung des Kopfes dar. Somit wurden diese beiden Parameter $p_0$ und $p_1$ im Hinblick auf ihre Aussagekraft bezüglich der Kopfpose untersucht.

Abbildung 4.27 zeigt den Zusammenhang zwischen den beiden Parametern $p_0$ und $p_1$ und den gemessenen Winkeln in horizontaler bzw. vertikaler Richtung. Hierbei wird ersichtlich, dass bei beiden Parametern ein quasi linearer Zusammenhang zwischen dem Parameterwert und dem Winkel existiert.

Die Abbildungen zeigen aber auch eine große Streuung der Winkel bezüglich der Parameter. Dies wird teilweise durch Fehler bei der Aufnahme hervorgerufen. Zum einen ist das Magnetfeld, welches für die Messung mit dem *Flock of Bird* erzeugt wird, Umwelteinflüssen ausgeliefert. Zum anderen konnte das Signal zwischen der Kamera und dem *Flock of Bird* nicht vollständig synchronisiert werden, das heißt, die Videoaufnahmen entstanden zeitlich versetzt. Der durchschnittliche Zeitversatz zwischen Videobild und Sensordaten lag zwischen $300 - 600ms$. Durch die Kopfbewegung der Probanden bei der Datenaufnahme sind somit Fehler entstanden.

Zur Überprüfung der Annahme, dass die Parameter $p_0$ und $p_1$ zur Beschreibung der vertikalen und horizontalen Kopfauslenkung verwendet werden können, wurde im Rahmen dieser Arbeit die *Mutual Information for Feature Selection (MIFS)* benutzt. Bei der MIFS handelt es sich um ein Verfahren aus der Informationstheorie zur Bestimmung der Relevanz von Pa-

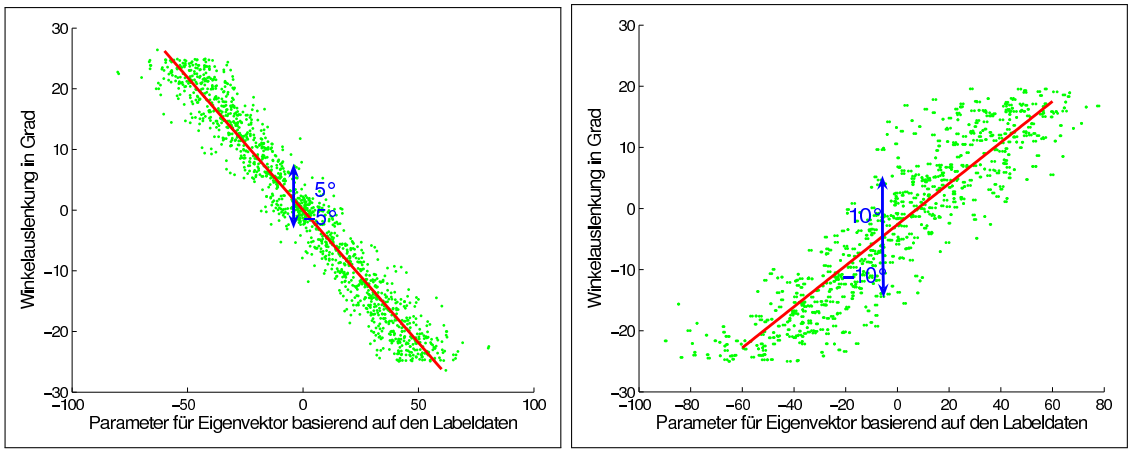



***Abbildung 4.27:*** *Verteilung der gemessenen Winkel bezüglich der adaptierten Parameter für die vertikale Kopfauslenkung (1) und die horizontale Auslenkung (2).*

rametern eines Klassifikations- oder Funktionsapproximationsproblems. Weitere Details zur MIFS finden sich im Anhang D.1. Für die Relevanz der Modellparameter in Bezug auf Pan- und Tilt-Winkel wurden folgende MIFS-Werte bestimmt:

| Pan | | | Tilt | |
|---|---|---|---|---|
| Parameter | MIFS | | Parameter | MIFS |
| $p_1$ | 1.644517 | | $p_0$ | 1.174072 |
| $p_2$ | 0.183370 | | $p_2$ | 0.143701 |
| $p_3$ | 0.016361 | | $p_3$ | 0.086005 |

***Tabelle 4.2:*** *Werte der MIFS der einzelnen Modellparameter $p_i$ (Formparameter) für Pan- und Tilt-Winkel. Nicht aufgelistete Parameter haben weniger Relevanz/Signifikanz als ein künstlich hinzugefügter Rauschkanal.*

Aus den MIFS-Werten in Tabelle 4.2 ist ersichtlich, dass die Parameter $p_0$ bzw. $p_1$ um fast eine Größenordnung gegenüber den anderen Parametern hervortreten. Damit konnte auch quantitativ nachgewiesen werden, dass eine Verwendung von lediglich $p_0$ und $p_1$ zur Schätzung der Kopfpose legitim ist. Die restlichen Modell-Parameter des AAMs spielen hierfür faktisch keine Rolle.

### 4.5.4 Auswertung auf Einzelbildern

Zur Schätzung der Kopfpose basierend auf den Formparametern des Active Appearance Models wurden auch hier zwei Varianten untersucht: Eine lineare Approximation und ein Multi-Layer-Perceptron.

### Kopfposenschätzung mittels linearer Approximation

Im Rahmen der Kopfposenschätzung mit Hilfe einer linearen Approximation wird eine lineare Abhängigkeit zwischen den Parametern $p_0$ bzw. $p_1$ und dem Winkel der horizontalen bzw. vertikalen Kopfpose angenommen. Die Arbeit von [Lanitis et al., 1997] und die Verteilung der gemessenen Winkel bezüglich der adaptierten Parameter auf dem Trainings- und Testdatensatz (siehe Abbildung 4.27) unterstützt diese Annahme.

Zur Approximation wurde die *gnuplot*[5] Implementierung des *non-linear least-squares (NLLS) Levenberg-Marquardt* Algorithmus eingesetzt. Der *Levenberg-Marquardt-Algorithmus* [Levenberg, 1944, Marquardt, 1963] ist ein numerischer Optimierungsalgorithmus zur Lösung nicht-linearer Probleme auf Basis der *Methode der kleinsten Quadrate*. Bei dem Algorithmus handelt es sich um eine Erweiterung des *Gauß-Newton-Algorithmus*. Im Allgemeinen ist der Levenberg-Marquardt-Algorithmus robuster. Wie beim Gauß-Newton-Verfahren ist aber auch hier eine Konvergenz nicht garantiert.

Als Ziel der Schätzung der horizontalen und vertikalen Kopfpose wurden also zwei lineare Funktionen folgender Form gesucht:

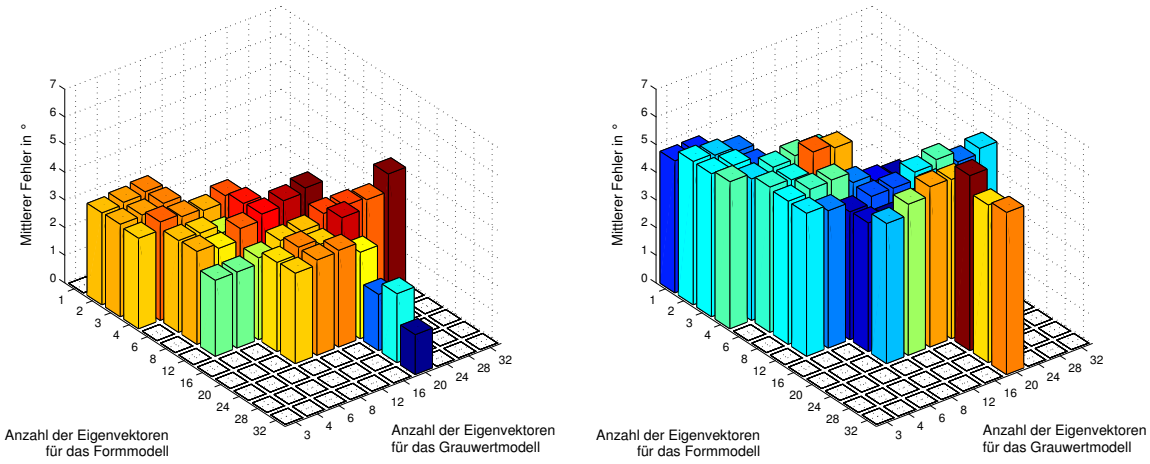
$$\begin{array}{rcl} \phi_{pan}(x) & = & m_{pan} \cdot x + n_{pan} \\ \phi_{tilt}(x) & = & m_{tilt} \cdot x + n_{tilt} \end{array} ,$$

wobei als freie Variable $x$ direkt die Modellparameter $p_0$ bzw. $p_1$ eingesetzt werden sollen. Der gesuchte Winkel kann somit direkt als $\phi(x)$ abgelesen werden. Insofern der zugrunde liegende Trainingsdatensatz symmetrisch bezüglich der horizontalen Achse ist bzw. die Symmetrie durch Hinzufügen von horizontal gespiegelten Trainingsbildern erzeugt wird, wird das Mittelwertgesicht $s_0$ eine horizontale Auslegung von 0° haben. Hierdurch vereinfacht sich das Problem zusätzlich, da dann $n_{pan} = 0$ gilt.

Abbildung 4.28 zeigt die Fehler für die Schätzung der horizontalen und vertikalen Kopfpose in Abhängigkeit der Modellkomplexität. Mit dieser Schätzung konnte ein mittlerer horizontaler Winkelfehler von knapp 3° erreicht werden. Die vertikale Kopfauslenkung kann etwas schlechter mit einem mittleren Fehler von knapp 5° geschätzt werden.
Bei der Interpretation von Abbildung 4.28 ist darauf zu achten, dass hier für jede Modellkonfiguration nur die Datensätze in die Fehlerbestimmung eingehen, auf denen auch eine erfolgreiche Modellanpassung durchgeführt werden konnte! Dies hat folgende Konsequenzen:

- Für ein Modell mit sehr vielen freien Parametern können nur sehr wenige Bilder der Testdatenbank korrekt angepasst werden (vgl. auch Abbildung 4.24). Insofern auf diesen die geschätzten Winkel eine geringe Abweichung aufweisen, wird auch ein geringer

[5] Gnuplot Homepage `http://www.gnuplot.info/`




Horizontale Kopfposenschätzung Vertikale Kopfposenschätzung

***Abbildung 4.28:*** *Die beiden Diagramme zeigen die Fehler bei der Posenschätzung mittels linearer Funktionen für verschiedene AAMs in Abhängigkeit von der Komplexität des Modells. Die linke Grafik zeigt den Schätzfehler bei der horizontalen Auslenkung. Die rechte Grafik den Fehler für die vertikale Kopfpose. Es ist zu sehen, dass der Schätzfehler bei zunehmender Komplexität ansteigt.*

mittlerer Fehler als Ergebnis entstehen. Dies darf jedoch nicht darüber hinwegtäuschen, dass das Modell auf einem Großteil der Bilder nicht angepasst und somit auch keine Winkelschätzung vorgenommen werden konnte und somit eine solche Konfiguration auch nicht für eine Kopfposenschätzung geeignet ist.

- Es ist ersichtlich, dass der Fehler bei zunehmender Komplexität auch leicht ansteigt, aber nicht deutlich größer wird. Dies kann mit der Tatsache begründet werden, dass Bilder mit einer schlechten Anpassung (auf denen dann auch nur eine schlechtere Winkelschätzung möglich wäre) nicht mit in die Auswertung eingehen. Oder umgekehrt ausgedrückt: Eine hinreichend genaue Modellanpassung führt automatisch zu einem beschränkten Winkelfehler.

Unter der Randbedingung, dass ein Modell hinreichend genau an ein gegebenes Bild angepasst werden kann, liegen beide Fehler insgesamt im Bereich der Streuung der Winkel, die bei der Datenaufnahme mit Hilfe des *Flock of Bird* entstanden sind.

### Kopfposenschätzung mittels MLP

Bei der Schätzung der Kopfpose handelt es sich um ein Funktionsapproximationsproblem, dass auch mittels einem *Multi Layer Perceptron (MLP)* gelöst werden kann. Details zum

MLPs finden sich im Anhang A.2.

Zunächst wurde untersucht, wie sich die verwendete Netzwerkarchitektur auf den resultierenden mittleren Fehler auswirkt. Da das zugrunde liegende Problem fast als linear betrachtet werden kann (siehe Abschnitt 4.5.3), zeigte sich schnell, dass bereits ein einfaches Perceptron in der Lage ist, die Funktionsapproximation mit geringem Fehler vorzunehmen. Mehrschichtige Netzwerke lieferten vergleichbare Ergebnisse, wobei die Generalisierung hier schlechter war, da diese beim Training schnell zum *over-fitting* tendierten. Da die Komplexität des Problems offenbar kein mehrschichtiges Netzwerk erfordert, wurden die weiteren Untersuchungen nur mit einem einschichtigen Perceptron durchgeführt.

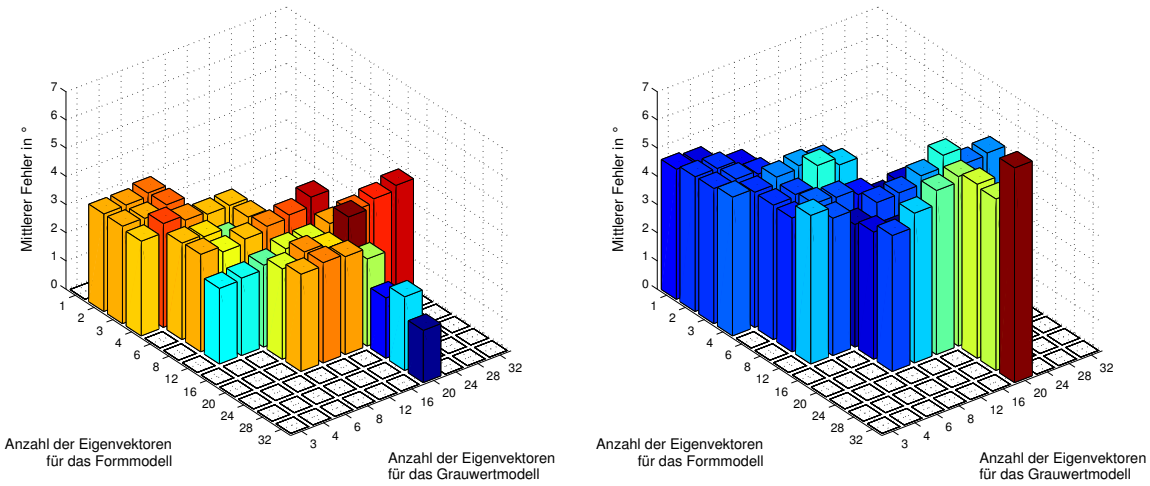
Zur Schätzung der Kopfpose mittels Perceptron wurden weiterhin zwei Varianten untersucht: Einerseits wurde der gesamte Vektor der Formparameter $\mathbf{p}$ genutzt und andererseits nur der Parameter $p_0$ bzw. $p_1$ als Input verwendet. Tabelle 4.3 zeigt die erzielten Ergebnisse. Hierbei ist ersichtlich, dass die Verwendung des vollständigen Formparametervektors $\mathbf{p}$ zu einem größeren mittleren Fehler führt. Dies kann damit begründet werden, dass quasi nur die Parameter $p_0$ bzw. $p_1$ überhaupt relevante Informationen zur Kopfpose beinhalten. Die restlichen Formparameter führen zu einem Fehlergebirge mit mehr lokalen Minima und somit zu einem tendenziell schlechteren Ergebnis des Trainings. Die Ergebnisse bei

| | Vektor $\mathbf{p}$ | $p_0$ bzw. $p_1$ |
|---|---|---|
| mittlerer Fehler horizontal | 5.0° | 3.1° |
| mittlerer Fehler vertikal | 6.8° | 5.3° |

***Tabelle 4.3:*** *Die Tabelle zeigt den durchschnittlichen Fehler bei der Kopfposenschätzung in horizontaler und vertikaler Richtung für ein einfaches Perceptron in Abhängigkeit des verwendeten Parametersatzes.*

der Verwendung allein von $p_0$ bzw. $p_1$ sind vergleichbar mit den Ergebnissen der linearen Approximation (siehe Abschnitt 4.5.4). Diese Tatsache ist damit erklärbar, dass ein einfaches Perceptron letztendlich die gleichen mathematischen Operationen wie eine lineare Funktionsapproximation durchführt.

Weiterhin wurde untersucht, wie sich die Komplexität des AAMs auf die Schätzung auswirkt. Abbildung 4.29 zeigt die erzielten Ergebnisse. Diese sind im Wesentlichen vergleichbar mit den Ergebnissen aus Abschnitt 4.5.4. Der mittlere Fehler für die horizontale Kopfpose liegt bei ca. 3° und für die vertikale Kopfauslenkung bei ca. 5°. Bei steigender Komplexität des AAMs nehmen beide Fehler zu. Die Erklärung für dieses Verhalten ist identisch mit der aus dem vorherigen Abschnitt.




Horizontale Kopfposenschätzung Vertikale Kopfposenschätzung

***Abbildung 4.29:*** *Die beiden Diagramme zeigen die Fehler bei der Posenschätzung mittels MLP für verschiedene AAMs in Abhängigkeit von der Komplexität des Modells. Die linke Grafik zeigt den Schätzfehler bei der horizontalen Auslenkung. Die rechte Grafik den Fehler für die vertikale Kopfpose. Es ist zu sehen, dass der Schätzfehler bei zunehmender Komplexität ansteigt.*

#### Fazit Kopfposenschätzung

Die beiden vorangegangenen Abschnitte haben gezeigt, dass eine gute Schätzung der Kopfpose für Drehung und Heben-Senken mit einer einfachen linearen Approximation oder einem einfachen einschichtigen Perceptron realisiert werden kann. Der mittlere Fehler für die horizontale Kopfdrehung liegt bei etwa 3° und ist damit recht genau. Der mittlere Fehler für das Heben und Senken liegt mit 5° minimal höher. Im Vergleich zur Literatur (siehe Abschnitt 4.2.2) ist der mittlere Fehler damit kleiner als der der *appearance-based* Verfahren (ca. 10°) und etwa gleich mit dem der *model-based* Verfahren (3...5°).
Der erzielte Fehler liegt weiterhin in der Größenordnung des Messfehlers, der durch das gegebene Setup bestehend aus Kamera und *Flock of Bird* entstanden ist. Insofern Möglichkeiten für eine Datenaufnahme mit geringerem Messfehler gegeben sind, kann der Fehler der Kopfposenschätzung möglicherweise noch weiter reduziert werden.

### 4.5.5 Auswertung auf Videosequenzen

In den vorangegangenen Abschnitten wurde eine detaillierte Auswertung der erzielten Ergebnisse der Kopfposenschätzung auf Einzelbildern präsentiert. In der Praxis ist jedoch nur selten eine Schätzung auf Einzelbildern sinnvoll. Typischerweise muss hier stattdessen

die Kopfpose kontinuierlich aus einem (Live)-Videodatenstrom ermittelt werden. Wie in Abschnitt 4.5.2 erläutert wurde, ist die Schätzung auf einer Videosequenz etwas einfacher, da für die Initialisierung jeweils die Modellschätzung des vorherigen Zeitschritts verwendet werden kann (siehe auch Abbildung 4.26). Andererseits stellt die kontinuierliche Schätzung hohe Anforderungen an die Performance und Effizienz des Algorithmus. Nur wenn das Verfahren in der Lage ist, die Anpassung schnell und hinreichend genau durchzuführen, ist letztendlich eine Kopfposenschätzung in Echtzeit möglich.

Im Rahmen dieser Dissertation wurde die Kopfposenschätzung auf Videosequenzen auf der in Abschnitt 4.5.1 vorgestellten Datenbank, einer weiteren Datenbank mit *Ground-Truth* Informationen und weiterhin auf verschiedenen freien Videosequenzen getestet.

#### Auswertung auf Videosequenzen mit Ground-Truth Daten

Als erste Untersuchung wurde die Kopfposenschätzung auf der in Abschnitt 4.5.1 vorgestellten Datenbank durchgeführt. Dabei wurden als Ground-Truth-Daten die ermittelte Kopfpose des *Flock of Bird* Sensors verwendet.

Weiterhin wurden Experimente auf Daten der *Boston University Face Tracking Database* [BU Database, 1998] durchgeführt. Dabei handelt es sich um Videosequenzen von 8 Probanden, die den Kopf vor der Kamera frei bewegen konnten. Abbildung 4.30 zeigt vier Beispielbilder. Je Proband wurden 9 Sequenzen unter verschiedenen Randbedingungen (Beleuchtungsbedingungen und Art der Kopfbewegung) aufgenommen. Bei 5 Probanden wurde eine konstante gleichmäßige Beleuchtung verwendet (*uniform-light*) und bei 3 Probanden eine sich verändernde Beleuchtung (*varying-light*). Bei allen Tests wurde die Kopfpose ebenfalls mit einem *Flock of Bird* Sensors aufgezeichnet und steht somit als *Ground-Truth*-Information zur Verfügung.

***Abbildung 4.30:*** *Beispielbilder der Videosequenzen aus der Boston University Face Tracking Database [BU Database, 1998].*

Zur Auswertung der Sequenzen wurde für jedes Bild zunächst eine Modellanpassung durchgeführt. Insofern aus dem vorherigen Zeitschritt $t-1$ bereits eine gültige Schätzung

vorhanden war, wurden die Modellparameter dieser Schätzung als Startwerte für die Anpassung zum Zeitpunkt $t$ verwendet. Wenn keine gültige Schätzung vorlag, wurde der *Viola-Jones-Gesichtsdetektor* zur Initialisierung des Modells eingesetzt. Wenn kein gültiges Modell gefunden werden konnte oder kein Gesicht zwecks Initialisierung gefunden werden konnte, wurde keine Kopfposenschätzung vorgenommen.

Die Tests wurden mit einem AAM-Modell der Größe 70x70 Pixel und dem *Project-Out-Algorithmus* durchgeführt. Als Vorverarbeitung des Inputs wurden ein Glättungsoperator (Gauß-Filter der Größe 3x3) und ein Histogrammausgleich eingesetzt.

Für jeden Zeitschritt, in dem eine gültige Modellanpassung vorgenommen werden konnte, wurde anschließend die Kopfpose basierend auf der in Abschnitt 4.5.4 vorgestellten linearen Approximation ermittelt. Durchgeführte Experimente mit einer Winkelschätzung basierend auf einem einschichtigen MLP zeigten die gleichen Ergebnisse.

Die Abbildungen 4.31 und 4.32 zeigen beispielhaft die Ergebnisse aus zwei von 28 vorliegenden Sequenzen aus Abschnitt 4.5.1. Es sind jeweils der zeitliche Verlauf der horizontalen (*pan*) und vertikalen (*tilt*) Kopfposenschätzung dargestellt. Zusätzlich sind jeweils drei ausgewählte Frames mit dem angepassten Modell und die beiden Histogramme der Winkelfehler im Bereich bis maximal 30° dargestellt.

Im Hinblick einer Bewertung der Ergebnisse ist zu beachten, dass die im vorangegangenen Abschnitt durchgeführten Auswertungen und Betrachtungen lediglich für Winkel von ± 25° in horizontaler und vertikaler Richtung durchgeführt wurden. In den hier vorliegenden Sequenzen treten aber auch Kopfposen mit deutlich größeren Auslenkungen im Bereich von ± 45° auf. Bei großen Winkeln treten bereits (Selbst-)Verdeckungen auf, und eine Anpassung des Modells in diesen Randbereichen wird deutlich schwieriger (siehe Abschnitt 4.4.3). In beiden Beispielen ist zu erkennen, dass vor allem bei großen Auslenkungen oftmals keine korrekte Modellanpassung mehr vorgenommen werden kann und somit auch keine Winkelschätzung möglich ist.

In beiden Beispielen ist auch zu sehen, dass größere Schätzfehler eher bei großen Winkeln auftreten. Im Bereich von ± 25° ist die Schätzung relativ passend. Je größer die Winkel, desto größer sind auch Ausreißer und Sprünge. Es ist auch zu erkennen, dass bereits unmittelbar vor dem Verlust der Modellschätzung größere Winkelfehler auftreten. Dies kann damit begründet werden, dass bereits zu diesen Zeitpunkten, die Modellanpassung recht ungenau ist und daher auch keine gute Winkelschätzung mehr vorgenommen werden kann.

Im gesamten zeitlichen Verlauf der Beispielsequenzen ist zu sehen, dass eine Winkelschätzung gut vorgenommen werden kann, wenn das Modell gut an das Inputbild angepasst

ist. Die beiden geschätzten Winkel entsprechen gut dem realen Verlauf, der mittels *Flock of Bird* Sensor aufgenommen wurde. Bei der Bewertung ist zu beachten, dass auch die Sensordaten nicht fehlerfrei sind. Auf Grund von Störungen des Magnetfeldes und einer nicht einwandfreien Synchronisation zwischen Sensor und PC ist hier mit Fehlern bis zu $3 - 5°$ zu rechnen. Bei schnellen Kopfbewegungen kann dieser möglicherweise noch höher liegen.

Die Abbildungen 4.33 und 4.34 zeigen Ergebnisse zweier Sequenzen mit konstanter gleichmäßiger Beleuchtung der *Boston University Face Tracking Database* [BU Database, 1998]. Die Sequenzen mit veränderlicher Beleuchtung zeigen einen deutlich schlechteren Verlauf. Grund hierfür sind Probleme bei der Modellanpassung in sehr dunklen oder überstrahlten Bildregionen. Zur Lösung dieses Problems ist eine verbesserte Vorverarbeitung und/oder eine Verbesserung des AAM-Anpassungsalgorithmus notwendig. Da dies jedoch kein Schwerpunkt dieser Dissertation ist, wurden die Sequenzen mit veränderlicher Beleuchtung hier nicht weiter betrachtet.

Anhand dieser beiden Beispiele ist trotzdem zu sehen, dass das im Rahmen dieser Dissertation vorgestellte System auch auf vollständig unbekannten Daten noch zufriedenstellend funktioniert. Der mittlere Fehler ist bei diesen Sequenzen höher als auf den bekannten Daten. Die Ursache hierfür sind hauptsächlich Probleme bei der Anpassung des Modells an die unbekannten Daten. Bei zwei der Probanden konnte auf Grund der ethnischen Herkunft keine stabile Anpassung erreicht werden, da die Besonderheiten ihres Gesichts (z.B. Kopfform, Augenabstand, Nasenform, ...) nicht durch das verwendete Active Appearance Model abgedeckt werden konnten. Mit Hilfe eines allgemeineren Modells und einer optimierten Anpassung sollte dieses Problem jedoch beseitigt werden können. Trotz dieser Einschränkung kann die Kopfpose auf den unbekannten Daten erfolgreich geschätzt werden.

Insgesamt konnten bei den drei verbleibenden Probanden der *uniform-light*-Auswahl die jeweils 9 Sequenzen erfolgreich getestet werden. Die erzielten Resultate sind vergleichbar mit denen der Beispiele aus den beiden Abbildungen 4.33 und 4.34.

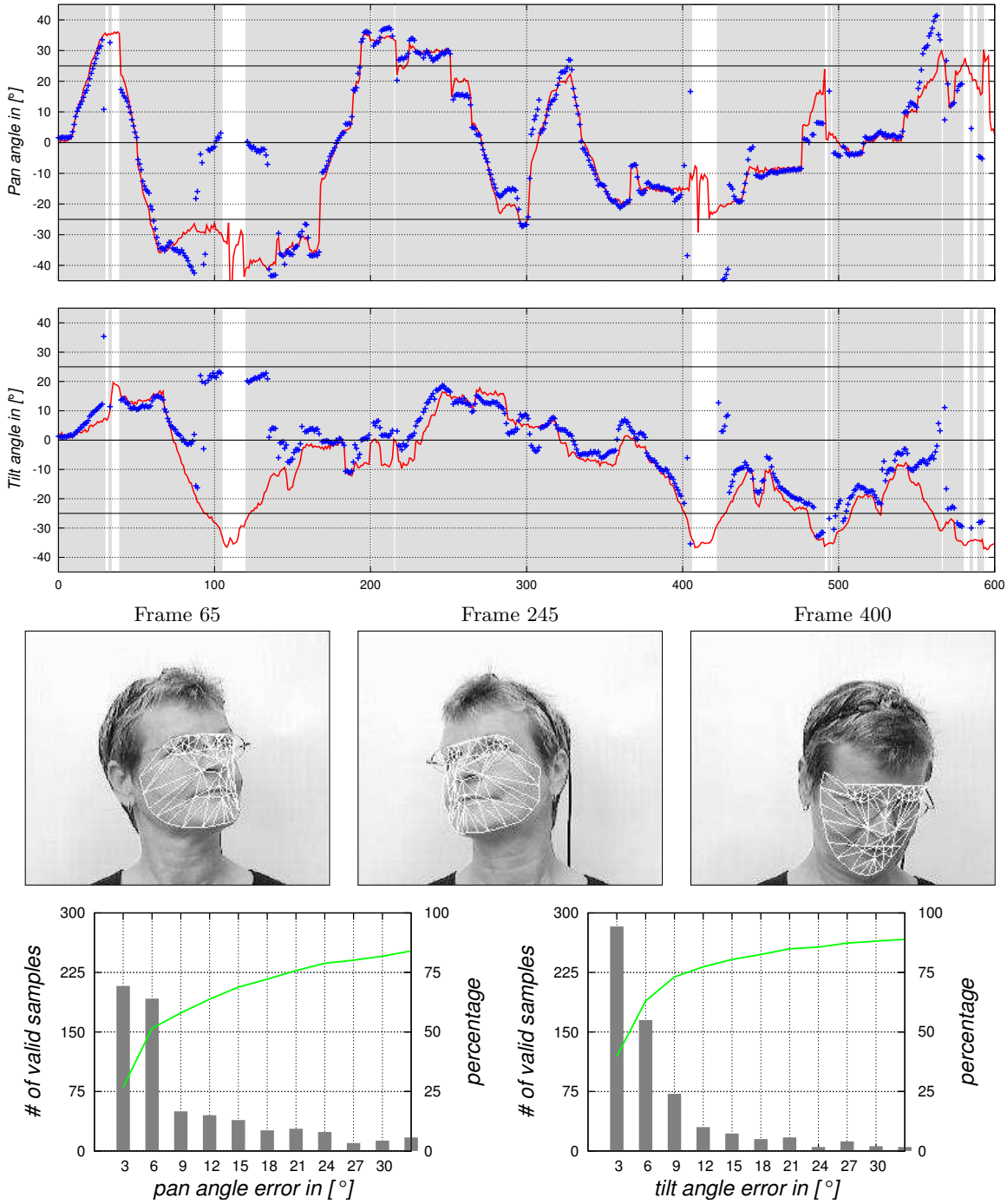



***Abbildung 4.31:*** *Ergebnisse auf Sequenz "Seq025" (bekannte weibliche Person aus Trainingsdatenbank): Die rote Linie zeigt den mit dem Flock of Bird Sensor aufgezeichneten Verlauf der horizontalen und vertikalen Kopfpose. In grau hinterlegten Bereichen konnte das Modell erfolgreich angepasst werden. Die Winkelschätzung ist blau dargestellt. Im unteren Teil der Grafik sind die Histogramme des Winkelfehlers als Anzahl pro Klasse (graue Balken) und in Prozent (grüne Linie) abgebildet.*

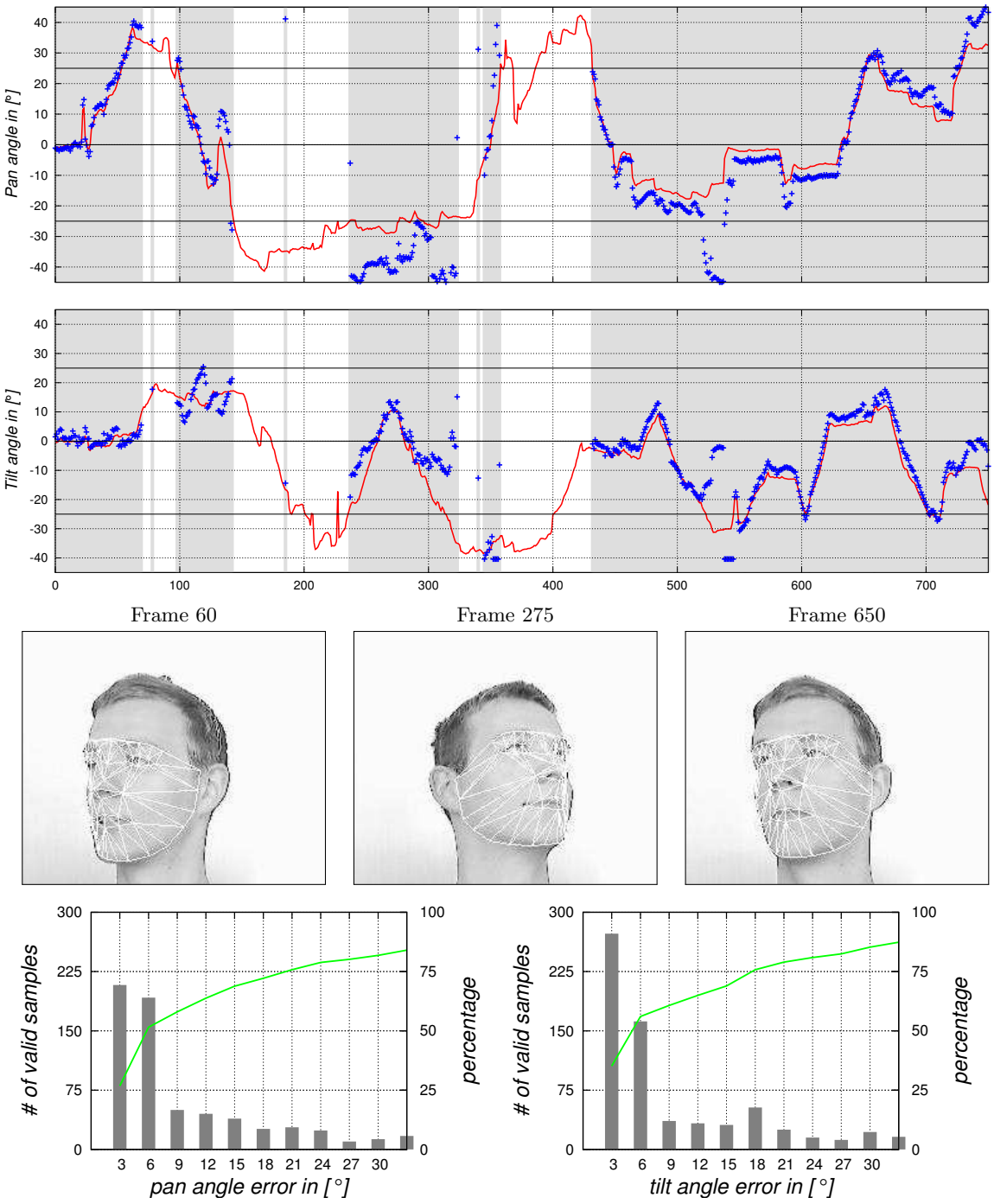



***Abbildung 4.32:*** *Ergebnisse auf Sequenz "Seq002" (bekannte männliche Person aus Trainingsdatenbank): Die rote Linie zeigt den mit dem Flock of Bird Sensor aufgezeichneten Verlauf der horizontalen und vertikalen Kopfpose. In grau hinterlegten Bereichen konnte das Modell erfolgreich angepasst werden. Die Winkelschätzung ist blau dargestellt. Im unteren Teil der Grafik sind die Histogramme des Winkelfehlers als Anzahl pro Klasse (graue Balken) und in Prozent (grüne Linie) abgebildet.*

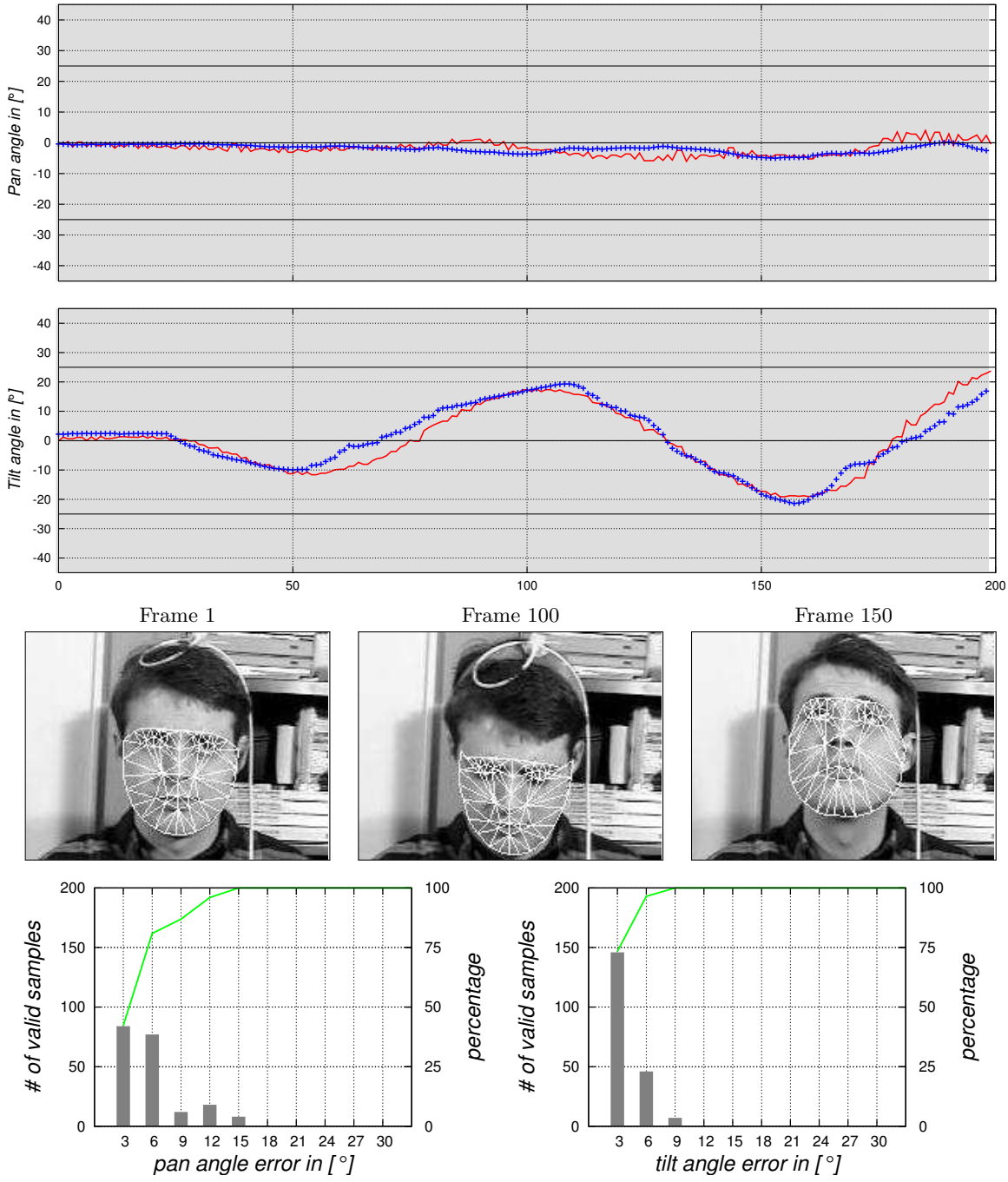



***Abbildung 4.33:*** *Ergebnisse auf Sequenz "BU-jim4" (Sequenz mit vorrangig Nickbewegung des Kopfes): Die rote Linie zeigt den mit dem Flock of Bird Sensor aufgezeichneten Verlauf der horizontalen und vertikalen Kopfpose. In grau hinterlegten Bereichen konnte das Modell erfolgreich angepasst werden. Die Winkelschätzung ist blau dargestellt. Im unteren Teil der Grafik sind die Histogramme des Winkelfehlers als Anzahl pro Klasse (graue Balken) und in Prozent (grüne Linie) abgebildet.*

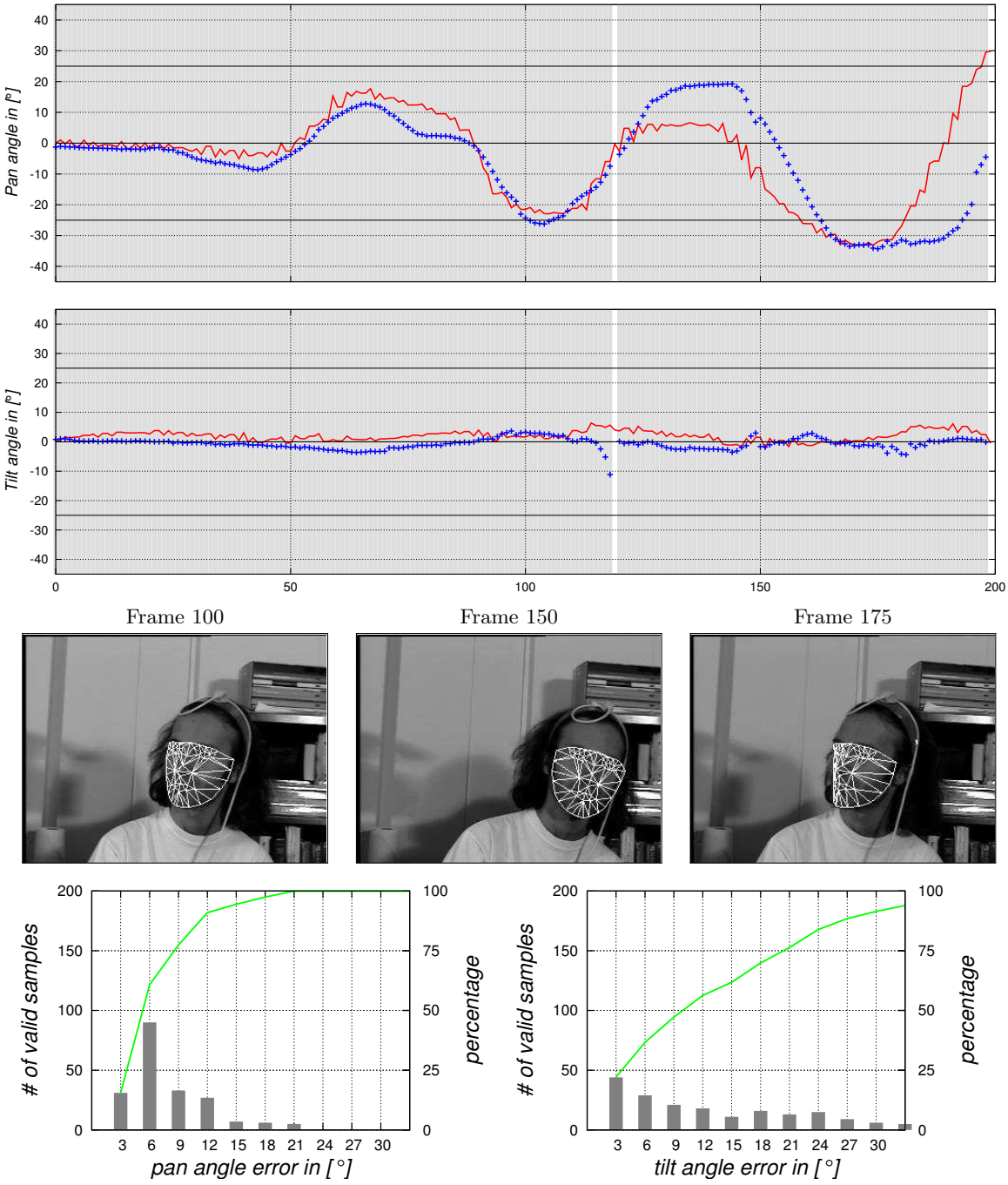



***Abbildung 4.34:*** *Ergebnisse auf Sequenz "BU-vam4" (Sequenz mit vorrangig horizontaler Drehbewegung des Kopf und Translation im Bild): Die rote Linie zeigt den mit dem Flock of Bird Sensor aufgezeichneten Verlauf der horizontalen und vertikalen Kopfpose. In grau hinterlegten Bereichen konnte das Modell erfolgreich angepasst werden. Die Winkelschätzung ist blau dargestellt. Im unteren Teil der Grafik sind die Histogramme des Winkelfehlers als Anzahl pro Klasse (graue Balken) und in Prozent (grüne Linie) abgebildet.*

### Auswertung auf freien Videosequenzen

Als eine weitere Untersuchung wurde das Verhalten der Kopfposenschätzung auf einer Reihe von verschiedenen freien Videosequenzen untersucht. Dabei wurden geeignete Videos aus dem Internet verwendet. Es wurden Sequenzen ausgewählt, in denen insbesondere der Kopf einer Person über einen längeren Zeitraum gut zu sehen ist und die Person den Kopf deutlich und eindeutig bewegt. Passend hierfür sind beispielsweise Aufnahmen von TV-Moderatoren oder von Nachrichtensprechern/-innen. Insgesamt wurden 10 Videosequenzen zwischen 30 und 60 Sekunden untersucht.

Da zu den Sequenzen keine *Ground-Truth-Daten* für die Kopfpose existieren, kann nur eine qualitative Einschätzung und Auswertung der Tests vorgenommen werden.

Die Abbildungen 4.35 und 4.36 zeigen zwei Ausschnitte je 40 Sekunden von zwei Nachrichtensprechern. Auf beiden Sequenzen konnte das AAM-Modell kontinuierlich korrekt auf die Bilder angepasst und eine Kopfposenschätzung vorgenommen werden.

In beiden Beispielen ist deutlich das regelmäßige Heben und Senken des Kopfes beim Ablesen der Nachrichten von der Vorlage zu erkennen. In der ersten Sequenz kann man auch sehen, dass der Sprecher seinen Kopf beim Absenken leicht nach rechts verdreht, da die Vorlage offenbar nicht mittig vor ihm liegt. Beim Anheben des Kopfes und dem Blick in die Kamera erreicht er wieder eine mittige Ausrichtung.

Abbildung 4.37 zeigt einen einminütigen Ausschnitt einer Diskussionsrunde. Obwohl die Köpfe der beiden Gesprächspartner relativ klein im Bild dargestellt sind (Kopfhöhe: 70 bis 100 Pixel), kann das AAM-Modell kontinuierlich korrekt angepasst werden. Auch beim Umschalten der Kameraposition, kann das Modell innerhalb eines Frames korrekt auf die andere Person angepasst werden. Bedingt durch die Positionen der beiden Kameras zeigen beide Gesprächspartner einen seitlich verdrehten Kopf. Dies wird auch korrekt durch die Kopfposenschätzung ermittelt. Die vertikale Kopfpose zeigt bei beiden Personen typische Nickbewegungen, die durch das Sprechen entstehen.

Diese Tests auf den freien Videosequenzen haben gezeigt, dass das realisierte Teilsystem auch mit unvollständig unbekannten Daten umgehen kann und eine Kopfposenschätzung realisiert werden kann. Eine wichtige Grundvoraussetzung ist jedoch, dass das zugrunde liegende Modell im Hinblick auf die repräsentierbaren Gesichter universell genug ist. Ohne eine korrekte Modellanpassung ist auch keine Kopfposenschätzung möglich.

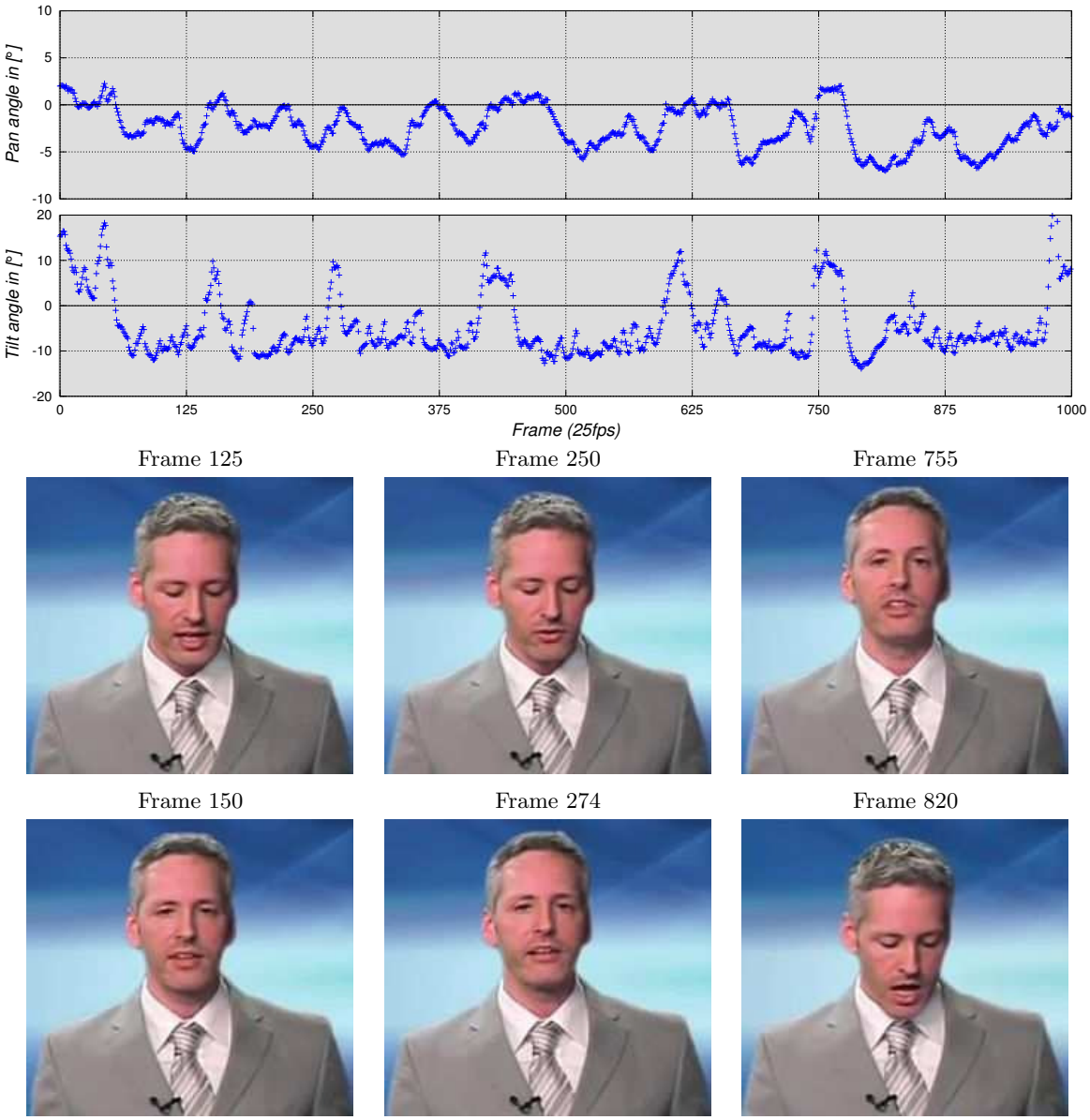



***Abbildung 4.35:*** *Ergebnisse auf Sequenz "Nachrichten 1": Die Frames 125, 250 und 820 zeigen ein deutliches Absenken des Kopfes verbunden mit einer horizontalen Verdrehung nach rechts. In den Frames 150, 274 und 755 schaut der Sprecher fast direkt ins Bild. Der zeitliche Verlauf der vertikalen Kopfpose zeigt die für einen Nachrichtensprecher typischen Nickbewegungen, die durch das Ablesen der Nachrichten bedingt sind.*

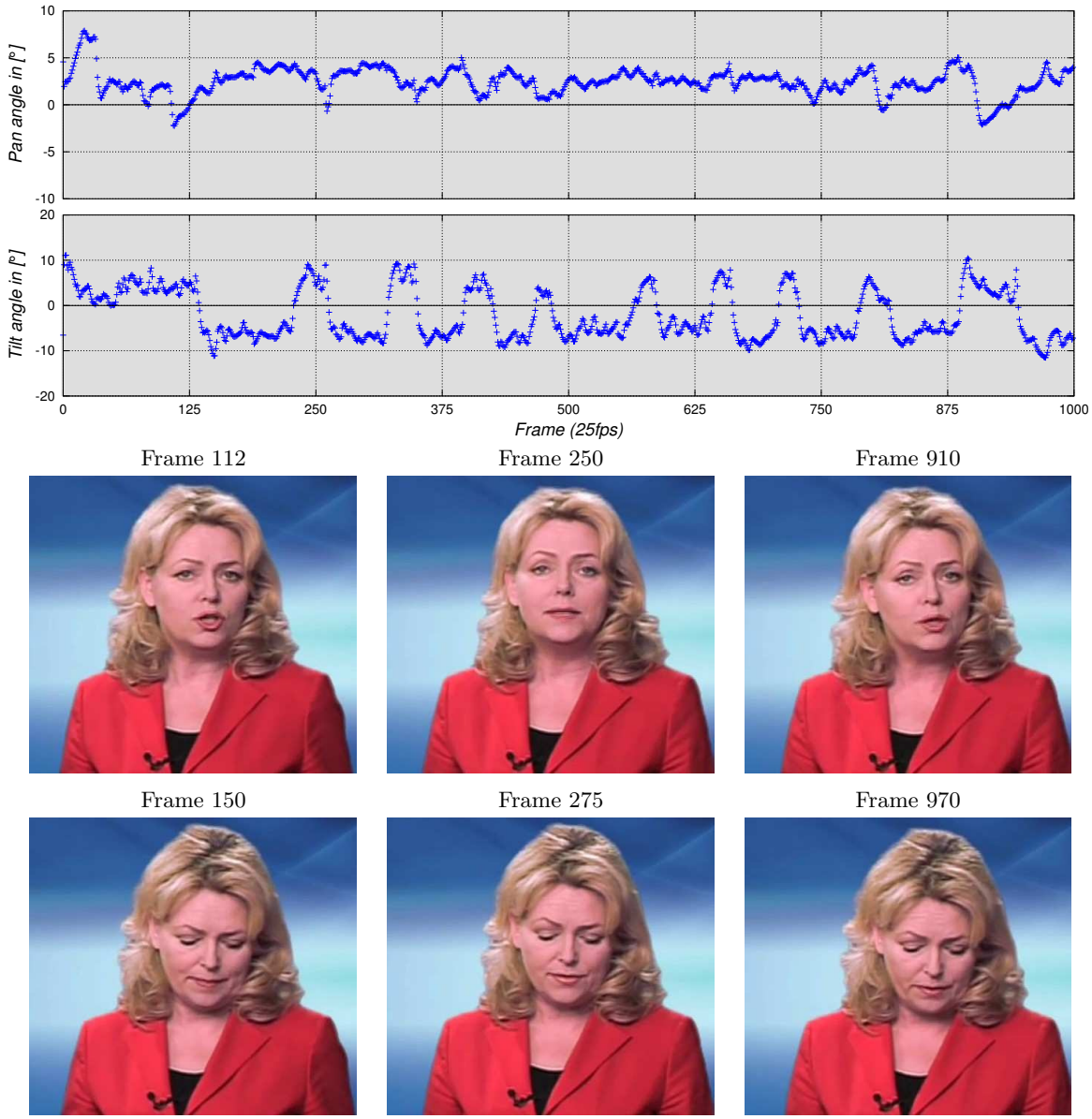
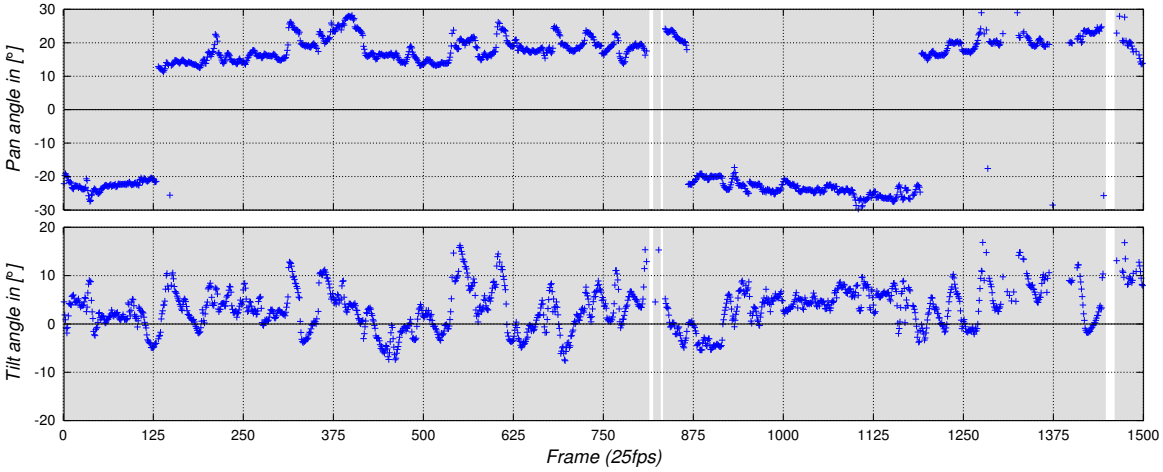



***Abbildung 4.36:*** *Ergebnisse auf Sequenz "Nachrichten 2": Die Frames 112 und 910 zeigen eine sichtbare horizontale Verdrehung nach rechts. Im Frame 250 ist die Kopfpose fast mittig. In den Frames 150, 275 und 970 ist eine deutliche Kopfbewegung nach unten sichtbar. Der zeitliche Verlauf der vertikalen Kopfpose zeigt die für eine Nachrichtensprecherin typischen Nickbewegungen, die durch das Ablesen der Nachrichten bedingt sind.*




Frame 125

Frame 500

Frame 1000

Frame 1420

***Abbildung 4.37:*** *Ergebnisse auf Sequenz "Nachrichten 7": Die Frames 0-130 und 870-1180 zeigen einen Moderator und der Rest der Sequenz einen Diskussionspartner. In grau hinterlegten Bereichen konnte das Modell erfolgreich angepasst werden. Durch die jeweils schräge Kameraposition haben beide einen seitlich verdrehten Kopf. Bei beiden Gesprächspartner ist eine typische Nickbewegung des Kopfes beim Sprechen sichtbar.*

## 4.6 Zusammenfassung

Neben dem in Kapitel 3 vorgestellten System zur Schätzung der Oberkörperpose, spielt die Kopfpose eine wichtige Rolle bei der Bestimmung des Interaktionsinteresses bzw. der Aufmerksamkeit eines Benutzers. In diesem Kontext wurde die Kopfpose bisher nur in sehr wenigen Arbeiten untersucht.

In den vorangegangenen Abschnitten wurde ein System präsentiert, das die Kopfpose eines Benutzers in Echtzeit auf einem mobilen Roboter schätzen kann. Hierzu wird zunächst ein Gesicht im Bild detektiert und anschließend ein parametrisches Modell auf dieses Gesicht angepasst. Die Modellparameter werden anschließend zur Schätzung der Kopfpose herangezogen.

Zur Gesichtsdetektion kommt ein herkömmlicher *Viola & Jones* - Gesichtsdetektor zum Einsatz, da dieser eine hohe Erkennungsrate, eine verhältnismäßig geringe *False-Positive-Rate* besitzt und in Echtzeit arbeiten kann. Die Detailanpassung arbeitet mit Hilfe von *Active Appearance Models*, die eine Weiterentwicklung der *Active Shape Models* darstellen. Dabei kommt der aus der Literatur bekannte *Project-Out-Algorithmus* zum Einsatz, der im Rahmen dieser Dissertation durch verschiedene Anpassungen ergänzt wurde, um eine höhere Robustheit zu erreichen. Diese Anpassungen erlauben letztendlich den Einsatz der AAMs unter den gegebenen Realwelt-Bedingungen auf einem mobilen Robotersystem.

Die Auswahl der relevanten Modell-Parameter erfolgt wie im vorherigen Kapitel mit Hilfe der bekannten Technik der *Mutual Information*. Als Ergebnis steht somit auch hier ein niedrigdimensionaler Merkmalsvektor zur Weiterverwendung zur Verfügung.

Zur eigentlichen Schätzung der Kopfpose wurden eine einfache lineare Approximation und ein *Multi-Layer-Perceptron* eingesetzt. Es wurde gezeigt, dass auf Grund der Struktur der resultierenden Modellparameter, eine einfache lineare Approximation oder ein einfaches Perceptron in der Lage ist, die Kopfpose auf 3° bis 5° genau auf der Testdatenbank zu schätzen.

Tests auf Videosequenzen mit unbekannten Daten haben gezeigt, dass das System auch in der Lage ist, sich an unbekannte Gesichter gut anzupassen und auch hier eine hinreichend genaue Kopfposenschätzung vorzunehmen. In diesem Punkt unterscheidet sich diese Dissertation von vielen anderen Arbeiten, in denen oftmals nur Tests und Auswertungen auf selbst aufgenommenen Bild- und Videomaterial durchgeführt werden.

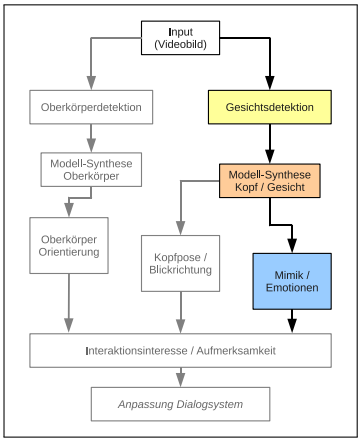
Probleme bei der Schätzung der Kopfpose treten insbesondere dann auf, wenn das Active Appearance Model nicht genau genug an das Inputbild angepasst werden kann. Auf

Grund eines falschen Modells entstehen dann falsche Parameter, die zu einer fehlerhaften Schätzung der Kopfpose führen können. Eine robuste Anpassung des Modells ist somit eine wichtige Voraussetzung. Zur Erhöhung der Robustheit der Modellanpassung wurden einige Möglichkeiten und Erweiterungen aufgezeigt, von denen einige im Rahmen dieser Dissertation eingesetzt wurden.

Zusammenfassend kann trotzdem festgestellt werden, dass der vorgestellte *HeadPoseTracker* unter der Randbedingung einer guten Modellanpassung geeignet ist, um die Kopfpose eines Interaktionspartners in Echtzeit zu schätzen. Somit steht eine weitere Grundlage zur Schätzung der Aufmerksamkeit bzw. des Interaktionsinteresses zur Verfügung.

# 5 Teilsystem 3: Emotionsschätzung

## 5.1 Einleitung




In den Kapiteln 3 und 4 wurde beschrieben, wie die Orientierung des Oberkörpers und die Kopfpose als zwei mögliche Merkmale zur Schätzung des Interaktionsinteresses bestimmt werden können. Diese beiden Systeme sind Teil eines Verfeinerungsprozesses, der durch eine weitere Stufe in diesem Kapitel ergänzt werden soll.

Nachdem sich ein (möglicher) Benutzer einem mobilen Robotersystem genähert hat und ein Dialog begonnen wurde, richtet der Benutzer seine Aufmerksamkeit typischerweise auf den Bildschirm, da meist dort die relevanten Informationen angezeigt werden. Hiermit ergibt sich die Möglichkeit, über eine im Bildschirm integrierte Kamera ein quasi frontales Bild des Gesichtes des Benutzers aufzunehmen und somit den Benutzer genau(er) beobachten und auch analysieren zu können.

Über den Gesichtsausdruck (Mimik) kann der emotionale Zustand des Kommunikationspartners bestimmt werden. Bevor ein Benutzer sich beispielsweise auf Grund von Langeweile wieder abwendet, zeigt sich das Desinteresse oftmals bereits in seinem Gesichtsausdruck. Weiterhin kann die Mimik auch benutzt werden, um z.B. im Rahmen eines kognitiven Übungsprogramms den Schwierigkeitsgrad so anzupassen, dass dieser optimal auf den Benutzer abgestimmt ist. Er sollte nicht unterfordert werden (z.B. sichtbar durch einen gelangweilten Gesichtsausdruck) aber auch nicht überfordert (z.B. Grübeln oder Frustration).

Die Mimik ist somit ein weiteres wichtiges Merkmal, dass zur Bestimmung des Interaktionsinteresses bzw. der Aufmerksamkeit genutzt werden kann. Im Rahmen dieses Kapitels soll ein *MimicDetector* vorgestellt werden, der dieses Merkmal bestimmen kann. Dabei soll nicht die Bestimmung von einzelnen Emotionen im Vordergrund stehen, sondern vielmehr eine generelle Einschätzung des emotionalen Zustands in einem kontinuierlichen Emotionsraum.

Im Folgenden werden im Abschnitt 5.2 zunächst einige Emotionsmodelle vorgestellt.

Anschließend werden im Abschnitt 5.3 verschiedene relevante Methoden aus der Literatur zum Thema Emotionsschätzung präsentiert. Im Anschluss werden die Struktur und die einzelnen Bestandteile eines Systems zur Mimikerkennung detailliert vorgestellt. Das Kapitel endet in Abschnitt 5.6 mit der Präsentation der erzielten Ergebnisse.

## 5.2 Emotionsmodelle

Um das Interaktionsinteresse eines Benutzers mit einem Serviceroboter zu bestimmen, ist es zunächst notwendig, menschliche Emotionen genauer zu betrachten. Emotionen entstehen bei Menschen, indem durch das kognitive System die Umwelt wahrgenommen wird, durch das emotionale System bewertet und Körperfunktionen darauf hin angepasst und moduliert werden. Emotionen zeigen sich beim Menschen beispielsweise durch den Gesichtsausdruck, die Muskelspannung, Bewegungen, Gesten, die Wortwahl, Sprache und die Hauttemperatur. Der Mensch kann ausgedrückte Emotionen jedoch in gewissem Maße beeinflussen. Beispielsweise können Emotionen gespielt oder bewusst unterdrückt werden.

Beim Menschen sind *Emotionen (emotions)* und *Stimmungen (moods)* von dem Begriff der *Persönlichkeit* zu unterscheiden:

| | **Emotionen/Stimmungen** | **Persönlichkeit** |
|---|---|---|
| Zeithorizont | kurz- bis mittelfristig | praktisch zeitinvariant |
| Veränderbarkeit | durch Wahrnehmung änderbar | durch Wahrnehmung kaum änderbar |
| Quelle/Ursprung | sind abhängig von eigenen Aktionen (z.B. Stolz oder Freude nach einem Erfolg) | vererbt und/oder durch die Erziehung herausgebildet |
| Vorhandensein | spezifisch an eine Situation oder ein Objekt gebunden | allgemein vorhanden |

Emotionen und Stimmungen sind sehr ähnlich und miteinander eng verknüpft. Im Gegensatz zu Emotionen, die typischerweise recht kurzfristig sind, sind Stimmungen eher mittelfristig und können sich beispielsweise abhängig von der Tages- oder Jahreszeit ändern.

Es gibt eine Vielzahl von Emotionstheorien. Seit dem Beginn der Erforschung und Entwicklung von Emotionsmodellen in den 1890er Jahren von William James wurden diese vielfach überarbeitet und auch revidiert. Eine relativ einfache und anschauliche Definition findet sich zum Beispiel in [Schmidt-Atzert, 1996]:

> „Eine Emotion ist ein qualitativ näher beschreibbarer Zustand, der mit Veränderungen auf einer oder mehreren der folgenden Ebenen einhergeht: Gefühl, körperlicher Zustand und Ausdruck.“

Eine andere Definition aus [Catrin, 2006] lautet:

> „Emotion ist ein psychophysiologischer Prozess, der durch die kognitive Bewertung eines Objekts ausgelöst wird und mit physiologischen Veränderungen, spezifischen Kognitionen, subjektivem Gefühlserleben und einer Veränderung der Verhaltensbereitschaft einhergeht. Emotionen treten beim Menschen und bei höheren Tieren auf."

Dies sind nur zwei von vielen vorhandenen Definitionen. Mehr als einhundert verschiedene wurden von [Kleinginna and Kleinginna, 2005] zusammengefasst.

William James betrachtete ausschließlich „gröbere" Emotionen, das heißt Emotionen, welche starke körperliche Reaktionen auslösen. Dazu gehören zum Beispiel Zorn, Furcht, Hass und Freude. Später wurden diese Theorien immer weiter verfeinert und von der bloßen körperlichen Äußerung getrennt.

Heute kann man die Emotionstheorien grundsätzlich in zwei Klassen einteilen. Zum einen gibt es die kategoriale Klassifikation, die Emotionen als Mischung aus bestimmten Basisemotionen ansieht. Zum anderen gibt es die dimensionale Klassifikation von Emotionen, die Emotionen als Ausprägung auf bestimmten Dimensionen ansieht.
Bei der kategorialen Klassifikation von Emotionen wird davon ausgegangen, dass sich die Emotionen aus bestehenden Basisemotionen zusammensetzen. Das sind Emotionen, die nicht weiter auf andere Emotionen zurückgeführt werden können beziehungsweise Emotionen, aus denen sich alle anderen - komplexeren - Emotionen zusammensetzen lassen.
Bei der dimensionalen Klassifikation von Emotionen wird angenommen, dass die Emotion das Resultat einer mehr oder minder starken Ausprägung aus mehreren bestimmten Dimensionen ist. Es besteht jedoch Uneinigkeit darüber, welche Dimensionen dies sind. Diese Dimensionen können zum Beispiel die folgenden drei sein: Spannung-Lösung, Lust-Unlust und Erregung-Beruhigung.
Im Folgenden sollen je ein weit verbreiteter Vertreter der Emotionsmodelle aus jeder der beiden Gruppen näher vorgestellt werden.

### 5.2.1 Diskrete Basisemotionen

Der Grundgedanke der diskreten Emotionsmodelle besteht in der Annahme der Existenz einer kleinen Anzahl von bestimmten Basisemotionen (*basic, primary or fundamental emotions*), mit Hilfe derer alle anderen Emotionen dargestellt werden können. Die Basisemotionen selbst können nicht auf andere Emotionen zurückgeführt werden. In [Ortony and Turner, 1990] werden eine Reihe solcher Modelle benannt und miteinander verglichen.
Im Rahmen dieser Dissertation sollen vorrangig solche Methoden und auch entsprechende Emotionsmodelle betrachtet werden, die den emotionalen Zustand eines Interaktionspartners

mit Hilfe des Gesichtsausdrucks bestimmen. Andere Modalitäten (wie z.B. Sprache oder Körperhaltung) sollen hier nicht weiter berücksichtigt werden.

### Ausdruck der Basisemotionen im Gesicht

Das menschliche Gesicht besteht aus einer Vielzahl von Muskeln, die in verschiedenen Kombinationen letztendlich den Gesichtsausdruck bestimmen. Einige dieser Muskeln bzw. Muskelgruppen können bewusst gesteuert werden, andere jedoch nicht. Daher kann es schwierig sein, bestimmte Gesichtsausdrücke bewusst darzustellen.
Ein Modell zur Katalogisierung von Bewegungen und Verformungen im Gesicht ist das in den 1970er Jahren von [Ekman and Friesen, 1978] entwickelte *FACS - Facial Action Coding System*. Eine überarbeitete Version wurde 2002 in [Hager et al., 2002] präsentiert. In FACS werden alle möglichen gesichtsspezifischen Veränderungen zusammengestellt, ohne dass dabei eine Interpretation bezüglich des emotionalen Zustandes einer Person vorgenommen wird. Das seit langer Zeit in der Verhaltensforschung angewendete FACS stellt ein Wertungssystem dar, dass für einen menschlichen Beobachter entwickelt wurde. Das System baut sich aus 46 sogenannten *Action Units (AU)* auf. Sie beschreiben elementare Aktionen im Gesicht, sowohl nach ihrem Ort als auch nach ihrer Intensität. Aus der Kombination verschiedener Action Units können im Ergebnis letztendlich (Basis-)Emotionen interpretiert werden. In Tabelle 5.1 sind einige mögliche Basisemotionen und deren Darstellung/Codierung im FACS System aufgeführt.

| **Basisemotion** | **Action Units (AU)** |
|---|---|
| Überaschung | 1 + 2 + 5 + 26 |
| Furcht | 1 + 2 + 4 + 5 + 7 + 20 + 25, 26 |
| Ekel | 4 + 9 + 17 |
| Wut | 4 + 5 + 7 + 10 + 25, 26 |
| Freude | 6 + 12 (+ 26) |
| Trauer | 1 + 4 + 15 |

***Tabelle 5.1:*** *Beispiele zur Interpretation von Basisemotionen mit Hilfe von Action Units aus dem FACS System entnommen aus [Kobayashi and Hara, 1997].*

### Modell der Basisemotionen von Ekman et.al.

Im Bereich der *Human-Computer-Interaction* und *Human-Robot-Interaction* werden oft die von [Ekman et al., 1982] vorgestellten sechs Basisemotionen verwendet: Ärger (*anger*), Ekel (*disgust*), Angst (*fear*), Freude (*joy/happy*), Traurigkeit (*sadness*) und Überraschung (*surprise*). Für automatische Emotionserkennungssysteme sind diese sechs Basisemotionen die bekanntesten und gebräuchlichsten Kategorien zur Klassifizierung von Gesichtsausdrücken.

In einigen Arbeiten werden diese sechs Basisemotionen noch zusätzlich um den „Neutral"-Zustand ergänzt. Abbildung 5.1 zeigt Beispielbilder für die sechs Basisemotionen aus der FEEDTUM-Gesichtsdatenbank [Wallhoff, 2006] (oder auch: *FG-NET Database with Facial Expressions and Emotions*) der Technischen Universität München.


***Abbildung 5.1:*** *Beispielbilder für die sechs Basisemotionen Ärger, Ekel, Angst, Freude, Traurigkeit und Überraschung (entnommen aus der FEEDTUM-Gesichtsdatenbank [Wallhoff, 2006]).*

#### Probleme der Modelle mit diskreten Basisemotionen

In verschiedenen Publikationen wurde die Existenz dieser universellen Klassen von Basisemotionen immer wieder in Frage gestellt. Es ist auch nicht endgültig geklärt, inwieweit diese Basisemotionen in Zusammenhang mit tatsächlichen emotionalen Zuständen zu bringen sind oder ob sie nur als Möglichkeit genutzt werden, um verbale Kommunikation visuell zu unterstützen. Trotz dieser Fragen, und auch wenn mit Sicherheit nicht jeder darstellbare Gesichtsausdruck einer dieser Klassen zugeordnet werden kann, wird in den meisten Studien, die sich mit der Analyse von Gesichtsausdrücken befassen, diese Kategorisierung in Basisemotionen genutzt. Probleme ergeben sich hierbei, sobald ein Gesichtsausdruck nicht eindeutig einer Emotionsklasse zugeordnet werden kann bzw. es sich um nur einen schwach ausgeprägtem Gesichtsausdruck handelt.

### 5.2.2 Kontinuierliches Emotionsmodell - The circumplex model of affect

Im Gegensatz zu den Modellen mit diskreten Basisemotionen, werden bei den kontinuierlichen Emotionsmodellen die verschiedenen Emotionen in einem $n$-dimensionalen Raum angeordnet. In verschiedenen Modellen werden oft folgende Dimensionen verwendet:

| **Dimension** | **Wertebereich** |
|---|---|
| *Stance/Haltung*: | offen ... geschlossen |
| *Valence/Wertigkeit*: | angenehm ... unangenehm |
| *Arousal/Erregung*: | stark ... schwach |

In [Breazeal, 1999] werden diese drei Dimensionen beispielsweise genutzt, um auf der Roboterplattform *Kismet* verschiedene Emotionen darzustellen. Dazu werden die einzelnen

Dimensionen auf verschiedene Aktoren des Gesichts von Kismet gemappt und zur Darstellung von Emotionen entsprechend angesteuert.

In [Russell, 1980] werden basierend auf den Arbeiten von Schlosberg (1952), Fillenbaum/Rapoport (1971) und Block (1957) acht Emotionen in einem kreisförmigen Modell angeordnet (siehe Abbildung 5.2). Die horizontale Achse des Modells beschreibt dabei die Dimension *pleasure* (Freude) ↔ *displeasure* (Unmut) und die vertikale Achse die Dimension *arousal* (Erregung) ↔ *sleep* (Schlaf). Die verbleibenden vier Emotionen des Modells sind dabei keine unabhängigen Zustände, sondern lassen sich mit Hilfe der vier anderen beschreiben. Diese Nebenzustände dienen mehr zur Charakterisierung der Quadranten des Zustandsraumes.

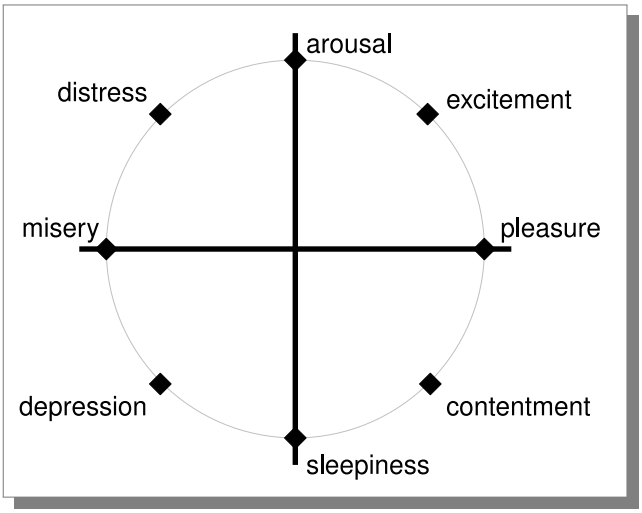



***Abbildung 5.2:*** *Acht Emotionen auf einem Kreis in einem zweidimensionalen Emotionsraum nach [Russell, 1980].*

In einer Reihe von Studien wurden Probanden aufgefordert, verschiedene Emotionsbegriffe in dieses Modell einzutragen. Dabei waren sowohl die acht Basisemotionen als auch 28 weitere Emotionsbegriffe einzuordnen. Als Ergebnis entstand eine Tabelle von Positionsangaben (Winkelpositionen) der ausgewählten Emotionen in dem vorgestellten kreisförmigen Modell.

Eine Fortsetzung dieser und ähnlicher Arbeiten findet sich beispielsweise in [Scherer, 2005]. Dort werden eine Reihe von Emotionen in einem kreisförmigen Modell angeordnet. Dabei werden die Emotionen nicht nur auf dem Kreis selbst, sondern auch im Inneren des Kreises plaziert (siehe Abbildung 5.3). In diesem Modell sind neben den 28 Emotionsbegriffen nach Russell (Begriffe mit Großbuchstaben in Abbildung 5.3) weitere 80 Emotionsbegriffe aus früheren Studien von Scherer (Begriffe mit Kleinbuchstaben in Abbildung 5.3) dargestellt.

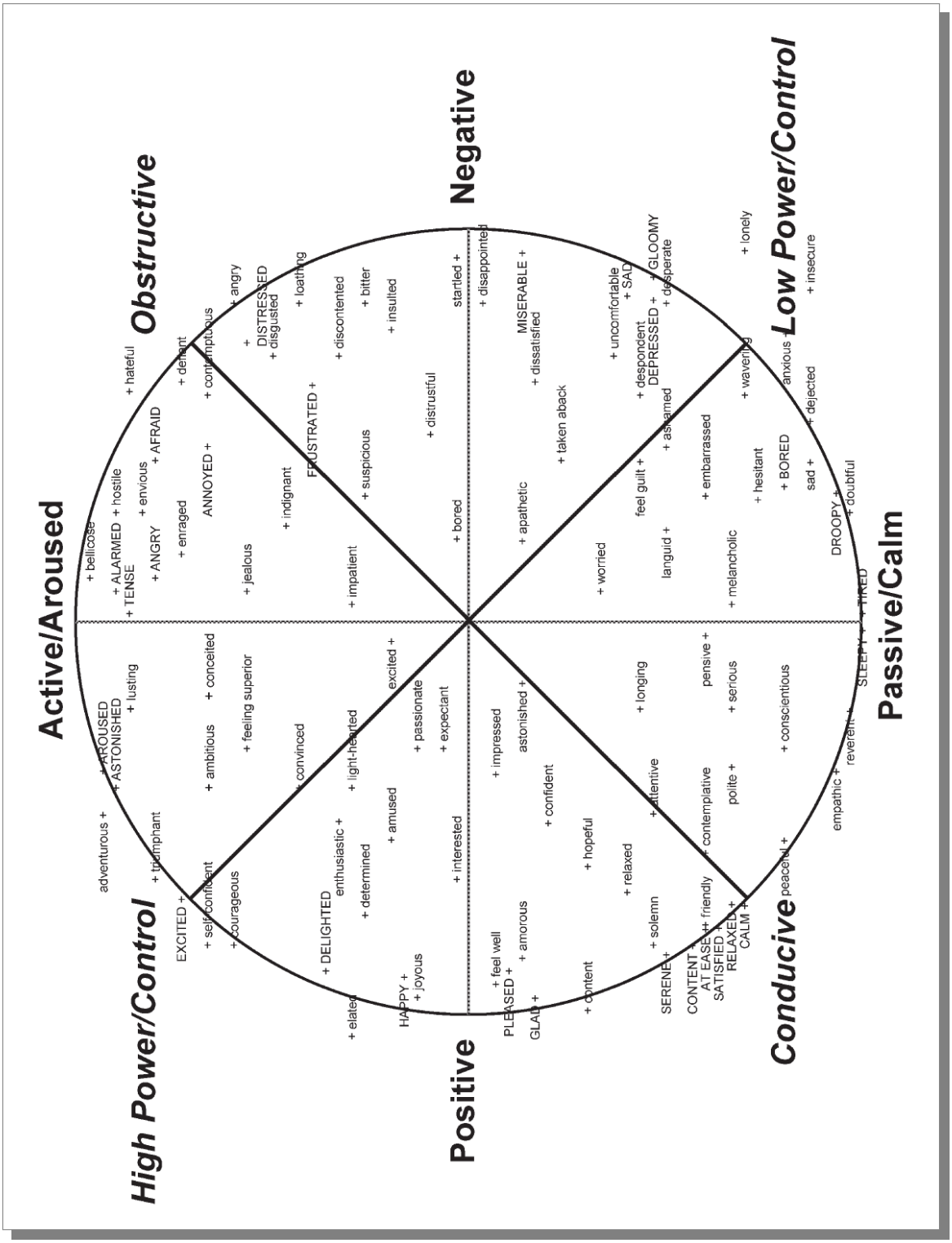



***Abbildung 5.3:*** *Emotionsmodell nach [Scherer, 2005]. Großbuchstaben: 28 Emotionsbegriffe nach Russell, Kleinbuchstaben: 80 Emotionsbegriffe nach Scherer.*

## 5.3 State-of-the-Art

Bevor Gesichtsmerkmale zur Schätzung von Emotionen aus einem Eingabebild extrahiert werden können, muss wie im vorangegangenen Kapitel das Gesicht zunächst detektiert werden. Eine kurze Übersicht zur Gesichtsdetektion findet sich im vorherigen Kapitel in Abschnitt 4.2.1.

#### Mimikschätzung mittels Features

Nach der Detektion eines Gesichtes, müssen die Gesichtsmerkmale extrahiert werden, mit denen der Gesichtsausdruck bestimmt werden soll. Ein Gesichtsausdruck entsteht durch die Kontraktion von Gesichtsmuskeln, die Veränderungen in Aussehen und Form von markanten Gesichtsregionen bewirken, wie zum Beispiel Augen und Lippen (siehe vorn: *FACS - Facial Action Coding System*). Zusammen mit ihrer Richtung und Geschwindigkeit bilden diese Veränderungen Gesichtsmerkmale. Diese beinhalten auch Texturen und Kanten im Gesicht, welche nicht permanent, sondern nur in Zusammenhang mit bestimmten Gesichtsausdrücken auftreten, auch wenn diese mit zunehmendem Alter einmal dauerhaft werden können, wie zum Beispiel bei Krähenfüßen. Um diese Gesichtsmerkmale verfolgen zu können, ist eine interne Repräsentation des Gesichts im Eingabebild notwendig.

Basierend auf den *Action Units (AUs)* kann beispielsweise eine regelbasierte Klassifikation wie in [Pantic and Rothkrantz, 2000a] durchgeführt werden. Dabei werden zunächst festgelegte Merkmalspunkte (Abbildung 5.4) im Gesicht lokalisiert und dann die Modellmerkmale an diesen Positionen im Bild bestimmt. Aus der Differenz der Modellmerkmale mit den Merkmalen, die bei einem neutralen Gesichtsausdruck derselben Person bestimmt wurden, kann die AU bestimmt werden, die diese Änderung bewirkt. Insgesamt können die Bewegungen von 31 verschiedenen AUs detektiert werden. Grundvoraussetzung bei diesem Verfahren ist, dass für jede Person immer auch eine (Vergleichs-)Aufnahme mit einem neutralen Gesichtsausdruck vorhanden ist, um die Bewegungen der AUs richtig detektieren zu können. Im Rahmen der geplanten realwelt-tauglichen Implementierung in dieser Dissertation ist die Aufnahme eines neutralen Gesichtsausdruck nicht praktikabel. Daher wird diese Variante hier nicht weiter betrachtet.

#### Mimikschätzung mittels Active Appearance Models

Eine weitere Möglichkeit sind modellbasierte Repräsentationen, bei denen ein Template eines Gesichtsmodells an das Eingabebild angepasst und verfolgt wird. Die resultierenden Modellparameter können nach der Anpassung genutzt werden, um Informationen zu dem gezeigten Gesichtsausdruck zu erhalten. Ein oft verwendetes Verfahren sind dabei die *Active Appearance Models* (AAMs). Diese werden beispielsweise in [Edwards et al., 1998], [van

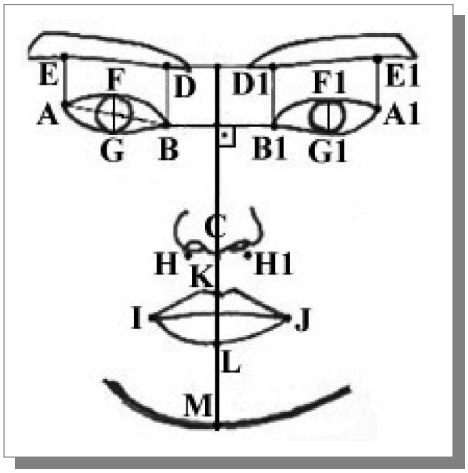



***Abbildung 5.4:*** *Knotenpunkte für eine regelbasierte Klassifikation. Aus den Differenzen der Modellmerkmale an diesen Knotenpunkten mit neutralen Referenzmerkmalen wird auf die AU geschlossen, die die Merkmalsänderung bewirkt hat.*

Kuilenburg et al., 2005], [Saatci and Town, 2006] und [Wilhelm, 2005] erfolgreich eingesetzt.

In [Edwards et al., 1998] wird ein Active Appearance Model an die Eingabebilder angepasst. Insgesamt wurden 88 Trainingsbilder per Hand markiert. Mit einer Hauptkomponentenanalyse wurden aus diesen Trainingsbildern ein Form- und ein Grauwertmodell erstellt. Mit Hilfe der Modellparameter soll das Individuum posen- und mimikinvariant identifiziert werden. Um dieses Ziel zu erreichen, wird das Mahalonobis Abstandsmaß auf einen repräsentativen Trainingsdatensatz angewendet. Bei dem Klassifikator wird davon ausgegangen, dass die Intraklassenvarianz (Pose und Gesichtsausdruck) für jedes Individuum ähnlich ist. Mit einer Diskriminanzanalyse wird die Interklassenvarianz (Identität) von der Intraklassenvarianz linear getrennt. Damit können zwei orthogonale Parameterunterräume konstruiert werden. Der eine umfasst Variationen in Pose, Gesichtsausdruck und Beleuchtung. Im zweiten Parameterraum können die Individuen posen- und mimikinvariant identifiziert werden. Allerdings wird nur auf einem festen Trainingsdatensatz operiert. Wie sich diese Methode bei unbekannten Individuen verhält, wurde nicht untersucht. Getestet wurde der Anpassungsalgorithmus auf 200 Testbildern. Bei knapp 20% der Bilder, wurde nicht die gewünschte Konvergenz bei der Modellanpassung erreicht, was für eine Erkennung von Gesichtsausdrücken entscheidend ist.

In [van Kuilenburg et al., 2005] werden AAMs erfolgreich für die Extraktion von Gesichtsmerkmalen nach dem FACS System und die Klassifikation in sieben Basisemotionen eingesetzt. Als Klassifikationsmerkmale werden ausschließlich die 96 Grauwertparameter

des Grauwertmodells verwendet. Mit diesen Merkmalsvektoren wird ein dreischichtiges Multi-Layer-Perceptron mit entsprechend 96 Eingabe- und sieben Ausgabeneuronen trainiert. Dafür wird aus den Trainingsdaten ein spezielles Grauwertmodell erstellt, welches die Texturinformationen von mimischen Gesichtsausdrücken darstellen kann. Abbildung 5.5 zeigt Beispiele für Grauwertkomponenten, die bestimmte Emotionszustände repräsentieren. Die Trainingsdaten umfassen dabei 1512 Grauwertmerkmalsvektoren von automatisch angepassten AAMs. Dafür werden 980 qualitativ hochwertige Bilder aus der KDEF-Datenbank [Lundqvist et al., 1998] genutzt. Es steht von jeder Person je ein Bild für jede Emotionsklasse für das Training zur Verfügung. Die Erkennungsraten auf dem Testdatensatz liegen bei 85% bis 97%. Mit den selben Merkmalsvektoren wird auch ein Neuronales Netz für die Klassifikation von Bewegungen von AUs aus dem FACS System trainiert. Dafür werden 858 automatisch erstellte Merkmalsvektoren aus den AU-codierten Bildern der Cohn-Kanade-Datenbank [Cohn, 1999] verwendet. Bei den 15 ausgewählten AUs werden damit durchschnittlich 86% richtig klassifiziert.

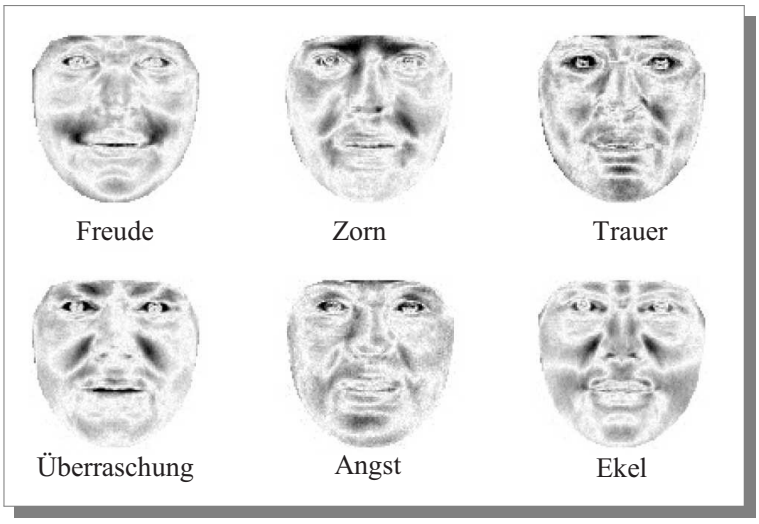



***Abbildung 5.5:*** *Spezielles Grauwertmodell für die Klassifikation von Gesichtsausdrücken. Dieses Modell enthält Grauwertkomponenten, die bestimmte Emotionsklassen repräsentieren [van Kuilenburg et al., 2005].*

In [Saatci and Town, 2006] werden AAMs verwendet, um Gesichtsmerkmale aus einem festen Datensatz zu extrahieren. Mit einer Kaskade von vier *Support Vector Maschines* (SVM) werden diese Merkmale in vier entsprechende emotionale Zustände klassifiziert. Dabei wird eine Kaskade von SVMs gewählt, um ein eindeutiges Klassifikationsergebnis zu erhalten. Bei einer parallelen Verwendung der SVMs kann es zu widersprüchlichen Ergebnissen kommen, wie zum Beispiel Freude und Trauer zur gleichen Zeit. Abbildung 5.6 zeigt den Aufbau dieser Klassifikatorkaskade. Aus einem Datensatz von 1135 Bildern aus

verschiedenen Mimikdatenbanken werden für die Trainingsmenge der AAMs lediglich 74 exakte, also per Hand markierte Beispiele genutzt. Über einen halbautomatischen *Bootstrapping* Algorithmus wird dieser Trainingsdatensatz auf 262 Bilder erweitert. Bootstrapping bedeutet, dass iterativ mit dem aktuellen Trainingsdatensatz ein AAM Modell erstellt wird, welches mit einem Anpassungsalgorithmus auf den Rest der unmarkierten Beispiele angepasst wird. Die manuell kontrollierten, augenscheinlich korrekt konvergierten Beispiele werden dem Trainingsdatensatz hinzugefügt und der Vorgang wird wiederholt. Mit dem endgültigen Modell aus diesem Verfahren wird eine Konvergenzrate von 81% auf dem gesamten Datensatz erreicht. Diese 809 Bilder werden dann für das Training und Testen des Klassifikators verwendet. Das endgültige Modell besteht aus hier 60 Parametern. Die Gesichtsmerkmalvektoren haben daher eine Länge von 60, mit denen die SVM-Kaskade trainiert wird. Die Kaskade enthält jeweils eine SVM für eine bestimmte Klasse von Emotionen, wie zum Beispiel eine SVM für Freude klassifiziert in „Freude oder keine Freude“. Von den vier Emotionsklassen Freude, Trauer, Zorn und Neutral werden von dem Testdatensatz mit diesem Verfahren 94%, 70%, 77% beziehungsweise 64% richtig klassifiziert.

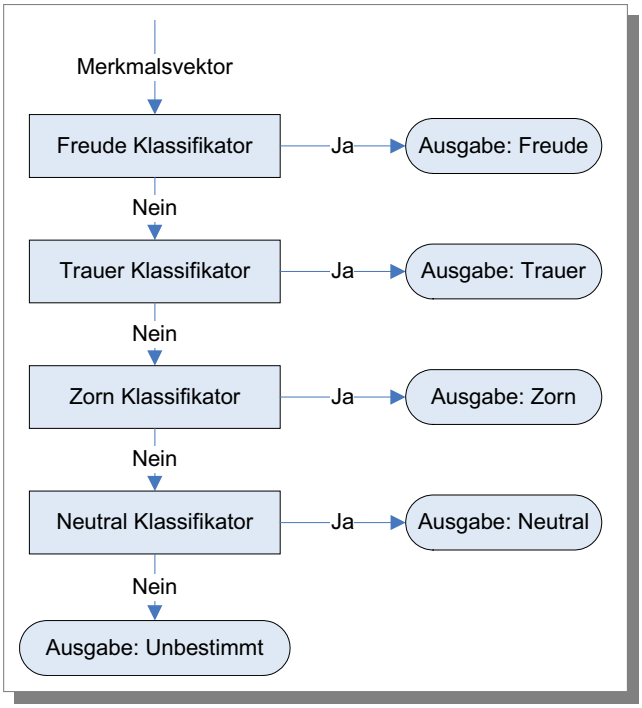



***Abbildung 5.6:*** *Kaskade von SVMs für die Klassifikation von Merkmalsvektoren der AAMs in die vier Basisemotionszustände Freude, Trauer, Zorn und Neutral [Saatci and Town, 2006].*

Eine andere Arbeit [Lucey et al., 2006] nutzt AAMs für eine Gesichtsmodellierung. Sie

bezieht sich auf die Arbeit [Donato et al., 1999] mit dem Ziel, die Bewegungen einzelner AUs nach dem FACS zu erkennen. Dabei wird sich auf die AUs 1, 1+2, 4 und 5 beschränkt. Diese beschreiben die Bewegungen der inneren und äußeren Augenbrauen und des oberen Augenlids. Dabei werden mit Hilfe einer AU-Datenbank Tests durchgeführt. Diese Datenbank enthält Bilder von 33 Personen unterschiedlicher ethnischer Herkunft mit realwelt-ähnlichen Gesichtsausdrücken. Mit dem System soll bei diesen Personen über die Bewegungen der AUs erkannt werden, ob sie zu einem bestimmten Sachverhalt die Wahrheit sagen oder lügen. Für das Tracking der Personen und die robuste Verfolgung der AU-Bewegungen werden spezielle, personenspezifische, zwei- und dreidimensionale Modelle für die relevanten Gesichtsregionen der AAMs verwendet. Aus dem angepassten Formmodell werden die Bewegungen der einzelnen AUs klassifiziert. Dies geschieht zum einen mit der *Nearest Neighbour* Methode in einem dimensionsreduzierten, linearen Unterraum durch eine PCA und zum anderen durch lineare SVMs, bei denen Trennhyperebenen optimal zwischen die positiven und negativen Beobachtungsvektoren plaziert werden. Diese Arbeit kommt zu dem Ergebnis, dass die dreidimensionalen Formen keine nennenswerten Steigerungen bei der Erkennung erzielen. Als zweites wird hervorgehoben, dass die Normalisierung der Form durch eine Entfernung der Ähnlichkeitstransformationsanteile eine deutliche Steigerung der Erkennungsrate bewirkt. Wenn das Grauwertbild durch ein lokales Mittelwertbild normalisiert wird, kann auch dieses für die Erkennung erfolgreich eingesetzt werden. Insgesamt ist die Erkennung der Bewegungen der AUs 1, 1+2 und 4, also der Augenbrauen mit Raten von etwa 70 bis 90 Prozent, am erfolgreichsten. Die Erkennung der Bewegung des oberen Augenlids der AU 5 erreicht diese Werte dagegen bei weitem nicht. Die Arbeit kommt zu dem Schluss, dass ein möglichst gut angepasstes Formmodell ein entscheidender Schritt für die automatische Erkennung der Bewegungen von AUs und damit für die Erkennung von beliebigen Gesichtsausdrücken ist.

***Abbildung 5.7:*** *Beispielemotionen aus [Wilhelm et al., 2005]: Neutral, Surprise, Sadness, Anger, Fear, Happiness, Disgust (v.l.n.r).*

Eine wichtige Grundlage für diese Dissertation bilden die Arbeiten von [Wilhelm, 2005]. In seinen Arbeiten wird u.a. die Leistungsfähigkeit verschiedener Ansätze im Bereich der Mimikanalyse untersucht. Dabei werden Active Appearance Modells, Elastic Graph Matching und die ICA miteinander verglichen. Bei einem AAM kombiniert mit einem MLP

wurde eine Klassifikationsrate von 72% auf den sechs Basisemotionen und Neutral erzielt. Als Datenbasis wurde hierbei eine Datenbank von 30 Personen verwendet, die alle jeweils die sechs Basisemotionen gut und sehr deutlich darstellen konnten. Ein Beispiel findet sich in Abbildung 5.7.

In [Ratliff and Patterson, 2008] werden Active Appearance Models zur Mimik-Klassifikation auf der FEEDTUM-Gesichtsdatenbank [Wallhoff, 2006] verwendet. Hierbei kommt ein einfacher auf dem euklidischen Abstand basierender Klassifikator zum Einsatz. Für jede Emotion wird dazu ein mittlerer Parametervektor aus den Trainingsdaten bestimmt. In der Kann-Phase wird immer die Klasse mit dem kleinsten euklidischen Abstand ausgewählt. Mit dieser einfachen Schätzung wird für die Basisemotionen *Sadness* und *Fears* eine Klassifikationsrate von ca. 65% erreicht. Für *Surprise* liegt die Erkennungsrate bei 80% und bei den anderen Emotionen bei knapp über 90%.

In der Arbeit von [Stricker et al., 2010] und deren Fortsetzung in [Hommel and Handmann, 2011], wurden Active Appearance Modelle zur Mimikschätzung in einem kontinuierlichen Emotionsraum eingesetzt. Mit Hilfe eines zur Laufzeit berechneten personenspezifischen Mittelwertgesichts wurde eine Schätzung auf der Positiv-Negativ-Achse des Emotionsmodells vorgenommen. Die Tests wurden ebenfalls auf der FEEDTUM-Gesichtsdatenbank [Wallhoff, 2006] durchgeführt. Es konnte gezeigt werden, dass das personenspezifischen Mittelwertgesicht die Schätzung deutlich verbessert. Jedoch bleibt unklar, wie sichergestellt werden kann, dass das Mittelwertgesicht nur auf Basis der Bilder eines neutralen Gesichtsausdrucks adaptiert werden kann.

#### Mimikschätzung ohne Active Appearance Models

Andere modellbasierte Methoden bei der Merkmalsextraktion für die Erkennung von Gesichtsausdrücken sind zum Beispiel *Gaborwavelets* in [Hong et al., 1998]. Dabei werden an bestimmten Knotenpunkten, die über *Labeled Graphs* (Abbildung 5.8 links) lokalisert werden, Gaborfilterantworten zu sogenannten *Jets* zusammengefasst und als Merkmale verwendet.

In [Zhang et al., 1998] dienen die geometrischen Positionen von bestimmten, im Gesicht plazierten Knotenpunkten als Grundlage für die Bestimmung des Gesichtsausdrucks. An jedem dieser Knotenpunkte werden die Koeffizienten von Gaborwavelets bestimmt. Abbildung 5.8 rechts zeigt Beispiele für Filterantworten verschiedener Gaborwavelets. Die Koeffizienten dienen als Eingabe für ein Neuronales Netzwerk. Klassifiziert wird in eine von sieben Emotionskategorien, die sechs Basisemotionen und einen neutralen Zustand umfassen. Damit hat die Ausgabeschicht ein Neuron für jede dieser Kategorien. Der Ausgabewert des jeweiligen Neurons ist die Wahrscheinlichkeit, dass der entsprechende Gesichtsausdruck

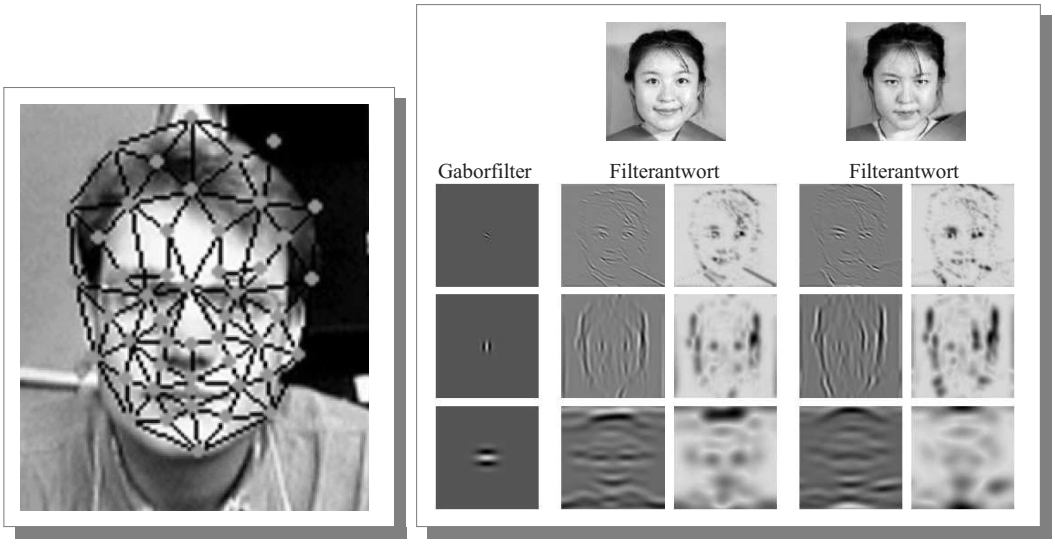



***Abbildung 5.8:*** **Links:** *Modell eines Labeled Graph mit 50 Jets. Jets bestehen aus Filterantworten von Gaborwavelets in verschiedenen Richtungen und Frequenzen [Hong et al., 1998].*
**Rechts:** *Filterantworten von verschiedenen Gaborwavelets. Die Koeffizienten dienen als Eingabe für ein Neuronales Netzwerk zur Klassifikation in Basisemotionen [Zhang et al., 1998].*

dargestellt wird. Trainiert wird das Netzwerk durch *Backpropagation* mit einem Datensatz von 213 Bildern, der in zehn Segmente unterteilt ist. Dabei werden jeweils neun Segmente des gesamten Datensatzes zum Training verwendet und eines zum Testen. Zu diesen Bildern sind die Positionen der festgelegten Knotenpunkte im Voraus exakt bekannt. Unter diesen Bedingungen werden sehr gute Erkennungsraten von bis zu 90% erreicht. Wie sich das Netzwerk bei Bildeingaben außerhalb des Trainingsdatensatzes verhält, ist nicht getestet worden. Der Einsatz dieses Systems auf unbekannten Bildern ist jedoch nicht ohne weiteres möglich, da die Positionen der Knotenpunkte exakt bekannt sein müssen. Außerdem ist das verwendete Verfahren der Segmentierung möglicherweise anfällig bezüglich einer Überanpassung, da zumindest indirekt auf denselben Daten getestet wird, mit denen das Netzwerk trainiert wurde.

In [Wilhelm, 2005] wurden neben den Active Appearance Modells auch andere Verfahren zur Mimikklassifikation untersucht. Eine Kombination von ICA mit einem Nearest-Neighbour-Klassifkator erreichte mit 74% das beste Ergebnis. Im Vergleich wurde auch das *Elastic-Graph-Matching* untersucht. Dieses erreicht mit 52% jedoch nur ein deutlich schlechteres Ergebnis.

#### Zusammenfassung

Es gibt eine Vielzahl von Arbeiten, die sich mit der Erkennung des Gesichtsausdrucks beschäftigen. Auf Grund der Auswahl der "Analyse durch Synthese" als methodisches Framework im Rahmen dieser Dissertation, sind insbesondere die Arbeiten basierend auf den Active Appearance Modellen von Interesse. Bei nahezu alle Arbeiten erfolgt die Klassifikation auf Basis von diskreten Emotionsklassen. Ein kontinuierliches Emotionsmodell wird nur in sehr wenigen Arbeiten verwendet. Als beste Klassifikationsergebnisse werden bis zu 90% auf den diskreten Emotionsklassen erreicht.

Die Arbeiten von [Wilhelm, 2005] und [Stricker et al., 2010] bilden eine Grundlage dieser Dissertation und wurden mit dem Ziel der Verwendung zur Schätzung des Interaktionsinteresses bzw. der Aufmerksamkeit weiterentwickelt.

## 5.4 Systembeschreibung

Im Rahmen dieser Dissertation sollen zur Emotionsschätzung ebenfalls *Active Appearance Modells* eingesetzt werden. Die dazu notwendige Architektur ist daher fast identisch mit der Systemstruktur der Kopfposenschätzung (siehe Abschnitt 4.3). Nach einer erfolgreichen Grobdetektion muss ein Modell an das Gesicht angepasst werden. Die eigentliche Emotionsschätzung erfolgt anschließend auf Basis ausgewählter Modellparameter. Wie im vorherigen Kapitel kann die Aufgabe der Emotionsschätzung also in drei Teilschritte untergliedert werden (siehe auch Abbildung 5.9):

1. Grobdetektion des Gesichts im Kamerabild:
   In der ersten Stufe der Verarbeitungskette soll lediglich die Grobdetektion des Gesichts im Inputbild vorgenommen werden. Wie im vorherigen Kapitel wird hierzu auch der in [Viola and Jones, 2001, Viola and Jones, 2002] vorgestellte Gesichtsdetektor verwendet.

2. Feinanpassung Gesichtsmodell:
   Nach der erfolgreichen Grobanpassung wird ein *Active Appearance Modell* (basierend auf den Vorarbeiten von [Wilhelm, 2005], [Werner, 2007][1] und [Stricker, 2008][1] an das Inputbild angepasst. Als Ergebnis steht somit ein an das Inputbild angepasstes Gesichtsmodell in parametrischer Form zur Verfügung.

3. Mimikschätzung:
   In dieser letzten Stufe wird die eigentliche Emotionsschätzung vorgenommen. Hierzu werden geeignet ausgewählte Modellparameter mit Hilfe eines Funktionsapproximators

[1] Diese Diplomarbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

bzw. Klassifikatorsystems in einen Emotionsparameterraum überführt. Die Klassifikation wird einerseits basierend auf den diskreten Basisemotionen und andererseits mittels einer Schätzung des emotionalen Zustands in einen kontinuierlichen Emotionsraum untersucht.

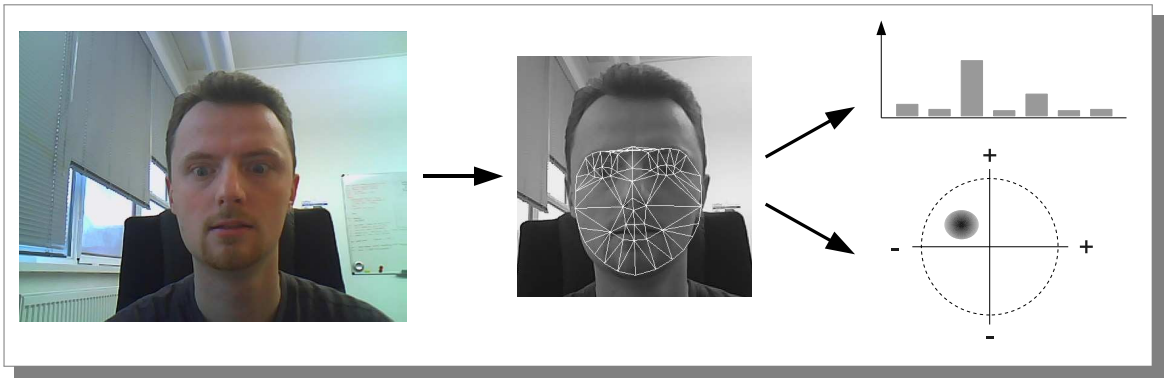



***Abbildung 5.9:*** *Systemaufbau zur Emotionsschätzung: Im Inputbild wird ein Gesicht mit Hilfe eines Gesichtsdetektors gesucht. In der relevanten ROI wird das parametrische Gesichtsmodell angepasst. Als Ergebnis entsteht ein Modell, aus dessen Parametern der Gesichtsausdruck als diskrete Basisemotion oder in einem kontinuierlichen Emotionsraum bestimmt werden kann.*

Als Alternative zu Schritt 1 kann auch der bereits bekannte Output des Teilsystems zur Kopfposenschätzung verwendet werden, insofern auf dem gleichen Kamerabild gearbeitet wird. In diesem Fall ist keine erneute Gesichtsdetektion notwendig, sondern es kann direkt eine Modellanpassung durchgeführt werden.

## 5.5 Active Appearance Models

Innerhalb des Teilsystems zur Emotionsschätzung werden wie bei der Kopfposenschätzung ebenfalls *Active Appearance Modells* eingesetzt. Zur Erläuterung der Grundlagen der AAMs wird daher an dieser Stelle auf Kapitel 4.4 verwiesen. Es gelten die gleichen Prinzipien der AAMs wie bei der Kopfposenschätzung. Es wird ebenfalls der echtzeitfähige *Project-Out-Algorithmus* eingesetzt. Jedoch gibt es Detailunterschiede, auf die im Folgenden kurz eingegangen werden soll.

### 5.5.1 Komplexität des Modells und Modellanpassung

Bei den in Kapitel 4 eingesetzten Active Appearance Modellen war es wichtig, dass diese die Blickrichtung (anhand der Kopfpose) repräsentieren können. Hierzu war es notwendig, die

durch die Bewegung des Kopfes entstehenden Änderungen der (äußeren) Form darzustellen. Die innere Form (z.B. Mund geöffnet oder geschlossen) spielen für die Blickrichtung keine Rolle. Diese mussten nur insoweit repräsentiert werden können, dass eine ggf. fehlende Komponente nicht zu einer fehlerhaften Anpassung des Gesamtbildes führt. Daher konnte ein Modell mit relativ wenig Komponenten verwendet werden.

Im Gegensatz dazu spielt bei der Mimikschätzung die Kopfpose nur eine untergeordnete Rolle. Stattdessen ist eine genaue Analyse der einzelnen Gesichtsbestandteile (wie Mund, Nase, Augen und Stirn) und der durch Muskelbewegungen entstehende Falten notwendig. Das Active Appearance Modell muss also vorrangig diese Komponenten (sowohl Form als auch Textur) darstellen können.

Im Folgenden wird davon ausgegangen, dass die Mimikschätzung nur bei einem frontal ausgerichteten Gesicht durchgeführt wird. Eine poseninvariante Mimikschätzung wie z.B. in [Fasel, 2002], [Kumano et al., 2009] oder [Rudovic and Pantic, 2010] ist für die anvisierte Schätzung des Interaktionsinteresse bzw. der Aufmerksamkeit nicht relevant. Sobald ein Nutzer nicht mehr auf den Bildschirm schaut, kann von einem Desinteresse oder von einer Ablenkung durch andere Dinge ausgegangen werden. Da die Kopfpose separat (siehe vorheriges Kapitel) erfasst wird, ist eine Analyse des Gesichtsausdrucks dann nicht mehr notwendig. Daher ist eine poseninvariante Mimikschätzung nicht Bestandteil dieser Dissertation.

Bei der Erstellung des Active Appearance Modells werden daher auch nur quasi frontal ausgerichtete Gesichter verwendet. Im Rahmen dieser Dissertation wird dazu die FEEDTUM-Gesichtsdatenbank [Wallhoff, 2006] eingesetzt. Details zu dieser Datenbank finden sich weiter hinten im Abschnitt 5.6.1.

Als Ergebnis entsteht ein Modell, dass im Vergleich zu Kopfposenschätzung aus wesentlich mehr Parametern besteht. Gleichzeitig ist eine genauere Modellanpassung notwendig, damit die Details im Gesicht gut erfasst werden und später zur Klassifikation genutzt werden können. Bei den durchgeführten Tests wurden Modelle mit typischerweise 10-25 Formparameter und 20-60 Texturparametern eingesetzt. Durch die größere Anzahl von Modellparametern entsteht aber auch ein komplexeres Fehlergebirge, dass die Modellanpassung erschweren kann (siehe dazu auch Abschnitt 4.4.3). Im Vergleich zur Kopfposenschätzung sind typischerweise auch mehr Iterationen pro Bild notwendig, um eine hinreichend genaue Modellanpassung zu erreichen. Bei den einfachen Modellen der Kopfposenschätzung waren teilweise weniger als 5 Iterationen notwendig, bei den komplexen Modellen zur Mimikschätzung sind dagegen oftmals pro neuem Bild mehr als 20 Iterationen für eine gute initiale Modellanpassung notwendig. Für nachfolgende Bilder ist die Anzahl der Iterationen jedoch wieder deutlich geringer, da auch hier die Parameter des vorhergehenden Bildes als Startschätzung verwendet werden können. Somit kann auch hier eine Echtzeitfähigkeit realisiert werden.

### 5.5.2 Verwendung unabhängiger Komponenten

Die im Kapitel 4.4 vorgestellten Standard - Active Appearance Modells verwenden sowohl beim Form- als auch beim Texturmodell eine Hauptkomponentenanalyse (*Principle Componten Analysis - PCA*), um lineare Abhängigkeiten zwischen den Labelvektoren zu beseitigen. Als Ergebnis entstehen jeweils ein Mittelwert (eine mittlere Gesichtsform und ein Mittelwertgesicht) und eine Reihe von Eigenvektoren, die die Varianz der zugrunde liegenden Daten repräsentieren. Im Rahmen der PCA werden die Eigenvektoren auch nach der Größe der zugehörigen Eigenwerte sortiert. Somit repräsentieren die ersten (größten) Eigenvektoren die Komponenten, die die größte Varianz in den Daten verursachen. Beim Formmodell handelt es sich dabei typischerweise um die Bewegung des Kopfes und beim Texturmodell um Beleuchtungseffekte und großflächige Änderungen im Gesicht.

Abbildung 5.10 zeigt beispielhaft vier Komponenten einer PCA auf dem Texturmodell. Die erste Zeile zeigt eine Variation der Beleuchtung. Zeile zwei zeigt ein Öffnen des Mundes verbunden mit dem Schließen der Augen. In der dritten Zeile ist ein Rümpfen der Nase und Öffnen des Mundes zu sehen. Die letzte Zeile zeigt ein Öffnen des Mundes und der Augen.
Ein Problem der PCA ist, dass die Bestimmung der Hauptkomponenten auf Basis der *second order information* erfolgt. Daher können lokale Merkmale kaum separat in den Hauptkomponenten herausgefiltert werden. Stattdessen werden eher größere Zusammenhänge (bzw. Komponenten) ermittelt. Bei der Bestimmung des Gesichtsausdrucks sind aber gerade einzelne Komponenten wichtig (z.B. Mund offen/zu, Augen zugekniffen/aufgerissen, etc.). Diese können mittels PCA nicht separat ermittelt werden. Zur Lösung dieses Problems wird in [Üzümcü et al., 2003] und [Zhan et al., 2008] vorgeschlagen, die *Independent Component Analysis (ICA)* anstatt der PCA zu verwenden. Es wurde gezeigt, dass hiermit eine bessere Segmentierung auf dem untersuchten Bildmaterial (Röntgen- und MRT-Aufnahmen) möglich ist, als mit der PCA. Auch in [Bartlett, 2001] und [Bartlett et al., 2002] wurde die ICA bereits erfolgreich zur Mimikschätzung (jedoch ohne Active Appearance Modells) eingesetzt. Es wurde gezeigt, dass die ICA beim Einsatz zur Gesichtsidentifikation Vorteile gegenüber der PCA hat und eine bessere Performance erreicht. Im Rahmen dieser Dissertation soll daher untersucht werden, wie sich die Verwendung der ICA beim Texturmodell auf die Eignung eines AAMs zur Mimikschätzung auswirkt.

Mit Hilfe der ICA können unabhängige *non-Gaussian* Komponenten bestimmt werden, indem neben den Momenten erster und zweiter Ordnung auch höhere statistische Momente über der Verteilung der Daten betrachtet werden. Zur Berechnung der ICA sind eine Reihe von Algorithmen bekannt. Dazu gehören die *FastICA* [Nyvärinen and Oja, 1997], *InfoMax* [Bell and Sejnowski, 1995] und *JADE* [Cardoso, 1999]. Im Rahmen dieser Dissertation wurde die FastICA eingesetzt. Abbildung 5.11 zeigt beispielhaft einige unabhängige Komponenten, die mittels ICA ermittelt wurden.

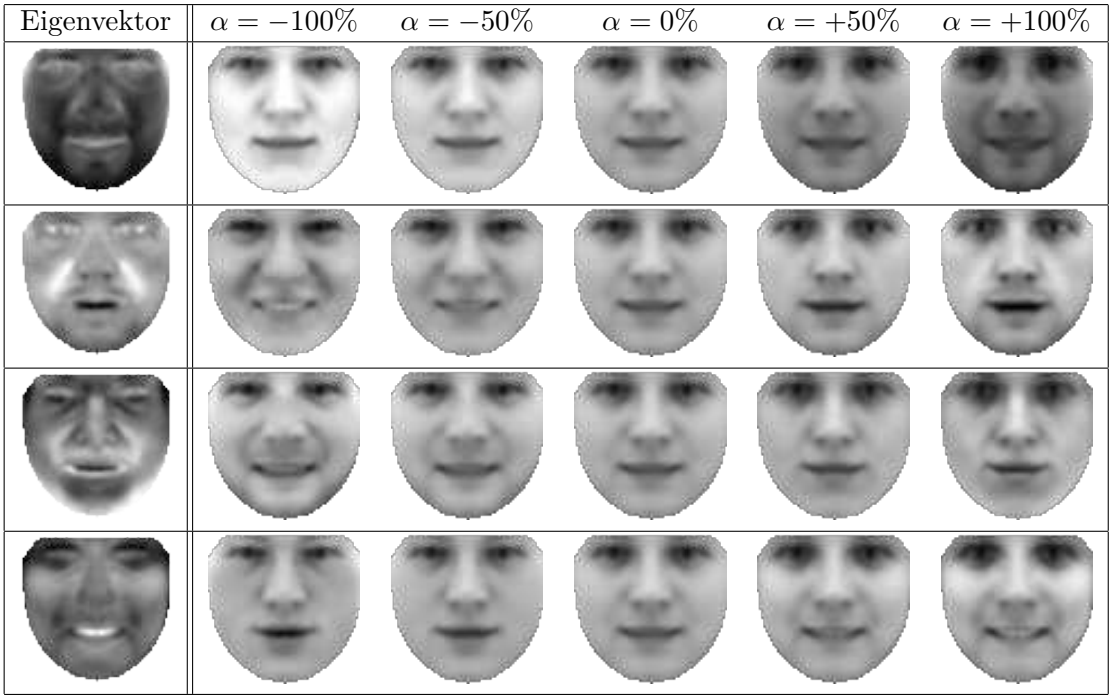



***Abbildung 5.10:*** *Beispiel für ein Texturmodell mittels PCA: In der ersten Spalte ist der jeweilige Eigenvektor der PCA als Bild dargestellt. Helle Bereiche stehen für positive Werte und dunkle Bereich repräsentieren negative Werte. Die nachfolgenden Spalten zeigen die Addition des Mittelwertgesichts und des Eigenvektors multipliziert mit einem Faktor $\alpha$. Bei $\alpha = 0\%$ ist jeweils das Mittelwertgesicht zu sehen.*

Ein Problem bei der Verwendung der ICA ist die Bestimmung einer Reihenfolge bzw. Ordnung der ermittelten Komponenten. Beim Einsatz der PCA ergibt sich diese auf Basis der Sortierung der zu den Eigenvektoren gehörigen Eigenwerte. Eine solche Sortierung der Komponenten ist nicht Bestandteil der ICA. Zur Ermittlung der relevanten Komponenten und einer anschließenden Dimensionsreduzierung des Modells ist diese jedoch zwingend notwendig. In [Üzümcü et al., 2003] und [Zhan et al., 2008] werden verschiedene Methoden zur Lösung dieses Problems vorgestellt:

- *Variance of the histogram*:
  Bei dieser Methode werden alle Elemente des Datensatzes auf jede der ICs (*Independent Component*) projiziert. Für jede IC wird anschließend ein Histogramm erstellt, in das eine Gauß-Kurve gefittet wird. Die ICs werden anschließend anhand der Varianzen der Gauß-Kurven sortiert. Eine IC mit kleiner Varianz beschreibt Rauschen und Ausreißer im Datensatz. Daher wird von groß nach klein sortiert.

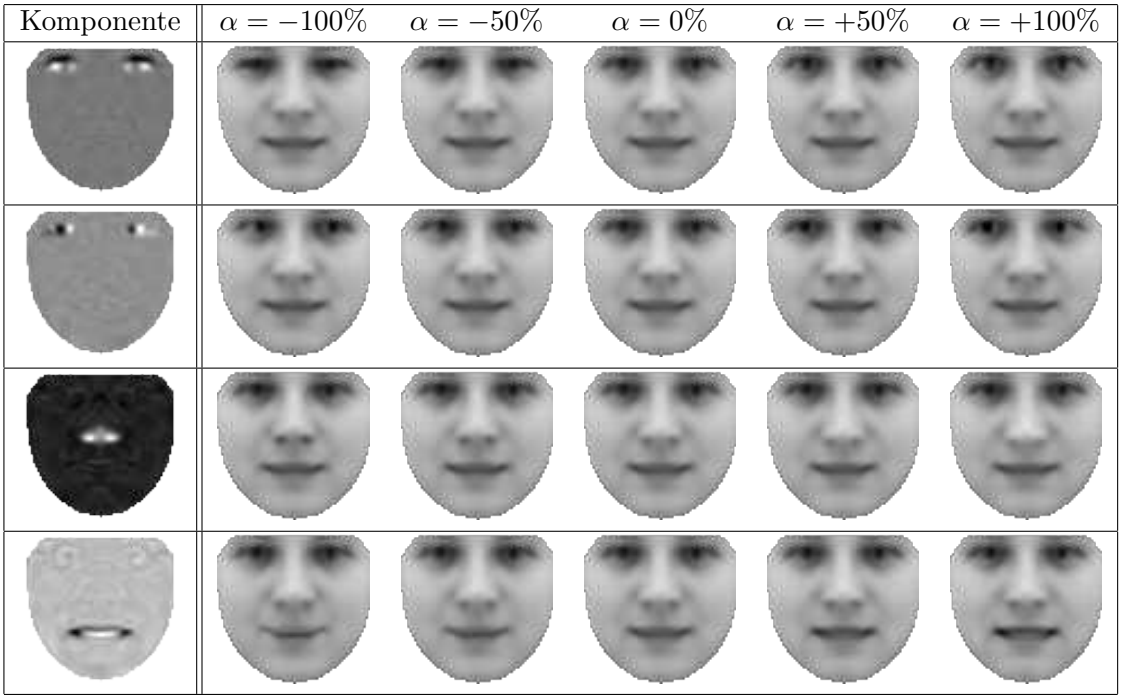



***Abbildung 5.11:*** *Beispiel für ein Texturemodell mittels ICA: In der ersten Spalte ist der jeweilige Eigenvektor der ICA als Bild dargestellt. Helle Bereiche stehen für positive Werte und dunkle Bereich repräsentieren negative Werte. Die nachfolgenden Spalten zeigen die Addition des Mittelwertgesichts und des Eigenvektors multipliziert mit einem Faktor $\alpha$. Bei $\alpha = 0\%$ ist jeweils das Mittelwertgesicht zu sehen.*

- *Alignment between data and ICs*:
  Die ICs spannen einen Raum zur Beschreibung des zugrunde liegenden Datensatzes auf. Jede IC steht für einen Richtungsvektor, entlang dessen die Projektion der Daten maximal nicht-gaußförmig ist. Für jede IC kann mittels Projektion ein mittlerer Winkel zwischen der Komponente und allen Datensätzen berechnet werden. Je kleiner dieser Winkel ist, desto stärker ist der Datensatz an dieser Komponente ausgerichtet. Die Sortierung erfolgt daher mit steigendem mittlerem Winkel.

- *Locality of Variation*:
  Dieser Ansatz versucht, die Komponenten anhand ihres (lokalen) Einflusses auf die Daten zu sortieren. Bezogen auf den Mittelwert zeigen bei Veränderung des Gewichtungsfaktors unterschiedliche Komponenten auch verschieden starken Einfluss. Unwichtige ICs erzeugen ein breit verteiltes Rauschen mit geringer Amplitude. Sehr wichtige ICs führen zu lokal sehr beschränkten Änderungen mit großer Amplitude. Durch Variation der Gewichte, kann der Einfluss auf den Mittelwert bestimmt werden. Komponenten mit lokal beschränkten Änderungen werden stärker gewichtet, als solche, die ein Rau-

schen erzeugen.

Im Rahmen dieser Dissertation wurde die Methode der *Locality of Variation* eingesetzt, da diese sich relativ einfach berechnen und implementieren lässt. Auf Basis der erstellten Sortierung der ICs kann anschließend auch die Dimensionsreduzierung vorgenommen werden, indem z.B. der gewünschte prozentuale Anteil der Gesamtmenge der ICs weiterverwendet wird.

Die Integration in den AAM-Anpassungsalgorithmus kann sehr einfach realisiert werden. Im Rahmen des *Project-Out-Algorthmus* werden die Formkomponenten wie bisher weiterhin mit einer PCA bestimmt. Die ICA wird nur für die Texturkomponenten verwendet. Anstatt einer PCA wird nun eine ICA (inkl. entsprechender Sortierung der Komponenten) mit einer anschließenden Orthogonalisierung (z.B. nach dem Gram-Schmidt-Verfahren, siehe Anhang D.4) ausgeführt. Da die eigentliche Bestimmung der gesuchten Texturparameter beim *Project-Out-Algorithmus* letztendlich nur eine einfache Matrixmultiplikation ist, müssen hier keine weiteren Änderungen vorgenommen werden.

## 5.6 Ergebnisse

Im folgenden Kapitel werden die erzielten Ergebnisse der Emotionsschätzung vorgestellt.

### 5.6.1 Datenbank zur Emotionsschätzung

Die Untersuchungen zur Mimikschätzung wurden auf Basis der FEEDTUM-Gesichtsdatenbank [Wallhoff, 2006] durchgeführt. Diese Datenbank enthält Videosequenzen von mehreren Personen, die jeweils eine der sechs Basisemotionen (siehe Abschnitt 5.2.1) zeigen.

Ein Hauptproblem bei der Erstellung einer Datenbank mit Gesichtsausdrücken ist das Phänomen, dass gespielte Emotionen sich von natürlichen Gesichtsausdrücken unterscheiden. Um diesen Effekt zu vermeiden bzw. so weit wie möglich zu reduzieren, wurde bei den Videoaufnahmen der FEEDTUM-Datenbank versucht, bei den Probanden natürliche Emotionen hervorzurufen. Dazu wurden den Probanden entsprechende Bilder gezeigt oder passende Video-Clips vorgespielt.

Die Datenbank enthält insgesamt Aufnahmen von 19 verschiedenen Personen. Für jede Person wurden jeweils drei Videos für die jeweils sechs Basisemotionen aufgezeichnet. Zusätzlich wurden Aufnahmen gemacht, in denen die Probanden keinen (bzw. einen neutralen) Gesichtsausdruck zeigen. Insgesamt stehen somit fast 400 Videoaufnahmen von wenigen Sekunden zur Verfügung. Jede Sequenz zeigt am Anfang einen neutralen Gesichtsausdruck, später eine der Basisemotionen und am Ende wieder eine neutrale Mimik. Zu jeder Videosequenz liegt

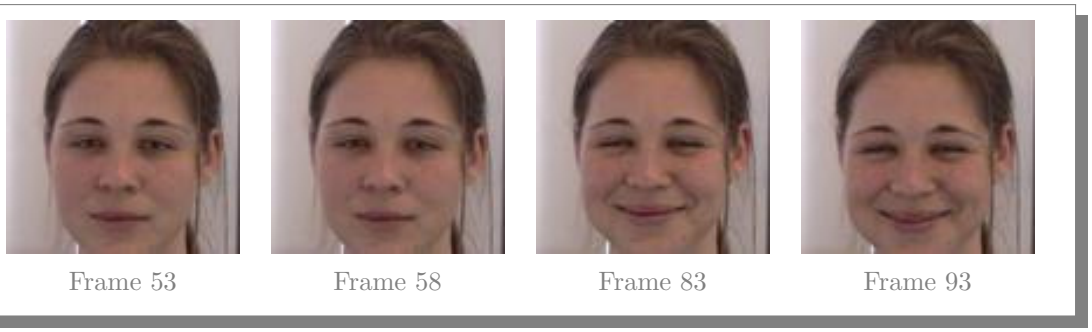



***Abbildung 5.12:*** *Bilder einer Beispielsequenz der FEEDTUM-Datenbank für die Basisemotion Freude.*

eine Beschreibung vor, in welchen Frames welche Zustände zu sehen sind. Abbildung 5.12 zeigt ein Beispiel für die Emotion Freude.

### 5.6.2 Anwendung auf diskrete Basisemotionen

Zunächst wurde die Emotionsschätzung bei der Verwendung von diskreten Basisemotionen untersucht. Dabei wurden die sechs Basisemotionen Ärger (*anger*), Ekel (*disgust*), Angst (*fear*), Freude (*happy*), Traurigkeit (*sadness*) und Überraschung (*surprise*) verwendet (siehe Abschnitt 5.2.1).

Die folgenden Ergebnisse wurden präsentiert in [Martin and Gross, 2008]. Hierbei wurde ein AAM basierend auf Kantenbildern (anstatt normalen Grauwertbildern) verwendet. Dort wurde gezeigt, dass hiermit eine bessere Modellanpassung gegenüber dem Standard *ProjectOut*-Algorithmus auf Grauwertbildern erreicht werden kann.

Die Experimente wurden auf der FEEDTUM-Datenbank (siehe Abschnitt 5.6.1) durchgeführt. Dabei wurden drei verschiedene Varianten der Klassifikation getestet:

- **AAM classifier set:** Hierbei wurde für jede Emotion ein eigenes AAM-Model trainiert. Dazu wurden die Trainingsdaten entsprechend der Labeldaten in die sechs Basisemotionen aufgeteilt. In der Kann-Phase wird eine Modellanpassung für alle sechs Modelle durchgeführt. Als Ergebnis wird die zu dem am besten angepassten Modell gehörige Emotionsklasse ausgegeben.

- **MLP-based classifier:** Bei dieser Variante wird ein *Multi-Layer-Perceptron (MLP)* eingesetzt. Als Input dienen im Rahmen dieser Auswertung die Form- und/oder Texturparameter des angepassten Modells. Als Output wird die ermittelte Emotionsklasse ausgegeben.

- **SVM-based classifier:** Als dritter Klassifikator wurde eine *Support Vector Machine (SVM)* anstatt eines MLP eingesetzt. Im Rahmen dieser Auswertung kommen ebenfalls die Form- und/oder Texturparameter des angepassten Modells zum Einsatz. Da es sich bei einer SVM nur um einen Zwei-Klassen-Klassifikator handelt, muss hier eine der Techniken zur Erweiterung auf mehrere Klassen eingesetzt werden. Auf Grund des Aufwands zum Training der Klassifikatoren wurde hier das *one-versus-one* Verfahren ausgewählt (siehe auch Abschnitt 3.6.7).

Bei den letzten beiden Varianten wurde zusätzlich untersucht, ob die ausschließliche Verwendung von Formparametern ausreicht, oder ob der Einsatz von Form- und Texturparametern das Ergebnis deutlich verbessert.

#### Ergebnisse beim *AAM classifier set*

Tabelle 5.2 zeigt die Ergebnisse aus [Martin and Gross, 2008] beim Einsatz emotionsspezifischer Modelle. Es ist deutlich zu sehen, dass nur *anger* sehr gut geschätzt werden konnte. Die anderen Emotionen dagegen weisen sehr schlechte Detektionsraten auf. Auch die *false-positive* Raten sind sehr hoch (insbesondere bei *anger*).

| | *anger* | *disgust* | *fear* | *happy* | *sadness* | *surprise* | FP |
|---|---|---|---|---|---|---|---|
| *anger* | 186 | 128 | 89 | 100 | 172 | 72 | 47.9% |
| *disgust* | 0 | 28 | 11 | 1 | 0 | 0 | 1.1% |
| *fear* | 1 | 25 | 70 | 14 | 21 | 26 | 7.8% |
| *happy* | 6 | 49 | 61 | 60 | 42 | 23 | 15.4% |
| *sadness* | 1 | 19 | 9 | 8 | 62 | 26 | 5.9% |
| *surprise* | 2 | 13 | 7 | 9 | 1 | 26 | 2.7% |
| Summe | 196 | 262 | 247 | 192 | 298 | 173 | |
| % | 94.9 | 10.7 | 28.3 | 31.3 | 20.8 | 15.0 | |

***Tabelle 5.2:*** *Ergebnisse der Klassifikation mit separaten AAMs: Die Tabelle zeigt die Detektionsrate (unteren Zeile) und die falsch-positiv Ergebnisse bei der Klassifikation mit mehreren AAMs. Als einzige Emotion kann anger robust geschätzt werden. Es treten außerdem hohe falsch-positiv Raten auf.*

Ein weiteres Problem dieser Variante ist, dass pro Bild jeweils sechs Modelle angepasst werden müssen. Damit ist ein deutlich höherer Rechenaufwand nötig. Unter Berücksichtigung der erzielten Ergebnisse und der Laufzeitproblematik wurde diese Variante im Rahmen dieser Dissertation daher nicht weiter untersucht.

#### Ergebnisse für *MLP* und *SVM* mit Formparametern

Beim MLP wurden zwei Hiddenschichten mit 15 bzw. 7 Neuronen eingesetzt. In der Ausgabeschicht wurden 7 Neuronen (Sechs Basisemotionen + Neutral) verwendet. Als

Ergebnis wurde jeweils das Neuron mit der größten Ausgabe gewählt. Das Training wurde mit dem Standard Backpropagation Algorithmus durchgeführt. Als Aktivierungsfunktion wurde der *tanh* verwendet. Bei Netzen mit nur einer Hiddenschicht zeigten sich schlechtere Ergebnisse. Eine größere Anzahl von Neuronen in beiden Hiddenschichten brachte keine deutliche Verbesserung, stattdessen tendieren solche Netze eher zum *over-fitting*.

Beim SVM wurde das *one-versus-one* Verfahren mit einem *Gauss*-Kernel verwendet. Als Input kamen wie beim MLP die Formparameter zum Einsatz. Bei Tests mit anderen Kerneln wurden keine besseren Ergebnisse erzielt.

| **MLP** | *anger* | *disgust* | *fear* | *happy* | *sadness* | *surprise* | *neutral* | FP |
|---|---|---|---|---|---|---|---|---|
| *anger* | 14 | 1 | 0 | 1 | 0 | 0 | 1 | 0.9% |
| *disgust* | 6 | 38 | 7 | 1 | 6 | 3 | 13 | 11.8% |
| *fear* | 10 | 1 | 40 | 1 | 4 | 2 | 15 | 10.9% |
| *happy* | 0 | 0 | 1 | 17 | 0 | 6 | 7 | 4.4% |
| *sadness* | 3 | 6 | 1 | 3 | 45 | 2 | 13 | 9.3% |
| *surprise* | 0 | 5 | 1 | 8 | 0 | 19 | 6 | 6.1% |
| *neutral* | 5 | 4 | 6 | 12 | 5 | 1 | 20 | 11.6% |
| Summe | 38 | 55 | 56 | 43 | 60 | 33 | 75 | |
| % | 37% | 69% | 71% | 40% | 75% | 58% | 27% | |

| **SVM** | *anger* | *disgust* | *fear* | *happy* | *sadness* | *surprise* | *neutral* | FP |
|---|---|---|---|---|---|---|---|---|
| *anger* | 32 | 0 | 1 | 3 | 0 | 1 | 2 | 1.9% |
| *disgust* | 1 | 51 | 1 | 1 | 1 | 1 | 6 | 3.0% |
| *fear* | 0 | 0 | 47 | 0 | 0 | 1 | 1 | 0.6% |
| *happy* | 1 | 2 | 0 | 31 | 2 | 2 | 2 | 2.5% |
| *sadness* | 2 | 0 | 3 | 4 | 53 | 0 | 19 | 7.7% |
| *surprise* | 2 | 2 | 2 | 4 | 4 | 27 | 2 | 4.4% |
| *neutral* | 0 | 0 | 2 | 0 | 0 | 1 | 43 | 0.8% |
| Summe | 38 | 55 | 56 | 43 | 60 | 33 | 75 | |
| % | 84% | 93% | 84% | 72% | 88% | 82% | 57% | |

***Tabelle 5.3:*** *Klassifikationsergebnisse bei $n = 10$ Formparametern: Die obere Tabelle zeigt die erzielten Detektionsergebnisse beim Einsatz eines MLP, die untere Tabelle die Ergebnisse bei einer SVM. In der unteren Zeile sind jeweils die Detektionsraten und in der rechten Spalte die false-postive (FP) Werte zu sehen. Die SVM-Klassifikation zeigt deutlich bessere Detektionsraten bei gleicher oder niedriger false-positve Rate.*

In Tabelle 5.3 sind die erreichten Ergebnisse bei ausschließlicher Verwendung von Formparametern dargestellt. Neben den sechs Basisemotionen wurde bei diesen Tests noch eine weitere Klasse *neutral* mit aufgenommen. Bei der Durchführung der Tests wurden nur die Datensätze verwendet, bei denen eine hinreichend genaue Modellanpassung vorgenommen

werden konnte. In diesem Test wurden $n = 10$ Formparameter eingesetzt. Dabei konnte bei 360 Bildern eine sehr gute Anpassung durchgeführt werden. Pro Klasse standen zwischen knapp 40 und 75 Datensätze zur Verfügung (siehe Tabelle 5.3). Tests mit mehr als $n = 10$ Formparametern zeigten ähnliche Resultate.

Beim MLP wurde eine mittlere Detektionsrate von 54% erreicht. Am besten konnte die Emotion *sadness* mit 75% und am schlechtesten der *neutral*-Zustand mit 27% klassifiziert werden. Bei der SVM liegt die mittlere Rate bei fast 80%. Sie schwankt zwischen 92% für *disgust* und 57% für *neutral*.

Gegenüber dem Einsatz verschiedener AAMs konnten somit hier deutlich bessere Detektionsraten für alle Basisemotionen erreicht werden. Auch die *false-positive* Rate ist deutlich geringer.

#### Ergebnisse für *MLP* und *SVM* mit Form- und Texturparametern

Als nächstes wurde untersucht, wie sich die Detektionsraten verändern, wenn sowohl Form- als auch Texturparameter zum Einsatz kommen. Hierbei wurde ein Modell mit $n = 10$ Form- und $m = 20$ Texturparametern verwendet.
Tabelle 5.4 zeigt die erzielten Ergebnisse bei Verwendung aller Form- und Texturparameter. Bei MLP konnte eine mittlere Klassifikationsrate von 75% erreicht werden. Diese schwankt zwischen 90% für *fear* und *sadness* und knapp unter 40% bei *neutral*. Die SVM erreichte eine mittlere Klassifikationsrate von 90%. Das beste Resultat wurde bei *fear* mit über 95% und das schlechteste bei *surprise* mit 78% erreicht.
Die erzielten Ergebnisse sind vergleichbar mit denen aus [Ratliff and Patterson, 2008], die eine abstandsbasierte Klassifikation auf einer leicht reduzierten Version der FEEDTUM Datenbank durchgeführt haben.

#### Reduktion der Modellparameter mittels *Mutual Information*

Wie in vorangegangenen Kapiteln wurde auch bei der diskreten Emotionsschätzung untersucht, wie sich eine Auswahl der relevanten Merkmale auf die Qualität der Schätzung auswirkt. Dazu wurde auch hier die *Mutual Information for Feature Selection (MIFS)* benutzt. Details zur MIFS finden sich in Anhang D.1.

Mit Hilfe der Berechnung der MIFS sind je nach Modell und vorhandener Form- und Texturparameter etwa 2-4 relevante Form- und 3-5 relevante Texturparameter ermittelt worden. Abbildung 5.13 zeigt einige der wichtigsten Modellparameter.

Durch die Reduktion der Modellparameter konnte das Training der Klassifikatoren deutlich vereinfacht werden. Beim MLP konnte das Netzwerk auf eine Hiddenschicht mit 10-15 Neu-

| **MLP** | *anger* | *disgust* | *fear* | *happy* | *sadness* | *surprise* | *neutral* | FP |
|---|---|---|---|---|---|---|---|---|
| *anger* | 38 | 1 | 0 | 3 | 1 | 0 | 0 | 1.6% |
| *disgust* | 3 | 46 | 0 | 1 | 1 | 3 | 3 | 3.7% |
| *fear* | 1 | 1 | 46 | 1 | 2 | 1 | 3 | 2.9% |
| *happy* | 1 | 4 | 1 | 34 | 2 | 6 | 2 | 5.1% |
| *sadness* | 2 | 3 | 0 | 3 | 79 | 0 | 1 | 3.3% |
| *surprise* | 1 | 4 | 3 | 6 | 1 | 39 | 2 | 5.5% |
| *neutral* | 0 | 0 | 1 | 0 | 2 | 1 | 7 | 1.2% |
| Summe | 46 | 59 | 51 | 48 | 88 | 50 | 18 | |
| % | 83% | 78% | 90% | 71% | 90% | 78% | 39% | |

| **SVM** | *anger* | *disgust* | *fear* | *happy* | *sadness* | *surprise* | *neutral* | FP |
|---|---|---|---|---|---|---|---|---|
| *anger* | 42 | 0 | 0 | 0 | 1 | 1 | 0 | 0.6% |
| *disgust* | 0 | 56 | 1 | 4 | 1 | 2 | 0 | 2.2% |
| *fear* | 0 | 0 | 49 | 0 | 4 | 5 | 1 | 2.8% |
| *happy* | 1 | 0 | 0 | 44 | 0 | 3 | 0 | 1.1% |
| *sadness* | 3 | 2 | 0 | 0 | 79 | 0 | 0 | 1.4% |
| *surprise* | 0 | 1 | 1 | 0 | 0 | 39 | 0 | 0.6% |
| *neutral* | 0 | 0 | 0 | 0 | 3 | 0 | 17 | 0.8% |
| Summe | 46 | 59 | 51 | 48 | 88 | 50 | 18 | |
| % | 91% | 95% | 96% | 92% | 90% | 78% | 94% | |

***Tabelle 5.4:*** *Klassifikationsergebnisse bei $n = 10$ Form- und $m = 20$ Texturparametern: Die obere Tabelle zeigt die Ergebnisse beim Einsatz eines MLP, die untere beim Einsatz einer SVM. In der unteren Zeile sind jeweils die Detektionsraten und in der rechten Spalte die false-postive (FP) Werte zu sehen. Die SVM-Klassifikation zeigt bessere Detektionsraten bei gleicher oder niedriger false-positve Rate.*

ronen reduziert werden. Damit konnten vergleichbare Erkennungsraten erreicht werden, wie bei einem MLP mit zwei Hiddenschichten (siehe vorheriger Abschnitt). Bei der SVM konnte die Trainingszeit stark reduziert werden. Die Ergebnisse selbst sind aber auch hier quasi unverändert geblieben.

#### Zusammenfassung diskrete Emotionsschätzung

Insgesamt kann also festgestellt werden, dass mit einer SVM die besten Klassifikationsergebnisse für eine diskrete Emotionsschätzung erreicht werden konnten. Der Einsatz eines MLP führt zu schlechteren, aber trotzdem noch guten Ergebnissen. Die Verwendung von AAMs für die einzelnen Basisemotionen hat am schlechtesten abgeschnitten. Mit Hilfe der *Mutual Information for Feature Selection (MIFS)* wurden die relevanten Modellparameter ermittelt. Hierdurch konnte dann das Training der Klassifikatoren vereinfacht werden, die erzielten Ergebnisse sind jedoch fast unverändert geblieben.

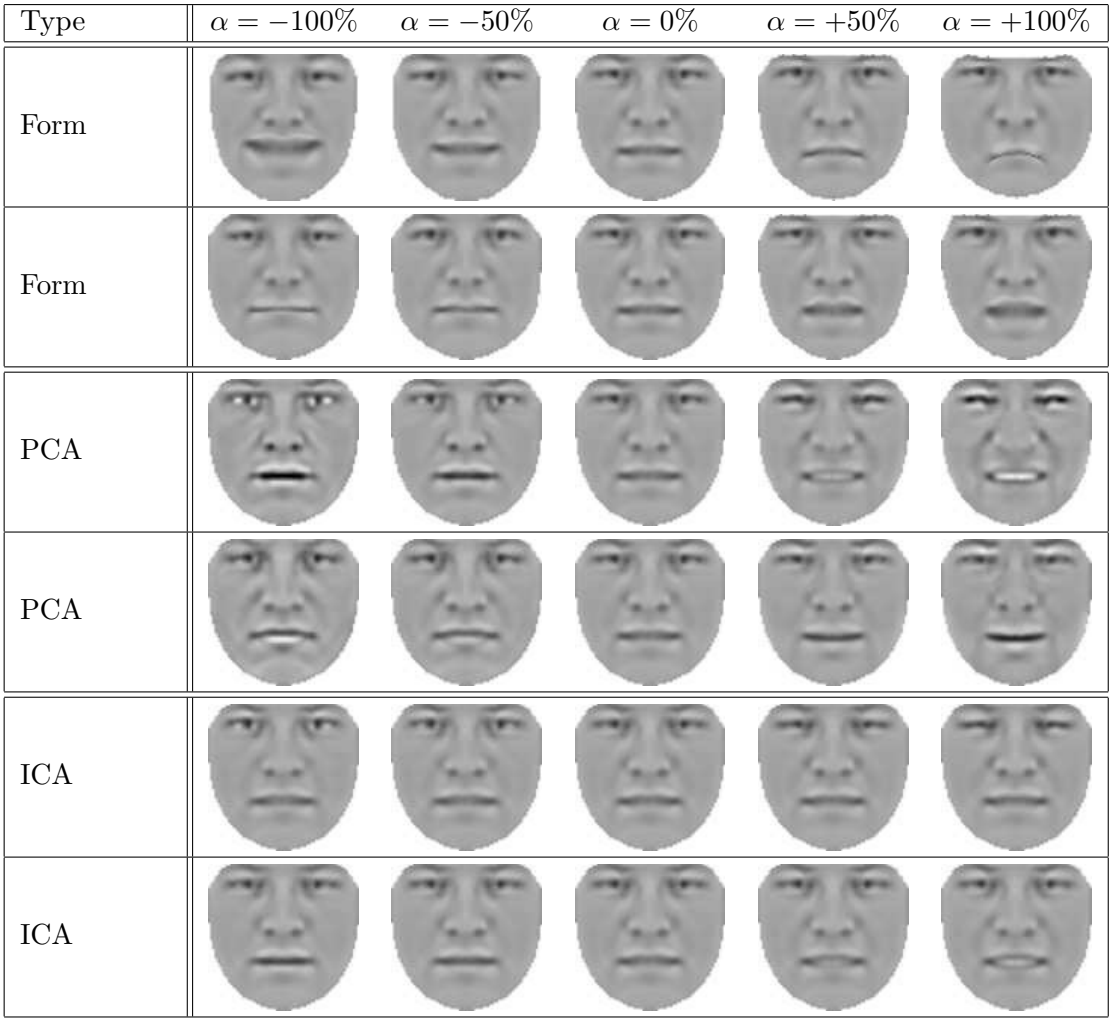



***Abbildung 5.13:*** *Beispiele für relevante Modellparameter zur Mimikschätzung: Die ersten beiden Zeilen zeigen die Wirkung zweier relevanter Formparameter. Die nachfolgenden Zeilen entsprechend jeweils zwei Texturparameter bei Nutzung von PCA bzw. ICA.*

Das bessere Abschneiden der SVM kann mit der besseren Separationsfähigkeit in einem hochdimensionalen Raum durch die eingepassten Hyperebenen erklärt werden. Bezüglich der Laufzeitanforderungen für ein MLP und eine SVM gibt es kaum Unterschiede, da der Anteil der benötigten Rechenzeit im Vergleich zur Modellanpassung fast vernachlässigt werden kann.

Die erzielten Ergebnisse liegen in der gleichen Größenordnung wie in [Ratliff and Patterson, 2008], die mit der gleichen Datenbank gearbeitet haben.

### 5.6.3 Ergebnisse im kontinuierlichen Emotionsraum

Zur Repräsentation des kontinuierlichen Emotionsraumes wurde ein Modell bzw. eine Struktur benötigt, die einen hochdimensionalen (Parameter-)Raum in niedrigdimensionale Zusammenhänge (die Achsen des Emotionsraums) abbilden kann. Im Rahmen dieser Dissertation wurden dazu selbstorganisierende Karten (*Self-organizing maps (SOM)* oder *Kohonen Maps*) verwendet. Dabei handelt es sich um ein unüberwachtes Lernverfahren aus dem Bereich der Neuronalen Netze. Sie können als eine topologieerhaltende Weiterentwicklung der Verfahren zur Vektorquantisierung (z.B. LVQ) betrachtet werden. Erstmals wurden diese von [Kohonen, 1982] vorgestellt. Weitere Details zu Kohonen Maps finden sich im Anhang A.4.

#### Varianten der Kohonen Maps

Es wurden zwei verschiedene Varianten von Kohonen Maps zur Abbildung des kontinuierlichen Emotionsraumes untersucht:

- *Standard 2D Kohonen Map*:
  Hierbei handelt es sich um eine selbstorganisierende Karte, bei der die Kohonen Neuronen in einem zweidimensionalen kartesischen Gitter liegen. Jede Position eines Neurons im Gitter wird beschrieben durch:

  $$\mathbf{p}_i = (x_i, y_i) \tag{5.1}$$

  Als Nachbarschaftsfunktion kommt die Standard Gauß-Funktion zum Einsatz:

  $$\mathrm{h}_{\mathrm{gauss}}(z, d) = \mathrm{e}^{(-z/d)^2} \quad \text{mit} \quad z = \|\mathbf{p}_{best} - \mathbf{p}_i\| \tag{5.2}$$

  Wobei $\mathbf{p}_{best}$ die Gitterposition des *Best-Matching-Neurons* und $d$ die Größe des zeitlich veränderlichen Nachbarschaftsradius innerhalb des Gitters ist.

- *Polar Kohonen Map*:
  Da, wie im Abschnitt 5.2.2 beschrieben wurde, aus psychophysiologischer Sicht der Emotionsraum als kreisförmiges Modell beschrieben werden kann, wurde im Rahmen dieser Dissertation eine Variante einer Kohonen Map entwickelt, bei der die Neuronen in einer ringförmigen Struktur in einem Polarkoordinatensystem plaziert sind. Hierzu wird die Position jedes Neurons beschrieben durch einen Winkel $\phi$ und einen Radius $r$:

  $$\mathbf{p}_i = (r_i, \phi_i) \tag{5.3}$$

Die Nachbarschaftsfunktion wurde wie folgt modifiziert:

$$\begin{aligned} \mathrm{h}_{\mathrm{gauss}}(z,d) = \mathrm{e}^{(-z/d)^2} \quad \text{mit} \qquad z &= \sqrt{\Delta r_i{}^2 + \Delta \phi_i{}^2} \\ \Delta r_i &= s_r \cdot |r_i - r_{best}| \quad \text{mit} \;\; s_r = 1/r_{max} \\ \Delta \phi_i &= s_\phi \cdot |\phi_i - \phi_{best}| \quad \text{mit} \;\; s_\phi = 1/\pi \end{aligned} \tag{5.4}$$

Wobei $r_{best}$ und $\phi_{best}$ den Radius und Winkel der Position des *Best-Matching-Neurons* angeben. Die Faktoren $s_r$ und $s_\phi$ stellen sicher, dass $0 \leq \Delta r_i \leq 1$ und $0 \leq \Delta \phi_i \leq 1$. Der zeitlich veränderliche Nachbarschaftsradius $d$ sollte bei dieser Variante immer zwischen 0 und 1 liegen.

Abbildung 5.14 zeigt die resultierende Gitterstruktur und den typischen Verlauf einer gaußförmigen Nachbarschaftsfunktion beider Typen im Vergleich.

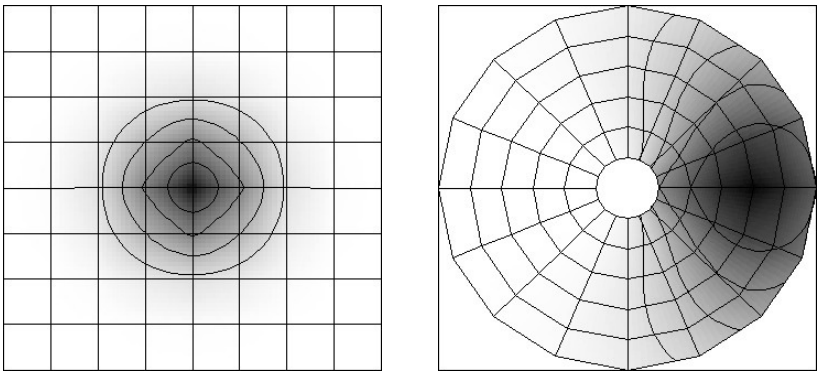


***Abbildung 5.14:*** *Gitterstruktur und Nachbarschaftsfunktion der Kohonen Maps: Links ist eine Standard-2D-Kohonen-Map der Größe 9x9 abgebildet. Rechts eine Polar-Kohonen-Map der Größe 6x16. In beiden Grafiken ist der Verlauf einer gaussförmigen Nachbarschaftsfunktion inkl. entsprechender Höhenlinien dargestellt.*

### Training der Kohonen Maps

Zum Training der Kohonen Maps wurden auch hier die Testbilder der FEEDTUM-Gesichtsdatenbank [Wallhoff, 2006] (siehe Abschnitt 5.6.1) eingesetzt. Hierzu wurden die Datensätze ausgewählt, auf denen eine Anpassung des Active Appearance Modells mit möglichst geringem Fehler erreicht werden konnte. Damit wird sichergestellt, dass die Trainingsdaten für die Kohonen Maps nur sinnvolle Parametersätze enthalten.

Aus den knapp 400 verfügbaren gelabelten Trainingsbildern aus der FEEDTUM-Datenbank konnten je nach Active Appearance Modells und gewählter maximaler Fehlerschwelle zwischen 270 und 350 Bilder zum Training verwendet werden. Diese waren über alle

Emotionen in etwa gleichverteilt.

Zusätzlich wurden, wie bei der diskreten Emotionsschätzung, nur die relevanten Modellparameter mit Hilfe der *Mutual Information* für das Training ausgewählt. Auf Grund der Notwendigkeit einer möglichst genauen Modellanpassung zur Bestimmung des Gesichtsausdrucks, wurde beim Erstellen der Modelle eine Abdeckung von 95% für das Form- und Texturmodell gewählt. Daher ergibt sich eine große Anzahl von Modellparametern, die aber auch eine Vielzahl von Redundanzen enthalten. Tabelle 5.5 zeigt die erreichte Reduktion der Modellparameter für ein Active Appearance Modell bei Verwendung der PCA bzgl. ICA für das Texturmodell bei Verwendung der *Mutual Information for Feature Selection (MIFS)*. Details zur MIFS finden sich in Anhang D.1.

| Variante | Anzahl Formparameter | Anzahl Texturparameter | Anzahl Parameter nach MIFS |
|---|---|---|---|
| PCA | 22 | 57 | 7 |
| ICA | 22 | 20 | 6 |

***Tabelle 5.5:*** *Reduktion der Modellparameter mittels Mutual Information zum Training der Kohonen Maps basierend auf einem Modell mit 95% Abdeckung von Form- und Textur.*

Beim Training der Kohonen Maps wurden die Trainingsdaten zufällig permutiert dem Netz präsentiert. Es wurden 100.000 Trainingsschritte durchgeführt. Beim Standard-Kohonen-Netz wurde ein 9x9 Gitter und beim Polar-Kohonen-Netz ein 6x16 (6 Radien, 16 Winkel) verwendet.

### Clusterbildung in den Kohonen Maps

Nach dem Training der Kohonen Maps wurde untersucht, wie sich die verwendeten Trainingsdaten im Netz wiederfinden. Dazu wurde für jeden Trainingsdatensatz das entsprechende Best-Matching-Neuron bestimmt und ein Histogramm der Aktivierungen für jede der Emotionsklassen gebildet. Die Abbildungen 5.15 und 5.16 zeigen die Ergebnisse für ein Standard-Kohonen-Netz und ein Polar-Kohonen-Netz. In beiden Fällen handelt es sich um die Ergebnisse bei Verwendung eines Active Appearance Modells mit einem Texturmodell basierend auf einer PCA. Die Grauwerte in den beiden Grafiken geben an, wie oft das Neuron an der entsprechenden Gitterstelle als Best-Matching-Neuron für die jeweilige Emotionsklasse aktiviert wurde. Je dunkler, desto häufiger wurde die entsprechende Gitterstelle aktiviert. Die Grauwerte zwischen den Stützstellen ergeben sich durch Interpolation und dienen nur der besseren Visualisierung der Clusterung.

In beiden Fällen ist je nach Emotion eine deutliche Clusterbildung zu erkennen. Insbesondere

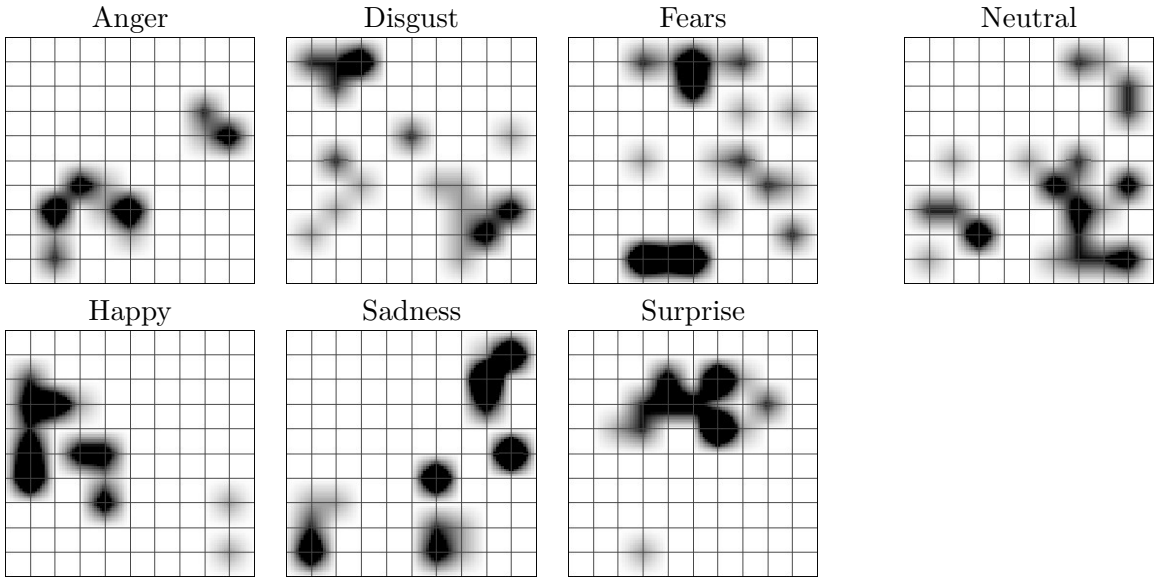



***Abbildung 5.15:*** *Clusterbildung mit einer Standard-Kohonenkarte der Parameter eines PCA-Active-Appearance-Modells. Für jede Basisemotion ist ein Histogramm der Best-Matching-Neuronen für die verwendeten Trainingsdaten abgebildet.*

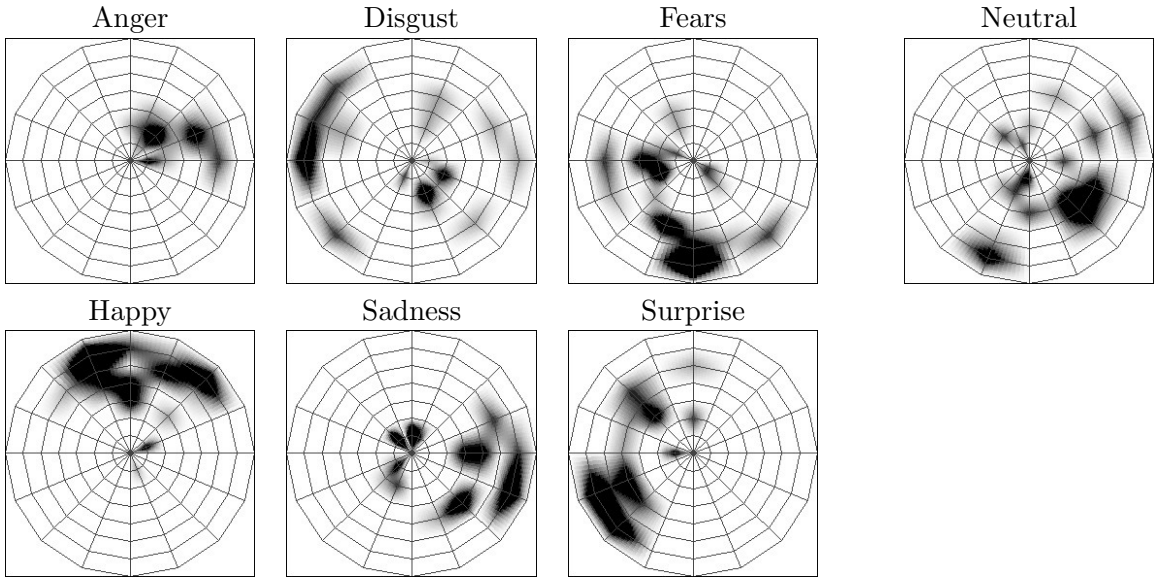



***Abbildung 5.16:*** *Clusterbildung mit einer Polar-Kohonenkarte der Parameter eines PCA-Active-Appearance-Modells. Für jede Basisemotion ist ein Histogramm der Best-Matching-Neuronen für die verwendeten Trainingsdaten abgebildet.*

bei den Emotionen *Happy*, *Sadness* und *Surprise* ist dies sichtbar. Dagegen sind z.B. *Neutral* und *Fears* relativ breit verteilt.

Beim Training hat sich gezeigt, dass die Clusterbildung je nach Durchlauf unterschiedlich gut erfolgt. Insbesondere der Neutral-Zustand hat sich nur sehr selten als ein Cluster herausgebildet. Typischerweise fand hier eher eine unterschiedlich starke Streuung statt.

Um eine stärkere Clusterbildung zu begünstigen, wurde eine Änderung am Algorithmus des Trainings der Kohonenkarte vorgenommen: Bei der Bestimmung des Best-Matching-Neurons $b$ zu einem Datensatz $\mathbf{x}$ mit bekannter Emotion $c(\mathbf{x})$ werden für *Neutral* nur die Neuronen im Zentrum der Karte berücksichtigt. Für die *Nicht-Neutral* Trainingsdaten werden nur Neuronen im äußeren Bereich der Karte betrachtet. Damit wird die Einbettung der Kohonenkarte im Inputraum gezielt so beeinflusst, dass die Klasse *Neutral* vorrangig in den inneren Bereich der Karte abgebildet wird. Die Bestimmung des Best-Matching-Neurons kann dann wie folgt beschrieben werden:

$$
\begin{aligned}
b &= \arg\min_i \left(f \cdot \|\mathbf{x} - \mathbf{w}_i\|\right) \\
f &= \begin{cases} 1.0 & \text{wenn } \; (c(\mathbf{x}) = Neutral) \wedge (r_i \leq r_{neutral}) \\ 1.0 & \text{wenn } \; (c(\mathbf{x}) \neq Neutral) \wedge (r_i > r_{neutral}) \\ \infty & \text{sonst} \end{cases}
\end{aligned}
\tag{5.5}
$$

Wobei $c(\mathbf{x})$ die bekannte Emotionsklasse des Inputs $\mathbf{x}$ und $r_{neutral}$ den festgelegten maximalen Radius für die Klasse *Neutral* in der polaren Kohonenkarte beschreiben.

Die beschriebene Modifikation beeinflusst nur die Lage der Neuronen der Klasse *Neutral*. Für die restlichen Trainingsdaten bleiben die Eigenschaften der Kohonenkarte als topologieerhaltende Abbildung unverändert bestehen. Die eigentlichen Emotionen werden sich also um den Zustand *Neutral* herum in den äußeren Bereichen der Kohonenkarte ausbilden.

Abbildung 5.17 zeigt ein Beispiel der veränderten Clusterung. Im Vergleich zu den Abbildungen 5.15 und 5.16 ist eine bessere Herausbildung der Cluster der einzelnen Emotionen zu erkennen. Es ist auch ersichtlich, dass beispielsweise die Emotionen *Happy* und *Surprise* sich teilweise überlappen. Die Klasse *Fears* ist teilweise mit der Klasse *Disgust* und teilweise mit der Klasse *Surprise* überlappt. Dies kann durch starke Ähnlichkeiten dieser Emotionen erklärt werden.

Vergleicht man die Ergebnisse aus Abbildung 5.17 mit dem kontinuierlichen Emotionsmodell aus Abschnitt 5.2.2 kann die Ost-West-Achse als Positiv-Negativ-Achse (Positive=West und Negativ=Ost) und die Nord-Süd-Achse als Passiv-Aktiv-Achse (Passiv=Nord, Aktiv=Süd) des 2D-Emotionsraumes angesehen werden. Jedoch kann keine optimale Deckung zwischen

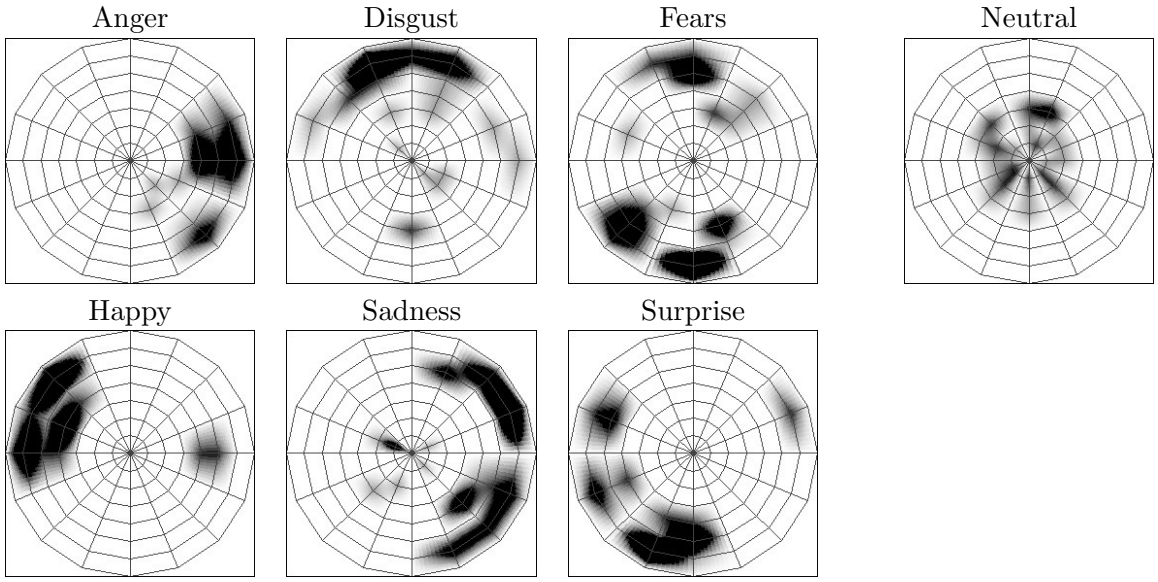



***Abbildung 5.17:*** *Clusterbildung mit einer Polar-Kohonenkarte mit Neutral im Zentrum der Parameter eines PCA-Active-Appearance-Modells. Für jede Basisemotion ist ein Histogramm der Best-Matching-Neuronen für die verwendeten Trainingsdaten abgebildet.*

den Emotionen in der Kohonenkarte und dem 2D-Emotionsraum erreicht werden. Trotzdem sind eindeutig positive Emotionen (von Süd bis Nord-West) von eher negativen (von Nord über Ost nach Süd) unterscheidbar. Die generelle Struktur des 2D-Emotionsraumen aus [Russell, 1980] bzw. [Scherer, 2005] konnte mit der Kohonen Karte somit nachgebildet werden.

#### Tests auf Videosequenzen

Mit Hilfe der im vorherigen Abschnitt vorgestellten Modellierung des kontinuierlichen Emotionsraumes auf Basis von Kohonen Maps wurden Tests auf den Videosequenzen der FEEDTUM-Datenbank [Wallhoff, 2006] durchgeführt. Für die vorliegenden Videosequenzen wurden folgende Schritte für jedes Bild durchgeführt:

1. Active Appearance Model auf Bild $I_t$ anpassen: Falls ein gültiges Modell aus dem Zeitschritt $t-1$ vorliegt, wird das letzte Modell als Basis für die Anpassung des neuen Inputs verwendet. Falls kein gültiges Modell vorliegt, wird zuerst eine Gesichtsdetektion und anschließend eine Modellanpassung vorgenommen (siehe auch Abschnitt 4.5.2).

2. Auswahl der relevanten Parameter: Aus den vorliegenden Modellparametern werden für die nachfolgende Klassifikation nur die gemäß *Mutual Information* ausgewählten relevanten Parameter verwendet (siehe vorn).

3. Suche nach dem Best-Matching-Neuron der Kohonen Map

Die resultierende Sequenz der Best-Matching-Neuronen kann als Trajektorie in der Kohonen Map visualisiert werden. Zwecks Unterdrückung von Ausreißern, wurde zusätzlich eine zeitliche Glättung vorgenommen. Die Abbildungen 5.18 bis 5.20 zeigen drei repräsentative Beispiele aus der FEEDTUM-Datenbank. Die erzielten Ergebnisse wurden mit den *Ground-Truth* Informationen der Datenbank qualitativ verglichen.

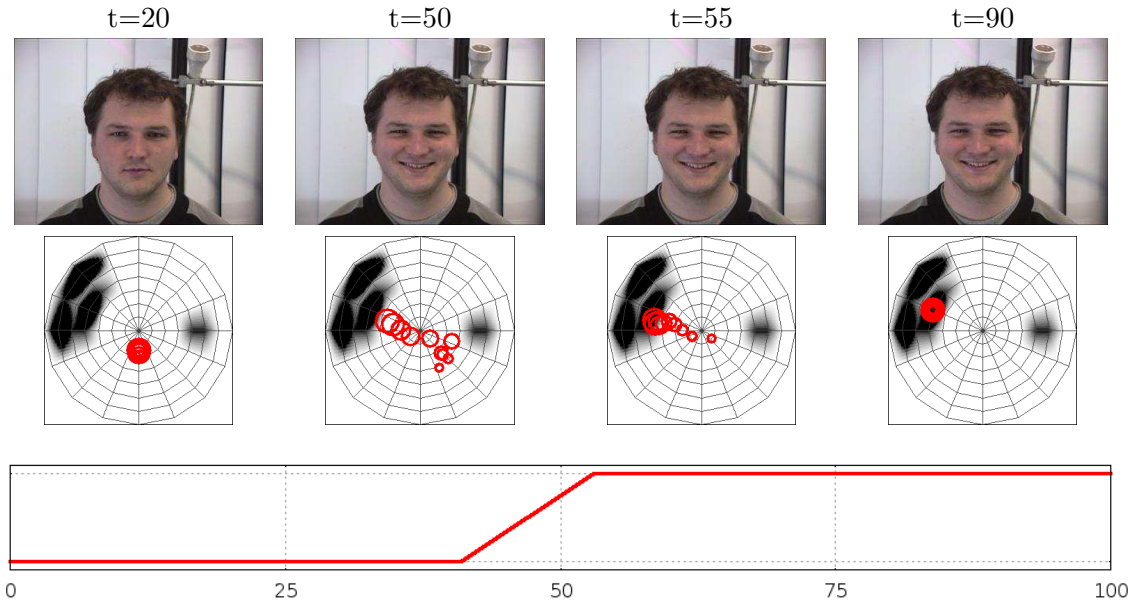



***Abbildung 5.18:*** *Beispiel für Emotionsschätzung bei Happy: Die obere Reihe zeigt ausgewählte Bilder der Sequenz. Die mittlere Reihe zeigt den Verlauf der geschätzten Emotion als Trajektorie über die letzten 10 Frames. Die untere Reihe zeigt die Aktivierung der Emotion laut Datenbank.*

Abbildung 5.18 zeigt eine Videosequenz der Kategorie *Happy*. Es ist zu sehen, dass der emotionale Zustand zunächst korrekt als Neutral in der Mitte der Kohonen Map geschätzt wird. Zum Zeitpunkt $t = 40$ beginnt der Proband zu lächeln und zum Zeitpunkt $t = 50$ ist der zu Zustand *Happy* erreicht. Mit einer (leichten durch den Tiefpass bedingten Verzögerung) verschiebt sich auch der geschätzte Zustand in den linken Bereich der Kohonen Map, der die Emotion *Happy* repräsentiert. Bis zum Ende der Sequenz verbleibt die Schätzung korrekt in diesem Bereich.
Analog ist der Verlauf der Schätzung für die Emotion *Disgust* in Abbildung 5.19 dargestellt. Nachdem der emotionale Zustand am Anfang wieder korrekt als *Neutral* in der Mitte geschätzt wurde, wird ab $t = 50$ die Emotion *Disgust* im oberen Teil der Kohonen Map erkannt.
Abbildung 5.20 zeigt einen beispielhaften Verlauf für die Emotion *Sadness*. Auch hier ist zunächst die korrekte Schätzung als *Neutral* und später als *Sadness* zu erkennen. Anzumerken ist bei diesem Beispiel, dass die Ausprägung der Emotion bei $t = 125$ relativ

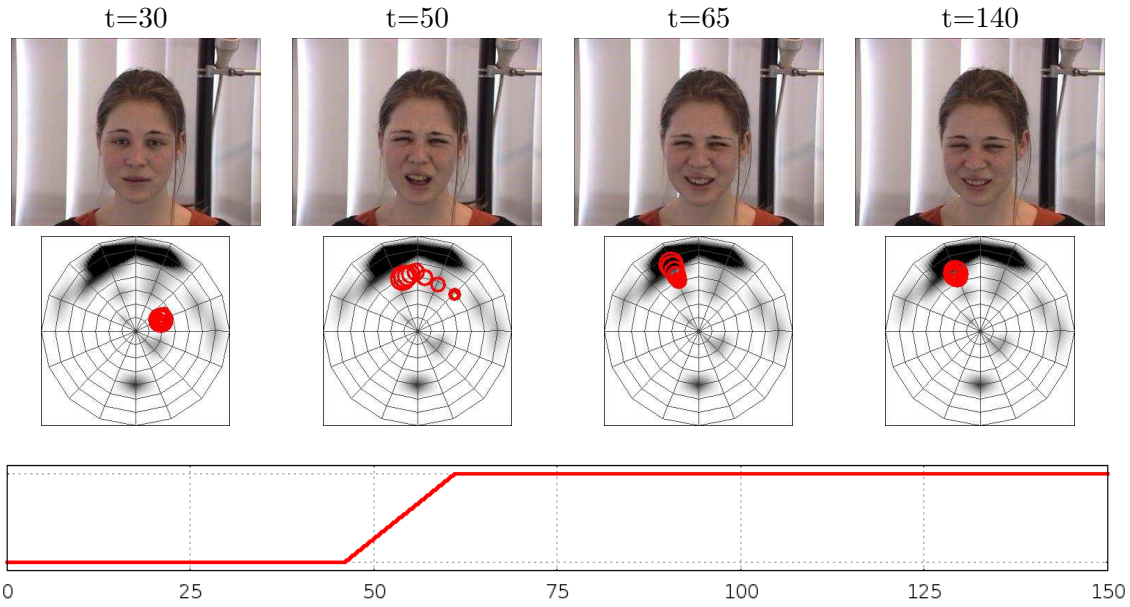



***Abbildung 5.19:*** *Beispiel für Emotionsschätzung bei Disgust: Die obere Reihe zeigt ausgewählte Bilder der Sequenz. Die mittlere Reihe zeigt den Verlauf der geschätzten Emotion als Trajektorie über die letzten 10 Frames. Die untere Reihe zeigt die Aktivierung der Emotion laut Datenbank.*

schwach ist und daher die Schätzung in der Kohonen Map wieder eher *Neutral* ist.

Bei der Durchführung weiterer Tests hat sich gezeigt, dass insbesondere sehr schwach ausgeprägte Emotionen nur relativ unsicher mit Hilfe der Kohonen Map klassifiziert werden können. In diesen Fällen springt das Best-Matching-Neuron mehr oder weniger zufällig um den *Neutral*-Zustand in der Mitte. Erst bei deutlichen Emotionen wird ein Best-Matching-Neuron im äußeren Teil der Kohonen Map aktiviert. Somit kann der Radius des Best-Matching-Neurons als Maß für die Sicherheit der Schätzung verwendet werden. Zusätzlich hat sich gezeigt, dass eine zeitliche Tiefpass-Filterung notwendig ist, um Ausreißer (z.B. durch eine nicht optimale Modellanpassung) zu unterdrücken.

### Resultate bei Verwendung der ICA

Im Abschnitt 5.5.2 wurde erläutert, wie die ICA anstatt der PCA in den AAM-Anpassungsalgorithmus integriert werden kann. Es wurde gezeigt, dass sich bei Verwendung der ICA lokal sehr begrenzte und gut interpretierbare Komponenten im Gesicht (z.B. Augen, Nase oder Mund) ermittelt werden.

Durchgeführte Experimente mit der ICA haben jedoch gezeigt, dass gegenüber der PCA

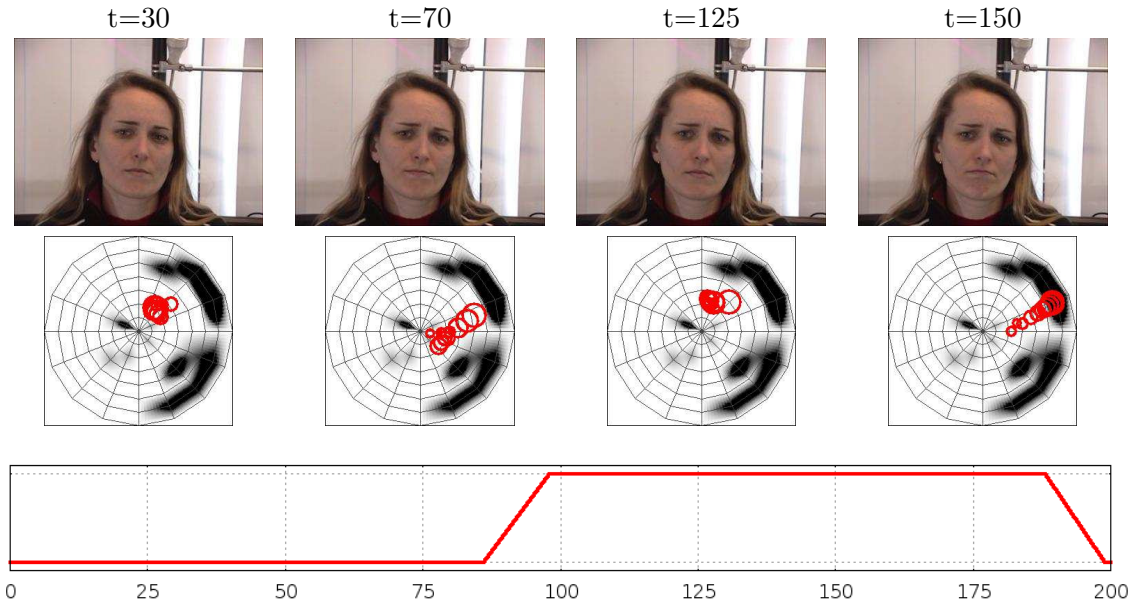



***Abbildung 5.20:*** *Beispiel für Emotionsschätzung bei Sadness: Die obere Reihe zeigt ausgewählte Bilder der Sequenz. Die mittlere Reihe zeigt den Verlauf der geschätzten Emotion als Trajektorie über die letzten 10 Frames. Die untere Reihe zeigt die Aktivierung der Emotion laut Datenbank.*

keine wesentliche Verbesserung erreicht werden kann. Wie in Tabelle 5.5 gezeigt wurde, kann die Größe des Inputraums von 7 Dimensionen bei der PCA auf 6 bei der ICA reduziert werden. Die eigentlichen Ergebnisse bleiben jedoch quasi gleich. Sowohl die Clusterbildung innerhalb der Kohonen Maps, als auch die Tests auf Videosequenzen zeigen im Vergleich zur PCA kaum Unterschiede. Je nach Trainingsdurchlauf kann sowohl die PCA als auch die ICA geringfügig bessere oder schlechtere Ergebnisse liefern.

Damit bleibt festzustellen, dass mit Hilfe der ICA zwar auch eine gute Modellanpassung vorgenommen werden kann, aber die eigentlichen Erkennungsraten damit nicht verbessert werden können.

## 5.7 Zusammenfassung

In den ersten beiden Kapiteln dieser Dissertation wurden zwei Systeme zur Bestimmung der Oberkörperpose und der Kopfpose präsentiert. Das erste kann bereits vor dem eigentlichen Start des Dialogs und das zweite kann vor dem Start und während des Dialogs genutzt werden. Ein weiteres wichtiges Merkmal, dass während des Dialogs verwendet werden kann, ist der emotionale Zustand des Benutzers.

In diesem Kapitel wurde ein System zur Schätzung des emotionalen Zustands des Benutzers auf Basis des Gesichtsausdrucks vorgestellt. Wie in Kapitel 4 wird hierzu zunächst das Gesicht im Bild detektiert und anschließend ein parametrisches Modell an das Gesicht angepasst. Die resultierenden Modellparameter dienen dann als Grundlage zur Bestimmung des emotionalen Zustands.

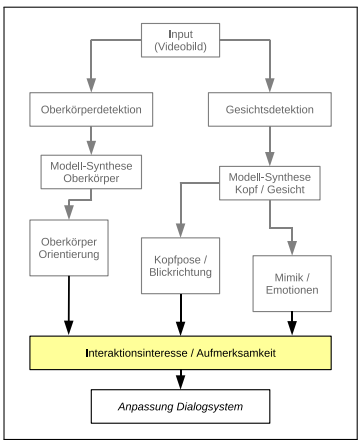
Zur Gesichtsdetektion kommt wieder der *Viola & Jones* - Gesichtsdetektor zum Einsatz. Die eigentliche Modellanpassung erfolgt wie im vorherigen Kapitel mit einem *Active Appearance Models*. Da die beiden Teilsysteme *HeadPoseTracker* und *MimicDetector* also nach den gleichen Verfahren arbeiten, können diese beiden sehr leicht miteinander kombiniert werden. Der Output des *HeadPoseTrackers* kann unmittelbar als Initialisierung für ein detailliertes *Active Appearance Model* zur Mimikschätzung genutzt werden. Als Zwischenergebnis entsteht ein hochdimensionaler Merkmalsvektor, der wieder mit Hilfe der *Mutual Information* auf die relevanten Elemente reduziert werden kann.

Die Schätzung des emotionalen Zustands kann entweder mit Hilfe einer Reihe von diskreten Basisemotionen oder in einem kontinuierlichen Emotionsraum erfolgen. Im Rahmen dieser Dissertation wurde ein zweidimensionaler Zustandsraum gewählt, da in diesem auch nicht so stark ausgeprägte Emotionen gut repräsentiert werden können und sich bei zeitlichen Betrachtungen auch Trajektorien im emotionalen Zustandsraum bilden lassen.

Zur Abbildung der Modellparameter des *Active Appearance Models* in den kontinuierlichen Emotionsraum wurde eine *Kohonen Map* als eine selbstorganisierende und topologieerhaltende Abbildung gewählt. Es konnte gezeigt werden, dass sich die relative Lage von bekannten Emotionsklassen im Emotionsraum auch in der relativen Lage von entsprechenden Regionen in der *Kohonen Map* wiederfindet.

Das vorgestellte System ist somit in der Lage, auf Basis des Gesichtsausdrucks den emotionalen Zustand in Form der Aktivierung von entsprechenden Neuronen in der *Kohonen Map* zu schätzen. Somit steht ein drittes Teilsystem zur Verfügung, dass zur Schätzung von Interaktionsinteresse und Aufmerksamkeit verwendet werden kann.

# 6 Schätzung von Aufmerksamkeit




In den vorangegangenen Kapiteln wurden drei (Teil-) Systeme vorgestellt, die es erlauben, auf Basis verschiedener Modalitäten, Informationen über das Interaktionsinteresse eines Benutzers bzw. über dessen Aufmerksamkeit zu gewinnen. Dabei handelt es sich um die Orientierung des Oberkörpers, die Blickrichtung bzw. Kopfpose und den Gesichtsausdruck. Wie im Abschnitt 1.3 beschrieben wurde, besteht die starke Vermutung, dass diese Informationen aus dem Bereich der Körpersprache dazu genutzt werden können, um eine Aussage über Interaktionsinteresse bzw. Aufmerksamkeit zu erhalten.

Es besteht nun das Problem, dass die drei Teilaussagen der Teilsysteme zu einem Gesamtergebnis fusioniert werden müssen. Hierbei ist zu berücksichtigen, dass die Teilergebnisse verrauscht und unsicher sein können. Darüber hinaus kann es vorkommen, dass zu einem konkreten Zeitpunkt möglicherweise nicht alle Ergebnisse vorliegen.

Im Rahmen dieses Kapitels wird beschrieben, wie sich aus den Ergebnissen der Teilsysteme ein Gesamtresultat bilden lässt und damit eine Aussage über das Interaktionsinteresse bzw. die Aufmerksamkeit des Benutzers in der konkreten Situation getroffen werden kann. Dabei kommen Verfahren aus der *probabilistischen Robotik* zum Einsatz, um mit den Unsicherheiten der Teilergebnisse umgehen zu können.

Auf Basis von Experimenten mit eingewiesenen Probanden wird die Funktionsfähigkeit der Teilmodule und der Fusionierung unter realen Einsatzbedingungen gezeigt.

Auf dieser Grundlage wird ein Gesamtsystem vorgestellt, dass als Grundlage für weitere sozialwissenschaftliche Untersuchungen in den angedachten Einsatzszenarien genutzt werden kann. Solche Untersuchungen sind notwendig, um den angenommenen Zusammenhang zwischen Oberkörperpose, Blickrichtung und Gesichtsausdruck und Interaktionsinteresse bzw. Aufmerksamkeit nachweisen zu können. Diese Untersuchungen sind jedoch bewusst nicht Bestandteil dieser Dissertation.

## 6.1 Architektur Gesamtsystem

Das im Rahmen dieser Dissertation vorgestellte Gesamtsystem zur Schätzung des Interaktionsinteresses bei der Mensch-Roboter-Interaktion besteht aus den in den vorherigen Kapiteln vorgestellten Teilsystemen, die zu einem Gesamtsystem fusioniert werden können (siehe Abbildung 6.1).

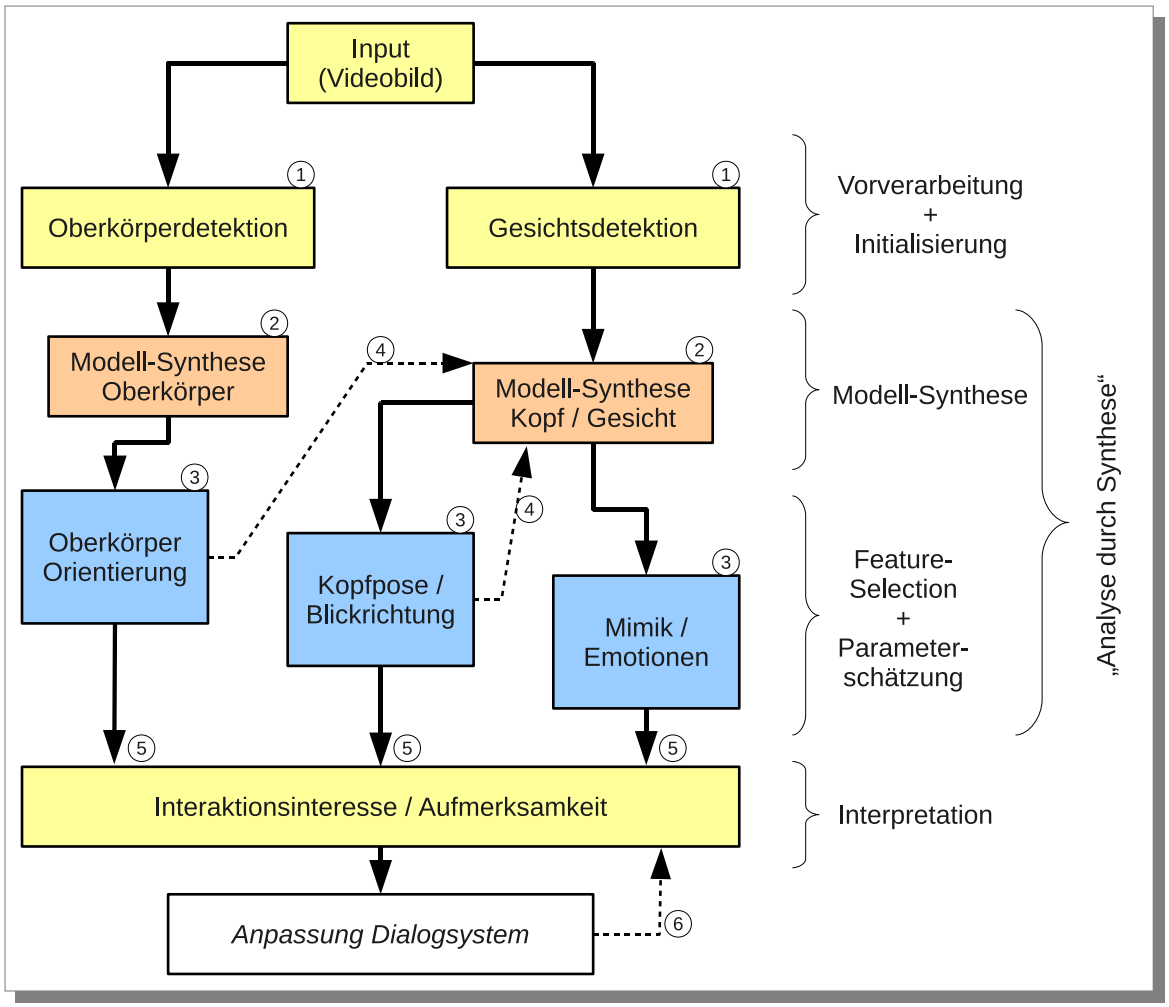



***Abbildung 6.1:*** *Architektur des Gesamtsystems: Der Inputdatenstrom wird bei allen drei Teilsystemen zunächst einer Vorverarbeitung unterzogen. Darauf basierend erfolgt die iterative Synthese eines Modells, aus dessen Parametern die gesuchten Größen geschätzt werden können. Alle drei Teilergebnisse werden abschließend zu einer Gesamtaussage fusioniert.*

In allen drei Teilsystemen werden drei wesentliche Schritte durchgeführt: eine Vorverarbeitung ① zwecks grober Initialisierung, die Anpassung eines parametrischen Modells

an den Input ② und abschließend eine Schätzung der gesuchten Größe auf Basis der Modellparameter ③. Diese drei Schritte werden typischerweise von allen drei Teilsystemen unabhängig voneinander ausgeführt. Je nach gegebener Geometrie des Kamerasystems können jedoch auch die Ergebnisse der Oberkörperschätzung zur Initialisierung der Modell-Synthese des Kopfes bzw. die Modellparameter der Kopfposenschätzung als Initialisierung zur Mimikschätzung verwendet werden ④. Als Teilergebnisse stehen letztendlich jeweils eine Messgröße (z.B. eine detektierte Richtung) und ein zugehöriger Konfidenzwert zur Verfügung. Abschließend werden die drei Teilergebnisse ⑤ zu einem Gesamtergebnis fusioniert, das beispielsweise zur Anpassung des Dialogsystems bzw. der zugrunde liegenden Applikation genutzt werden kann. Weiterhin kann das Dialogsystem bzw. die Anwendung optional über einen Rückkanal ⑥ auch die Aufmerksamkeitsschätzung beeinflussen und steuern. Beispielsweise könnten so gewünschte (Ziel-)Bereiche im Emotionsraum vorgegeben werden, die die Emotion(en) beinhalten, die durch die aktuelle Dialogführung beim Nutzer erreicht werden soll.

Die Adaption des Dialogs bzw. der Anwendung ist nicht Bestandteil dieser Dissertation. Hierzu wären psychologische und sozialwissenschaftliche Untersuchungen zur korrekten Interpretation der Teilergebnisse notwendig. Stattdessen soll im Rahmen dieser Dissertation nur das Grundprinzip der Fusion untersucht und die Machbarkeit der Interessenschätzung demonstriert werden.

Bei der Betrachtung des Gesamtsystems ist zu beachten, dass die drei Teilsysteme nur dann sinnvoll arbeiten können, wenn auch die notwendigen Inputdaten vorliegen. Im Idealfall gibt es mehrere Kamerasysteme, die zu jedem Zeitpunkt der Interaktion für jedes Teilsystem den passenden Input liefern. Auf einem mobilen Robotersystem ist dies jedoch kaum realisierbar. Daher sind dann nicht immer alle drei Teile aktiv, sondern es entsteht ein hierarchisches System, bei dem sich die Stufen teilweise überlappen können bzw. die Information der vorherigen Stufe für die nächste als Initialisierung genutzt werden können. Die konkreten Randbedingungen und Einschränkungen sind abhängig von der eingesetzten mobilen Roboterplattform.

Im Rahmen dieser Dissertation wurden das Gesamtsystem auf Basis der in [Martin et al., 2005b] vorgestellten Software-Architektur realisiert.

## 6.2 Einschränkungen der eingesetzten Roboterplattform

Die im Rahmen dieser Dissertation durchgeführten Integrationsexperimente wurden auf einer *SCITOS G5* Roboterplattform der Firma MetraLabs GmbH [MetraLabs, 2011] durchgeführt. Dabei handelt es sich um eine universelle Roboterplattform, die auch in den Abschnitten 2.1 und 2.2 vorgestellten Szenarien eingesetzt wird. Abbildung 6.2 zeigt ein

Bild der eingesetzten Plattform aus dem Projekt *ALIAS* [ALIAS, 2010].

Die nachfolgenden Erläuterungen und Ergebnisse beziehen sich auf das verwendete Robotersystem *ALIAS*. Daher sollen hier kurz die technischen Randbedingungen der Roboterplattform vorgestellt werden. Generell können die eingesetzten Verfahren und Methoden auch auf andere Robotersysteme übertragen werden.

***Abbildung 6.2:*** *Das für die Integrationsexperimente eingesetzte Robotersystem ALIAS - basierend auf der SCITOS Roboterplattform der Firma MetraLabs.*

Für die Integrationsexperimente wurden zwei Kamerasysteme des Roboters eingesetzt: Zum einen die omnidirektionale Kamera auf dem Kopf des Roboters und zum anderen eine am Display integrierte Minikamera. Die omnidirektionale Kamera liefert Bilder mit einer Auflösung von 1400x240 Pixel mit 15fps. Da diese Kamera aus vier Einzelkameras mit Weitwinkelobjektiven (Öffnungswinkel: horizontal ca. 100° und vertikal ca. 60°) besteht,

können im Überlappungsbereich der Teilbilder Artefakte oder Doppelabbildungen entstehen. Um einen möglichen negativen Einfluss dieser Störungen auf die eingesetzten Verfahren zu vermeiden, wurden die Experimente so durchgeführt, dass der relevante Erfassungsbereich sich immer innerhalb eines Teilbildes befindet. Auf Grund der relativ geringen Auflösung der Kamera und deren Erfassungsbereich wird diese nur für die Bestimmung der Oberkörperpose verwendet.

Die am Display integrierte Minikamera arbeitet mit einer Auflösung von 640x480 bei 30fps. Diese Kamera ist so ausgerichtet, dass ein vor dem Roboter sitzender Benutzer gut erkannt werden kann. Im Idealfall kann hiermit ein hochaufgelöstes Bild des Gesichts aufgenommen werden. Daher wird diese Kamera zur Bestimmung der Kopfpose und der Mimik eingesetzt. Die Oberkörperpose kann mit Hilfe dieser Kamera nicht bestimmt werden, da der Sichtbereich/Öffnungswinkel zu klein ist.

Abbildung 6.3 zeigt eine schematische Darstellung der Öffnungswinkel und Sichtbereiche der beiden Kamerasysteme des ALIAS-Roboters.

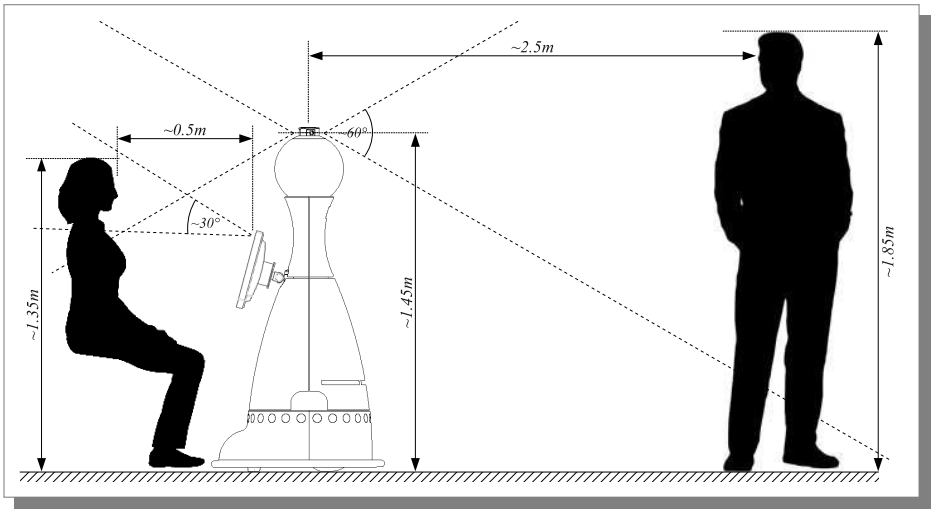



***Abbildung 6.3:*** *Schematische Darstellung der Sichtbereiche der Kameras: Mit Hilfe der omnidirektionalen Kamera kann der Oberkörper einer stehenden Person (ca. > 2.5m entfernt) gut erfasst werden. Bei einem Abstand kleiner 1m, ist die Person nicht mehr vollständig sichtbar. Die im Display eingebaute Kamera dient zur Aufnahme des Gesichts, wenn sich der Nutzer in Interaktion mit dem Roboter befindet (Entfernung < 0.5m).*

Somit sind im Rahmen der Integrationsexperimente auf der verwendenten Roboterplattform folgende Fälle/Situation zu unterscheiden:

- Kein Nutzer im Erfassungsbereich der Kameras.

- Ein Benutzer nähert sich dem Roboter und kann von der omnidirektionalen Kamera erfasst werden. Auf Basis dieser Bilddaten kann nach dem in Kapitel 3 vorgestellten Verfahren die Oberkörperpose geschätzt werden. Eine Kopfposen- und Emotionsschätzung ist noch nicht möglich, da die Auflösung typischerweise zu gering ist.

- Sobald sich der Benutzer dem Roboter nähert (Abstand $< 1m$), kann mit Hilfe der Displaykamera der Kopf erfasst werden. Gemäß dem Verfahren aus Kapitel 4 kann die Kopfpose bestimmt werden. Die Bestimmung der Oberkörperpose ist noch möglich, solange der Nutzer nicht zu dicht am Roboter steht. Sobald sich der Benutzer setzt oder unmittelbar vor dem Roboter steht, ist der Oberkörper nicht mehr vollständig im Bild sichtbar.

- Während des Dialogs kann der Nutzer mit Hilfe der Displaykamera beobachtet werden. Hierbei ist eine kontinuierliche Schätzung der Kopfpose möglich. Die Emotionsschätzung kann ebenfalls durchgeführt werden, wenn der Benutzer annähernd frontal auf das Display schaut. Eine poseninvariante Emotionsschätzung ist nicht Bestandteil dieser Dissertation.

Die Modellierung dieser Zustände und der entsprechenden Übergänge kann mit Hilfe einer einfachen Zustandsmaschine (siehe Abbildung 6.4) realisiert werden.

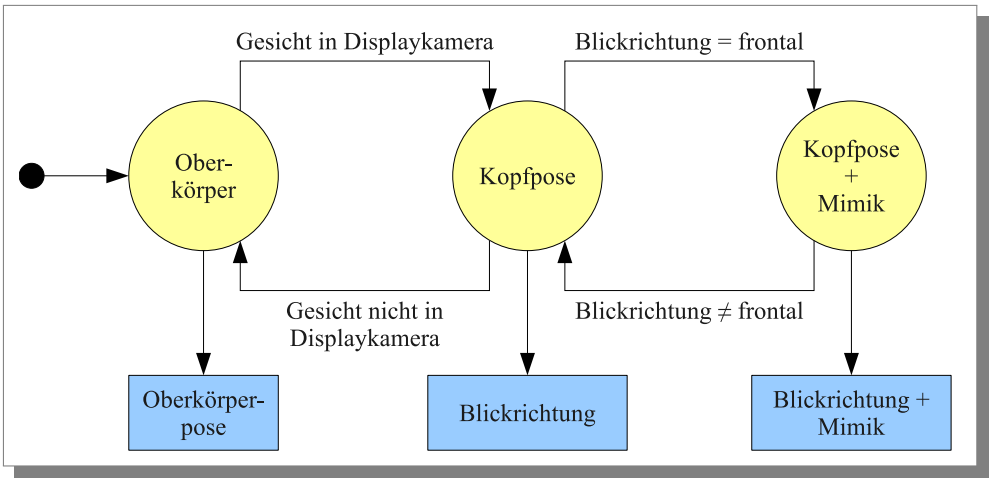



***Abbildung 6.4:*** *Zustandsmaschine der Gesamtstruktur: Auf Grund der gegebenen Kamerageometrie können nicht alle drei Teilsysteme gleichzeitig Informationen liefern. Wenn der Nutzer sich dem Roboter nähert, kann nur die Oberkörperpose bestimmt werden. Sobald der Benutzer im Erfassungsbereich der Displaykamera ist, können die Blickrichtung und der Gesichtsausdruck geschätzt werden.*

Während der Integrationsexperimente wurden Daten von beiden Kamerasystemen aufgezeichnet. Je nach aktivem Zustand wurden dabei entsprechend die Oberkörperpose, die Kopfpose und/oder der emotionale Zustand geschätzt. Diese Daten bilden die Grundlage für die nachfolgende Schätzung des Interaktionsinteresses bzw. der Aufmerksamkeit.

## 6.3 Integration der Teilergebnisse

Die Aufgabe, aus den gegebenen drei Teilinformationen eine Gesamtaussage zu bilden, kann dem klassischen Themenfeld der *Datenfusionierung* oder der *Sensor fusion* zugeordnet werden. Dabei handelt es sich nicht um eine *low-level* Sensorfusionierung auf *Signalebene*, sondern um eine *high-level* Fusionierung von Berechnungsergebnissen auf der *Merkmalsebene* oder noch höher auf der *Symbolebene* (vgl. [Ruser and Puente-Leon, 2006]).

In der Literatur sind eine Vielzahl von möglichen Varianten zur Fusionierung von Daten bekannt. Ein Überblick und eine Taxonomie findet sich beispielsweise in [Ruser and Puente-Leon, 2006]. Da Methoden zur Datenfusionierung bzw. die Suche nach einer optimalen Variante zur Fusionierung der Teilergebnisse keinen Schwerpunkt dieser Dissertation darstellen, wird bei den durchgeführten Experimenten nur ein probabilistischer Ansatz in Kombination mit einem einfachen Regelwerk eingesetzt. Andere Methoden der Datenfusionierung (z.B. merkmalsbasierte Ansätze, Fuzzy-basierte Methoden oder neuronale Ansätze) sollen nicht weiter untersucht werden. Die probabilistische Variante wurde gewählt, da hiermit vorhandene Unsicherheiten in den Teilergebnissen ohne Zusatzaufwand direkt mit modelliert und berücksichtigt werden können.

### 6.3.1 Varianten der Fusionierung

Mit der in den Kapiteln 3 bis 5 vorgestellten Architektur sind grundsätzlich zwei verschiedene Varianten der Fusionierung der Teilergebnisse möglich: In einer ersten Variante werden die Teilergebnisse direkt verwendet, um eine Gesamtaussage über das Interaktionsinteresse bzw. die Aufmerksamkeit zu gewinnen. In einer zweiten Variante werden die Ergebnisse jedes Teilsystems verwendet, um unabhängig von den anderen eine Aussage über das Interaktionsinteresse bzw. die Aufmerksamkeit zu treffen. Anschließend können diese drei Teilaussagen zu einer Gesamtaussage fusioniert werden (siehe Abbildung 6.5).

Die erste Variante ist insbesondere dann sinnvoll, wenn alle Teilergebnisse immer gleichzeitig vorliegen und auf ähnliche Art und Weise zum Gesamtergebnis beitragen können. Dies ist auf Grund der Einschränkungen der verwendeten Roboterplattform im Rahmen dieser Dissertation nicht möglich. Daher wird im Rahmen dieser Dissertation die zweite Variante näher untersucht, da hiermit das Fehlen einzelner Teilergebnisse und die unterschiedliche Bedeutung besser modelliert und ausgeglichen werden kann.

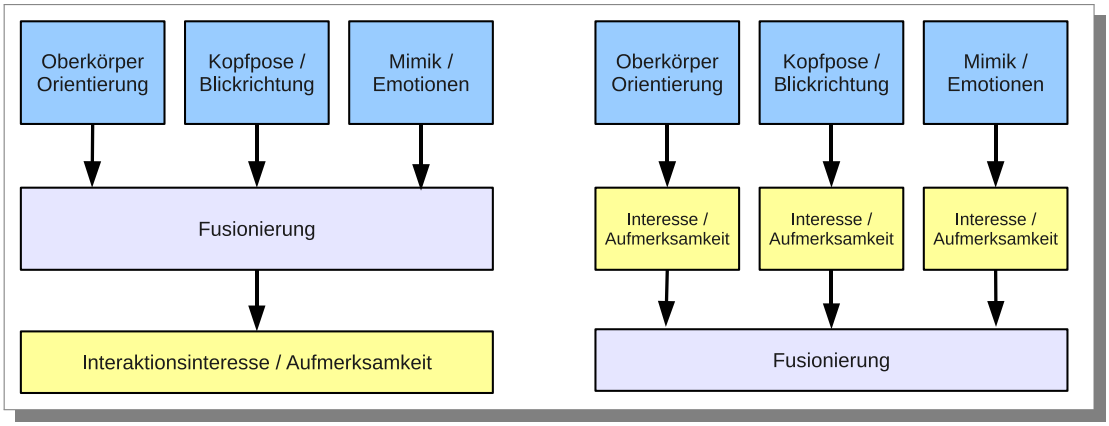



***Abbildung 6.5:*** *Varianten der Fusionierung: Bei der linken Variante werden alle drei Teilergebnisse direkt zu einer Gesamtaussage über Interaktionsinteresse/Aufmerksamkeit fusioniert. Bei der rechten Variante wird aus jedem Teilergebnis eine Aussage über Interaktionsinteresse/Aufmerksamkeit erzeugt und anschließend fusioniert.*

### 6.3.2 Glättung der Teilergebnisse

Im Folgenden wird davon ausgegangen, dass alle drei Teilsysteme in der Lage sind folgende Daten bereitzustellen, insofern – abhängig von den konkreten Randbedingungen der verwendeten Roboterplattform – geeignete Inputdaten zur Verfügung stehen:

- **Oberkörperpose:** Die Oberkörperpose wird als Richtung bzw. als Winkel relativ zum Kamerasystem betrachtet. Ein Winkel von Null bedeutet, dass der Oberkörper direkt frontal zur Kamera ausgerichtet ist. Zusätzlich wird die Position und Größe des Oberkörpers im Bild bestimmt.

- **Blickrichtung:** Die Blickrichtung wird in Form der zwei Winkel *pan* und *tilt* (bzw. *yaw* und *pitch*) ermittelt. Wenn beide Werte Null sind, dann schaut der Benutzer frontal in Richtung Kamera. Weiterhin kann die Position und die Größe des Gesichts im Bild bestimmt werden.

- **Gesichtsausdruck:** Der emotionale Zustand wird in einem 2D-Emotionsraum in Form eines Abstands zum Zentrum (=*Neutral*) und einer Richtung bzw. eines Winkels repräsentiert. Zusätzlich sei hierbei die Lage der *Positiv-Negativ-* und der *Aktiv-Passiv-Achse* (vgl. Abschnitt 5.2.2) aus den Trainingsdaten bekannt.

Als zusätzliche Messgröße können alle Teilsysteme einen Konfidenzwert liefern, der als Maß für die Sicherheit der eigentlichen Schätzung verwendet wird. Ein solcher Wert kann z.B. aus dem verbliebenen Fehler nach der Modellanpassung des ASM oder AAM gewonnen

werden.

Weiterhin wird angenommen, dass alle Systeme in der Lage sind, ungültige Resultate (z.B. außerhalb des abgedeckten Bereichs) frühzeitig zu erkennen und resultierende Fehlschätzungen als solche zu kennzeichnen bzw. den entsprechenden Konfidenzwert zu senken.

Um aus Ausreißern resultierende Schätzfehler zu glätten, kann zu allen drei Systemen eine passende zeitliche Glättung hinzugefügt werden. Im Folgenden sei $y_t$ o.B.d.A. eine der Ergebnisgrößen der Schätzung der Oberkörperorientierung, der Blickrichtung bzw. des emotionalen Zustands. Bereits ein einfacher Tiefpassfilter bzw. eine exponentielle Glättung (ein exponentiell geglätteter Mittelwert) liefert gute Ergebnisse:

$$y_t^* = \alpha \cdot y_t + (1 - \alpha) \cdot y_{t-1}^* \tag{6.1}$$

Da diese Form der Glättung zu einem Schleppfehler bei trendbehafteten Zeitreihen führt, kann auch eine exponentielle Glättung zweiter Ordnung verwendet werden:

$$y_t^{**} = \alpha \cdot y_t + (1 - \alpha) \cdot y_{t-1}^{**} \qquad \widehat{y}_{t+1} = 2 \cdot y_t^* - y_{t-1}^{**} \tag{6.2}$$

Als Resultat dieser zeitlichen Glättung liegen somit die Ergebnisse der Schätzung der Oberkörperpose, der Blickrichtung und des Gesichtsausdrucks in einer Form vor, die für die weitere Verarbeitung von den gröbsten Fehlern bereinigt ist.

Für die nachfolgenden Betrachtungen wird angenommen, dass bereits zeitlich geglättete Werte $y_t^*$ für die jeweiligen Messgrößen vorliegen und diese im Folgenden werden.

### 6.3.3 Probabilistische Betrachtung

Nach einer Glättung der absoluten Messgröße, muss diese in ein geeignetes Maß für das Interaktionsinteresse bzw. die Aufmerksamkeit überführt werden. Anders ausgedrückt findet also eine Zustandsschätzung (in dieser Arbeit: das Interaktionsinteresse oder die Aufmerksamkeit) basierend auf einer zeitlichen Folgen von Beobachtungen (in dieser Arbeit: der Messgrößen) statt.

Diese Umwandlung lässt sich sehr gut mittels Wahrscheinlichkeiten modellieren. In der probabilistischen Robotik hat sich für diese Zwecke der *Bayes-Filter* etabliert [Thrun et al., 2005]:

$$Bel_t(x) = p\left(x|z_{1:t}\right) = 1 - \left(1 + \frac{p(x|z_t)}{1 - p(x|z_t)} \cdot \frac{1 - p(x)}{p(x)} \cdot \frac{Bel_{t-1}(x)}{1 - Bel_{t-1}(x)}\right)^{-1} \tag{6.3}$$

wobei $x$ der gesuchte Zustand und $z$ die vorliegende Messgröße (oft auch *observation* ge-

nannt) ist. Der Wert $p(x|z_t)$ wird auch *inverse measurement model* genannt und beschreibt die Verteilung der Zustandsvariablen als eine Funktion über der Messgröße $z$. Im einfachsten Fall kann hier bereits eine Gauß-Verteilung sehr gute Ergebnisse erzielen. Der Term $p(x)$ wird auch als *prior* bezeichnet und beschreibt die a-priori-Wahrscheinlichkeit des Zustands $x$.

Im Folgenden werden mehrere solcher Bayes-Filter zur Bestimmung der Wahrscheinlichkeit(en) des Interaktionsinteresses bzw. der Aufmerksamkeit eingesetzt.

## 6.4 Aufmerksamkeitsschätzung mittels Bayes-Filter

In den folgenden Abschnitten wird beschrieben, wie auf Basis der Oberkörperpose, der Blickrichtung und der Mimik mit Hilfe mehrerer Bayes-Filter ein Maß für die Aufmerksamkeit bzw. das Interaktionsinteresse eines Benutzers bestimmt werden kann.

### 6.4.1 Aufmerksamkeitsschätzung auf Basis der Oberkörperpose

Im Folgenden sei $Bel_t(Body)$ ein Maß für das Interaktionsinteresse bzw. die Aufmerksamkeit auf Basis der Oberkörperpose zum Zeitpunkt $t$. Dabei steht $Bel_t(Body) = 0$ für ein stark ausgeprägtes Desinteresse und $Bel_t(Body) = 1$ ein sicheres Interesse bzw. hohe Aufmerksamkeit. Bei $Bel_t(Body) = 0.5$ ist der Zustand unbekannt.

Es kann davon ausgegangen werden, dass der Benutzer dann Interesse an einer Interaktion hat, wenn er sich direkt dem Robotersystem zuwendet, der geschätzte Winkel der Oberkörperpose $\phi_{Body}$ also nah bei 0° liegt. Mit einem ersten Bayes-Filter wird daher das Maß $Bel_t(Body, \phi)$ gemäß (6.3) geschätzt. Als inverses Sensormodell kommt dabei eine Gauß-Verteilung über dem Winkel $\phi_{Body}$ mit dem Erwartungswert $\phi_{Body} = 0°$ und einer Streuung $\sigma_{Body,\phi}$ zum Einsatz. Wenn man zwecks Vereinfachung von einer a-priori-Wahrscheinlichkeit von 0.5 ausgeht, ergibt sich eine rekursive Schätzung der folgenden Form:

$$
\begin{aligned}
Bel_t(Body, \phi) &= 1 - \left(1 + \frac{p(Body, \phi|z_t)}{1 - p(Body, \phi|z_t)} \cdot \frac{Bel_{t-1}(Body, \phi)}{1 - Bel_{t-1}(Body, \phi)}\right)^{-1} \\
&\text{mit:} \\
p(Body, \phi|z_t) &= \frac{1}{\sigma_{Body,\phi}} \cdot exp\left(-\frac{1}{2} \cdot \frac{z(t)^2}{\sigma^2_{Body,\phi}}\right) \\
z_t &= \phi^*_{Body}(t)
\end{aligned}
\tag{6.4}
$$

wobei $\phi^*_{Body}(t)$ für die zeitlich geglättete Orientierung (siehe Abschnitt 6.3.2) des Oberkörpers steht. Zum Zeitpunkt $t = 0$ (bzw. nach einer Re-Initialisierung) wird angenommen, dass $Bel_0(Body, \phi) = 0.5$ ist.

Weiterhin wird die Annahme getroffen, dass nur dann ein Interesse besteht, wenn die Person sich relativ langsam bewegt oder vor dem Roboter steht. Wenn sich eine Person mit hoher Geschwindigkeit am Roboter vorbei bewegt, besteht mit hoher Wahrscheinlichkeit kein Interesse an einer Interaktion. Daher wird mit einem zweiten Bayes-Filter das Maß $Bel_t(Body, speed)$ analog zu (6.4) geschätzt. Auch hierbei kommt wieder eine Gauß-Funktion als inverses Sensormodell zum Einsatz.

In beiden Bayes-Filtern kann über die Streuung in der Gauss-Verteilung die Breite des zulässigen Winkel- oder Geschwindigkeitsbereichs gesteuert werden. Bei einem Erwartungswert $\neq 0$ kann auch eine andere gewünschte Zielgröße in Bezug auf Winkel und/oder Geschwindigkeit vorgegeben werden.

Da die Orientierung des Oberkörpers und die Geschwindigkeit der Person als weitestgehend statistisch unabhängig voneinander angenommen werden, gilt letztendlich:

$$Bel_t(Body) = Bel_t(Body, \phi) \cdot Bel_t(Body, speed) \tag{6.5}$$

Abbildung 6.6 zeigt einen beispielhaften Kurvenverlauf. In den ersten Sekunden ist die Person noch nicht im Bild sichtbar. Daher kann keine Schätzung vorgenommen werden. Bei $t = 6s$ liegt eine gültige Modellanpassung vor. Da sowohl der Winkel relativ groß ist und auch eine deutliche Bewegung im Bild stattfindet, geht $Bel_t(Body)$ gegen Null. Bei $t = 7...8s$ ist der Oberkörper fast direkt auf den Roboter gerichtet. Daher steigt $Bel_t(Body, \phi)$ an. Da aber weiterhin eine hohe Geschwindigkeit geschätzt wird, bleibt $Bel_t(Body)$ klein. Erst bei $t = 11s$ liegen sowohl Winkel als auch Geschwindigkeit im gewünschten Bereich. Daher steigt das Aufmerksamkeitsmaß deutlich an. Am Ende der Sequenz bewegt sich der Benutzer aus dem Bild heraus und setzt sich vor dem Roboter hin. Daher kann keine gültige Schätzung mehr vorgenommen werden, und das Aufmerksamkeitsmaß sinkt.

### 6.4.2 Aufmerksamkeitsschätzung auf Basis der Kopfpose

Das Interaktionsinteresse auf Basis der Kopfpose wird im Folgenden als $Bel_t(Head)$ bezeichnet. Wie im vorherigen Abschnitt werden auch hier zwei separate Bayes-Filter zur Schätzung eingesetzt.

Der erste Filter schätzt die Aufmerksamkeit $Bel_t(Head, \phi)$ auf Basis der Winkel *pan* und *tilt* analog zu (6.4). Wenn die verwendete Kamera im Display eingebaut ist und es das Ziel ist, dass der Nutzer seine Aufmerksamkeit auf den Bildschirm richtet, kann hier eine Gauß-Verteilung über die beiden Winkel mit dem Erwartungswert Null verwendet werden. Wenn die Kamera nicht direkt im Display verbaut ist, sondern beispielsweise seitlich oder darüber, und damit die Zielgröße bei einem Winkel $\neq 0$ ist, kann ein entsprechender Erwartungswert im inversen Sensormodell vorgegeben werden. Über die Varianz der Gauß-Verteilung kann

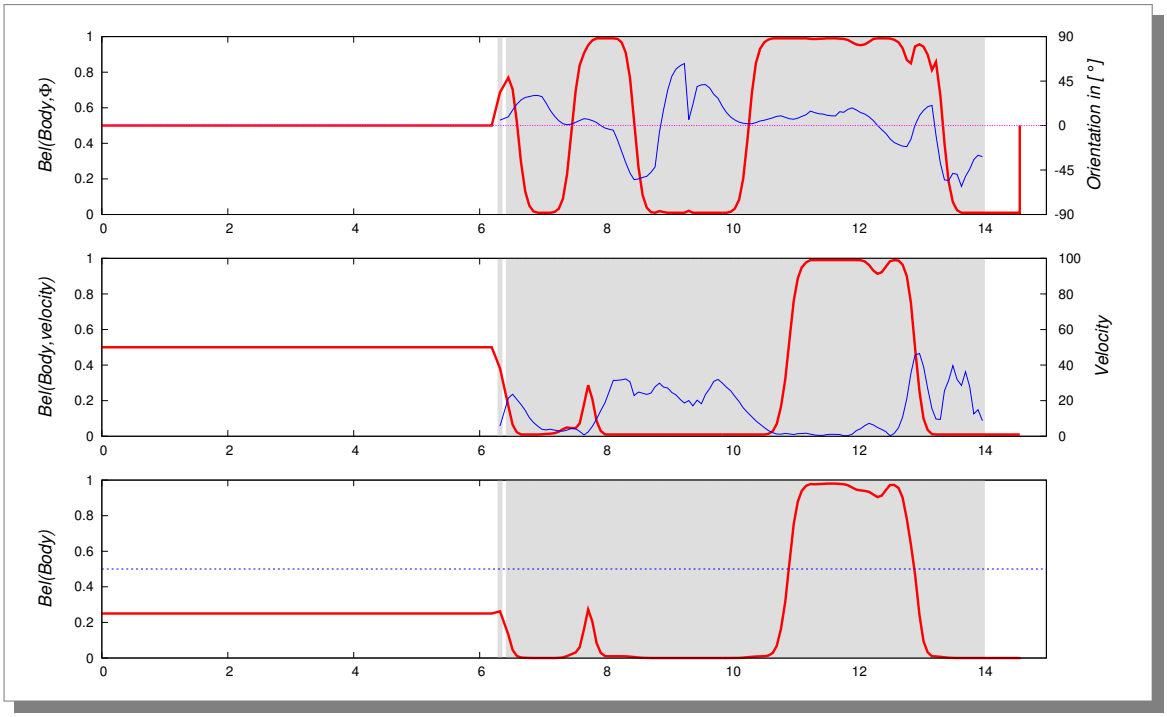



***Abbildung 6.6:*** *Schätzung der Aufmerksamkeit aus der Oberkörperorientierung: Im grau hinterlegten Bereich lag eine korrekte Modellanpassung vor. Oben: Winkelschätzung in Blau und resultierendes Maß $Bel_t(Body, \phi)$ in Rot. Mitte: Geschwindigkeit in Blau und resultiendes Maß $Bel_t(Body, speed)$ in Rot. Unten: Letztendliche Schätzung $Bel_t(Body)$.*

gesteuert werden, wie weit der Nutzer an der Null-Achse vorbeischauen darf und trotzdem noch als interessiert eingeschätzt wird. Typischerweise ist hier $\sigma = 10°$ ein geeigneter Wert.

Als zweite Messgröße wird die Skalierung des Gesichts ausgewertet. Diese kann direkt aus den globalen Parametern des AAM-Modells bestimmt werden. Im Folgenden wird angenommen, dass ein Interesse dann besteht, wenn der Benutzer sich innerhalb einer gewissen Entfernung zum Bildschirm befindet und somit die Skalierung einen entsprechenden Schwellwert überschreitet. Auf dieser Basis wird mit einem zweiten Bayes-Filter das Maß $Bel_t(Head, scale)$ analog zu (6.4) geschätzt.

Da die Kopfpose und die Skalierung auch als weitestgehend statistisch unabhängig voneinander angenommen werden, können auch diese beiden Maße multipliziert werden:

$$Bel_t(Head) = Bel_t(Head, \phi) \cdot Bel_t(Head, scale) \tag{6.6}$$

Abbildung 6.7 zeigt einen beispielhaften Kurvenverlauf. Bis $t = 18s$ wird das Gesicht noch

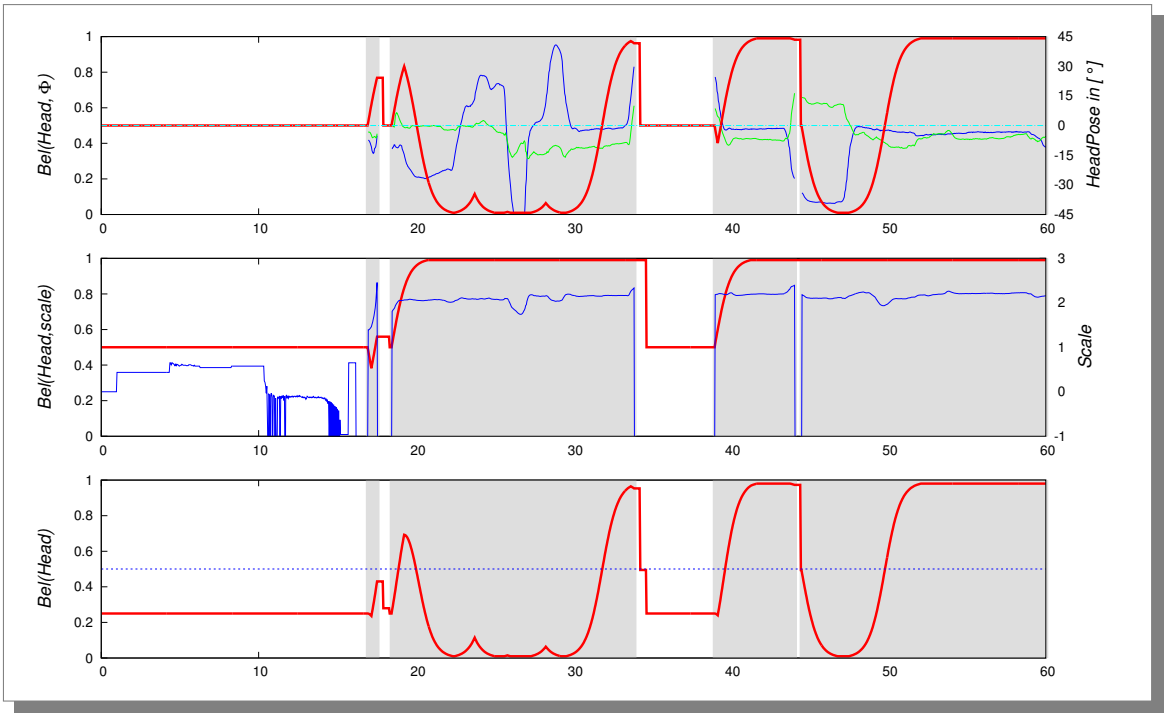



***Abbildung 6.7:*** *Schätzung der Aufmerksamkeit aus der Kopfpose: Im grau hinterlegten Bereich lag eine korrekte Modellanpassung vor. Oben: Winkelschätzung in Blau und Grün und resultierendes Maß $Bel_t(Head, \phi)$ in Rot. Mitte: Größe/Skalierung in Blau und resultiendes Maß $Bel_t(Head, scale)$ in Rot. Unten: Letztendliche Schätzung $Bel_t(Head)$.*

nicht von der Kamera erfasst. Daher wird hier keine Schätzung vorgenommen. Ab $t = 20s$ hat die Größe des Gesichts den Schwellwert überschritten, und daher geht $Bel_t(Head, scale)$ gegen 1. Da bis $t = 35s$ jedoch noch große Kopfbewegungen stattfinden, geht $Bel_t(Head, \phi)$ und damit auch $Bel_t(Head)$ gegen Null. Im Zeitraum $t = 35...40s$ liegt keine gültige Modellanpassung vor. Von $t = 40s$ bis $t = 45s$ schaut der Nutzer gerade in die Kamera und somit geht das geschätzte Interessen-Maß gegen 1. Nach einer erneuten größeren Kopfbewegung, während der $Bel_t(Head)$ wieder sinkt, schaut der Nutzer ab t=50$s$ kontinuierlich direkt in Richtung Kamera und $Bel_t(Head)$ steigt wieder an.

### 6.4.3 Aufmerksamkeitsschätzung auf Basis der Mimik

Im Gegensatz zur Oberkörperpose und der Blickrichtung, kann aus dem Gesichtsausdruck nicht nur eine Aussage über das Interesse gewonnen werden, sondern es kann direkt der (über den Gesichtsausdruck sichtbare) emotionale Zustand des Benutzers abgeschätzt werden. Durch die Modellierung des emotionalen Zustands in einem kontinuierlichen Emotionsraum (siehe Abschnitt 5.2.2) kann durch die Anwendung oder das Dialogsystem ein erwünsch-

ter Zielbereich und/oder eine -größe vorgegeben werden. Im Fall des im Abschnitt 5.6.3 verwendeten zweidimensionalen Emotionsraums stehen als Messgrößen zwei Varianten zur Verfügung:

- Variante 1: Abstand zum neutralen Zentrum und der Winkel. Der Abstand gibt dabei an, wie stark eine Emotion ausgeprägt ist. Über den Winkel wird die eigentliche Emotion kodiert. Dies entspricht der Interpretation in Polarkoordinaten.
- Variante 2: Einordnung der Emotion in Bezug auf die *Positiv-Negativ-Achse* und die *Aktiv-Passiv-Achse*. Diese Variante kann mit normalen kartesischen Koordinaten verglichen werden.

Da beide Varianten eindeutig ineinander umgerechnet werden können, wird im Rahmen dieser Arbeit nur die erste weiter untersucht.

Die eigentliche Schätzung, wann ein gewünschter emotionaler Zustand erreicht ist, kann auch hier über einen Bayes-Filter realisiert werden. Als inverses Sensormodell kommt auch hier wieder eine Gauß-Verteilung um die gewünschte Zielregion im zweidimensionalen Emotionsraum zum Einsatz. Diese entspricht der verwendeten Nachbarschaftsfunktion der Kohonen-Karten (siehe Abschnitt 5.6.3) bei der Emotionsschätzung. Die Schätzung von $Bel_t(Mimic)$ wird dabei nur vorgenommen, wenn der Nutzer frontal in die Kamera schaut, also $Bel_t(Head)$ einen festgelegten Schwellwert überschreitet.

Abbildung 6.8 zeigt einen beispielhaften Verlauf von $Bel_t(Mimic)$, bei dem geschätzt werden sollte, ob der Nutzer der Region *Happy* zugeordnet werden kann. Die grau hinterlegten Bereiche kennzeichnen die Zeiträume, in denen der Nutzer frontal in Richtung Kamera schaut. Der zugehörige Verlauf der Kopfpose ist in Abbildung 6.7 dargestellt. Die Region *Happy* liegt im dabei zugrunde liegenden Modell bei einem Winkel von $\approx -135°$.

Der Kurvenverlauf zeigt deutlich, dass die Mimikschätzung zunächst sehr verrauscht ist. Der Grund hierfür ist, dass das AAM-Modell zur Mimikschätzung nur mit frontalen Bildern trainiert wurde und daher bei großen Kopfbewegungen keine gute Modellanpassung erreicht werden kann. Nur in den grau hinterlegten Bereichen liefert die Mimik verlässliche Aussagen. Im Bereich $t = 50...52s$ zeigt der Benutzer einen deutlich freudigen Gesichtsausdruck. Daher steigt hier das Maß $Bel_t(Mimic)$ auch unmittelbar an.

### 6.4.4 Fusionierung der Ergebnisse

Wie im Abschnitt 6.3.1 erläutert und in Abbildung 6.5 dargestellt, sollen die drei Werte $Bel_t(Body)$, $Bel_t(Head)$ und $Bel_t(Mimic)$ in einem nachgeschalteten Schritt fusioniert werden. Auf Grund der zeitlichen Verfügbarkeit und Bedeutung der Schätzwerte wird hierzu ein einfaches Regelwerk vorgeschlagen:

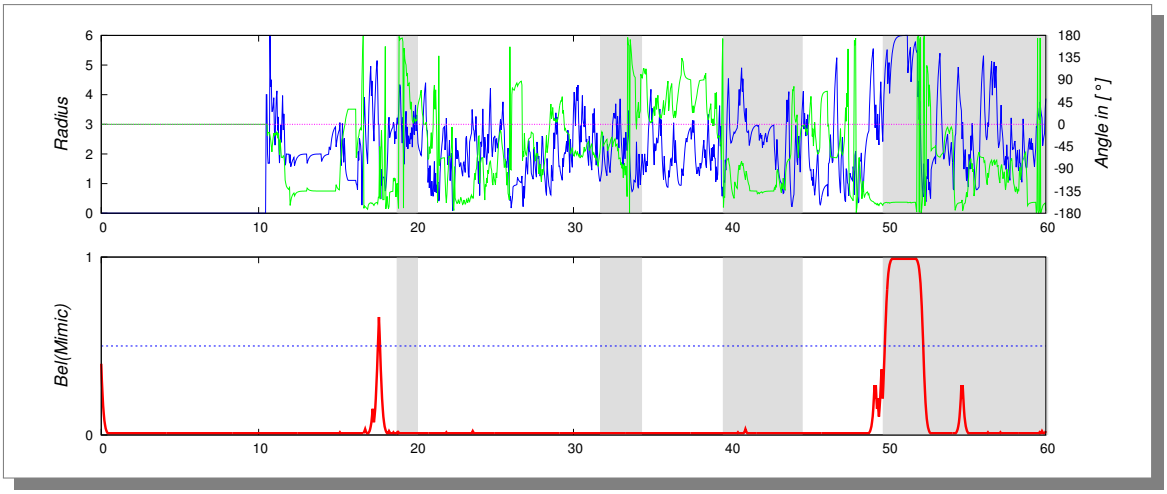



***Abbildung 6.8:*** *Schätzung eines emotionalen Zustands. Im grau hinterlegten Bereich schaut der Nutzer frontal in die Kamera. Nur in diesem Bereich ist eine sinnvolle Mimikschätzung möglich. Oben: Radius (Blau) und Winkel (Grün) im Emotionsraum. Unten: Schätzung $Bel_t(Mimic)$.*

- Wenn $Bel_t(Body) < \theta_{Body}$ und $Bel_t(Head) < \theta_{Head}$ :

  $\Rightarrow$ Kein Nutzer vorhanden.

- Wenn $Bel_t(Body) > \theta_{Body}$ und $Bel_t(Head) < \theta_{Head}$ :

  $\Rightarrow$ Nutzer steht vor dem Roboter und richtet seine Aufmerksamkeit in Richtung des Roboters.

- Wenn $Bel_t(Body) < \theta_{Body}$ und $Bel_t(Head) > \theta_{Head}$ :

  $\Rightarrow$ Nutzer schaut in die Displaykamera und richtet seine Aufmerksamkeit auf das Display des Roboters.

Wenn die letzte Regel aktiv ist, kann zusätzlich $Bel_t(Mimic)$ ausgewertet werden. Diese Auswertung ist jedoch abhängig von der konkreten Anwendung und dem verwendeten Dialogsystem und soll im Rahmen dieser Dissertation nicht weiter betrachtet werden.

## 6.5 Ergebnisse

Auf der Basis des in den letzten Abschnitten vorgestellten Verfahrens zur Schätzung des Interaktionsinteresses und der Aufmerksamkeit wurden eine Reihe von Tests durchgeführt, um die Leistungfähigkeit des Verfahrens zu untersuchen. Dazu wurde ein einfaches Szenario in einer Büroumgebung erstellt, das mit mehreren Probanden durchgespielt wurde.

Jeder Proband wurde gebeten, zum Roboter zu gehen und sich auf einen Stuhl vor dem Roboter zu setzen. Auf dem Bildschirm sollte dann ein kurzer Text gelesen werden. Parallel dazu wurden die Personen zufällig abgelenkt, indem sie von der Seite angesprochen wurden. Am Ende der Sequenz sollten die Personen zusätzlich noch ein freundliches Gesicht oder ein Lächeln zeigen. Für jeden Probanden wurden dabei die Daten der omnidirektionalen Kopfkamera und der Displaykamera als Videosequenzen aufgenommen. Die mittlere Dauer eines Testlauf betrug ca. 60 Sekunden. Abbildung 6.9 zeigt Beispielbilder eines typischen Testlaufs. Nachträglich wurden die aufgezeichneten Videodaten in Bezug auf die Oberkörperpose, die Kopfpose und den Gesichtsausdruck ausgewertet. Gemäß dem Verfahren aus Abschnitt 6.4 wurde der zeitliche Verlauf von $Bel_t(Body)$ und $Bel_t(Head)$ bestimmt und ausgewertet.

(A) Nutzer nähert sich dem Roboter

(B) Nutzer sitzt vor dem Roboter

(C) Nutzer ist abgelenkt und zur Seite

(D) Nutzer zeigt ein freudiges Gesicht

***Abbildung 6.9:*** *Beispielbilder der beiden Kameras aus den Testsequenzen. Für die vier Situationen (A)-(D) ist jeweils der Bildausschnitt der omnidirektionalen Kamera (links) und der Displaykamera (rechts) dargestellt.*

Insgesamt wurden 15 Sequenzen von insgesamt 5 verschiedenen Personen aufgenommen und untersucht. Anhand dieser soll in diesem Abschnitt exemplarisch gezeigt werden, dass das vorgestellte System die gewünschte Funktion der Schätzung des Interaktionsinteresses bzw. der Aufmerksamkeit realisieren kann.

Um eine Vergleichsgrundlage zur Auswertung zu haben, wurden die Sequenzen zusätzlich durch einen menschlichen Betrachter in Bezug auf Interesse und/oder Aufmerksamkeit gelabelt. Diese Daten werden im Folgenden als *Ground-Truth-Information* verwendet.

Die Abbildung 6.10 zeigt drei ausgewählte Beispiele. Jeweils im oberen Teil der Grafiken sind

die Ergebnisse der Verfahren aus dieser Dissertation zu sehen. Es sind jeweils der zeitliche Verlauf von $Bel_t(Body)$ in Rot und $Bel_t(Head)$ in Grün dargestellt. Der grau hinterlegte Bereich ist der, in dem der gewählte Schwellwert von 0.5 überschritten wird und der Nutzer somit als interessiert oder aufmerksam eingestuft wird. Im unteren Teil der Grafiken ist jeweils der durch den menschlichen Betrachter als relevant eingestufte Bereich dargestellt.

In den Beispielen ist ersichtlich, dass eine gute Übereinstimmung zwischen den erzielten Ergebnissen und der *Ground-Truth-Information* erreicht werden konnte. Jedoch sind auch Differenzen vorhanden:

(A): In allen Beispielen ist zu erkennen, dass der menschliche Betrachter in der Übergangsphase zwischen "Nutzer steht vor dem Roboter" und "Nutzer sitzt vor dem Roboter" den Vorgang des Hinsetzens als solchen erkennt und den Benutzer weiterhin als interessiert einstuft. Die Verfahren zur Schätzung der Oberkörperpose und der Blickrichtung können diese Übergänge jedoch nicht erkennen und können damit auch kein korrektes Ergebnis liefern.

(B): Durch die Glättung der Rohdaten der Oberkörper- und Blickrichtungsschätzung (siehe Abschnitt 6.3.2) und der Integration der Ergebnisse durch die Bayes-Filterung ergibt sich ein zeitlicher Versatz von ca. 1-2$s$. Dieser ist insbesondere jeweils im zweiten Teil der Testsequenzen zu sehen, wenn der Nutzer durch eine Ablenkung den Kopf zur Seite bewegt und $Bel_t(Head)$ entsprechend sinkt und später wieder ansteigt.

Das Problem (A) ist bedingt durch das gewählte Szenario und kann dem System nicht direkt angelastet werden. Durch eine geeignete Wahl der Parameter und eine Behandlung der Übergänge können diese Probleme für eine konkrete Anwendung jedoch reduziert bzw. vollständig vermieden werden.

Das Problem (B) spielt ebenfalls nur eine untergeordnete Rolle, da ein zeitlicher Versatz von wenigen Sekunden für die Analyse von Aufmerksamkeit oder Interesse kaum relevant ist.

Als quantitatives Vergleichsmaß zwischen den berechneten Daten und der Beobachtung des menschlichen Betrachters werden die Übereinstimmungsrate (EQ), die Falsch-Positiv-Rate (FP), die Falsch-Negativ-Rate (FN) und der Korrelationskoeffizient $\rho$ zwischen den Kurven berechnet. Zusätzlich wurde das Problem (B) herausgerechnet, in dem der ermittelte Kurvenverlauf um $t = 1s$ verschoben wurde:

| | direkt | | | | shifted | | | |
|---|---|---|---|---|---|---|---|---|
| **Datensatz** | EQ | FP | FN | $\rho$ | EQ' | FP' | FN' | $\rho'$ |
| final-05 | 78% | 5% | 17% | 0.59 | 86% | 2% | 12% | 0.75 |
| final-11 | 82% | 0% | 18% | 0.68 | 80% | 0% | 20% | 0.65 |
| final-16 | 74% | 8% | 18% | 0.48 | 80% | 5% | 15% | 0.62 |

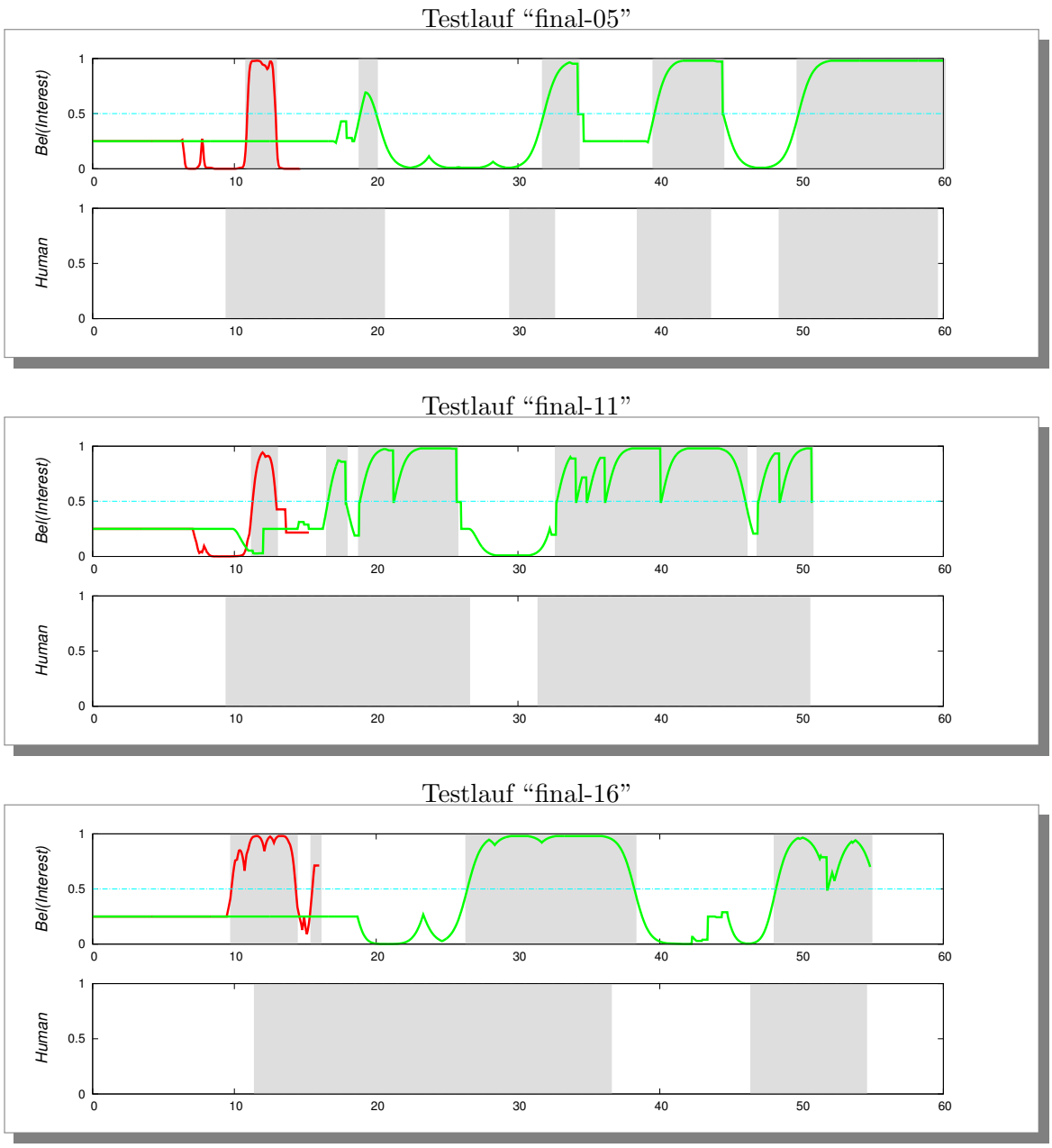



***Abbildung 6.10:*** *Schätzung der Aufmerksamkeit auf drei Datensätzen: Die obere Grafik zeigt jeweils $Bel_t(Body)$ in Rot und $Bel_t(Head)$ in Grün. Der grau hinterlegte Bereich wird als Aufmerksamkeit eingestuft. Die untere Grafik zeigt jeweils dazu im Vergleich die Einstufung des Interesses durch einen menschlichen Tester auf denselben Daten.*

Über alle aufgenommenen Datensätze gemittelt ergibt sich eine Übereinstimmungsrate von 76%, eine Falsch-Positiv-Rate von 4% und eine Falsch-Negativ-Rate von 18%. Die hohe Falsch-Negativ-Rate ist bedingt durch das Problem (A) und kann in dem vorgestellten Szenario und den vorhandenen Verfahren daher nicht weiter reduziert werden.

Zusammenfassend kann also festgestellt werden, dass die präsentierten Verfahren im untersuchten Szenario eine gute Performance erreicht haben. Solange die darunterliegenden Verfahren zur Schätzung der Oberkörperpose und der Blickrichtung stabile und korrekte Ergebnisse liefern, kann auch eine korrekte Schätzung von $Bel_t(Body)$ und $Bel_t(Head)$ erfolgen und somit das Interaktionsinteresse des Benutzers bestimmt werden.

## 6.6 Anpassung Dialogsystem

In den vorangegangenen Abschnitten wurde gezeigt, wie sich aus den Rohdaten der Oberkörperorientierung, der Kopfpose und des emotionalen Zustands Aussagen über das Interaktionsinteresse bzw. die Aufmerksamkeit eines Benutzers in Form von Wahrscheinlichkeiten gewinnen lassen.

Auf dieser Basis kann ein nachgeschaltetes Dialogsystem eine Anpassung des Dialogs bzw. der Interaktion zwischen Benutzer und Robotersystem vornehmen, um die konkrete Zielstellung oder den Zweck einer Anwendung besser zu erreichen. Einige mögliche Stellgrößen zur Anpassung des Dialogs könnten z.B. sein:

- Lautstärke der Sprachausgabe
- Geschwindigkeit der Dialogführung
- Überspringen von Teilen des Dialogs
- Anpassung des Schwierigkeitsgrad von Anwendungsinhalten (z.B. Spielen)
- Auswahl von passenden Motiven in einem Memory-Spiel
- etc.

Die eigentliche Anpassung des Dialogs und der damit notwendigen Methoden, Verfahren und Strategien sind nicht Gegenstand dieser Dissertation und sollen daher hier nicht weiter untersucht werden.

Neben der Anpassung des Dialogs auf Basis der Schätzung von Interesse und Aufmerksamkeit kann auch umgekehrt das Dialogsystem die Parameter der Bayes-Filterung anpassen. Beispielsweise kann das inverse Sensormodell für $Bel_t(Mimic)$ so angepasst werden, dass entweder gezielt nur ein speziell gewünschter Gesichtsausdruck angestrebt wird oder aber

ein sehr großer Bereich im emotionalen Zustandsraum als für die aktuelle Aufgabe relevant und adäquat definiert ist. Beispiel: Bei einem kognitiven Trainingsprogramm (z.B. einem adaptiven Memory-Spiel) ist es ausreichend, wenn der Nutzer nicht gelangweilt wird und daher der Gesichtsausdruck tendenziell eher dem positiven Bereich des Emotionsraumes zugeordnet werden kann. Das Sensormodell könnte in diesem Fall eine Gauß-Verteilung mit großem Sigma sein. Wenn das Ziel aber direkt die Erkennung der Emotion *Happy* ist, dann sollte das Sensormodell nur in diesem Bereich des Emotionsraumes ansprechen. Dies kann realisiert werden, wenn das Sigma entsprechend klein gewählt wird.

## 6.7 Zusammenfassung

Im Rahmen dieser Dissertation wurde in den Kapiteln 3 bis 5 vorgestellt, wie aus einem geeigneten Kamerabild die Oberkörperpose, die Blickrichtung und der Gesichtsausdruck eines Benutzers im Rahmen der Mensch-Roboter-Interaktion bestimmt werden können.

Basierend auf diesen Ergebnissen wurde in diesem Kapitel vorgestellt, wie aus diesen Daten eine Aussage über das Interaktionsinteresse oder die Aufmerksamkeit gewonnen werden kann. Die Zusammenführung der Ergebnisse kann nach [Ruser and Puente-Leon, 2006] einer *high-level* Fusionierung auf *Merkmals-* oder *Symbolebene* zugeordnet werden.

Unter der Vielzahl von möglichen Varianten zur Datenfusionierung wurde ein probabilistischer Ansatz gewählt, da sich damit die vorhandenen Unsicherheiten in den Rohdaten direkt modellieren lassen und *Bayes-Filter* bereits in vielen anderen Teilgebieten der mobilen Robotik erfolgreich eingesetzt wurden.

Die eingesetzten Bayes-Filter haben weiterhin den Vorteil, dass über das *inverse measurement model* die gewünschte Zielgröße direkt beeinflusst werden kann. Im einfachsten Fall können dies feste Parameter sein. Alternativ können die verwendeten Gauß-Verteilungen jedoch auch im Merkmalsraum verschoben werden, um eine andere Zielregion in Bezug auf die Schätzung von Interesse und Aufmerksamkeit auszuwählen.

Die vorgestellten Verfahren wurden auf einem realen Robotersystem aus dem Projekt ALIAS [ALIAS, 2010] evaluiert. Durch die Verwendung dieses Robotersystems ergeben sich Einschränkungen, die durch die Sichtbereiche der integrierten Kamerasysteme bedingt sind. Die eingesetzten Methoden und Verfahren sind aber nicht auf dieses System beschränkt, sondern können durch geeignete Anpassungen (insbesondere Parametereinstellungen oder auch Koordinatentransformationen) auch leicht auf andere Robotersysteme mit deren Spezifika übertragen werden.

Anhand von Tests mit eingewiesenen Probanden wurde in einer Büroumgebung gezeigt, dass die vorgestellten Methoden und Verfahren in der Lage sind, die Aufmerksamkeit oder das Interaktionsinteresse eines Benutzers gut zu schätzen. Die erzielten Ergebnisse wurden mit denen eines menschlichen Beobachters verglichen. Dabei konnte eine hohe Übereinstimmung erreicht werden. Grundvoraussetzung für die korrekte Analyse ist, dass die Rohdaten der Schätzung der Oberkörperpose, der Blickrichtung und des Gesichtsausdrucks stabil und korrekt sind.

Damit die vorgestellten Verfahren gut funktionieren, ist es notwendig, dass diese gut parametrisiert sind. Nur wenn sämtliche Zeitkonstanten, die Sensormodelle und sonstige Parameter gut eingestellt sind, wird das gewünschte Ergebnis erreicht. Im Rahmen dieser Dissertation wurden sämtliche Parameter anhand von praktischen Gegebenheiten abgeschätzt und somit in Form von Designer-Wissen eingebracht. Die Wahl bzw. die Anpassung der Parameter müsste Bestandteil der ausstehenden sozialwissenschaftlichen Untersuchungen sein. Diese müssten so ausgelegt werden, dass einerseits allgemein verwendbare Parameter bestimmt werden können, aber andererseits auch spezielle Parameterkombinationen, die in einer bestimmten Anwendung bzw. einem konkreten Szenario notwendig sind. Dabei sind auch die speziellen Gegebenheiten und technischen Randbedingungen der verwendeten Roboterplattform(en) zu berücksichtigen.

# 7 Zusammenfassung und Ausblick

## 7.1 Zusammenfassung

Durch die zunehmende „Technisierung des Alltags“ kommt man im täglichen Leben immer häufiger in Kontakt mit interaktiven Systemen. Zu diesen gehören beispielsweise Navigationssysteme, moderne Smartphones oder auch *Serviceroboter*. Ein Problem in der Benutzung solcher Systeme besteht darin, dass die Interaktion zwischen dem menschlichen Nutzer und dem technischen System oft sehr einseitig ist. Die Menüführung ist beispielsweise oftmals nur wenig flexibel und läuft unabhängig vom Nutzer bzw. dessen Kenntnissen ab. Um letztendlich intuitiv bedienbare, intelligente Systeme realisieren zu können, ist es aber notwendig, dass sich die Systeme an die Eingaben des Benutzers anpassen können. Hierzu müssen sie in der Lage sein, gewisse Zustandsinformationen über den Benutzer erfassen zu können. Dazu gehört beispielsweise die Schätzung des Interaktionsinteresses und der Aufmerksamkeit, basierend auf einem Videodatenstrom, welche den Kern der vorliegenden Dissertation bildet.

Im Rahmen dieser Dissertation wurde diese Problematik im Forschungsumfeld der *interaktiven mobilen Serviceroboter* untersucht. Nach Kenntnisstand des Autors gibt es bisher keine vergleichbaren Arbeiten, die das Interaktionsinteresse und die Aufmerksamkeit eines Benutzers bei der Interaktion mit einem mobilen Robotersystem untersuchen. In Kapitel 2 wurden zwei mögliche Einsatzfelder vorgestellt: ein Shoppingroboter für den Einzelhandel und ein Roboter in einem häuslichen Umfeld. Aus der Literatur ist bekannt, dass sich bei der zwischenmenschlichen Kommunikation eine Interaktionsbereitschaft typischerweise über Kopf- und Körperausrichtung und über den Blickkontakt zeigt. Daher wurden drei Merkmale ausgewählt, die zur Schätzung des Interaktionsinteresses bzw. der Aufmerksamkeit geeignet sind: die Oberkörperpose, die Blickrichtung und der Gesichtsausdruck.

Um die Arbeit auf eine einheitliche methodische Basis zu stellen, wurden im Rahmen der vorliegenden Dissertation Verfahren und Algorithmen aus der Gruppe der „Analyse durch Synthese“-Methoden ausgewählt und in die Teilsysteme zur Schätzung der genannten drei Merkmale integriert. Als weitere Randbedingungen wurden die Beschränkung auf Standardkameras und der Ausschluss von auditiven oder 3D-Informationen festgelegt.

In Kapitel 3 wurde ein System zur Schätzung der Oberkörperorientierung vorgestellt. Dabei handelt es sich um ein mehrstufiges System. In der ersten Stufe erfolgt eine Grobdetektion

des Oberkörpers auf Basis eines *HOG-Detektors*. Die detaillierte Erfassung des Oberkörpers wird mit Hilfe eines *Active-Shape-Models* in der zweiten Stufe realisiert. Aus den Modellparametern wurden auf Basis der *Mutual Information* diejenigen ausgewählt, die auch Informationen über die Oberkörperorientierung enthalten. Hierdurch konnte eine erhebliche Reduzierung des Merkmalsraumes erreicht werden. Auf Basis dieser Parameter wird in der letzten Stufe die eigentliche Schätzung der Oberkörperorientierung vorgenommen. Hierzu wurden vier verschiedene Funktionsapproximatoren untersucht. Die besten Ergebnisse wurden mit einer *Support Vector Machine* erzielt. Jedoch ist dafür der Trainingsaufwand am größten. Das beste Aufwand-Nutzen-Verhältnis konnte mit einem *Multi Layer Perceptron* erreicht werden. Die Leistungsfähigkeit dieses ersten Teilsystems wurde auf verschiedenen Videodatenströmen getestet. Die Tests wurden durchgeführt auf Daten, für die mit einem Referenzsystem auch die entsprechenden *Ground-Truth-Information* gewonnen werden konnten, und zusätzlich auch auf Realweltdaten aus einer Shopping-Anwendung. Es konnte gezeigt werden, dass die vorgestellte Architektur zur Schätzung der Oberkörperpose geeignet ist. Bei Recherchen des Autors wurden in der Literatur kaum vergleichbare Verfahren gefunden, die eine solche Schätzung der Oberkörperorientierung realisieren. Die meisten Arbeiten beschränken sich auf die Erfassung der Kontur, aber werten diese nicht weiter aus.

Als zweites Teilmodul wurde in Kapitel 4 ein System zur Schätzung der Blickrichtung präsentiert. Im Rahmen der vorliegenden Arbeit wird hierzu die Kopfpose verwendet. Die eigentliche Blickrichtung auf Basis der Augen und Pupillen wurde nicht berücksichtigt, da in der zwischenmenschlichen Kommunikation die Kopfpose als zeitliche Tiefpassfilterung der Augenbewegungen betrachtet werden kann und damit Rückschluss auf den visuellen Aufmerksamkeitsfokus erlaubt. Wie bei der Schätzung der Oberkörperorientierung wurde auch hier ein dreistufiger Ansatz gewählt. In der ersten Stufe erfolgt die Initialdetektion des Kopfes auf Basis des *Viola-und-Jones* Gesichtsdetektors. Dieser Detektor benötigt nur wenig Rechenleistung und besitzt andererseits eine geringe Falsch-Positiv-Rate. Die im Bild gefundene Position des Gesichts wird zur Initialisierung eines *Active-Appearance-Modells* verwendet. Mit Hilfe der *Mutual Information* werden auch bei diesem Teilsystem nur die Modellparameter ausgewählt, die Informationen über die Kopfpose enthalten. Die eigentliche Schätzung der Kopfpose wurde mit Hilfe einer einfachen linearen Approximation und einem *Multi Layer Perceptron* durchgeführt. Dabei konnte eine Genauigkeit von 3° bis 5° auf der Testdatenbank erreicht werden. Diese Ergebnisse entsprechen denen, die bekannte Verfahren aus der Literatur ebenfalls erreichen. Zusätzlich wurde die Leistungsfähigkeit auf bekannten und unbekannten Videodatenströmen überprüft. Auch hierbei konnte gezeigt werden, dass das System zur Schätzung der Blickrichtung in Echtzeit geeignet ist.

In Kapitel 5 wurde als drittes Teilmodul ein System zur Erkennung des Gesichtsausdrucks auf Frontalbildern vorgestellt. Auch hierbei kommt wie bei den anderen Teilen ein mehrstufiges Verfahren zum Einsatz. Die Initialdetektion erfolgt auch hier mit Hilfe des

*Viola-und-Jones Gesichtsdetektors* und anschließend wird ein *Active-Appearance-Modell* an das Gesicht angepasst. Im Gegensatz zur Blickrichtung kommt hierbei ein Modell mit wesentlich mehr Parametern zum Einsatz, um die Vielfalt der möglichen Gesichtsausdrücke abbilden zu können. Auch in diesem Teilsystem werden nur die Parameter bei der eigentlichen Mimikschätzung weiterverwendet, die auch einen entsprechenden Informationsgehalt besitzen. Zur Auswertung der Leistungsfähigkeit wurde einerseits eine Klassifikation bekannter Gesichtsausdrücke auf Basis der sechs Basisemotionen untersucht und andererseits die Repräsentation in einem kontinuierlichen 2D-Emotionsraum. Zur Abbildung in den zweidimensionalen Emotionsraum wurde eine *Kohonen Map* als topologieerhaltende Abbildung gewählt. Im Rahmen dieser Dissertation konnte gezeigt werden, dass sich bei der Verwendung des kontinuierlichen Emotionsraums die, aus der Psychologie bekannte Lage von wichtigen Emotionsklassen zueinander, auch in der relativen Lage der entsprechenden Regionen in der *Kohonen Map* wiederfindet. Nach Kenntnisstand des Autors wurde dies bisher noch in keiner anderen Arbeit gezeigt. Das vorgestellte System ist in der Lage, unter den gegebenen Randbedingungen den Gesichtsausdruck eines vor ihm befindlichen Interaktionspartners auf Frontalbildern zu schätzen.

Für alle drei Teilsysteme gilt die Einschränkung, dass eine korrekte Schätzung der gesuchten Parameter nur dann vorgenommen werden kann, wenn auch die Modellanpassung korrekt ist. Bei schlecht angepassten Modellen ist der Informationsgehalt der relevanten Parameter stark gestört und damit die Schätzung sehr fehlerbehaftet. Es gibt zwei wichtige Grundvoraussetzungen für eine hinreichend gute Modellanpassung. Einerseits sind leistungsfähige Anpassungsalgorithmen notwendig, die auch mit Störungen im Input umgehen können. Andererseits muss das zum Training verwendete Datenmaterial die gesamte Bandbreite des später möglichen Inputs abdecken, da sonst die generierten Modellparameter nicht ausreichend sind, um das Modell hinreichend genau an den Input anpassen zu können, weil dieser in dem Modell nicht repräsentiert ist.

Die eigentliche Schätzung der Aufmerksamkeit bzw. des Interaktionsinteresses wurde in Kapitel 6 vorgestellt. Hierbei wurde zunächst untersucht, wie aus den einzelnen Messgrößen (Oberkörperorientierung, Blickrichtung und Gesichtsausdruck) ein Maß für das Interaktionsinteresse bzw. die Aufmerksamkeit gewonnen werden kann. Um mit den dabei gegebenen Unsicherheiten umgehen zu können, wurden hierzu mehrere *Bayes-Filter* eingesetzt. Für jedes Teilsystem wurde dies anhand von Beispielen nachgewiesen und im Rahmen dieser Dissertation präsentiert. Das Gesamtsystem wurde auf einem realen Robotersystem aus dem Projekt ALIAS evaluiert. Dazu wurden die Teilmodule innerhalb der in [Martin et al., 2005b] vorgestellten Software-Architektur implementiert. Für die Tests wurden Daten mit eingewiesenen Probanden aufgenommen und diese nachträglich in Bezug auf Oberkörperpose, Blickrichtung und Gesichtsausdruck ausgewertet und darauf basierend das Maß für das Interaktionsinteresse bzw. die Aufmerksamkeit gewonnen. Die dabei erreichten

Ergebnisse wurden mit denen eines menschlichen Betrachters verglichen, wobei im Rahmen der durch das Robotersystem und das Szenario gegebenen Einschränkungen eine hohe Übereinstimmung erreicht wurde.

Somit konnte gezeigt werden, dass das vorgestellte Gesamtsystem in der Lage ist, die gesuchte Information bezüglich des Interaktionsinteresses bei der Kontaktanbahnung und der Aufmerksamkeit während der Interaktion zu bestimmen. Diese Informationen können im Rahmen eines adaptiven Dialogsystems genutzt werden, um die Interaktion zwischen Benutzer und Roboter anzupassen und das gewünschte Ziel der zugrunde liegenden Anwendung zu erreichen. Ein solches situationsadaptives Dialogsystem ist nicht Bestandteil dieser Dissertation.

## 7.2 Erweiterungsmöglichkeiten und Ausblick

Auf Grundlage der vorliegenden Arbeit gibt es eine Reihe von möglichen Anknüpfungspunkten und Erweiterungsmöglichkeiten für zukünftige Arbeiten, von denen im Folgenden einige kurz benannt werden sollen. Als erste Gruppe gehören dazu Optimierungsmöglichkeiten für die drei Teilsysteme, um deren Erkennungsleistung zu verbessern. Zum Beispiel:

- Erhöhung der Robustheit bei der Anpassung der Active-Shape- und Active-Appearance-Modelle: Eine gute Modellanpassung ist eine Grundvoraussetzung für die anschließende Schätzung der gesuchten Größe. In [Stricker et al., 2009] wurden bereits eine Reihe von Verbesserungsmöglichkeiten evaluiert, die teilweise auch im Rahmen dieser Dissertation genutzt wurden.

- Berücksichtigung von Verdeckungen: Bei allen drei Teilsystemen können Verdeckungen auftreten. Beim Oberkörper kann dies zum Beispiel eine vorbeilaufende Person sein. Bei der Blickrichtung oder der Mimikschätzung können Selbstverdeckungen bei Profilgesichtern oder Verdeckungen durch andere Objekte (wie eine Brille) auftreten. Solche Verdeckungen werden im Rahmen dieser Dissertation nicht behandelt. In der Literatur wird beispielsweise in [Theobald et al., 2006] und [Gross et al., 2006] die Behandlung von Verdeckungen bei Active-Appearance-Modellen untersucht.

- Verwendung von Farbinformationen: Im Rahmen dieser Dissertation wurde fast ausschließlich auf Grauwertbildern gearbeitet. Farbbilder enthalten mehr Informationen, die beispielsweise zur besseren Trennung von Vorder- und Hintergrund bei der Detektion des Oberkörpers eingesetzt werden können.

- Verwendung anderer Sensorik: Wenn die gewählte Randbedingung der Verwendung von Standardkameras aufgegeben wird und auch z.B. die Verwendung moderner 3D-Sensorik erlaubt ist, dann sollten auch dreidimensionale Modelle eingesetzt werden, um

die gewonnenen Sensorinformationen optimal auszunutzen. In diesem Umfeld existieren bereits Algorithmen, die auf deren Tauglichkeit für die hier vorgestellte Anwendung untersucht werden müssen. Einige Ansätze finden sich beispielsweise in [Girshick et al., 2011] und [Shotton et al., 2011].

Im Rahmen eines zweiten wichtigen Komplexes zukünftiger Arbeiten gilt es, den Zusammenhang zwischen den gewählten Merkmalen und der Interaktionsbereitschaft bei der Kontaktanbahnung oder der Aufmerksamkeit während der Interaktion nachzuweisen. Hierzu sind eine Reihe von sozialwissenschaftlichen Untersuchungen in den gewählten Anwendungen notwendig. Die in dieser Dissertation vorgestellten Verfahren und Methoden können als Grundlage für solche Untersuchungen eingesetzt werden. Im Rahmen dieser sollten u.a. folgende Fragestellungen untersucht werden:

- Verhalten gegenüber einem Roboter: In dieser Dissertation wird davon ausgegangen, dass sich die Nutzer gegenüber einem mobilen Serviceroboter prinzipiell genauso verhalten, wie in der zwischenmenschlichen Kommunikation. Trifft diese Annahme immer zu? Ist das Verhalten möglicherweise abhängig vom Roboter, dessen Gestalt oder auch dem Anwendungsszenario?

- Parametrisierung der Verfahren: Die vorgestellten Methoden und Verfahren können nur gut funktionieren, wenn diese hinreichend gut parametrisiert sind. Im Rahmen dieser Dissertation wurden die Parameter als Designer-Wissen auf Basis von praktischen Gegebenheiten eingebracht. Zukünftige sozialwissenschaftliche Untersuchungen sollten Parameter ermitteln, die allgemeingültig eingesetzt werden können und solche Parameterkonfigurationen, die in speziellen Szenarien notwendig sind.

- Eignung und Gewichtung der ausgewählten Merkmale: Es ist zu untersuchen, ob die Oberkörperpose, die Blickrichtung und der Gesichtsausdruck die geeigneten Merkmale sind, um das Interaktionsinteresse oder die Aufmerksamkeit zu schätzen. Weiterhin kann davon ausgegangen werden, dass nicht alle Merkmale den gleichen Informationsgehalt haben. Welche sind also am zuverlässigsten oder am eindeutigsten?

- Anpassung der Anwendung: Letztendlich sollte auch untersucht werden, welche Änderungen am Dialog oder der Anwendung das Interesse bzw. die Aufmerksamkeit eines Benutzers am besten beeinflussen können.

## 7.3 Fazit

Das Schaffen neuer methodischer Grundlagen war bewusst kein Ziel dieser Dissertation. Stattdessen sollten bekannte Verfahren und Techniken innerhalb des gewählten methodischen Frameworks so erweitert und anpasst werden, dass sie zur Schätzung des Interaktionsinteresses und der Aufmerksamkeit bei der Mensch-Roboter-Interaktion geeignet

sind. Der wesentliche Schwerpunkt dieser Dissertation lag somit in der Integration der Teilmodule innerhalb eines Gesamtszenarios auf einem mobilen Robotersystem unter Realwelt-Bedingungen. Um die Anforderungen zu erfüllen, wurden verschiedene Teilverfahren erweitert oder mit anderen bekannten Methoden ergänzt.

Eine solche Kombination der Verfahren mit dem Ziel der Schätzung des Interaktionsinteresses und/oder der Aufmerksamkeit ist nach Kenntnisstand des Autors bisher in keiner anderen Arbeit präsentiert wurden.

Die Frage, wie schnell und in welcher Form mobile Roboter zukünftig im Alltag anzutreffen sein werden, die die Grundregeln der zwischenmenschlichen Kommunikation berücksichtigen, konnte in dieser Dissertation nicht beantwortet werden. Jedoch wurde eine Lösungsmöglichkeit für ein Teilproblem aus diesem spannenden Forschungsgebiet gezeigt. Es bleibt zu hoffen, dass hier zukünftig weitere Fortschritte erzielt werden, um in nicht all zu ferner Zukunft mobile Serviceroboter vorzufinden, die das menschliche Alltagsleben wirklich erleichtern können.

# A Details zu Klassifikatoren und Funktionsapproximatoren

## A.1 Nearest Neighbour Klassifikation

Die Nearest-Neighbour-Klassifikation ist eine einfache Methode der Klassifikation von Daten. Sie beruht darauf, die Merkmale eines Objektes einer unbekannten Klasse mit den Merkmalen von Objekten, deren Klassenzugehörigkeit bekannt ist, zu vergleichen. Die Klasse des Objektes, das dem unbekannten Objekt am meisten ähnelt, wird dann übertragen auf das unbekannte Objekt. Sofern die Ausprägungen der Merkmale eines Objektes numerisch bekannt sind, kann ein Distanzmaß benutzt werden, um die Ähnlichkeit der Objekte zu bestimmen. In dieser Dissertation wird dafür die euklidische Distanz $d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ für den Abstand zweier Punkte $\mathbf{x} = (x_1, ..., x_n)$ und $\mathbf{y} = (y_1, ..., y_n)$ benutzt. Ein Beispiel für die Vorgehensweise der Nearest-Neighbour-Klassifikation ist in Abbildung A.1 dargestellt. Im Merkmalsraum, bestehend aus zwei Merkmalen, wird die euklidische Distanz des zu klassifizierenden Objektes zu allen bekannten Objekten überprüft und dann die Klasse des nächstliegenden Objektes auf das unbekannte Objekt übertragen.

Nearest-Neighbour-Klassifikation ist dabei eine reine Klassifikationsmethode, mit der sich keine Funktionsapproximation durchführen lässt. Gehört ein Objekt beispielsweise zur Klasse 3 und ein anderes Objekt zur Klasse 1, so wird für ein unbekanntes Objekt, das zwischen diesen beiden Objekten liegt, die Klasse 1 oder 3 gewählt, nicht aber die Klasse 2 als ein Mittel aus beiden. Für kontinuierliche Probleme (z.B. eine Winkelschätzung) wird jedoch genau eine solche Funktionsapproximation benötigt. Mit einer einfachen Nearest-Neighbour-Klassifikation ist dies nicht möglich.

Ein erster Schritt zur Lösung des Problems hierzu ist die *k-Nearest-Neighbour-Methode*. Das Prinzip der Nearest-Neighbour-Klassifikation wird dabei so abgewandelt, dass nicht nur das dem zu klassifizierenden Objekt nächste Objekt betrachtet wird, sondern die $k$ nächsten Objekte. Für $k = 3$ würden also die 3 nächsten Nachbarn bestimmt und die Klasse, die unter diesen drei Nachbarn am häufigsten auftritt, wird dem unbekannten Objekt zugewiesen. Damit gewinnt das Klassifikationsverfahren Stabilität gegenüber Ausreißern, es kann aber immer noch keine Funktionsapproximation durchgeführt werden.

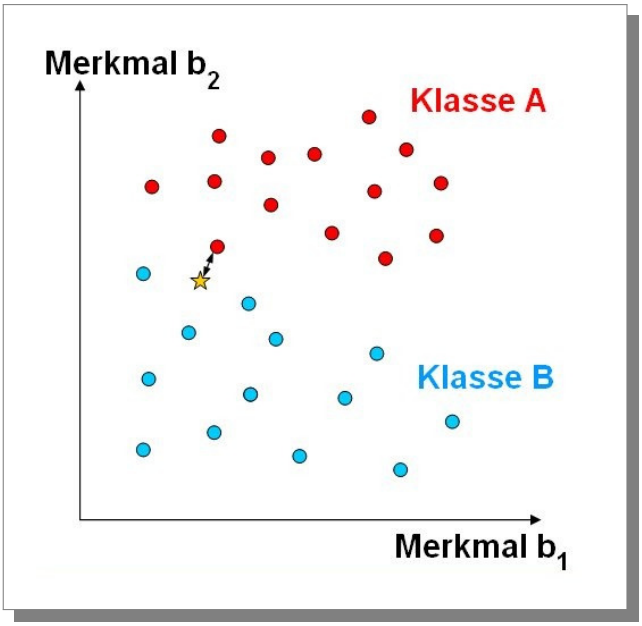



***Abbildung A.1:*** *Nearest-Neighbour-Klassifikation. Im Merkmalsraum sind Beispiele aus den zwei Klassen A und B vorhanden. Die Klasse eines neuen Objektes (im Bild als gelber Stern) wird bestimmt, indem das nächste Beispielobjekt gesucht und dessen Klasse darauf übertragen wird. Im abgebildeten Fall würde das neue Objekt zur Klasse A gezählt.*

Durch eine leichte Modifikation an der $k$-Nearest-Neighbour-Methode lässt sich dies erreichen: man betrachtet immer noch die $k$ nächsten Nachbarn, die Klasse des unbekannten Objektes wird allerdings nicht mehr nach der am häufigsten auftretenden Klasse unter den $k$ Nachbarn bestimmt, sondern aus dem Mittelwert der auftretenden Klassen.

Noch genauer wird diese Methode, wenn statt des normalen Mittelwertes ein gewichteter Mittelwert benutzt wird, der die Distanz der $k$ Nachbarn zu dem unbekannten Objekt bei der Bildung des Funktionswertes berücksichtigt:

$$f_k(\mathbf{x}) = \sum_i c_i \cdot \left( \frac{1/d_i}{\sum_j 1/d_j} \right) \tag{A.1}$$

Das heißt, der Funktionswert $f_k(\mathbf{x})$ für $k$ Nachbarn zu einem Input $\mathbf{x}$ ergibt sich aus der Summe ihrer Klassen $\sum_i c_i$ gewichtet mit der Distanz $d_i$ des jeweiligen Nachbarn gegenüber der Gesamtdistanz aller Nachbarn $\sum_j d_j$. Damit der Nachbar mit der kleinsten Distanz das größte Gewicht erhält und die Summe aller Gewichte 1 ergibt, muss jeweils der Kehrwert der Distanz benutzt werden.

Dieses Klassifikationsverfahren wird in Abbildung A.2 noch einmal graphisch verdeutlicht, allerdings ohne Anwendung eines gewichteten Mittelwertes. Zusätzliche Informationen zur Nearest-Neighbour-Klassifikation und der $k$-Nearest-Neighbour-Methode finden sich auch in [Duda and Hart, 1973] und [Jafar-Shaghaghi, 1994].

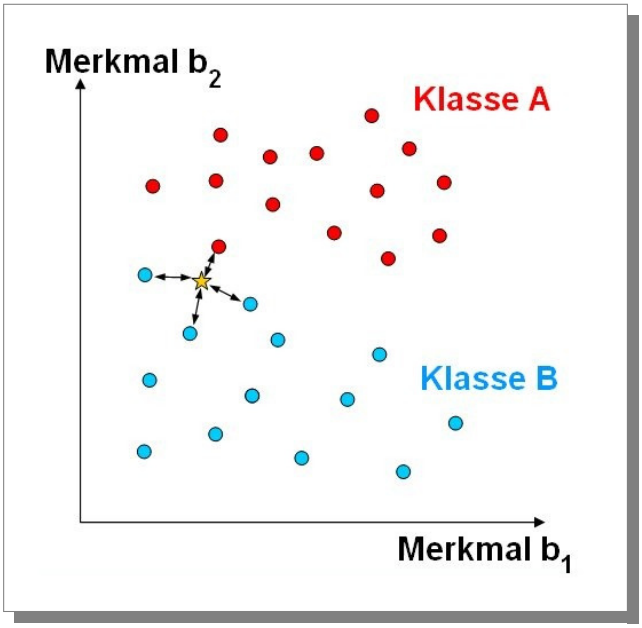



***Abbildung A.2:*** *Anwendung der k-Nearest-Neighbour-Klassifikation (im Beispiel für k=4). Im Gegensatz zur Nearest-Neighbour-Klassifikation werden die vier nächsten Nachbarn des zu klassifizierenden Objektes bestimmt. Das nächstliegende Objekt gehört zwar zur Klasse A, aber die drei anderen Objekte gehören zur Klasse B und sind nur geringfügig weiter vom Objekt entfernt. Im abgebildeten Fall würde das neue Objekt deshalb zur Klasse B gezählt.*

## A.2 Multi Layer Perceptron

Ein *Multi Layer Perceptron (MLP)* ist ein mehrschichtiges, einfach verkoppeltes, neuronales Netzwerk. In Abbildung A.3 ist ein Beispiel für ein solches Netzwerk zu sehen. Es hat drei Eingabevariablen und drei Ausgabevariablen bei insgesamt drei Schichten. Diese Schichten sind die Eingabeschicht, die Hiddenschicht und die Ausgabeschicht. Sie sind jeweils vollständig mit der vorangegangenen Schicht verbunden.

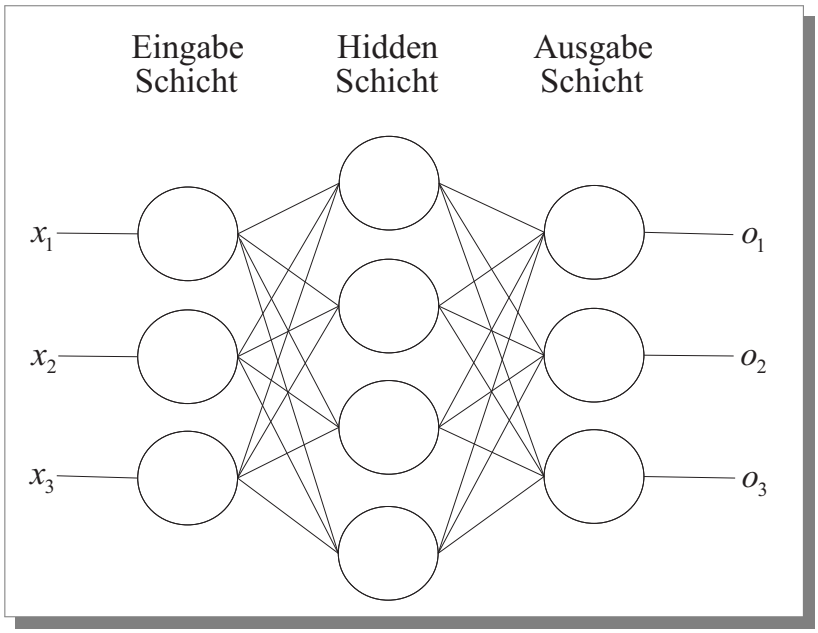



***Abbildung A.3:*** *MLP mit drei Eingabeneuronen $x_1$ bis $x_3$, vier Neuronen in der Hiddenschicht und drei Ausgabeneuronen $o_1$ bis $o_3$.*

Die Abbildung A.4 zeigt das Modell eines einzelnen künstlichen Neurons. Dieses besteht im Wesentlichen aus einer Aktivierungsfunktion $z = z(x)$ und einer Transfer- oder Ausgabefunktion $y = y(z)$.
Als Aktivierungsfunktion wird zumeist eine Skalarproduktaktivierung mit einer Gewichtsmatrix $\mathbf{W}$ verwendet. Mit dieser Gewichtsmatrix werden die Eingangsverbindungen der Neuronen gewichtet und das Neuron $j$ wird damit folgendermaßen aktiviert:

$$z\left(x\right) = \sum_{i=1}^{n} x_i w_{ij} \tag{A.2}$$

Als Transferfunktion kann zum Beispiel eine Sigmoidfunktion für Ausgabewerte im Intervall [0..1] oder die Tangens-Hyperbolicus Funktion für Ausgabewerte im Intervall $[-1..1]$ genutzt werden. Bei diesen beiden Transferfunktionen gibt es zusätzlich einen Schwellwert $T$, mit dem die Aktivierungswerte verschoben werden können. Mit der Tangens-Hyperbolicus Funktion

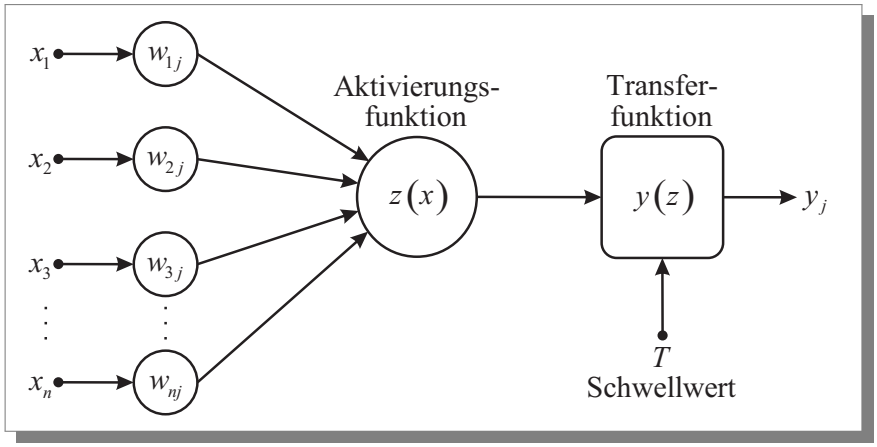



***Abbildung A.4:*** *Modell des Neurons $j$: $\mathbf{x} = (x_1, \ldots, x_n)^T$ ist der Eingabevektor, der mit der Gewichtsmatrix $\mathbf{W} = (w_{ij})$ gewichtet wird. Mit der Funktion $z(\mathbf{x})$ wird der Aktivierungswert des Neurons berechnet. Die Ausgabe $y_j$ wird aus dem Aktivierungswert und der Transferfunktion $y(z)$ mit dem Schwellwert $T$ berechnet.*

ist die Ausgabefunktion

$$y(z) = \tanh(z - T) = \frac{e^{2(z-T)} - 1}{e^{2(z-T)} + 1} \tag{A.3}$$

Der gesamte Ablauf in einer bestimmten Schicht $l$ könnte zum Beispiel folgendermaßen aussehen: Der Eingabevektor $\mathbf{x}^{(l)}$ wird mit der Gewichtsmatrix $\mathbf{W}^{(l)}$ gewichtet und die Neuronen werden mit der Skalarproduktaktivierung $\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{x}^{(l)}$ aktiviert. Die Ausgabe berechnet sich mit dem Ausgabegewichtsvektor mit $\mathbf{y}^{(l)} = \tanh\left(\mathbf{z}^{(l)} - \mathbf{T}^{(l)}\right)$, wobei der Aktivierungsvektor $\mathbf{z}^{(l)}$ mit dem Schwellwertvektor $\mathbf{T}^{(l)}$ verschoben wird. Bei der nächsten Schicht $l+1$ wird die Ausgabe $\mathbf{y}^{(l)}$ zur Eingabe $\mathbf{x}^{(l+1)}$ und die Berechnung erfolgt genauso. Zusammengefasst ergibt sich:

$$\mathbf{x}^{(l+1)} = \tanh\left(\mathbf{W}^{(l)}\mathbf{x}^{(l)} - \mathbf{T}^{(l)}\right) \tag{A.4}$$

Es gibt verschiedene Algorithmen, mit denen ein MLP trainiert werden kann. Das bekannteste Verfahren ist der *Backpropagation-Algorithmus*, bei dem zu jedem Eingabevektor des Trainingsdatensatzes der gewünschte Ausgabevektor des MLPs bekannt ist. Dieser Vektor wird Teachvektor $\mathbf{t}$ genannt. Dabei wird der Eingabevektor $\mathbf{x}$ an das MLP angelegt und der Fehlerwert $E$ zwischen dem Teachvektor $\mathbf{t}$ und der tatsächlichen Netzwerkausgabe $\mathbf{o}$ gebildet.

Dafür wird zumeist die Summe der quadratischen Fehler gewählt:

$$E = \frac{1}{2} \|\mathbf{t} - \mathbf{o}\|_2^2 = \frac{1}{2} \sum_{j=1}^{n} (t_j - o_j)^2 \tag{A.5}$$

Der Faktor $\frac{1}{2}$ dient nur zur Vereinfachung bei der Ableitung. Dieser Fehler $E$ soll nun durch die folgende Anpassungsregel der einzelnen Kantengewichte $w_{ij}$ minimiert werden:

$$w_{ij}^{neu} = w_{ij}^{alt} + \Delta w_{ij} \tag{A.6}$$

Dazu wird die Fehlerfunktion (A.5) nach den einzelnen Kantengewichten $w_{ij}$ partiell abgeleitet. Die entgegengesetzte Richtung dieses Gradienten ist die Richtung des steilsten Abstiegs. Damit ist

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}}.$$

Dabei ergibt die Ableitung der Gleichung (A.5) $\frac{\partial E}{\partial y_j} = (t_j - o_j)$, was dem Fehlerwert des Neurons $j$ entspricht. Die Ableitung der Ausgabefunktion (A.3) nach dem Aktivierungswert ist $\frac{\partial y_j}{\partial z_j} = y'(z_j)$ und die Ableitung der Aktivierungsfunktion (A.2) nach dem Kantengewicht ergibt $\frac{\partial z_j}{\partial w_{ij}} = 1$. Daraus kann die folgende Vorschrift für die Anpassung der Kantengewichte gebildet werden:

$$w_{ij}^{neu} = w_{ij}^{alt} - \eta \delta_j x_i. \tag{A.7}$$

Dabei ist $\eta$ die Lernrate und

$$\delta_j = \begin{cases} y'(z_j)(t_j - o_j) & \text{falls } j \text{ Ausgabeneuron ist} \\ y'(z_j) \sum_{k=1}^{m} \delta_k w_{jk} & \text{falls } j \text{ Neuron einer Hiddenschicht ist} \end{cases}$$

wobei $m$ die Anzahl der Neuronen der vorangegangenen Schicht ist.

In der Praxis gibt es weitere Lernalgorithmen und auch zahlreiche Erweiterungen zum Backpropagation Verfahren, wie zum Beispiel den Momentum Lernalgorithmus. Diese sollen jedoch an dieser Stelle nicht weiter betrachtet werden. Weitere ausführliche Details zum MLP finden sich beispielsweise in [Zell, 1994].

Im Rahmen dieser Dissertation wurde für die MLPs die Implementierung der *Fast Artifical Neural Network Library (FANN)* [FANN, 2010] verwendet.

## A.3 Support Vector Machine

Eine *Support Vector Machine (SVM)* ist ein Klassifikator, mit dem eine Menge von Objekten in zwei Klassen unterteilt werden. Dabei geschieht diese Unterteilung so, dass um die Klassengrenze herum ein möglichst breiter Bereich frei von Objekten bleibt. Die Objekte werden dabei durch Vektoren in einem Eingaberaum repräsentiert. Abbildung A.5 zeigt die Trennung zweier Klassen von Objekten mittels Hyperebenen. Dabei sind die Hyperebenen in den Eingaberaum projiziert und können daher gekrümmt oder auch unterteilt sein.

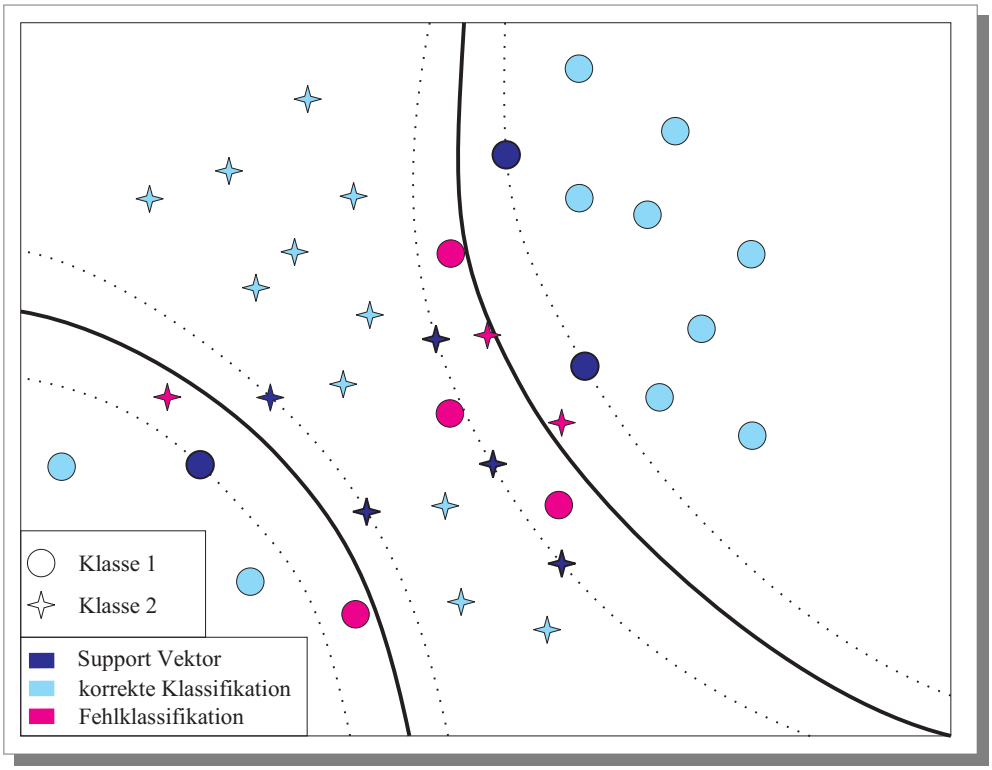



***Abbildung A.5:*** *Support Vektor Maschine zur Klassifikation. Die fett gedruckten Linien markieren die Klassengrenzen. Sie sind die in den Eingaberaum projizierten Hyperebenen des Merkmalsraums. Die gepunkteten Linien zeigen die Trennbreite an. Die dunkelblauen Instanzen auf den gepunkteten Linien sind die Support Vektoren.*

Gegeben sei die folgende endliche Trainingsmenge $L$:

$$L = \left\{ \left( \mathbf{f}^i, g^i \right) | i = 1, ..., M \right\} \tag{A.8}$$

mit $\mathbf{f}^i \in \mathbb{R}^r$ der Objektvektoren im Eingaberaum und $g^i \in \{+1, -1\}$ der Klassenzugehörigkeit. Die $\mathbf{f}^i$ mit $g^i = +1$ sind die positiven Instanzen und die $\mathbf{f}^i$ mit $g^i = -1$ sind die negativen Instanzen.

Gesucht wird eine durch den Vektor $\mathbf{w} \in \mathbb{R}^r$ und den Bias $b$ beschriebene Hyperebene $h$ mit

$$h : h(\mathbf{w}, b) = \left\{ \mathbf{f} \in \mathbb{R}^r \,\middle|\, \mathbf{w}^T \mathbf{f} + b = 0 \right\}, \tag{A.9}$$

die die positiven von den negativen Instanzen trennt. Der vorzeichenbehaftete Abstand $d$ eines Punktes $f$ von der Hyperebene $h$ ist

$$d = \frac{\mathbf{w}^T \mathbf{f} + b}{\|\mathbf{w}\|} \tag{A.10}$$

Die Trennbreite $\mu_L(\mathbf{w}, b)$ der Hyperebene $h$ bezüglich der Trainingsmenge $L$ wird als das Minimum der Beträge der Abstände aller Punkte $\mathbf{f}^1, \ldots, \mathbf{f}^M$ zu $h$ definiert:

$$\mu_F(w, b) = \min_{i=1,\ldots,M} |d_i| = \min_{i=1,\ldots,M} \frac{\left| w^T x^i + b \right|}{\|w\|}. \tag{A.11}$$

Ziel ist es, die optimale Hyperebene $h^* = h(\mathbf{w}^*, b^*)$ zu finden, welche die Objekte der Trainingsmenge $L$ mit einer maximalen Trennbreite $\mu_L(\mathbf{w}, b)$ separiert. Für diese optimale Hyperebene ist klar, dass es eine Menge von positiven Instanzen $\mathbf{f}^i$ mit $\mathbf{w}^T \mathbf{f}^i + b = +1$ und eine Menge von negativen Instanzen $\mathbf{f}^i$ mit $\mathbf{w}^T \mathbf{f}^i + b = -1$ gibt. Diese Instanzen mit der größten Nähe zur Hyperebene $h^*$ werden Support Vektoren von $h^*$ bezüglich $L$ genannt. Die Berechnung der optimalen Hyperebene erfolgt mit Hilfe von Lagrange'schen Multiplikatoren. Auf die Herleitung wird an dieser Stelle verzichtet. Sie kann der Literatur [Burges, 1998] entnommen werden. Nach der Herleitung ist die Funktion

$$Q(\alpha) = -\frac{1}{2} \sum_{i}^{M} \sum_{j}^{M} \left[ \alpha_i \cdot \alpha_j \cdot g^i \cdot g^j \cdot \left( \left( \mathbf{f}^i \right)^T \mathbf{f}^j \right) \right] + \sum_{i}^{M} \alpha_i \tag{A.12}$$

mit den Lagrang'schen Multiplikatoren $\alpha$ zu maximieren unter den Nebenbedingungen

$$\sum_{i=1}^{M} \alpha_{\mathrm{i}} \cdot g^i = 0 \;\text{ und }\; \alpha_1 \geq 0, ..., \alpha_M \geq 0. \tag{A.13}$$

Die Funktion $Q(\alpha)$ ist eine konvexe Funktion und die Nebenbedingungen linear. Daher ist das Maximum von $Q(\alpha)$ leicht zu bestimmen. Mit den optimalen Werten für $\alpha$ ergeben sich die Gewichte $\mathbf{w}$ wie folgt:

$$\mathbf{w} = \sum_{i=1}^{M} \alpha_i \cdot g^i \cdot \mathbf{f}^i. \tag{A.14}$$

Dabei gehen die Trainingsvektoren in die Optimierungsfunktion $Q(\alpha)$ in Gleichung (A.12) nur in Form von paarweisen Skalarprodukten $(\mathbf{f}^i)^T \mathbf{f}^j$ für $i, j = 1, ..., M$ ein.

### A.3.1 Merkmalsraum und Kernel-Funktion

Beliebige Trainingsmengen sind in der Regel nicht linear separierbar. Deshalb wird der Eingaberaum $\mathbb{R}^r$, in dem die Vektoren von $L$ liegen, in einen höherdimensionalen Merkmalsraum $\mathbb{R}^R$ tranformiert. Dabei ist $R \gg r$. Die Abbildung in den Merkmalsraum wird mit

$$\Phi : \mathbb{R}^r \rightarrow \mathbb{R}^R \tag{A.15}$$

bezeichnet. Die Trainingsmenge wird in den Merkmalsraum abgebildet, um die Klassen dort linear zu trennen:

$$\Phi(L) = \{(\Phi(\mathbf{f}^m), d^m) \,|m = 1, ..., M\}. \tag{A.16}$$

Gleichung (A.12) im Eingaberaum wird bei der Abbildung in den Merkmalsraum zu folgendem Optimierungsproblem:

$$Q^{\Phi}(\alpha) = -\frac{1}{2}\sum_{i}^{M}\sum_{j}^{M}\left[\alpha_i \cdot \alpha_j \cdot g^i \cdot g^j \cdot \left(\left(\Phi\left(f^i\right)\right)^T \Phi\left(f^j\right)\right)\right] + \sum_{i}^{M}\alpha_i, \tag{A.17}$$

wobei die Randbedingungen gleich bleiben. Bei der Optimierung müssen nicht die Merkmalsvektoren $\Phi(f^i)$ berechnet werden, sondern nur die paarweisen Skalarprodukte $\Phi(f^i)^T \Phi(f^j)$. Für die Skalarprodukte wird eine sogenannte Kernel-Funktion verwendet, die wie folgt definiert ist:

$$K : \mathbb{R}^r \times \mathbb{R}^r \rightarrow \mathbb{R} \tag{A.18}$$

Für die Abbildungsfunktion $\Phi$ ist diese Kernel-Funktion

$$K^{\Phi}(x, y) = \Phi(x)^T \Phi(y) \tag{A.19}$$

Mit der Kernel-Funktion kann $Q^{\Phi}(\alpha)$ effizient maximiert werden. Die Gewichtsvektoren der Support Vektoren sind dann

$$w = \sum_{i=1}^{M}\alpha_i \cdot g^i \cdot \Phi\left(f^i\right) \tag{A.20}$$

Zentrale Aufgabe beim Entwerfen einer SVM ist es, eine passende Abbildungsfunktion vom Eingaberaum in den Merkmalsraum zu finden und die dazugehörige Kernel-Funktion $K^{\Phi}$ zu bestimmen. Oft verwendete Kernel-Funktionen sind die Gauß-Kernels oder Radial Basis Funktionen (RBF) mit

$$K_{\sigma}(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}, \tag{A.21}$$

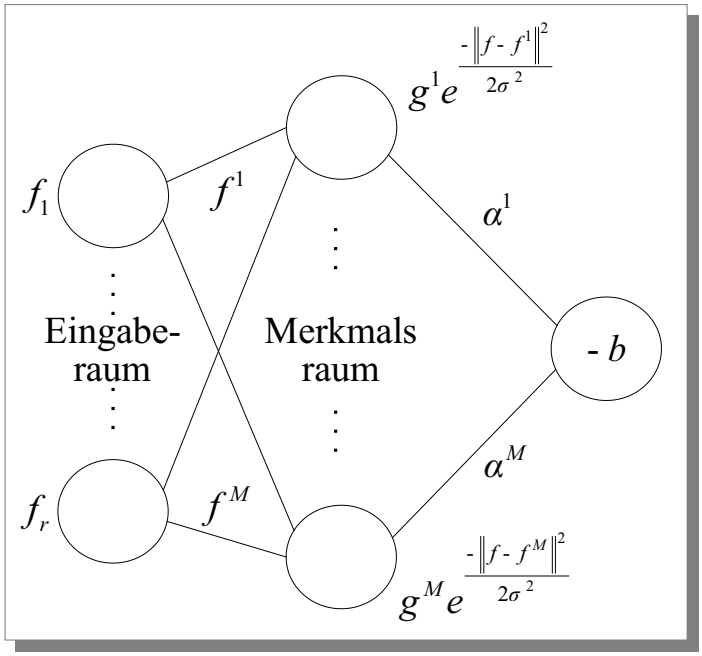



***Abbildung A.6:*** *Support Vector Maschine: Die erste Schicht repräsentiert den Eingaberaum. Die Eingabe* $\mathbf{f} = (f_1, \ldots, f_r)^T$ *wird in den Merkmalsraum der zweiten Schicht abgebildet. Die dritte Schicht trennt die positiven und negativen Instanzen mit Hilfe einer Hyperebene, repräsentiert durch die Koeffizienten* $\alpha$ *und den Bias* $b$*.*

wie sie in der Darstellung der SVM in Abbildung A.6 verwendet werden.
Mit diesen Kernel-Funktionen lassen sich selbst eng ineinander verschachtelte Klassen gut voneinander trennen. Auch mit MLPs lassen sich ineinander verschachtelte Klassen durch entsprechend erhöhte Netzwerkkomplexität voneinander trennen. Bei zu hoher Netzwerkkomplexität kann es bei MLPs jedoch zu Überanpassungen kommen. Eine Überanpassung wird bei SVMs durch einen speziellen Lernalgorithmus vermieden. Für Klassifikationsaufgaben bei unbekannter Verteilung der Klassenexemplare sind SVMs daher gut geeignet.

Im Rahmen dieser Dissertation wurde für die SVMs die Implementierung der *LIBSVM - A Library for Support Vector Machines* [libSVM, 2010] verwendet.

### A.3.2 Mehrklassen-Klassifikation mittels SVMs

Bei SVMs handelt es sich vom Prinzip her nur um Zwei-Klassen-Klassifikatoren. Eine oft verwendete Technik zur Erweiterung auf $k$ Klassen, ist das sogenannte *one-versus-rest* oder *one-versus-all* Verfahren. Dabei werden insgesamt $k$ separate Klassifikatoren erzeugt, die als Trainingsdaten jeweils die Daten einer Klasse und die Menge der Daten sämtlicher anderer

Klassen verwenden. Als Ergebnis wird die Klasse ausgewählt, die einen unbekannten Input mit dem größten Abstand zur Hyperebene klassifiziert.

Eine andere Variante ist das *one-versus-one* Verfahren [Knerr et al., 1990, Kreßel, 1999]. Dabei müssen insgesamt $k(k-1)/2$ Klassifikatoren aus den Daten von je zwei verschiedenen Klassen der Trainingsmenge erstellt werden. Als Ergebnis einer Klassifikation wird dann die Klasse ausgewählt, die von den meisten Klassifikatoren selektiert wird. Obwohl bei diesem Verfahren deutlich mehr Klassifikatoren erstellt werden müssen, ist die Trainingszeit oft kürzer, da die Anzahl der Trainingsdaten für jeden einzelnen Klassifikator deutlich geringer ist. Weitere Varianten für Multi-Klassen-SVMs sind in [Hsu et al., 2003] vorgestellt.

Im Rahmen dieser Dissertation wurde das *one-versus-one* Verfahren aus der *LIBSVM - A Library for Support Vector Machines* [libSVM, 2010] eingesetzt.

## A.4 Self organizing maps - Kohonen Maps

Bei selbstorganisierenden Karten (*Self-organizing maps (SOM)* oder *Kohonen Maps*) handelt es sich um ein unüberwachtes Lernverfahren aus dem Bereich der Neuronalen Netze. Sie können als eine Weiterentwicklung der Verfahren zur Vektorquantisierung (z.B. LVQ) betrachtet werden. Erstmals wurden sie von [Kohonen, 1982] vorgestellt. Die folgende Kurzvorstellung der Kohonen Maps ist sinngemäß entnommen aus [Zell, 1994].

### A.4.1 Grundidee Kohonen Maps

Bei den Kohonen Maps handelt es sich um ein einschichtiges neuronales Netz, dass mit einem unüberwachten Lernverfahren trainiert wird. Das Netz selbst besteht aus einer Menge von Neuronen, deren Gewichtsvektoren $\mathbf{w}_i$ sich in einem $n$-dimensionalen Raum so verteilen sollen, dass sie diesen entsprechend der Verteilung der Inputdaten möglichst gut abdecken. Gleichzeitig wird eine Nachbarschaft zwischen den Neuronen definiert. Typischerweise kommt hier ein $m$-dimensionales Gitter zum Einsatz.
Mit Hilfe dieser Nachbarschaftsbeziehung soll die Vektorquantisierung des $n$-dimensionalen Inputraums so durchgeführt werden, dass die durch das $m$-dimensionale Gitter definierte Ordnung erhalten bleibt. Benachbarte Eingabevektoren sollen auch auf benachbarte Neuronen im Gitter abgebildet werden. Somit soll eine topologieerhaltende Abbildung vom $\mathbb{R}^n$ nach $\mathbb{R}^m$ erreicht werden.

### A.4.2 Lernverfahren der Kohonen Maps

Für einen neuen Input $\mathbf{x}$ erfolgt folgende Anpassung der Kohonen Maps. Zunächst wird das Neuron gesucht, dessen Gewichtsvektor am ähnlichsten zu $\mathbf{x}$ ist:

$$b = \arg\min_i \left( \|\mathbf{x} - \mathbf{w}_i\| \right) \tag{A.22}$$

Dieses Neuron $b$ wird auch als *Best-matching Neuron* bezeichnet. Als Vergleichsfunktion kann hierfür jede beliebige Norm eingesetzt werden. In der Praxis verwendet man am häufigsten die euklidische Norm.

Anschließend erfolgt die Anpassung der Gewichte sämtlicher Neuronen wie folgt:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t) \cdot \mathrm{h}_{b,i}(t) \cdot (\mathbf{x} - \mathbf{w}_i(t)) \tag{A.23}$$

Dabei ist $\eta(t)$ eine zeitlich veränderliche Lernfunktion (typischerweise eine monoton fallende Funktion mit $0 < \eta(t) < 1$) und $h_{b,i}(t)$ eine veränderliche Nachbarschaftsfunktion mit

$$\mathrm{h}_{b,i}(t) = \mathrm{h}\left( \|\mathbf{p}_b - \mathbf{p}_i\|, t \right) = \mathrm{h}(z, t) \tag{A.24}$$

wobei $\mathbf{p}_b$ und $\mathbf{p}_i$ die Vektoren der Positionen der Neuronen $b$ und $i$ innerhalb der Gitterstruktur des Netzes beschreiben. Die Nachbarschaftsfunktion h sollte so gewählt werden, dass mit zunehmendem Abstand $z = \|\mathbf{p}_b - \mathbf{p}_i\|$ gilt: $\mathrm{h}_{b,i}(z,t) \to 0$.

Je näher ein Neuron $i$ innerhalb der definierten Gitterstruktur also zum *Best-matching Neuron* $b$ liegt, desto stärker wird dieses Neuron in Richtung des Inputs $\mathbf{x}$ verschoben. Je weiter weg das Neuron liegt, desto weniger wird es in Richtung $\mathbf{x}$ bewegt.

Anstatt einer zeitabhängigen Nachbarschaftsfunktion $\mathrm{h}(z,t)$ kann auch eine Nachbarschaftsfunktion mit veränderlichem Radius $\mathrm{h}(z,d)$ eingesetzt werden. Dabei definiert der Radius $d$ die Größe des Nachbarschaftsradius innerhalb des Gitters.

Eine typische Nachbarschaftsfunktion ist die Gauß-Funktion:

$$\mathrm{h}_{\mathrm{gauss}}(z,t) = \frac{1}{\sigma(t)\sqrt{2\pi}} \cdot \mathrm{e}^{-z^2/(2\sigma^2(t))} \tag{A.25}$$

bzw.

$$\mathrm{h}_{\mathrm{gauss}}(z,d) = \mathrm{e}^{(-z/d)^2} \tag{A.26}$$

Nach dem Training einer Kohonen Map muss diese noch kalibriert werden. Dabei handelt es sich um einen anwendungsabhängigen Vorgang, bei dem beobachtet wird, auf welches Neuron bekannte Inputs abbilden. Auf diese Weise kann eine Art von Koordinatensystem oder eine Reihe von Referenzpunkten festgelegt werden, die bei der Interpretation von unbekannten Inputs helfen sollen (z.B. durch Interpolation oder Extrapolation).

# B Weitere Verfahren zur Gesichtsdetektion

Der Mensch erkennt scheinbar spielend Gesichter in einer Vielzahl von Umgebungen, unter schlechten Lichtverhältnissen und in großer Entfernung. Nach [Russell and Fernandez-Dols, 1997] sind Schwarz-Weiß-Bilder mit einer Größe von $10 \times 10$ Pixeln im Allgemeinen die untere Grenze, bei der ein Mensch noch Gesichter erkennen kann (siehe Abbildung B.1). Zudem erkennt der Mensch das Gesicht als Ganzes und nicht als eine Menge von einzelnen Merkmalen. Die geometrische Lage, in der die einzelnen Merkmale zueinander in Beziehung stehen, ist wichtiger, als die Details der einzelnen Merkmale [Bruce, 1988]. Sind Teile des Gesichts verdeckt, zum Beispiel durch Hand, Brille oder Haare, so ist der Mensch trotzdem in der Lage, das Gesicht als Ganzes wahrzunehmen und dabei die fehlenden Teile zu ersetzen. Der Mensch ist daher der Maßstab, an dem sich automatische Gesichtsdetektoren messen lassen müssen.

***Abbildung B.1:*** *Ein Gesicht in verschiedenen Auflösungsstufen:* $100 \times 100$, $50 \times 50$, $25 \times 25$ *und* $13 \times 13$ *Pixel. Selbst in der geringsten Auflösung kann man das Gesicht noch recht deutlich erkennen.*

Automatische Gesichtsdetektoren arbeiten in der Regel nur innerhalb zuvor festgelegter Bedingungen. Grundsätzlich kann man zwischen merkmalsbasierten und modellbasierten Ansätzen bei Gesichtsdetektoren unterscheiden. Bei merkmalsbasierten Methoden werden bestimmte unveränderliche Gesichtsmerkmale, wie Kanten oder Helligkeitsverteilungen genutzt, um Gesichter zu erkennen. Bei modellbasierten Verfahren wird dagegen im Voraus ein Modell erzeugt, welches mit möglichen Gesichtern verglichen wird. Über Schwellwerte eines definierten Ähnlichkeitsmaßes werden damit Gesichter erkannt.

Ein modellbasierter Ansatz wird zum Beispiel in [Pantic and Rothkrantz, 2000b] verfolgt. Dabei wird ein Punktverteilungsmodell verwendet, um ein Gesicht in einem Bild zu repräsentieren. Um dieses Modell zu initialisieren, wird ein Canny Kantendetektor verwendet,

mit dem die Intensitätskante zwischen den Lippen und die beiden symmetrischen, vertikalen Kanten der seitlichen Kopfränder erkannt wird. Haare im Gesicht und Brillen stören dabei die Erkennung erheblich. Außerdem dürfen keine starken Kopfbewegungen oder Beleuchtungsschwankungen auftreten.

In [Pantic and Rothkrantz, 2000a] wird das Gesicht über eine Histogrammanalyse mit bestimmten Schwellwerten lokalisiert. Dabei muss jede Testperson einen Helm mit einer darauf montierten Kamera tragen, die das Gesicht der Testperson aufnimmt. Auf den Bildern dieser Helmkamera befindet sich das Gesicht daher immer an der gleichen Position. Für einen Realwelteinsatz ist dies nicht praktikabel.

Bei einem merkmalsbasierten Ansatz in [Kobayashi and Hara, 1997] wird diese Lokalisierung über die Analyse der Helligkeitsverteilung erreicht. Auch hier dürfen keine beliebigen Kopfbewegungen auftreten, was die praktische Anwendbarkeit drastisch reduziert.

In beliebigen Bildern ist die Erkennung von Gesichtern deutlich schwieriger. In [Hong et al., 1998] wird ein System namens *PersonSpotter* [Steffens et al., 1998], welches ein Rechteck als Kopfumrandung liefert, für eine Echtzeitverfolgung eingesetzt. Dieses Rechteck wird für die Initialisierung eines *Labeled Graph* verwendet, der dann genauer an das Gesicht angepasst wird. Mit Hilfe von Stereo Algorithmen werden die Kopfbewegungen in aufeinanderfolgenden Bildern ausgewertet. Die aktuelle Position und Geschwindigkeit wird mittels eines linearen Filters vorhergesagt. In [Steffens et al., 1998] wird angegeben, dass das System auch bei bewegtem Hintergrund erfolgreich arbeitet, allerdings bei verdeckten oder in Pose gedrehten Gesichtern scheitert.

In [Essa and Pentland, 1997] werden lineare Unterräume (nach der sogenannten Eigenspace-Methode von [Pentland et al., 1994]) verwendet, um Gesichter in beliebigen Szenen zu lokalisieren. Mit einer Hauptkomponentenanalyse aus Trainingsgesichtern wird ein sogenannter Gesichtsraum aus den Eigenvektoren der Kovarianzmatrix gebildet. Um Gesichter in statischen Bildern zu erkennen, wird der Abstand des Eingabebildes zu dem Gesichtsraum bestimmt. In Bildsequenzen wird ein spatio-temporaler Filter verwendet, um Bewegungen in Bildregionen zu analysieren. Jede dieser Bildregionen kann einen menschlichen Kopf beinhalten und wird wie ein einzelnes Bild behandelt. Diese Methode funktioniert erfolgreich auf Bildern mit frontalen und unverdeckten Gesichtern.

In den bisher vorgestellten Arbeiten werden die Bilder oder Bildsequenzen grundsätzlich in einer kontrollierten Umgebung aufgenommen. Damit sind zum einen schlechte Beleuchtungsverhältnisse ausgeschlossen, zum anderen befinden sich die aufgenommenen Gesichter in einer nahezu frontalen Ansicht oder es ist zumindest die Position und Größe durch Markierung in irgendeiner Form im Voraus bekannt. Im Allgemeinen ist das Erkennen von Gesichtern

ein komplexes Problem. Bei feststehenden Kameras können Gesichter in verschiedenen Größen und Winkeln in den Bildern auftauchen. In Realweltanwendungen sind wechselnde Beleuchtungsverhältnisse und Glanzeffekte, zum Beispiel durch Sonneneinstrahlung, ein großes Problem. Ein statisches Template für ein ganzheitliches Modell ist daher schwer zu finden. Das Auftreten von Rauschen und Verdeckungen erschwert das Problem zusätzlich.

Abgewandte oder partiell verdeckte Gesichter können mit den bisherigen Verfahren zumeist nicht richtig erkannt und damit auch nicht analysiert werden. Dabei sind mimische Gesichtsveränderungen oft gerade im Zusammenhang mit schnellen Kopfbewegungen zu beobachten, zum Beispiel, wenn sich eine Person überrascht oder freudig einer anderen Person zuwendet. Für dieses Problem sind verschiedene Lösungen denkbar. In [Chang et al., 2005b] wird ein Echtzeit-3D-Scanner eingesetzt, um mit einer Stereo-Kameraanordnung reale dreidimensionale Daten zu erhalten. Eine andere Lösung ohne 3D Scanner ist zum Beispiel mit einem zylindrischen dreidimensionalen Kopfmodell [Xiao et al., 2003] möglich. Mit Hilfe des optischen Flusses wird hierbei die Pose der Testperson mit einem mittleren Fehler von 3 Grad genau geschätzt. Mit diesem Modell kann die Frontalansicht der Personen auch bei starker Verdrehung bis zu 60 Grad gut rekonstruiert werden. Weitere Arbeiten, bei denen statt eines zylindrischen Modells ein reales Kopfmodell verwendet wird, finden sich z.B. in [Kán, 2010]. Obwohl ein reales Kopfmodell weit komplexer zu erstellen und zu handhaben ist, ist die Rekonstruktion von Frontalbildern wesentlich genauer möglich und damit auch für die Extraktion der Gesichtsmerkmale von Vorteil.

# C Details zu Active Appearance Modells

## C.1 Warp - stückweise, affine Transformation

Ziel des Warps ist die Überführung einer beliebigen Form $\mathbf{s_s}$ in eine Zielform $\mathbf{s_d}$. Wird die Triangulation der Form vernachlässigt, so ist diese Überführung trivial, da die neuen Positionen der Knotenpunkte direkt durch $\mathbf{s_s}$ bestimmt sind. Anders sieht es hingegen aus, wenn die Form als Oberfläche betrachtet wird. In diesem Fall reicht eine Überführung der Knotenpunkte nicht aus, vielmehr muss die Überführung für jeden beliebigen Punkt $\mathbf{x}$ innerhalb der Form definiert werden.

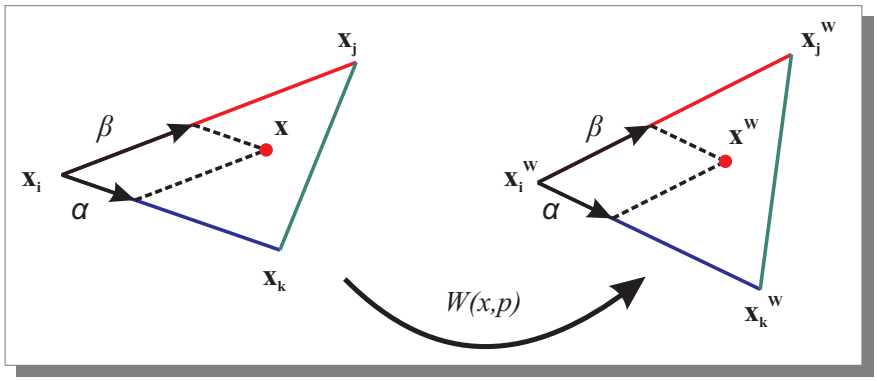



***Abbildung C.1:*** *Überführung des Punktes $x$ von der Quellform in die Zielform über eine affine Transformation mit $W(\mathbf{x}, \mathbf{p})$.*

Zunächst wird ein beliebiges Dreieck der Triangulation der Form $\mathbf{s_s}$ betrachtet. Ohne Einschränkung der Allgemeinheit sei das Dreieck durch die Knotenpunkte $\mathbf{x_i}$, $\mathbf{x_j}$ und $\mathbf{x_k}$ definiert. In der Zielform widerum seien die Knotenpunkte durch $\mathbf{x_i^W}$, $\mathbf{x_j^W}$ und $\mathbf{x_k^W}$ definiert (Abb. C.1). Es ist offensichtlich, dass eine Abbildung, welche die gewünschte Transformation durchführt, mindestens die Eckpunkte der Dreiecke eineindeutig aufeinander abbilden muss. Weiterhin ergeben sich noch zwei weitere Anforderungen an die gesuchte Transformation:

- Bestehende Abstandsverhältnisse sollen beibehalten werden.
- Punkte, die vor der Transformation auf einer Linie lagen, sollen auch nach der Transformation wieder auf einer Linie liegen. Die Transformation soll demzufolge auch kollinear sein.

Eine Abbildung, welche die geforderten Eigenschaften besitzt, wird auch als affine Abbildung bezeichnet. Die gesuchte Abbildung eines beliebigen Punktes $\mathbf{x}$ auf den Zielpunkt $\mathbf{x^W}$ kann über eine solche affine Transformation definiert werden. Dazu wird der Punkt $\mathbf{x}$ in ein neues Basissystem projiziert, welches durch die Vektoren $\overline{\mathbf{x_i x_j}}$ und $\overline{\mathbf{x_i x_k}}$ gebildet wird. Ohne Einschränkung der Allgemeinheit wird der Punkt $\mathbf{x}$ dabei auf die Position $(\alpha\beta)^T$ im neuen Basissystem projiziert:

$$\alpha = \frac{(\mathbf{x}-\mathbf{x_i})\cdot(\mathbf{x_k}-\mathbf{x_i})^{\perp}}{(\mathbf{x_j}-\mathbf{x_i})\cdot(\mathbf{x_k}-\mathbf{x_i})^{\perp}}, \quad \beta = \frac{(\mathbf{x}-\mathbf{x_i})\cdot(\mathbf{x_j}-\mathbf{x_i})^{\perp}}{(\mathbf{x_k}-\mathbf{x_i})\cdot(\mathbf{x_j}-\mathbf{x_i})^{\perp}} \tag{C.1}$$

Daraus ergeben sich die Berechnungsvorschriften von $\alpha$ und $\beta$ in Bezug auf die $x$ und $y$ Koordinaten der Dreieckspunkte und des zu transformierenden Punktes $\mathbf{x}$:

$$\alpha = \frac{(x-x_i)(y_k-y_i)-(y-y_i)(x_k-x_i)}{(x_j-x_i)(y_k-y_i)-(y_j-y_i)(x_k-x_i)} \tag{C.2}$$

$$\beta = \frac{(x-x_i)(y_j-y_i)-(y-y_i)(x_j-x_i)}{(x_k-x_i)(y_j-y_i)-(y_k-y_i)(x_j-x_i)} \tag{C.3}$$

Als Resultat lässt sich die Position des Punktes $\mathbf{x}$ durch einen beliebigen Dreieckspunkt $\mathbf{x_i}$ und die Skalierungsfaktoren $\alpha$ und $\beta$ beschreiben:

$$\mathbf{x} = \mathbf{x_i} + \alpha\cdot[\mathbf{x_j}-\mathbf{x_i}] + \beta\cdot[\mathbf{x_k}-\mathbf{x_i}] \tag{C.4}$$

Da die gewünschte Abbildung die Bedingung der Kollinearität erfüllen soll, muss sich der gewarpte Punkt $\mathbf{x^W}$ ebenfalls an der Position $(\alpha\beta)^T$ befinden. Einzig das zugrunde liegende Basissystem hat sich während des Warps geändert. Es wird nun nicht mehr über die Vektoren der Quellform $\mathbf{s_s}$ definiert, sondern ergibt sich aus den Vektoren, welche das Dreieck der Zielform $\mathbf{s_d}$ aufspannen:

$$W(\mathbf{x},\mathbf{p}) = \mathbf{x_i^W} + \alpha\cdot\left[\mathbf{x_j^W}-\mathbf{x_i^W}\right] + \beta\cdot\left[\mathbf{x_k^W}-\mathbf{x_i^W}\right] \tag{C.5}$$

Die Berechnung der Skalierungsfaktoren $\alpha$ und $\beta$ erfolgt dabei wie in Gleichung (C.2) und (C.3) beschrieben.

## C.2 Warp - Umkehrtransformation und Warpkomposition

### Umkehrtransformation

Die Umkehroperation zu $W(\mathbf{x},\Delta\mathbf{p})$ wird als $W^{-1}(\mathbf{x},\Delta\mathbf{p})$ bezeichnet. Es gilt

$$W(\mathbf{x},\Delta\mathbf{p}) = W(\mathbf{x},\mathbf{0}) + \frac{\partial W}{\partial\mathbf{p}}\Delta\mathbf{p} + O\left(\Delta\mathbf{p}^2\right) = \mathbf{x} + \frac{\partial W}{\partial\mathbf{p}}\Delta\mathbf{p} + O\left(\Delta\mathbf{p}^2\right) \tag{C.6}$$

Da $W(\mathbf{x}, \mathbf{0}) = \mathbf{x}$ die Identitätstransformation darstellt, ist

$$W(\mathbf{x}, \Delta\mathbf{p}) \circ W(\mathbf{x}, -\Delta\mathbf{p}) = \mathbf{x} + \frac{\partial W}{\partial \mathbf{p}}\Delta\mathbf{p} - \frac{\partial W}{\partial \mathbf{p}}\Delta\mathbf{p} + O\left(\Delta\mathbf{p}^2\right) = \mathbf{x} + O\left(\Delta\mathbf{p}^2\right) \quad \text{(C.7)}$$

In erster Ordnung von $\Delta\mathbf{p}$ gilt daher:

$$W^{-1}(\mathbf{x}, \Delta\mathbf{p}) = W(\mathbf{x}, -\Delta\mathbf{p}) \quad \text{(C.8)}$$

#### Verknüpfungsansatz der Transformationsfunktion

Mit dieser Umkehrtransformation muss nun die aktuelle Transformationsfunktion $W(\mathbf{x}, \mathbf{p})$ durch eine Verknüpfung mit $W^{-1}(\mathbf{x}, \Delta\mathbf{p})$ angepasst werden, um bei einem Iterationsschritt den neuen Parametersatz zu erhalten. Aus den aktuellen Transformationsparametern $\mathbf{p}$ können mittels

$$\mathbf{s} = \mathbf{s_0} + \sum_{i=1}^{n} p_i \mathbf{s_i} \quad \text{(C.9)}$$

die aktuellen Knotenpositionen $\mathbf{s} = (x_1, y_1, \ldots, x_v, y_v)$ bestimmt werden. Wie in Gleichung (C.8) gezeigt, hat $W^{-1}(\mathbf{x}, \Delta\mathbf{p})$ näherungsweise die Parameter $-\Delta\mathbf{p}$. Daher kann mittels (C.9) die geschätzte Änderung der Knotenpositionen in Bezug auf die Grundform bestimmt werden:

$$\Delta\mathbf{s}_0 = -\sum_{i=1}^{n} \Delta p_i \mathbf{s}_i \quad \text{(C.10)}$$

Dabei sind $\Delta\mathbf{s}_0 = (\Delta x_1^0, \Delta y_1^0, \ldots, \Delta x_v^0, \Delta y_v^0)^T$ die Änderungen zur Grundform in Bezug auf $W^{-1}(\mathbf{x}, \Delta\mathbf{p})$. Um nun $W^{-1}(\mathbf{x}, \Delta\mathbf{p})$ mit $W(\mathbf{x}, \mathbf{p})$ zu verknüpfen, müssen die Änderungen in Bezug auf die aktuelle Form $\Delta\mathbf{s} = (\Delta x_1, \Delta y_1, ..., \Delta x_v, \Delta y_v)^T$ berechnet werden. Wenn diese Änderungen $\Delta\mathbf{s}$ bekannt sind, können die Parameter von $W(\mathbf{x}, \mathbf{p}) \circ W^{-1}(\mathbf{x}, \Delta\mathbf{p})$ durch Lösen von Gleichung (C.9) bestimmt werden:

$$p_i' = \mathbf{s}_i \cdot (\mathbf{s} + \Delta\mathbf{s} - \mathbf{s}_0)\,. \quad \text{(C.11)}$$

Es wird jeweils das Skalarprodukt mit den orthonormalen Formkomponenten $\mathbf{s}_i$ gebildet, um den entsprechenden Formparameter $p_i'$ zu erhalten.

Anschließend wird $\Delta s$ aus $\Delta s_0$ bestimmt. Dazu wird der $i$-te Knotenpunkt der Form betrachtet. Dabei muss $(\Delta x_i, \Delta y_i)^T$ in der aktuellen Form $\mathbf{s}$ aus $(\Delta x_i^0, \Delta y_i^0)^T$ in der Grundform $\mathbf{s}_0$ bestimmt werden. Für alle Dreiecke, die den $i$-ten Knotenpunkt enthalten, ist die Transformationsfunktion von der Grundform $\mathbf{s}_0$ zur aktuellen Form $\mathbf{s}$ bekannt (Abschnitt 4.4.1). Eine Möglichkeit $(\Delta x_i, \Delta y_i)^T$ aus $(\Delta x_i^0, \Delta y_i^0)^T$ zu bestimmen ist daher, die Trans-

formationsfunktion eines bestimmen Dreiecks auf $(x_i^0, y_i^0)^T + (\Delta x_i^0, \Delta y_i^0)^T$ anzuwenden, um $(x_i, y_i)^T + (\Delta x_i, \Delta y_i)^T$ zu erhalten. Es ist zu klären, welches Dreieck für diese Transformation zu wählen ist. Verschiedene Dreiecke bewirken verschiedene Transformationen. Eine Möglichkeit ist, das Dreieck auszuwählen, in dem der Punkt $(x_i^0, y_i^0)^T + (\Delta x_i^0, \Delta y_i^0)^T$ tatsächlich liegt. Hier ist das Problem, dass dieser Punkt auch außerhalb der Grundform liegen könnte. Es ist daher besser, die Transformation des Punktes $(x_i, y_i)^T + (\Delta x_i, \Delta y_i)^T$ für alle Dreiecke zu bestimmen, die den $i$-ten Knotenpunkt enthalten und daraus den Mittelwert zu bilden.

Diese Anpassung der Transformationsfunktion wird auch *Verknüpfungsansatz* oder *Warpkomposition* genannt.

## C.3 AAM-Anpassungsalgorithmen im Detail

Anknüpfend an Abschnitt 4.4.2 soll hier ein detaillierter Überblick über die verschiedenen Anpassungsalgorithmen präsentiert werden. Der Inhalt ist sinngemäß entnommen aus [Cootes et al., 1998] [Cootes et al., 2001] [Baker and Matthews, 2001] [Baker and Matthews, 2004] und [Matthews and Baker, 2004].

### Lucas-Kanade-Algorithmus

Ausgangspunkt für die Anpassung eines Active Appearance Models bildet der von Lucas und Kanade in [Lucas and Kanade, 1981] vorgestellte und nach den Autoren benannte Algorithmus. Der Algorithmus wurde ursprünglich entwickelt, um einen bestimmten Bildbereich, ein sogenanntes Template, in einer Bildfolge aufzufinden. Dazu wird über dem Template ein Warp mit globalen Tranformationen definiert, mit dessen Hilfe eine möglichst gute Anpassung an das Bild erreicht werden soll. Diese Voraussetzungen sind mit denen der Active Appearance Modelle identisch. Die Prinzipien der Anpassung der Active Appearance Modelle lassen sich daher sehr gut anhand dieses Algorithmus verdeutlichen.

Das Ziel des Lucas-Kanade-Algorithmus besteht in der Ermittlung eines optimalen Parametervektors $\mathbf{p}$, der den Unterschied zwischen einem Template $T(\mathbf{x})$ und dem, in das Templatekoordinatensystem gewarpten, Eingangsbild $I(x)$ minimiert (analog zu (4.10)).

Die Minimierung erfolgt in Bezug auf den Parametervektor $\mathbf{p}$ und ist im Allgemeinen nicht linear lösbar. Es wird daher davon ausgegangen, dass eine hinreichend genaue Initialschätzung für den Parametervektor $\mathbf{p}$ bereits bekannt ist. Dennoch bleibt das Minimierungsproblem nichtlinear und lässt sich ohne Weiteres nicht effizient berechnen. Um eine Lösung erhalten zu können, wird daher von Lucas und Kanade ein Gauß-Newton Minimierungsverfahren vorgeschlagen. Dabei wird die zu lösende Minimierung in einen iterativen Minimierungsprozess überführt. Anstatt direkt einen Parametervektor $\mathbf{p}$ zu bestimmen, der das Fehlerbild

minimiert, wird in jeder Iteration nach einem Parameterinkrement $\Delta\mathbf{p}$ gesucht, welches den Fehler schrittweise mit Hilfe der Methode der kleinsten Fehlerquadrate verringert. Eine Iteration besteht entsprechend aus zwei Schritten: Im ersten Schritt erfolgt die Bestimmung eines Parameterinkrements $\Delta\mathbf{p}$ und im zweiten Schritt erfolgt eine Anpassung von $\mathbf{p}$ mit dem Parameterinkrement. Die zu lösende Problemstellung ändert sich entsprechend zu:

$$\sum_{\mathbf{x}\in T} \left[I\left(W(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})\right) - T(\mathbf{x})\right]^2 \rightarrow Min \tag{C.12}$$

Wobei die Minimierung nicht mehr im Hinblick auf $\mathbf{p}$ sondern auf $\Delta\mathbf{p}$ durchgeführt wird. Der zweite Schritt, in welchem die Anpassung von $\mathbf{p}$ vorgenommen wird, erfolgt über:

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \tag{C.13}$$

Wird während dieses Prozesses ein Minimum durchlaufen, so geht die durch $\Delta\mathbf{p}$ vorgeschlagene Änderung gegen 0 und der Anpassungsprozess kann abgebrochen werden.

Bei kleinen Parameteränderungen $\Delta\mathbf{p}$ kann eine lineare Approximation vorgenommen werden. Diese lässt sich durch eine Taylorapproximation ersten Grades an der Stelle $\mathbf{p}$ realisieren:

$$\sum_{\mathbf{x}\in T} \left[I\left(W(\mathbf{x};\mathbf{p})\right) + \nabla I \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x})\right]^2, \quad \nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) \tag{C.14}$$

Dabei repräsentiert $\nabla I$ die Ableitung des Eingangsbildes, die an der Stelle $\mathbf{p}$ erfolgt. Der zweite durch die Taylorapproximation entstehende Term ist $\frac{\partial W}{\partial \mathbf{p}}$, welcher auch als *Jacobi-Matrix*[1] bezeichnet wird. Für einen Warp, der im Zweidimensionalen definiert ist durch $W(\mathbf{x};\mathbf{p}) = (W_x(\mathbf{x},\mathbf{y})W_y(\mathbf{x},\mathbf{y}))^T$, ergibt sich somit die folgende Jacobi-Matrix:

$$\frac{\partial W}{\partial p} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_n} \end{pmatrix} \tag{C.15}$$

Abbildung C.2 visualisiert die einzelnen Elemente der Matrix für ein beispielhaftes Appearance Modell. Die Jacobi-Matrix beschreibt letztendlich wie stark einzelne Bildregionen in Abhängigkeit eines Modellparameters durch einen Warp beeinflusst werden.

Die durch die Taylorapproximation entstandene Gleichung C.14 ist quadratisch. Ihr Minimum in Abhängigkeit von $\Delta\mathbf{p}$ kann damit durch partielles Ableiten nach $\Delta\mathbf{p}$ bestimmt

[1] Die Jacobi-Matrix, benannt nach Carl Gustav Jacob Jacobi wird auch als Ableitungsmatrix bezeichnet und ist die Matrixdarstellung der Ableitung einer Funktion $f$. Dazu enthält sie alle partiellen Ableitungen der Funktion.

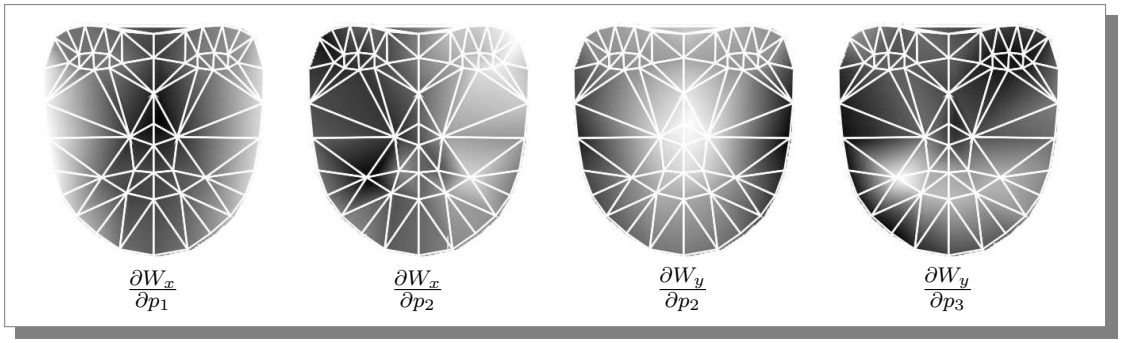



***Abbildung C.2:*** *Durch die weiße Farbe wird eine besonders starke Beeinflussung durch den Warp codiert, während schwarze Bereiche nicht beeinflusst werden. $p_1$ beschreibt eine Drehung des Kopfes in der Horizontalen während durch $p_2$ eine Drehung in der Vertikalen erreicht werden kann. $p_3$ wiederum bewirkt starke Änderungen an der Form des Mundes.*

werden:

$$2\sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T \left[I\left(W(x;p)\right) + \nabla I \frac{\partial W}{\partial p}\Delta p - T(x)\right] \tag{C.16}$$

Das Minimum lässt sich nun durch Gleichsetzen dieses Terms mit 0 und anschließendes Umstellen nach $\Delta\mathbf{p}$ ermitteln und hat die Form:

$$\Delta p = H^{-1}\sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T \left[T(x) - I\left(W(x;p)\right)\right] \tag{C.17}$$

Die Matrix $H$ ist die Gauß-Newton Approximation der zweiten Ableitungen der Funktion und bildet somit eine Annäherung an die Hessematrix[2] nach der folgenden Form:

$$H = \sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T \left[\nabla I \frac{\partial W}{\partial p}\right] = \sum_x SD(x)^T SD(x) \tag{C.18}$$

Der Term $\nabla I \frac{\partial W}{\partial \mathbf{p}} = SD(x)$ wird dabei auch als Gradientenabstiegsbild (*Steepest Descent*) bezeichnet. Zusammenfassend beschreiben diese Gradientenabstiegsbilder die „Richtung“ der Parameteränderung, während die Hessematrix eine Wichtung zwischen den einzelnen Parametern vornimmt und die endgültige Parameteränderung bestimmt.

Das Hauptproblem des Lucas-Kanade-Algorithmus ist, dass keinerlei Vorabberechnungen vorgenommen werden können. Sowohl die Jacobi- als auch die Hesse-Matrix müssen in jedem Iterationsschritt neu berechnet werden. Daher erreicht dieser Algorithmus nur eine sehr

[2]Die Hessematrix, benannt nach Otto Hesse, fasst die partiellen Ableitungen zweiter Ordnung einer mehrdimensionalen Funktion $f$ zusammen.

schlechte Performance und hat praktisch kaum Relevanz.

#### Additiver Ansatz nach Cootes

In [Cootes et al., 1998, Cootes et al., 2001] wird ein weit effizienterer Algorithmus vorgestellt. In diesem Ansatz geht man von der Annahme aus, dass es einen linearen Zusammenhang zwischen dem Fehlerbild $E(\mathbf{x})$ und einem additiven Parameterinkrement für die Form und das Grauwertbild gibt:

$$\Delta p_i = \sum_{\mathbf{x} \in \mathbf{s_0}} R_i(\mathbf{x}) E(\mathbf{x}) \text{ und } \Delta \lambda_i = \sum_{\mathbf{x} \in \mathbf{s_0}} S_i(\mathbf{x}) E(\mathbf{x}) \tag{C.19}$$

Dabei sind $R_i(\mathbf{x})$ und $S_i(\mathbf{x})$ konstante Bilder über der Grundform, d.h. sie sind nicht von $p_i$ und $\lambda_i$ abhängig. Sowohl $R_i(\mathbf{x})$ als auch $S_i(\mathbf{x})$ können dabei mit Hilfe gezielter kleiner Parameteränderungen auf bekannten Trainingsbildern ermittelt werden, indem für die jeweiligen Änderungen $\Delta p_i$ oder $\Delta \lambda_i$ das resultierende Differenzbild betrachtet wird. $\Delta p_i$ und $\Delta \lambda_i$ werden auch als Prädiktormatrizen bezeichnet. Mit dieser Annahme lassen sich die Parameteränderungen $\Delta p_i$ und $\Delta \lambda_i$ berechnen und die Parametervektoren in der Form $p_i \leftarrow p_i + \Delta p_i$ und $\lambda_i \leftarrow \lambda_i + \Delta \lambda_i$ linear anpassen.

In [Matthews and Baker, 2004] wird gezeigt, dass der lineare Zusammenhang nur in einem sehr kleinen Bereich gilt. Für eine effizientere Anpassung muss davon ausgegangen werden, dass $R_i(\mathbf{x})$ und $S_i(\mathbf{x})$ nicht konstant sind, sondern abhängig von $p_i$ und $\lambda_i$. Daher müssten $R_i(\mathbf{x})$ und $S_i(\mathbf{x})$ in jeder Iteration neu berechnet werden. Dies führt dazu, dass der additive Anpassungsalgorithmus mit linearem Inkrement ineffizient ist.

In [Matthews and Baker, 2004] wird auch gezeigt, dass es besser ist, die gesamte Transformationsfunktion durch eine Verknüpfung der aktuellen Transformation $W(\mathbf{x}, \mathbf{p})$ mit einem berechneten Transformationsinkrement mit dem Parameter $\mathbf{\Delta p}$ anzupassen. Die Anpassung verläuft dann folgendermaßen:

$$W(x, p) \leftarrow W(x, p) \circ W^{-1}(x, \Delta p) \tag{C.20}$$

Für diese Operation ist die Umkehrtransformation $W^{-1}(\mathbf{x}, \mathbf{\Delta p})$ notwendig. In erster Ordnung von $\mathbf{\Delta p}$ gilt (siehe Anhang C.2):

$$W^{-1}(\mathbf{x}, \mathbf{\Delta p}) = W(\mathbf{x}, -\mathbf{\Delta p}) \tag{C.21}$$

Basierend auf der Umkehrtransformation kann die Verknüpfung der Transformationsfunktion beschrieben werden (siehe Anhang C.2). Diese Anpassung der Transformationsfunktion wird auch *Verknüpfungsansatz* oder *Warpkomposition* genannt und ist die Grundlage für die nun folgenden Algorithmen.

### Vorwärts verknüpfter Algorithmus ohne Grauwertvariation

Im sogenannten "Vorwärts verknüpften Algorithmus" wird statt einer linearen Paramerteranpassung mit $\Delta\mathbf{p}$ der Verknüpfungsansatz verwendet. Dazu wird eine inkrementelle Transformation $W(\mathbf{x}, \Delta\mathbf{p})$ berechnet, mit der die aktuelle Transformation $W(\mathbf{x}, \mathbf{p})$ angepasst wird (Gleichung (C.20)). Mit der Verknüpfung der aktuellen Transformation mit der Inkrementtransformation lässt sich das folgende Minimierungsproblem formulieren:

$$\mathbf{p} = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}} [A_0(\mathbf{x}) - I(W(W(\mathbf{x}, \Delta\mathbf{p}), \mathbf{p}))]^2 \tag{C.22}$$

Zur Lösung wird eine Taylorreihenentwicklung erster Ordnung von (C.22) durchgeführt:

$$\sum_{\mathbf{x}} \left[ A_0(\mathbf{x}) - I(W(W(\mathbf{x}, \mathbf{0}), \mathbf{p}) - \nabla I(W(\mathbf{x}, \mathbf{p})) \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p} \right]^2 \tag{C.23}$$

Unter der Annahme, dass $W(\mathbf{x}, \mathbf{0}) = \mathbf{x}$ die Identitätstransformation darstellt, ergibt sich:

$$\sum_{\mathbf{x}} \left[ A_0(\mathbf{x}) - I(W(\mathbf{x}), \mathbf{p}) - \nabla I(W(\mathbf{x}, \mathbf{p})) \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p} \right]^2 \tag{C.24}$$

Dabei wird der Gradient von $I(W(\mathbf{x}, \mathbf{p}))$ berechnet. Dieser ändert sich bei jeder Iteration und kann daher nicht vorberechnet werden. Die für einen Gradientenabstieg notwendige Jacobi-Matrix wird jedoch an der Stelle $(\mathbf{x}, \mathbf{0})$ berechnet. Damit bleibt sie während der Iterationen konstant und kann im Voraus berechnet werden.

Bei der Anpassung der Transformation nach Gleichung (C.20) ist der Verknüpfungsansatz berechnungsintensiver als die einfache additive Parameteranpassung. Dafür braucht man jedoch die Jacobi-Matrix nicht in jeder Iteration neu zu berechnen, was ein weit größerer Aufwand wäre.

### Rückwärts verknüpfter Algorithmus ohne Grauwertvariation

Der "Rückwärts verknüpfte Algorithmus" ist eine Modifikation des "Vorwärts verknüpften Algorithmus", bei dem die Rollen des Modells und des Eingabebildes vertauscht werden. Anstatt die Inkrementtransformation in Bezug auf $I(W(\mathbf{x}, \mathbf{p}))$ zu berechnen, wird sie nun in Bezug auf das Mittelwertgesicht $A_0(\mathbf{x})$ berechnet. Einen Beweis, dass das Ergebnis dieses Rollentauschs äquivalent zum "Vorwärts verknüpften Algorithmus" ist, findet man in [Baker and Matthews, 2001] und [Baker and Matthews, 2004]. Das Minimierungsproblem von Gleichung (C.22) wird dabei wie folgt modifiziert:

$$\mathbf{p} = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}} [I(W(\mathbf{x}, \mathbf{p})) - A_0(W(\mathbf{x}, \Delta\mathbf{p}))]^2 \tag{C.25}$$

Die Anpassung der Transformationsfunktion geschieht dabei nun folgendermaßen:

$$W(\mathbf{x},\mathbf{p}) \leftarrow W(\mathbf{x},\mathbf{p}) \circ W^{-1}(\mathbf{x},\Delta\mathbf{p}) \tag{C.26}$$

Entwickelt man den rechten Teil von Gleichung (C.25) ebenfalls in einer Taylorreihe bis zum ersten Grad, so erhält man

$$\sum_{\mathbf{x}} \left[ I(W(\mathbf{x},\mathbf{p}) - A_0(W(\mathbf{x},\mathbf{0})) - \nabla\mathbf{A}_0 \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p} \right]^2 \tag{C.27}$$

Dabei ist $W(\mathbf{x},\mathbf{0}) = \mathbf{x}$ erneut die Identitätstransformation und es ergibt sich eine geschlossene Lösung der kleinsten Quadrate für diesen Term mit:

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[ \nabla\mathbf{A}_0 \frac{\partial W}{\partial \mathbf{p}} \right]^T [I(W(\mathbf{x},\mathbf{p})) - A_0(\mathbf{x})] \tag{C.28}$$

Die Hessematrix $\mathbf{H}$ ist nun nicht mehr von $I(\mathbf{x})$ abhängig, sondern wird aus dem Mittelwertgesicht $A_0(\mathbf{x})$ berechnet:

$$\mathbf{H} = \sum_{\mathbf{x}} \left[ \nabla\mathbf{A}_0 \frac{\partial W}{\partial \mathbf{p}} \right]^T \left[ \nabla\mathbf{A}_0 \frac{\partial W}{\partial \mathbf{p}} \right] \tag{C.29}$$

Da das Mittelwertgesicht $A_0(\mathbf{x})$ konstant ist und die Jacobi-Matrix immer an der Stelle $(\mathbf{x},\mathbf{0})$ berechnet wird, kann der Hauptteil der Berechnungen von Gleichung (C.28) und (C.29) im Voraus geschehen. Das ist der entscheidende Vorteil gegenüber dem "Vorwärts verknüpften Algorithmus". Hierdurch erhält man einen effizienten, echtzeitfähigen Algorithmus, der während der Iterationsphase nur noch wenige, schnell berechenbare Schritte enthält.

#### Rückwärts verknüpfter Algorithmus mit Grauwertvariation

Im vorherigen Abschnitt wurde ein schneller Gradientenabstiegsalgorithmus beschrieben, der jedoch Variationen durch die Grauwertkomponenten nicht betrachtet. Um die Variationen durch die Grauwertkomponenten zu berücksichtigen, muss die Gleichung (4.10) gelöst werden. In [Hager and Belhumeur, 1998] wird eine Projektionstechnik vorgestellt, die hier verwendet werden kann, um die Grauwertkomponenten zu berücksichtigen und trotzdem einen schnellen Algorithmus wie im vorherigen Abschnitt zu erhalten. Der Term (4.10) wird dazu wie folgt umgeformt:

$$\sum_{x \in s_0} \left[ A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(W(x,p)) \right]^2 = \left\| A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(W(x,p)) \right\|^2 \tag{C.30}$$

Dabei ist $\|...\|^2$ die $L_2$-Norm. Dieser Term wird simultan in Bezug auf $\mathbf{p} = (p_1, p_2, ..., p_m)$ und $\lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ minimiert.

Der lineare Unterraum, den die Grauwertbildkomponenten $\mathbf{A}_i$ aufspannen, wird als $span(\mathbf{A}_i)$ bezeichnet. Der zu diesem Raum orthogonale Unterraum ist $span(\mathbf{A}_i)^\perp$. Damit kann der rechte Teil der Gleichung (C.30) wie folgt aufgespalten werden:

$$\left\| A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(W(\mathbf{x}, \mathbf{p})) \right\|^2_{span(\mathbf{A}_i)^\perp} + \left\| A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(W(\mathbf{x}, \mathbf{p})) \right\|^2_{span(\mathbf{A}_i)} \tag{C.31}$$

$\|\mathbf{u}\|^2_L$ ist dabei die $L_2$-Norm des Vektors $\mathbf{u}$ projiziert in den linearen Unterraum $L$.

Der erste Summand kann sofort vereinfacht werden, da die Norm nur die Anteile berücksichtigt, die im orthogonalen Vektorraum zu $span(\mathbf{A}_i)$ liegen. Damit entfallen alle Anteile, die in $span(\mathbf{A}_i)$ liegen. Der Term (C.31) vereinfacht sich damit zu

$$\|A_0(\mathbf{x}) - I(W(\mathbf{x}, \mathbf{p}))\|^2_{span(\mathbf{A}_i)^\perp} + \left\| A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(W(\mathbf{x}, \mathbf{p})) \right\|^2_{span(\mathbf{A}_i)} \tag{C.32}$$

Der erste dieser Terme hängt nicht mehr von $\lambda_i$ ab. Für jedes $\mathbf{p}$ ist der Wert des zweiten Terms gleich Null. Daher kann das Minimum sequenziell gefunden werden, wenn man zuerst den ersten Term allein in Bezug auf $\mathbf{p}$ minimiert und dann dieses $\mathbf{p}$ konstant hält und den zweiten Term in Bezug auf $\lambda$ minimiert. Wenn man annimmt, dass die $\mathbf{A}_i$ eine orthonormale Basis bilden, kann der zweite Term geschlossen minimiert werden zu

$$\lambda_i = \sum_{\mathbf{x} \in \mathbf{s}_0} A_i(\mathbf{x}) \left[ I(W(\mathbf{x}, \mathbf{p})) - A_0(\mathbf{x}) \right] \tag{C.33}$$

Dies ist das Skalarprodukt von $\mathbf{A}_i$ mit dem resultierenden Fehlerbild nach der ersten Minimierung. Somit können die $\lambda_i$ mit wenig Aufwand berechnet werden, insofern die erste Minimierung gelöst wurde.

Die Minimierung des ersten Summanden aus Gleichung (C.32) erfolgt ähnlich wie beim "Rückwärts verknüpften Algorithmus" ohne Grauwertvariation im vorherigen Abschnitt. Der einzige Unterschied besteht darin, dass hier in dem linearen Unterraum $span(\mathbf{A}_i)^\perp$ operiert wird. Dabei muss das Fehlerbild $E(\mathbf{x})$, da es ausschließlich in Skalarprodukten verwendet wird, nicht in diesen Unterraum projiziert werden. Nur $\nabla \mathbf{A}_0 \partial W / \partial \mathbf{p}$ muss in den Unterraum $span(\mathbf{A}_i)^\perp$ projiziert werden. Das Projektionsergebnis wird als Abstiegsbild $SD_j(\mathbf{x})$, $j =$

$1, ..., m$ für jede Formkomponente $\mathbf{s}_j$ bezeichnet. Es wird wie folgt berechnet:

$$SD_j(\mathbf{x}) = \nabla \mathbf{A}_0 \frac{\partial W}{\partial p_j} - \sum_{i=1}^{m} \left[ \sum_{\mathbf{x} \in \mathbf{s}_0} A_i(\mathbf{x}) \cdot \nabla \mathbf{A}_0 \frac{\partial W}{\partial p_j} \right] \cdot A_i(\mathbf{x}) \tag{C.34}$$

Dabei werden durch den Term $\nabla \mathbf{A}_0 \partial W / \partial p_j$ alle Anteile entfernt, die in $span(\mathbf{A}_i)$ liegen, indem die Projektionen auf die orthonormalen Vektoren $\mathbf{A}_i$ abgezogen werden. Die Gauß-Newton Hessematrix wird analog zur Gleichung (C.29) wie folgt berechnet:

$$\mathbf{H} = \sum_{\mathbf{x}} SD(\mathbf{x})^T SD(\mathbf{x}) \tag{C.35}$$

Daraus ergibt sich für die inkrementelle Transformationsanpassung eine Parameteränderung von

$$\Delta \mathbf{p} = -\mathbf{H}^{-1} \sum_{\mathbf{x}} SD(\mathbf{x}) \left[ I(W(\mathbf{x}, \mathbf{p})) - A_0(\mathbf{x}) \right] \tag{C.36}$$

Somit ist dieser Algorithmus genau so schnell wie der "Rückwärts verknüpfte Algorithmus" ohne Grauwertvariation, berücksichtigt jedoch auch die Grauwertkomponenten. Dieser Algorithmus wird auch als *Project-Out-Algorithmus* bezeichnet.

# D Weitere eingesetzte Verfahren

## D.1 Merkmalsauswahl mittels MIFS

Ein wichtiger Bestandteil eines allgemeinen Musterkennungssystems ist die Merkmalsextraktion, die sich zwischen einer problemspezifischen Vorverarbeitung und der Kodierung der Merkmale für einen Klassifikator, Funktionsapproximater oder Clusterer einordnet.

Die *Merkmalsselektion* ist von der *Merkmalstransformation* zu unterscheiden: Bei Ersteren sollen die signifikanten/relevanten Merkmale ermittelt werden. Bei der Merkmalstransformation sollen neue Merkmale durch Kombination(en) vorhandener Merkmale gebildet werden. Die Ziele der *Merkmalsselektion* bzw. *Signifikanzanalyse* für ein konkretes Problem sind:

- Finden und Weglassen von *irrelevanten* Kanälen,
- Beibehalten aller *relevanten* Kanäle und
- Identifikation (und gegebenenfalls Weglassen) von *redunanten* Kanälen.

Letztendlich kann damit die Anzahl der Eingangsgrößen reduziert werden. Durch diese geringere Dimensionalität des Eingaberaums wird die Suche nach einer Lösung in einem weniger schwierigen Fehlergebirge erleichtert. Zusätzlich wird die Anzahl der freien Parameter reduziert, wodurch ein schnelleres Lernen der Parameter möglich ist, eine bessere Generalisierungsfähigkeit erreicht werden kann und die Wahrscheinlichkeit eines Overfittings reduziert wird.

Eine einfache Signifikanzanalyse kann mittels linearer Korrealationsanalyse, der Fisher-Diskriminate oder linearer Diskriminanzanalyse durchgeführt werden. Das Problem dieser linearen Analyseverfahren ist jedoch, dass sie nur lineare Zusammenhänge detektieren können. In vielen Problemen existieren teilweise hochgradig nichtlineare Zusammenhänge, die über Kovarianzen oder Korrelationskoeffizienten nicht erkannt werden können. Einen Ausweg hierfür bilden nichtlineare Konzepte aus der Informationstheorie.

Die *Entropie* $H(X)$ beschreibt ein Maß für die Unsicherheit einer Zufallsvariable $X$, quantifiziert durch die durchschnittliche Information dieser Variablen:

$$H(X) = -\int p(x) \cdot \log\left(p(x)\right) dx \tag{D.1}$$

Die maximale Entropie entsteht dann, falls alle Ereignisse von $X$ gleich wahrscheinlich sind (Gleichverteilung). Die Entropie ist dann gleich der logarithmischen Anzahl der Ereignisse $|X|$. Somit gilt:

$$0 \leq H(X) \leq \log |X| \tag{D.2}$$

In der Informatik wird typischerweise der *logarithmus dualis* ($\log_2$) verwendet.

Die *bedingte Entropie* $H(X|Y)$ ist ein Maß für die Unsicherheit einer Zufallsvariablen $X$ unter Kenntnis einer anderen Zufallsvariablen $Y$:

$$H(X|Y) = -\int\int p(x,y) \log\left(p(x|y)\right) dxdy \tag{D.3}$$

Die Kenntnis einer Variablen $Y$ kann die Unsicherheit über die Variable $X$ im Mittel nicht erhöhen:

$$H(X|Y) \leq H(X) \tag{D.4}$$

Die *Transinformation* $I(X;Y)$ (auch *Mutual Information (MI)* genannt) ist ein Maß für die Information einer Zufallsvariable $X$ über eine andere Zufallsvariable $Y$:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ I(X;Y) &= H(Y) - H(Y|X) \\ I(X;Y) &= H(X) + H(Y) - H(X|Y) \end{aligned} \tag{D.5}$$

Die Transinformation ist symmetrisch: $I(X;Y) = I(Y;X)$.

Eine grafische Interpretation der Transinformation und Entropie ist in Abb. D.1 dargestellt:

Die Mutual Information kann als Distanz von Wahrscheinlichkeitsverteilungen interpretiert werden. Somit entspricht die Mutual Information dann der *Kullback-Leibler-Diverenz* der Verteilungen über $X$ und $Y$:

$$I(X;Y) = \int_x \int_y p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} dxdy \tag{D.6}$$

Das Problem hierbei ist die Ermittlung der Verteilungen $p(x)$, $p(y)$ und $p(x,y)$ aus vorhandenen Daten. Diese Verteilungen sind typischerweise nicht bekannt. Eine Möglichkeit besteht in einer *Kernel Density Estimation*, wobei jeder Datenpunkt mit einer Gauß-Kurve in die Schätzung eingeht. Problem hierbei ist jedoch, dass ein Doppelintegral gelöst werden muss. Praktikabler ist eine diskrete Approximation der Verteilungen über Histogramme. Sowohl die

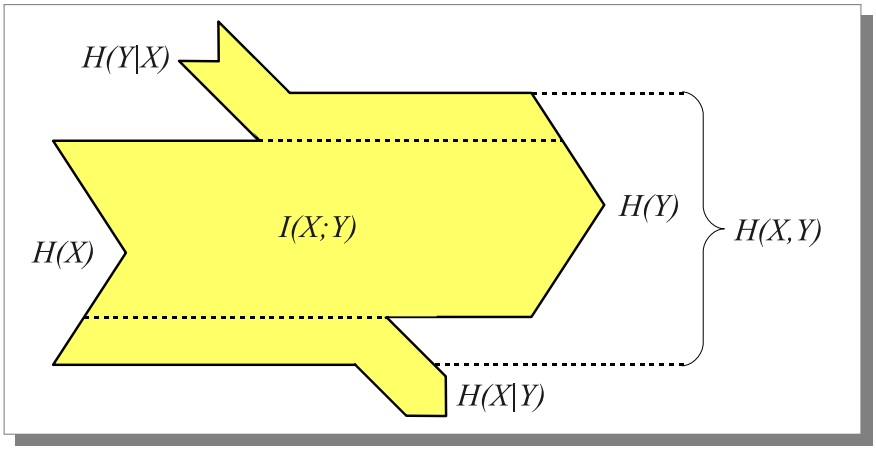



***Abbildung D.1:*** *Grafische Interpretation der Zusammenhänge zwischen der Entropie und der Transinformation.*

Verbund- als auch die Einzelverteilungen können als 2D bzw. 1D Histogramme beschrieben werden.
Ein Problem ist die Bestimmung der geeigneten Anzahl von Bins bzw. die Breite $W$ der Bins. Nach der (Daumen-)Regel von Scott gilt:

$$W = 3.5\sigma N^{-\frac{1}{3}} \tag{D.7}$$

Wobei $\sigma$ die Standardabweichung und $N$ die Anzahl des Samples ist.

Ein positiver Nebeneffekt der Approximation durch Histogramme ist, dass sich die Berechnung der Mutual Information deutlich vereinfacht:

$$I(X;Y) = \sum_x \sum_y p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} dxdy \tag{D.8}$$

Die eigentliche Merkmalsauswahl erfolgt entsprechend dem Ranking der Mutual Information über die einzelnen Merkmale. Ein Problem hierbei ist, dass mit dieser Vorgehensweise keine Redundanzen erkannt werden können. Die theoretisch optimale Lösung für dieses Problem ist die *Joint Mutual Information (JMI)* (Verbund-Transinformation):

$$I(X;Y) = \int_x \int_y p(x_1,...,x_n,y) \log_2 \frac{p(x_1,...,x_n,y)}{p(x_1,...,x_n)p(y)} dxdy \tag{D.9}$$

Die praktische Umsetzung scheitert an der Bestimmung der benötigten hochdimensionalen Verbundverteilung, da ein exponentieller Zusammenhang zur Dimensionalität des Problems besteht.

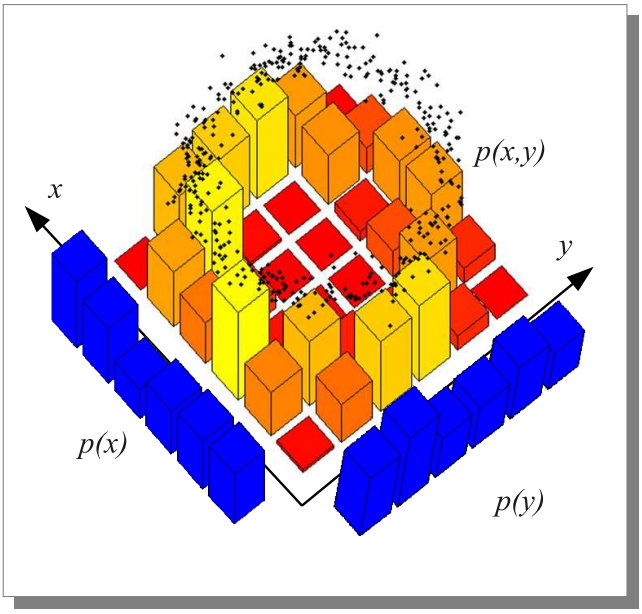



***Abbildung D.2:*** *Approximation der Verteilungen durch 1D bzw. 2D Histogramme.*

Eine Alternative wurde von [Battiti, 1994] als *Mutual Information for Feature Selection (MIFS)* vorgestellt. Neben der MI zwischen dem zu bewertenden Merkmal und dem Zielkanal wird die paarweise MI zwischen den gewählten Merkmalen des Subsets und des Kandidatenmerkmals einbezogen:

$$\underset{x}{\operatorname{argmax}} \left( I(x_i, y_i) - \beta \sum_{s \in S} I(s_k; x_i) \right) \tag{D.10}$$

Ein typischer Wert für $\beta$ liegt zwischen 0.1 und 0.3.

Letztendlich entsteht folgender Algorithmus:

1. $S \leftarrow \emptyset$
2. Bestimme Merkmal $x_{max}$ mit höchstem MIFS-Wert (siehe oben)
3. $S \leftarrow S \cup x_{max}$
4. Solange das Abbruchkriterium nicht erfüllt ist, gehe zu Schritt 2

Als mögliche Abbruchkriterien können z.B. eine maximale Anzahl von Merkmalen im Subset $S$ oder ein maximaler MIFS-Wert $x_{max} < 0$ verwendet werden.

In einer praktischen Implementierung ist es hilfreich, ein oder mehrere künstliche Merkmale $r_i$ hinzuzufügen, die nur Rauschen (also keine Informationen) enthalten. Sobald im Schritt (2) des vorgestellten Algorithmus eines dieser Rausch-Merkmale $r_i$ ausgewählt wird, kann der Algorithmus abgebrochen werden, da alle noch nicht zu $S$ hinzugefügten Merkmale weniger Informationen als ein Rauschkanal $r_i$ besitzen und daher nicht für die Lösung des Problems relevant sind.

## D.2 Generalized Orthogonal Procrustes Analysis

Die Generalized Orthogonal Procrustes Analysis (GPA) dient der Ausrichtung einer gegebenen Menge von $n$ Formen/Labels im zweidimensionalen Raum [Ross, 2004]. Jede Form $s^{(i)},\ i = 1, ..., n$ besteht dabei aus $v$ Punkten $(x, y)$:

$$s^{(i)} = \left(x_1^{(i)}, y_1^{(i)}, ..., x_v^{(i)}, y_v^{(i)}\right) \tag{D.11}$$

Jede Form wird im Rahmen der GPA einer Rotation, Translation und Skalierung unterzogen, so dass der quadratische Fehler aller Vektoren $s^{(i)}$ zur mittleren Form $\overline{s}$ minimal wird:

$$E = \sum_{i=1}^{n} |s^{(i)} - \overline{s}|^2. \tag{D.12}$$

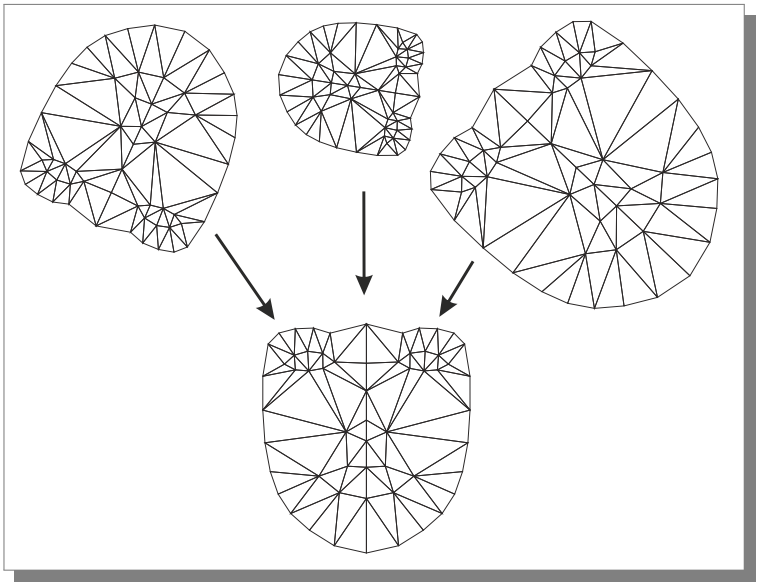


***Abbildung D.3:*** *Ausrichtung von Formen durch Procrustes Analyse: Ziel ist die Entfernung von Rotations-, Translations- und Skalierungsanteilen in der gegebenen Formmenge.*

Der Algorithmus geht dabei iterativ vor. Die Ausrichtung der Labels zueinander wird solange weitergeführt bis eine definierte Abbruchbedingung erfüllt ist. So kann die Berechnung z.B. solange fortgesetzt werden, wie sich der Fehler reduziert bzw. abgebrochen werden, wenn der Unterschied zwischen den Fehlerwerten der aktuellen Ausrichtung und dem vorhergehendem Zeitschritt eine Schwelle unterschreitet oder eine maximale Anzahl von Iterationen überschritten wurde.

Der Algorithmus arbeitet dabei wie folgt:

1. Bestimmung der mittleren Form $\overline{s}$
2. Berechnung des Fehlers $E$
3. Ausrichten der Formen zueinander
   a) Translation: Verschieben aller Schwerpunkte in den Ursprung
   b) Skalierung: Normalisieren aller Formen
   c) Rotation der Formen
4. Berechnen des neuen Fehlers. Wenn die Abbruchbedingung nicht erfüllt ist, setzt der Algorithmus bei Schritt 3 wieder ein.

Im Folgenden werden die einzelnen Schritte detailliert erläutert.

### D.2.1 Bestimmung der mittleren Form

Neben dem Mittelwert über allen Labels:

$$\overline{s} = (\overline{x}_1, \overline{y}_1, ..., \overline{x}_v, \overline{y}_v), \tag{D.13}$$

$$(\overline{x}_k,\ \overline{y}_k) = \left( \sum_{i=1}^{b} x_k^{(i)}, \sum_{i=1}^{b} y_k^{(i)} \right), \quad k = 1, ..., v \tag{D.14}$$

kann beispielsweise auch die erste Form der Menge ausgewählt werden. Typischerweise wird jedoch der Mittelwert aller Labels $s^{(i)}$ verwendet.

### D.2.2 Verschieben aller Schwerpunkte in den Ursprung

In diesem Schritt wird für jede Form $s^{(i)}$ der Schwerpunkt $\overline{s}^{(i)} = \left(\overline{x}^{(i)}, \overline{y}^{(i)}\right)$

$$\left(\overline{x}^{(i)}, \overline{y}^{(i)}\right) = \left( \frac{1}{v} \sum_{k=1}^{v} x_k^{(i)}, \frac{1}{v} \sum_{k=1}^{v} y_k^{(i)} \right) \tag{D.15}$$

ermittelt. Anschließend wird der Schwerpunkt der Labels in den Ursprung verschoben:

$$\left(x_k'^{(i)},\ y_k'^{(i)}\right) = \left(x_k^{(i)} - \overline{x}^{(i)},\ y_k^{(i)} - \overline{y}^{(i)}\right), \quad k = 1, ..., v, \tag{D.16}$$

$$s'^{(i)} = \left(x_1'^{(i)}, y_1'^{(i)}, ..., x_v'^{(i)}, y_v'^{(i)}\right) \tag{D.17}$$

Somit ergibt sich eine Menge von Formen $s'^{(i)}$, die alle zentriert zum Ursprung liegen.

### D.2.3 Normalisieren der Labelmenge

Im nächsten Schritt erfolgt die Normierung der zentrierten Labels $s'^{(i)}$ auf die Länge Eins:

$$s''^{(i)} = \frac{s'^{(i)}}{\left\| s'^{(i)} \right\|}. \tag{D.18}$$

Als Ergebnis dieses Zwischenschritts liegt nun eine Menge von Formen $s''^{(i)}$ vor, die zentriert zum Ursprung liegen und normiert auf die Länge Eins sind.

### D.2.4 Rotation der Konfiguration

Im Anschluss an die Translation und die Skalierung erfolgt die Rotation. Dabei ist das Ziel, die Minimierung des quadratischen Fehlers zwischen den Formen $s''^{(i)}$ und der mittleren Form $\overline{s''}$. Dieser lässt sich durch:

$$E^{(i)} = \left| s''^{(i)} - \overline{s''} \right|^2 = \sum_{k=1}^{v} \left( \left( x_k''^{(i)} - \overline{x_k''} \right)^2 + \left( y_k''^{(i)} - \overline{y_k''} \right)^2 \right) \tag{D.19}$$

berechnen. Die Berechnung der mittleren Form $\overline{s''}$ erfolgt dabei analog zu (D.13).
Die Rotation des Punktes $\left( x_k''^{(i)}, y_k''^{(i)} \right)$ in Richtung $\left( \overline{x_k''}, \overline{y_k''} \right)$ lässt sich durch folgende Matrixmultiplikation darstellen:

$$\begin{pmatrix} \overline{x_k''} \\ \overline{y_k''} \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x_k''^{(i)} \\ y_k''^{(i)} \end{pmatrix}. \tag{D.20}$$

Damit ergibt sich folgende Gleichung zur Minimierung des Fehlers:

$$E^{(i)} = \sum_{k=1}^{v} \left( \left( x_k''^{(i)} \cos\alpha - y_k''^{(i)} \sin\alpha - \overline{x_k''} \right)^2 + \left( x_k''^{(i)} \sin\alpha + y_k''^{(i)} \cos\alpha - \overline{y_k''} \right)^2 \right) \rightarrow min \tag{D.21}$$

Über die Lösung von (D.21) mittels Ableitung lässt sich der gesuchte Drehwinkel $\alpha$ wie folgt bestimmen:

$$\alpha = \arctan \frac{\sum_{k=1}^{v} \left( y_k''^{(i)} \overline{x_k''} - x_k''^{(i)} \overline{y_k''} \right)}{\sum_{k=1}^{v} \left( x_k''^{(i)} \overline{x_k''} + y_k''^{(i)} \overline{y_k''} \right)} \tag{D.22}$$

Die neuen Positionen der Labelpunkte ergeben sich damit zu:

$$x_k^{*(i)} = x_k''^{(i)} \cos\alpha - y_k''^{(i)} \sin\alpha \tag{D.23}$$

$$y_k^{*(i)} = x_k''^{(i)} \sin\alpha + y_k''^{(i)} \cos\alpha \tag{D.24}$$

Als Ergebnis liegt nun eine Menge von Formen $s^{*(i)} = \left( x_1^{*(i)}, y_1^{*(i)}, ..., x_v^{*(i)}, y_v^{*(i)} \right)$ vor, die zentriert zum Ursprung liegen, normiert auf die Länge Eins sind und näherungsweise gleichförmig ausgerichtet sind.

## D.3 Hintergrund-Vordergrund-Segmentierung mittels Differenzbildverfahren und Closing- und Connected-Regions-Algorithmus

Der Inhalt dieses Abschnitt ist entnommen aus [Steege, 2007][1]

Das Differenzbild-Verfahren ist ein sehr einfaches Verfahren zur Trennung eines statischen Hintergrundes von einem bewegten Vordergrund. Es werden dafür zeitlich versetzt zwei Bilder aufgenommen. Anschließend werden die beiden Bilder miteinander verglichen. Hat ein Pixel in einem Bild einen Grauwert, der um mehr als eine festgesetzte Schwelle $S$ vom Grauwert des gleichen Pixels im anderen Bild abweicht, so wird angenommen, dass an dieser Stelle des Bildes Bewegung stattgefunden hat bzw. sich ein Objekt im Vordergrund befindet. Das entsprechende Pixel wird markiert. In Abbildung D.4(3) ist ein so entstandenes Differenzbild dargestellt, wobei markierte Pixel weiß eingezeichnet sind. Wie zu sehen ist, sind in dem resultierenden Differenzbild viele Ausreißer an Stellen, wo tatsächlich aber keine Bewegung stattfand. An anderen Stellen ist das Differenzbild hingegen unvollständig. Dies ist zurückzuführen auf das Bildrauschen und leichte Helligkeitsschwankungen einerseits und zum anderen darauf, dass Hintergrund und Vordergrund zum Teil ähnliche Grauwerte haben und so durch den Differenzbild-Algorithmus nicht erfasst werden. Ein Anheben oder Absenken der Schwelle $S$ bringt keine Verbesserung, da bei einer Absenkung die Menge der Ausreißer größer wird und bei einer Anhebung die Lücken im Differenzbild zunehmen.


***Abbildung D.4:*** *Ablauf einer Hintergrundsubtraktion mittels Differenzbild. (1.): Bild einer Person vor einer Zeigegeste und (2.): während einer Zeigegeste. (3.): Differenzbild. Weiß markiert sind Stellen, an denen sich Bild 1 und 2 um mehr als eine festgelegte Schwelle im Grauwert unterscheiden. (4.): Durch Anwenden eines Closing- und Connected-Regions-Algorithmus entsteht aus dem Differenzbild eine Maske, die über das Eingangsbild gelegt wird und die zeigende Person extrahiert.*

Mehr Erfolg bringt das Anwenden eines *Closing-Algorithmus* und die Suche nach verbundenen Regionen im Bild (*Connected-Regions*). Der in dieser Dissertation verwendete Closing-Algorithmus sucht im Bild nach markierten Pixeln, die weniger als eine definierte Distanz $d$

[1] Diese Diplomarbeit wurde vom Autor im Rahmen dieser Dissertation betreut.

in horizontaler oder vertikaler Richtung von einem anderen markierten Pixel entfernt sind. Werden solche Pixel gefunden, so werden die Pixel dazwischen ebenfalls markiert. Mit dieser Verfahrensweise können Lücken im Differenzbild geschlossen werden. Die Funktionsweise des Algorithmus ist in Abbildung D.5 genauer dargestellt.

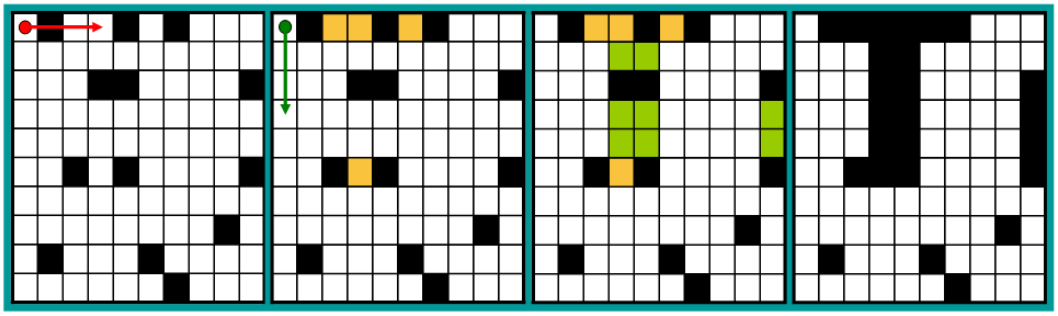


***Abbildung D.5:*** *Closing-Algorithmus: Das Bild wird zuerst zeilenweise, dann spaltenweise durchlaufen. Ist ein schwarzes Pixel weniger als eine definierte Schwelle (in diesem Beispiel 3 Pixel) von einem weiteren schwarzen Pixel entfernt, so werden die dazwischen liegenden Pixel ebenfalls schwarz gefärbt. Die im ersten Durchlauf eingefärbten Pixel sind in diesem Beispiel orange markiert, die im zweiten Durchgang eingefärbten Pixel sind hellgrün markiert.*

Durch den Closing-Algorithmus können Lücken im Differenzbild geschlossen werden. Um die Zahl der Ausreißer im Differenzbild zu begrenzen, wird im Bild nach zusammenhängenden Regionen gesucht. Jedes markierte Pixel im Bild erhält die Nummer einer Region, falls es an eine bestehende Region angrenzt, oder eröffnet eine neue Region. Gleichzeitig wird die Pixelanzahl in den bestehenden Regionen gezählt. Am Ende werden alle Regionen verworfen, die eine bestimmte Mindestgröße $g_{min}$ nicht erreichen und die Markierungen der zur jeweiligen Region gehörenden Pixel werden gelöscht. So können kleine Regionen von Ausreißern im Bild vermieden werden. Der Ablauf des Verfahrens ist in Abbildung D.6 graphisch dargestellt.

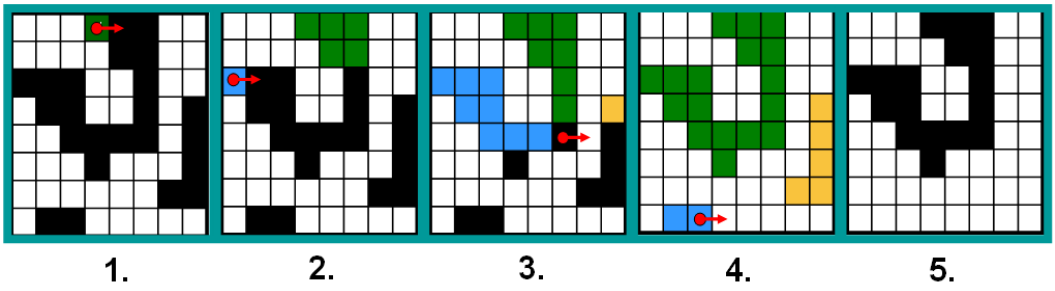



***Abbildung D.6:*** *Suche nach Connected-Regions: Das Bild wird zeilenweise durchlaufen. Die aktuelle Position ist mit dem roten Punkt markiert. Wird ein neuer schwarzer Pixel gefunden, der nicht an eine Region grenzt, wird eine neue Region erstellt (1.) (grün gekennzeichnet). Pixel, die an eine bestehende Region angrenzen, werden zu dieser Region gezählt. In (2.) trifft der Cursor erneut auf einen Pixel, der an keine bestehende Region grenzt. Es wird eine neue Region angelegt (blau gekennzeichnet). An der Stelle (3.) trifft der Cursor auf einen Pixel, der an zwei Regionen grenzt. Beide Regionen werden vereint mit der zuerst benutzten Farbe (hier grün). (4.) zeigt die drei gekennzeichneten Regionen nach Durchlaufen des Bildes. Anschließend werden Regionen gelöscht, die kleiner als die minimale Grenze sind (im Beispiel gleich 10) und es entsteht Bild (5.).*

## D.4 Gram-Schmidtsches Orthogonalisierungsverfahren

Beim *Gram-Schmidtschen Orthogonalisierungsverfahren* handelt es sich um einen Algorithmus aus der linearen Algebra, der zu einem gegebenen System linear unabhängiger Vektoren ein Orthogonalsystem für denselben Untervektorraum erzeugt. Das Verfahren ist benannt nach Jørgen Pedersen Gram und Erhard Schmidt.

Im Folgenden sei $\langle \mathbf{a}, \mathbf{b} \rangle$ das Skalarprodukt der Vektoren $\mathbf{a}$ und $\mathbf{b}$. Gegeben sind die linear unabhängigen Vektoren $\mathbf{w_1}, ..., \mathbf{w_n}$. Für diese wird ein Orthogonalsystem von $n$ paarweisen orthogonalen Vektoren $\mathbf{v_1}, ..., \mathbf{v_n}$ gesucht, das denselben Unterraum erzeugt. Diese Vektoren berechnen sich wie folgt:

$$
\begin{aligned}
\mathbf{v_1} &= \mathbf{w_1} \\
\mathbf{v_2} &= \mathbf{w_2} - \frac{\langle \mathbf{v_1}, \mathbf{w_2} \rangle}{\langle \mathbf{v_1}, \mathbf{v_1} \rangle} \cdot \mathbf{v_1} \\
\mathbf{v_3} &= \mathbf{w_3} - \frac{\langle \mathbf{v_1}, \mathbf{w_3} \rangle}{\langle \mathbf{v_1}, \mathbf{v_1} \rangle} \cdot \mathbf{v_1} - \frac{\langle \mathbf{v_2}, \mathbf{w_3} \rangle}{\langle \mathbf{v_2}, \mathbf{v_2} \rangle} \cdot \mathbf{v_2} \\
&\vdots \\
\mathbf{v_n} &= \mathbf{w_n} - \sum_{i=1}^{n-1} \frac{\langle \mathbf{v_i}, \mathbf{w_n} \rangle}{\langle \mathbf{v_i}, \mathbf{v_i} \rangle} \cdot \mathbf{v_i}
\end{aligned}
\tag{D.25}
$$

Eine Erweiterung stellt das *Gram-Schmidtschen Orthonormalisierungsverfahren* dar: Dieses bestimmt zu einer gegebenen Menge von linear unabhängigen Vektoren $\mathbf{w_1}, ..., \mathbf{w_n}$ ein Orthonormalsystem. Dazu wird wie folgt vorgegangen:

$$
\begin{aligned}
\mathbf{v_1} &= \frac{\mathbf{w_1}}{\|\mathbf{w_1}\|} && \text{(Normalisieren des Vektors } w_1\text{)} \\
\mathbf{v_2'} &= \mathbf{w_2} - \langle \mathbf{v_1}, \mathbf{w_2} \rangle \cdot \mathbf{v_1} && \text{(Orthogonalisieren des Vektors } w_2\text{)} \\
\mathbf{v_2} &= \frac{\mathbf{v_2'}}{\|\mathbf{v_2'}\|} && \text{(Normalisieren des Vektors } v_2'\text{)} \\
\mathbf{v_3'} &= \mathbf{w_3} - \langle \mathbf{v_1}, \mathbf{w_3} \rangle \cdot \mathbf{v_1} - \langle \mathbf{v_2}, \mathbf{w_3} \rangle \cdot \mathbf{v_2} && \text{(Orthogonalisieren des Vektors } w_3\text{)} \\
\mathbf{v_3} &= \frac{\mathbf{v_3'}}{\|\mathbf{v_3'}\|} && \text{(Normalisieren des Vektors } v_3'\text{)} \\
&\vdots \\
\mathbf{v_n'} &= \mathbf{w_n} - \sum_{i=1}^{n-1} \langle \mathbf{v_i}, \mathbf{w_n} \rangle \cdot \mathbf{v_i} && \text{(Orthogonalisieren des Vektors } w_n\text{)} \\
\mathbf{v_n} &= \frac{\mathbf{v_n'}}{\|\mathbf{v_n'}\|} && \text{(Normalisieren des Vektors } v_n'\text{)}
\end{aligned}
\tag{D.26}
$$

## D.5 Histograms of Oriented Gradients (HOG)

In [Dalal and Triggs, 2005] wurde ein Verfahren zur Detektion von Personen in einem Inputbild basierend auf *Histograms of Oriented Gradients (HOG)* vorgestellt, dass deutlich bessere Detektionsergebnisse als bisherige Verfahren erreicht hat. Im Folgenden werden die Grundidee und die angewandte Vorgehensweise zur Erstellung eines HOG-Detektors für Oberkörper beschrieben.

### D.5.1 Grundidee der HOG

Der Grundgedanke des Verfahrens ist, dass die lokale Form und das Aussehen eines Objektes sehr gut durch eine Verteilung von lokalen Gradientenorientierungen beschrieben werden können, ohne die genaue Position der Gradienten kennen zu müssen.
Die Bestimmung eines HOG-Featurevektors wird realisiert, indem ein Inputbild in kleine Zellen (die sich auch überlappen können) zerlegt wird. Für jede dieser Zellen wir ein einfaches 1-D-Histogramm der Gradientenorientierungen über alle Pixel der Zelle erstellt. Zwecks Robustheit gegenüber Helligkeitsschwankungen wird vor der Berechnung des Histogramms eine Kontrastnormalisierung unter Berücksichtigung benachbarter Bildregionen durchgeführt. Als Ergebnis entsteht ein hochdimensionaler HOG-Featurevektor.

Beim Einsatz der HOGs zur Detektion von Personen, wird ein Suchfenster (ggf. in verschiedenen Größenstufen) über das Inputbild geschoben und jeweils der HOG-Featurevektor bestimmt. Mit Hilfe einer herkömmlichen SVM wird für jede Position im Inputbild eine Detektionsentscheidung getroffen. In einem Nachverarbeitungsschritt können noch sehr eng zusammenliegende Detektionen zusammengefasst werden.

Das erstellte HOG-Framework wurde in [Dalal and Triggs, 2005] auf der *MIT Pedestrian Database* und der *INRIA Database* [Dalal, 2005] getestet. Das Training des SVM-Klassifikators erfolgte auf Basis von ca. 1.200 Positivbeispielen und ca. 12.000 Negativbeispielen (je 10 zufällig gewählte Ausschnitte aus 1218 Fotos). Abbildung D.7 zeigt einige Beispiele aus der Datenbank.

In [Dalal and Triggs, 2005] wird mit einer Zellgröße von 8x8 Pixeln und einer Fenstergröße von 64x128 Pixeln gearbeitet. Bei einem Inputbild der Größe von 320x240 ergeben sich etwa 4.000 Detektionsfenster, die in weniger als einer Sekunde berechnet werden konnten. Eine Echtzeitfähigkeit des Gesamtsystem konnte jedoch nicht gezeigt werden.

### D.5.2 Training eines HOG-Detektors für Oberkörper

Das *INRIA Object Detection and Localization Toolkit* [Dalal, 2007] enthält einen vortrainierten HOG-Detektor für vollständig abgebildete Personen. Bei der Anwendung auf

Quelle: [Dalal, 2005]

***Abbildung D.7:*** *Beispielbilder aus der INRIA-Database [Dalal, 2005]. Alle Personen stehen aufrecht, sind jedoch teilweise verdeckt und in verschiedenen Posen abgebildet.*

Bildern, die nur einen Oberkörper enthalten, liefert dieser Detektor nur sehr schlechte Ergebnisse [Marin-Jimenez et al., 2008].

Von der *Robotics Research Group* der *University of Oxford* wird ein HOG-Detektor für den Oberkörper als Software zur Verfügung gestellt [Marin-Jimenez et al., 2008]. Dieser Detektor wurde hauptsächlich auf Frontalbildern und leicht seitlich stehenden Personen (bis ca. 30°) trainiert.

Zwecks Nachvollziehbarkeit wurde im Rahmen dieser Dissertation ein eigener HOG-Detektor mit Hilfe eines Bootstrapping-Prozesses auf Basis der Software von [Dalal, 2007] und [Marin-Jimenez et al., 2008] trainiert. Dazu wurde im Detail wie folgt vorgegangen:

- Anwendung des Oberkörper-HOG-Detektors von [Marin-Jimenez et al., 2008] auf den 460 Personenbildern der *GRAZ01-Database* [Opelt and Pinz, 2003].
- Als Ergebnis wurden 552 Detektionen gefunden. Ein erheblicher Teil davon sind jedoch Fehldektionen und müssen daher manuell aussortiert werden. Als Ergebnis sind 287 Positiv-Beispiele entstanden.
- Für das Training eines HOG-Detektors werden weiterhin eine Reihe von Negativ-Beispielen benötigt. Dazu wurden die entsprechenden Bilder der *INRIA Database* [Dalal, 2005] verwendet.
- Anschließend werden die HOGs für die Positiv- und Negativ-Beispiele berechnet. Danach wird eine erste SVM erstellt.
- Die trainierte SVM trennt die beiden Regionen typischerweise aber noch nicht optimal. Daher werden sog. *hard examples* ermittelt, die bei einem zweiten nachfolgendem SVM-Training mitbenutzt werden. Als Ergebnis entsteht eine SVM, welche die ursprünglichen Positiv- und Negativ-Beispiele deutlich besser trennen kann.

Abbildung D.8 zeigt Bilder der Positiv- und Negativ-Beispiele aus dem HOG-Training.

***Abbildung D.8:*** *Beispielbilder aus dem HOG-Trainingsprozess: Oben: Bilder mit Detektionen aus GRAZ01-Database [Opelt and Pinz, 2003]. Mitte: Ausgeschnittene Positiv-Beispiele. Unten: Negativ-Beispiele aus INRIA Database [Dalal, 2005]*

Als Ergebnis entstand ein HOG-Detektor, der die Anforderungen zur Initialisierung des Teilsystems zur Schätzung der Oberkörperpose hinreichend gut erfüllen konnte.

# E Weitere Ergebnisgrafiken

## E.1 Oberkörperschätzung - weitere Grafiken

Im folgenden Abschnitt findet sich die Darstellung weiterer Ergebnisse für die Oberkörperschätzung basierend auf einer *Nearest-Neighbour* Klassifikation (siehe Abschnitt 3.6.5), einem *Multi-Layer-Perceptron* (siehe Abschnitt 3.6.6) und einer *Support-Vector-Machine* (siehe Abschnitt 3.6.7).

Die Testdatenbank enthält für jede Person eine Reihe von Bildern, die als Videosequenz eine volle Umdrehung der Testperson vor der Kamera zeigt. Für jedes dieser Bilder wurde die Schätzung der Oberkörperorientierung mit Hilfe der drei vorgestellten Verfahren durchgeführt. Betrachtet man dies als zeitliche Abfolge, kann somit die geschätzte Oberkörperorientierung bezüglich der Zeit bzw. der Umdrehung der Person vor der Kamera grafisch dargestellt werden.

Die Abbildungen E.1 bis E.6 zeigen die Ergebnisse im 180°- und 360°-System für sechs ausgewählte Testpersonen. Im Idealfall würden alle Ergebnispunkte der Schätzung (rote Kreuze) auf der gestrichelten Linie (*Ground Truth*) liegen.

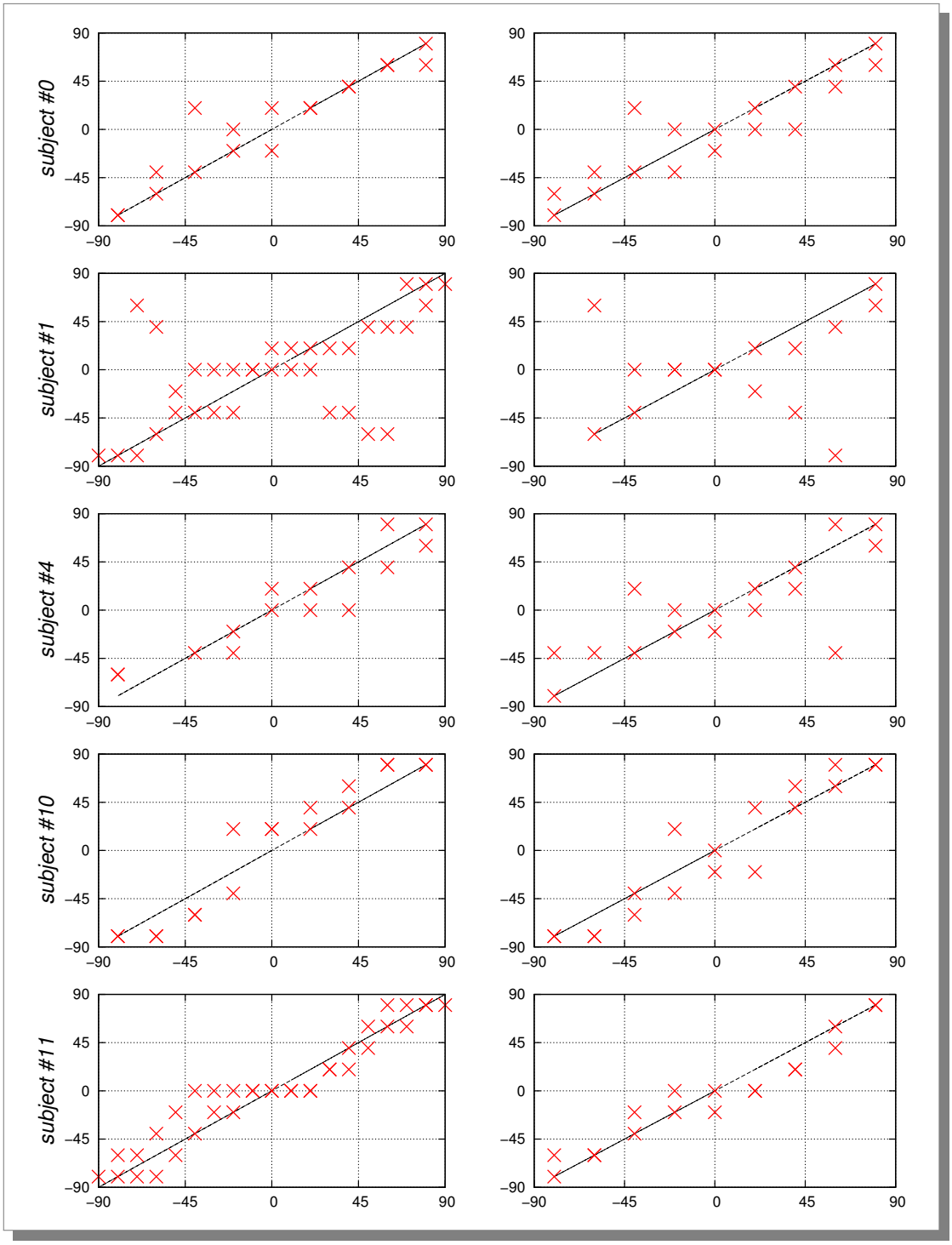



***Abbildung E.1:*** *Weitere Ergebnisse im 180°-System für eine Schätzung der Oberkörperorientierung mit Hilfe eines Nearest-Neighbour-Schätzers.*

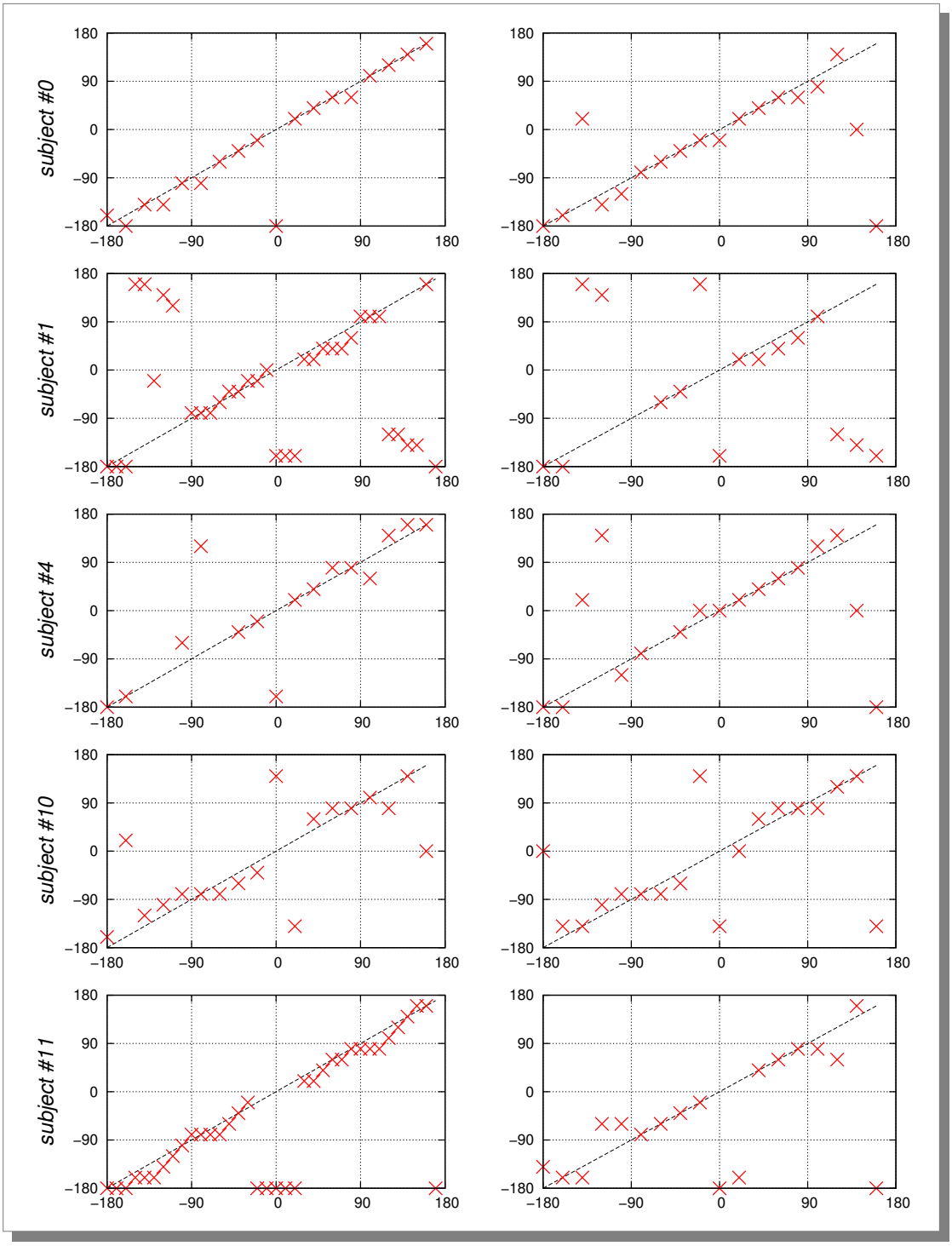



***Abbildung E.2:*** *Weitere Ergebnisse im 360°-System für eine Schätzung der Oberkörperorientierung mit Hilfe eines Nearest-Neighbour-Schätzers.*

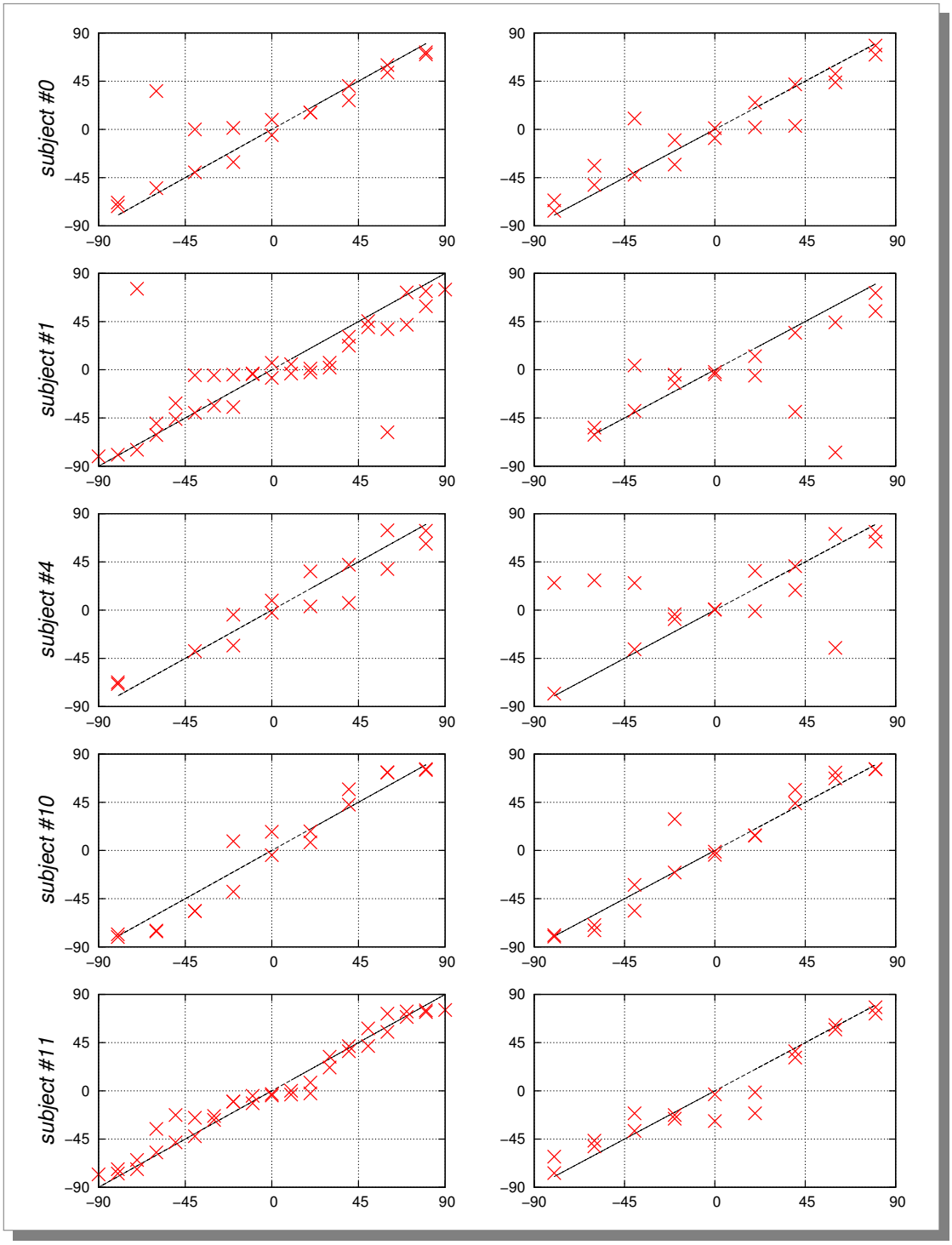



***Abbildung E.3:*** *Weitere Ergebnisse der MLP-basierten Schätzung der Oberkörperorientierung im 180°-System bei einer 3-6-7 Architektur.*

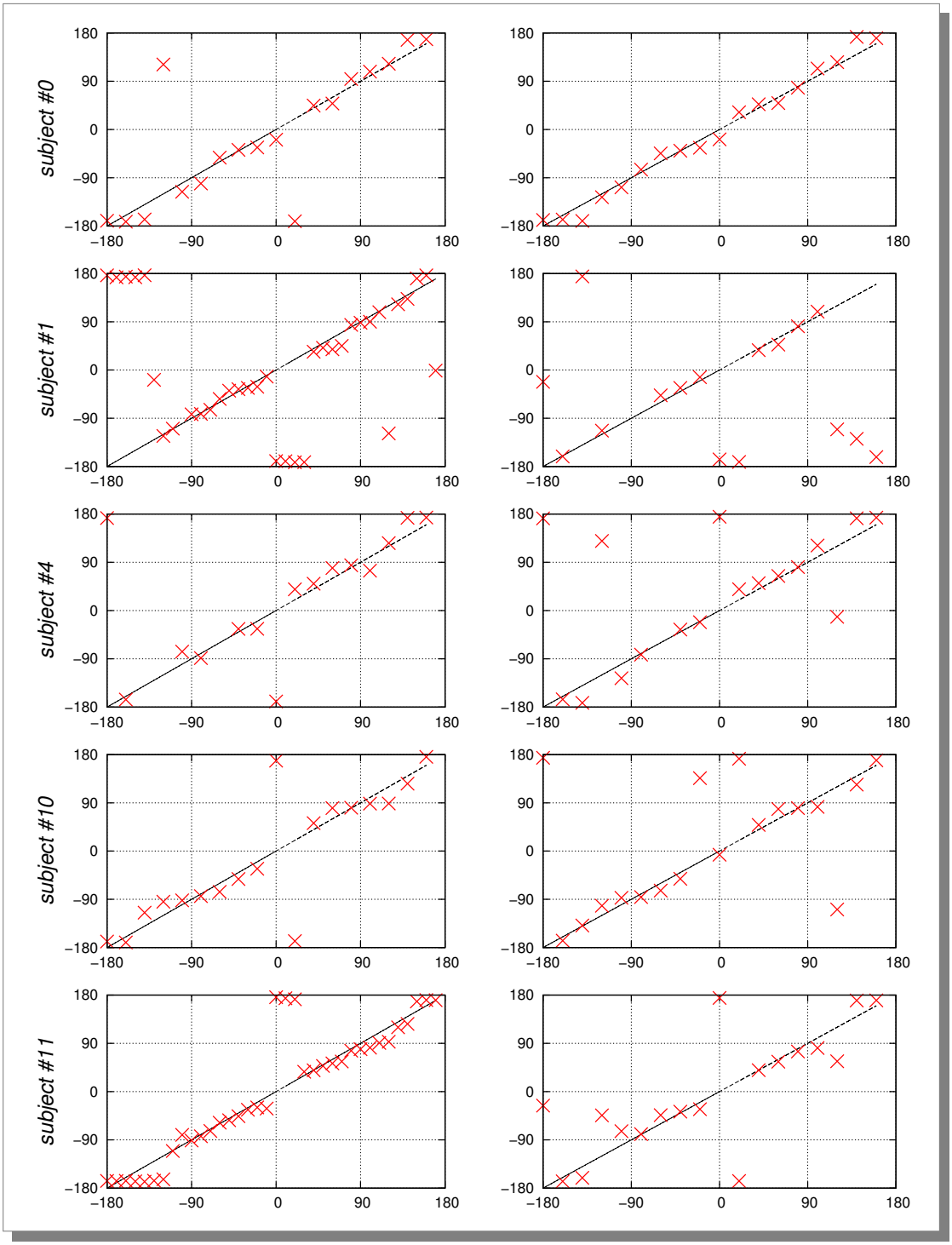



***Abbildung E.4:*** *Weitere Ergebnisse der MLP-basierten Schätzung der Oberkörperorientierung im 360°-System bei einer 3-6-9 Architektur.*

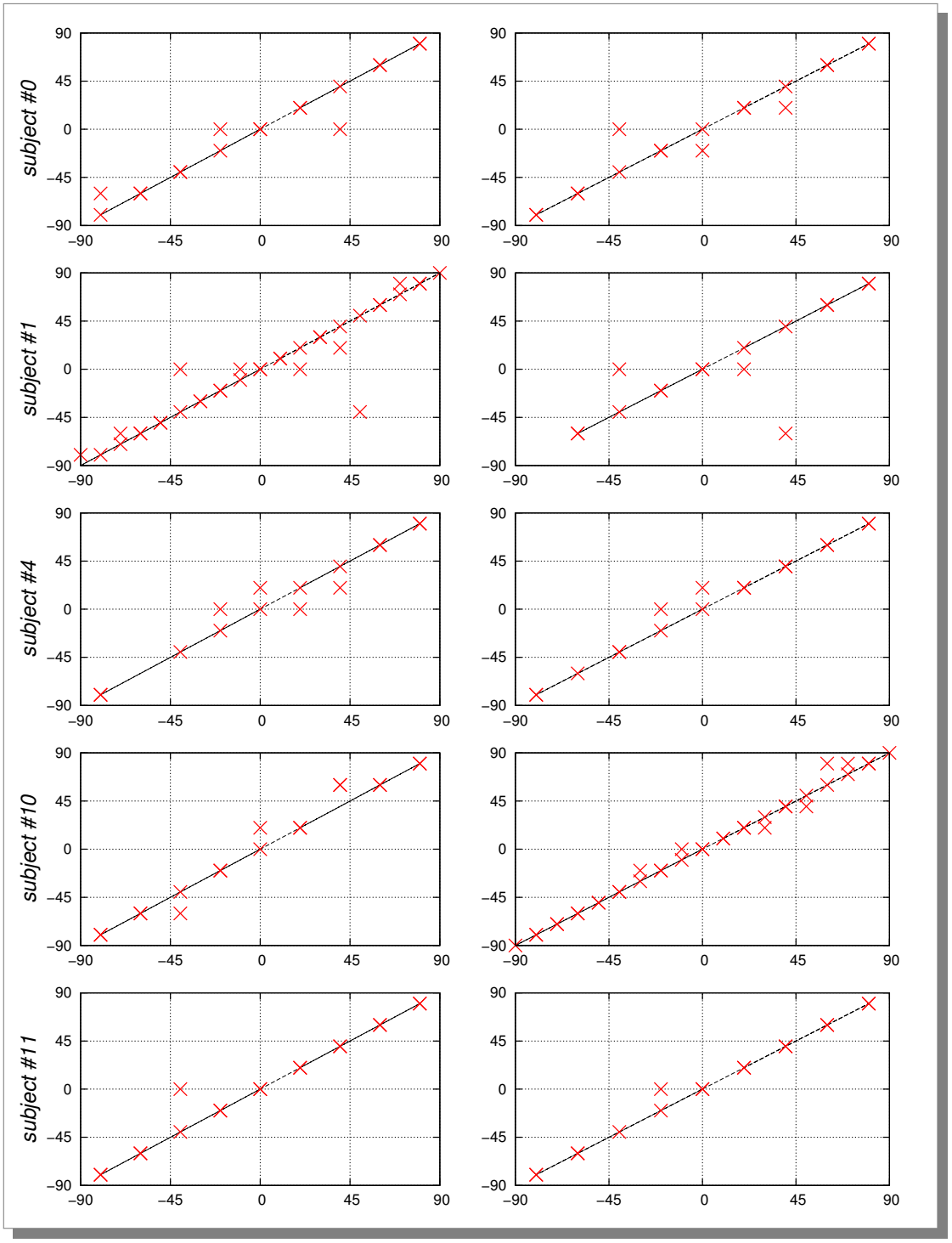



***Abbildung E.5:*** *Weitere Ergebnisse der SVM-basierten Schätzung der Oberkörperorientierung im 180°-System.*

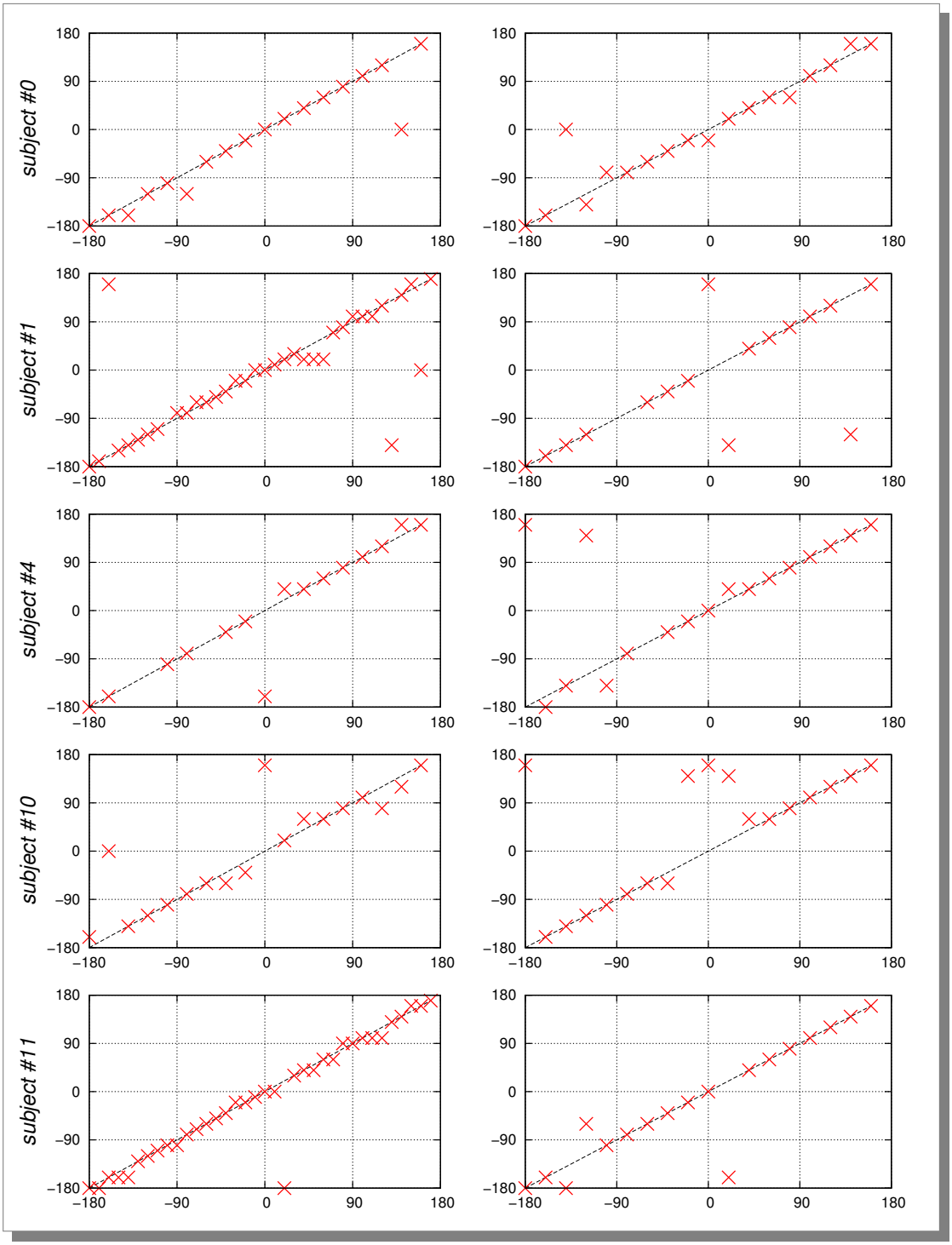



***Abbildung E.6:*** *Weitere Ergebnisse der SVM-basierten Schätzung der Oberkörperorientierung im 360°-System.*

# Literaturverzeichnis

[ALIAS, 2010] ALIAS (2010). The ALIAS Project - Adaptable Ambient LIving ASsistant. Webseite. `http://www.aal-alias.eu`.

[Allili and Ziou, 2007] Allili, M. and Ziou, D. (2007). Object of Interest segmentation and Tracking by Using Feature Selection. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1–8.

[Andriluka et al., 2009] Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1014–1021.

[Argyle and Cook, 1976] Argyle, M. and Cook, M. (1976). *Gaze and mutual gaze*. Cambrigde Univerity Press, Cambrigde.

[Babies, 2007] Babies, B. (2007). Bestimmung der Kopfpose und Untersuchung der Posenunabhängigkeit der Alters-, Geschlechts- und Indentifikationsschätzung mit Hilfe von Active Appearance Models. Technische Universität Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik, Diplomarbeit. Betreuer: Thorsten Wilhelm, Christian Martin, Betreuender Hochschullehrer: Horst-Michael Groß.

[Baker et al., 2003a] Baker, S., Gross, R., and Matthews, I. (2003a). Lucas-Kanade 20 Years On: A Unifying Framework: Part2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, Pittsburgh.

[Baker et al., 2003b] Baker, S., Gross, R., and Matthews, I. (2003b). Lucas-Kanade 20 Years On: A Unifying Framework: Part3. Technical Report CMU-RI-TR-03-35, Robotics Institute, Carnegie Mellon University, Pittsburgh.

[Baker and Matthews, 2001] Baker, S. and Matthews, I. (2001). Equivalence and Efficiency of Image Alignment Algorithms. In *Proc. of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1090–1097.

[Baker and Matthews, 2004] Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255.

[Baker et al., 2004] Baker, S., Matthews, I., Xiao, J., Gross, R., Kanade, T., and Ishikawa, T. (2004). Real-time non-rigid Driver Head Tracking for Driver Mental State Estimation. In *Proc. of 11th World Congress Intelligent Transportation Systems*.

[Bartlett, 2001] Bartlett, M. (2001). *Face Image Analysis by Unsupervised Learning*. Kluwer Academic Publishers.

[Bartlett et al., 2002] Bartlett, M., Movelan, J., and Sejnowski, T. (2002). Face Recognition by Independent Component Analysis. *IEEE Transactions on Neural Networks*, 13(6):1450–1464.

[Battiti, 1994] Battiti, R. (1994). Using Mutal Information for Selecting Features in Supervised Neural Network Learning. *IEEE Trans. Neural Networks*, 5(4):537–550.

[Bell and Sejnowski, 1995] Bell, A. and Sejnowski, T. (1995). An Information Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(.):1129–1159.

[Belz, 2008] Belz, S. (2008). Optimierung eines Personentrackers auf Basis von Konturen zum Einsatz auf einem mobilen Robotersystem. Technische Universität Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik, Diplomarbeit. Betreuer: Christian Martin, Betreuender Hochschullehrer: Horst-Michael Groß.

[Blanz and Vetter, 1999] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194.

[Breazeal, 1999] Breazeal, C. (1999). Robot in Society: Friend or Appliance? In *Proc. of Agents99 Workshop on Emotion-Based Agent Architectures*, pages 18–26.

[Bruce, 1988] Bruce, V. (1988). *Recognising Faces*. Hillsdale, NJ: Lawrence Erlbaum Associates Ltd.

[Brueckmann et al., 2007] Brueckmann, R., Scheidig, A., and Gross, H.-M. (2007). Adaptive Noise Reduction and Voice Activity Detection for Improved Verbal Human-Robot Interaction using Binaural Data. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1782–1787.

[Bruske et al., 1998] Bruske, J., Abraham-Mumm, E., Pauli, J., and Sommer, G. (1998). Head-pose estimation from facial images with Subspace Neural Networks. In *Proc. of the Int. Conf. on Neural Networks and Brain*, pages 528–531.

[Bruske and Sommer, 1997] Bruske, J. and Sommer, G. (1997). Topology Representing Networks for Intrinsic Dimensionality Estimation. In *Proceedings of the 7th International Conference on Artificial Neural Networks*, pages 595–600.

[Bruske and Sommer, 1998] Bruske, J. and Sommer, G. (1998). Intrinsic Dimensionality Estimation With Optimally Topology Preserving Maps. Technical Report 9703, Computer Science Institute, Christian-Albrechts-Universität.

[BU Database, 1998] BU Database (1998). Boston University Face Tracking Database. Image and Video Computing Group - Boston University : `http://www.cs.bu.edu/groups/ivc/HeadTracking`.

[Burges, 1998] Burges, C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[Böhme et al., 2006] Böhme, H.-J., Scheidig, A., Wilhelm, T., Schröter, C., Martin, C., König, A., Müller, S., and Gross, H.-M. (2006). Progress in the Development of an Interactive Mobile Shopping Assistant. In *Proc. of the Joint Conf. on Robotics: 37th International Symposium on Robotics (ISR 2006) and 4th German Conference on Robotics (Robotik 2006)*, page 20 pages.

[Cardoso, 1999] Cardoso, J. (1999). Higher Order Contrasts for Independent Component Analysis. *Neural Computation*, 11(1):157–192.

[Catrin, 2006] Catrin, K. (2006). Emotion. Online im Internet: `http://de.wikipedia.org/wiki/Emotion`.

[Chang et al., 2005a] Chang, C., Ansari, R., , and Khokhar, A. (2005a). Multiple Object Tracking with Kernel Particle Filter. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 566–573.

[Chang et al., 2005b] Chang, Y., Vieira, M., Turk, M., and Velho, L. (2005b). Automatic 3D Facial Expression Analysis in Videos. In *Analysis and Modelling of Faces and Gestures, Proceedings*, pages 293–307.

[Cohn, 1999] Cohn, J. (1999). Cohn-Kanade AU-Coded Facial Expression Database. CD ROM. Pittsburgh University: `http://vasc.ri.cmu.edu/idb/html/face/facial\_expression/index.html`.

[CompanionAble, 2009] CompanionAble (2009). The CompanionAble Project - Integrated Cognitive Assistive & Domotic Companion Robotic Systems for Ability & Security. Webseite. `http://www.CompanionAble.com`.

[Cootes et al., 1998] Cootes, T., Edward, G., and Tylor, C. (1998). Active Appearance Models. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 484–498.

[Cootes et al., 2001] Cootes, T., Edwards, G., and Taylor, C. (2001). Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

[Cootes et al., 2000] Cootes, T., Walker, K., and Taylor, C. (2000). View-Based Active Appearance Models. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2000)*, pages 227–232.

[Cootes and Taylor, 2001] Cootes, T. F. and Taylor, C. (2001). Statistical models of appearance for medical image analysis and computer vision. In *Proceedings of SPIE Medical Imaging 2001*, pages 236–248.

[Cootes et al., 1995] Cootes, T. F., Taylor, C., Cooper, D. H., and Graham, J. (1995). Active shape models—their training and application. *Journal of Computer Vision and Image Understanding*, 61(1):38–59.

[Dalal, 2005] Dalal, N. (2005). INRIA Person Dataset. `http://pascal.inrialpes.fr/data/human`.

[Dalal, 2007] Dalal, N. (2007). INRIA Object Detection and Localization Toolkit. `http://pascal.inrialpes.fr/soft/olt/`.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *Int. Conf. on Computer Vision & Pattern Recognition*, pages 886–893.

[Dautenhahn, 2004] Dautenhahn, K. (2004). Robots We Like to Live With?! – A Developmental Perspective on a Personalized, Life-Long Robot Companion. In *Proc. of 13th IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN 2004)*, pages 17–22.

[Dautenhahn, 2007] Dautenhahn, K. (2007). Methodology & Themes of Human-Robot Interaction: A Growing Research Field. *Int. Journal of Advanced Robotic Systems*, 4(1):103–108.

[De la Torre et al., 2007] De la Torre, F., Collet, A., Quero, M., Cohn, J., and Kanade, T. (2007). Filtered Component Analysis to Increase Robustness to Local Minima in Appearance Models. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1–8.

[Deparis, 2004] Deparis, K. (2004). The harris corner detector. Departement of Computer Science and Engineering, York University, Canada. Seminar paper.

[Dimitrijevic et al., 2005] Dimitrijevic, M., Lepetit, V., and Fua, P. (2005). Human Body Pose Recognition Using Spatio-Temporal Templates. In *ICCV Workshop on Modeling People and Human Interaction*.

[Donato et al., 1999] Donato, G., Bartlett, M. S., Hager, J. C., Ekman, P., and Sejnowski, T. J. (1999). Classifying Facial Actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989.

[Dornaika and Ahlberg, 2004] Dornaika, F. and Ahlberg, J. (2004). Model-based head and facial motion tracking. In *Computer Vision in Human-Computer Interaction, ECCV 2004 Workshop on HCI*, pages 221–232.

[Duda and Hart, 1973] Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons Inc.

[Edwards et al., 1998] Edwards, G., Cootes, T., and Taylor, C. (1998). Face Recognition Using Active Appearance Models. $5^{th}$ *European Conference on Computer Vision 1998*, LNCS-Series 1406–1607:581–595.

[Einecke, 2006] Einecke, N. (2006). Contour Tracker. Department of Neuroinformatics and Cognitive Robotics, Ilmenau Technical University, Germany. Seminar paper.

[Ekman and Friesen, 1978] Ekman, P. and Friesen, W. (1978). *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto.

[Ekman et al., 1982] Ekman, P., Friesen, W. V., and Ellsworth, P. (1982). What emotion categories or dimensions can observers judge from facial behavior? In *Emotion in the human face*, pages 39–55.

[Essa and Pentland, 1997] Essa, E. and Pentland, A. (1997). Coding, Analysis, Interpretation, and Recognition of Facial Expressions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7).

[FANN, 2010] FANN (2010). Fast Artifical Neural Network Library (FANN). `http://leenissen.dk/fann/`.

[Fasel, 2002] Fasel, B. (2002). Multiscale Facial Expression Recognition using Convolutional Neural Networks. In *Proc. IEEE Int. Conf. on Multimodal Interfaces (ICMI 02)*.

[Ferrari et al., 2009] Ferrari, V., Marin-Jimenez, M., and Zisserman, A. (2009). Pose Search: Retrieving People using Their Pose. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997). A decisiontheoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science*, 55(1):119–139.

[Gavrila and Munder, 2007] Gavrila, D. M. and Munder, S. (2007). Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. *International Journal of Computer Vision (IJCV)*, 73(1):41–59.

[Girshick et al., 2011] Girshick, R., Shotton, J., Kohli, P., Criminisi, A., and Fitzgibbon, A. (2011). Efficient Regression of General-Activity Human Poses from Depth Images. In *Proc. of IEEE Int. Conference on Computer Vision (ICCV)*, pages 415–422.

[Goffman, 1963] Goffman, E. (1963). *Stigma*. Prentice-all, Englewood Cliffs, New Jersey.

[Goffman, 1971] Goffman, E. (1971). *Verhalten in sozialen Situationen - Strukturen und Regeln der Interaktion im öffentlichen Raum*. Fachverlag Bertelsmann, Gütersloh.

[Goodwin, 1981] Goodwin, C. (1981). *Conversational organization: Interaction between speakers and hearers*. Academic Press, New York.

[Gross et al., 2008] Gross, H.-M., Böhme, H.-J., Schröter, C., Müller, S., König, A., Martin, C., Merten, M., and Bley, A. (2008). ShopBot: Progress in Developing an Interactive Mobile Shopping Assistant for Everyday Use. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (IEEE-SMC 2008)*, pages 3471–3478.

[Gross et al., 2011] Gross, H.-M., Schroeter, C., Mueller, S., Volkhardt, M., Einhorn, E., Bley, A., Martin, C., Langner, T., and Merten, M. (2011). Progress in Developing a Socially Assistive Mobile Home Robot Companion for the Elderly with Mild Cognitive Impairment. In *Proc. IEEE/RJS Int. Conf. on Intelligent Robots and Systems (IROS 2011)*, pages 2430–2437.

[Gross et al., 2005] Gross, R., Matthews, I., and Baker, S. (2005). Generic vs. Person Specific Active Appearance Models. *Image and Vision Computing*, 23(1):1080–1093.

[Gross et al., 2006] Gross, R., Matthews, I., and Baker, S. (2006). Active Appearance Models with Occlusion. *Image and Vision Computing*, 24(1):593–604.

[Hager and Belhumeur, 1998] Hager, G. and Belhumeur, P. (1998). Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039.

[Hager et al., 2002] Hager, J. C., Ekman, P., and Friesen, W. (2002). *Facial action coding system.* A human face, Salt Lake City.

[Hanek and Beetz, 2004] Hanek, R. and Beetz, M. (2004). The Contracting Curve Density Algorithm: Fitting Parametric Curve Models to Images Using Local Selfadapting Separation Criteria. *International Journal of Computer Vision (IJCV)*, 59(3):233–258.

[Holthaus et al., 2010] Holthaus, Patrick amd Lütkebohle, I., Hanheide, M., and Wachsmuth, S. (2010). Can I Help You? - A Spatial Attention System for a Receptionist Robot. In *Proc. of the 2th Int. Conf. on Social robotics*, pages 325–334.

[Hommel and Handmann, 2011] Hommel, S. and Handmann, U. (2011). AAM based Continuous Facial Expression Recognition for Face Image Sequences. In *IEEE Int. Symposium on Computational Intelligence and Informatics (CINTI)*, pages 189–194.

[Hong et al., 1998] Hong, H., Neven, H., and von der Malsburg, C. (1998). Online Facial Expression Recognition based on Personalized Gallery. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 1998)*, pages 354–359.

[Hsu et al., 2003] Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A Practical Guide to Support Vector Classification. Technical Report, Department of Computer Science, National Taiwan University. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

[Hu et al., 2004] Hu, Y., Chen, L., Zhou, Y., and Zhang, H. (2004). Estimating Face Pose by Facial Asymmetry and Geometry. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004)*, pages 651–656.

[Huber, 1981] Huber, P. (1981). *Robust Statistics.* Wiley & Sons.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). Condensation - Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.

[Jafar-Shaghaghi, 1994] Jafar-Shaghaghi, F. (1994). *Maschinelles Lernen, neuronale Netze und statistische Lernverfahren zur Klassifikation und Prognose: theoretische Analyse und ökonomische Anwendung.* PhD thesis, Universität Karlsruhe.

[Jobson et al., 1997a] Jobson, D., Rahman, Z., and Woodell, G. (1997a). A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing*, 6(7):965–976.

[Jobson et al., 1997b] Jobson, D., Rahman, Z., and Woodell, G. (1997b). Properties and Performance of a Center/Surround Retinex. *IEEE Transactions on Image Processing*, 6(3):451–462.

[Kalliomäki and Lampinen, 2003] Kalliomäki, I. and Lampinen, J. (2003). Modeling of Pose Effects in Oriented Filter Responses for Head Pose Estimation. In *Image Analysis, 13th Scandinavian Conference, SCIA 2003*, pages 156–162.

[Kleinginna and Kleinginna, 2005] Kleinginna, P. and Kleinginna, A. (2005). A Categorized List of Emotion Definitions with Suggestions for a Consensual Definition. *Motivation and Emotion*, 5(4):345–379.

[Knerr et al., 1990] Knerr, S., Personnaz, L., and Dreyfus, G. (1990). Single-layer learning revisited: a stepwise procedure for building and training a neural network. In Fogelman, J., editor, *Neurocomputing: Algorithms, Architectures and Applications.* Springer-Verlag.

[Kobayashi and Hara, 1997] Kobayashi, H. and Hara, F. (1997). Facial Interaction Between Animated 3D Face Robot and Human Beings. *IEEE International Conference on Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'*, 4(4):3732–3737.

[Kohonen, 1982] Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43:59–69.

[Kreßel, 1999] Kreßel, U. (1999). Pairwise classification and support vector machines. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods & Support Vector Learning*, pages 255–268, Cambridge, MA. MIT Press.

[Krüger et al., 1997] Krüger, N., Pötzsch, M., and von der Malsburg, C. (1997). Determination of Face Position and Pose With a Learned Representation Based on Labelled Graphs. *Image Vision Comput.*, 15(8):665–673.

[Krüger and Sommer, 2002] Krüger, V. and Sommer, G. (2002). Gabor Wavelet Networks for Efficient Head Pose Estimation. *Image Vision Comput.*, 20(9-10):665–672.

[Kumano et al., 2009] Kumano, S., Otsuka, K., Yamato, J., Maeda, E., and Sato, Y. (2009). Pose-Invariant Facial Expression Recognition Using Variable-Intensity Templates. *International Journal of Computer Vision (IJCV)*, 83(2).

[Kán, 2010] Kán, P. (2010). 3d head model reconstruction from photographs. *Computer Graphics & Geometry (CG&G) Internal-Journal*, 12(1):17–39.

[Lanitis et al., 1997] Lanitis, A., Taylor, C., and Cootes, T. (1997). Automatic Interpretation and Coding of Face Images Using Flexible Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756.

[Lee and Cohen, 2004] Lee, M. W. and Cohen, I. (2004). Human Upper Body Pose Estimation in Static Images. In *Proc. of European Conf. Computer Vision*, pages 126–138.

[Levenberg, 1944] Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least Squares. *The Quarterly of Applied Mathematics*, pages 164–168.

[Li et al., 2010] Li, L., Hoe, K. E., Yu, X., Dong, L., and Chu, X. (2010). Human Upper Body Pose Recognition Using Adaboost Template for Natural Human Robot Interaction. In *Proc. of the 2010 Canadian Conference on Computer and Robot Vision*, pages 370–377.

[libSVM, 2010] libSVM (2010). LIBSVM - A Library for Support Vector Machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[Lucas and Kanade, 1981] Lucas, B. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. of Internation Joint Conference on Artificial Intelligenz*, pages 674–679.

[Lucey et al., 2006] Lucey, S., Matthews, I., Hu, C., Ambadar, Z., De la Torre Frade, F., and Cohn, J. (2006). AAM Derived Face Representations for Robust Facial Action recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2006)*, pages 155–160.

[Lundqvist et al., 1998] Lundqvist, D., Flykt, A., and Öhman, A. (1998). The Karolinska Directed Emotional Faces. CD ROM. Department of Clinical Neuroscience, Psychology section, Karolinska Institutet.

[Marin-Jimenez et al., 2008] Marin-Jimenez, M., Ferrari, V., and Zisserman, A. (2008). Visual Geometry Group. Upper-body detector. `http://www.robots.ox.ac.uk/~vgg/software/UpperBody/index.html`.

[Marquardt, 1963] Marquardt, D. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441.

[Martin and Gross, 2008] Martin, C. and Gross, H.-M. (2008). A Real-time Facial Expression Recognition System based on Active Appearance Models using Gray Images and Edge Images. In *Proc. of the 8th IEEE Int. Conference on Face and Gesture Recognition (FG'08)*, pages paper no. 299, 6 pages.

[Martin et al., 2005a] Martin, C., Schaffernicht, E., Scheidig, A., and Gross, H.-M. (2005a). Sensor Fusion using a Probabilistic Aggregation Scheme for People Detection and Tracking. In *Proc. of the 2nd European Conference on Mobile Robots (ECMR 2005)*, pages 176–181.

[Martin et al., 2005b] Martin, C., Scheidig, A., Wilhelm, T., Schröter, C. Böhme, H.-J., and Gross, H.-M. (2005b). A new Control Architecture for Mobile Interaction Robots.. In *Proc. of the 2nd European Conference on Mobile Robots (ECMR 2005)*, pages 224–229.

[Martin et al., 2010] Martin, C., Steege, F.-F., and Gross, H.-M. (2010). Estimation of Pointing Poses for Visual Instructing Mobile Robots under Real-World Conditions. *Robotics and Autonomous Systems*, 58(2):174–185.

[Matthews and Baker, 2004] Matthews, I. and Baker, S. (2004). Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164.

[MetraLabs, 2011] MetraLabs (2011). Homepage der MetraLabs GmbH - Neue Technologien und Systeme. Webseite. `http://www.MetraLabs.com`.

[Mokhtarian and Suomela, 1998] Mokhtarian, F. and Suomela, R. (1998). Robust Image Corner Detection Through Curvature Scale Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1376–1381.

[Molcho, 1983] Molcho, S. (1983). *Körpersprache.* Mosaik Verlag GmbH, München.

[Molcho, 2006] Molcho, S. (2006). *ABC der Körpersprache.* Ariston Verlag, München.

[Murphy-Chutorian and Trivedi, 2009] Murphy-Chutorian, E. and Trivedi, M. (2009). Head Pose Estimation in Computer Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626.

[Mutlu et al., 2009] Mutlu, B., Yamaoka, F., Kanda, T., Ishiguro, H., and Hagita, N. (2009). Nonverbal Leakage in Robots: Communication of Intentions through Seemingly Unintentional Behavior. In *Proc. of the 4th ACM/IEEE Int. Conf. on Human Robot Interaction*, pages 69–76.

[Müller et al., 2008] Müller, S., Hellbach, S., Schaffernicht, E., Ober, A., Scheidig, A., and Gross, H.-M. (2008). Whom to talk to? Estimating user interest from movement trajectories. In *Proc. of the 17th IEEE Int. Symposium on Robot and Human Interactive Communication, (RO-MAN 08)*, pages 532–538.

[Nordstrøm et al., 2004] Nordstrøm, M., Larsen, M., Sierakowski, J., and Stegmann, M. (2004). The IMM Face Database - An Annotated Dataset of 240 Face Images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU. `http://www2.imm.dtu.dk/~aam/`.

[Nyvärinen and Oja, 1997] Nyvärinen, A. and Oja, E. (1997). A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation*, 9(7).

[Oka et al., 2005] Oka, K. amd Sato, Y., Nakanishi, Y., and Koike, H. (2005). Head Pose Estimation System Based on Particle Filtering with Adaptive Diffusion Control. In *Proc. Int.Conf. Machine Vision Applications*, pages 586–589.

[Opelt and Pinz, 2003] Opelt, A. and Pinz, A. (2003). GRAZ-01 Database. `http://www.emt.tugraz.at/~pinz/data/GRAZ_01`.

[Ortony and Turner, 1990] Ortony, A. and Turner, T. J. (1990). What's basic about basic emotions? *Psychological Review*, 97(3):315–331.

[Panin et al., 2006] Panin, G., Ladikos, A., , and Knoll, A. (2006). An effcient and robust real-time contour tracking system. In *Proc. of IEEE Int. Conf. on Computer Vision Systems (ICVS)*, pages 44–52.

[Pantic and Rothkrantz, 2000a] Pantic, M. and Rothkrantz, L. (2000a). An Expert System for Recognition of Facial Actions and their Intensity. *Image and Vision Computing*, 18:881–905.

[Pantic and Rothkrantz, 2000b] Pantic, M. and Rothkrantz, L. (2000b). Automatic Analysis of Facial Expressions: The State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445.

[Pentland et al., 1994] Pentland, A., Moghaddam, B., and Starner, T. (1994). View-based and Modular Eigenspaces for Face Recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, pages 84–91.

[Piccardi, 2004] Piccardi, M. (2004). Background subtraction techniques: a review. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (SMC)*.

[Pomerleau, 1989] Pomerleau, D. (1989). ALVINN: An Autonomous Land Vehicle In a Neural Network. In *Advances in Neutral Information Processing Systems 1*, pages 305–313.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1):81–106.

[Ratliff and Patterson, 2008] Ratliff, M. S. and Patterson, E. (2008). Emotion Recognition using Facial Expressions with Active Appearance Models. In *Proc. of Int. Conference on Human Computer Interaction*.

[Ritter et al., 1991] Ritter, H., Martinetz, T., and Schulten, K. (1991). *Neuronale Netze*. Addison-Wesley.

[Ross, 2004] Ross, A. (2004). Procrustes analysis. Technical Report, Department of Computer Science and Engineering, University of South Carolina, SC 29208. `http://www.cse.sc.edu/~songwang/CourseProj/proj2004/ross/ross.pdf`.

[Rosten and Drummond, 2005] Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511.

[Rosten and Drummond, 2006] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 430–443.

[Rudovic and Pantic, 2010] Rudovic, O. Patras, I. and Pantic, M. (2010). Coupled Gaussian Process Regression for Pose-Invariant Facial Expression Recognition. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 350–363.

[Ruser and Puente-Leon, 2006] Ruser, H. and Puente-Leon, F. (2006). Methoden der Informationsfusion: Taxonomie und Überblick. *Informationsfusion in der Mess- und Sensortechnik*, pages 1–20.

[Russell and Fernandez-Dols, 1997] Russell, J. and Fernandez-Dols, J. (1997). *The Psychology of Facial Expression*. Cambridge Univ. Press.

[Russell, 1980] Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.

[Rühle, 2009] Rühle, J. (2009). Active Appearance Modelle mit Berücksichtigung von Verdeckungen. Technische Universität Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik, Bachelorarbeit. Betreuer: Christian Martin, Ronny Stricker, Betreuender Hochschullehrer: Horst-Michael Groß.

[Saatci and Town, 2006] Saatci, Y. and Town, C. (2006). Cascaded Classification of Gender and Facial Expression using Active Appearance Models. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2006)*, pages 393–400, Washington, DC, USA. IEEE Computer Society.

[Salvini et al., 2010] Salvini, P., Laschi, C., and Dario, P. (2010). Design for Acceptability: Improving Robots' Coexistence in Human Society. *Int. Journal of Social Robotics*, 2(4):451–460.

[Schenk et al., 2011] Schenk, K., Eisenbach, M., Kolarow, A., and Gross, H.-M. (2011). Comparison of Laser-based Person Tracking at Feet and Upper-Body Height. In *34th German Conference on Artificial Intelligence (KI-2011)*, pages 277–288.

[Scherer, 2005] Scherer, K. R. (2005). What are emotions? And how can they be measured? *Social Science Information*, 44(4):695–729.

[Schmidt et al., 2006] Schmidt, J., Fritsch, J., , and Kwolek, B. (2006). Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images. In *Proc. of 7th IEEE Int. Conf. on Automatic Face and Gesture Recognition*.

[Schmidt-Atzert, 1996] Schmidt-Atzert, L. (1996). *Lehrbuch der Emotionspsychologie*. Kohlhammer, Stuttgart, Berlin, Köln.

[Sclaroff and Isidoro, 2003] Sclaroff, S. and Isidoro, J. (2003). Active Blobs: Region-Based, Deformable Appearance Models. *Computer Vision and Image Understanding*, 89(2-3):197–225.

[Shewchuk, 1996] Shewchuk, J. (1996). Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *1st Workshop on Applied Computational Geometry (WACG): Towards Geometric Engineering*, pages 203–222.

[Shotton et al., 2011] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-Time Human Pose Recognition in Parts from a Single Depth Image. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1297–1304.

[Steege, 2007] Steege, F. (2007). Weiterentwicklung eines Verfahrens zur robusten videobasierten Einweisung eines Roboters mittels Zeigegesten. Technische Universität Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik, Diplomarbeit. Betreuer: Christian Martin, Betreuender Hochschullehrer: Horst-Michael Groß.

[Steffens et al., 1998] Steffens, J., Elagin, E., and Neven, H. (1998). PersonSpotter - Fast and Robust System for Human Detection, Tracking and Recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 1998)*, pages 516–521.

[Stiefelhagen et al., 2002] Stiefelhagen, R., Yang, J., and Waibel, A. (2002). Modeling Focus of Attention for Meeting Indexing. *IEEE Transactions on Neural Networks*, 13(4):928–938.

[Stiene, 2005] Stiene, S. (2005). Konturbasierte Objekterkennung aus Tiefenbildern eines 3D-Laserscanners. Fachbereich Informatik, Universität Osnabrück. Master thesis.

[Stricker, 2008] Stricker, R. (2008). Robotertauglicher vision-basierter Schätzer für das Interaktionsinteresse im Mensch-Roboter-Dialog auf Basis von Active Appearance Modellen. Technische Universität Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik, Diplomarbeit. Betreuer: Christian Martin, Klaus Debes, Betreuender Hochschullehrer: Horst-Michael Groß.

[Stricker et al., 2010] Stricker, R., Hommel, S., Martin, C., and Gross, H.-M. (2010). Realtime User Attention and Emotion Estimation on a Mobile Robot. In *Proc. 55th Int. Scientific Colloquium, Ilmenau, Germany*, pages 629–634.

[Stricker et al., 2009] Stricker, R., Martin, C., and Gross, H.-M. (2009). Increasing the Robustness of 2D Active Appearance Models for Real-World Applications. In *Proc. 7th Int. Conf. on Computer Vision Systems (ICVS)*, pages 364–373.

[Syrdal et al., 2006] Syrdal, D. s., Dautenhahn, K., Woods, S., Walters, M., and Koay, K. (2006). 'Doing the right thing wrong' – Personality and tolerance to uncomfortable robot approaches. In *Proc. of 15th IEEE Int. Symposium on Robot and Human Interactive Communication (ROMAN 2006)*, pages 183–188.

[Taylor, 2000] Taylor, C. (2000). Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image. *Computer Vision and Image Understanding*, 80:677–684.

[Theobald et al., 2006] Theobald, B.-J., Matthews, I., and Baker, S. (2006). Evaluating error functions for robust active appearance models. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2006)*, pages 149–154.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probalistic Robotics*. The MIT Press, New York, Inc., Secaucus, NJ, USA.

[Trefflich, 2010] Trefflich, B. (2010). *Videogestützte Überwachung der Fahreraufmerksamkeit und Adaption von Fahrerassistenzsystemen*. PhD thesis, Technische Universität Ilmenau, Audi Ingolstadt.

[Treptow et al., 2005] Treptow, A., Cielniak, G., and Duckett, T. (2005). Comparing Measurement Models for Tracking People in Thermal Images on a Mobile Robot. In *Proc. of European Conference on Mobile Robots (ECMR)*, pages 146–151.

[Tsoligkas and Xu, 2007] Tsoligkas, N. and Xu, D. (2007). Hybrid Object Based Video Compression Scheme Using a Novel Content-Based Automatic Segmentation Algorithm. In *Proc. of IEEE Int. Conference on Communications (ICC)*, pages 2654–2659.

[van Kuilenburg et al., 2005] van Kuilenburg, H., Wiering, M., and den Uyl, M. (2005). A Model Based Method for Automatic Facial Expression Recognition. In *Proceedings of the European Conference on Machine Learning 2005*.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simples Features. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 511–520.

[Viola and Jones, 2002] Viola, P. and Jones, M. (2002). Robust Real-time Object Detection. *Internationl Journal of Computer Vision.*, 57(2):137–154.

[Voit et al., 2006] Voit, M., Nickel, K., and Stiefelhagen, R. (2006). Neural Network-based Head Pose Estimation and Multi-view Fusion. In *CLEAR Evaluation Workshop*.

[Voit and Stiefelhagen, 2008] Voit, M. and Stiefelhagen, R. (2008). Deducing the Visual Focus of Attention from Head Pose Estimation in Dynamic Multi-view Meeting Scenarios. In *ACM & IEEE Int. Conf. on Multimodal Interfaces*, pages 173–180.

[Wallhoff, 2006] Wallhoff, F. (2006). Facial Expressions and Emotion Database. Technische Universität München: `http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html`.

[Wang et al., 2004] Wang, H., Li, S., and Wang, Y. (2004). Face recognition under varying lighting conditions using self quotient image. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004)*, pages 819–824.

[Weiss et al., 2010] Weiss, A., Igelsböck, J., Wurhofer, D., and Tscheligi, M. (2010). Looking Forward to a Robotic Society. *Int. Journal of Social Robotics*, 3(2):111–123.

[Werner, 2007] Werner, U. (2007). Mimikerkennung in Echtzeit unter Realwelt-Bedingungen mittels Active Appearance Models. Technische Universität Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik, Diplomarbeit. Betreuer: Christian Martin, Betreuender Hochschullehrer: Horst-Michael Groß.

[Wilhelm, 2005] Wilhelm, T. (2005). *Methoden der vision-basierten Nutzerwahrnehmung für eine natürliche Interaktion mit mobilen Servicerobotern*. PhD thesis, Technische Universität Ilmenau.

[Wilhelm et al., 2003] Wilhelm, T., Böhme, H.-J., and Gross, H.-M. (2003). Looking Closer. In *Proc. European Conference on Mobile Robots (ECMR 2003)*, pages 65–70.

[Wilhelm et al., 2005] Wilhelm, T., Böhme, H.-J., and Gross, H.-M. (2005). Classification of Face Images for Gender, Age, Facial Expression, and Identity. In *Proc. of ICANN 2005*, pages 569–547.

[Wimmer, 2005] Wimmer, T. (2005). Objekte mittels Background Subtraction erkennen. `http://www.prip.tuwien.ac.at/Teaching/WS/ProSemSab/prosem05/wimmer.html%`, Proseminar Visual Surveillance, Technischen Universität Wien.

[Wu and Nevatia, 2005] Wu, B. and Nevatia, R. (2005). Detecion of Multiple, Partically Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In *Proc. of IEEE Int. Conference on Computer Vision (ICCV)*, pages 90–97.

[Wu and Toyama, 2000] Wu, Y. and Toyama, K. (2000). Wide-Range, Person- and Illumination-Insensitive Head Orientation Estimation. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000)*, pages 183–188.

[Xiao et al., 2002] Xiao, J., Kanade, T., and Cohn, J. (2002). Robust Full-Motion Recovery of Head by Dynamic Templates and Re-Registration Techniques. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2002)*, pages 163–169.

[Xiao et al., 2003] Xiao, J., Moriyama, T., Kanade, T., and Cohn, J. (2003). Robust Full-Motion Recovery of Head by Dynamic Templates and Re-registration Techniques. *International Journal of Imaging Systems and Technology*, 13:85–94.

[Yamazaki et al., 2007] Yamazaki, K., Kawashima, M., Kuno, Y., Akiya, N., Burdelski, M., Yamazaki, A., and Kuzuoka, H. (2007). Prior-to-request and request behaviours within elderly day care: Implications for developing service robots for use in multiparty settings. In *Proc. of the Tenth European Conference on Computer Supported Cooperative Work*, pages 61–78.

[Zell, 1994] Zell, A. (1994). *Simulation neuronaler Netze.* Addison-Wesley.

[Zhan et al., 2008] Zhan, S., Chang, H., Jiang, J.-g., and Li, H. (2008). Spinal Images Segmentation Based on Improved Active Appearance Models. In *Proc. of. The 2nd Int. Conf. on Bioinformatics and Biomedical Engineering (ICBBE)*, pages 2315–2318.

[Zhang et al., 1998] Zhang, Z., Lyons, M., Schuster, M., and Akamatsu, S. (1998). Comparison between Geometry-based and Gaborwavelets -based Facial Expression Recognition using Multi-Layer Perceptron. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR 1998)*, pages 454–459.

[Zhao et al., 2002] Zhao, L., Pingali, G., and Carlbom, I. (2002). Real-time head orientation estimation using neural networks. In *Proc. Int. Conf. on Image Processing*, pages 297–300.

[Zivkovic and van der Heijden, 2001] Zivkovic, Z. and van der Heijden, F. (2001). A Stabilized Adaptive Appearance Changes Model for 3D Head Tracking. In *Proc. of the Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Realtime Systems*, pages 175–181.

[Zou et al., 2007] Zou, X., Kittler, J., and Messer, K. (2007). Illumination Invariant Face Recognition: A Survey. In *IEEE Int. Conf. on Biometrics: Theory, Applications, and Systems*, pages 1–8.

[Üzümcü et al., 2003] Üzümcü, M., Frangi, A., Reiber, J. H. C., and Lelieveldt, B. P. F. (2003). Independent Component Analysis in Statistical Shape Models. In *SPIE Medical Imaging*, pages 375–383.