# VIDEOBASIERTE GESTENERKENNUNG IN EINER INTELLIGENTEN UMGEBUNG

Dissertation
zur Erlangung des Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

JAN RICHARZ

Dortmund

2011

Tag der mündlichen Prüfung: 14.12.2011

Dekanin: Prof. Dr. Gabriele Kern-Isberner

Gutachter: Prof. Dr.-Ing. Gernot A. Fink
Prof. Dr. Heinrich Müller

# DANKSAGUNG

Mein besonderer Dank gilt all den Menschen, die mich beruflich und privat während der Entstehung dieser Dissertation begleitet haben und ohne die diese Arbeit nicht möglich gewesen wäre. An erster Stelle danke ich meinem Betreuer, Prof. Dr.-Ing. Gernot A. Fink, der immer ein offenes Ohr für meine Fragen hatte. Sein Rat und seine Ideen waren von großem Wert. Genauso danke ich Prof. Dr. Heinrich Müller, der sich trotz eines immer prall gefüllten Terminkalenders nicht lange überreden ließ, das Zweitgutachten zu erstellen und nebenbei auch noch die Zeit für wertvolle Anregungen fand.

Weiterhin danke ich meinen aktuellen und ehemaligen Kollegen in der Arbeitsgruppe intelligente Systeme, Marius Hennecke, Christian Kleine-Cosack, Dr.-Ing. Thomas Plötz, Boris Schauerte, Dr.-Ing. Christian Thurau und Dr. Szilárd Vajda, für jahrelange gute Zusammenarbeit und ein überaus angenehmes kollegiales Arbeitsklima. Besonders hervorheben möchte ich Dr. Kai Lienemann, der seine Freizeit für das Probelesen opferte. Ebenfalls danken möchte ich den studentischen Mitarbeitern, insbesondere Andreas Niemöller und Felix Gabriel, die durch ihre Arbeit zum Gelingen dieser Dissertation beitrugen.

Bei einer Arbeit, die sich über mehrere Jahre erstreckt, gibt es immer auch Rückschläge und Misserfolge. In diesen Phasen des Zweifelns gaben meine Familie und meine Freunde mir den benötigten Rückhalt. An allererster Stelle waren dies meine Eltern Willi und Brigitte Richarz und meine Schwester Astrid. Ein großes Dankeschön geht auch an Iyad Deeb, Florian Evers und Sebastian Göbel. Danke, dass ihr immer an meinen Erfolg geglaubt habt.

# INHALTSVERZEICHNIS

# 1 EINLEITUNG

## 1.1 NOTWENDIGKEIT NEUER MENSCH-MASCHINE-INTERAKTIONSKONZEPTE

Vor nicht allzu langer Zeit waren Computer und elektronische Geräte teure Spezialausrüstung, die nur von hochqualifiziertem Personal bedient werden konnten. Mit der rasanten Entwicklung der Computertechnik in den letzten Jahrzehnten und der damit einhergehenden Leistungssteigerung sowie besseren Verfüg- und Bedienbarkeit hielt elektronische Rechentechnik Einzug in nahezu alle Bereiche des menschlichen Lebens. Computer sind mittlerweile weit mehr als Arbeitsmittel oder „Rechenknechte", sie sind ein unverzichtbarer Bestandteil der modernen Informations- und Mediengesellschaft. Diese Durchdringung führt dazu, dass Rechentechnik und elektronische Geräte im wahrsten Wortsinne allgegenwärtig geworden sind. Gleichzeitig wird die Vielfalt und Komplexität der Dienste, die sie bereitstellen, und – eng damit verbunden – auch die ihrer Bedienung immer größer. Gute Beispiele hierfür sind moderne Smart Phones, die längst nicht mehr nur Telefone sind, sondern die Funktionalitäten von Organizern, Multimedia-Unterhaltungs-Plattformen, Breitband-Kommunikationsgeräten, Kameras etc. vereinen. Die Vision einer vernetzten Welt, in der der Mensch durch elektronische Helfer jederzeit und überall Zugriff auf eine Vielzahl an Informationen und Mehrwertdiensten hat, ist zumindest teilweise bereits Realität.

Als direkte Folge ergeben sich bisher unbekannte Fragestellungen in Hinsicht auf Bedienung, Vernetzung, Privatsphäre und Sicherheit. Elektronische Geräte müssen heute so konzipiert sein, dass ihre Bedienung auch ungeschulte und mit Computertechnik nicht vertraute Benutzer nicht überfordert, bzw. von diesen in kurzer Zeit intuitiv erlernbar ist. Gleichzeitig muss vermieden werden, dass diese Benutzer zu „gläsernen Menschen" werden, d.h. unwissentlich private und sensitive Daten für andere Personen zugänglich machen, sei es durch böswilligen Zugriff Dritter oder eigene Unbedarftheit. Das Ziel muss also sein, Missbrauch auf der einen Seite zu vermeiden, auf der anderen Seite aber jedem potentiellen Nutzer Zugriff auf die gleichen rechnergestützten Dienste zu ermöglichen, unabhängig davon, ob der jeweilige Benutzer mit der Bedienung des Systems vertraut ist oder nicht. Für eine breite Akzeptanz ist es wichtig, dass die Dienste des Rechners als Mehrwert wahrgenommen werden, ohne aufdringlich, störend, gefährlich oder überfordernd zu sein. Diese Aufgabe wird

angesichts der Vielfalt an gebotenen Funktionalitäten, an unterschiedlichen Geräten und Nutzertypen, nicht einfacher.

Das Konzept der Allgegenwärtigkeit und jederzeitigen Verfügbarkeit von Rechentechnik wird gemeinhin als *Ubiquitous Computing* oder auch *Pervasive Computing* bezeichnet. Dahinter stecken technische Entwicklungen und Forschungsvorhaben, deren Ziel es ist, genau jener zunehmenden Vernetzung und Durchdringung der modernen Gesellschaft mit elektronischen Helfern Rechnung zu tragen bzw. sie voranzutreiben. Die dahinter steckende Vision geht aber weit über den heute erreichten Stand hinaus. Sie besagt, dass einerseits die Dienste, die ein Rechnersystem anbieten kann, jederzeit und überall in unserer Umgebung verfügbar sein sollen, andererseits die Rechner selbst als solche nicht zu erkennen sind, weil sie harmonisch und weitgehend unsichtbar in die Umgebung integriert sind (*Disappearing Computer*). Der Computer ist somit nicht länger ein Werkzeug, das Eingeweihte benutzen und das von anderen Personen ggf. sogar als störend wahrgenommen wird, sondern er tritt in den Hintergrund als ein unauffälliger, allgegenwärtiger Assistent. Verfolgt man diese Idee konsequent weiter, führt das zum Konzept der Umgebungsintelligenz (*Ambient Intelligence*). Sie stellt den Schritt von einer Vielzahl verschiedener, an unterschiedliche Benutzer gebundener und auf höherer Ebene vernetzter, persönlicher elektronischer Apparaturen zu einer komplett in die Umgebung integrierten Rechnerintelligenz dar. Oder anders gesagt: Der Benutzer führt die ihn umgebende rechnerbasierte Intelligenz nicht mit sich und definiert sie somit für sich persönlich durch die Geräte, die er besitzt. Vielmehr ist die Umgebung selbst intelligent und bietet ihre Dienste und Funktionalitäten allen Benutzern gleichermaßen an. Im Idealfall ist hierfür nicht einmal eine gezielte Benutzungsintention des Benutzers vonnöten. Er muss sich noch nicht einmal der Tatsache bewusst sein, dass die Umgebungsintelligenz existiert, weil sie laufend den Zustand von Entitäten in ihrem Einflussbereich analysiert und automatisch unterstützend reagiert. Trotzdem muss natürlich ein Benutzer weiterhin die Möglichkeit haben, aktiv auf diese Reaktionen einzuwirken bzw. sie gezielt zu steuern. Das Ziel muss bleiben, dass der Mensch seine Umgebung kontrolliert, und nicht umgekehrt.

All dies macht die Entwicklung neuer Konzepte für Mensch-Maschine-Interaktionsschnittstellen nötig. Lange wurden derartige Schnittstellen durch mechanische Bedienelemente (Schalter und Tasten) realisiert. Diese erweisen sich aber angesichts der Flut unterschiedlicher in einem Gerät vereinter Funktionen als zunehmend ungeeignet. Ein Beipiel hierfür sind moderne Infotainment-Systeme in Fahrzeugen, deren Funktionsumfang sich längst nicht mehr auf Schalterbatterien abbilden lässt, sondern in komplexen mehrschichtigen Menüstrukturen repräsentiert werden muss. Deshalb dominiert mittlerweile das WIMP-Paradigma (Windows, Icons, Menus, Pointers), d.h. symbolbasierte grafische Benutzeroberflächen, die mittels speziell zugeschnittener

Eingabegeräte bedient werden. Diese stellen zwar eine wesentliche Verbesserung der Bedienbarkeit dar, geraten aber mittlerweile auch an ihre Grenzen. Das zeigt z.B. die zunehmende Verbreitung von Touchscreens in Smart Phones, weil ihre Bedienung über Tastatur oder Trackball zu kompliziert und unintuitiv wäre.

Die Vision der Umgebungsintelligenz und des *Disappearing Computers* schließt auch das Verschwinden solcher expliziter Eingabehardware ein. Fest installierte Bedien- oder Anzeigegeräte widersprechen der Forderung nach der Allgegenwärtigkeit, insbesondere der Möglichkeit, jederzeit und von jeder beliebigen Stelle der Umgebung Zugriff auf die Dienste des Rechners zu haben. Weiterhin werden die Zugriffsmöglichkeiten der Benutzer stark eingeschränkt, da immer nur ein oder wenige Benutzer gleichzeitig Zugriff auf die Bedienelemente haben können. Neuere Entwicklungen zielen daher immer mehr darauf ab, die Kopplung der Bedienelemente an physische Entitäten abzuschwächen und beispielsweise als „virtuelle Touchscreens“ auf beliebige Oberflächen zu projizieren. Ein aktuelles Beispiel hierfür ist das Microsoft-Projekt LightSpace [209]. Die Notwendigkeit einer grafischen Oberfläche – auch wenn sie an beliebige Stellen projiziert werden kann – impliziert aber immer noch, dass diese Oberfläche irgendwo in der Umgebung angezeigt werden muss. Dies kann auf unbeteiligte Personen störend wirken, schränkt die Designmöglichkeiten der Umgebung und des Systems ein oder ist in bestimmten Fällen – beispielsweise bei privaten Daten – aus Sicherheitsgründen nicht praktikabel.

Der größte grundlegende Nachteil WIMP-basierter Interfaces und dedizierter Bedienelemente ist aber, dass zur fehlerfreien Bedienung eine Unterweisung oder Eingewöhnung nötig ist, denn sowohl die Bedienung mit Tastatur und Maus als auch per ikonischer Oberflächen ist kein natürliches menschliches Kommunikationsmittel. Somit ist die Bedienung für viele Menschen nicht intuitiv, obgleich geschicktes Design der Bedienelemente dieses Problem abmildern kann. Demzufolge ist es wünschenswert, Interaktionsmöglichkeiten zu entwickeln, die vom Benutzer als natürlich wahrgenommen werden und intuitiv erlernbar sind.

## 1.2 INTELLIGENTE UMGEBUNGEN

Die ersten Realisierungsversuche einer Umgebungsintelligenz, welche die Menschen in ihren Handlungen unterstützt und jederzeit bestimmte Dienste anbietet, sind sog. *Intelligente Umgebungen* (auch: *Intelligente Häuser/Räume*, eng. *Smart/Intelligent Environments*). Die interessante Frage ist, was in diesem Zusammenhang das Wort „intelligent“ bedeutet.

Es ist ein verbreiteter Irrtum, die Intelligenz des Systems über die pure Masse an Elektronik und Funktionalitäten zu definieren, welche die Umgebung bereitstellt. Sicher führt dies mitunter zu sinnvollen Anwendungen, und die Vernetzung und Synchronisation einer Vielzahl unterschiedlicher Sensoren, Aktoren, Netzwerke und Rechenelemente stellt schon für sich allein eine große Herausforderung und ein interessantes Forschungsfeld dar. Aber ein derartig hochkomplexes System ist nicht notwendigerweise intelligent, es ist in erster Linie komplex. Das impliziert auch Fehleranfälligkeit und insbesondere Komplexität der Bedienung. Als Beispiel sei ein einfacher Lichtschalter genannt. Solange es nur wenige Lampen gibt, die per Schalter bedient werden müssen, funktioniert dieses Konzept. Stellt die Umgebung mehrere Lampen zur Verfügung, die unabhängig voneinander gedimmt oder zu bestimmten Lichtgruppen und Beleuchtungsszenarien zusammengeschaltet werden können, wird die Bedienung über Schalter verwirrend und fordert die Intelligenz des Benutzers, nicht des Systems. Das Bedienkonzept des Lichtschalters, das an sich sehr einfach und intuitiv verständlich ist, funktioniert aufgrund der gewachsenen Umgebungskomplexität nicht mehr. Ein wirklich intelligentes System sollte weiterhin mit nur wenigen Schaltern auskommen und das jeweilige Beleuchtungsszenario durch Beobachtung des Nutzers oder anhand von Kontextwissen auswählen, bzw. der Nutzer sollte in der Lage sein, das gewünschte Szenario auf intuitive Weise zu definieren.

Was macht ein intelligentes System also „intelligent“? Wir Menschen nehmen einen Interaktionspartner dann als intelligent wahr, wenn wir mit ihm in gewohnter Weise kommunizieren können und er sinnvolle Reaktionen auf unsere Aktionen und den gegenwärtigen Zustand der Umgebung zeigt. Intelligenz definiert sich also einerseits durch ein Situationsbewusstsein (engl. *Situation Awareness*) und entsprechend an den jeweiligen Kontext angepasstes Verhalten, andererseits durch hochentwickelte Interaktionsfähigkeiten. Übertragen auf den Kontext der Intelligenten Umgebungen bzw. der Intelligenten Systeme bedeutet das: Die Intelligenz eines Systems definiert sich nicht nur durch die Menge oder Komplexität der Funktionalitäten, die es zur Verfügung stellt. Sie definiert sich auch und in erster Linie durch die Möglichkeit des intuitiven Zugriffs auf diese Funktionalitäten durch die Nutzer sowie durch ein gewisses Kontextbewusstsein, so dass das System bei unterschiedlichen Umgebungszuständen auf die gleiche Nutzeraktion ggf. unterschiedlich, aber stets in sinnvoller Weise reagiert. Ein mobiler Roboter, der einen festen Pfad abfährt und bestimmte Funktionalitäten per Touchpad zur Verfügung stellt, wird zwar als interessantes technisches System, nicht jedoch als intelligent wahrgenommen. Ist derselbe Roboter in der Lage, dynamisch Hindernissen auszuweichen, sich Personen zuzuwenden oder auf Sprachkommandos und Gesten zu reagieren, so wirkt er intelligent.

Eine Intelligente Umgebung muss also drei grundlegende Voraussetzungen erfüllen:

- Bereitstellung sinnvoller, den Nutzer unterstützende Dienste, automatisch oder durch explizite Aktivierung.
- Wahrnehmung und Modellierung des Zustandes der Umgebung und anwesender Personen, also des Kontextes, um Systemreaktionen darauf abzustimmen.
- Intuitive Bedienbarkeit, d.h. Zugriff auf Systemfunktionen und -dienste mit Hilfe natürlicher menschlicher Interaktionsmodalitäten.

Alle diese Voraussetzungen führen zu Problemstellungen, deren Lösung nicht trivial ist. Während der erste Punkt eher dem Bereich Software- und Systemdesign zuzuordnen ist, führen die anderen beiden zu Aufgabenstellungen aus den Bereichen maschinelles Lernen, automatische Mustererkennung und *Computer Vision*. Diese Arbeit beschäftigt sich vorwiegend mit der Realisierung der intuitiven Bedienbarkeit. Die Wichtigkeit intuitiver Eingabe- bzw. Bedienmethoden für moderne technische Systeme wurde weiter oben bereits erläutert. Was aber sind solche intuitiven Eingabemethoden? Intuitiv bedeutet in diesem Zusammenhang, dass die Bedienung einerseits leicht und schnell erlernbar ist, andererseits die Bedeutung der Eingabekommandos für den Benutzer leicht ersichtlich ist. Dafür eignen sich offensichtlich insbesondere Modalitäten, die auch in der täglichen natürlichen zwischenmenschlichen Interaktion vorkommen. Hier sind in erster Linie Sprache, Gestik sowie Mimik zu nennen. Während für die Erkennung natürlicher Sprache bereits seit längerem gute kommerzielle Lösungen vefügbar sind, ist die Analyse von Gestik in uneingeschränkten Szenarien noch ein ungelöstes Problem und ein aktives Forschungsfeld. In dieser Arbeit wird deshalb die Entwicklung einer gestenbasierten Mensch-Maschine-Schnittstelle angestrebt.

## 1.3 MOTIVATION UND ZIELSETZUNG DIESER ARBEIT

Die vorliegende Arbeit hat die Entwicklung von Methoden zur Nutzerlokalisation und zur Erkennung bestimmter Gesten zum Zweck der intuitiven Mensch-Maschine-Interaktion (MMI) zum Ziel. Als Ziel- und Integrationsumgebung dient ein intelligenter Konferenzraum (FINCA, vgl. Kapitel 6.2), der über Sensorik – in erster Linie Kameras und Mikrofone – verfügt und dessen Aufgabe es ist, seine Benutzer in typischen Konferenz- und Vortragssituationen bestmöglich zu unterstützen. In diesem Szenario ist die Interpretation von Gesten aus verschiedenen Gründen interessant.

Intelligente Konferenzräume bieten eine Vielzahl von technischen Geräten und entsprechend vielfältige Konfigurationsmöglichkeiten. Beispielsweise könnte die Beleuchtung in verschiedene Gruppen aufgeteilt sein, die unabhängig voneinander gesteuert werden können, ein Videoprojektor oder Display kann angesteuert und

konfiguriert werden, aktive Kameras können je nach Bedarf auf unterschiedliche Bereiche des Raumes ausgerichtet werden. Die Gesamtheit der Konfigurationen der technischen Ausstattung definiert, zusammen mit dem Umgebungskontext und dem Verhalten der Benutzer, den Zustandsraum einer intelligenten Umgebung. Wie bereits erwähnt, muss es das langfristige Ziel sein, dass diejenigen Parameter des Zustandes, welche kontrollierbar sind – eben die Konfiguration der technischen Ausstattung – kontextbezogen automatisch in sinnvoller Weise gewählt werden. So erfordert etwa eine Vortragssituation häufig, dass die Beleuchtung abgedunkelt und die Kameras auf den Präsentationsbereich ausgerichtet werden, während es bei einer Konferenz hell sein sollte und die Kameras z.B. den aktuellen Sprecher fokussieren könnten. Da es aber Abweichungen von „typischen“ Situationen geben kann, die Vorlieben der Benutzer mitunter sehr unterschiedlich sind und auch mit Fehlern bei der Situations- und Kontextanalyse zu rechnen ist, muss zusätzlich den Benutzern die Möglichkeit gegeben werden, die Konfiguration der intelligenten Umgebung jederzeit zu korrigieren und ihren Bedürfnissen anzupassen. In diesem Zusammenhang kann Gestik ein mächtiges Werkzeug sein, um die Bedienung einerseits effizient, andererseits leicht erlernbar und verständlich zu machen:

- Gesten sind effizienter und weniger störend als eine Bedienung mittels Schalterfeldern oder ähnlichen fest installierten Bedienelementen. Der Nutzer muss sich nicht von seinem Platz entfernen, um die Umgebungskonfiguration zu verändern, sondern kann die Änderungen idealerweise von jedem beliebigen Punkt per Geste vornehmen. Das spart Zeit und ist z.B. in Vortragssituationen oder Diskussionsrunden von Vorteil, weil der Diskussionsfluss nicht unterbrochen wird. Insbesondere können somit fehlerhafte oder unerwünschte Aktionen der intelligenten Umgebung sofort unterbunden oder korrigiert werden.

- Eine Bedienung mittels Gesten (und ggf. Sprache) ist natürlich und intuitiv zu erlernen, sofern die Kommandogesten natürlichen in zwischenmenschlicher Kommunikation vorkommenden Gesten entsprechen (vgl. Kapitel 3). Dieser Vorteil ist umso stärker ausgeprägt, je vielfältiger und komplexer die Bedienmöglichkeiten und Dienste der intelligenten Umgebung sind. Ab einem bestimmten Punkt werden selbst sorgfältig gestaltete Bedienelemente und -oberflächen unübersichtlich und verwirrend, erfordern somit eine längere Eingewöhnungszeit oder sogar eine Schulung des Benutzers. Dieses Problem kann durch die Verwendung von Gestik abgemildert werden.

- Bestimmte Gesten eröffnen Interaktionsmöglichkeiten, die mit herkömmlichen Bedienelementen oder auch mit reiner Sprachsteuerung nicht realisierbar wären.

Hier sind insbesondere Zeigegesten zu nennen, die in intuitive Weise eindeutige Referenzen zu Personen, Objekten oder Orten in der Umgebung definieren.

- Gesten oder Aktionen können Aufschluss geben über den aktuellen Kontext bzw. die Intentionen der Benutzer. Die passive Beobachtung und Erkennung von Gestik kann somit ein Hilfsmittel für eine Situationserkennung sein. Wie oben bereits erwähnt, ist ein Situationsbewusstsein eine wichtige Voraussetzung für intelligente Systeme.

Aus diesen Punkten wird ersichtlich, dass eine gestenbasierte Schnittstelle einen wichtigen Schritt hin zu intuitiver, leicht erlernbarer und jederzeit verfügbarer Mensch-Maschine-Interaktion innerhalb einer intelligenten Umgebung darstellt. Ein besonderer Schwerpunkt liegt dabei auf der Erkennung und Auswertung von Zeigegesten sowie der expliziten Berechnung der Zeigerichtung. Damit wird eine Verfolgung der angezeigten Richtung durch den Raum möglich, was das Deuten auf weit entfernte Bereiche sowie auf Objekte außerhalb des aktuellen Kamerasichtfeldes ermöglicht. Diese Information kann in verschiedenen Szenarien genutzt werden. Beispielsweise kann die Sensorik des Raumes aktiv durch den Benutzer auf einen angezeigten Bereich gelenkt werden, oder ein mehrdeutiger Steuerbefehl, der sich auf mehrere Entitäten beziehen kann, wird konkretisiert („Schalte *dieses* Licht an."). Weiterhin sind Zeigegesten in Hinblick auf das angestrebte Situationsbewusstsein interessant, da die intelligente Umgebung hierfür Kenntnisse über den Zustand der Szene, insbesondere über anwesende Objekte und Personen, benötigt. Durch den Einsatz von Zeigegesten können neu hinzugekommene Objekte dem System explizit und ohne künstliche Kalibrationsphase „präsentiert" werden. Weiterhin können Zeigegesten in Verbindung mit Spracherkennung dazu dienen, Objekten bestimmte Attribute (wie z.B. Namen, Farbe) zuzuordnen, die hinterher zur Identifikation genutzt werden können.

Neben Zeigegesten sind bestimmte bedeutungstragende Gesten (sog. *Embleme*, vgl. Kapitel 3) von Interesse, die direkt als Steuerkommandos interpretiert werden können, etwa Winken um die Aufmerksamkeit auf sich zu ziehen oder „nach oben" bzw. „nach unten" zur Steuerung der Jalousien und der Beleuchtungshelligkeit. Hierbei stellt die Auswahl geeigneter Gesten ein nicht triviales Problem dar, weil diese einerseits unterschiedlich und strukturiert genug sein müssen, um zuverlässig erkannt werden zu können. Andererseits dürfen sie nicht zu abstrakt oder stilisiert sein, da sie sonst für den Benutzer nicht mehr intuitiv sind.

Die Gestenerkennung soll nutzerunabhängig sein, d.h. kein vorheriges Training oder sonstige Anpassungen für unterschiedliche Nutzer benötigen. Sie soll in 3D erfolgen, da nur so eindeutige Referenzen innerhalb eines Raumes ohne weitere Einschränkungen möglich sind und die Erkennung invariant gegenüber der relativen Positionen

von Nutzer und Kameras realisiert werden kann. Weiterhin sollen die entwickelten Methoden möglichst flexibel einsetzbar sein, z.B. soll die Art der Kamerainstallation so wenig wie möglich eingeschränkt werden. Daher kommt keine spezialisierte Sensorik, wie Stereokameras, Infrarotkameras oder synchronisierte Multikamerasysteme, zum Einsatz. Außerdem ist bei der Auswahl der eingesetzten Methoden auf eine effiziente algorithmische Umsetzbarkeit zu achten, so dass ein reaktives Demonstrationssystem im Umfeld des intelligenten Konferenzraumes FINCA realisiert werden kann. Reaktivität ist insbesondere für die Akzeptanz eines technischen Systems notwendig, da es einem Benutzer nicht zuzumuten ist, dass er längere Zeit (länger als wenige Sekunden) auf eine Reaktion des Systems warten muss.

Zusammenfassend leistet diese Arbeit Beiträge in folgenden Bereichen:

- Robuste Detektion von Personen und deren Händen in einem realistischen Innenraumszenario.
- Punktbasierte 3D-Rekonstruktion in einem verteilten *unsynchronisierten* Multikamerasystem mit beliebiger Kameraanordnung.
- Reaktive trajektorienbasierte Gestenerkennung in 3D in einem weitgehend uneingeschränkten Interaktionsszenario, d.h. insbesondere unabhängig von Position und Orientierung des Benutzers.

## 1.4 AUFBAU DIESER ARBEIT

Nachdem die Fragestellungen, die in dieser Arbeit untersucht werden sollen, motiviert wurden, werden in den folgenden Kapiteln zunächst einige zum Verständnis nötige theoretische Grundlagen behandelt. Begonnen wird mit einer kurzen Einführung in grundlegende Begriffe und Methoden der Mustererkennung und *Computer Vision* zum Zweck der Definition einer einheitlichen begrifflichen Basis. Es folgt eine Vorstellung verbreiteter Kategorisierungen von Gestik und menschlicher Körpersprache, anhand derer definiert wird, was im Rahmen dieser Arbeit als relevante Geste verstanden werden soll. Kapitel 4 verwendet die zuvor eingeführten Begriffe und Kategorien, um einen ausführlichen und systematischen Überblick über verwandte Arbeiten zu geben. Im darauf folgenden Kapitel 5 werden Methoden, die im Rahmen dieser Arbeit zum Einsatz kommen, detaillierter erläutert. Darauf aufbauend stellt Kapitel 6 die konkrete Realisierung der einzelnen Verarbeitungsschritte vor, welche in Kapitel 7 ausführlich evaluiert werden. Die Arbeit schließt mit einer Zusammenfassung und einem Fazit.

# 2 GRUNDBEGRIFFE UND GRUNDLEGENDE METHODEN

Im Folgenden werden zum Zwecke einer einheitlichen begrifflichen Grundlage einige Begriffe aus dem Umfeld der Mustererkennung und *Computer Vision* eingeführt und kurz erläutert, die für diese Arbeit von Bedeutung sind. Mustererkennung beschäftigt sich mit der Analyse von Daten, welche die physikalische Welt abbilden. In diesem Sinne ist ein *Muster* die Gesamtheit der durch die Sensorik erfassten Messwerte zu einem bestimmten Zeitpunkt, die einen Zustand der physikalischen Umgebung beschreibt [133]. Ziel der Mustererkennung ist es, Perzeptionsfähigkeiten des Menschen in einem maschinellen Umfeld nachzubilden.

Menschen erbringen scheinbar mühelos erstaunliche Leistungen. Als Beispiel sei die Erkennung von Gesichtern genannt: Wir Menschen können Gesichter unterschiedlichster Ausprägung erkennen, unabhängig von Hautfarbe, Alter, Geschlecht, Beleuchtung, Frisur, Kopfstellung und Gesichtsausdruck. Weiterhin sind wir selbst dann in der Lage, Gesichter als solche wahrzunehmen, wenn große Teile davon verdeckt sind. Dies wird erreicht durch einerseits hochflexible, massiv parallele Verarbeitung perzeptioneller Stimuli im Gehirn, andererseits durch Rückgriff auf einen großen, über Jahre angelernten Erfahrungsschatz und Auswertung von Kontextwissen. Eine exakte Nachbildung dieser Fähigkeiten in einem technischen System ist nach heutigem Stand weder technisch noch methodisch möglich. Daher ist die Realisierung eines Mustererkennungssystems für Realweltdaten eine große Herausforderung.

*Computer Vision* oder *maschinelles Sehen* beschäftigt sich speziell mit der Verarbeitung visueller Muster, also von Kamerabildern, und demzufolge mit der maschinellen Nachbildung visueller Erkennungsleistungen des Menschen. Seit einigen Jahren ist dies eines der dynamischsten und größten Spezialgebiete innerhalb des Forschungsumfeldes der Mustererkennung und gewinnt auch im kommerziellen Umfeld immer größere Bedeutung. Aufgrund der speziellen Natur der verwendeten Daten – Bilder bieten einerseits einen im Vergleich zu anderen Sensordaten enorm hohen Informationsgehalt, andererseits ist auch eine entsprechend größere Datenmenge und eine große Anzahl von Variationen und Distraktoren in den Mustern zu bewältigen – wurden im Bereich des maschinellen Sehens viele spezialisierte Methoden und Ansätze entwickelt.

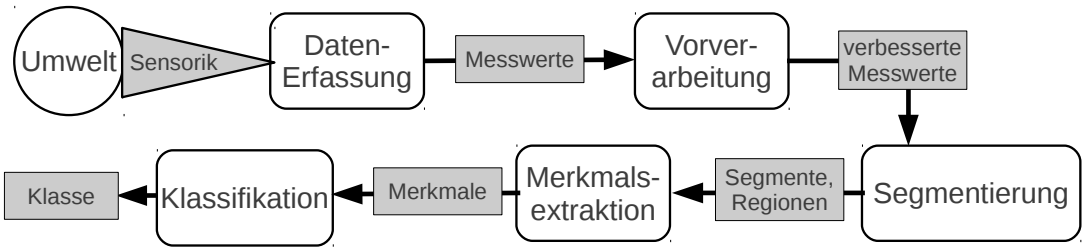


Abbildung 1: Aufbau eines typischen Mustererkennungssystems.

## 2.1 AUFBAU EINES MUSTERERKENNUNGSSYSTEMS

Ein typisches Mustererkennungssystem besteht aus den Teilen *Datenerfassung*, *Vorverarbeitung*, *Segmentierung*, *Merkmalsextraktion* und *Klassifikation* (Abbildung 1, vgl. z.B. [133]). Unter *Datenerfassung* versteht man die Abbildung der (analogen) realen Umwelt in von einem Rechnersystem verarbeitbare digitale Messwerte mittels entsprechender Sensorik. Diese Abbildung ist nicht immer eindeutig und i.d.R. aufgrund der Charakteristik der Sensoren (z.B. Sensorrauschen, Linsenverzerrung etc.) verfälscht oder gestört. Die *Vorverarbeitung* hat das Ziel, den Einfluss von Störungen in den aufgenommenen Messwerten zu eliminieren oder zu verringern. Allgemeiner gesagt, sollen ungewollte Variationen oder redundante Informationen in den Messwerten entfernt werden, welche für die Aufgabe nicht relevant sind, aber das Ergebnis der Datenanalyse beeinflussen könnten. Typische Vorverarbeitungsschritte umfassen Glättung, Rauschelimination, Normierung oder lineare Transformationen.

*Segmentierung* bedeutet die Aufteilung eines aufgenommenen Musters in zusammenhängende, nicht überlappende Teile. Die Zusammengehörigkeit der Elemente eines Segmentes ist durch geeignete Distanz- bzw. Ähnlichkeitsmaße oder Diskontinuitäten im Muster definiert. Ein Beispiel ist die Aufteilung eines Bildes in Regionen einheitlicher Farbe oder Textur.

Für die weiterführende Analyse eines Musters bzw. der nach der Segmentierung entstandenen Segmente ist meist eine *Merkmalsextraktion* nötig. Als *Merkmal* bezeichnet man eine Datenrepräsentation, welche bestimmte für die Mustererkennungsaufgabe wichtige Eigenschaften eines Musters beschreibt, also eine numerische Größe, die von den Mustern (d.h. den Sensormessungen) abstrahiert. Das Ziel hierbei ist zum Einen die bessere Analysierbarkeit durch gezielte Extraktion bestimmter Eigenschaften des Musters, zum Anderen eine Reduktion der Datenmenge durch die Berechnung einer kompakteren Repräsentation. Konkrete Ausprägungen eines Merkmals charakteri-

sieren einen für die Aufgabe des Mustererkennungssystems relevanten Zustand des beobachteten Musters, definieren also die *Klasse*, zu der das Muster gehört.

Die Zuordnung eines Musters (bzw. eines musterbeschreibenden Merkmals) zu einem Klassenkennzeichen aus einer Menge bekannter Klassen heißt *Klassifikation*. Ein *Klassifikator* ist eine Funktion, welche die Abbildung eines Merkmals auf sein zugehöriges Klassenkennzeichen vornimmt. Diese Funktion ist meist nicht analytisch bestimmbar, da nicht alle möglichen Ausprägungen aller Klassen bekannt sind und die Abbildungsfunktion ggf. nicht eindeutig ist. Daher wird sie typischerweise anhand einer sog. *Trainingsstichprobe* bestimmt. Die Stichprobe besteht aus einer Menge von Merkmalsvektoren und dem Kennzeichen der Klasse, zu der das jeweilige Merkmal gehört. Unter der Annahme, dass die Stichprobe repräsentativ für das betrachtete Problem ist, kann eine Approximation der Abbildungsfunktion mit Methoden des *maschinellen Lernens* ermittelt werden. Die Bezeichnung Klassifikator bezieht sich dann auf die so ermittelte approximative Funktion.

Ein klassisches Mustererkennungsproblem ist die Aufgabe, eine Ausprägung einer bestimmten Musterinstanz in einem Datenstrom zu finden. Als Beispiel sei die Lokalisierung von Personen in Kamerabildern genannt. Obwohl dies nichts anderes als ein binäres Klassifikationsproblem ist, wird dieser Vorgang gemeinhin als *Detektion* und ein entsprechendes Mustererkennungssystem als *Detektor* bezeichnet.

Für die Realisierung der Verarbeitungsphasen eines solchen Mustererkennungssystems kann auf eine große Menge von Methoden und Konzepten zurückgegriffen werden, welche die jeweilige Aufgabenstellung oder Teile davon lösen. Im Folgenden werden nun einige davon, die im Rahmen dieser Arbeit relevant sind, kurz erläutert.

## 2.2 KANTENBILDER

Als *Kantenbild* oder *Gradientenbild* $\mathbf{G}(x,y) = \{\mathbf{G}_x(x,y), \mathbf{G}_y(x,y)\}, \quad \mathbf{G}_x(x,y) = \{g_x(x,y)\}$ bezeichnet man die erste räumliche Ableitung eines Bildes $\mathbf{B}(x,y)$ entlang der primären Bildachsen (vgl. z.B. [64] Kapitel 10.2.5):

$$\mathbf{G}_x^c(x,y) = \frac{\delta}{\delta x}\mathbf{B}^c(x,y), \quad \mathbf{G}_y^c(x,y) = \frac{\delta}{\delta y}\mathbf{B}^c(x,y) \tag{2.1}$$

Hierbei bezeichnet $c = 1...n$ den Farbkanal. Besitzt $\mathbf{B}(x,y)$ $n$ Farbkanäle, so hat $\mathbf{G}(x,y)$ $2n$ Kanäle. Man erhält also je Kanal zwei getrennte Komponenten des Gradientenbildes für x- und y-Richtung. Gebräuchlicher und intuitiv verständlicher ist jedoch die Darstellung mit Betrag $\mathcal{M}$ und Orientierung $\phi$, welche sich wie folgt berechnen lässt:

$$\mathcal{M}^c(i,j) = \sqrt{g_x^c(i,j)^2 + g_y^c(i,j)^2}, \quad \phi^c(i,j) = \arctan\left(\frac{g_y^c(i,j)}{g_x^c(i,j)}\right) \tag{2.2}$$

Hierbei bezeichnet $\mathcal{M}^c(i,j)$ den Gradientenbetrag des Pixels $b^c(i,j)$ mit den Bildkoordinaten $i$ und $j$ für den Farbkanal $c$, die weiteren Bezeichnungen sind analog dazu. Üblicherweise wird das Ausgangsbild $\mathbf{B}(x,y)$ vor der Gradientenberechnung in ein Graustufenbild konvertiert, so dass pro Pixel nur 2 statt $2n$ Komponenten entstehen. Alternativ kann nur jeweils der maximale Betrag und die zugehörige Orientierung oder der Mittelwert über alle Farbkanäle betrachtet werden. Der Betrag wird groß, wenn die lokalen Farb- oder Grauwertunterschiede im Bild groß sind. Das ist im Allgemeinen bei Objekt- und Texturkanten der Fall. Werden die Vorzeichen von $g_x(i,j)$ und $g_y(i,j)$ zur Bestimmung des Quadranten beachtet, so kann die Orientierung Werte zwischen $-\pi$ und $\pi$ annehmen, wobei $\phi$ und $\phi - \pi$ die gleiche Kantenorientierung, aber unterschiedliche Übergänge hell-dunkel oder dunkel-hell beschreiben.

Die Berechnung eines Kantenbildes erfolgt üblicherweise mit Hilfe von *Kantenoperatoren*, die als diskrete *Faltungsmasken* realisiert sind. Eine Faltungsmaske ist eine Pixelmaske, welche dem zentralen Pixel einen Wert zuweist, der sich durch die gewichtete Summe aller anderen Pixel unter der Maske ergibt. Verbreitete Beispiele für Kantenoperatoren sind der *Prewitt-Operator* oder der *Sobel-Operator*. Letzterer integriert eine Gradienten- mit einer Gauss'schen Glättungsmaske und erreicht damit eine gewisse Robustheit gegenüber Bildrauschen. Ein etwas aufwändigerer Algorithmus ist der *Canny-Algorithmus*, der auf einem Sobel-Kantenbild beruht, auf die extrahierten Kantenpixel aber zusätzlich eine Nebenmaxima-Unterdrückung und einen Zusammenhangstest anwendet.

## 2.3 HISTOGRAMME

Histogramme sind eine sehr einfache und effiziente Art der Merkmalsrepräsentation (vgl. z.B. [133]). Verbreitet sind in der *Computer Vision* Farb- und Kantenhistogramme, die sich als sehr geeignet erwiesen haben, um Eigenschaften von Bildregionen in kompakter Weise zu repräsentieren. Dabei kann ein Histogramm sowohl selber ein Merkmal sein als auch ein Werkzeug, um verschiedenartige Merkmale oder Merkmale aus verschiedenen Zeitschritten auf einer höheren Ebene zusammenzufassen (gewissermaßen also ein Metamerkmal). Ein Histogramm $\mathbf{h}$ ist eine diskrete Repräsentation der Häufigkeitsverteilung von Messwerten. Gegeben sei eine Messgröße $x$ und eine Menge von $n$ Messwerten $\mathcal{X} = \{x_i, i = 1...n\}$. Weiterhin seien die Wertebereichsgrenzen $\gamma_1$ und $\gamma_2$ von $x$ bekannt, d.h. $x_i \in [\gamma_1, \gamma_2] \in \mathbb{R}$. Ein Histogramm teilt den Wertebereich in $m$ – üblicherweise, aber nicht notwendigerweise, gleich große – Zellen auf, sog. *Bins*, d.h. $\mathbf{h} = (h_j, j = 1...m)$. Durch die Aufteilung lässt sich eine Funktion $f_h(x)$ definieren,

welche die Messwerte den Bins zuordnet: $f_h : \mathbb{R} \to [1, m]$. Das Histogramm kann somit wie folgt berechnet werden:

$$\mathbf{h} = (h_j), \quad h_j = \sum_{i=1}^{n} \delta(f_h(x_i) - j). \tag{2.3}$$

Nach Betrachtung aller Messwerte in $\mathcal{X}$ repräsentieren die Histogrammbins die absoluten Häufigkeiten der Messwerte[1], die in den Wertebereich fallen, welchen das entsprechende Bin repräsentiert, d.h. $\sum_j h_j = n$. Wird das Histogramm normalisiert, so dass $\sum_j h_j = 1$, dann repräsentieren die Bins die relativen Häufigkeiten. Ist $x$ eine Zufallsvariable, so ist $\mathbf{h}$ eine diskrete Schätzung ihrer Wahrscheinlichkeitsverteilung $p(x)$, deren Genauigkeit durch die Bingröße bestimmt wird. Die Erweiterung auf mehrdimensionale Messgrößen ist einfach, es ergeben sich entsprechend mehrdimensionale Histogramme. Ein Histogramm eignet sich also, um einfache Statistiken einer Menge von Messwerten oder Merkmalen kompakt zu kodieren.

## 2.4 HAUPTKOMPONENTENANALYSE

Die Hauptkomponentenanalyse (HKA engl. *Principal Component Analysis*, PCA, auch als *diskrete Karhunen-Loeve-Transformation* bekannt, vgl. z.B. [43]) ist ein grundlegendes Verfahren der multivariaten Datenanalyse. Es handelt sich um eine varianzerhaltende lineare Transformation des Merkmalsraumes, die üblicherweise zur Verringerung der Dimensionalität der Merkmalsvektoren eingesetzt wird. Konkret soll eine Menge von $k$ $n$-dimensionalen Merkmalen (oder statistischen Variablen) näherungsweise durch Linearkombination von $l$ orthonormalen Basisvektoren, $l \ll n$ dargestellt werden. Die HKA ist also eine Projektion $\mathcal{P} : \mathbb{R}^n \to \mathbb{R}^l$ des Merkmalsraumes $\Gamma$ auf einen niedrigdimensionaleren Raum $\Gamma'$. Dabei soll der Informationsverlust durch die Dimensionsreduktion möglichst gering sein. Die Grundannahme der HKA ist, dass die in den Daten enthaltene Informationsmenge mit ihrer Varianz korreliert ist und diese nicht für alle Dimensionen von $\Gamma$ gleichmäßig verteilt ist[2]. Demzufolge ist der Informationsverlust am geringsten, wenn die Basisvektoren den Richtungen im Merkmalsraum entsprechen, entlang derer die Varianz der Merkmale am größten ist. Die Lösung dieses Problems sind die sog. *Hauptkomponenten* (HK), d.h. diejenigen Eigenvektoren der Kovarianzmatrix der mittelwertfreien Merkmalsvektoren, die zu den $l$ größten Eigenwerten korrespondieren. Für eine detaillierte Herleitung sei auf

1 Streng genommen ergibt sich die Häufigkeit durch das Integral über die Binbreite, dies kann im Falle von gleich großen Bins aber vernachlässigt werden.

2 Diese Annahme hat sich häufig als nützlich erwiesen, ist aber nicht notwendigerweise gültig.

[43] Kapitel 3.8.1 verwiesen. Ein bekanntes Beispiel für die direkte Anwendung der HKA zur Musterklassifikation sind die sog. *Eigenfaces* [198].

## 2.5 KEYPOINTS UND LOKALE DESKRIPTOREN

Der Begriff *Keypoint* oder *Interest Point* bezeichnet einen Bildpunkt[3] bzw. eine Bildregion um diesen herum, die gemäß eines mathematisch wohldefinierten Kriteriums für die gestellte Aufgabe besonders geeignet oder bedeutungsvoll ist (vgl. z.B. [124]). Oder anders gesagt, ein *Keypoint* hebt sich in einer bestimmten von der Anwendung definierten Weise von den ihn umgebenden Punkten ab. Die Idee ist, nicht das gesamte Bild zu verarbeiten, sondern lediglich eine vergleichsweise kleine Menge von Bildpunkten oder -regionen, die aber besonders informationstragend sind. Das hat verschiedene Vorteile. Die zu verarbeitende Datenmenge verringert sich ggf. beträchtlich und die Verarbeitung wird robuster gegenüber Störungen, weil nur Informationen verarbeitet werden, welche für die Anwendung relevant sind. Zudem ergibt sich auf natürliche Weise eine gewisse Robustheit gegenüber teilweiser Verdeckung, und kleinen Verformungen einer Bildregion, da *Keypoints* sich als kleine, lokal diskriminative Regionen interpretieren lassen, welche lokale Eigenschaften beschreiben. Sie definieren also gewissermaßen eine Zerlegung einer Bildregion in Teile, von denen einige fehlen können und deren gegenseitige Lage sich verändern kann. Anhand der vorhandenen *Keypoints* ist oft trotzdem eine Identifikation möglich.

Es existieren viele verschiedene Möglichkeiten zur Lokalisierung [124] und Repräsentation [123] von *Keypoints*. I.d.R. sind sie speziell entworfen, um Punkte oder Regionen mit genau definierten Eigenschaften zu finden. Die Merkmalsrepräsentation eines *Keypoints*, d.h. die Beschreibung der ihn umgebenden Bildregion, wird gemeinhin als (lokaler) *Deskriptor* bezeichnet.

## 2.6 CLUSTERING UND VEKTORQUANTISIERUNG

Unter *Clustering* versteht man die Gruppierung ähnlicher Ausprägungen eines Musters (vgl. [43], Kapitel 10.6), d.h. die Identifikation von Häufungsgebieten von Datenpunkten in einem Datenraum. Mögliche Anwendungsfälle hierfür finden sich z.B. in den Bereichen des *unüberwachten maschinellen Lernens*, etwa zur rein datengetriebenen Extraktion von Klassensymbolen, oder bei der automatischen Konstruktion hierarchischer Suchbäume. Werden die identifizierten Häufungsgebiete (engl. *Cluster*) durch einen

3 Außerhalb der *Computer Vision* wird dieser Begriff kaum benutzt. Das Konzept ist jedoch nicht nur auf Bilddaten beschränkt, z.B. existieren Erweiterungen auf sog. *Space-Time Keypoints* (vgl. Kapitel 4)

Repräsentanten ersetzt, spricht man von *Vektorquantisierung*. Derartige Repräsentanten können z.B. in exemplar-basierten Klassifikationsmethoden zum Einsatz kommen.

Soll die Verteilung der Datenpunkte über den Datenraum in einem parametrischen probabilistischen Modell abgebildet werden, so kann ein *Clustering* zur Initialisierung eingesetzt werden. Beispielsweise werden bei Gauss'schen Mischverteilungsmodellen ([57], Kapitel 4.4) die Erwartungswerte der einzelnen Komponenten mit den *Cluster*-Repräsentanten initialisiert und die initialen Kovarianzen anhand der Datenpunkte des jeweiligen *Clusters* geschätzt. Weit verbreitete *Clustering*-Methoden sind der *k-means* Algorithmus und der Algorithmus nach Lloyd (vgl. [57], Kapitel 4).

## 2.7 TRACKING

*Tracking* (deutsch: (Ver)folgen) bezeichnet die Verfolgung eines Musters über die Zeit (vgl. z.B. [213]). Die Grundannahme hinter *Tracking*-Ansätzen ist, dass ein Objekt weder seine Position noch seine Repräsentation im Merkmalsraum sprunghaft ändert, sondern dass die Übergänge über die Zeit kontinuierlich geschehen. Somit kann für kleine Zeitunterschiede angenommen werden, dass das gesuchte Objekt sich ungefähr an der letzten bekannten Position befindet und seine Repräsentation sich kaum geändert hat. Ein *Tracker* versucht nun, ausgehend von einer bekannten Objektrepräsentation und -position das gesuchte Objekt im nächsten Zeitschritt wiederzufinden.

Insbesondere die Annahme der konstanten Position ist für sich schnell bewegende Objekte offensichtlich problematisch. Daher benutzen gängige Trackingverfahren stattdessen häufig ein Bewegungsmodell (vorgegeben oder aus der bisher beobachteten Bewegung gelernt), um die nächste wahrscheinliche Position des Objektes vorherzusagen. Dies geschieht unter der Annahme, dass die Art der Bewegung sich nicht sprunghaft ändert. Im Bereich um die durch den *Tracker* „vorhergesagte" Position wird dann der am besten zur Objektrepräsentation passende Bereich lokalisiert und das Modell ggf. mit der neuen Beobachtung adaptiert. In gewisser Weise sind Detektion und *Tracking* sehr ähnliche Probleme, *Tracking* kann als adaptives Detektionsverfahren mit geschickter Suchraumeinschränkung angesehen werden.

Der entscheidende Unterschied besteht darin, dass *Tracker* nicht zu jedem Zeitpunkt isolierte Objektdetektionen liefern, sondern eine zeitliche Referenz zwischen verfolgten Objekten herstellen. Es wird also nicht ein beliebiges Objekt einer Objektklasse gesucht, sondern eine ganz bestimmte Objektinstanz wird über die Zeit verfolgt. Damit können insbesondere Raum-Zeit-Trajektorien auf integrierte Weise ermittelt werden.

## 2.8 HINTERGRUND-MODELLIERUNG

In Innenräumen oder urbanen Umgebungen ist es oft zulässig anzunehmen, dass die Umgebung sich über die Zeit nicht oder nur langsam ändert. Übertragen auf die *Computer Vision* bedeutet das, dass der Bildhintergrund über einen längeren Zeitraum weitgehend statisch bleibt. Diese Annahme wird häufig ausgenutzt, um die Datenmenge zu verringern und die Robustheit zu erhöhen, indem die Verarbeitung frühzeitig auf Vodergrundregionen eingeschränkt wird (vgl. z.B. [152]). Im Folgenden bezeichnet *Hintergrund* alle nichtrelevanten Bereiche der Szene, während *Vordergrund* die für die Anwendung interessanten Bereiche, im vorliegenden Fall der Mensch-Maschine-Interaktion insbesondere Personen, bezeichnet.

Einer Hintergrund-Modellierung[4] liegen zwei Annahmen zugrunde. Erstens: Der Hintergrund weist gewisse Eigenschaften auf, die sich im Laufe der Zeit nicht oder nur wenig ändern, so dass sie in einem Modell erfasst werden können. Zweitens: Vordergrund und Hintergrund unterscheiden sich in diesen Eigenschaften, so dass mit Hilfe des Modells eine Trennung vorgenommen werden kann. Beide Annahmen können im weiteren Verlauf zu Problemen führen.

Die Annahme unveränderlicher Eigenschaften ist offensichtlich verletzt, wenn sich bewegte oder veränderliche Objekte in der Szene befinden, die nach obiger Definition nicht zum Vordergrund gehören, oder wenn sich globale Eigenschaften der Szene ändern. Ein Beispiel hierfür sind Bäume in einer Außenszene, die sich im Wind bewegen, oder Beleuchtungsänderungen in einem Innenraum. Um damit umzugehen, muss das Modell entweder so gewählt werden, dass es robust gegen derartige Änderungen ist, oder es muss in der Lage sein, sich diesen Änderungen anzupassen.

Die Annahme der Unterscheidbarkeit ist dann verletzt, wenn Vorder- und Hintergrund unter dem Modell eine ähnliche Ausprägung haben. Beispielsweise kann in einem rein farbbasierten Modell eine Person dann nicht von einer Wand unterschieden werden, wenn ihre Kleidung eine ähnliche Farbe wie die Wand hat. Dieses Problem kann durch die Kombination unterschiedlicher Modalitäten in einem Modell abgeschwächt werden. In der Regel ist aber auch mit einem guten Modell keine perfekte Trennung möglich, es ist mit Fehlern in der ermittelten Vordergrundmaske zu rechnen.

4 Natürlich kann ebenso der Vordergrund modelliert werden. Die Vorgehensweise ist in beiden Fällen identisch.

### 2.8.1 *Offline- und Online-Modelle*

Prinzipiell besteht bei einer Modellierung immer die Wahl zwischen einem *offline* (bzw. *statischen*) oder *online* (bzw. *dynamischen* oder *adaptiven*) Modell. Ein offline Modell wird einmal anhand einer Stichprobe trainiert und während der Laufzeit des Systems ohne weitere Modifikation benutzt. Es ist leicht einzusehen, dass so ein Modell empfindlich auf Abweichungen von den Trainingsbedingungen reagieren kann. Es ist somit nur in Umgebungen zuverlässig einsetzbar, in denen kontrollierte und vorab bekannte Bedingungen herrschen, welche in dem Modell erfasst werden können.

Im Gegensatz dazu kommt ein reines online Modell ohne vorheriges Training aus. Es wird zur Laufzeit datengetrieben erstellt und über die Zeit laufend an veränderte Umgebungsbedingungen angepasst. Ein solches Modell ist wesentlich flexibler einsetzbar, jedoch besteht das Problem der Initialisierung und Validierung: Um ein Modell zur Laufzeit zu lernen, muss eine initiale Hypothese gegeben sein. Ist diese fehlerhaft bzw. ändern sich die Umgebungsbedingungen so stark, dass das bisher gelernte Modell versagt, wird das Modell mit fehlerhaften Daten erstellt und degeneriert.

Deshalb bietet es sich an, eine Kombination aus beiden Ansätzen zu verwenden: Ein statisches Modell wird zur initialen Segmentierung verwendet. Die erhaltene Segmentierung wird dann dazu benutzt, das Modell zu verbessern und über die Zeit zu aktualisieren. Zusätzlich kann mit einem unabhängigen Verfahren eine Validierung des Modells zur Laufzeit erfolgen, so dass eine Adaption mit fehlerhaften Daten vermieden bzw. ein degeneriertes Modell erkannt werden kann.

### 2.8.2 *Modellierungsarten*

Die einfachste Art der Hintergund-Modellierung ist die *Hintergrundsubtraktion*. Hierbei wird als Modell ein Referenzbild der Szene gespeichert und zu jedem Zeitschritt von der aktuellen Beobachtung subtrahiert. Durch eine *Schwellwertoperation* kann das Differenzbild in eine binäre Vordergrund/Hintergrund-Maske überführt werden.

Komplexere Methoden benutzen *probabilistische Modelle* der Verteilung geeigneter Merkmale. In diesem Fall repräsentiert das Modell eine Wahrscheinlichkeitsdichte $p(b_t = HG|m(b_t) \in \Gamma)$ über einem geeigneten *Merkmalsraum* $\Gamma \subseteq \mathbb{R}^n$. Die Wahrscheinlichkeit, dass ein Pixel $b_t$ zum Zeitpunkt $t$ zum Hintergrund gehört, ergibt sich dann zu $P(b_t|m(b_t)))$, mit der Merkmalsrepräsentation des Pixels $m(b_t)$. Für die konkrete Modellierung von $p(...)$ ist prinzipiell jedes parametrische probabilistische Modell denkbar, wie z.B. *Histogramme* [4] oder *Gauss'sche Mischverteilungsmodelle* (GMM) [189].

Andere, weniger verbreitete Ansätze benutzen Methoden des *maschinellen Lernens* zur pixelweisen Klassifikation [166, 193] oder Distanzmaße zwischen Deskriptoren lokaler Bildregionen [138, 216].

Als Merkmale haben sich insbesondere Farbrepräsentationen als geeignet erwiesen [4, 48, 166, 189, 193], aber auch Intensitätskanten und andere strukturelle Merkmale wurden erfolgreich eingesetzt [94, 138, 147, 168, 216].

## 2.9 PROJEKTIVE GEOMETRIE

Die Aufnahme einer Szene mit einer herkömmlichen Kamera bildet eine dreidimensionale Szene durch ein Linsensystem auf eine zweidimensionale *Bildebene* ab. Dieser Vorgang wird als *Projektion* bezeichnet. Er stellt eine Dimensionsreduzierung dar, die zu einem erheblichen Informationsverlust führt, da der Abstand der zu den Bildpunkten korrespondierenden Szenepunkte von der Bildebene aus den Bilddaten nicht mehr rekonstruiert werden kann. Es ist jedoch möglich, die verloren gegangene Tiefeninformation zu rekonstruieren, wenn die Szene von mehreren räumlich verteilten Kameras aufgezeichnet wird[5] (vgl. [73]). Dazu müssen die Abbildungseigenschaften der Kameras sowie ihre relativen Positionen im Raum bekannt sein. Die Abbildungseigenschaften und die Rekonstruktion eines Szenepunktes aus den Bildpunkten mehrerer Kameras lassen sich mit Hilfe der *projektiven Geometrie* beschreiben.

### 2.9.1 *Das Lochkamera-Modell*

Die Projektion einer dreidimensionalen Szene auf eine Bildebene lässt sich sehr einfach durch das sog. Lochkamera-Modell beschreiben. Es geht von einer idealen Lochblende aus, d.h. die Optik der Kamera besteht nur aus einem infinitesimal kleinen Loch, so dass keine Linsenverzerrungen oder sonstige Artefakte auftreten. In diesem Fall lässt sich die Projektion vollständig mit Hilfe des Strahlensatzes beschreiben. Abbildung 2 zeigt schematisch die Abbildung eines Szenenpunktes in einer Lochkamera.

Sei $\mathbf{p} = (p_x, p_y, p_z)^T$ ein dreidimensionaler Punkt in der Szene. Der von ihm ausgehende Sichtstrahl fällt durch die Lochblende – das sog. *projektive Zentrum* $\mathbf{z}_p$ der Kamera – auf die im Inneren der Kamera im Abstand $f$ von $\mathbf{z}_p$ liegende Bildebene $\mathcal{B}$ und erzeugt dort den Bildpunkt $\mathbf{p}' = (p'_x, p'_y, p'_z)^T$. Durch diese Zentralprojektion erscheint das Abbild der Szene um 180 Grad gedreht. Die Betrachtung wird daher

5 Bei statischen Szenen ist eine Rekonstruktion ebenfalls möglich, wenn eine einzelne bewegte Kamera die Szene mehrfach zu verschiedenen Zeitpunkten aufnimmt. Die Formulierung des Rekonstruktionsproblems ist dann völlig analog zum Multikamera-Fall.

einfacher, wenn die Projektion auf eine virtuelle Bildebene $\mathcal{B}'$ betrachtet wird, die im Abstand $f$ vor der Linsenöffnung liegt. $f$ wird als *fokale Länge* der Kamera bezeichnet.

Im Folgenden wird zunächst angenommen, dass Kamera- und Weltkoordinatensystem identisch sind und ihren Ursprung in $\mathbf{z}_p$ haben. Die $x$- und $y$- Achsen definieren die Bildebene $\mathcal{B}'$, die $z$-Achse sei identisch zur *optischen Achse* der Kamera. Weiterhin sei $f = 1$. Dann lässt sich die Projektion durch folgende lineare Abbildung beschreiben:

$$\begin{pmatrix} p'_x \\ p'_y \\ p'_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} \tag{2.4}$$

Diese extrem einfache Beziehung gilt nur für die oben getroffenen vereinfachenden Annahmen. Für eine Bildebene im Abstand $f$ von $\mathbf{z}_p$ hat der Bildpunkt die gleichen projektiven Koordinaten, seine kartesischen Koordinaten ergeben sich somit durch Skalierung mit $f$. Im Falle einer realen (digitalen) Kamera besteht zudem die Bildebene aus diskreten Bildelementen der Breite $w_x$ und Höhe $w_y$, d.h. die kontinuierlichen Koordinaten $\mathbf{p}'$ werden abgebildet auf diskrete Pixelwerte $\hat{\mathbf{p}}$. Weiterhin besteht eine Diskrepanz zwischen dem Kamerakoordinatensystem, dessen Ursprung in $\mathbf{z}_p$ liegt, und dem Bildkoordinatensystem, dessen Koordinatenursprung üblicherweise in der linken oberen Bildecke liegt. Die Gesamttransformation für eine beliebige Szenengeometrie lässt sich demnach schreiben als ([73], Kapitel 6):

$$\begin{aligned} \begin{pmatrix} \hat{p}_x \\ \hat{p}_y \\ \hat{p}_z \end{pmatrix} &= \begin{pmatrix} \frac{f}{w_x} & \rho & c_x \\ 0 & \frac{f}{w_y} & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{o}_3^T & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} \\ &= \mathbf{K}_k \mathbf{K}_p \mathbf{K}_e \mathbf{p} \\ &= \mathbf{K}\mathbf{p} \end{aligned} \tag{2.5}$$

Hierbei ist $\mathbf{o}_3$ der Nullvektor der Dimension drei und $\rho = \tan(\alpha)\frac{f}{w_y}$ ein Korrekturfaktor für nicht exakt rechtwinklig aufeinander stehende Bildachsen. Dieser ist für moderne Digitalkameras normalerweise vernachlässigbar. $c_x$ und $c_y$ sind die Bildkoordinaten von $\mathbf{z}_p$, dem sog. *Hauptpunkt*. Die Matrix $\mathbf{K}_k$ heißt *intrinsische Kalibrationsmatrix* der Kamera. $\mathbf{K}_e$ heißt *extrinsische Kalibrationsmatrix* und ist definiert durch eine affine Transformation mit der 3D-Rotationsmatrix $\mathbf{R}$ und dem Translationsvektor

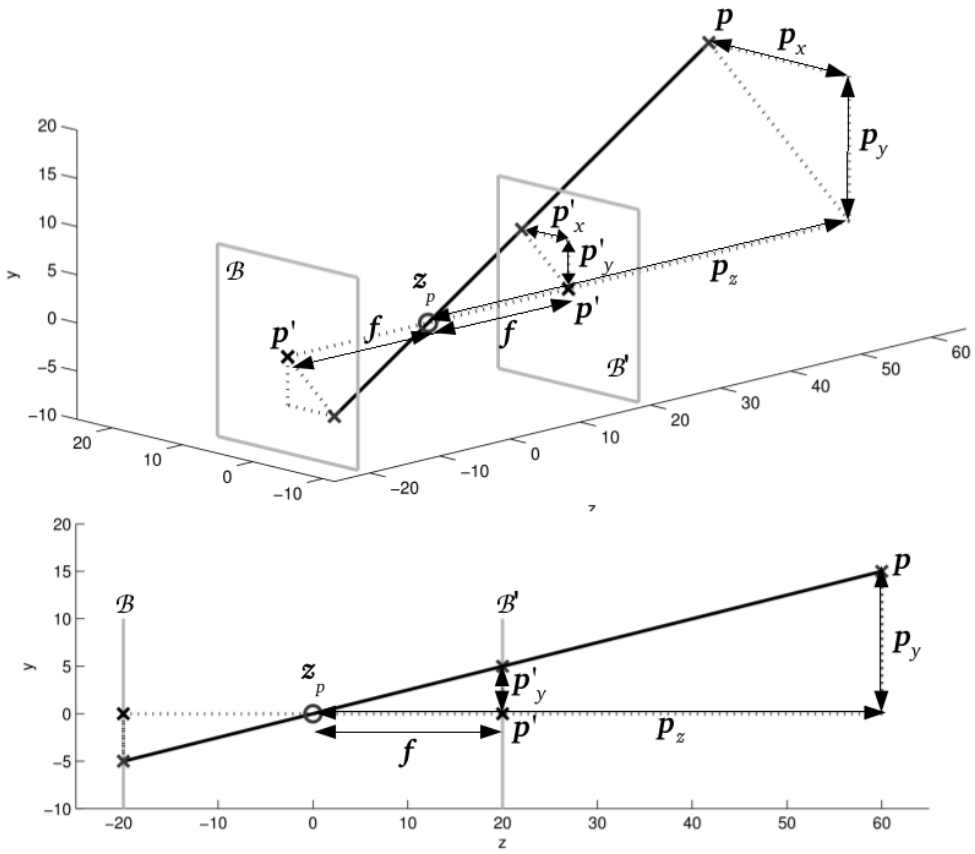


Abbildung 2: Grafische Darstellung der Beziehungen beim Lochkamera-Modell. Oben: 3D-Ansicht. Unten: Projektion auf $y-z$-Ebene.

**t**, welche das Kamerakoordinatensystem in das Weltkoordinatensystem überführt. Die 3x4 Matrix **K** heißt *Projektionsmatrix*. Damit lassen sich die Abbildungseigenschaften einer Kamera also vollständig durch $\mathbf{K}_k$ und $\mathbf{K}_e$ beschreiben. Die Bestimmung dieser Matrizen wird als *Kalibration* bezeichnet. Eine detaillierte Betrachtung des Kalibrationsproblems würde an dieser Stelle zu weit führen, deshalb sei für eine Einführung auf [51] verwiesen.

### 2.9.2 *3D-Rekonstruktion*

Sind die Abbildungseigenschaften der Kameras und ihre Positionen im Weltkoordinatensystem bekannt, kann der Bildpunkt jedes Szenepunktes in den Kamerabildern

analytisch bestimmt werden. Genauso ist es möglich, mittels der inversen Transformation die zweidimensionalen Pixelkoordinaten in dreidimensionale Weltkoordinaten zu überführen. Für eine einzelne Kamera lassen diese sich allerdings nicht eindeutig bestimmen. Es lässt sich lediglich die *projektive Linie* des Bildpunktes berechnen, d.h. seine Lage in der Szene ist nur bis auf einen Skalierungsfaktor bekannt.

Eine vollständige Rekonstruktion ist möglich, wenn mindestens zwei verschiedene Ansichten (d.h. mit unterschiedlichen extrinsischen und/oder intrinsischen Parametern) der Szene zur Verfügung stehen. Die projektiven Linien der Abbilder des selben Szenepunktes auf unterschiedlichen Bildebenen (sog. *korrespondierende Punkte*) schneiden sich an der realen Position des Szenepunktes.

# 3 GESTIK UND GESTENERKENNUNG

In Kapitel 1 wurde motiviert, warum Gestik eine wichtige Rolle bei der Realisierung intuitiv bedienbarer Mensch-Maschine-Schnittstellen einnehmen kann. Leider wird der Begriff "Geste" in der Literatur in sehr unterschiedlichen Kontexten verwendet, von Ganzkörperbewegungen über Gesichtsausdrücke bis hin zu Maus"gesten" zur Bedienung einer grafischen Oberfläche. Im Folgenden wird deshalb zur Verdeutlichung zunächst eine allgemeine theoretische Definition einer Geste vorgenommen, gefolgt von einer Aufschlüsselung unterschiedlicher Gestenarten, zum Einen aus der theoretischen Sicht der Semiotik, zum Anderen aus der systemorientierten Sicht der Mensch-Maschine-Interaktion. Im Sinne einer einheitlichen Grundlage zum Verständnis der Intention dieser Arbeit wird definiert, was im Rahmen dieser Dissertation unter einer Geste verstanden wird und welche Arten von Gesten von Interesse sind. Daran anschließend werden die technischen und methodischen Anforderungen an ein Gestenerkennungssystem herausgearbeitet.

## 3.1 DEFINITION: GESTIK

Unter Gestik versteht man ein nichtverbales, aber ggf. sprachbegleitendes, Mittel menschlicher Kommunikation unter Zuhilfenahme des Körpers bzw. einzelner Körperteile. Der allgemeine Begriff umfasst alle Arten der nichtverbalen körperbasierten Informationsübermittlung, im weitesten Sinne auch Mimik (d.h. Gesichtsausdruck) und unbewusste Körpersprache. Gesten können sowohl alleinstehend (d.h. die Geste trägt die komplette Information) oder aktionsbegleitend (d.h. als Mittel zur Verstärkung oder Verdeutlichung) auftreten, bei ihrer Interpretation ist also immer auch der Kontext zu beachten, in dem sie auftreten.

## 3.2 TAXONOMIE AUS SICHT DER SEMIOTIK

Aus Sicht der Semiotik ist Gestik „im engeren Sinne [...] das semiotische Ausdruckspotential des menschlichen Körpers mittels der Arme, der Hände und des Kopfes." ([139] S. 298). Hier wird die Definition also explizit auf Aktionen der Gliedmaßen

beschränkt, Ganzkörperposen und -bewegungen sind nach dieser Sichtweise keine Gesten, sehr wohl aber Kopfbewegungen und insbesondere auch Mimik.

Teilweise wird eine Unterteilung in Gesten und Gebärden vorgenommen, wobei Gebärden persönlichkeitsgebundener und weniger stark konventionialisiert sind als Gesten. Gesten werden in diesem Zusammenhang als „Ausdrucksbewegungen" bezeichnet, welche nur Bewegungen der Hände und Arme, nicht aber die Mimik umfassen ([139] S. 298). Der Übergang ist jedoch fließend und nicht klar begrenzt.

Eine bedeutende und verbreitete Gestenklassifikation stammt von Ekman und Friesen [47], die fünf Gestenkategorien definieren ([139] S. 299):

- **Embleme** haben „eine direkte sprachliche Übersetzung oder eine lexikalische Bedeutung" (ebd.). D.h. es sind konventionalisierte Zeichen, die autonom (auch ohne begleitende Sprache) bedeutsam sind. Dies kann beispielsweise das Heranwinken einer anderen Person sein, auch vordefinierte Kommandogesten und Zeichensprache-Zeichen fallen in diese Kategorie. Embleme bzw. ihre Bedeutung sind mitunter stark kulturabhängig.

- **Illustratoren** sind sprachbegleitende Gesten, die zur Untermalung oder Verdeutlichung der Sprache dienen. Ein anderer Begriff hierfür ist Gestikulation. Diese Gesten sind häufig unbewusst und persönlichkeitsabhängig, aber auch Zeigegesten (Deiktika) sind Illustratoren.

- **Regulatoren** sind ebenfalls „redebegleitende Gesten mit der Funktion, die verbale Interaktion zwischen Sprechern und Hörern zu steuern." (ebd.). Dazu gehören z.B. Nicken bzw. Kopfschütteln, um Zustimmung oder Ablehnung auszudrücken.

- **Affektäußerungen** sind Emotionsausdrücke, häufig mimisch und unbewusst.

- **Körpermanipulatoren** bezeichnet Gesten, die eine Interaktion mit einem Objekt oder dem eigenen Körper beinhalten, bspw. am Kopf kratzen.

Etwas vereinfacht lassen sich menschliche Gesten in drei große Gruppen einteilen (vgl. z.B. [46, 92]), die sich durch die Art der übermittelten Information, Ausdruckskraft und Intuitivität unterscheiden. Am einen Ende des Spektrums liegt die Gestikulation. Diese umfasst Körper- und Handbewegungen, die in hohem Maße mit anderen Ausdrucksmodalitäten, wie z.B. Sprache, korreliert sind und eine inhärente Multimodalität aufweisen [46]. Ihre automatische Interpretation ist deshalb schwierig, da einerseits zur vollen Erfassung ihrer Bedeutung die komplizierte und subtile Interaktion aller beteiligten Modalitäten analysiert werden müsste, andererseits die konkrete Ausprägung

von Gestikulation stark personalisiert ist und sie häufig unbewusst ausgeführt wird. Der Informationsgehalt und die Ausdruckskraft solcher Gesten ist, für sich alleine betrachtet, gering. Gestikulation stellt aber die natürlichste und intuitivste Art von gestenbasierter Informationsübermittlung dar.

Am anderen Ende des Spektrums befinden sich künstliche, wohldefinierte Zeichen- und Gebärdensprachen sowie gestenbasierte Kommandoalphabete. Sie zeichnen sich durch eine definierte Struktur und Semantik aus, ihre Interpretation ist daher eindeutig und ihr Informationsgehalt hoch. Allerdings haben derartig formalisierte Gesten wenig mit natürlichen menschlichen Gesten gemein. Sie sind daher wenig intuitiv und erfordern umfassendes Training.

Für natürliche gestenbasierte Mensch-Maschine-Interaktion ist die dritte Gestenklasse besonders interessant, die analog zu obiger Taxonomie als Embleme bezeichnet wird. Embleme sind wohldefinierte Zeichen, die für sich alleine betrachtet eine eindeutige Information übermitteln, dabei aber, typischerweise innerhalb einer bestimmten ethnischen oder kulturellen Gruppe, allgemein etabliert sind. Obwohl sie in gewisser Weise konventionalisierte Zeichen sind, ist ihre Ausführung und Interpretation daher intuitiv und natürlich.

## 3.3 TAXONOMIE AUS SICHT DER MMI

In der Mensch-Maschine-Interaktion spielen Gesten bzw. deren Detektion und Klassifikation seit längerer Zeit eine entscheidende Rolle (s. Kapitel 4). Dabei sind die folgenden Hauptforschungsrichtungen zu unterscheiden:

- Die **Gesteninterpretation** zum Zwecke der Kommunikation - in engerem Sinne also gestenbasierte Mensch-Maschine-Interfaces mit dem Ziel der intuitiven Bedienbarkeit.
- **Zeichen- und Gebärdensprache-Erkennung** ebenfalls zum Zwecke der Kommunikation, häufig mit einem starken Fokus auf der Unterstützung behinderter Menschen. Der Unterschied zum ersten Punkt liegt in der Art der zu erkennenden Gestik sowie im Zeichenrepertoire, das hier meist wesentlich größer und stärker konventionalisiert ist.
- **Aktionserkennung**, also Interpretation von Gesten bzw. häufig von Ganzkörperbewegungen, nicht zur Kommunikation, sondern zur Klassifikation des aktuellen Kontextes. Das Ziel besteht hier darin, die Intention eines oder mehrerer Nutzer zu erkennen, ohne dass eine explizite (bewusste) Kommunikation zwischen Nutzer und technischem System stattfindet. Diese Sichtweise ist insbesondere

im Überwachungsbereich interessant, z.B. zur automatischen Erkennung von tätlichen Angriffen oder Diebstählen durch Überwachungskameras.

- **Posenschätzung**, d.h. die Ermittlung der Körperkonfiguration einer Person, z.B. in Form von modellbasierten Gelenkwinkeln. Dies kann zum Zwecke des Motion Capturing in der Computergrafik dienen, zur Szenenbeschreibung oder als Vorstufe zur Gesteninterpretation bzw. Aktionserkennung.

- Schließlich die **Emotionserkennung**, die zum Ziel hat, aus der Körpersprache und Mimik eines Menschen auf dessen Gemütszustand zu schließen.

Typischerweise erfolgt die Datenakquisition videobasiert, d.h. mit Methoden der Bildverarbeitung, seltener auch mit spezialisierten Sensoren, beispielsweise Beschleunigungssensoren. Weiterhin kann - insbesondere bei Mensch-Maschine-Interfaces und bei der Emotionserkennung - unterstützend eine Spracherkennung erfolgen.

Im vorhergehenden Abschnitt wurde eine umfassende Gestentaxonomie aus Sicht der Semiotik vorgestellt. Im Kontext der Mensch-Maschine-Interaktion hat sich eine wesentlich einfachere, technischere Sicht auf Gestik etabliert: Unterschieden wird zwischen Ganzkörpergesten und Gesten, die mit einzelnen Körperteilen ausgeführt werden. Erstere sind hauptsächlich für die Bereiche Aktions- und Emotionserkennung von Interesse, letztere werden häufig auf einarmige Gesten oder Handgesten bzw. -zeichen reduziert und dienen vor allem zur expliziten, benutzerinitiierten Kommunikation. Weiterhin wird zwischen **statischen** und **dynamischen** Gesten unterschieden.

Statische Gesten sind Körperkonfigurationen, deren Information hauptsächlich in der spatialen Anordnung der Körperteile liegt, also gestalt- oder formbasiert. Sie bleiben über einen gewissen Zeitraum konstant und können mit Methoden der erscheinungsbasierten Objekterkennung bzw. durch Extraktion von gestaltbeschreibenden Merkmalen interpretiert werden. Typische Vertreter dieser Gestenart sind z.B. das „Victory“ Zeichen oder „Daumen hoch“ zur Signalisierung von Zustimmung. Diese Art Gesten wird häufig auch als Posen oder Posturen[1] bezeichnet.

Unter dynamischen Gesten versteht man räumlich-zeitliche Bewegungsmuster des Körpers oder einzelner Körperteile, wobei die Information ausschließlich oder hauptsächlich durch die Bewegung übermittelt wird. Hier liegt der Fokus auf der Extraktion

1 Obwohl diese Begriffe häufig parallel verwendet werden, sind sie streng genommen nicht gleichbedeutend. Eine Pose bezeichnet die Gesamtkonfiguration einer Entität im umgebenden Raum, also beispielsweise auch Ausrichtung und Geschwindigkeit eines Fahrzeuges, und ist somit ein allgemeiner gefasster Begriff als Postur, welcher bewusst herbeigeführte, signifikante Körperzustände beschreibt. Statische Gesten im Verständnis der og. Taxonomie sind also Posturen.

und Klassifikation geeigneter, die Bewegung beschreibender Merkmale, z.B. Trajektorien, also der Zeitreihenanalyse. Eine typische dynamische Geste ist beispielsweise das Winken, um den Wunsch nach Aufmerksamkeit zu signalisieren.

Eine gewisse Sonderstellung nehmen Zeigegesten (deiktische Gesten) ein, da sie sich sowohl in ihrer Bewegungscharakteristik bzw. Posturinterpretation als auch in der Art der übertragenen Information von gewöhnlichen Gesten unterscheiden. Zeigegesten können sowohl dynamisch als auch statisch sein (bzw. bestehen aus einem dynamischen und statischen Teil) und tragen für sich alleine betrachtet i.d.R. keine Information. Eine Zeigegeste muss immer im Kontext der Umgebung betrachtet werden. Sie stellt einen der wichtigsten Aspekte sprachbegleitender nonverbaler Kommunikation dar, da sie Sprache in einfacher und eindeutiger Weise mit referenzierten Objekten in der Umgebung verknüpft.

## 3.4 GESTEN IM KONTEXT DIESER ARBEIT

Im Weiteren wird der Begriff „Geste“ für dynamische Gesten verwendet, während „Postur“ eine statische Geste bezeichnet. Diese Arbeit hat die Entwicklung eines Gesteninterpretationssystems zur Interaktion mit einem Intelligenten Raum zum Ziel. Besonders interessant sind in diesem Zusammenhang Zeigegesten (Deiktika, Illustratoren) sowie bewusst informationstragende (statische wie dynamische) Einzelgesten (Embleme bzw. Regulatoren).

Nicht betrachtet werden Ganzkörpergesten und -posturen sowie Mimik und unbewusste Körpersprache. Der Begriff „Geste“ wird im Rahmen dieser Arbeit auf Aktionen und Posturen der Arme und Hände beschränkt, welche vom Nutzer des intelligenten Raumes bewusst mit der Intention ausgeführt werden, eine Information zu übermitteln bzw. eine bestimmte Reaktion eines technischen Systems zu bewirken. Es wird von einem kooperativen Nutzer ausgegangen, der über die gestenbasierten Interaktionsmöglichkeiten informiert ist und die Absicht hat, diese zu benutzen.

## 3.5 ANFORDERUNGEN AN EIN GESTENERKENNUNGSSYSSTEM

Wir Menschen können ohne jede Anstrengung eine Vielzahl unterschiedlicher Gesten unserer Interaktionspartner wahrnehmen und interpretieren. Die Leistung des menschlichen Gehirns besteht vor allem darin, dass dies auch für unterschiedlichste Ausprägungen der selben Geste, bei Anwesenheit von Störungen, teilweiser Verdeckung usw. funktioniert, also in der Robustheit und Fähigkeit zur Verallgemeinerung. Die Nachbildung dieser Fähigkeiten in einem automatischen System ist eine hochkom-

plexe Aufgabe, in deren Rahmen mehrere Einzelprobleme zu lösen sind. Im Folgenden soll ein kurzer Überblick über die Natur der Problemstellungen gegeben werden.

DATENAKQUISITION Rechnersysteme arbeiten mit digitalisierten Daten. Um Zustände der Umgebung wahrnehmen zu können, werden Sensoren benötigt, welche die analoge reale Welt in eine diskrete, vom Rechner verarbeitbare Repräsentation überführen. Sensoren stellen somit das technische Gegenstück zu menschlichen Sinnesorganen dar. Für die nichtinvasive[2] Gestenerkennung sind hauptsächlich Bilddaten von Belang, es werden also Kameras benötigt. Da Gesten sich mitunter durch eine hohe Dynamik auszeichnen, ist eine entsprechend hohe Bildaufnahme- und Verarbeitungsfrequenz notwendig. Die Datenmenge, die von den Sensoren geliefert wird, kann deshalb sehr groß werden, und die effiziente Weiterleitung und Verteilung der Daten vom Sensor zum System ist ein wichtiger Punkt, der zu beachten ist.

PERSONENDETEKTION Im Mittelpunkt eines intelligenten Systems steht der Nutzer. Er gibt Kommandos, er definiert zu einem großen Teil den Kontext der Umgebung und alle Aktionen des Systems sind auf ihn ausgerichtet. Deshalb muss dem System bekannt sein, ob und wo sich Personen in seinem Wirkungsbereich befinden. Die Detektion anwesender Personen stellt also eine zentrale Anforderung dar.

Ein Detektor vollführt immer einen Balanceakt zwischen Generalisierungsfähigkeit und Zuverlässigkeit (bzw. Detektionsrate und Anzahl Fehldetektionen). Generalisierungsfähigkeit ist nötig, um unterschiedliche Ausprägungen eines Musters (z.B. teilweise verdeckte Personen, unterschiedliche Größe und Kleidung, verschiedene Stimmmuster) erkennen zu können. Dies stellt eine Grundvoraussetzung für den Einsatz in realen Umgebungen dar, denn ein System, das nur auf ganz spezifische Nutzer abgestimmt ist, kann nur in wenigen eingeschränkten Szenarien zum Einsatz kommen. Typischerweise sollen möglichst viele (im Idealfall alle) anwesenden Personen gefunden werden, auch wenn sich darunter unbekannte Personen befinden.

Zuverlässigkeit heißt, dass die Hypothesen der Personenerkennung mit hoher Wahrscheinlichkeit verlässlich sind. Detektierte Personen sollten auch wirklich Personen sein und ihre Lokalisierung sollte im Rahmen einer gewissen Toleranz korrekt sein. Das Ziel der hohen Zuverlässigkeit steht dabei in Konkurrenz zur Generalisierungsfähigkeit, da eine hohe Generalisierungsfähigkeit ein allgemeineres (und damit „schwächeres“) Modell voraussetzt, was die Gefahr von Fehldetektionen erhöht. In einem realen Mus-

2 D.h. ohne Eingriffe in die Freiheit des Benutzers. Dies schließt ein, dass der Nutzer zur Benutzung des Systems keinerlei künstliche Hilfsmittel, wie z.B. Farbmarkierungen, Datenhandschuhe oder elektronische Messvorrichtungen, tragen muss.

tererkennungssystem werden immer Detektionsausfälle und Fehldetektionen auftreten, das System muss also damit umgehen können.

DETEKTION EINZELNER KÖRPERTEILE Für die Analyse von Gestik ist die Lokalisierung einer Person alleine nicht ausreichend. Gesten sind durch bestimmte Stellungen oder Bewegungen der Körperteile relativ zueinander bestimmt (vgl. Kapitel 3.3). Daher ist es notwendig, einzelne Körperteile (insbesondere Kopf, Arme, Hände) zu lokalisieren.

Dieses Problem ist analog zur vorher beschriebenen Personendetektion, es ist jedoch schwieriger. Im Verhältnis zum menschlichen Körper sind einzelne Körperteile, insbesondere Hände, in den Bilddaten schlecht aufgelöst und mitunter stark artikuliert, d.h. verformbar. Es ist mit schnellen Bewegungen (schnelle Positionsänderung in Verbindung mit Bewegungsunschärfe) und Änderungen der Konfiguration sowie mit häufig auftretenden Verdeckungen zu rechnen. Detektoren für einzelne Körperteile sind daher i.d.R. deutlich unzuverlässiger als Ganzkörperdetektoren. Dennoch sind sie für das hier angestrebte System von entscheidender Wichtigkeit.

TRACKING Für ein Gestenerkennungssystem ist *Tracking* in dreierlei Hinsicht interessant: Erstens sind *Tracker* aufgrund des eingeschränkten Suchraumes typischerweise um Größenordnungen schneller als Detektoren, die eine erschöpfende Suche durchführen. Dies kann in Hinblick auf Echtzeitanforderungen wichtig sein. Zweitens können Gesten dynamisch sein (vgl. Kapitel 3.3), sind also genau durch die Raum-Zeit-Trajektorien von Körperteilen definiert, die durch *Tracking* ermittelt werden können. Drittens können die Ergebnisse eines zuverlässigen *Trackers* zur Elimination von Fehldetektionen und zur Überbrückung von Detektionsausfällen genutzt werden.

In diesem Zusammenhang ist insbesondere das *Initialisierungsproblem* zu lösen: Das Modell eines *Trackers* muss geeignet mit der aktuell zu verfolgenden Objektinstanz initialisiert werden. Dies kann aus offensichtlichen Gründen nicht manuell geschehen, es muss also ein Ansatz zur automatischen Initialisierung entwickelt werden. Weiterhin muss es möglich sein, die Hypothesen des *Trackers* zu verifizieren, so dass eine fehlgeschlagene Verfolgung oder ein degeneriertes Modell detektiert und der *Tracker* ggf. reinitialisiert werden kann.

GESTENDETEKTION Nicht alle beobachteten Gesten sind aussagekräftig und nicht jede Geste des Nutzers ist an die intelligente Umgebung gerichtet. Weiterhin kann nicht davon ausgegangen werden, dass unterschiedliche Gesten klar voneinander abgegrenzt auftreten. Ein zu lösendes Problem besteht also in der Segmentierung der

beobachteten Daten und der Detektion von bewussten, an das System gerichteten Gesten der Nutzer.

Üblicherweise werden systemrelevante Aktionen relativ selten auftreten. Demzufolge wird der Datenstrom zum größten Teil aus unbekannten bzw. nicht aussagekräftigen Ereignissen bestehen. Relevante Aktionen tauchen darin unvermittelt und ohne Vorankündigung auf. Die Aufgabe besteht darin, die unrelevanten Informationen zu ignorieren und nur auf die wenigen relevanten Ereignisse zu reagieren. Dieses Problem ist als *(Event) Spotting* bekannt. Technisch betrachtet muss zu diesem Zweck eine kontinuierliche Analyse des Datenstromes stattfinden. Dabei muss ein Rückweisungskriterium für unbekannte Ereignisse existieren, um Fehldetektionen zu minimieren. Die Auswertung von Kontextinformation kann dabei hilfreich sein, beispielsweise könnte ein Sprachkommando als Hinweis dienen, dass gerade etwas für das System Interessantes passiert ist.

GESTENKLASSIFIKATION Um auf Gesten des Benutzers reagieren zu können, muss ein Gestenerkennungssystem entscheiden, welche Geste gerade vollführt wurde. Es muss also in der Lage sein, die aktuelle Beobachtung einer bekannten Kommandogeste zuzuordnen. Dies ist ein klassisches Klassifikationsproblem[3]. Konkret muss jedem Segment des betrachteten Datenstroms ein Klassenkennzeichen zugewiesen werden. Die Menge der bekannten Klassen entspricht dabei der Menge an Kommandogesten, welche innerhalb der intelligenten Umgebung erkannt und interpretieret werden sollen. Sobald eine solche Zuordnung erfolgreich vorgenommen wurde, gilt die entsprechende Aktion als erkannt und die intelligente Umgebung kann darauf reagieren. Zusätzlich muss ein Rückweisungskriterium definiert sein, d.h. Ereignisse, die nicht gut zu den bekannten Klassen passen, müssen als unbekannt verworfen werden.

Die Leistung eines Klassifikators hängt entscheidend von der Qualität der Daten und dem Informationsgehalt der gewählten Datenrepräsentation ab. Der Vorgang des Klassifikatorentwurfs besteht aus der Wahl einer geeigneten, möglichst diskriminativen Datenrepräsentation (Merkmale, vgl. Kapitel 2) und der Bestimmung eines Verfahrens, welches die Klassentrennung vornimmt. Ist die Repräsentation so gewählt, dass Instanzen der unterschiedlichen Klassen sich sehr deutlich voneinander unterscheiden, wird das Klassifikationsproblem trivial. Da dies bei realen Daten i.d.R. nicht möglich ist, muss mit Fehlern gerechnet werden. Insbesondere kann es vorkommen, dass Klassen nicht eindeutig trennbar sind. Der Klassifikator muss so beschaffen

3 Genau genommen sind die im Vorfeld angesprochene Personendetektion, Körperteildetektion sowie das *Spotting* ebenfalls Klassifikationsaufgaben.

sein, dass der Klassifikationsfehler (also der Anteil fehlerhafter Klassenzuordnungen) minimiert wird. Dies ist umso schwieriger, je größer die Anzahl der Klassen ist.

3D-KOMBINATION Neben dem in Kapitel 2.9 bereits erwähnten Informationsverlust der bei der Projektion einer dreidimensionalen Szene auf eine zweidimensionale Bildebene auftritt, kann es bei Einsatz einer einzelnen Kamera zusätzlich zu Verdeckungen von Personen oder einzelner Körperteile kommen, wodurch eine an sich bedeutungstragende Geste ggf. gar nicht oder nur teilweise wahrgenommen werden kann. Zudem ist der menschliche Körper ein dreidimensionales Gebilde, d.h. sämtliche Gesten, die durch Körperstellungen oder -bewegungen charakterisiert werden, müssen zur korrekten Deutung in 3D ausgewertet werden, sofern nicht einige deutliche Einschränkungen bezüglich des Szenarios getroffen werden. Ein gutes Beispiel hierfür sind Zeigegesten. Die Ergebnisse in [160] zeigen z.B., dass die erreichbare Genauigkeit bei der Schätzung der Zeigerichtung aus monokularen Daten sehr beschränkt ist. Zudem kann ohne zusätzliches Kontextwissen nicht zwischen Objekten unterschieden werden, die in unterschiedlicher Tiefe im Raum vorhanden sind, aber auf ähnliche Positionen im Bild projiziert werden (vgl. z.B [169]. Aus diesen Gründen sollte die Analyse von Gestik im dreidimensionalen Raum stattfinden.

Eine Lösung des Problems ist entweder mit dedizierter Sensorik möglich (z.B. kalibrierte Stereokameras, Time-of-flight-Kameras), die bereits 3D Daten misst, oder durch die Verwendung mehrerer Kameras. Die letzte Variante hat den Vorteil, dass durch eine geschickte Verteilung im Raum eine sehr gute Abdeckung des Raums mit den Blickfeldern der Kameras erreicht werden kann. Verdeckungen können somit weitgehend ausgeschlossen werden.

MODELLIERUNG VON UMGEBUNG UND KONTEXT Unter Kontext versteht man den zeitlichen, räumlichen und kausalen Zusammenhang, in dem eine Aktion zu bewerten ist. Dazu gehören u.A. der aktuelle Systemzustand, die Menge der anwesenden Personen und ihre Rollen, eine Bewertung der momentanen Situation (handelt es sich z.B. um eine Vortragssituation, eine Konferenz oder ein lockeres Gespräch), oder die Konfigurationen und Zustände von Gegenständen im Raum. Eine komplette kontextuelle Beschreibung einer Umgebung ist also sehr komplex und mit automatischen Methoden nur sehr schwer möglich. Jedoch können bereits einfache Beziehungen helfen, Mehrdeutigkeiten aufzulösen oder Problemstellungen zu vereinfachen (vgl. z.B. [129, 114]). Beispielsweise kann eine bedeutungsvolle Zeigegeste dadurch definiert werden, dass sie ein bekanntes Objekt im Raum referenziert. Die Interaktion eines Nutzers mit einem bestimmten Objekt kann einen Hinweis auf die aktuelle Situation

und zu erwartende Aktionen geben, oder die Interpretation des Kommandos „Licht“ kann davon abhängen, ob das Licht im Moment an- oder ausgeschaltet ist.

Bis zu welcher Komplexität Kontext modelliert wird, ist eine wichtige Designentscheidung, und das automatische Inferieren von Kontextwissen ist ein ungelöstes Forschungsproblem. Im Rahmen eines intelligenten Systems müssen mindestens die relevanten Systemzustände zu jeder Zeit bekannt sein. Weiterhin muss eine Repräsentation der Umgebung existieren, damit sinnvoll auf Aktionen reagiert werden kann. Beispielsweise ist die Berechnung einer Zeigerichtung sinnlos, wenn nicht bekannt ist, wo im Raum sich die Person befindet und worauf gezeigt werden kann.

REAKTIVITÄT Losgelöst von der konkreten Implementierung spielen bei der algorithmischen Umsetzung der gewählten Problemlösungen Überlegungen zu Laufzeitverhalten und Latenz eine Rolle. Damit ein solches System sinnvoll eingesetzt werden kann, muss es einerseits Daten schnell genug verarbeiten können, um z.B. schnell ausgeführte dynamische Gesten erfassen zu können, andererseits muss die Reaktion auf Aktionen des Nutzers innerhalb zumutbarer Zeit geschehen. Dies wird gemeinhin als Echtzeitfähigkeit bezeichnet, wobei die Verwendung des Begriffes „Echtzeit“ in diesem Zusammenhang irreführend ist[4]. In dieser Arbeit wird daher der Begriff *Reaktivität* verwendet. Ein ausreichend reaktives System muss in der Lage sein, so schnell auf Eingaben zu reagieren, dass die entstehenden Verzögerungen vom Benutzer nicht als störend wahrgenommen werden. Im Folgenden wird davon ausgegangen, dass eine Reaktionszeit von wenigen ($< 5$) Sekunden ausreichend ist.

Verzögerungen entstehen nicht nur aufgrund Latenzen bei der Übertragung und Verarbeitung von Daten, sondern insbesondere auch als Folge der Auswertung von Zeitreihen. So hat z.B. ein Verfahren, das zur Zeitreihenanalyse Messwerte innerhalb eines Zeitfensters betrachtet, eine Latenz, die mindestens der Länge des Fensters entspricht. Deshalb beeinflussen Anforderungen an die Reaktivität nicht nur die konkrete technische Umsetzung, sondern auch maßgeblich die Wahl der Methoden. Bestimmte Verfahren der Mustererkennung verbieten sich automatisch durch ihre Komplexität oder inhärente Latenz.

4 Streng genommen bedeutet Echtzeitfähigkeit, dass Daten garantiert in der Rate verarbeitet werden, in der sie von den Sensoren geliefert werden. Im vorliegenden System würde das z.B. für die Kameras eine Verarbeitungsgeschwindigkeit von ca. 20 Bildern pro Sekunde erfordern. Dies ist nicht nur schwierig zu erreichen, sondern im Allgemeinen auch nicht notwendig.

# 4 VERWANDTE ARBEITEN

Dieses Kapitel soll einen Überblick über Arbeiten geben, deren Ziel die videobasierte Erkennung von Gesten ist. Hierbei werden ebenfalls die thematisch und/oder methodisch stark verwandten Gebiete Posenschätzung und Aktionserkennung betrachtet. Methoden, die sich nicht auf visuelle Daten, sondern z.B. auf die Messdaten von körpermontierten Beschleunigungssensoren verlassen (vgl. z.B. [86]), werden nicht betrachtet. Die Gesten- und Aktionserkennung ist ein sehr dynamisches Forschungsgebiet mit einer großen Anzahl an jährlichen Veröffentlichungen (vgl. Abschnitt 4.1). Deshalb kann dieses Kapitel nicht den Anspruch erheben, einen vollständigen Überblick zu geben. Im Folgenden werden zunächst einige aktuelle Artikel vorgestellt, die eine Übersicht über Teile des Forschungsfeldes präsentieren. Im Anschluss daran wird die Kategorisierung eingeführt, mit deren Hilfe relevante Literatur strukturiert wird. Das Kapitel schließt mit einer detaillierteren Übersicht über ausgewählte Arbeiten, die aus Sicht dieser Dissertation von besonderem Interesse sind, sowie einem kurzen Fazit. Literatur, die sich nicht auf die Erkennung von Gesten bezieht, sondern auf einzelne im Rahmen dieser Arbeit eingesetzte Verfahren und Methoden, wird im Folgenden zum Zwecke besserer Übersichtlichkeit zunächst nicht betrachtet und im späteren Verlauf an geeigneteren Stellen referenziert.

## 4.1 LITERATURÜBERSICHTEN UND TAXONOMIEN

Aufgrund der großen Aufmerksamkeit, welche das weite Feld der videobasierten Bewegungs-, Aktions- und Gestenanalyse seit einigen Jahren in der Forschung genießt, existiert eine Reihe von Übersichtsartikeln, die das Forschungsfeld beleuchten und Taxonomien für die verschiedenen Ansätze einführen. Einen sehr umfassenden Überblick über verschiedene Aspekte der Bewegungsanalyse und Aktionserkennung – von der Initialisierung über Trackingansätze bis zur Klassifikation – geben Moeslund et al. [127, 128]. Sie unterscheiden zwischen Aktionsprimitiven[1], Aktionen[2] und Aktivitäten[3]. Poppe [150] übernimmt diese Taxonomie und gibt zusätzlich einen Überblick über

1 Elementare, kurze Bewegungen einzelner Körperteile.
2 Elementare Ganzkörperbewegungen, die ggf. aus mehreren Aktionsprimitiven zusammengesetzt sind.
3 Komplexere Folgen von Aktionen.

aktuelle Referenz-Datensätze. Er beschränkt sich allerdings auf Ganzkörper-Aktionen und klammert die (arm- oder handbasierte) Gestenerkennung explizit aus.

Im Gegensatz dazu erweitern Turaga und Kollegen [197] den Begriff der Aktivität auf Interaktionen zwischen mehreren Personen und behandeln somit auch Ansätze, die dem Themengebiet der Überwachung zuzuordnen sind. Explizit auf die Erkennung von Hand- und Armgesten in Interaktionsszenarien zielt der Artikel von Mitra und Acharya [125]. Zusätzlich betrachten sie die Analyse von Gesichtsausdrücken und Emotionen. Ong und Ranganath [142] beschäftigen sich mit der Erkennung von Zeichensprache, also künstlichen Gestenalphabeten mit definierter Struktur.

## 4.2 KATEGORISIERUNG

Die folgende Kategorisierung der relevanten Literatur orientiert sich grob an derjenigen in [128], unterscheidet sich aber an verschiedenen Stellen von dieser. Aufgrund der großen Zahl von Veröffentlichungen und der methodischen Vielfalt empfiehlt sich eine orthogonale Kategorisierung, in der hybride Ansätze ggf. in mehreren Kategorien auftauchen. Im Folgenden wird die Einteilung gemäß folgender Kriterien vorgenommen:

- Zielsetzung: Für welche Aufgabe ist ein Verfahren konzipiert?
- Lokalisierung: Wie werden aufgabenrelevante Bildregionen identifiziert?
- Merkmale: Welche Repräsentationen werden zur Beschreibung von Einzelbildern verwendet?
- Temporale Repräsentation und Integration: Welche Merkmale werden benutzt, um zeitliche Beziehungen zwischen aufeinanderfolgenden Videobildern zu erfassen?
- Körpermodell: Wie werden Lagebeziehungen sowie kinematische und anthropologische Beschränkungen zwischen Körperteilen modelliert?
- Erkennung: Welche Methoden existieren, um anhand der extrahierten Merkmale und Modellrepräsentationen Entscheidungen zu treffen?

## 4.3 ZIELSETZUNG

Diese Unterteilung folgt im Wesentlichen der in Kapitel 3.3 vorgestellten Systematik, d.h. die Arbeiten werden gemäß ihrer Intention eingeteilt in Posenschätzung, Akti-

onserkennung, Zeichenspracheerkennung und Gesteninterpretation. Ausgespart wird der Bereich der Emotionserkennung, weil diese Aufgabenstellung sich sehr stark von den hier verfolgten Zielen unterscheidet. Ebenfalls nicht betrachtet wird die Aktivitätserkennung in Überwachungsszenarien, für einen Einstieg in dieses Thema sei auf [62, 80, 118] verwiesen.

### 4.3.1 *Posenschätzung*

Unter Posenschätzung versteht man die automatische Ermittlung der Konfiguration eines Objektes relativ zu seiner Umgebung anhand beobachteter Daten und ggf. a-priori Modellwissen. Besteht ein Objekt aus mehreren gegeneinander beweglichen Teilen, wie es z.B. beim menschlichen Körper der Fall ist, so schließt die Posenschätzung die Ermittlung der gegenseitigen relativen Positionen der Modellteile (d.h. der Postur des Objektes) ein. Viele Arbeiten in diesem Feld verfolgen nicht ausdrücklich das Ziel, die ermittelte Pose des Objektes zu klassifizieren und zu interpretieren. Da statische Gesten aber durch eine bestimmte Konfiguration von Körperteilen – also die Postur eines Teiles des Objektes – gegeben sind und dynamische Gesten aus einer bestimmten Abfolge von Posturen bestehen, ist die Zielsetzung der Posenschätzung jener der Gestenerkennung ähnlich und beide Gebiete verwenden mitunter ähnliche Ansätze.

Die Ermittlung einer zweidimensionalen Pose aus monokularen Bilddaten ist das Ziel von [11, 24, 38, 54, 55, 97, 154]. Die Reduzierung der Körperpostur auf 2D umgeht das grundlegende Problem der monokularen videobasierten Posenschätzung, dass durch die Projektion auf die Bildebene und den damit verbundenen Verlust von Tiefeninformation Mehrdeutigkeiten in der Poseninterpretation entstehen. Allerdings sind derartige Ansätze nur bedingt für den Einsatz in realen, uneingeschränkten Interaktionsszenarien geeignet, da eine 2D-Posenbeschreibung grundsätzlich nicht ansichtsinvariant ist und somit entscheidend von der relativen Orientierung der Person zur Kamera abhängt. Um sicherzustellen, dass immer eine geeignete Ansicht verfügbar ist, wäre ein Multikamerasystem mit sehr guter Sichtfeld-Abdeckung und entsprechend vielen Kameras notwendig. Aus diesem Grund ist eine dreidimensionale Posenrepräsentation wünschenswert.

Im Falle der Schätzung einer solchen Repräsentation aus monokularen Daten [1, 16, 39, 72, 120, 131, 134, 175, 183, 199] stellt die fehlende dritte Dimension eine versteckte Variable dar, die im Zuge der Inferenz modellbasiert oder durch zeitliche Integration ermittelt werden muss. Dabei ist sicherzustellen, dass die inhärenten Mehrdeutigkeiten zu sinnvollen und anthropologisch plausiblen Ergebnissen aufgelöst werden, die trotzdem die Beobachtung gut erklären. Eine andere Herangehensweise besteht in der

Verwendung von Stereo- oder Multikamerasystemen. Dies ermöglicht eine explizite 3D-Rekonstruktion [19, 28, 29, 90, 112, 122] oder eine Integration der Ergebnisse verschiedener Ansichten [78, 126, 211], wodurch Mehrdeutigkeiten zumindest zum großen Teil vermieden bzw. aufgelöst werden können.

### 4.3.2 *Aktionserkennung*

Aktionserkennung (oder auch Verhaltenserkennung) unterscheidet sich von der Gestenerkennung im Verständnis dieser Arbeit in zweierlei Hinsicht. Erstens spielt für die Aktionserkennung häufig die Dynamik eine entscheidende Rolle, es geht also um die Klassifikation einer Zeitreihe bzw. Folge von Ereignissen mit ggf. im Vergleich zu einer einzelnen Geste langen Zeitbasen. Zweitens ist das Ziel der Aktionserkennung weniger die Erkennung einzelner dynamischer „Zeichen" – entgegen z.B. der Gestenerkennung zum Zwecke der Mensch-Maschine-Interaktion – sondern von komplexeren Aktionen, die ggf. auch Interaktionen zwischen mehreren Personen beinhalten können (dann auch als „Aktivitätserkennung" bezeichnet, vgl. [197]). Eine Aktion befindet sich also typischerweise auf einem höheren Abstraktionsniveau als eine Geste, kann aber verschiedene Gesten oder eine definierte Abfolge von Posturen enthalten. Dennoch kommen in beiden Anwendungsfällen oft ähnliche Methoden zum Einsatz, und viele Ansätze aus dem Bereich der Aktionserkennung könnten ebenso zur Gesten- oder Posenerkennung verwendet werden oder beinhalten diese als Teilschritte.

Analog zur Posenerkennung lässt sich auch bei der Aktionserkennung eine Einteilung vornehmen in Ansätze, die für Multikamera-Anwendungen ausgelegt sind [88, 205, 206] und solche, die mit monokularen Bilddaten arbeiten. Hier muss wiederum unterschieden werden zwischen Methoden, die in gewissen Grenzen durch Verwendung geeigneter Repräsentationen ansichtsinvariant sind [42, 65, 102, 108, 130, 135, 155, 180] und Ansätzen, die eher für die Aktionserkennung unter festen Blickwinkeln bzw. nur geringen Blickwinkelvariationen geeignet sind [14, 56, 69, 81, 84, 100, 140, 145, 151, 153, 172, 173, 194, 195, 215].

Die Beurteilung der genannten Ansätze hinsichtlich ihrer Robustheit gegenüber Variationen des Blickwinkels und anderer Faktoren, wie z.B. Hintergrund, ist anhand der publizierten Ergebnisse oft schwierig. Das liegt daran, dass sich im Bereich der Aktionserkennung als Benchmark-Datensätze hauptsächlich der sog. KTH [174] und der Weizmann Datensatz [65] etabliert haben. Dies erleichtert zwar die Vergleichbarkeit der Ergebnisse, allerdings enthalten diese Datensätze monokulare Bildsequenzen mit geringen Variationen der og. Faktoren. Zwar existieren andere frei verfügbare Datensätze

mit Multikamera-Daten [182, 206] oder realen, sehr anspruchsvollen Filmsequenzen [114], diese haben sich aber bislang nicht auf breiter Front etabliert.

### 4.3.3 *Erkennung von Zeichen- und Gebärdensprache*

Zeichensprache-Zeichen sind durch statische Posturen [25, 109, 141, 178, 190, 196, 202] oder Bewegungsmuster der Hände [17, 23] definiert. Die Intention der Zeichenspracheerkennung ist nicht die Realisierung möglichst natürlicher und intuitiver Mensch-Maschine-Interfaces, schließlich ist das gewählte Zeichenrepertoire in den seltensten Fällen intuitiv und erfordert ohnehin spezielles Training der gestikulierenden Personen. Demzufolge ist das Aufnahmeszenario oft eingeschränkt, indem beispielsweise Hilfsmittel wie Handschuhe oder farbige Markierungen verwendet werden [109, 203] oder die Hand sich in einem bestimmten vordefinierten Bereich des Bildes befinden muss bzw. der Kameraausschnitt so eingeschränkt wird, dass nur die Hand im Bild enthalten ist [23, 25, 141, 178, 190, 196, 202]. Allerdings zeichnen sich Systeme zur Zeichenspracheerkennung häufig durch ein großes Zeichenrepertoire aus (z.B. 46 Zeichen in [109] oder 100 in [178]).

### 4.3.4 *Gesteninterpretation*

Unter Gesteninterpretation soll im Rahmen dieser Arbeit die automatische Erkennung von (konventionalisierten) Kommandogesten und -posturen verstanden werden, die explizit bedeutungstragend (Embleme) oder bedeutungsunterstreichend (Illustratoren) sind. Der Unterschied zur Zeichenspracheerkennung liegt in Art, Ausführung und Intention der Gestik: Betrachtet werden Posturen oder Trajektorien, die nicht als Zeichen eines Alphabetes interpretiert werden, sondern als Kommando eine direkte Systemreaktion bewirken. Die Ausführung dieser Gesten ist hierbei nicht nur auf die Hand beschränkt, sondern umfasst ggf. die Arme oder den gesamten Oberkörper.

Bei der Gesteninterpretation in monokularen Kamerakonfigurationen [2, 3, 12, 18, 35, 45, 77, 95, 97, 103, 104, 105, 113, 164, 176, 179] besteht grundsätzlich das gleiche Problem der fehlenden Blickwinkel-Unabhängigkeit wie bei der monokularen Posen- und Aktionserkennung. Typischerweise wird daher vorausgesetzt, dass die gestikulierende Person frontal der Kamera zugewandt ist und die Geste in einer Ebene annähernd parallel zur Bildebene ausführt. Diese Annahme ist allerdings nicht so restriktiv, wie es auf den ersten Blick scheint: Bei einer bewussten Interaktionsabsicht wenden Menschen sich üblicherweise ihrem Interaktionspartner zu. Wenn der Interaktionspartner im Falle eines Gestenerkennungssystems also eine Kamera ist, kann mit einiger Berechtigung

angenommen werden, dass der Benutzer sich dieser zuwendet. Dieser Effekt kann durch das Vorhandensein eines Rückmeldungskanals (z.B. Avatare, Interface-Agenten) vom technischen System hin zum Nutzer noch verstärkt werden. Voraussetzung ist offensichtlich, dass der Nutzer weiß, wer und wo sein Interaktionspartner ist, was in gewissem Sinne dem *Disappearing Computer* Paradigma widerspricht.

Neben rein monokularen Ansätzen haben sich im Bereich der Gesteninterpretation hauptsächlich Stereosysteme etabliert [20, 27, 33, 34, 49, 85, 132, 137, 143, 204]. Diese lösen das Problem der Blickwinkel-Invarianz nur bedingt, da der Kameraaufbau starr ist und insbesondere bei Stereosystemen mit kurzer Baseline die gleichen Verdeckungsprobleme auftreten, wie bei monokularen Kameras. Stereo- bzw. Disparitätsbilder können aber z.B. die Lokalisierung oder Segmentierung einzelner Körperteile entscheidend unterstützen, da Objektgrenzen und zusammenhängende Segmente nicht mehr nur rein ansichtsbasiert, sondern zusätzlich anhand der Tiefenwerte ermittelt werden können. Echte Multikamera-Ansätze sind dagegen relativ selten [26, 91, 93].

Ein erheblicher Teil der publizierten Literatur legt einen besonderen Schwerpunkt auf die Erkennung und Analyse von Zeigegesten bzw. beschäftigt sich ausschließlich damit. In diesem Zusammenhang ist neben der eigentlichen Erkennung der Zeigegeste insbesondere die Schätzung der angezeigten Richtung interessant. Diese kann beispielsweise zur Definition von Objektreferenzen [26, 71, 77, 143, 176], zur gestenbasierten Einweisung eines mobilen Roboters [27, 68, 132, 164], zur Steuerung von Virtual Reality Anwendungen bzw. grafischer Oberflächen [85, 91, 137] oder zur Steuerung der Aufmerksamkeit eines technischen Systems bzw. zur Etablierung eines gemeinsamen Aufmerksamkeitsfokus [169] dienen. Die zuletzt genannte Veröffentlichung stützt sich unter Anderem auf einige der im Rahmen dieser Dissertation entwickelten Methoden.

## 4.4 LOKALISIERUNG

Der erste Schritt eines Verfahrens zur Analyse menschlicher Postur und Bewegung besteht in der Lokalisierung einer Person im Bild. Darunter versteht man die Eingrenzung des Suchbereiches auf interessante Bildausschnitte bzw. die explizite Detektion von relevanten Entitäten (im Falle der Aktions- und Gestenerkennung Personen oder deren Körperteile). Ansätze hierfür basieren i.d.R. entweder auf ansichtsbasierter Hintergrund- bzw. Vordergrundmodellierung oder Detektion. Weiterhin gibt es lokalisierungsfreie Verfahren, bei denen Lokalisierung und Inferenz entweder integriert sind oder keine explizite Lokalisierung benötigt wird.

### 4.4.1 *Vordergrund- und Hintergrundmodellierung*

Eine Möglichkeit zur Lokalisierung besteht in der Segmentierung des Bildes in Vorder- und Hintergrundregionen (vgl. Kapitel 2.8), wobei Vordergrund stets die für die jeweilige Aufgabe potentiell relevanten Bildbereiche bezeichnet und der Rest des Bildes als Hintergrund betrachtet wird.

Der einfachste Ansatz zur Vordergrundsegmentierung besteht in der Hintergrundsubtraktion. Hierbei ist das Modell des Hintergrundes der Hintergrund selbst, also das Bild. Dies setzt das Vorhandensein eines geeigneten Referenzbildes (ohne Vordergrund) sowie entweder einen statischen Hintergrund und kontrollierte Umgebungsbedingungen oder einen Adaptionsmechanismus voraus. Die Segmentierung kann dann durch einfachen pixelweisen Vergleich vorgenommen werden. Aufgrund der Einfachheit und der sehr effizienten algorithmischen Umsetzbarkeit erfreut dieser Ansatz sich großer Beliebtheit [1, 2, 3, 14, 16, 25, 28, 29, 33, 35, 54, 65, 69, 72, 78, 81, 88, 90, 95, 97, 103, 105, 108, 115, 135, 195, 199, 205, 206].

Eine etwas flexiblere und robustere Herangehensweise ist die Modellierung von Vorder- oder Hintergrund. D.h. das Modell abstrahiert von den rohen Bilddaten und repräsentiert typische Eigenschaften von Vorder- bzw. Hintergrund. Die Segmentierung beruht dann auf einer einer pixel- oder regionsweisen Klassifikation. Weit verbreitet sind in diesem Zusammenhang Modelle der Farbverteilung [11, 19, 24, 39, 85, 91, 112, 120, 122, 126, 190, 211], für die Segmentierung von Personen insbesondere der Hautfarbe [2, 3, 16, 17, 23, 25, 26, 35, 45, 49, 77, 104, 109, 115, 132, 141, 143, 178]. Weiterhin ist eine Modellierung mit Strukturinformation, beispielsweise den Antworten lokaler Filterbänke [71] oder Intensitätskanten [93], denkbar.

Die Verfügbarkeit von Tiefeninformation, also dreidimensionaler Daten, erleichtert die Segmentierungsaufgabe erheblich, falls davon ausgegangen wird, dass der Vordergrund auch im Tiefensinne Vordergrund ist, sich also deutlich durch geringere Tiefe vom Hintergrund abhebt [19, 49, 85, 132, 143, 204]. Gelegentlich wird das Lokalisierungsproblem auch durch die Verwendung von Hilfsmitteln wie farbiger Marker [18, 20] oder manuelle Initialisierung [176] gelöst. Die beiden letzten Herangehensweisen sind für nichtinvasive natürliche Mensch-Maschine-Interaktion ungeeignet.

### 4.4.2 *Detektoren*

Eine weitere Möglichkeit der Lokalisierung besteht in der Verwendung von anhand einer Trainingsstichprobe trainierten Personen- oder Körperteildetektoren. Bei dieser Vorgehensweise wird ein Bild in viele überlappende – typischerweise rechteckige –

Bildausschnitte eingeteilt. Innerhalb des jeweiligen Ausschnittes werden Merkmale berechnet, anhand derer ein Klassifikator die Entscheidung trifft, ob der Ausschnitt eine Person enthält oder nicht. Das Ergebnis ist also die Eingrenzung des Suchraumes auf eine oder mehrere Bildregionen. Derartige Detektoren kommen z.B. in [55, 131, 134, 135, 164, 172, 173, 194] zum Einsatz.

In gleicher Art kann eine Kombination mehrerer Detektoren zur unabhängigen Detektion verschiedener Körperteile eingesetzt werden. Diese Vorgehensweise wird gelegentlich in der Posen- und Aktionserkennung eingesetzt, um einen ersten Satz von Körperteilhypothesen zu erhalten, der dann mit anderen Methoden verfeinert und validiert wird [120, 145, 183]. Insbesondere in der Zeichenspracheerkennung existieren auch Ansätze, die Lokalisierung und Posen-/Gestenklassifikation integrieren, indem eine Hierarchie von Detektoren für verschiedene bekannte Posturen verwendet wird [109, 141]. Ferner kann die Detektion von Gesichtern bzw. Köpfen eine Möglichkeit zur Initialisierung und Adaption von personalisierten online-Farbmodellen zur Vordergrund-Modellierung sein [3, 132, 143].

### 4.4.3 *Lokalisierungsfreie Ansätze*

Neben den vorgestellten Lokalisierungsansätzen existieren auch Arbeiten, die keine explizite Lokalisierung benötigen, da entweder die Erkennung bzw. Inferenz die Lokalisierung als integralen Bestandteil enthält oder die genaue Position der Person im Bild weder für die Inferenz benötigt noch Teil des Ergebnisses ist. Bei Ersterem sind beispielsweise bestimmte Körpermodell-basierte Ansätze zu nennen, deren Inferenzmethode auch die Anpassung der Modellteile an die Bilddaten optimiert [11, 24, 27, 54, 154, 196][4]. Auch die im vorherigen Abschnitt bereits erwähnten integrierten hierarchischen Detektions- und Klassifikationsansätze [109, 141] können als Vertreter dieser Gruppe betrachtet werden.

Zur zweiten genannten Gruppe gehören insbesondere Keypoint- oder Interestpoint-basierte Ansätze (vgl. Kapitel 5.1) [42, 56, 88, 100, 102, 108, 140, 215]. Hierbei werden an mathematisch wohldefinierten „interessanten“ Bildpunkten lokale Deskriptoren berechnet. Die Inferenz oder Erkennung erfolgt dann aufgrund des Vorhandenseins bestimmter Deskriptoren oder Gruppen von Deskriptoren, ggf. in einem losen räumlichen Zusammenhang. Die Suche nach den Keypoints wird i.d.R. nicht auf bestimmte

4 Einige der hier genannten Ansätze wurden bereits als Beispiele für eine Lokalisierung per Vordergrundsegmentierung genannt. Sie benutzen Silhouetten oder binäre Vordergrundmasken als Kriterium für die Güte der Übereinstimmung zwischen Modell und Bild, nicht explizit zur Lokalisierung oder Suchraumeinschränkung. Eine genaue Einordnung ist in diesen Fällen schwierig, da implizit trotzdem eine Vorauswahl möglicher Bildpositionen stattfindet

Bildregionen eingeschränkt und ihre genaue Lage spielt für die Erkennung keine Rolle. Darüberhinaus existieren auch nicht Keypoint-basierte Ansätze, welche lokalisierungsfrei mit globalen Repräsentationen der Bilddaten arbeiten [84, 113, 155].

## 4.5 MERKMALE

Merkmale sind numerische Werte, die von den rohen Messwerten abstrahieren. Ziel der Merkmalsextraktion ist eine möglichst kompakte und diskriminative Repräsentation der zugrunde liegenden Daten, um eine nachfolgende Erkennung bzw. Inferenz zu erleichtern. Statische Merkmale werden aus Einzelmessungen oder sonstigen statischen Repräsentationen berechnet, betrachten also nicht die temporale Komponente eines Datenstroms. Für die Merkmalsextraktion in Bildern kann nach Art und Aufbau der eingesetzten Merkmalsrepräsentation unterschieden werden zwischen Silhouettenbasierten Merkmalen, Farbmerkmalen und Strukturmerkmalen.

### 4.5.1 *Silhouettenbasierte Merkmale*

Silhouettenbasierte Merkmale beschreiben den Umriss (Kontur) oder die Gestalt einer Vordergrundregion. Die Eingabedaten der Merkmalsextraktion bestehen normalerweise aus Binärbildern, in denen Vordergrundpixel den Wert 1 und Hintergrundpixel den Wert 0 zugewiesen bekommen. Somit erfordert die Extraktion einer Silhouette oder einer Kontur die Segmentierung des Ursprungsbildes in Vorder- und Hintergrund. Insbesondere in der Posenerkennung werden binäre Silhouetten oft direkt (d.h. ohne weitere Merkmalsextraktion) verwendet, um die Übereinstimmung zwischen einem Körpermodell und den Bilddaten anhand der Überdeckung der Modellteile mit der Silhouette zu bewerten [11, 24, 54] oder Körperteile direkt durch Segmentierung oder Anpassung parametrischer Modelle zu finden [71, 85, 91, 126]. Ähnliche Ansätze für dreidimensionale Daten arbeiten mit volumetrischen Rekonstruktionen der Silhouetten aus mehreren Kamerabildern [29, 78, 112, 122, 205]. In [151] werden Differenzbilder aus zeitlich aufeinanderfolgenden Silhouetten berechnet und direkt als Merkmal zur Aktionserkennung genutzt.

Konturen können ebenfalls direkt als Merkmale benutzt werden [16, 17, 27, 183]. Häufig wird ein exemplarischer Abgleich zwischen extrahierten Konturen und entsprechenden Beispielschablonen anhand kantenbasierter Ähnlichkeitsmaße, wie der Chamfer- oder Hausdorff-Distanz [184], vorgenommen [24, 78, 120, 130, 190, 199].

Neben der direkten Verwendung von Silhouetten- oder Konturpixeln findet sich in der Literatur eine große Vielfalt davon abgeleiteter Merkmale, wie zum Beispiel sog.

*Shape Context Deskriptoren* [1, 28, 69, 72, 103, 141, 195], *Fourier-Koeffizienten* [25, 105, 206] und diverse andere Varianten [14, 33, 34, 65, 81, 97, 108, 115].

### 4.5.2 *Farbbasierte Merkmale*

Die Verwendung von Farbe zur Lokalisierung bzw. Segmentierung von Personen oder Körperteilen im Bild wurde bereits im Abschnitt 4.4 behandelt. Im Folgenden soll es hauptsächlich um Arbeiten gehen, in denen Farbinformation darüber hinausgehend als Bestandteil des Modellierungs- und Inferenzprozesses genutzt wird. Unter der Annahme, dass die Farbverteilung eines Körperteiles sich zwischen aufeinanderfolgenden Bildern nur wenig ändert, können körperteilspezifische Farbmodelle online gelernt und während der Inferenz als zusätzliches Erkennungsmerkmal genutzt [55, 90, 126, 131, 154, 211] oder in einem Tracking-Verfahren zur Generierung guter initialer Hypothesen eingesetzt werden [19, 35]. In [175] werden statische personenspezifische Farbmodelle für das Tracking eines Körpermodells verwendet. Diese müssen allerdings manuell initialisiert werden. Statische globale Modelle der Farbverteilung werden in [190, 196] in der Erkennungsphase genutzt. In [155] ist eine sog. Farbopponentenkarte integraler Bestandteil einer räumlich-zeitlichen Aktionsrepräsentation.

### 4.5.3 *Strukturbasierte Merkmale*

Neben binären Silhouetten und Farbe lässt sich auch die Struktur eines Bildes mit Merkmalen beschreiben. Strukturmerkmale erfassen Eigenschaften wie die Textur oder die Dichte und Orientierung von Intensitätsgradienten in einem lokalen Bereich um den Merkmalspunkt herum. Die einfachste Art eines Strukturmerkmals stellt die direkte Verwendung der Bilddaten (ggf. in kleinen lokalen Bildausschnitten, sog. Patches) oder von Kantenbildern dar [12, 38, 56, 84, 153, 154, 175, 176, 202].

Eine weitere Möglichkeit besteht in der Faltung des Bildes mit Filtermasken unterschiedlicher Größe und Orientierung (sog. *Jets*). Der Vektor der Filterantworten am Aufpunkt stellt dann den Merkmalsvektor dar. Weil die verwendeten Filter immer einen lokal begrenzten Einzugsbereich haben, beschreibt ihre Antwort nicht einen einzelnen Punkt, sondern eine lokale Bildregion um den Aufpunkt herum. Deshalb ermöglicht die Anwendung von Jets in einem groben Raster oder an einzelnen markanten Bildpunkten eine kompakte Repräsentation der Bildstruktur. Verbreitet sind in diesem Zusammenhang Jets von Gabor-Filtern [137, 164, 172, 173, 196], aber auch andere Filterarten sind möglich [95, 100, 155].

In ähnlicher Weise kann die Struktur lokaler Bildbereiche durch sog. *lokale Deskriptoren* beschrieben werden. Hierbei wird die Region um den Merkmalspunkt gemäß einer vorgegebenen Deskriptor-Topologie in Zellen aufgeteilt. In jeder Zelle werden dann gewisse Eigenschaften der in der Zelle enthaltenen Pixel extrahiert. Die Repräsentationen der einzelnen Zellen werden in einer festen Reihenfolge konkateniert und bilden so den Merkmalsvektor. In den meisten Fällen sind die Zellenmerkmale Histogramme der Orientierungen von Pixelgradienten (vgl. SIFT, Kapitel 5.1 und HOG, Kapitel 5.3) [39, 55, 88, 134, 135, 194] oder räumlich-zeitliche Erweiterungen davon [42, 56, 102, 108, 155, 215]. Manche Arbeiten integrieren auch Bewegungs- und Flussinformationen in den Deskriptor [102, 195] oder beschreiben die Verteilung an den Vordergrund angepasster geometrischer Primitive [81]. Auch die bei den silhouettenbasierten Merkmalen erwähnten Shape Context Deskriptoren können als eine Form lokaler strukturbasierter Deskriptoren betrachtet werden.

## 4.6 TEMPORALE REPRÄSENTATION UND INTEGRATION

Dynamische Gesten und Aktionen lassen sich nur schwer anhand statischer Einzelbilder klassifizieren, da auch ihre dynamische Entwicklung über einen gewissen Zeitraum informationstragend ist. Demzufolge muss diese Dynamik in geeigneter Weise repräsentiert werden. Die hier existierenden Methoden lassen sich einteilen in räumlich-zeitliche Repräsentationen, Einzelpostur-basierte Ansätze, dichte Flußfelder sowie Trajektorienanalyse.

### 4.6.1 *Räumlich-zeitliche Repräsentationen*

Räumlich-zeitliche (engl. *spatiotemporal*) Repräsentationen integrieren räumliche und temporale Informationen in einem einheitlichen Datenraum. Sie entstehen, indem eine zeitliche Folge von räumlichen Daten konkateniert und die resultierende Repräsentation als Einheit betrachtet wird. Im Falle eines Videos oder einer Bildsequenz entsteht durch die Konkatenation der (zweidimensionalen) Bilddaten ein dreidimensionales Volumen mit der Zeit bzw. dem Bildindex als dritter Koordinatenachse. Analog zur lokalen Merkmalsextraktion in rein räumlichen Daten können in einem solchen Volumen lokale Kuboide (Subvolumen) durch Merkmale beschrieben werden. Diese repräsentieren dann sowohl die Gestalt und Erscheinung der Bilddaten als auch deren zeitliche Entwicklung in einer einheitlichen Merkmalsrepräsentation. Insbesondere in der Aktionserkennung werden räumlich-zeitliche Volumen häufig verwendet [42, 56, 65, 69, 100, 102, 108, 140, 155]. Eine Sonderform dieses Ansatzes

stellt die Konkatenierung binärer Silhouetten dar. Der zeitliche Verlauf kann dann in einem Graustufenbild über den Grauwert kodiert werden, so dass die resultierende Repräsentation die gleiche Dimensionalität hat wie die Eingabedaten. Diese Form räumlich-zeitlicher Daten wird als Motion History Image [14] oder Motion History Volume [206] bezeichnet. In [103] wird eine Matrix aus konkatenierten eindimensionalen Silhouettendeskriptoren verwendet.

Neben der expliziten Konstruktion räumlich-zeitlicher Datenrepräsentationen gibt es auch Ansätze, die einen zeitlichen Kontext implizit durch (ggf. zeitlich gewichtete) Integration der Merkmale mehrerer aufeinanderfolgender Videobilder in einem gemeinsamen Deskriptor realisieren [81, 88, 195, 215]. In [55] wird ein räumlich-zeitliches Körpermodell vorgestellt, das neben kinematischen auch temporale Abhängigkeiten in einem festen Zeitfenster enthält.

### 4.6.2 *Einzelpostur-basierte Ansätze*

Im Gegensatz zum integrierten räumlich-zeitlichen Ansatz gehen Einzelpostur-basierte Verfahren zweistufig vor. Zunächst wird auf Einzelbildebene eine Beschreibung der Postur anhand statischer Merkmale extrahiert. Die einzelnen Ergebnisse aufeinanderfolgender Bilder werden anschließend zeitlich integriert.

Im einfachsten Fall wird jedem Einzelbild ein Symbol aus einer diskreten Symbolmenge zugewiesen. Die Integration kann dann durch einen Mehrheitsentscheid [81, 105], durch die Verwendung eines Histogrammes der relativen Symbolhäufigkeiten [151, 194] oder durch explizite Modellierung der zeitlichen Symbolfolgen geschehen [17, 33, 93, 115]. Dieser Vorgehensweise liegt die Annahme zugrunde, dass Aktionen sich durch eine relativ kleine Menge eindeutig unterscheidbarer statischer Schlüsselposturen beschreiben lassen, was offensichtlich voraussetzt, dass die Schlüsselposturen a priori bekannt sind. Aus diesem Grund verfolgen solche Ansätze auf Einzelbildebene eine Exemplar-basierte Erkennungsstrategie.

Der grundlegende Nachteil obiger Ansätze ist, dass bereits auf Einzelbildebene „harte“ Entscheidungen – im Sinne von Symbolzuordnungen – getroffen werden. Jeder Zuordnungsfehler beeinflusst also direkt die temporale Integration. Aus diesem Grund wurden sog. „Softvote“ Ansätze entwickelt, die mit kontinuierlichen Distanzwerten bzw. Zuordnungswahrscheinlichkeiten zu Symbolen oder Exemplaren arbeiten und diese über eine Bildsequenz akkumulieren [95, 134, 172, 173, 180]. Auch die explizite Modellierung zeitlicher Folgen von Posturen ist ohne „harte“ Entscheidungen möglich, indem probabilistische graphische Modelle verwendet werden, die statt auf Symbol- auf Merkmalsebene operieren [33, 130, 131, 135, 205].

### 4.6.3 *Dichte Flussfelder*

Flussfelder erfassen die Bewegungsrichtung und -geschwindigkeit von Pixeln oder kleinen Bildbereichen zwischen aufeinanderfolgenden Einzelbildern in einem Verschiebungsvektorfeld. Derartige Vektorfelder bezeichnet man als optischen Fluss (vgl. [51] Kapitel 9, [184] Kapitel 16). Sie enthalten sowohl globale Bewegung (wie z.B. Kamerabewegungen) als auch lokale Bewegungsmuster (z.B. einzelner Körperteile). Repräsentationen dichter Flussfelder werden z.B. in [3, 25, 113, 145, 172, 173, 195] als dynamische Merkmale benutzt. Ein Nachteil der Verwendung dichter Flussfelder ist, dass der optische Fluß sich nur in strukturierten Bildbereichen und an Objektkanten zuverlässig berechnen lässt, weil in unstrukturierten Bereichen keinerlei Informationen für die Ermittlung von Korrespondenzen zwischen aufeinanderfolgenden Bildern vorhanden sind. Deshalb werden in [84, 130] spärlichere Flußrepräsentationen benutzt, die nur an starken Intensitätsgradienten bzw. Silhouettenpunkten berechnet werden.

### 4.6.4 *Trajektorienanalyse*

Es ist leicht einzusehen, dass die im vorherigen Abschnitt erwähnten dichten Flussfelder viel redundante Informationen enthalten, weil die Flussvektoren benachbarter Pixel auf dem gleichen bewegten Objekt nahezu identisch sind. Daher ist es naheliegend, die Bewegungsanalyse nur auf die Flussvektoren einzelner repräsentativer Merkmalspunkte zu beschränken. Die Konkatenation der Flussvektoren - oder, analog, der räumlich-zeitlichen Koordinaten - eines Merkmalspunktes wird als (spatiotemporale) Trajektorie bezeichnet und beschreibt die Bewegungscharakteristik dieses Punktes über einen gewissen Zeitraum. Trajektorienbasierte Ansätze sind insbesondere im Bereich der (Hand-) Gesten- und Zeichensprache-Erkennung verbreitet [2, 3, 12, 17, 18, 20, 23, 25, 34, 35, 45, 77, 91, 104, 132, 143, 179, 203].

Trajektorien lassen sich jedoch nicht nur anhand der aufeinanderfolgenden Bildkoordinaten eines Punktes definieren. Prinzipiell ist jede temporale Folge von Merkmalsvektoren eine Trajektorie in einem entsprechend hochdimensionalen Raum. So kann beispielsweise die zeitliche Folge von Gelenkwinkeln eines Körpermodells [19, 175, 176, 204] oder eine Trajektorie von Posturen in einem Eigenraum [153, 178] analysiert werden

## 4.7 REPRÄSENTATION DES MENSCHLICHEN KÖRPERS

Die Analyse von Körperposturen oder -bewegungen setzt eine explizite oder implizite Modellierung des menschlichen Körpers voraus, auf deren Basis Konfigurationen von Körperteilen beschrieben werden können. Hier existieren Verfahren, welche die Körperstruktur sowie Beziehungen zwischen Körperteilen explizit modellieren oder probabilistische Modelle der Lagebeziehungen zwischen Körperteilen verwenden. Weiterhin sind Exemplar-basierte Methoden sowie modellfreie Ansätze zu nennen.

### 4.7.1 *Kinematische und anthropologische Körpermodelle*

Eine Möglichkeit, die Konfiguration der Körperteile während einer bestimmten Pose oder Geste zu beschreiben, besteht in der Verwendung eines parametrischen Körpermodells, welches neben der Erscheinung auch Lagebeziehungen und ggf. kinematische Beschränkungen zwischen den einzelnen Körperteilen modelliert. Bei der überwiegenden Mehrheit der Verfahren sind dies ungerichtete Graphen, deren Knoten die Körperteile und assoziierte Erscheinungsmodelle repräsentieren, und deren Kanten Beziehungen zwischen den Körperteilen definieren. Diese Beziehungen können entweder anhand von Trainingsdaten gelernt oder explizit anhand anthropologischer oder kinematischer Erkenntnisse formuliert werden. Der Vorteil dieser Modellierung besteht darin, dass rein ansichtsbasiert plausibel erscheinende Modellkonfigurationen, die anthropologisch nicht möglich sind, ausgeschlossen werden können. Demnach finden modellbasierte Ansätze eine Modellkonfiguration, die einerseits die Beobachtung gut erklärt, andererseits im Sinne der Modellbeziehungen plausibel ist. Auf diese Weise werden modellbasierte Verfahren in gewissen Grenzen robust gegen fehlerhafte Hypothesen für einzelne Körperteile, fehlende Körperteile oder Überdeckungen. Allerdings hängt die Qualität der Ergebnisse stark von der Angemessenheit des verwendeten Modells für das gegebene Szenario ab. Das Anpassen eines detaillierten Modells kann zudem aufgrund des hochdimensionalen Suchraumes sehr rechenaufwändig sein. Typische Ansätze dieser Art sind *Pictorial Structures* [11, 38, 39, 54, 55, 175, 176] oder *kinematische Bäume* bzw. *Skelettmodelle* [19, 112, 131, 183]. In [90] wird ein mit einem Skelettmodell verbundenes 3D-*Gittermodell* benutzt, um Ganzkörperposen mit 24 Freiheitsgraden zu tracken.

### 4.7.2 *Probabilistische Verteilungsmodelle*

Neben der anthropologisch motivierten Modellierung mit Körperteilen und Gelenken existieren auch schwächere Modelle, welche die relative Lage der Modellteile zueinander in allgemeinerer Form beschreiben. Die Struktur des Modells entspricht dann nicht notwendigerweise derjenigen des modellierten Körpers. Dazu gehören die konzeptionell sehr eng mit Pictorial Structures verwandten *probabilistischen Sterngraphen* [24, 56] und andere graphenbasierte Ansätze [154, 196]. Darüber hinaus existieren auch Repräsentationen, die keine explizite strukturelle Information in Form von Graphen verwenden, sondern Wahrscheinlichkeitsverteilungen über relative Positionen der Körperteile [24, 120, 132] oder probabilistische Modelle des Umrisses bzw. der Gestalt[5] [16, 27]. Derartige Modelle lernen zulässige Verformungen und Posen anhand von Trainingsdaten und repräsentieren somit gleichzeitig implizite Beschränkungen der möglichen Modellausprägungen.

Viele der im vorherigen Abschnitt genannten anthropologischen Modelle modellieren die Freiheitsgrade der Gelenke ebenfalls probabilistisch (z.B. lassen sich Pictorial Structures sowohl als Energieminimierungsproblem als auch in einem probabilistischen Rahmen formulieren [54]). Daher können sie ebenfalls zu den probabilistischen Verteilungsmodellen gezählt werden.

### 4.7.3 *Exemplar-basierte Methoden*

Die obigen modellbasierten Methoden ermöglichen überwiegend eine (quasi-) kontinuierliche Repräsentation der Pose (z.B. in Form von Gelenkwinkeln oder Lagebeziehungen). Im Gegensatz dazu existieren Ansätze, welche die Pose einer Person oder eines Körperteiles anhand eines Vergleiches mit einer Datenbank bekannter Posen [14, 17, 23, 42, 69, 78, 81, 84, 95, 97, 100, 105, 115, 130, 151, 155, 180, 190, 194, 195, 196, 205, 206, 215] oder durch Klassifikation ermitteln [28, 34, 102, 109, 141, 172]. Das Ergebnis dieser Herangehensweise ist demzufolge kein kontinuierlicher Posenvektor, sondern ein Symbol aus einer diskreten Symbolmenge. Die Inferenz der Modellpose wird somit zu einem Klassifikationsproblem. Für Anwendungen der Mensch-Maschine-Interaktion stellt dies keinen Nachteil dar, weil zur Interpretation einer Körperpostur oder Geste ohnehin eine Abbildung auf einen Satz von bekannten Kommandos vorgenommen werden muss. Zudem sind Exemplar-basierte Ansätze im Vergleich oft deutlich effizienter,

5 Das ist eine sehr subtile Unterscheidung und im Grunde nur eine Frage der Sichtweise: Stellt man die Körperteilmodelle als Knoten und die Beziehungen zwischen ihnen als Kanten dar, lassen sich die meisten hier genannten Ansätze als Graph visualisieren.

weil die zeitaufwändige Suche über den hochdimensionalen Konfigurationsraum eines Körpermodells entfällt.

### 4.7.4 *Modellfreie Ansätze*

Modellfreie Ansätze kommen ohne die explizite Modellierung von Körperteilen und kinematischen Beziehungen zwischen diesen aus. Im Gegensatz zu exemplar-basierten Methoden ist ihr Ergebnis aber nicht ein diskretes Symbol, sondern ein kontinuierlicher Posenvektor. In diesem Sinne handelt es sich hierbei also um hybride Ansätze, die zwar ggf. implizit ein Körpermodell verwenden, jedoch keinen expliziten Abgleich zwischen Modell, Bilddaten und im Modell kodierten anthropologischen oder kinematischen Beschränkungen vornehmen und auch ohne Exemplar-Modelle auskommen. Ein Beispiel hierfür ist die Verwendung (probabilistischer) Regressoren, welche erscheinungsbasierte Merkmale direkt auf einen Posenvektor abbilden [1, 72, 93, 134, 199]. Einige Autoren nehmen eine Anpassung eines aus geometrischen Primitiven zusammengesetzten Körpermodells [29, 122] oder eines Skelettmodells [211] an Bild- oder Voxeldaten vor . Hierbei wird im Gegensatz zu den weiter oben beschriebenen modellbasierten Verfahren die Anpassung rein datengetrieben realisiert, d.h. ohne Modellierung und Ausnutzung kinematischer Beschränkungen. In [126] werden Körperteile modellfrei durch Segmentierung von Silhouetten gefunden und durch 3D-Kombination der Ergebnisse mehrerer Ansichten verifiziert, während in [78] ein hybrider Ansatz realisiert wird, bei dem erst exemplar-basiert eine initiale Pose ermittelt und dann durch lokale Optimierung verfeinert wird. Eine weitere Möglichkeit besteht in der integrierten Repräsentation probabilistischer Erscheinungsmodelle und parametrischer Körpermodelle [16]. Die aus der Anpassung des Erscheinungsmodelles an die Bilddaten resultierende Deformation deformiert gleichzeitig das eingebettete Körpermodell und liefert so einen kontinuierlichen modellbasierten Posenvektor ohne explizite Auswertung eines Körpermodells.

## 4.8 ERKENNUNG

Erkennung stellt das Bindeglied zwischen Merkmalsrepräsentation, zeitlichen Beziehungen, Modellparametern und der Zielsetzung dar: Ihr Ziel ist die Integration und Interpretation aller vorhandenen Informationen zu einem durch die Intention des Verfahrens vorgegebenen Ergebnis. Das kann beispielsweise ein Gestensymbol im Falle der Gestenklassifikation sein, oder ein posenbeschreibender Parametervektor im Falle der Posenschätzung. Ansätze zur Erkennung lassen sich grob kategorisie-

ren in probabilistische Inferenz, Funktionsapproximation und Optimierung sowie Nächster-Nachbar- und Bag-of-Words-Ansätze. Nicht näher betrachtet werden heuristische Entscheidungsregeln [85, 91, 202], da diese speziell auf bestimmte Szenarien zugeschnitten und daher nicht allgemein einsetzbar sind.

### 4.8.1 *Probabilistische Inferenz*

Probabilistische Inferenzverfahren modellieren Beobachtungen und Ergebnisse als Zufallsvariablen, die einem zugrunde liegenden stochastischen Prozess folgen. Ihr Ziel ist die Beschreibung dieses Prozesses anhand von Trainingsbeispielen und daraus gelernter parametrischer probabilistischer Modelle. Hier lassen sich zwei prinzipielle Herangehensweisen unterscheiden [13]: *Diskriminative* Verfahren modellieren die a-posteriori Wahrscheinlichkeit einer Hypothese in Abhängigkeit der Beobachtungen, definieren also eine – im Allgemeinen nicht eindeutige – probabilistische Abbildung vom Beobachtungs- in den Ergebnisraum [24, 55, 134, 183, 199, 204]. Im Gegensatz dazu repräsentieren *generative* Modelle die Verbundwahrscheinlichkeit von Beobachtung und Hypothese durch klassenbedingte Dichten und a-priori Wahrscheinlichkeiten [19, 38, 39, 54, 56, 126]. Die Inferenz geschieht durch Maximierung der Beobachtungswahrscheinlichkeit, d.h. Suchen derjenigen Hypothese, welche die Beobachtungen am plausibelsten erklärt und gleichzeitig die Wahrscheinlichkeit des Modells maximiert. Beiden Vorgehensweisen ist der Vorteil gemein, dass durch die Modellierung als parametrische Wahrscheinlichkeitsverteilung „weiche“ Entscheidungen oder die Generierung mehrerer guter Hypothesen möglich sind. Daneben existieren auch Methoden, die beide Prinzipien kombinieren. So kann beispielsweise eine diskriminative Abbildung benutzt werden, um den Suchbereich im Ergebnisraum eines generativen Modelles einzuschränken [72].

Eine besondere Form der probabilistischen Inferenz, der aufgrund ihrer weiten Verbreitung und sehr erfolgreichen Anwendung insbesondere bei der Klassifikation von dynamischen Gesten und Aktionen an dieser Stelle besondere Aufmerksamkeit gewidmet wird, sind probabilistische graphische Sequenzmodelle. Sie gehören zu den generativen Ansätzen, betrachten aber neben Einzelergebnissen auch deren zeitliche Folge und berechnen eine a-posteriori-Wahrscheinlichkeit oder einen Gütewert auf Sequenzebene. Graphische Sequenzmodelle bestehen aus Knoten oder Zuständen, welche Einzelergebnisse repräsentieren. Dies können z.B. klassenbedingte Dichten über Merkmale eines einzelnen Kamerabildes oder Beobachtungspotentiale sein, aber auch Symbole einer höheren Ebene, z.B. einzelne Posen- oder Aktionsprimitive (vgl. Abschnitt 4.6). Die Knoten sind über Kanten verbunden, welche Übergangswahrschein-

lichkeiten oder -potentiale zwischen Knoten repräsentieren. Derartige Modelle sind also probabilistische Erweiterungen endlicher Automaten, wie sie in frühen Ansätzen zur Sequenzklassifikation [115] eingesetzt wurden. Die Inferenz geschieht durch Suche nach der Zustandssequenz, welche für eine gegebene Sequenz von Eingabemerkmalen oder -symbolen die Ausgabewahrscheinlichkeit für das jeweilige Modell maximiert. Die verbreitetesten Varianten dieses Prinzips sind *Markov-Ketten* [17, 19], *(Hidden) Markov Modelle* ((H)MM, vgl. Kapitel 5.6) [20, 23, 25, 33, 45, 49, 93, 104, 113, 132, 143, 178, 205] oder *(Hidden) Conditional Random Fields* ((H)CRF) [130, 135, 204].

### 4.8.2 *Funktionsapproximation und Klassifikation*

Viele Inferenzverfahren basieren nicht auf einem probabilistischen Framework, sondern auf Trennfunktionen oder deterministischen Abbildungen von Merkmalen auf Ergebnissymbole[6]. Im Allgemeinen spricht man in diesem Fall nicht von Inferenz, sondern von Klassifikation. Im einfachsten Fall ist eine Trennfunktion durch einen Schwellwert oder ein einfaches binäres Entscheidungskriterium über Merkmalen gegeben. Derartige Klassifikatoren sind extrem schnell, aufgrund der sehr reduzierten Komplexität des Entscheidungskriteriums aber auch unzuverlässig. Sehr erfolgreich sind jedoch sog. *Boosting*-Verfahren [109, 120, 141, 201], die mehrere einfache Klassifikatoren (*Schwache Klassifikatoren*) zu einem komplexeren und leistungsfähigen Gesamtklassifikator kombinieren. Ein klassisches Klassifikationsverfahren, das lineare Trennfunktionen in hochdimensionalen Merkmalsräumen verwendet, sind *Support Vektor Maschinen* (SVM) [22, 28, 102, 172, 173] oder Varianten davon [1, 140]. Im Unterschied dazu können neuronale Funktionsapproximatoren, wie z.B *künstliche Neuronale Netze* (KNN, vgl. Kap 5.5) [35, 137, 164], in der Theorie beliebig komplizierte funktionale Beziehungen bzw. Abbildungen zwischen Eingabe- und Ausgabesymbolen repräsentieren, benötigen dafür aber eine sehr große Zahl repräsentativer Trainingsbeipiele.

### 4.8.3 *Nächster Nachbar und Suchverfahren*

Die bisher genannten Methoden nehmen eine Inferenz aufgrund gelernter probabilistischer oder deterministischer Abbildungen vor. Im Gegensatz dazu verwenden *Nächster Nachbar* (NN) Verfahren [12, 23, 65, 78, 81, 88, 95, 97, 105, 120, 176, 180, 190, 195, 206] eine gespeicherte Datenbank repräsentativer Beispiele (*Prototypen*), um einem zu klassifizierenden Merkmalsvektor ein Symbol aus einer bekannten Symbolmenge zuzuordnen. Das Entscheidungskriterium ist ein geeignetes Distanz- oder Ähnlich-

6 In diesem Sinne können derartige Verfahren ebenfalls zu den diskriminativen Ansätzen gezählt werden.

keitsmaß zwischen der aktuellen Beobachtung und den gespeicherten Beispielen. Der nächste Nachbar wird im einfachsten Fall durch erschöpfende Suche über alle Prototypen gefunden. Durch *Suchbäume* [78, 131] lässt sich ein NN-Vergleich mit einer großen Menge von Beispielen effizienter realisieren. Die Qualität und Verläßlichkeit einer NN Suche hängt entscheidend von der Repräsentativität der gewählten Prototypen für die jeweilige Aufgabe und ihrer Robustheit in Bezug auf Variationen ab. Mit einer guten Auswahl an Prototypen und geeigneten, gegen bestimmte Variationen invarianten Merkmalen erreichen NN-Ansätze trotz ihrer Einfachheit oft Ergebnisse, die sehr wohl mit komplizierteren Methoden konkurrieren können.

Das Prinzip der NN-Suche lässt sich auch auf Beobachtungssequenzen unterschiedlicher Länge anwenden. In diesem Fall werden Beobachtungs- und Prototypsequenz bestmöglich zueinander ausgerichtet und ihre Ähnlichkeit durch die Kosten der Ausrichtung beschrieben. Dabei werden die Sequenzen ggf. nichtlinear „verformt“ (im Englischen als *warping* bezeichnet), so dass ihre Start- und Endpunkte übereinstimmen und alle dazwischen liegenden Datenpunkte gemäß ihrer geringsten Distanz paarweise zugeordnet werden. Der nächste Nachbar ist dann diejenige Prototypsequenz, die sich mit den geringsten Kosten an die Beobachtung anpassen lässt. Dieses Problem lässt sich durch *Dynamische Programmierung* (DP) lösen, die entsprechenden Verfahren werden demzufolge als *Dynamic (Space) Time Warping* (D(S)TW) bezeichnet [2, 3, 12, 81, 103, 180, 203].

### 4.8.4 *Optimierung*

Optimierungsverfahren sind implizit Bestandteil vieler Modellierungs- und Inferenzverfahren. An dieser Stelle soll unter Optimierung im engeren Sinne verstanden werden, dass die Inferenz als Suche nach Extremwerten in einer Gütefunktion oder a-posteriori Wahrscheinlichkeitsverteilung realisiert ist. Im Allgemeinen wird die Gütefunktion so gewählt, dass sie glatt und differenzierbar ist, so dass eine gezielte Suche mittels *Gradientenauf- oder Abstieg* möglich ist [78, 84, 100]. Gelegentlich werden auch *evolutionäre Algorithmen* [11, 211], lokale Suche [196] oder informationstheoretische Ansätze wie ein *Minimum Description Length* (MDL) Kriterium eingesetzt [85, 153].

### 4.8.5 *Bag-of-Words Modelle*

*Bag-of-Words* (BoW) Modelle (deutsch etwa „Sack von Wörtern“) stellen streng genommen keinen eigenen Erkennungs- oder Inferenzansatz dar. Ihre Besonderheit liegt in der Art, wie Merkmale berechnet und repräsentiert werden. Aufgrund ihrer in

den letzten Jahren – besonders im Bereich der Aktionserkennung – immer größer werdenden Bedeutung werden sie hier gesondert behandelt.

Ein BoW ist ein Metamerkmal, welches entweder verschiedenartige Merkmale oder Merkmale aus unterschiedlichen Zeitschritten in einer kompakten Histogrammform integriert. Die Zellen des Histogramms korrespondieren zu Symbolen oder Prototypen („Wörtern"), ihr Wert gibt die Häufigkeit des Auftretens des jeweiligen Symbols an. Somit gehen örtliche und zeitliche Beziehungen zwischen Symbolen verloren, was zugleich der größte Vor- und Nachteil der Methode ist: Einerseits geht potentiell wertvolle Information verloren, andererseits wird der Deskriptor robust gegen eine Vielzahl von Variationen, wie z.B. Anzahl der Merkmale, Länge einer Sequenz, unterschiedliche Start- und Endpunkte einer Sequenz, oder verschiedene Zeitbasen. Weitere Vorteile sind, dass sich Merkmale aus völlig unterschiedlichen Domänen in einfacher Weise kombinieren lassen und dass „weiche" Enscheidungen auf natürliche Weise durch nichtbinäre Inkremente der Histogrammzellen realisiert werden können.

Die eigentliche Erkennung eines BoW-Deskriptors erfolgt häufig NN-basiert [42, 100, 108, 151, 155, 194, 215]. Auch Klassifikatoren wurden erfolgreich eingesetzt [1, 81, 102, 134, 135, 172, 173].

## 4.9 AUSGEWÄHLTE ARBEITEN

Im Folgenden sollen einige Arbeiten detaillierter vorgestellt werden, die im Kontext dieser Dissertation von besonderem Interesse sind, weil sie ähnliche Ziele in einem vergleichbaren Szenario verfolgen oder interessante Modelle und Ansätze verwenden.

Eine Methode, die zwar die Erkennung von Zeichensprache-Gesten zum Ziel hat, dennoch aber interessante Ansätze zur allgemeinen Gestenerkennung aufzeigt, ist die Arbeit von Bowden et al. [17]. Hierbei werden einzelne Gesten und Posturen durch abstrakte linguistische Regeln beschrieben, die lediglich die Position der Hände relativ zu anderen Körperteilen und die Handform beinhalten. Zunächst werden der Kopf und die Hände anhand von Hautfarbe und eines Konturmodelles der Kopf-Schulter-Linie lokalisiert. Davon ausgehend werden Schlüsselpositionen auf dem Körper (z.B. die Schultern) durch gelernte Verteilungen ihrer relativen räumlichen Positionen gefunden. Handposturen werden durch Exemplarvergleich ihrer Silhouetten klassifiziert. Die so ermittelte Konfiguration des Oberkörpers wird anhand heuristischer Regeln in einen binären linguistischen Merkmalsvektor überführt. Dies resultiert in einer Beschreibung der Postur auf einer hohen Abstraktionsebene (z.B. "Die rechte Hand befindet sich auf Höhe des Kopfes, die linke bewegt sich abwärts auf Höhe der Schulter. Die Handposturen sind dabei X und Y."). Die Merkmalsvektor-Sequenzen werden anschließend

mit Markov-Ketten klassifiziert. Das Verfahren läuft in Echtzeit und erreicht sehr gute Ergebnisse auf einem Alphabet, welches 43 Zeichen der britischen Zeichensprache umfasst. Dies zeigt, dass bereits einfache Beziehungen zwischen wenigen Körperpunkten und eine sehr grobe Beschreibung ihrer Bewegung ausreichen, um komplexe Gesten mit hoher Genauigkeit zu erkennen. Das Problem der genauen Lokalisierung und der Erkennung unter verschiedenen Blickwinkeln wird allerdings durch Beschränkung auf ein sehr eingeschränktes Szenario gelöst, indem die Person frontal zur Kamera und in definiertem Abstand annähernd zentral im Bild stehen muss.

Den Prototypen eines Kamera-Projektor-Systems, mit dem sich bestimmte Oberflächen einer Umgebung in Touchscreens verwandeln lassen, stellen Wilson und Benko in [209] vor. Sie verwenden mehrere kalibrierte 3D-Kameras[7], um eine dreidimensionale Szenenrepräsentation zu erhalten. In dieser können anwesende Personen einfach durch 3D-Hintergrundsubtraktion detektiert werden. Die dichte 3D-Repräsentation erlaubt zudem die Berechnung von virtuellen zweidimensionalen Kameraansichten definierter Ebenen im Raum. Damit können spezialisierte Ansichten für bestimmte Interaktionstypen berechnet werden. Beispielsweise kann eine virtuelle Ansicht berechnet werden, welche nur eine Interaktionsoberfläche und ein geringes Volumen darüber enthält. Auf diese Weise sind die Detektion und das Tracking von Händen, die mit der Oberfläche interagieren, sehr einfach zu realisieren. Wilson und Benko benutzen diese Herangehensweise, um eine Reihe von einfachen Interaktionen mit projizierten grafischen Oberflächen zu realisieren. Diese Arbeit zeigt sehr eindrucksvoll, welche Möglichkeiten die neue Technologie der 3D-Kameras eröffnen kann. Allerdings bleibt auch mit 3D-Kameras das Problem bestehen, dass durch Vordergrundobjekte verdeckte Bereiche nicht erfassen werden können. Für eine echte 3D-Rekonstruktion der Szene werden trotzdem mehrere Kameras mit unterschiedlichen überlappenden Sichtfeldern benötigt. Die Kombination der verschiedenen 3D-Repräsentationen stellt dann – ähnlich wie im zweidimensionalen Fall – ein nichttriviales Problem dar. Zudem verlässt sich das vorgestellte Interaktionskonzept weiterhin auf WIMP-Oberflächen und ist zwar innovativ, aber zumindest teilweise wenig intuitiv zu bedienen.

Li und Greenspan [103] schlagen einen räumlich-zeitlichen Konturdeskriptor zur Erkennung von Armgesten vor. Nach einer Konturextraktion per Hintergrundsubtraktion und Konturverfolgung werden zu jedem Zeitpunkt eindimensionale Konturdeskriptoren berechnet, indem die Distanz der Konturpunkte zum Zentroid der Kontur aufgetragen wird. Die Kontur wird hierfür mit einer festen Anzahl von Punkten neu abgetastet und größennormalisiert, um Länge und Dynamik des Deskriptors zu

7 Die Kameras entsprechen denjenigen, die in Microsofts *Kinect*-Interface (zuvor Projekt Natal [121]) für die Xbox 360 zum Einsatz kommen und arbeiten mit strukturiertem infrarotem Licht.

normieren. Die zeitliche Konkatenation dieses Deskriptors für eine Gestensequenz stellt dann das Modell der Geste dar. Unterschiedliche Ausführungsgeschwindigkeiten können durch Interpolation der zeitlichen Achse des Deskriptors erfasst werden. Die Klassifikation erfolgt zweistufig: Zunächst werden Modellkandidaten anhand ihrer DTW-Distanz ausgewählt. Der beste Kandidat wird dann durch erschöpfende Suche anhand eines Entropie-basierten Distanzmaßes gefunden. Die Autoren erreichen sehr gute Ergebnisse für die Erkennung von acht verschiedenen Armgesten. Insbesondere ist das Verfahren in weiten Grenzen robust gegen Unterschiede in der Ausführungsgeschwindigkeit der Gesten. Zur Klassifikation wird allerdings die gesamte Gestensequenz (oder ein großer Teil davon) benötigt. Das Problem der Segmentierung bzw. des *Gesture Spottings* wird nicht behandelt. Zudem beschränkt sich das Szenario auf frontale monokulare Bildsequenzen mit statischem, einfachen Hintergrund.

Die große Mehrheit der Arbeiten zur Gestenerkennung verwendet Trajektorien von Punkten auf dem Körper der Person. Malgireddy und Kollegen [113] verwenden Histogramme der Orientierung des optischen Flusses und ein HMM-basiertes Klassifkationsframework zur Erkennung fünf verschiedener Handgesten. Dabei werden Gesten als Konkatenation atomarer Bewegungen (sog. Subgesten) modelliert. Die gleichzeitige Segmentierung eines kontinuierlichen Datenstroms und Erkennung der Segmente wird durch *Spotting-Modelle* erreicht. Hierbei werden unbekannte Gesten durch zufällige Folgen von Subgesten modelliert. Davon ausgehend wird ein sog. *Filler*-Modell und für jede Geste ein *Komplettierungs*-Modell konstruiert. Ist die a-posteriori-Wahrscheinlichkeit des Letzteren größer als die des Ersteren, wird angenommen, dass eine bekannte Geste beobachtet wurde und die Sequenz wird durch Rückverfolgung des Viterbi-Pfades klassifiziert. Dieses Vorgehen entspricht im Wesentlichen der verbreiteten Methode des *Nullmodelles* zur Modellierung unbekannter Beobachtungen. Interessant an dieser Arbeit ist insbesondere die Modellierung von Gesten als Folge von Subgesten. Die Autoren zeigen, dass dies die Erkennungsleistung im Vergleich mit einer holistischen Modellierung deutlich verbessert. Sowohl die Subgesten als auch ihre Zusammensetzung zu Gesten sind jedoch von Hand vorgegeben, was das Verfahren unflexibel macht. Zudem wird auch hier ein sehr eingeschränktes monokulares Szenario benutzt.

Auch Alon et al. [3] modellieren Subgesten, verfolgen jedoch einen etwas anderen Weg: Ausgehend von der Beobachtung, dass manche einfache und kurze Gesten Bestandteile komplexerer Gesten sind, werden derartige Subgesten-Relationen gezielt modelliert und automatisch aus Trainingsdaten gelernt. Nach einer auf Hautfarbe und Bewegung basierenden Handdetektion werden Trajektorienmerkmale (Position des Handzentroids und gemittelter optischer Fluß in der Handregion) berechnet. Als Gestenmodelle werden Markov-Modelle verwendet. Die Besonderheit ist, dass keine

perfekte Handdetektion angenommen wird, sondern davon ausgegangen wird, dass in jedem Bild eine Anzahl von Handhypothesen extrahiert wird. Die korrekte Hypothese wird durch Maximierung der Modellwahrscheinlichkeit mittels dynamischer Programmierung (DP) gefunden. Unwahrscheinliche Pfade werden dabei frühzeitig ausgeschlossen (*Pruning*), um eine kombinatorische Explosion zu vermeiden. Dieses Vorgehen realisiert eine gleichzeitige Segmentierung und Klassifikation der Merkmalssequenz. Mit dem gleichen Algorithmus werden Subgesten-Relationen identifiziert, indem geprüft wird, ob eine bestimmte Geste einen DP-Pfad mit hoher Wahrscheinlichkeit unter dem Modell einer anderen Geste aufweist. Die Entscheidung, ob eine bekannte Geste klassifiziert wurde, sowie die Behandlung von Subgesten-Relationen erfolgen anhand heuristischer Regeln. Diese Arbeit verfolgt einige zur vorliegenden Dissertation sehr ähnliche Ziele und Ansätze, beschränkt sich jedoch auf ein monokulares Szenario. Die Gestenerkennung ist daher nicht ansichtsinvariant und die Position der Person in Relation zur Kamera ist fest vorgegeben.

Bemerkenswerterweise existieren nur wenige Arbeiten, deren Ziel die Erkennung von Gesten im dreidimensionalen Raum ist. Kim und Kollegen [93] verwenden ein Multikamerasystem. Sie repräsentieren normalisierte 2D-Silhouetten und 3D-Gelenkpositionen mittels Selbstorganisierender Neuronaler Karten (*Self Organising Maps*, SOM) und lernen eine direkte Abbildung zwischen deren Knoten. Die Klassifikation erfolgt anhand der so errechneten 3D-Gelenktrajektorien in einem HMM-Framework. Das *Gesture Spotting* wird dabei durch den Vergleich der Ausgabewahrscheinlichkeiten des besten Gestenmodelles und eines Nullmodelles realisiert. Sie erkennen acht verschiedene ein- und zweiarmige Gesten und steuern damit die Vorhänge und Beleuchtung eines intelligenten Raumes. Die direkte Verwendung von Silhouetten erfordert jedoch eine sehr genaue Vordergrundsegmentierung, weshalb eine statische Umgebung mit sehr einfachem Hintergrund angenommen wird. Die präsentierten Beispiele und Ergebnisse lassen zudem darauf schließen, dass die Methode nicht ansichtsinvariant ist, weil die Abbildung von 2D nach 3D auf einer kleinen Anzahl von durch die SOM repräsentierten Schlüsselposturen basiert. Zudem werden die Silhouettenmerkmale der verschiedenen Kameraansichten in einer festen Reihenfolge zu einem Merkmalsvektor konkateniert, was die Flexibilität weiter einschränkt.

Im Gegensatz dazu verwenden Wang et al. [204] ein dreidimensionales zylindrisches Körpermodell, das mit einer Stereokamera getrackt wird. Die Gelenkkoordinaten und -winkel dieses Modelles bilden die Merkmale für die Gestenklassifikation mit *Hidden Conditional Random Fields* (HCRF). Die Autoren erreichen vielversprechende Ergebnisse auf einem Datensatz aus sechs beidhändigen Armgesten. Allerdings ist auch dieser Ansatz nicht ansichtsinvariant, weil das Tracking des Körpermodelles eine der Kamera

frontal zugewandte Person erfordert. Generell kann eine Stereokamera aufgrund des geringen Abstandes der beiden Bildebenen das Verdeckungsproblem nicht lösen.

Die Erkennung von Zeigegesten und die Schätzung der angezeigten Richtung im dreidimensionalen Raum ist das Ziel von Nickel und Stiefelhagen [132]. Zu diesem Zweck wird ebenfalls eine Stereokamera verwendet und das Gesicht sowie die Hände werden anhand einer kombinierten Disparitäts- und Hautfarbkarte gesucht. Das Hautfarbmodell wird dabei zunächst statisch initialisiert und zur Laufzeit mit aktuellen Beobachtungen adaptiert. Zeitlich aufeinanderfolgende Kopf- und Handpositionen werden in einem einfachen probabilistischen Multihypothesen-Trackingframework zu Trajektorien aggregiert. Zusätzlich wird die Orientierung des Kopfes durch ein KNN geschätzt, welches als Eingabe den normalisierten Kopfausschnitt des Bildes erhält. Das Auftreten einer Zeigegeste wird anhand der Handtrajektorie mit einem HMM-Klassifikator verifiziert. Der letzte Schritt umfasst die Schätzung der Zeigerichtung. Hier werden drei verschiedene Ansätze untersucht: Die Ausrichtung des Unterarmes, die Verbindungslinie zwischen Kopf- und Handposition („Sichtlinie“) sowie die Kopforientierung (unter der Annahme, dass die Person das angezeigte Ziel visuell fixiert). Die geschätzte Zeigerichtung wird zur Einweisung eines mobilen Roboters eingesetzt. Diese Arbeit ist eine der wenigen, die sich mit der expliziten Zeigerichtungsschätzung in 3D in einem realistischen Szenario beschäftigen. Die präsentierte Evaluierung ist jedoch leider sehr lückenhaft und gibt wenig Aufschluß über die tatsächliche Leistungsfähigkeit.

Chien und Kollegen [26] beschreiben ein System zur Zeigegestenerkennung in einem Multikamera-Szenario. Sie definieren die Zeigerichtung als Richtung der Hauptachse des Unterarmes. Armkandidaten werden durch ein Histogramm-Hautfarbmodell segmentiert. Ihre Positionen im dreidimensionalen Raum werden mit einem *Partikelfilter* getrackt. Die Auswertung des Beobachtungsmodelles geschieht durch Rückprojektion der Positionshypothesen in die Kameraansichten und Vergleich mit den extrahierten Vordergrundregionen. Die geschätzte Lage und Orientierung des Armkandidaten wird anschließend unter Ausnutzung der Epipolargeometrie zwischen verschiedenen Kameraansichten iterativ verfeinert. Während dieses Prozesses wird anhand einer Heuristik auch eine einfache Ansichtswahl vorgenommen, indem dasjenige Kamerapaar bestimmt wird, welches das beste Ergebnis liefert. Diese an sich sehr einfache Methode wurde erfolgreich eingesetzt, um acht in einem Büro verteilte Ziele zu referenzieren, wobei die Testpersonen sich frei im Sichtfeld der Kameras bewegen konnten. Die Autoren treffen aber verschiedene vereinfachende Annahmen, welche die reale Einsetzbarkeit stark einschränken. So erfordert die Armdetektion über Hautfarbe, dass die Unterarme der Person unbekleidet sind. Weiterhin wird angenommen, dass jede

erfolgreiche Detektion einer Zeigegeste entspricht, es erfolgt keine Gestenerkennung im eigentlichen Sinn und keine Rückweisung.

## 4.10 FAZIT

Die Vielzahl verschiedener Veröffentlichungen zeigt, dass die Analyse menschlicher Posturen und Bewegungen ein sehr aktives und relevantes Forschungsfeld ist. Die Entwicklung der letzten Jahre verschob den Fokus eher auf Aktions- und Aktivitätserkennung auf einem höheren Abstraktionsniveau, als auf die Erkennung von expliziten Kommandogesten. Die Gründe hierfür sind vielfältig: Die automatisierte Erkennung insbesondere ungewöhnlicher und aggressiver Verhaltensweisen in Überwachungsszenarien gewinnt im kommerziellen und politischen Umfeld seit Jahren immer mehr an Bedeutung. Die Verfügbarkeit großer, gut dokumentierter Datensätze erleichtert die Entwicklung und Evaluation sowie den Vergleich von entsprechenden Aktionserkennern. Nicht zuletzt legen die publizierten Ergebnisse den Schluß nahe, dass die holistische Erkennung von Ganzkörperaktionen einfacher und weniger fehleranfällig ist, als die Analyse von Körperteiltrajektorien, die mitunter eine genaue Lokalisierung relevanter Körperteile benötigt.

Im Bereich der Mensch-Maschine-Schnittstellen geht – möglicherweise ebenfalls aus kommerziellen Gründen – die aktuelle Entwicklung eher in die Richtung von Multitouch-fähigen WIMP-basierten Oberflächen und entsprechenden Eingabetechniken (vgl. z.B. [33, 76, 188, 209, 210]). Zwar ist auch hier eine rasante und interessante Entwicklung zu verzeichnen, ihre Einschränkungen hinsichtlich Intuitivität und Natürlichkeit bleiben dennoch bestehen. Aus diesem Grund ist die Entwicklung gestenbasierter berührungsloser Interfaces immer noch relevant, und das Problem ist noch weit davon entfernt, für uneingeschränkte Szenarien gelöst zu sein. Im Folgenden werden die verschiedenen in der Literatur zu findenden Ansätze hinsichtlich ihrer Anwendbarkeit für das vorliegende Szenario bewertet.

**Lokalisierung:** Bei der Lokalisierung sind prinzipiell alle Methoden denkbar, die effizient genug und hinreichend robust gegen Änderungen der Umgebungsbedingungen sind. Dazu zählen statistische adaptive Hinter-/Vordergrundmodelle, insbesondere sind Modelle der Hautfarbe für die Lokalisierung von Händen interessant. Genauso existiert mittlerweile eine Vielzahl effizienter ansichtsbasierter Detektoren. Hierbei ist allerdings darauf zu achten, dass das Erscheinungsbild der Person sich aufgrund des uneingeschränkten Szenarios (Ansichtsinvarianz), teilweiser Verdeckung und nicht zuletzt durch die Ausführung von Gesten stark verändern kann. Die Detektion einer Person als Ganzes erscheint somit nicht geeignet, sehr wohl jedoch die Detektion ein-

zelner Körperteile, wie z.B. des Kopfes oder Rumpfes. Lokalisierungsfreie Verfahren sind offensichtlich nur dann einsetzbar, wenn die absolute Position der Person in der Erkennung keine Rolle spielt. Dies ist zumindest für Zeigegesten nicht der Fall, weshalb diese Verfahrensgruppe ausscheidet.

**Merkmale:** Silhouettenbasierte Merkmale sind nur dann robust einsetzbar, wenn die segmentierten Vordergrundregionen von hoher Qualität sind. Das stellt wiederum hohe Anforderungen an das verwendete Hinter-/Vordergrundmodell. In einem natürlichen Szenario mit veränderlichen Umgebungsbedingungen kann nicht von einer perfekten Segmentierung ausgegangen werden, weshalb diese Merkmalsart nicht geeigent erscheint. Farb- und strukturbasierte Merkmale sind hingegen gut geeignet, weil sie ggf. auch in Grenzen gegen fehlerhafte Lokalisierung und affine Transformationen robust sind. Insbesondere sind hier kantenbasierte lokale Deskriptoren (z.B. HOG) zu nennen, die in einer Vielzahl von Aufgaben erfolgreich eingesetzt wurden.

**Zeitliche Integration:** Die vor allem in der Aktionserkennung sehr populären räumlich-zeitlichen Repräsentationen haben sich als sehr mächtig erwiesen, weil sie räumliche und temporale Zusammenhänge in einem kompakten Merkmal kodieren. Allerdings haben sie den Nachteil, dass zu ihrer Konstruktion eine längere Sequenz von Beobachtungen benötigt wird[8], was zu einer inhärenten Latenz führt. Dies ist für ein reaktives System unerwünscht. Sehr gute Ergebnisse wurden mit der Klassifikation von Trajektorien einzelner Körperteile und Flussfeldern erreicht. Die Berechnung von dichten Flußfeldern ist jedoch sehr rechenaufwändig und liefert eine große Datenmenge mit hoher Redundanz, so dass die Verwendung von Trajektorien einiger weniger Punkte geeigneter erscheint. Einzelpostur-basierte Verfahren sind ebenfalls potentiell interessant. Die publizierten Ergebnisse lassen vermuten, dass sich Körperbewegungen sehr gut als Folge von Schlüsselposturen repräsentieren lassen. Deren zuverlässige ansichtsbasierte Erkennung kann aber ein schwieriges Problem sein, wenn keine vereinfachenden Annahmen hinsichtlich der relativen Posen der Person und der Kamera getroffen werden. Viele Verfahren aus der Literatur, die dieses Prinzip verwenden, sind deshalb nicht oder nur in sehr engen Grenzen ansichtsinvariant.

**Körpermodellierung:** Die Verwendung eines Körpermodelles ist für die Gestenerkennung prinzipiell interessant, weil die zeitliche Folge der Modellparameter die menschliche Bewegung gut beschreibt. Grafische Körpermodelle haben jedoch typischerweise eine hohe Zahl von Freiheitsgraden, was ihre Lokalisierung sehr aufwändig macht. Zudem ergibt sich – auch bei anderen körpermodell-basierten Ansätzen – für das vorliegende Szenario ein schwieriges Problem bei der Integration der Ergebnisse

8 Viele der vorgestellten Ansätze segmentieren den Datenstrom überhaupt nicht, sondern berechnen einen Deskriptor für eine gesamte Aktionssequenz

mehrere Kameraansichten: Es wird von unsynchronisierten Kameras ausgegangen. Das bedeutet, dass die Kameras ihre Bilder zu unterschiedlichen Zeitpunkten aufnehmen. Damit wird einerseits eine voxelbasierte 3D-Rekonstruktion mit anschließender Anpassung eines 3D-Körpermodelles erschwert, weil mit großen Rekonstruktionsfehlern, Mehrdeutigkeiten und Inkonsistenzen zu rechnen ist. Andererseits wird bei unabhängiger Suche in den einzelnen Kamerabildern das ohnehin schon nichttriviale Problem, ggf. widersprüchliche Ergebnisse zu kombinieren, durch den zeitlichen Versatz noch schwieriger. Ohnehin stellt sich die Frage – angesichts der guten Ergebnisse, die bei der trajektorienbasierten Klassifikation schon mit sehr wenigen Punkttrajektorien erreicht werden – ob eine explizite Körpermodellierung für das vorliegende Problem überhaupt notwendig ist. Unbestritten kann jedoch eine Modellierung z.B. der relativen Körperteilpositionen dabei helfen, fehlerhafte Hypothesen zu eliminieren.

**Erkennung:** Die meisten der vorgestellten Erkennungsmethoden sind unabhängig von der konkreten Art der Daten und Merkmale. Daher sind sie prinzipiell alle für den Einsatz im angestrebten Gestenerkennungssystem geeignet, bzw. in verschiedenen Verarbeitungsphasen können unterschiedliche Erkennungsmethoden zum Einsatz kommen. Für die Aufgabe der Personenlokalisation bieten sich ansichtsbasierte Klassifikatoren an. Hier existieren mittlerweile sehr leistungsfähige und effiziente Verfahren. Im Bereich der Sequenzanalyse nehmen generative probabilistische Sequenzmodelle wie HMM in der Literatur eine herausragende Rolle ein. Hier existieren sowohl für das Training als auch für die Inferenz effiziente, theoretisch gut untersuchte Methoden. Zudem können derartige Modelle mit inkrementell anlaufenden Daten arbeiten und den Datenstrom gleichzeitig segmentieren und klassifizieren, weshalb dieser Ansatz für eine trajektorienbasierte Gestenerkennung besonders geeignet erscheint. Aber auch BoW-Modelle wurden sehr erfolgreich für die Sequenzanalyse eingesetzt und bilden somit – auch aufgrund ihrer Einfachheit – eine interessante Alternative.

# 5 METHODISCHE GRUNDLAGEN

Dieses Kapitel beschreibt verschiedene etablierte Methoden der Mustererkennung und *Computer Vision*, die im Rahmen dieser Arbeit zum Einsatz kommen oder zum Verständnis wichtig sind. Auf die konkrete Realisierung, Integration und Benutzung dieser Methoden im Rahmen des entwickelten Gestenerkennungssystems wird dann im nächsten Kapitel eingegangen.

## 5.1 SCALE INVARIANT FEATURE TRANSFORM

Die *Scale Invariant Feature Transform* (SIFT) [111] ist eines der verbreitetesten *Keypoint*-basierten Merkmalsextraktionsverfahren. SIFT kombiniert eine automatische *Keypoint*-Detektion mit einem gradientenbasierten lokalen Deskriptor, um Bildregionen zu repräsentieren. Der Deskriptor ist speziell entworfen, um bestimmte Eigenschaften aufzuweisen. So ist er z.B. invariant gegen Rotation, Skalierung, geringe Beleuchtungsänderungen und – in engen Grenzen – auch gegen affine Transformationen. Der zugehörige Klassifikationsansatz trifft Entscheidungen nicht aufgrund einzelner Merkmale, sondern Gruppen von Merkmalen, was ihn relativ robust gegen Ausreißer, Fehldetektionen und fehlende *Keypoints* macht.

### 5.1.1 *Keypoint-Detektion*

SIFT wurde ursprünglich zur Bildregistrierung und zur Detektion nicht verformbarer Objekte in Bildern entwickelt. Objekte bzw. Bildstrukturen sollen dabei unabhängig von ihrer Größe gefunden werden, d.h. der Deskriptor soll so weit wie möglich invariant sein gegenüber Größenänderungen der Bildregion, die er repräsentiert. Oder anders gesagt: Es wird gezielt nach Bildregionen gesucht, deren Merkmalsrepräsentation unabhängig von ihrer Skalierung ist. Dies wird erreicht durch die Verwendung eines sog. *Skalenraumes* (engl. *Scale Space*) [106].

Der Skalenraum ist eine kontinuierliche Funktion $L(x, y, \sigma)$, mit den Bildkoordinaten $x, y$ und der Skalierung $\sigma$. Er ergibt sich durch Faltung des Bildes $\mathbf{B}(x, y)$ mit Gauss'schen Glättungsfiltern $\mathcal{N}(x, y, \sigma)$ über alle möglichen Skalierungen $\sigma \in \mathbb{R}^+$. Somit umfasst der Skalenraum unendlich viele unterschiedlich stark geglättete Versionen

des ursprünglichen Bildes. Eine effiziente Suche nach stabilen Merkmalen ist möglich, indem die Differenz je zweier benachbarter, durch einen konstanten Skalierungsfaktor $s$ getrennter Skalenebenen berechnet und in den entstehenden Differenzbildern $\mathbf{D}_j(x, y), j = 1...n$ nach Extrempunkten gesucht wird (für eine Begründung bzw. einen Beweis der Richtigkeit siehe [111] bzw. [106]):

$$\begin{aligned}\mathbf{D}_j(x, y) &= (\mathcal{N}(x, y, \sigma_j) - \mathcal{N}(x, y, \sigma_{j-1})) * \mathbf{B}(x, y) \qquad (5.1)\\ &= L(x, y, \sigma_j) - L(x, y, \sigma_{j-1})\end{aligned}$$

$$\text{mit} \quad \sigma_i = s\sigma_{i-1}, \quad L(x, y, \sigma_i) = \mathcal{N}(x, y, \sigma_i) * \mathbf{B}(x, y) \qquad (5.2)$$

Somit müssen in der Praxis nur $n + 1$ Ebenen des Skalenraumes berechnet werden. Die Startskalierung $\sigma_0$, der Skalierungsfaktor $s$ sowie $n$ bzw. $\sigma_n = s^n\sigma_0$ sind dabei Parameter des Verfahrens. Es ergibt sich eine Variante einer Gauss'schen Auflösungspyramide, deren benachbarte Ebenen jeweils voneinander subtrahiert werden müssen. Das Ergebnis wird deshalb als *Difference of Gaussian* (DoG) Skalenraum bezeichnet.

*Keypoints* werden anhand ihrer Stabilität im DoG-Skalenraum identifiziert. Die lokalen Extrema der $\mathbf{D}_j(x, y)$ ergeben die initiale Menge der *Keypoint*-Kandidaten. Vor der eigentlichen Merkmalsberechnung werden Punkte mit geringem Kontrast innerhalb ihrer zugehörigen Bildregion verworfen, weil sie anfällig gegen Rauschen sind. Weiterhin werden *Keypoints* auf Intensitätskanten verworfen, da sie potentiell auf Objektgrenzen liegen und somit Hintergrundinformation kodieren könnten.

### 5.1.2 *Aufbau des Deskriptors*

Die Berechnung des SIFT-Deskriptors ist in Abbildung 3 dargestellt. Im Wesentlichen handelt es sich um ein in einem Fenster um den Keypoint herum berechnetes Gradientenorientierungshistogramm mit einer zusätzlichen Gauss'schen Gewichtung der Gradientenbeträge. Dabei wird jeder *Keypoint* annotiert mit seiner Position $\mathbf{x}$, der Skalierung $\sigma$, auf welcher er detektiert wurde, sowie seiner primären Orientierung $\rho$. Die primäre Orientierung korrespondiert zur dominanten Gradientenrichtung in der Deskriptorregion und wird anhand des globalen Gradientenhistogrammes bestimmt. Gibt es mehrere dominante Orientierungen, wird eine entsprechende Anzahl von Deskriptoren mit gleichem $\sigma$ und $\mathbf{x}$, aber unterschiedlichem $\rho$ generiert.

Für die Berechnung des eigentlichen Deskriptors wird die *Keypoint*-Region in $n$ nichtüberlappende rechteckige Bereiche eingeteilt, in denen jeweils ein Gradientenhistogramm $\mathbf{h}_i = (h_{ij}, j = 1...m), i = 1...n$, mit $m$ Bins berechnet wird. Der Merkmalsvektor besteht aus der Konkatenation aller Teilhistogramme und wird global normiert.

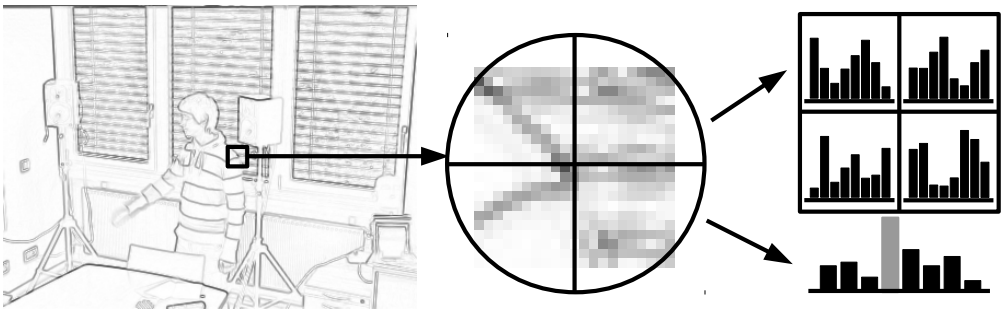

Abbildung 3: Konstruktion des SIFT-Deskriptors. Um eine detektierte *Keypoint*-Position herum wird ein rechteckiger Ausschnitt des Gradientenbildes extrahiert, dessen Größe proportional zur Skale des *Keypoints* ist. Über den Ausschnitt wird eine zweidimensionalen Gauss'sche Gewichtsfunktion gelegt, mit der die Gradientenbeträge multipliziert werden. Anschließend werden Gradientenhistogramme berechnet: Ein globales Histogramm (rechte Seite unten), anhand dessen die Orientierung $\rho$ des *Keypoints* bestimmt wird (grauer Balken), und mehrere lokale Histogramme (rechte Seite oben). Für letzteren Schritt wird der Bildbereich in nichtüberlappende Zellen aufgeteilt.

Durch die Histogrammbildung wird er robust gegen geringe Verschiebungen, während die Normalisierung Robustheit gegen globale Beleuchtungsänderungen bewirkt. Der finale Deskriptor ist ein Vektor $\mathbf{m} = (\{\mathbf{h}_k, k = 1...n \cdot m\}, \mathbf{x}, \sigma, \rho)$.

Die Invarianz gegenüber Skalierung und Rotation wird durch Normierung des Deskriptors relativ zu $\sigma$ und $\rho$ erreicht. Durch die Pose $(\mathbf{x}, \sigma, \rho)$ werden relative räumliche Beziehungen zwischen *Keypoints* im Deskriptor kodiert.

### 5.1.3 *Deskriptor-Klassifikation*

Die Klassifikation von SIFT-Deskriptoren besteht aus mehreren Schritten. In einer Trainingsphase wird eine Deskriptordatenbank aufgebaut, welche Deskriptoren der zu detektierenden Objekte bzw. Bildbereiche und deren assoziierte Posen enthält. Zunächst werden extrahierte Deskriptoren ähnlichen Beispielen aus der Datenbank zugeordnet. Diese Zuordnung ist i.d.R. nicht sehr zuverlässig und liefert eine hohe Zahl von Fehldetektionen und Fehlzuordnungen. Die Entscheidung, ob ein gesuchtes Objekt gefunden wurde oder nicht, wird daher durch eine Art Mehrheitsentscheid getroffen. Weil sowohl die Datenbank-Deskriptoren als auch die extrahierten Deskriptoren mit dem Posenvektor $(\mathbf{x}, \sigma, \rho)$ annotiert sind, lässt sich für jeden zugeordneten Deskriptor eine relative Pose berechnen. Wenn mehrere Deskriptoren das gleiche Objekt beschrei-

ben, sollte – unter der Annahme, dass das Objekt planar ist und sich nicht bzw. nur leicht verformen kann – die relative Pose aller dieser Keypoints annähernd gleich sein, weil sie eine affine Transformation des Objektes beschreibt. In [111] wird dieser Mehrheitsentscheid mittels einer generalisierten Hough-Transformation ([64] S. 733 ff.) mit anschließender iterativer Kleinste-Quadrate-Optimierung der Posenparameter (s. [111] für Details) realisiert. Dieses recht aufwändige Vorgehen liefert, neben dem Vorteil der höheren Robustheit und Zuverlässigkeit, zusätzlich für jedes gefundene Objekt eine Schätzung seiner Pose. Prinzipiell lassen sich mit diesem Ansatz auch die Parameter komplexerer Modelle als affine Transformationen planarer Objekte schätzen, etwa interne Parameter artikulierter Objekte. Dies würde jedoch eine große Zahl an Keypoint-Matches erfordern.

### 5.1.4 *Eigenschaften und Anwendungen*

SIFT hat sich als sehr robustes Verfahren zur Detektion von nicht oder wenig veränderlichen Strukturen in Bildern erwiesen. Die Invarianz der Merkmale gegen Skalierung, Rotation, Translation und (moderate) Beleuchtungsänderungen lässt es für viele verschiedene Einsatzzwecke als geeignet erscheinen, da es z.B. auch mit unterschiedlichen Kamerablickwinkeln oder teilweisen Verdeckungen eines Objektes bis zu einem gewissen Grad umgehen kann. SIFT und ähnliche Keypoint-basierte Verfahren finden Anwendung in den Bereichen Bildregistrierung [89, 111], Detektion bzw. Erkennung starrer Objekte [8, 110], automatische Kamerakalibration [107], 3D-Registrierung [41] sowie landmarkenbasierte Roboternavigation und -lokalisation [40, 192].

Nachteilig an SIFT ist, dass es nur für hinreichend strukturierte und kontraststarke Bilddaten zuverlässig funktioniert. Zudem ist die Berechnung des Skalenraumes, der eigentlichen Deskriptoren und insbesondere das Matching gegen große Datenbanken relativ rechenaufwändig. Es existieren allerdings Optimierungen und Verfahren, um die Berechnung wesentlich zu beschleunigen [8, 9, 75].

## 5.2 INTEGRALBILDER

In der Bildverarbeitung und videobasierten Mustererkennung werden häufig – sowohl bei der Merkmalsextraktion als auch in Vorverarbeitungsschritten, z.B. bei der Segmentierung – anstelle einzelner Pixel kleine Bildausschnitte betrachtet. Der Grund hierfür ist, dass die Farbwerte einzelner Pixel sehr rausch- und störungsanfällig sind, die Beschreibung einer kleinen Bildregion somit eine höhere Robustheit und Aussagekraft erwarten lässt. Typische Operationen sind z.B. die Berechnung durchschnittlicher

Farb- oder Grauwerte oder einfacher Merkmale, die auf blockweisen Pixeldifferenzen basieren. Derartige Operationen werden sehr ineffizient, wenn viele überlappende Bildbereiche betrachtet werden, was an einem einfachen Beispiel deutlich wird: Gegeben sei ein Grauwertbild der Größe $100 \times 100$ Pixel, über dem in einem gleitenden Fenster der Größe $10 \times 10$ mittlere Grauwerte berechnet werden sollen. An jeder Fensterposition müssen somit 100 Arrayzugriffe, 99 Additionen und eine Division durchgeführt werden. Innerhalb des Bildes existieren $(100 - 10 + 1)^2 = 8281$ mögliche Fensterpositionen. Für eine komplette Berechnung des gleitenden Mittelwertes sind also selbst bei dieser einfachen Situation mit einem Bild sehr geringer Auflösung 828100 Arrayzugriffe, 819819 Additionen und 8281 Divisionen notwendig. Es ist leicht einzusehen, dass mit größeren Bildern, mehreren Farbkanälen oder gleitenden Fenstern unterschiedlicher Größe eine derartige naive Berechnung ineffizient wird.

Für achsenparallele rechteckige Bildausschnitte existiert eine Möglichkeit, Pixelsummen innerhalb des Ausschnittes wesentlich effizienter zu berechnen, mittels eines sog. *Integralbildes*. Sei $\mathbf{B}(x, y)$ ein (einkanaliges) Bild. Der Wert des zugehörigen Integralbildes $\mathbf{I}(x, y)$ an der Position $(x_i, y_j)$ ergibt sich zu

$$\mathbf{I}(x_i, y_j) = \sum_{k=0}^{i} \sum_{l=0}^{j} \mathbf{B}(x_k, y_l). \tag{5.3}$$

Somit entspricht der Wert eines Pixels des Integralbildes der Summe der Farbwerte aller im Bild links oberhalb von ihm liegenden Pixel. Eine derartige Repräsentation lässt sich durch einmaliges Durchlaufen des Bildes berechnen (Abbildung 4). Durch Verwendung von Akkumulatorvariablen lässt sich ein Integralbild der Größe $m \times n$ mit $(2n - 1) \cdot m$ Arrayzugriffen und $2(m - 1)(n - 1) + m - 1$ Additionen berechnen.

Der Vorteil dieser Repräsentation liegt in der effizienten Berechnung von Pixelsummen, die insbesondere unabhängig von der Größe des betrachteten Bildausschnittes ist. Seien $(x_1, y_1), (x_2, y_2)$ zwei Punkte in Bildkoordinaten, die einen rechteckigen achsenparallelen Bildausschnitt definieren (Abbildung 4). Die Summe aller Pixel innerhalb des Ausschnittes ergibt sich zu

$$\mathbf{I}(x_2, y_2) + \mathbf{I}(x_1, y_1) - \mathbf{I}(x_1, y_2) - \mathbf{I}(x_2, y_1), \tag{5.4}$$

also mit vier Arrayzugriffen und drei Additionen. Dies ist offensichtlich unabhängig von den konkreten Bildkoordinaten der beiden Punkte.

In obigem Beispiel werden also zur Berechnung des gleitenden Mittelwertes $8281 \cdot 4 = 33124$ Arrayzugriffe und $8281 \cdot 3 = 24843$ Additionen benötigt. Dazu kommen die Kosten zur Berechnung des Integralbildes mit 19900 Arrayzugriffen und 19701 Additionen. Insgesamt spart die Verwendung des Integralbildes in diesem Beispiel also

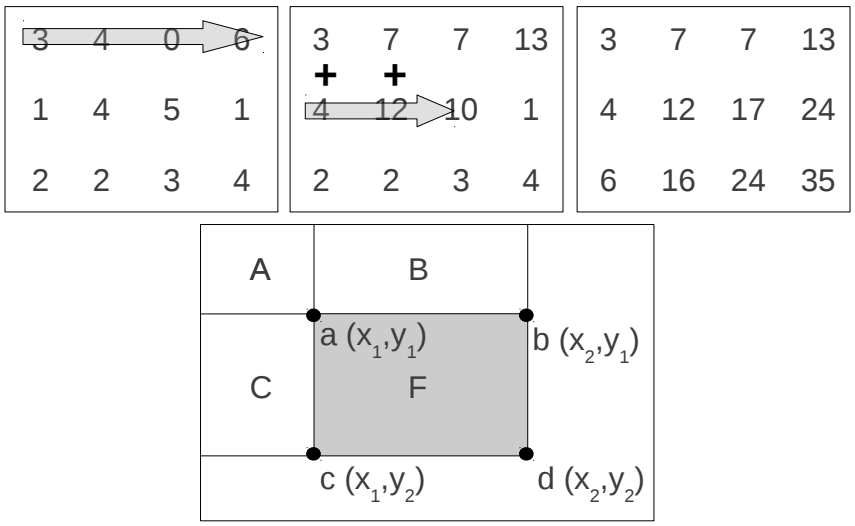


Abbildung 4: **Oben**: Berechnung eines Integralbildes. Ausgehend vom Ursprungsbild (links) werden pro Zeile kumulative Zeilensummen berechnet. Der Wert eines Pixels des Integralbildes ergibt sich dann aus der bisherigen kumulativen Summe über seine Zeile plus dem Wert seines oberen Nachbarn.
**Unten**: Die Pixelsumme im Bildausschnitt F lässt sich aus den Werten der vier Punkte $a, b, c, d$ wie folgt berechnen:
$\sum_F = \sum_{A \cup B \cup C \cup F} - \sum_{A \cup B} - \sum_{A \cup C} + \sum_A = d - b - c + a$.

mehr als 93% der Arrayzugriffe und 94% der Additionen ein. Diese Einsparung wird umso größer, je größer die Anzahl der betrachteten Bildausschnitte ist. Der Einsatz von Integralbildern an geeigneter Stelle stellt demnach eine sehr interessante Modifikation in Hinblick auf die Echtzeitfähigkeit eines Mustererkennungssystems dar, wie nicht zuletzt der Erfolg des populären Viola-Jones Detektors [201] zeigt.

## 5.3 HISTOGRAMS OF ORIENTED GRADIENTS

Bildgradienten werden häufig als Merkmale zur Objektdetektion benutzt, weil sie die Struktur oder Form eines Objektes unabhängig von dessen Farbe beschreiben. Ein auf Gradientenhistogrammen basierender Deskriptor wurde von Dalal und Triggs [37] vorgeschlagen, die sog. ***H**istograms of **O**riented **G**radients* (HOG, dt. Histogramme orientierter Gradienten). Im Folgenden werden zunächst die grundlegenden Prinzipien erläutert, gefolgt von einer detaillierten Beschreibung des HOG-Ansatzes.

### 5.3.1 *Gradientenhistogramme*

Gegeben sei ein Gradientenbild $\mathbf{G}(x, y) = \{\mathcal{M}(x, y), \phi(x, y), x = 1...w, y = 1...h\}$, mit der Breite $w$ und der Höhe $h$. Gesucht ist eine Repräsentation der Verteilung der Kantenorientierungen von $\mathbf{G}(x, y)$. Diese kann wie folgt durch ein Histogramm (vgl. Kapitel 2.3) beschrieben werden: Der Wertebereich der Kantenorientierung wird durch $m$ Bins diskretisiert. Da die Kantenrichtungsschätzung aufgrund der kleinen betrachteten Pixelumgebung i.d.R. ohnehin nicht sehr genau ist, reichen wenige Bins aus. Diese Anzahl kann weiter verringert werden, wenn statt dem Intervall $[-\pi, \pi]$ das Intervall $[0, \pi]$ zur Repräsentation der Orientierungen verwendet wird. Dies bedeutet, dass nur die tatsächliche Richtung der Kante von Interesse ist, nicht jedoch, ob der Übergang von dunkel nach hell oder umgekehrt erfolgt. In praktischen Anwendungen, etwa der Objektdetektion, ist dies häufig der Fall, weil es unerheblich ist, ob das Objekt heller oder dunkler als der Hintergrund ist.

Für jeden Pixel von $\mathbf{G}(x, y)$ wird nun anhand seiner Orientierung bestimmt, zu welchem Bin er beiträgt. Der Wert des entsprechenden Bins wird um den Gradientenbetrag des Pixels erhöht:

$$\mathbf{h} = (h_j), \quad h_j = \sum_{x=1}^{w} \sum_{y=1}^{h} \mathcal{M}(x, y)\delta(f_h(\phi(x, y)) - j). \tag{5.5}$$

Starke Kanten tragen also in höherem Maße zu den Werten des Histogrammes bei, als schwache. Auf diese Weise ergibt sich ein eindimensionales Histogramm der Kantenorientierungen, dessen Bins mit den Beträgen gewichtet sind. Man erhält somit eine extrem kompakte Darstellung der Kantenhäufigkeiten.

### 5.3.2 *Der HOG-Deskriptor*

Das soeben beschriebene einfache Gradientenhistogramm hat in praktischen Anwendungen nur wenig Nutzen, weil in einem Histogramm jegliche Information über örtliche Zusammenhänge zwischen den Gradienten verloren geht. Dies bedeutet, dass die tatsächliche Form des Objektes, welches durch das Histogramm repräsentiert wird, völlig unerheblich ist. Diese Eigenschaft ist für Objektdetektion und -klassifikation aus offensichtlichen Gründen unerwünscht.

Ziel des HOG-Verfahrens ist es, die Vorteile der kompakten Repräsentation in einem Histogramm mit Informationen über räumliche Zusammenhänge zu kombinieren. Der HOG-Deskriptor besteht deshalb aus mehreren konkatenierten Histogrammen, die jeweils kleine Bildausschnitte beschreiben. Innerhalb der einzelnen Bildausschnitte

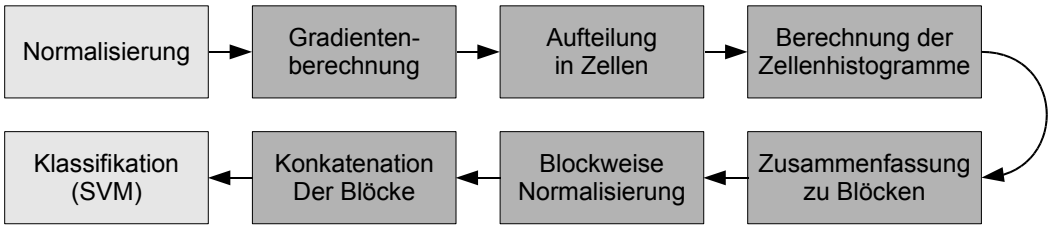


Abbildung 5: Prinzipieller Ablauf der Personendetektion mit HOG, nach [37]. Die hell hinterlegten Schritte sind nicht Teil der eigentlichen Deskriptorberechnung.

gehen die räumlichen Informationen zwar verloren. Die Verwendung mehrerer Teilhistogramme, die in einer festgelegten Reihenfolge konkateniert werden, erhält aber schwache räumliche Beziehungen der Bildausschnitte. Der Deskriptor ähnelt in seinem Aufbau somit dem zuvor vorgestellten SIFT-Deskriptor, geht aber noch einen Schritt weiter, indem Teilbereiche des Bildes zu überlappenden Blöcken zusammengefasst werden und der entstehende Deskriptor blockweise normiert wird.

Das Merkmalextraktionsverfahren operiert auf rechteckigen oder kreisförmigen Bildausschnitten und umfasst die folgenden Schritte (vgl. Abbildung 5 und 6):

- (Optional): Normalisierung des Bildes und Transformation in einen geeigneten Farbraum. Die Experimente in [37] lassen vermuten, dass die Wahl des Farbraumes nur einen sehr geringen Einfluss auf die Leistungsfähigkeit des HOG-Deskriptors hat, die Verwendung von Graustufenbildern jedoch zu einer Verschlechterung führt.

- Berechnung des Gradientenbildes. Interessanterweise führt eine vorherige Glättung des Bildes laut [37] zu einer signifikanten Verschlechterung der Ergebnisse. Ein möglicher Grund hierfür ist, dass eine Glättung nicht nur Rauschen, sondern auch kleine Details des Umrisses eliminiert.

- Unterteilung des Bildausschnittes in $m \times n$ achsenparallele Zellen und Berechnung des Gradientenhistogrammes für jede Zelle.

- Gruppierung benachbarter Zellenhistogramme zu überlappenden Blöcken von je $k \times l$ Zellen, dergestalt, dass die horizontale und vertikale Überlappung jeweils ein ganzzahliges Vielfaches der Zellengröße ist.

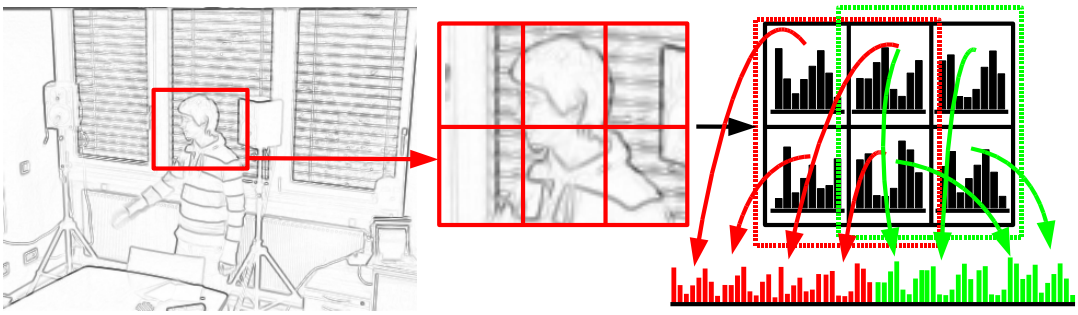

Abbildung 6: Berechnung eines HOG-Deskriptors. Der Bildausschnitt wird in Zellen eingeteilt. Innerhalb der Zellen werden Histogramme der Gradientenverteilung erstellt. Mehrere Zellen werden zu Blöcken zusammengefasst (hier: zwei Blöcke, grün und rot, die je $2 \times 2$ Zellen enthalten und sich horizontal um eine Zelle überlappen). Die Zellenhistogramme jedes Blocks werden in einer festen Reihenfolge aneinandergereiht und blockweise normalisiert. Die Aneinanderreihung der Deskriptoren aller Blöcke des Bildausschnittes bildet den HOG-Deskriptor.

- Lokale Kontrastnormalisierung für jeden Blockdeskriptor, um Robustheit gegen Beleuchtungsänderungen zu erreichen.
- Konkatenation der Blockdeskriptoren gemäß einer festen Reihenfolge zum finalen Merkmalsvektor.

Die Zusammenfassung der lokalen Gradientenhistogramme zu überlappenden Blöcken sorgt dafür, dass jede Zelle (mit Ausnahme der Zellen am Außenrand des Bildausschnittes) in mehreren Blöcken enthalten ist. Durch die blockweise Normalisierung wird jede Zelle mehrmals in verschiedenen räumlichen Kontexten betrachtet. Diese auf den ersten Blick redundante Vorgehensweise ist vermutlich einer der Hauptgründe für die guten Ergebnisse, die mit HOG-Merkmalen erzielt werden können.

Der so berechnete Deskriptor kann nun als Grundlage für weitere Verarbeitungsschritte dienen (z.B. Dimensionsreduktion per PCA) oder direkt klassifiziert werden. Dalal und Triggs klassifizieren den Deskriptor direkt mit einer ***S**upport **V**ektor **M**aschine* (SVM) und erreichen insgesamt sehr gute Ergebnisse.

Für den Einsatz zur Objektdetektion müssen HOG-Merkmale an allen möglichen Positionen im Bild in verschiedenen Skalierungen berechnet und klassifiziert werden. Dies lässt sich mit einem gleitenden Detektionsfenster über einer Auflösungspyramide realisieren. Der große Nachteil des HOG-Ansatzes besteht darin, dass die hochdimensionalen Deskriptoren nicht mit trivialen Klassifikatoren (z.B. einfachen

Schwellwert-Klassifikatoren) klassifiziert werden können, was zu einem erhöhten Rechenaufwand führt.

Ein weiterer potentieller Nachteil ist die große Zahl an Parametern. Die Zellgröße, die Anzahl der horizontalen und vertikalen Zellen pro Block, die Anzahl der horizontalen und vertikalen Blöcke pro Deskriptor, die Anzahl überlappender Zellen sowie die Anzahl der Orientierungsbins in den Zellenhistogrammen sind Parameter des Algorithmus, die entsprechend der Anwendung gewählt werden müssen. Somit ergibt sich eine sehr große Zahl möglicher Parameterkombinationen, was eine systematische Suche nach optimalen Parametern sehr aufwändig macht.

### 5.3.3 *Effiziente Berechnung mit Integralhistogrammen*

Die Objektdetektion mittels HOG-Merkmalen ist in der originalen Form für Echtzeitanwendungen zu rechenintensiv. Dies liegt, neben den benötigten relativ aufwändigen Klassifikatoren, die viele Male (für jedes gleitende Detektionsfenster) ausgewertet werden müssen, auch an der Berechnung des Deskriptors. In ihrer trivialen Form ist diese ineffizient, da zum Aufbau der Zellenhistogramme mehrmals über die lokalen Pixelumgebungen iteriert werden muss.

Zhu et al. [217] stellen einen Ansatz vor, der eine mögliche Lösung für beide Probleme enthält, indem HOG-Merkmale mit dem Detektionsverfahren von Viola und Jones [201] kombiniert werden. Dies umfasst eine Beschränkung der Deskriptor- und Klassifikatorkomplexität durch den Aufbau einer Detektorkaskade mittels *Boosting* sowie die Einführung einer Integralrepräsentation für Histogramme.

Integralbilder wurden in Kapitel 5.2 vorgestellt und es wurde erläutert, wie mit Hilfe dieser Repräsentation Pixelsummen in rechteckigen achsenparallelen Bereichen effizient berechnet werden können. Dieses Konzept lässt sich direkt auf (eindimensionale) Histogramme übertragen.

Sei $\mathbf{h} = (h_j, j = 1...m)$ ein Histogramm mit $m$ Bins, $\mathbf{G}(x, y)$ ein Gradientenbild und $f_h : \mathbb{R} \rightarrow [1, m]$ eine Funktion, welche die Gradientenorientierungen $\phi$ den Histogrammbins zuordnet. Das $k$-te Bin $h_k$ enthält die Summe der Beträge aller Gradienten, für deren Orientierung gilt $f_h(\phi) = k$. Betrachten wir nun eine Matrix $\mathbf{G}^k(x, y)$ welche nur die Gradientenbeträge $\mathcal{M}^k$ enthält, deren zugehöriges Orientierungsbin $k$ ist. Der Wert von $h_k$ für einen Bildausschnitt der Größe $i \times j$ ist die Summe über alle Pixelwerte innerhalb des Ausschnittes, d.h. $h_k = \sum_i \sum_j \mathbf{G}^k(x, y)$. Dies ist eine Pixelsumme in einem rechteckigen achsenparallelen Bildbereich, demzufolge lässt sich ein Integralbild zur effizienten Berechnung nutzen. Für jedes der $m$ Histogrammbins ist ein separates Integralbild erforderlich. Der Aufwand zur Berechnung eines

Integralhistogramms ist also $m$ mal so groß wie zur Berechnung eines einzelnen Integralbildes. Bei sehr großen Histogrammen wird dieser Aufwand inakzeptabel, im Falle der Gradientenorientierungs-Histogramme ist $m$ aber üblicherweise klein.

Ein Integralhistogramm kann also als Integralbild mit $m$ Kanälen angesehen werden. Analog zu einem Integralbild kann das Histogramm in einem rechteckigen Bildbereich unabhängig von der Größe des Bereiches mit drei Additionen pro Kanal, also mit insgesamt $4m$ Arrayzugriffen und $3m$ Additionen, berechnet werden.

### 5.3.4 *Eigenschaften und Anwendungen*

HOG-Merkmale haben sich als sehr mächtige Werkzeuge zur Beschreibung von Umrissen oder strukturierten Objekten erwiesen. Durch die Verwendung lokaler Histogramme sind sie weitgehend invariant gegen Rauschen, kleine lokale Deformation, geringe Translation (sofern diese klein ist im Vergleich zur Zellengröße) und Rotation (wenn der Rotationswinkel klein ist im Vergleich zur Orientierungsdiskretisierung durch die Histogrammbins). Sie sind nicht invariant gegen Skalierung, die Repräsentation als Integralhistogramm ermöglicht jedoch eine effiziente Berechnung in mehreren Skalierungsstufen. Durch die Normierung des Deskriptors ergibt sich eine hohe Robustheit gegenüber Kontrast- und Helligkeitsschwankungen. Der Deskriptor ist zudem unabhängig von der Farbe des Objektes, nicht jedoch von dessen Textur.

Aufgrund der guten Eigenschaften und des relativ einfachen Aufbaus erfreuen sich HOG-Merkmale in der Literatur großer Beliebtheit. Dalal und Triggs [37] sowie Ferrari et al. [55] verwenden sie zur Detektion aufrecht stehender oder laufender Personen. Thurau und Hlaváč [194] verwenden HOG-Deskriptoren, um Aktionsprimitive zur Aktionserkennung zu beschreiben. Laptev et al. [102] benutzen HOG-Merkmale und sog. *Histograms of Oriented Flow* (HOF), einen HOG-ähnlichen Deskriptor der Verteilung lokaler optischer Flussvektoren, um Aktionen in Videos zu lernen und zu identifizieren. Felzenszwalb et al. [53] benutzen Distanzen zwischen HOG-Deskriptoren als erscheinungsbasiertes Ähnlichkeitsmaß zur Anpassung eines Körpermodelles an Bildmaterial. Ikizler und Duygulu [81] erweitern den HOG-Ansatz, indem sie statt lokaler Kanten die Verteilung orientierter rechteckiger Patches durch einen HOG-ähnlichen Merkmalsvektor beschreiben. Diese sog. *Histograms of Oriented Rectangles* werden zur videobasierten Aktionserkennung eingesetzt.

Die starre Anordnung von Blöcken in einem regulären Raster wird in [217] durch Merkmalsselektion mittels *Boosting* ersetzt, wobei die Anordnung und Größe der HOG-Blöcke anhand eines Trainingsdatensatzes gelernt wird. Die Anwendbarkeit wird für einen Personendetektor demonstriert. Laptev [101] verfolgt einen ähnlichen Ansatz

und erreicht zusätzlich eine Geschwindigkeitssteigerung, indem die SVM durch *Fisher Linear Discriminant*-Klassifikatoren ersetzt wird. Der so gelernte Klassifikator wird zur Detektion von Personen und unterschiedlichen Objektklassen verwendet.

## 5.4 MEAN SHIFT

*Mean Shift* ist eine Technik der multivariaten Datenanalyse, die ursprünglich von Fukunaga und Hostetler [61] entwickelt wurde. Mit den Arbeiten von Comaniciu und Meer [30] gewann *Mean Shift* zunehmend an Popularität. Die Methode ist verwandt mit Verfahren der kernelbasierten Dichteschätzung. Sie findet auf effiziente Weise Modalwerte von Wahrscheinlichkeitsdichten und wurde erfolgreich in den Bereichen Segmentierung, Clustering und Tracking angewendet. Im Folgenden werden die theoretischen Grundlagen des Verfahrens vorgestellt. Anschließend wird gezeigt, wie sich damit effiziente Cluster- und Trackingverfahren realisieren lassen.

### 5.4.1 *Grundlagen*

Der folgende Abschnitt basiert größtenteils auf den Ausführungen von Comaniciu und Meer [30]. Wie bereits erwähnt, ist Mean Shift eng verwandt mit kernelbasierten Dichteschätzern. Formal ist ein *Kernel* (dt. Kern, in dieser Arbeit wird der gebräuchlichere englische Begriff verwendet) eine Abbildung $K : X \times X \to \mathbb{R}$ über dem Eingaberaum $X$ mit der Eigenschaft, dass ein Skalarproduktraum $(\Gamma, <.,.>)$ und eine Abbildung $\Phi : X \to \Gamma$ existieren, so dass $K(\mathbf{x}, \mathbf{y}) =< \Phi(\mathbf{x}), \Phi(\mathbf{y}) > \quad \forall \mathbf{y}, \mathbf{x} \in X$. $\Gamma$ wird als Merkmalsraum bezeichnet. Die Anwendung eines Kernels auf zwei Datenpunkte $\mathbf{x}$ und $\mathbf{y}$ entspricht also dem Skalarprodukt der zugehörigen Repräsentationen in $\Gamma$. Vereinfacht gesagt definiert ein Kernel ein skalares Ähnlichkeitsmaß zwischen zwei Datenpunkten, ohne dass eine explizite Berechnung der Transformation $\Phi$ vorgenommen werden muss. Kernelfunktionen haben große Bedeutung im Bereich des maschinellen Lernens oder als Faltungskerne in der Bildverarbeitung. Für eine ausführliche Einführung in Kernelmethoden sei auf [177] verwiesen.

Gegeben sei eine Stichprobe aus $n$ $d$-dimensionalen Datenpunkten $\{\mathbf{x}_i, i = 1...n\}, \mathbf{x}_i \in \mathbb{R}^d$, und ein Kernel $K(\mathbf{x}) =< \Phi(\mathbf{x}), \Phi(\mathbf{x}) >$. Der multivariate kernelbasierte Dichteschätzer am Punkt $\mathbf{x}$ ist dann

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_H(\mathbf{x} - \mathbf{x}_i), \quad K_H(\mathbf{x}) = \frac{1}{\sqrt{|\mathbf{H}|}} K\left(\mathbf{H}^{-\frac{1}{2}}\mathbf{x}\right). \tag{5.6}$$

Die symmetrische, positiv definite $d \times d$ Matrix $\mathbf{H}$ wird als Bandbreitenmatrix bezeichnet, $|\mathbf{H}|$ ist ihre Determinante. Diese Vorgehensweise berechnet eine Schätzung der Dichtefunktion $f(\mathbf{x})$, die durch die Stichprobe repräsentiert wird, mit der Kernelfunktion als Gewichtung.

Betrachten wir nun einen radialsymmetrischen Kernel, der sich durch Rotation einer eindimensionalen Funktion $k(\|\mathbf{x}\|^2)$, eines sog. Kernelprofils, ergibt:

$$K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2). \tag{5.7}$$

Das Profil $k(x)$ muss hierbei nur für $x \geqslant 0$ definiert sein. $c_{k,d}$ ist ein positiver Normalisierungsfaktor, der sicherstellt, dass $\int_{R^d} K(\mathbf{x}) d\mathbf{x} = 1$.

Zur Vereinfachung der Dichteschätzung wird häufig angenommen, dass $\mathbf{H}$ ein Vielfaches der Einheitsmatrix ist, d.h. $\mathbf{H} = h^2\mathbf{I}$. Damit ergibt sich der kernelbasierte Dichteschätzer zu

$$\hat{f}_{h,K}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right). \tag{5.8}$$

Bei der Analyse der Dichte $f(\mathbf{x})$ sind in der Regel die Modalwerte, d.h. lokale Maxima, von Interesse, die durch Nullstellen des Gradienten $\nabla f(\mathbf{x})$ gekennzeichnet sind. Da $f(\mathbf{x})$ unbekannt ist, ist eine exakte Berechnung nicht möglich, sondern es kann lediglich der Gradient der geschätzten Dichtefunktion $\hat{f}_{h,K}(\mathbf{x})$ berechnet werden. Dieser ist aber aufgrund der Linearität von Ausdruck (5.8) einfach durch die erste Ableitung des Dichteschätzers zu berechnen:

$$\nabla\hat{f}_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}) g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right), \tag{5.9}$$

mit $g(x) = -k'(x)$. Durch Ausmultiplizieren und Umstellen erhält man

$$\nabla\hat{f}_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right]. \tag{5.10}$$

Der erste Term entspricht der Dichteschätzung $\hat{f}_{h,G}(\mathbf{x})$ mit dem Kernel $G(\mathbf{x}) = c_{g,d} g(\|\mathbf{x}\|^2)$, der zweite Term wird als *Mean Shift Vektor* $m_{h,G}$ unter dem Kernel $G$ mit der Bandbreite $h$ bezeichnet,

$$m_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}, \tag{5.11}$$

und ist nichts anderes als die Differenz zwischen dem gewichteten Mittel mit dem Kernel als Gewichtsfunktion und dem Zentrum des Kernels. Setz man diese Terme in Ausdruck (5.10) ein, ergibt sich

$$\nabla \hat{f}_{h,K}(\mathbf{x}) = \hat{f}_{h,G}(\mathbf{x}) \frac{2c_{k,d}}{h^2 c_{g,d}} m_{h,G}(\mathbf{x})$$

$$\Rightarrow m_{h,G}(\mathbf{x}) = \frac{1}{2} h^2 c \frac{\nabla \hat{f}_{h,K}(\mathbf{x})}{\hat{f}_{h,G}(\mathbf{x})}. \tag{5.12}$$

Das bedeutet, dass der *Mean Shift* Vektor, berechnet mit Kernel $G$, bis auf eine Konstante dem normalisierten Gradienten der Dichteschätzung unter dem Kernel $K$ entspricht. Damit zeigt $m_{h,G}$ immer in Richtung des maximalen Anstieges der Dichtefunktion und der Betrag des Vektors ist normalisiert mit der Dichteschätzung mit Kernel $G$. Eine Verschiebung des Kernelzentrums in Richtung des *Mean Shift* Vektors führt also zu einem höheren Dichtewert. Modalwerte können demnach durch iterative Berechnung des *Mean Shift* Vektors und Verschiebung des Kernels bis zur Konvergenz gefunden werden.

Die Normalisierung sorgt dabei für eine automatische Adaption der Länge des Verschiebungsvektors, abhängig vom Wert der Dichteschätzung. Deshalb konvergiert der Mean Shift Algorithmus immer zu einem lokalen Maximum der Dichtefunktion (für den Beweis dieser Eigenschaft sei auf [30] verwiesen) und benötigt typischerweise nur wenige Iterationen. Die einzigen Parameter, welche das Verhalten des Algorithmus beeinflussen, sind die Profilfunktion $k(\|\mathbf{x}\|^2)$ und die Bandbreite $h$. Beides kann anwendungsabhängig gewählt werden. In [31] werden zudem zwei Möglichkeiten aufgezeigt, die Bandbreite automatisch anhand der beobachteten Daten zu wählen.

### 5.4.2 *Mean Shift Clustering*

Obiger Algorithmus kann zum Clustering von Datenpunkten verwendet werden. Im Gegensatz zu anderen Clusterverfahren (z.B. *k-Means*) muss die Anzahl der Cluster nicht vorab bekannt sein oder geschätzt werden. Weiterhin gibt es keine Beschränkung für die Form der Cluster. Das Ergebnis des Clustering-Vorgangs hängt allerdings von der Art und Bandbreite des gewählten Kernels ab.

Betrachten wir, wie im vorigen Abschnitt, $n$ $d$-dimensionale Datenpunkte $\{\mathbf{x}_i, i = 1...n\}, x_i \in R^d$. Die Aufgabe besteht darin, Cluster von Datenpunkten zu identifizieren. Cluster sind kompakte Bereiche des Datenraumes mit einer hohen Konzentration von Datenpunkten, d.h. einer hohen Datenpunktdichte. Unter dem oben beschriebenen Dichteschätzer ist das äquivalent zu einem hohen Dichtewert und die Modalwerte

**Algorithmus 1** Mean Shift Clusteralgorithmus

**Eingabe:** Datenpunkte $\{\mathbf{x}_i, i = 1...n\}$, Kernel $K_h(\mathbf{x})$ mit Bandbreite $h$, Abbruchkriterium $\epsilon$, Toleranz $\delta$.
$C = \{\}$, die Menge der Clusterzentren $\mathbf{c}_j$
$k = 0$
**for** j=1...n **do**
  Initialisiere *Mean Shift* Prozedur mit $\mathbf{x} = \mathbf{x}_j, m_{h,G}(\mathbf{x}) = \mathbf{0}$
  Berechne *Mean Shift* Vektor:
  $m_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}$
  **while** $\|m_{h,G}(\mathbf{x})\| > \epsilon$ **do**
    $\mathbf{x} = \mathbf{x} + m_{h,G}(\mathbf{x})$
    Berechne *Mean Shift* Vektor:
    $m_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}$
  **end while**
  **if** $\exists \mathbf{c}_l \in C : \quad \|(\mathbf{x} - \mathbf{c}_l)\| \leqslant \epsilon, \quad l = 1...k$ **then**
    $P_l = P_l \cup \mathbf{x}_j$
  **else**
    $k = k + 1, \quad \mathbf{c}_k = \mathbf{x}, \quad C = C \cup \mathbf{c}_k, \quad P_k = \{\mathbf{x}_j\}$
  **end if**
**end for**
**Ausgabe:** Menge der Clusterzentren $C = \{\mathbf{c}_j, j = 1...k\}$ und die zugehörigen Punktemengen $P_j$.

der Dichteschätzung entsprechen den Clusterzentren $\mathbf{c}_j$. Die Menge der zum Cluster j gehörenden Punkte $P_j$ besteht aus allen Datenpunkten, deren *Mean Shift* Prozedur zum Punkt $\mathbf{c}_j$ konvergiert (unter Beachtung einer Toleranz $\delta$). Damit lässt sich *Mean Shift* direkt zum Clustern einsetzen, wie in Algorithmus 1 dargestellt (vgl. *Mean Shift Segmentation* in [30]).

### 5.4.3 *Mean Shift Tracking*

Die vermutlich bekannteste Anwendung, in der *Mean Shift* Verwendung findet, ist das sog. *kernelbasierte Tracking* [32]. Die Grundidee besteht darin, das Trackingziel räumlich mit einem symmetrischen Kernel zu wichten. Dies erlaubt die Berechnung

einer glatten Ähnlichkeitsfunktion, deren Maxima mögliche Objektpositionen repräsentieren, die mit Gradientenaufstiegsverfahren gefunden werden können. In diesem Zusammenhang sind die Begriffe *Zielmodell* und *Kandidat* wichtig: Das Zielmodell ist eine Modellierung des Trackingzieles in einem geeigneten Merkmalsraum. Ein Kandidat ist eine Position oder ein Bildausschnitt in der korrespondierenden Merkmalsrepräsentation, für den entschieden werden soll, ob das Trackingziel sich dort befindet. Im Falle einer Darstellung des Zielmodelles und des Kandidaten als diskrete Dichteschätzungen lässt sich dieser Vorgang sehr einfach und elegant mit *Mean Shift* lösen. Die folgenden Ausführungen basieren größtenteils auf [32].

Das Trackingproblem wird formuliert als die Aufgabe, das Maximum der bedingten Wahrscheinlichkeit $P(\mathbf{x}|I(\mathbf{x}), \mathcal{M})$ mit der aktuellen Beobachtung $I(\mathbf{x})$ und dem Zielmodell $\mathcal{M}$ zu finden. $\mathcal{M}$ ist als eine Wahrscheinlichkeitsdichtefunktion in einem Merkmalsraum gegeben. Ein Kandidat $\mathcal{A}(\mathbf{y})$ an Position $\mathbf{y}$ ist ebenfalls durch eine Wahrscheinlichkeitsdichte $P(\mathbf{y})$ repräsentiert. Beide Dichtefunktionen sind unbekannt und sollen aus den Daten geschätzt werden.

Im Weiteren werden sowohl $\mathcal{M}$ als auch $\mathcal{A}(\mathbf{y})$ durch normalisierte Histogramme mit $n$ Bins repräsentiert:

$$\begin{aligned} \mathcal{M} &= (m_i, i = 1...n), \quad \sum_{i=1}^{n} m_i = 1, \\ \mathcal{A}(\mathbf{y}) &= (a_i(\mathbf{y}), i = 1...n), \quad \sum_{i=1}^{n} a_i(\mathbf{y}) = 1. \end{aligned} \tag{5.13}$$

In der Praxis hat sich farbbasiertes Tracking als sehr robust herausgestellt, d.h. der Merkmalsraum ist ein geeigneter Farbraum und $\mathcal{M}$ sowie $\mathcal{A}(\mathbf{y})$ sind Histogramme der Farbwertverteilung.

Sei $\hat{\rho}(\mathbf{y}) \equiv \rho[\mathcal{A}(\mathbf{y}), \mathcal{M}]$ eine Ähnlichkeitsfunktion zwischen $\mathcal{A}(\mathbf{y})$ und $\mathcal{M}$. Lokale Maxima dieser Funktion über alle Positionen $\mathbf{y}$ im aktuellen Bild $I(\mathbf{x})$ kennzeichnen mögliche Positionen des gesuchten Objektes. Weil bei der Verwendung von Histogrammen sämtliche Informationen über räumliche Zusammenhänge verloren gehen, weist die Funktion $\hat{\rho}(\mathbf{y})$ i.d.R. sehr starke Variationen und viele lokale Maxima auf, was die direkte Anwendung von Gradientenaufstiegsverfahren erschwert. Nach [32] kann dieses Problem behoben werden, wenn die Ähnlichkeitsfunktion mit einem symmetrischen Kernel $K(\mathbf{x})$ mit dem Profil $k(\mathbf{x})$ regularisiert wird. Wird o.B.d.A. ange-

nommen, dass $\mathcal{M}$ im Nullpunkt zentriert ist, ergibt sich die Wahrscheinlichkeit des $i$-ten Merkmals (Bin) somit als

$$m_i = c \sum_{j=1}^{l} k(\|\mathbf{x}_j\|^2)\delta(f(\mathbf{x}_j) - i). \tag{5.14}$$

Hierbei ist $\{\mathbf{x}_j\}, j = 1...l$ die Menge aller Pixel innerhalb der Region des Zielmodells, $f(\mathbf{x})$ ist eine Funktion, welche das Pixel $\mathbf{x}$ auf ein Bin des Histogrammes abbildet, und $c$ ist ein Normalisierungsfaktor, der sicherstellt, dass $\sum_{i=1}^{n} m_i = 1$.

Analog ergibt sich die Wahrscheinlichkeit von Merkmal $i$ des Kandidaten an Punkt $\mathbf{y}$ zu

$$a_i(\mathbf{y}) = c \sum_{j=1}^{l} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_j}{h}\right\|^2\right) \delta(f(\mathbf{x}_j) - i). \tag{5.15}$$

mit der Kernelbandbreite $h$.

Hierbei ist die einzige Anforderung an $K(\mathbf{x})$, dass sein Profil $k(\mathbf{x})$ konvex und mit steigendem Abstand vom Kernelzentrum monoton fallend ist. Der Effekt ist, dass $\hat{\rho}(\mathbf{y})$ geglättet wird und die Eigenschaften von $k(\mathbf{x})$ erbt, d.h. $\hat{\rho}(\mathbf{y})$ wird differenzierbar, wenn $k(\mathbf{x})$ differenzierbar ist. Somit sind effiziente gradientenbasierte Optimierungsverfahren anwendbar.

Die Ähnlichkeitsfunktion $\hat{\rho}(\mathbf{y})$ misst die Ähnlichkeit zweier Histogramme. In [32] wird folgende Metrik vorgeschlagen:

$$d(\mathbf{y}) = \sqrt{1 - \hat{\rho}(\mathbf{y})}, \quad \hat{\rho}(\mathbf{y}) = \sum_{i=1}^{n} \sqrt{a_i(\mathbf{y}) m_i}. \tag{5.16}$$

Sie ergibt sich aus dem Bhattacharyya-Koeffizienten und hat gegenüber anderen verbreiteten Ähnlichkeitsmaßen für diskrete Verteilungen – wie etwa die Kullback-Leibler-Divergenz oder die Chi-Quadrat-Distanz (vgl. z.B. [43, 99]) – den Vorteil, dass sie robust gegen leere Histogrammbins ist. Ihr Wertebereich ist außerdem auf $[0, 1]$ beschränkt, d.h. sie liefert eine normalisierte Distanz. Weil $d(\mathbf{y})$ ein Distanzmaß ist, ist das Trackingproblem äquivalent zur Minimierung von $d(\mathbf{y})$, bzw. zur Maximierung von

$\hat{\rho}(\mathbf{y})$. Durch Taylor-Reihenentwicklung von $\hat{\rho}(\mathbf{y})$ um den Punkt $a_i(\mathbf{y}_0)$ und Abbruch nach dem ersten Differenzialglied (d.h. lineare Approximation) ergibt sich

$$\begin{aligned}\hat{\rho}(\mathbf{y}_0) &\approx \sum_{i=1}^{n} \sqrt{a_i(\mathbf{y}_0) m_i} + \sum_{i=1}^{n} \left[ \frac{1}{2} \left( \frac{m_i}{\sqrt{a_i(\mathbf{y}_0) m_i}} \right) \cdot (a_i(\mathbf{y}) - a_i(\mathbf{y}_0)) \right] \\ &= \sum_{i=1}^{n} \sqrt{a_i(\mathbf{y}_0) m_i} + \frac{1}{2} \left[ \sum_{i=1}^{n} \left( \frac{a_i(\mathbf{y}) m_i}{\sqrt{a_i(\mathbf{y}_0) m_i}} - \frac{a_i(\mathbf{y}_0) m_i}{\sqrt{a_i(\mathbf{y}_0) m_i}} \right) \right] \\ &= \frac{1}{2} \sum_{i=1}^{n} \sqrt{a_i(\mathbf{y}_0) m_i} + \frac{1}{2} \sum_{i=1}^{n} a_i(\mathbf{y}) \sqrt{\frac{m_i}{a_i(\mathbf{y}_0)}}. \end{aligned} \tag{5.17}$$

Einsetzen von (5.15) führt zu

$$\hat{\rho}(\mathbf{y}) \approx \frac{1}{2} \sum_{i=1}^{n} \sqrt{a_i(\mathbf{y}_0) m_i} + \frac{C_h}{2} \sum_{j=1}^{l} w_j k \left( \left\| \frac{\mathbf{y} - \mathbf{x}_j}{h} \right\|^2 \right), \tag{5.18}$$

mit den Gewichten

$$w_j = \sum_{i=1}^{n} \sqrt{\frac{m_i}{a_i(\mathbf{y}_0)}} \delta(f(\mathbf{x}_j) - i). \tag{5.19}$$

In (5.18) ist der erste Term unabhängig von $\mathbf{y}$. Maximieren des zweiten Terms über die Bildpositionen $\mathbf{y}$ maximiert also $\hat{\rho}(\mathbf{y})$ und minimiert somit $d(\mathbf{y})$. Der zweite Term entspricht aber genau der Dichteschätzung mit dem Kernelprofil $k(\mathbf{x})$ (vgl. (5.8)) mit einer zusätzlichen Gewichtung $w_j$, die von der Ähnlichkeit des Kandidaten mit dem Zielmodell abhängt. Lokale Maxima dieser Dichteschätzung sind somit Lösungen des Trackingproblems, d.h. mögliche Objektpositionen. Aus Kapitel 5.4.1 ist bekannt, dass diese lokalen Maxima sich effizient mittels *Mean Shift* bestimmen lassen. Der *Mean Shift* Vektor ergibt sich analog zu (5.11) unter Einbeziehung der Gewichtung $w_j$ zu

$$m_{h,G}(\mathbf{y}) = \frac{\sum_{j=1}^{m} \mathbf{x}_j w_j g \left( \left\| \frac{\mathbf{y} - \mathbf{x}_j}{h} \right\|^2 \right)}{\sum_{j=1}^{m} w_j g \left( \left\| \frac{\mathbf{y} - \mathbf{x}_j}{h} \right\|^2 \right)} - \mathbf{y}, \tag{5.20}$$

und die neue Position $\mathbf{y}_1$ ausgehend von der initialen Schätzung $\mathbf{y}_0$ ist $\mathbf{y}_0 + m_{h,G}(\mathbf{y}_0)$.

Damit lässt sich das Trackingproblem wie in Algorithmus 2 formulieren. Im realen Einsatz lässt sich der Algorithmus noch deutlich vereinfachen (vgl. [32] Kapitel 4.2) und liefert einen echtzeitfähigen histogrammbasierten Tracker.

**Algorithmus 2** Iterativer Mean Shift Tracking Algorithmus.

**Eingabe:** Zielmodell $\mathcal{M} = \{m_i\}, i = 1...n$, Kernel $K(\mathbf{x})$, letzte Objektposition $\mathbf{y}_0$, Abbruchkriterium $\epsilon$
Initialisiere Position in aktuellem Bild mit $\mathbf{y}_0$.
Berechne Merkmalsrepräsentation $\mathcal{A}(\mathbf{y}_0) = \{a_i(\mathbf{y}_0)\}$
und $\hat{\rho}(\mathbf{y}_0) = \sum_{i=1}^{n} \sqrt{a_i(\mathbf{y}_0) m_i}$
**loop**
  Berechne Gewichte $w_j$ gemäß (5.19).
  Berechne Mean Shift Vektor gemäß (5.20) und neue Kandidatenposition $\mathbf{y}_1$.
  Berechne $\mathcal{A}(\mathbf{y}_1) = \{a_i(\mathbf{y}_1)\}$ und $\hat{\rho}(\mathbf{y}_1) = \sum_{i=1}^{n} \sqrt{a_i(\mathbf{y}_1) m_i}$
  **while** $\hat{\rho}(\mathbf{y}_1) < \hat{\rho}(\mathbf{y}_0)$ **do**
    $\mathbf{y}_1 \leftarrow 0.5(\mathbf{y}_0 + \mathbf{y}_1)$, berechne $\hat{\rho}(\mathbf{y}_1)$
  **end while**
  **if** $\|\mathbf{y}_1 - \mathbf{y}_0\| < \epsilon$ **then**
    **break**
  **else**
    $\mathbf{y}_0 \leftarrow \mathbf{y}_1$
  **end if**
**end loop**
**Ausgabe:** Neue Objektposition $\mathbf{y}_1$, Gütebewertung $\hat{\rho}(\mathbf{y}_1)$

Die originale Formulierung hat jedoch zwei wesentliche Nachteile: Erstens erfordert sie nach jeder Iteration des Algorithmus eine Neuberechnung des Kandidatenmodells $\mathcal{A}(\mathbf{y})$ und der Gewichte $w_j$. Zweitens ist das gewählte Distanzmaß integraler Bestandteil des Algorithmus und kann nicht ohne Weiteres gegen ein anderes Maß ausgetauscht werden. In der Praxis ist es deshalb häufig sinnvoller, auch das Tracking-Problem als Modalwertsuche in einer Wahrscheinlichkeitsdichte zu interpretieren. Die Dichte ist im Falle des Trackings in Bilddaten eine zweidimensionale diskrete reellwertige Funktion über den Bildkoordinaten, beispielsweise eine anhand eines anderen unabhängigen Modelles errechnete Gütekarte. Das Trackingproblem besteht dann im Finden eines Maximums der Dichte in der Nähe der letzten bekannten Objektposition. In diesem Fall entfällt in obigem Algorithmus die Berechnung der Gewichte $w_j$, $\mathcal{A}(\mathbf{y}_n)$ sowie $\hat{\rho}(\mathbf{y}_1)$ und die $w_j$ sind gegeben durch den Wert der zugrunde liegenden Gütekarte an der jeweiligen Pixelposition.

## 5.5 KÜNSTLICHE NEURONALE NETZE

Der Begriff Künstliches Neuronales Netz (KNN) beschreibt eine Klasse nichtlinearer diskriminativer Klassifikatoren. Ein KNN realisiert eine Abbildung eines $n$-dimensionalen Eingabevektors $\mathbf{e} = (e_i, i = 1...n)$ auf einen $m$-dimensionalen Ausgabevektor $\mathbf{o} = (o_j, j = 1...m)$, üblicherweise mit $n \neq m$. Es kann als ein Funktionsapproximator angesehen werden, der Entscheidungs- bzw. Diskriminanzfunktionen in einem eingebetteten Merkmalsraum definiert mit dem Ziel, die durch Trainingsbeispiele vorgegebene Abbildung von $\mathbf{e}$ auf $\mathbf{o}$ mit geringstmöglichem Fehler zu approximieren.

Das Prinzip eines KNN ist biologisch motiviert und realisiert ein – stark vereinfachtes – mathematisches Modell der neuronalen Informationsverarbeitung im menschlichen Gehirn. Es stellt ein sehr mächtiges Werkzeug zur Musteranalyse und -klassifikation dar. In der Tat wurde nachgewiesen, dass zumindest theoretisch beliebig komplizierte Entscheidungsfunktionen repräsentiert und somit beliebige Klassifikationsprobleme gelöst bzw. beliebige a-posteriori-Wahrscheinlichkeitsverteilungen nachgebildet werden können [43]. Die mathematische Formulierung ist zudem einfach und lässt sich algorithmisch effizient implementieren.

Im Folgenden werden kurz – soweit zum Verständnis des Prinzips notwendig – die biologischen Grundlagen neuronaler Informationsverarbeitung umrissen. Davon ausgehend wird eine häufig verwendete Form eines KNN, das sog. Feed-Forward-Netzwerk oder Multi-Layer Perzeptron, beschrieben. Abschließend wird ein verbreiteter Lernalgorithmus für KNN, das Error Backpropagation Learning, vorgestellt.

### 5.5.1 *Biologische Grundlage*

Das menschliche Gehirn stellt ein sehr mächtiges Informationsverarbeitungs- und Mustererkennungssystem dar. Es besteht aus vielen einzelnen Verarbeitungsknoten, den Neuronen, die für sich betrachtet weder besonders schnell noch besonders leistungsfähig sind. Die gewaltige Leistungsfähigkeit des Gehirns wird durch eine massiv parallele Informationsverarbeitung erreicht. Ein Mensch besitzt im Mittel ca. 14 Milliarden Neuronen [187], die durch Zellfortsätze (Synapsen) miteinander vernetzt sind. Jeweils eine dieser Verbindungen (Axon) dient zur Übertragung von Informationen zu anderen Zellen, die restlichen (Dendriten) dienen als Eingänge, die bioelektrische Signale anderer Zellen empfangen. Stark vereinfacht dargestellt (für Details s. z.B. [187]) besitzen Neuronen ein elektrisches Potential gegenüber der Umgebung, das durch Signale an den Synapsen verändert werden kann. Übersteigt dieses Potential eine Schwelle, wird das Soma angeregt und generiert bioelektrische Impulse, deren

Stärke und Frequenz von der Stärke der Zellerregung abhängen. Über die Synapsen werden diese Impulse an benachbarte Zellen weitergeleitet und können diese wiederum anregen. Auf diese Weise pflanzt die Information sich im Gehirn fort.

Die Wissensrepräsentation im menschlichen Gehirn ist also hochgradig parallel und verteilt. Sie wird sowohl durch die Neuronen als auch durch die Art und Stärke ihrer synaptischen Verbindungen repräsentiert. Das Gehirn beweist, dass eine derartige Informationsverarbeitungsstruktur enorm flexibel und leistungsfähig sein kann.

### 5.5.2 *Künstliche Neuronen*

Angesichts der Fähigkeiten des menschlichen Gehirns liegt der Gedanke nahe, seine Struktur und sein Prinzip der parallelen verteilten Informationsverarbeitung in einem Modell nachzubilden. Das Prinzip, viele sehr einfache und per se unzuverlässige Verarbeitungseinheiten zu einem viel leistungsfähigeren Konstrukt zu kombinieren, wird beispielsweise beim sog. Boosting (vgl. Kapitel 4.8.2) angewendet. Künstliche Neuronale Netze stellen einen Versuch dar, nicht nur das grundlegende Prinzip sondern auch die Struktur und Topologie eines menschlichen Gehirns zu modellieren. Eine exakte mathematische Beschreibung der Funktionsweise wäre allerdings für eine algorithmische Umsetzung viel zu komplex, und die parallele Verarbeitung von Milliarden von Neuronen übersteigt die Kapazität heutiger Rechensysteme bei Weitem. Deshalb können KNN nur ein stark vereinfachtes und verkleinertes Modell darstellen, welches sich aber trotzdem als sehr mächtig erwiesen hat.

Die Grundbausteine eines KNN sind die Verarbeitungsknoten, die analog zum biologischen Vorbild als Neuronen bezeichnet werden. Eines der ersten künstlichen, mathematisch wie neurophysiologisch fundierten, Neuronenmodelle stammt von McCulloch und Pitts [116]. Im Laufe der Zeit wurden viele Modifikationen und Änderungen dieses Modells vorgeschlagen. Abbildung 7 zeigt ein verbreitetes einfaches Neuronenmodell, das sog. formale statische Neuron. Seine Eingabe ist ein p-dimensionaler Vektor $\mathbf{x} = (x_k, k = 1...p)$. Das Neuron besteht aus dem Eingabegewichtsvektor $\mathbf{w} = (w_k, k = 1...p)$, der Aktivierungs- oder Übertragungsfunktion $z = f(\mathbf{x}, \mathbf{w}))$ und der Ausgabefunktion $y = g(z - \delta)$. Die Aktivierungsfunktion berechnet aus der Eingabe $\mathbf{x}$ und den Gewichten $\mathbf{w}$ einen skalaren Wert, die sog. Eingabeaktivierung $z$. Die Ausgabefunktion generiert daraus die skalare Ausgabeaktivierung (oder einfach Aktivierung) $y$. $\delta$ wird als Schwellwert (eng. *Bias*) bezeichnet und entspricht der Membranschwelle einer biologischen Nervenzelle. Ein Neuron definiert somit eine Abbildung $F : \mathbb{R}^p \rightarrow \mathbb{R}^1$. Die Form von $f(...)$ und $g(...)$ ist prinzipiell beliebig, daher ist $F : y = g(f(\mathbf{x}, \mathbf{w}) - \delta)$ im Allgemeinen eine nichtlineare Abbildung (auch

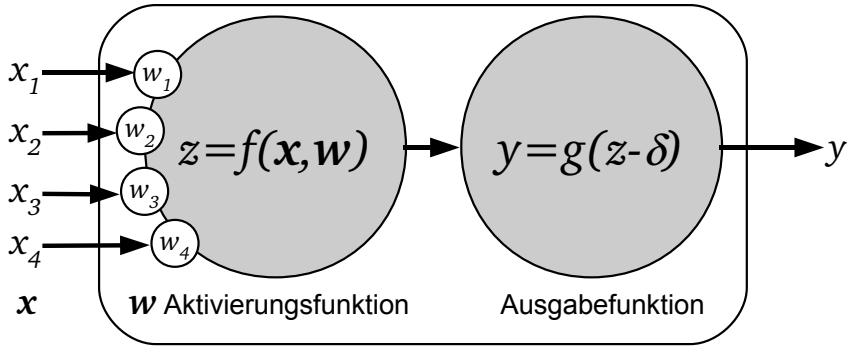


Abbildung 7: Das formale statische Neuron besteht aus einem Vektor der Eingabegewichte $\mathbf{w} = \{w_i\}, i = 1...k$ (hier: $k = 4$) sowie einer Aktivierungs- und Ausgabefunktion. Die Aktivierungsfunktion kombiniert den Eingabevektor **x** mit **w** zu einer skalaren Eingabeaktivierung $z$. Diese wird durch die Ausgabefunktion in die Ausgabeaktivierung $y$ überführt.

als *Transferfunktion* bezeichnet). Die Ausgabe einer kontinuierlichen Aktivierung stellt dabei bereits eine Verallgemeinerung des ursprünglichen McCulloch-Pitts Modelles dar, bei dem nur binäre Ausgaben vorgesehen sind.

### 5.5.3 *Das Perzeptron*

Kombiniert man mehrere Neuronen mit gleichem Eingabevektor in einer Netzstruktur, erhält man das sog. Perzeptron (Abbildung 8) [167]. Im Sinne einer konsistenten Darstellung werden gewöhnlich die Eingänge eines KNN ebenfalls als Neuronen dargestellt. Das Perzeptron erhält somit eine Eingabeschicht mit $k$ Neuronen. Diese haben je einen Ein- und Ausgang, die Transferfunktion ist die Identität $y = x$. Der Schwellwert $\delta$ lässt sich ebenfalls als zusätzliches Eingabeneuron (sog. Bias- oder On-Neuron) mit der Aktivierung 1 und dem Gewicht $-\delta$ repräsentieren. Die Ausgabeaktivierungen der letzten Schicht bilden den Ausgabevektor **o**, diese Schicht wird deshalb als Ausgabeschicht, ihre Neuronen als Ausgabeneuronen bezeichnet. Ein Perzeptron besteht immer mindestens aus Ein- und Ausgabeschicht. Besitzt es zudem noch weitere Zwischenschichten (sog. versteckte Schichten), so spricht man von einem mehrschichtigen Perzeptron (engl. Multi Layer Perceptron, MLP). Ein derartiges MLP stellt den „klassischen“ KNN-Klassifikator dar.

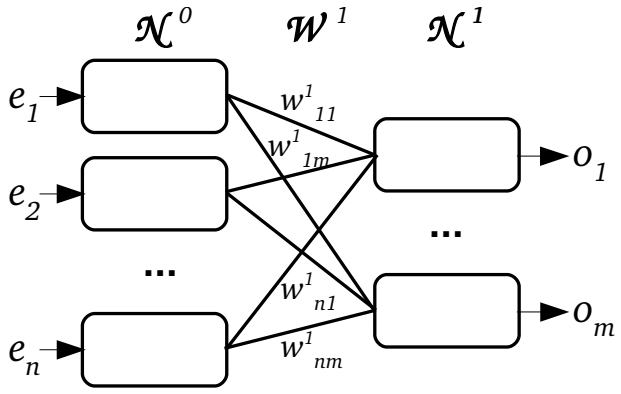


Abbildung 8: Ein einfaches Perzeptron, bestehend aus Ein- und Ausgabeschicht.

Sei $\mathbf{e} = (e_i, i = 1...n)$ ein $n$-dimensionaler Eingabevektor und $\mathbf{o} = (o_j, j = 1...m)$ ein $m$-dimensionaler Ergebnisvektor. Ferner sei das $l$-te Neuron einer Schicht $N_l$ gegeben durch seine Transferfunktionen:

$$N_l = \{f_l(\mathbf{x}, \mathbf{w}_l), g_l(z_l), \delta\}. \tag{5.21}$$

Die Menge der Neuronen in der $k$-ten Schicht des Perzeptrons $\mathcal{N}^k$ (wobei $k = 0$ im Folgenden die Eingabeschicht bezeichnet) ist demzufolge

$$\mathcal{N}^k = \{N_l^k\}. \tag{5.22}$$

Somit lässt sich ein Perzeptron $\mathcal{P}$ mit $q$ Schichten vollständig wie folgt beschreiben:

$$\mathcal{P} = \{\mathcal{N}^0, \{\mathcal{N}^k, \mathbf{W}^k\}, k = 1...q-1\}. \tag{5.23}$$

Hierbei ist $\mathbf{W}^k = (w_{ij}^k)$ die Matrix der interneuronalen Gewichte $w_{ij}^k$ für Eingang $i$ von Neuron $N_j^k$. Der Eingabevektor $\mathbf{x}^k$ von Neuron $N_j^k$ ist der Vektor der Ausgabeaktivierungen der vorherigen Schicht $\mathbf{y}^{k-1}$.

Die konkrete Form und Funktion eines MLP hängt entscheidend von der Art der Aktivierungs- und Ausgabefunktionen ab. Ein Überblick über alle verschiedenen Möglichkeiten würde an dieser Stelle zu weit führen, für einen Einstieg sei z.B. auf [43] verwiesen. Welche Aktivierungs- und Ausgabefunktion für ein bestimmtes Problem am besten geeignet ist, lässt sich a priori schwer sagen. Genauso existiert

kein bekanntes analytisches Verfahren, um die optimale Netzwerk-Topologie für ein gegebenes Problem zu ermitteln[1]. Weil MLP aber effizient und einfach trainierbar sind, können diese Parameter empirisch optimiert werden.

Neben dem hier vorgestellten Modell, das nur von vorwärts gerichteten interneuronalen Verbindungen zwischen benachbarten Schichten ausgeht (daher der Name *Feed-Forward-Netzwerk*) ist eine Vielzahl komplizierterer Netzwerktopologien denkbar (vgl. [43], Kapitel 6.10).

### 5.5.4 *Trainingsalgorithmen*

Wie oben bereits erwähnt, stellen KNN Modelle mit verteilter Wissensrepräsentation dar. Das im Netzwerk gespeicherte Wissen steckt zum größten Teil in den Gewichten der interneuronalen Verbindungen. Eine Festlegung der Gewichte „von Hand" ist nicht nur nicht wünschenswert, sondern für die meisten Probleme und alle nichttrivialen Netzwerke nicht möglich. Daher müssen sie durch einen Trainingsalgorithmus datengetrieben gemäß einer geeigneten Fehlerfunktion optimiert werden.

Der bekannteste Trainingsalgorithmus ist das sog. *Error Backpropagation Learning* ([43], Kapitel 6.3.). Er gehört zur Klasse der überwachten Lernalgorithmen, d.h. er benötigt eine annotierte Trainingsstichprobe $\mathcal{M} = \{\mathbf{e}_l, \mathbf{t}_l, l = 1...n\}$ mit $n$ Mustern oder Merkmalsvektoren $\mathbf{e}_l$ und den zugehörigen gewünschten Netzwerk-Ausgabeaktivierungsvektoren $\mathbf{t}_l$. $\mathbf{t}_l$ wird auch als *Feedback-Vektor* bezeichnet. Im Folgenden wird zunächst ein zweischichtiges Netzwerk (ohne versteckte Schichten) betrachtet. Der quadratische Ausgabefehler $J(\mathbf{w})$ zwischen der durch das Netzwerk nach Präsentation der Eingabe $\mathbf{e}$ generierten Ausgabeaktivierung $\mathbf{o}$ und dem zugehörigen Feedbackvektor $\mathbf{t}$ (der Stichprobenindex wird im Folgenden zu Gunsten besserer Lesbarkeit vernachlässigt) ist gegeben durch die Fehlerfunktion

$$J(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^{m} (t_k - o_k)^2. \tag{5.24}$$

$J(\mathbf{w})$ ist dabei implizit abhängig vom Gewichtsvektor $\mathbf{w}$ der Ausgabeschicht. Die Grundidee des *Error Backpropagation* Algorithmus besteht in einer iterativen Anpassung der Netzwerkgewichte

$$\mathbf{w}(c+1) = \mathbf{w}(c) + \Delta\mathbf{w}(c), \tag{5.25}$$

1 Verfahren wie Cascade Correlation ([43], S.329) konstruieren automatisch minimale Netze für ein gegebenes Problem. Die Struktur dieser Netze entspricht aber nicht mehr einem MLP und garantiert keine Optimalität.

dergestalt, dass sich der Gesamtfehler in jeder Iteration verringert. Dies kann durch einen Gradientenabstieg über $J(\mathbf{w})$ erreicht werden:

$$\Delta\mathbf{w} = -\eta\frac{\delta J(\mathbf{w})}{\delta\mathbf{w}}. \tag{5.26}$$

Hierbei bezeichnet $\eta$ die sog. Lernrate, die zur Schrittweite des Gradientenabstieges korrespondiert. Im einfachsten Fall eines skalarproduktaktivierten[2] Netzes ohne Bias-Neuronen und mit $m$ Ausgabeneuronen mit identischer Ausgabefunktion $g(z)$ führt dies zu folgender Lernregel (für Details zur Herleitung siehe [43], Kapitel 6.3.):

$$\Delta w_{jk} = \eta\epsilon_k x_j = \eta(t_k - o_k)g'(z_k)x_j, \tag{5.27}$$

wobei $x_j$ die j-te Komponente des Eingabevektors von Neuron $N_k$ ist. Im Falle eines zweischichtigen Netzwerkes ist dieser Eingabevektor identisch mit der Netzwerkeingabe, es gilt also $x_j = e_j$. $\epsilon_k$ ist die sog. *Sensitivität* des Neurons $N_k$. Sie besagt, wie der Gesamtfehler von der internen Aktivierung eines Neurons abhängt, beschreibt also den Einfluß des Neurons auf die Fehlerfunktion. Hierbei wird angenommen, dass die Ausgabefunktion $g(z)$ differenzierbar ist[3].

Diese sehr einfache Lernregel bildet die Grundlage aller Varianten des *Error Backpropagation* Lernalgorithmus. Sie lässt sich für eine Vielzahl von Variationen der Netzwerkstruktur in ähnlicher Form herleiten. Insbesondere sind die Beschränkung einer identischen Ausgabefunktion $g(z)$ und die Annahme einer identischen Lernrate für alle Neuronen des Netzwerkes nicht notwendig.

Weiterhin lässt sich das Prinzip auf Netzwerke mit einer oder mehreren versteckten Schichten erweitern. Die obige Lernregel (5.27) kann direkt für das Training der Gewichte zwischen der letzten versteckten Schicht und der Ausgabeschicht verwendet werden. Für die Gewichte zwischen der vorletzten und letzten versteckten Schicht ergibt sich analog zu (5.27) ([43], Kapitel 6.3.):

$$\Delta w_{ij} = \eta\epsilon_j x_i = \eta g'(z_j)\sum_{k=1}^{m} w_{jk}\epsilon_k x_i. \tag{5.28}$$

Hieraus wird auch ersichtlich, dass die Netzwerkgewichte nicht mit Null initialisiert werden dürfen, weil $\Delta w_{ij}$ dann immer Null wäre. In der Regel werden die Gewichte eines MLP daher zufällig mit Werten ungleich Null initialisiert. Die Sensitivität $\epsilon_j$ in obiger Formel hängt von den Sensitivitäten $\epsilon_k$ der Neuronen der nachfolgenden

2 D.h. die Eingabeaktivierung ist das Skalarprodukt von Eingabevektor und Gewichtsvektor.

3 Aus diesem Grund werden als Ausgabefunktion zumeist differenzierbare monoton zunehmende Funktionen gewählt.

Netzwerkschicht ab, ist also rekursiv definiert. Für Netze mit mehr als einer versteckten Schicht ergeben sich entsprechend komplexere Ausdrücke für die Sensitivitäten der früher im Netz liegenden Schichten, die sich aber immer rekursiv mit den Sensitivitäten der nachfolgenden Schicht berechnen lassen. Daher der Name *Error Backpropagation*: Der Ausgabefehler muss, beginnend bei den Ausgabeneuronen, schrittweise rückwärts durch das Netz propagiert werden.

Es existieren viele verschiedene Varianten und Erweiterungen dieser ursprünglichen Version des Lernalgorithmus, für Details sei z.B. auf [43], Kapitel 6.3.2 ff. verwiesen.

## 5.6 HIDDEN MARKOV MODELLE

Hidden Markov Modelle (HMM) sind generative probabilistische grafische Modelle, die in der Sequenz- oder Zeitreihenanalyse mit großem Erfolg eingesetzt werden (vgl. Kapitel 4.8.1). Sie können als probabilistische endliche Automaten betrachtet werden, wobei jeder Zustand des Automaten eine Ausgabe generiert. Es wird angenommen, dass die Folge von Ausgaben eines HMM beobachtbar ist, die zugrunde liegende Zustandsfolge, die sie generiert hat, jedoch unbekannt ist. Das eigentliche Modell ist also „versteckt" (engl. *hidden*). Weiterhin wird angenommen, dass die Markov-Annahme erfüllt ist, d.h. dass die statistischen Eigenschaften des Modells von einer endlichen zeitlichen Historie abhängen. Daher der Name Hidden Markov Modell. Im Folgenden werden, soweit für das Verständnis des Prinzips nötig, eine Beschreibung des Modells sowie gängiger Trainings- und Inferenzalgorithmen präsentiert. Für eine erschöpfende Behandlung der theoretischen Hintergründe sei z.B. auf [57] verwiesen.

### 5.6.1 *Definition*

Sei $\mathcal{X} = \{\mathbf{x}_t, t = 1...n\}, \quad \mathbf{x}_t \in \mathbb{R}^d$, eine geordnete Folge d-dimensionaler Beobachtungen. Dies können rohe Messwerte, Merkmale oder auch abstrakte Symbole sein. Die einzelnen Beobachtungen $\mathbf{x}_t$ werden als multivariate Zufallsvariablen betrachtet, die einem zugrunde liegenden kausalen stochastischen Prozess entspringen. D.h. ihre Entstehung folgt einer – üblicherweise unbekannten – multivariaten Wahrscheinlichkeitsdichte $p(\mathbf{x}_t|x_1, x_2, ...x_{t-1})$ über dem Beobachtungsraum. Nimmt man weiterhin an, dass der stochastische Prozess verschiedene Zustände annehmen kann, deren Übergänge ebenfalls einem stochastischen Prozess folgen (d.h. der Prozess ist zweistufig), dann hängt zu jedem Zeitpunkt $t$ der aktuelle Zustand $S_t$ von allen Vorgängerzuständen ab. Die Wahrscheinlichkeitsdichte über dem Zustandsraum zum Zeitpunkt $t$ ist somit gegeben als $p(S_t|S_1, S_2, ...S_{t-1})$. Damit wird die Dichte über dem Beobach-

tungsraum ebenfalls abhängig von der Zustandssequenz, einschließlich des aktuellen Zustandes: $p(\mathbf{x}_t|\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t-1}, S_1, ..., S_t)$. Ziel der Analyse eines solchen Prozesses ist es, seine statistischen Eigenschaften, d.h. die Form von $p(S_t|...)$ und $p(\mathbf{x}_t|...)$, anhand der beobachteten Daten bestmöglich zu modellieren.

Ein HMM trifft vereinfachende Annahmen bezüglich der Struktur des stochastischen Prozesses, um die Modellkomplexität zu beschränken. Es wird angenommen, dass der Zustandsraum diskret und endlich ist, d.h. zu jedem Zeitpunkt $t$ befindet sich der Prozess in einem Zustand aus einer diskreten Zustandsmenge $\mathcal{S} = \{s_i, i = 1...m\}$. Weiterhin sei der Prozess stationär und einfach, d.h. seine Parameter seien konstant und die Zustandsverteilung sei nur vom direkten Vorgängerzustand abhängig. In diesem Fall vereinfachen sich die Zustands- und Beobachtungsverteilungen zu

$$\begin{aligned} p(S_t|S_1, S_2, ...S_{t-1}) &= p(S_t|S_{t-1}), \\ p(\mathbf{x}_t|\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t-1}, S_1, ..., S_t) &= p(\mathbf{x}_t|\mathbf{x}_{t-1}, S_t). \end{aligned} \tag{5.29}$$

Nimmt man zusätzlich an, dass die Beobachtungen statistisch unabhängig voneinander sind, vereinfacht sich die Beobachtungsverteilung weiter zu $p(\mathbf{x}_t|S_t)$. Damit lässt sich ein HMM als probabilistischer Graph wie in Abbildung 9 darstellen. Ein HMM $\Lambda$ ist demnach vollständig beschrieben durch

$$\begin{aligned} \Lambda &= \{\mathcal{S}, \mathbf{A}, \bar{\pi}, b_j(x_k)\}, \quad \text{mit} \\ \mathbf{A} &= (a_{ij}|a_{ij} = P(S_t = s_j|S_{t-1} = s_i)), \\ \bar{\pi} &= (\pi_i|\pi_i = P(S_1 = s_i)), \\ b_j(\mathbf{x}) &= p(\mathbf{x}|S_t = s_j), \end{aligned} \tag{5.30}$$

mit der Matrix der Zustandsübergangswahrscheinlichkeiten $\mathbf{A}$, dem Vektor der Startwahrscheinlichkeiten $\bar{\pi}$ und den zustandsspezifischen Emissionswahrscheinlichkeitsdichten $b_j(\mathbf{x})$. Das Modell befindet sich also zu Beginn der Beobachtung mit der Wahrscheinlichkeit $\pi_i$ im Zustand $s_i$, vollführt während der Beobachtungsdauer Zustandsübergänge gemäß $\mathbf{A}$ und generiert bzw. emittiert zu jedem Beobachtungszeitpunkt eine Beobachtung $\mathbf{x}_k$ gemäß $b_j(\mathbf{x})$. Die so entstehende Beobachtungsfolge ist der einzige beobachtbare Teil des Modells. Die Folge von Zustandsübergängen, welche die Beobachtungsfolge generiert hat, ist versteckt.

Werden die Emissionswahrscheinlichkeitsdichten $b_j(\mathbf{x})$ als kontinuierliche Dichtefunktionen und individuell für jeden Zustand des HMM modelliert, so spricht man von *kontinuierlichen* HMM. Üblicherweise werden für die Repräsentation GMM (vgl. Kapitel 6.3.1) verwendet. In dieser Arbeit kommen sog. *semi-kontinuierliche* HMM zum Einsatz. Der Unterschied ist, dass hier ein einziges Codebuch Gauss'scher Dichten

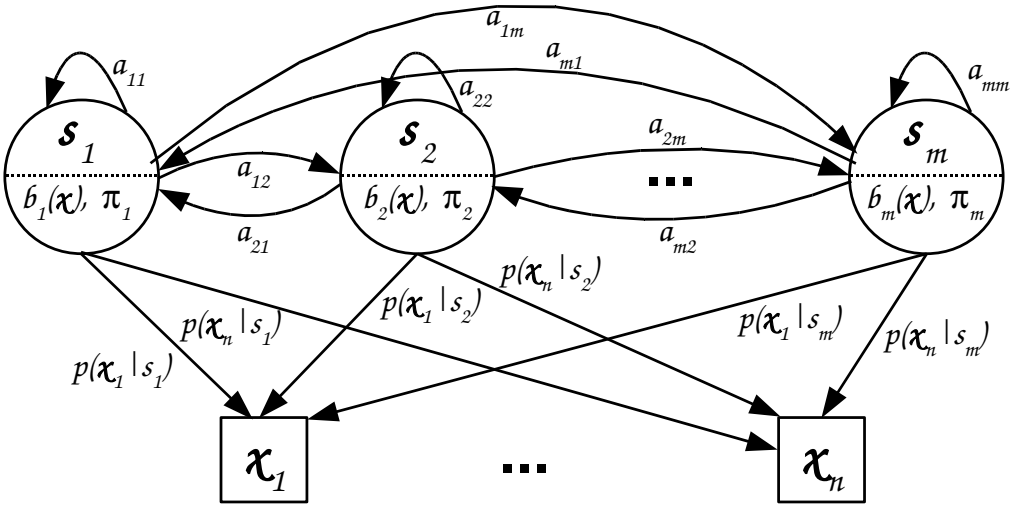


Abbildung 9: Grafische Darstellung eines HMM und seiner Komponenten.

existiert, das von allen Zuständen des HMM benutzt wird. Lediglich die Mischverteilungsgewichte $\gamma_i$ werden für jeden Zustand individuell bestimmt. Diese Art der Modellierung bietet Vorteile bei schlechter Datenlage, weil die Anzahl freier Parameter im Vergleich zu einem kontinuierlichen HMM geringer ist. Somit werden weniger Trainingsbeispiele benötigt.

### 5.6.2 *Training*

Leider existiert keine bekannte Methode, um für eine gegebene Stichprobe automatisch ein HMM zu generieren, welches ein Optimalitätskriterium erfüllt. Es ist lediglich möglich, ausgehend von einem Startpunkt die freien Parameter des Modells iterativ zu optimieren, also ein lokales Optimum zu erreichen. Die erreichbare Qualität hängt somit entscheidend von der Initialisierung und von der gewählten Struktur und Topologie des Modells ab. Deshalb verlässt man sich an dieser Stelle i.d.R. auf die Intuition und Erfahrung eines Experten. Die verbreitetesten Algorithmen für die Optimierung der Modellparameter sind der *Baum-Welch-Algorithmus* und das *Viterbi-Training* ([57] Kapitel 5.7). Sie unterscheiden sich im Wesentlichen durch das verwendete Qualitätskriterium.

### 5.6.3 *Dekodierung*

Der typische Anwendungsfall eines HMM ist die Analyse einer Beobachtungsfolge $\mathcal{X}$ fester Länge (bzw. vom ersten Beobachtungszeitpunkt $t_0$ bis zum aktuellen Zeitpunkt $t$). Es seien $q$ Modelle $\Lambda_l, l = 1...q$ gegeben, im Fall der Gestenerkennung z.B. je ein Modell pro Gestentyp, oder bei einer Spracherkennungsaufgabe ein Modell pro Sprachlaut. Die Inferenzaufgabe besteht darin, in der Beobachtungsfolge das Auftreten eines oder mehrerer der mit den $\Lambda_l$ modellierten Muster zu finden und gleichzeitig die zeitliche Begrenzung des Auftretens (d.h. die zugehörige Subsequenz von $\mathcal{X}$) zu ermitteln (*Segmentierung*). Zusätzlich soll jedem Segment ein Klassenkennzeichen zugewiesen werden, abhängig davon, welches der $\Lambda_l$ das jeweilige Segment am besten beschreibt (*Klassifikation*). Eine große Stärke von HMM ist, dass Segmentierung und Klassifikation in einem einheitlichen probabilistischen Rahmen integriert vorgenommen werden können. Dieser Vorgang wird als Dekodierung bezeichnet.

Um dies zu realisieren, müssen die internen Abläufe des Modelles analysiert werden. Die Grundannahme eines HMM ist, dass die Beobachtungen $\mathcal{X}$ gemäß der Emissionswahrscheinlichkeiten $b_{lj}(\mathbf{x}_k)$ aus den Modellen $\Lambda_l$ durch Durchlaufen einer bestimmten Zustandssequenz mit Zustandsübergängen gemäß $\mathbf{A}_l$ generiert wurden. Die im probabilistischen Sinne optimale Lösung ist demzufolge gegeben durch die Zustandssequenz, welche $\mathcal{X}$ mit maximaler a-posteriori Wahrscheinlichkeit erzeugt:

$$\mathbf{s}^* = \arg\max_{s} P(\mathbf{s}|\mathcal{X}, \tilde{\Lambda}). \tag{5.31}$$

Hierbei bezeichnet $\tilde{\Lambda}$ ein Verbundmodell, das aus den $\Lambda_l$ gemäß einer der jeweiligen Mustererkennungsaufgabe angepassten Struktur konstruiert wird (näheres hierzu folgt später in Kapitel 6.7.4). Die optimale Zustandsfolge $\mathbf{s}^*$ lässt sich effizient mit Hilfe des *Viterbi-Algorithmus* ([57], Kapitel 5.6) bestimmen. Die Segmentierung ergibt sich dann durch Übergänge zwischen Modellen in $\mathbf{s}^*$. Solche Übergänge sind nur zwischen End- und Startzuständen möglich. Demzufolge existiert für jedes Segment des optimalen Pfades $\mathbf{s}^*$ genau ein aktives Teilmodell $\Lambda_l$. Die Klassifikation ist somit durch das im jeweiligen Segment aktive Teilmodell gegeben.

# 6 KONZEPTION UND REALISIERUNG

In Kapitel 3.5 wurden Teilaufgaben und Anforderungen erarbeitet, die im Rahmen dieser Arbeit umzusetzen sind. Aus den Teilaufgaben ergeben sich direkt die einzelnen Verarbeitungsschritte und letztlich die realisierte Systemarchitektur, wie sie in Abbildung 10 vereinfacht dargestellt ist. Die konkrete Realisierung der einzelnen Bestandteile wird in den folgenden Abschnitten behandelt.

## 6.1 GRUNDLEGENDE ÜBERLEGUNGEN

Die meisten in Kapitel 4 vorgestellten Arbeiten zur Gesten- und Aktionserkennung gehen von erheblichen vereinfachenden Annahmen aus oder verwenden speziell auf die Aufgabe zugeschnittene Sensorkonfigurationen. Für eine intuitive Interaktion eines Nutzers mit einer Umgebungsintelligenz ist eine Bindung der Interaktion an bestimmte physische Entitäten oder eine Beschränkung des Interaktionsbereiches nicht akzeptabel. Der Nutzer sollte sich idealerweise innerhalb der intelligenten Umgebung frei bewegen und von jedem beliebigen Punkt aus Kommandos an die Umgebungsintelligenz richten können. Das bedeutet, dass sowohl sein Standpunkt als auch seine Orientierung relativ zur umgebenden Sensorik keinen Einschränkungen unterliegen sollten.

Praktisch ergeben sich trotzdem einige Einschränkungen, z.B. ist eine dreidimensionale visuelle Erkennung nur möglich, wenn der Nutzer in mindestens zwei Kameras sichtbar ist und die relevanten Körperteile – insbesondere Kopf und Hände – nicht verdeckt sind. Deshalb soll die Erkennung in einem prinzipiell uneingeschränkten Multikamera-Setup geschehen. D.h. in der Umgebung befinden sich mehrere Kameras mit sehr unterschiedlichen Sichtfeldern, die eine gute visuelle Abdeckung der Szene ermöglichen. Je mehr Kameras vorhanden sind und je besser diese verteilt sind, umso besser wird die Abdeckung und umso weniger Einschränkungen ergeben sich für das Interaktionsszenario. Aber auch mit relativ wenigen Kameras kann eine gute Abdeckung erreicht werden, wenn diese geschickt platziert oder aktiv ausgerichtet werden, so dass die Erfassung der Szene gemäß gewisser Kriterien optimiert wird [170]. Es wird daher im Folgenden davon ausgegangen, dass die Anordnung der Kameras prinzipiell beliebig sein kann (aber zur Laufzeit bekannt ist) und die Kameras aktiv

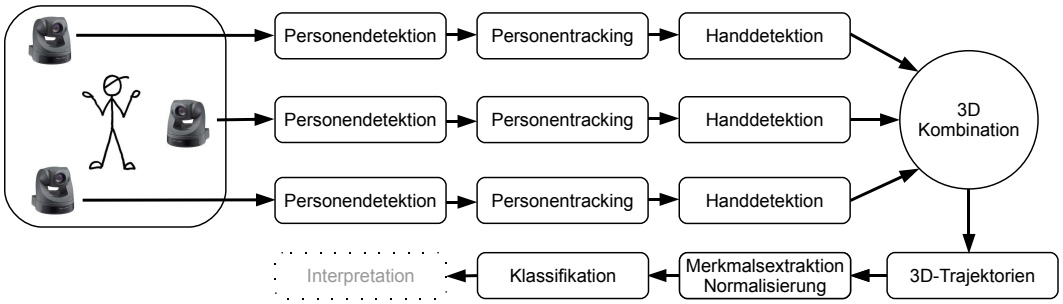


Abbildung 10: Prinzipieller Aufbau des realisierten Gestenerkennungssystems.

sind. D.h. ihre Parameter können sich zur Laufzeit ändern, was eine entsprechende Nachführung der Kalibrierung nötig macht.

Darüberhinaus sollen die entwickelten Techniken nicht von hochspezialisierter Sensorik abhängig sein. Deshalb kommen keine kalibrierten Stereosysteme, *Time of flight* Kameras oder ähnliche teure Spezialhardware zum Einsatz, sondern handelsübliche, unsynchronisierte Videokameras. Daraus ergeben sich einige zusätzliche Schwierigkeiten, auf die im weiteren Verlauf noch eingegangen wird.

Hinsichtlich der Reaktivität ist es wünschenswert, dass nur Methoden zum Einsatz kommen, die geringe inhärente Latenzen aufweisen. Ansätze, die auf spatiotemporalen Strukturen mit großer temporaler Ausdehnung oder der Analyse längerer Bildsequenzen basieren, sind für die Realisierung einer reaktiven Mensch-Maschine-Schnittstelle aus offensichtlichen Gründen nicht geeignet. Der Fokus liegt auf Methoden, die einerseits einfach genug für eine effiziente Umsetzung sind, andererseits Potential zur Parallelisierung bieten. Letzteres ist beispielsweise für nahezu alle Pixeloperationen oder regionsbasierten Ansätze der Fall.

In diesem Zusammenhang stellt ein Multikamerasystemes mit einer prinzipiell beliebigen Anzahl von Kameras eine nicht unerhebliche Schwierigkeit dar, denn viele Kameras bedeuten auch, dass ein Vielfaches der Datenmenge zu verarbeiten ist. Um die Skalierbarkeit jederzeit zu gewährleisten, wird die Datenverarbeitung so lange wie möglich unabhängig auf den einzelnen zweidimensionalen Bildströmen ausgeführt. Damit ergibt sich auf natürliche Weise eine parallele Verarbeitungsstruktur, die ggf. auf mehrere Rechner verteilt werden kann. Die letztendliche 3D-Kombination erfolgt nicht auf der Ebene der Bilddaten, sondern auf der höheren Ebene der Auswertungsergebnisse der einzelnen Bildströme. Das stellt gewisse Anforderungen an Synchronisation, Kombination und Auswahl der Einzelergebnisse, verhindert aber die sehr rechenaufwändige 3D-Rückprojektion kompletter Bilder.

Trotz allem liegen auch dieser Arbeit einige vereinfachende Annahmen zugrunde, um die Komplexität des Problems zu begrenzen. Es wurde bereits mehrfach erwähnt, dass von einem kooperativen Nutzer ausgegangen wird. Darüberhinaus wird angenommen, dass sich nur jeweils eine Person im Sichtfeld der Kameras befindet, die während der Ausführung einer Geste an einer Position im Raum stehen bleibt. Weiterhin muss die Person jederzeit in mindestens zwei der Kameras sichtbar sein, so dass eine 3D Kombination der Einzelergebnisse möglich ist.

## 6.2 INTEGRATIONSUMGEBUNG

Die intelligente Umgebung, in der die entwickelten Methoden evaluiert werden, ist das im Institut für Roboterforschung der TU Dortmund errichtete intelligente Haus „FINCA“[1] (Abbildung 11). Die FINCA dient dort als Integrationsumgebung für Forschungsprojekte im Bereich der multimodalen Mensch-Maschine-Interaktion, automatischen Situationserkennung und Umgebungsintelligenz. Sie besteht aus zwei Teilen, einem offenen Mehrzweckbereich und einem Konferenzraum. Die vorgestellte Arbeit wird in letzteren integriert.

Der Konferenzraum verfügt über verschiedene Sensorik, unter anderem verteilte Mikrofonfelder, Bewegungsmelder und mehrere aktive deckenmontierte Sony EVI D70P Schwenk-Neige-Zoom Kameras, die über eine analoge Verbindung Farbbilder in PAL-Auflösung (768 x 576 Pixel) mit einer Frequenz von maximal 25 Hertz liefern. Die Positionen dieser Kameras können in weiten Grenzen frei gewählt werden, üblicherweise befinden sie sich ungefähr in den Ecken des Raumes, wie in Abbildung 11 angedeutet. Jede Kamera ist mit einem dedizierten Rechner verbunden, der ihre Bilder empfängt und ggf. als Server fungiert, der die Daten über eine TCP/IP-Verbindung über Netzwerk bereitstellt. Die Kameras selber können über das von Sony standardisierte VISCA-Protokoll [185] konfiguriert und gesteuert werden.

Darüberhinaus ist die FINCA mit einer KNX-Gebäudeinstallation [96] ausgestattet, über die sich beispielsweise die Jalousien oder die Beleuchtung vielfältig ansteuern lassen und jederzeit der Status angeschlossener Geräte und Sensoren abgefragt werden kann. Somit sind über die Gebäudeinstallation einerseits Informationen über den aktuellen Zustand und Kontext der intelligenten Umgebung verfügbar, andererseits kann über sie der Zustand gezielt verändert werden.

1 ***F***lexible ***I***ntelligent E***n***vironment with ***C***omputational ***A***ugmentation

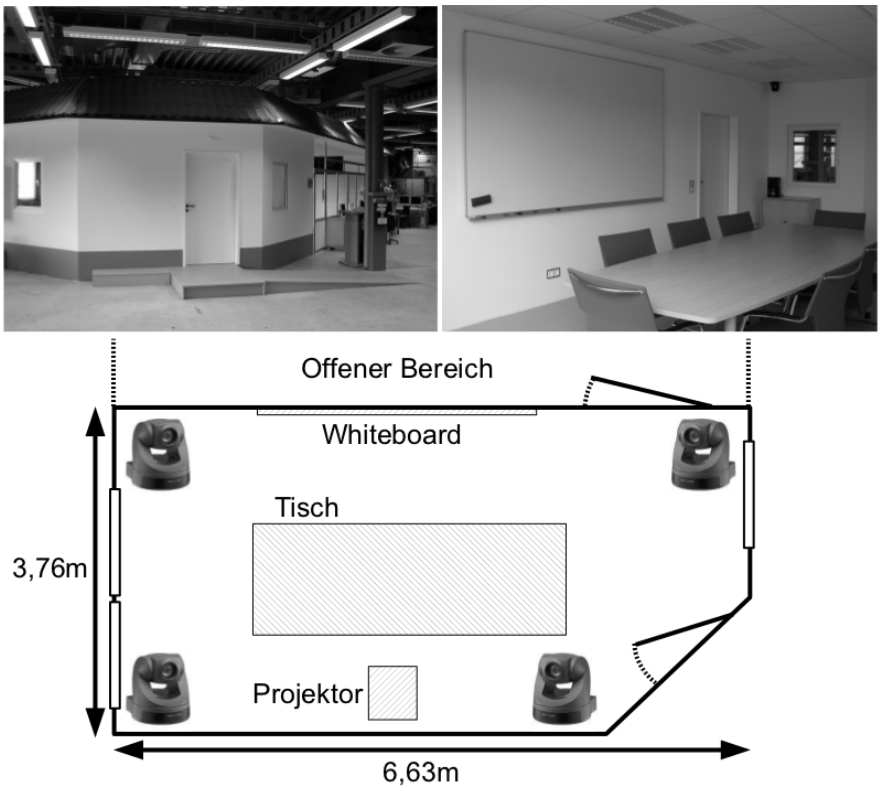


Abbildung 11: Die Integrationsumgebung FINCA. Von oben links nach unten: Außenansicht, Innenansicht des intelligenten Konferenzraumes, Grundriss-Skizze des intelligenten Konferenzraumes. In der Innenansicht ist rechts oben eine der verwendeten aktiven Kameras erkennbar.

## 6.3 PERSONENDETEKTION

Damit eine intelligente Umgebung auf Anweisungen von Nutzern reagieren kann, muss zuerst bekannt sein, ob ein Nutzer anwesend ist und wo er sich befindet. Der erste Schritt besteht somit aus einer Personendetektion. Das Ziel ist, innerhalb jedes Kamerabildes einen Bereich zu identifizieren, in dem sich mit hoher Wahrscheinlichkeit Personen befinden. Dies geschieht durch Anwendung eines Hintergrundmodelles. Weiterhin müssen die geschätzten Positionen aus mehreren Kamerabildern später zu Hypothesen im dreidimensionalen Raum kombiniert werden. Um eine aufwändige voxelbasierte Rekonstruktion zu vermeiden, wird eine Reduzierung der Vordergrund-

regionen auf einige wenige Punkthypothesen angestrebt. Zu diesem Zweck wird innerhalb des durch die Hintergrundmodellierung eingeschränkten Suchbereiches ein ansichtsbasierter Detektor eingesetzt.

### 6.3.1 *Hintergrundmodellierung*

Der intelligente Konferenzraum der FINCA ist eine weitgehend statische Innenraum-Umgebung. Aus diesem Grund erscheint eine Modellierung des Szenenhintergrundes erfolgversprechend. Hierbei sind allerdings einige Besonderheiten zu beachten:

- Der Konferenzraum verfügt über Außenfenster und einen Projektionsbereich für Vorträge und Präsentationen. Die Annahme eines statischen Hintergrundes ist zumindest in diesen Bereichen potentiell verletzt. Demzufolge ist mit Segmentierungsfehlern zu rechnen.
- Beleuchtung und Jalousien gehören zu den Funktionen, die durch das Gestenerkennungssystem gesteuert werden könnten. Der Einsatz der Gestenerkennung führt also mitunter zu einer Veränderung der Umgebungsbedingungen, von denen seine ersten Verarbeitungsschritte direkt abhängig sind.
- Die verwendeten Kameras sind aktive Schwenk-Neige-Zoom Kameras. Aktive Kameras bieten viele Vorteile für adaptive Mensch-Maschine-Schnittstellen, wie beispielsweise die Möglichkeiten der Nachführung, Optimierung der Szenenabdeckung oder aktiven Fokussierung auf bestimmte Szenebereiche. Damit kann aber nicht mehr von einem statischen Hintergrundbild für jede Kamera ausgegangen werden, die einfache Methode der statischen Hintergrundsubtraktion ist somit nicht anwendbar.

Das verwendete Hintergrundmodell muss also adaptiv sein, d.h. es muss in der Lage sein, sich innerhalb kurzer Zeit an geänderte Umgebungsbedingungen anzupassen. Im Folgenden werden einige Möglichkeiten der Modellierung vorgestellt, welche im Rahmen dieser Arbeit umgesetzt wurden.

#### *Adaptive Hintergrundsubtraktion mit Vodergrundhistorie*

Die einfachste Modellierungsart basiert auf pixelweiser Hintergrundsubtraktion, wobei das vorgehaltene Hintergrundbild adaptiv realisiert ist. Das Modell $\mathcal{H}_0$ wird zur Laufzeit mit dem ersten von der jeweiligen Kamera empfangenen Bild $\mathbf{B}_0$ initialisiert. Die Entscheidung, ob ein Pixel $b(t, x, y)$ zum Zeitpunkt $t$ Vorder- oder Hintergrund ist,

wird anhand eines Schwellwertes $\delta$ über Pixelfarbdifferenzen getroffen. Das Resultat ist eine binäre Vordergrundmaske $\mathbf{M}(t,x,y) = (m(t,x,y))^2$:

$$m(t,x,y) = \begin{cases} 1 & \text{falls} \quad \sum_c |b^c(t,x,y) - \mathcal{H}^c(t,x,y)| > \delta \\ 0 & \text{sonst} \end{cases}. \tag{6.1}$$

Hierbei bezeichnet $c$ die Farbkanäle.

In der Folge wird das Modell mit der neuen Beobachtung aktualisiert:

$$\mathcal{H}(t+1,x,y) = (1-\lambda)\mathcal{H}(t,x,y) + \lambda\mathbf{B}(t,x,y), \quad \lambda \in [0,1]. \tag{6.2}$$

Die Adaptivitätskonstante (oder Lernrate) $\lambda$ definiert dabei, wie schnell die Adaption geschieht. Das führt dazu, dass Änderungen der Szene, die über einen längeren Zeitraum statisch bleiben, zunehmend in das Hintergrundmodell integriert und nach einer bestimmten Zeit nicht mehr als Vordergrund detektiert werden. Das Modell realisiert also eine einfache Form der Bewegungsdetektion. Die Adaptivität ist notwendig, um mit veränderlichen Umgebungsbedingungen umgehen zu können, führt aber auch dazu, dass ein sich schnell adaptierendes Modell einmal detektierte Vordergrundregionen schnell „vergisst“, sobald diese statisch werden. Aus Stabilitätsgründen ist dieses Verhalten nicht immer wünschenswert. Insbesondere ist die Dauer, über die ein Objekt als Vordergrund detektiert wird, direkt davon abhängig, wie stark sich sein Farbwert von demjenigen des Hintergrundmodelles an dieser Stelle unterscheidet.

Eine Möglichkeit, dieses Problem zu lösen, besteht in der Einführung einer zeitlichen Vordergrundhistorie, d.h. eines pixelbasierten „Gedächtnisses“, das als Verfallsfunktion $f(t)$ über Pixelwerte definiert ist:

$$m(t,x,y) = \begin{cases} 1, \quad t_d(x,y) = t & \text{falls} \quad \sum_c |b^c(t,x,y) - \mathcal{H}^c(t,x,y)| > \delta \\ f(t - t_d(x,y)) & \text{sonst.} \end{cases} \tag{6.3}$$

Hierbei bezeichnet $t_d(x,y)$ den letzten Zeitpunkt, zu dem der betreffende Pixel als Vordergrund detektiert wurde. Die Vordergrundmaske $\mathbf{M}(t,x,y)$ ist nun nicht mehr binär, sondern besitzt kontinuierliche Werte. Pixel mit dem Wert 1 kennzeichnen Vordergrundregionen, die im aktuellen Zeitpunkt aufgrund des Vergleichs mit dem Hintergrundmodell detektiert wurden. Werte kleiner als 1 kennzeichnen Regionen,

2 Die explizite Notation der Zeitabhängigkeit wird im Folgenden nur benutzt, wenn es zum Verständnis wichtig ist, und ansonsten zu Gunsten besserer Lesbarkeit vernachlässigt. Es gilt, dass alle Modelle, Beobachtungen und Hypothesen grundsätzlich mit dem aktuellen Beobachtungszeitpunkt assoziiert sind.

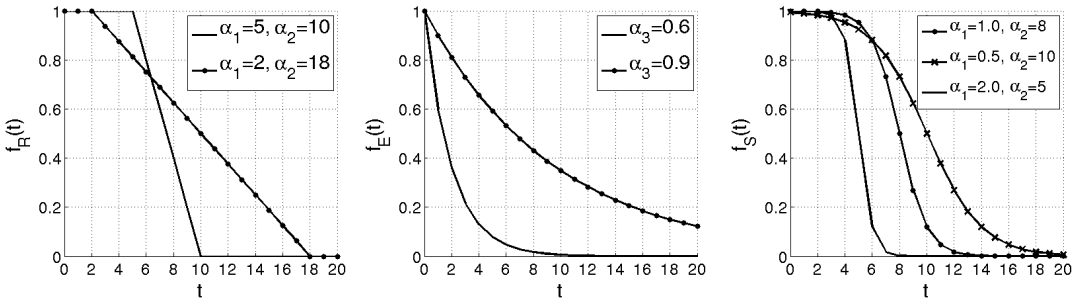


Abbildung 12: Beispiele für verwendete Verfallsfunktionen in der adaptiven Hintergrundsubtraktion.

die zu einem früheren Zeitpunkt detektiert wurden. Je kleiner der Wert der Vordergrundmaske an einer Position ist, umso länger liegt die Klassifikation dieses Pixels als Vordergrund zurück. Damit ist diese Vorgehensweise sehr ähnlich zu *Motion History Images* [14]. Die Funktion $f(t)$ ist eine beliebige monoton fallende Funktion, die anwendungs- und umgebungsabhängig gewählt werden kann. Im Rahmen dieser Arbeit wurden gute Ergebnisse mit Rampen- ($f_R(t)$), asymptotischen Exponential- ($f_E(t)$) sowie Sigmoidfunktionen ($f_S(t)$) erzielt:

$$f_R(t) = \begin{cases} 1 & \text{falls} \quad t < \alpha_1 \\ 0 & \text{falls} \quad t > \alpha_2 \\ 1 - \frac{(t-\alpha_1)}{\alpha_2 - \alpha_1} & \text{sonst} \end{cases}, \tag{6.4}$$

$$f_E(t) = \alpha_3^t,$$

$$f_S(t) = 1 - \frac{1}{1 + e^{-\alpha_1(t-\alpha_2)}}.$$

Hierbei sind $\alpha_1, \alpha_2 \in \mathbb{R}^+$ und $\alpha_3 \in ]0,1[$ Parameter der Funktionen. In Abbildung 12 sind die Funktionen dargestellt. Der Effekt ist ein „Nachklingen“ der Werte der Vordergrundmaske. Obige Verfallsfunktionen sind nur von $t$ abhängig und somit unabhängig vom Betrag der Pixeldifferenzen.

Der Vorteil dieses Ansatzes liegt in seiner Einfachheit und Effizienz: Die zeitabhängigen Terme können – bei Annahme einer konstanten Bildrate – durch separate Zähler für jeden Pixel der Maske ersetzt werden. Damit kann $t$ nur ganzzahlige Werte annehmen, für welche die Verfallsfunktion vorberechnet werden kann. Die Verwendung eines vorgehaltenen Hintergrundbildes bedeutet jedoch, dass starke Änderungen der Umgebungsbedingungen und insbesondere Kamerabewegungen sofort zu großen

Fehlern führen. Aufgrund der Adaptivität werden diese zwar nach einer gewissen Zeit ausgeglichen. Dieses Modell hat aber aufgrund dessen eine gewisse Antwortzeit $t_A$ auf Änderungen, innerhalb derer es fehlerhafte Ergebnisse liefert. Diese Antwortzeit hängt von der „Nachklingdauer" $t_{N,f}$ der Verfallsfunktion[3] und der Lernrate $\lambda$ ab. Ihre obere Grenze lässt sich somit abschätzen als $t_A = \lambda^{-1} + t_{N,f}$. Im einfachsten Fall kann damit durch Ignorieren einer entsprechenden Anzahl von Bildern umgegangen werden, dies führt jedoch zu inhärenten Latenzen und Totzeiten. Insbesondere häufige schnell aufeinander folgende Änderungen können zu Problemen führen.

In Abbildung 13 sind einige Beispielergebnisse dargestellt. In den Bildreihen drei bis fünf ist klar zu erkennen, dass die Verwendung der Vordergrundhistorie die Stabilität des Ergebnisses verbessert, allerdings auf Kosten einer größeren Anzahl von Fehldetektionen. Dies liegt zum Einen daran, dass einerseits bei bewegten Objekten die Positionen nachklingen, an denen sich das Objekt in vorherigen Zeitschritten befand (hierdurch entsteht in der letzten Bildreihe ein Bewegungsschatten links von der Person). Zum Anderen klingen auch fehlerhaft detektierte Vordergrundregionen länger nach, zu erkennen an der hellen Region am linken Bildrand.

*Statistische Farbmodelle*

Geht man davon aus, dass der Hintergrund aus wenigen charakteristischen Farben besteht, deren Verteilung ansichtsunabhängig ist – eine Annahme, die in Innenräumen oft erfüllt ist –, kann diese Farbverteilung in einem statistischen Modell repräsentiert werden. Objekte, deren Farbe sich deutlich von der des Hintergrundes unterscheidet, werden somit als Vordergrund detektiert. Statistische Modelle können – bekannte und hinreichend statische Umgebungsbedingungen vorausgesetzt – offline gelernt und ohne Initialisierungsphase eingesetzt werden. Anders als bei der pixelbasierten Modellierung ist das Modell zudem unabhängig von den Bildkoordinaten. Somit ist es robuster gegen Änderungen der Kameraausrichtung.

Die einfachste und effizienteste statistische Repräsentation ist ein normalisiertes Farbhistogramm mit $k$ Bins:

$$\mathcal{H} = (h_i, i = 1...k), \quad h_i = \frac{1}{n_b} \sum_{x,y} \delta(f_h(b(x,y)) - i). \tag{6.5}$$

3 Die Nachklingdauer ist die Zeit, die bis zum Abklingen der Funktion auf 0 bzw. bei asymptotischen Funktionen auf einen genügend kleinen Wert vergeht.

Abbildung 13: Beispiele für die Vordergrundsegmentierung per adaptiver Hintergrundsubtraktion. Die beiden oberen Reihen zeigen eine Situation, in der eine Person die Szene betritt und durch das Modell segmentiert wird. Die dritte und vierte Reihe zeigen eine Situation, in der die Person still steht und durch die Adaption des Modells in den Hintergrund übergeht. Die letzte Reihe zeigt die gleiche Szene unter Verwendung der Vordergrundhistorie aus (6.3). (Parameter: Lernrate $\lambda = 0.1$, Schwellwert $\delta = 20$, Verfallsfunktion Sigmoid, $\alpha_1 = 0.5$, $\alpha_2 = 8$.)

Dabei ist $n_b$ die Anzahl der Bildpixel. Die Wahrscheinlichkeit, dass ein Bildpixel $b(x, y)$ zu der durch $\mathcal{H}$ repräsentierten Klasse gehört, entspricht dann einfach der relativen Häufigkeit des zugehörigen Histogrammbins:

$$P(\mathcal{H}|b(x, y)) = h_l, \quad l := f_h(b(x, y)). \tag{6.6}$$

Eine adaptive Modellierung ist auch für Histogramme möglich, entweder durch Beibehaltung der absoluten Binhäufigkeiten und Aktualisierung mit den Farbwerten des aktuellen Bildes oder durch eine gewichtete Mittelwertsbildung.

Eine etwas mächtigere Modellierungsmöglichkeit stellen GMM dar. Hierbei wird die klassenbedingte Wahrscheinlichkeitsdichte der Farbwerte durch eine gewichtete Summe multivariater Normalverteilungen repräsentiert:

$$\mathcal{H} = \{\{\gamma_i, \mu_i, \Sigma_i\}, \quad i = 1...k\} \tag{6.7}$$
$$p(b(x,y)|\mathcal{H}) = \sum_i \gamma_i \mathcal{N}(b(x,y) - \mu_i, \Sigma_i), \quad \sum_i \gamma_i = 1.$$

Die Parameter $\gamma_i, \mu_i, \Sigma_i$ einer Mischverteilung lassen sich anhand einer Trainingsstichprobe und einer beispielsweise per Vektorquantisierung erhaltenen Vorsegmentierung mit Hilfe des iterativen *Expectation Maximization* (EM) Algorithmus bestimmen (vgl. [57] Kapitel 4.4). Eine Adaption des Modelles zur Laufzeit an neu beobachtete Daten kann z.B. mit dem Algorithmus von McKenna [117] erfolgen. Die Adaption eines GMM ist aber generell vergleichsweise rechenaufwändig, weshalb diese Modellierungsart sich vor allem für statische Modelle eignet.

Die pixelweise a-posteriori Wahrscheinlichkeit $P(\mathcal{H}|b(x,y))$ ergibt sich durch die Bayes-Regel zu

$$P(\mathcal{H}|b(x,y)) = \frac{P(b(x,y)|\mathcal{H})P(\mathcal{H})}{P(b(x,y))} \tag{6.8}$$

und ist somit proportional zur klassenbedingten Wahrscheinlichkeit $P(b(x,y)|\mathcal{H})$. Deshalb kann die Auswertung direkt durch Einsetzen der Farbwerte von $b(x,y)$ in (6.7) erfolgen.

*HOG-basierte Modellierung*

Alternativ zur Modellierung basierend auf Farbwerten bietet sich eine Modellierung des Hintergrundes anhand struktureller Information, d.h. Intensitätskanten oder Textur, an (vgl. Kapitel 2.8). In vorliegender Arbeit wurde eine Modellierung basierend auf HOG-Deskriptoren (Kapitel 5.3) realisiert. Da ein HOG-Deskriptor immer einen kleinen rechteckigen Bildausschnitt beschreibt, ist diese Modellierungsart nicht pixel- sondern regionenbasiert. Sie vermeidet damit potentiell das Problem, dass Intensitätskanten keine großflächige Ausdehnung haben und eine kantenbasierte Modellierung somit typischerweise nur wenige Pixel des Bildes erfasst.

Die Idee besteht darin, das Konzept der adaptiven farbbasierten Hintergrundsubtraktion direkt auf HOG-Merkmale zu übertragen. Das Modell entspricht somit wie in Abschnitt 6.3.1 einem dynamisch adaptierten Bild der Szene. In jedem

Zeitschritt wird ein gleitendes Fenster über das aktuelle Kamerabild und das Modellbild geschoben und an jeder Position jeweils ein identisch aufgebauter HOG-Deskriptor berechnet. Als Ergebnis erhält man für jede Fensterposition jeweils einen Modelldeskriptor $\mathcal{D}_m(b(x,y)) = (d_{m,i}, i = 1...n)$ und einen Beobachtungsdeskriptor $\mathcal{D}_b(b(x,y)) = (d_{b,i}, i = 1...n)$. Hierbei bezeichnet $b(x,y)$ den zentralen Pixel des Fensters und $n$ die Deskriptorlänge. Diese Deskriptoren können nun mit einem geeigneten histogrammbasierten Distanzmaß verglichen werden. Es wird ein normalisiertes Distanzmaß benötigt, um die Deskriptordistanz definiert auf Intensitäts- oder Wahrscheinlichkeitswerte abbilden zu können. Weiterhin ist bei HOG-Deskriptoren mit leeren Histogrammbins zu rechnen, das Distanzmaß muss also dagegen robust sein. Diese Anforderungen erfüllt die in Kapitel 5.4.3 eingeführte Bhattacharyya-Distanz:

$$d(\mathcal{D}_m(b(x,y)), \mathcal{D}_b(b(x,y))) = \sqrt{1 - \sum_i \sqrt{d_{m,i} d_{b,i}}}. \tag{6.9}$$

Die Bhattacharyya-Distanz ist nur für normalisierte Histogramme sinnvoll, d.h. die HOG-Deskriptoren müssen so normalisiert werden, dass $\sum_i d_{m,i} = \sum_i d_{b,i} = 1$.

Ausdruck (6.9) liefert immer einen Distanzwert $d(...) \in [0,1]$, wobei 1 maximale Unterschiedlichkeit der beiden Deskriptoren bedeutet. Der Distanzwert kann also direkt als Vordergrundwahrscheinlichkeit $P(\mathcal{H}|b(x,y))$ des Pixels $b(x,y)$ interpretiert werden. Eine Implementierung in einer Auflösungspyramide mit mehreren gleitenden Fenstern unterschiedlicher Größe ist ebenfalls möglich:

$$P(\mathcal{H}|b(x,y) = \max_l(d^l(\mathcal{D}_m(b(x,y)), \mathcal{D}_b(b(x,y))). \tag{6.10}$$

Hierbei bezeichnet $d^l(...)$ die Bhattacharyya-Distanz für den Pixel $b(x,y)$, ermittelt für die $l$-te Stufe der Auflösungspyramide. In der Praxis wird die Fenstergröße für eine gegebene Anzahl von Auflösungsstufen jeweils mit einem konstanten Faktor skaliert. Abbildung 14 zeigt einige Beispielergebnisse.

Dieses Modell ist anfälliger gegen Bildrauschen als die farbbasierten Ansätze (vgl. mit Abbildung 13), weil jeder Rauschpixel eine Kantenantwort induziert und somit den Deskriptor beeinflusst. Die Berechnung von (6.9) für die Vielzahl an Positionen des gleitenden Fensters ist zudem sehr rechenaufwändig, so dass diese Art der Modellierung weder die Qualität noch die Effizienz der farbbasierten Modelle erreicht.

*Kombination von Vorder- und Hintergrundmodellen*

Bisher war immer nur von einem einzelnen Hinter- oder Vordergrundmodell die Rede. Natürlich ist es genauso möglich, jeweils ein separates Modell zu benutzen und deren

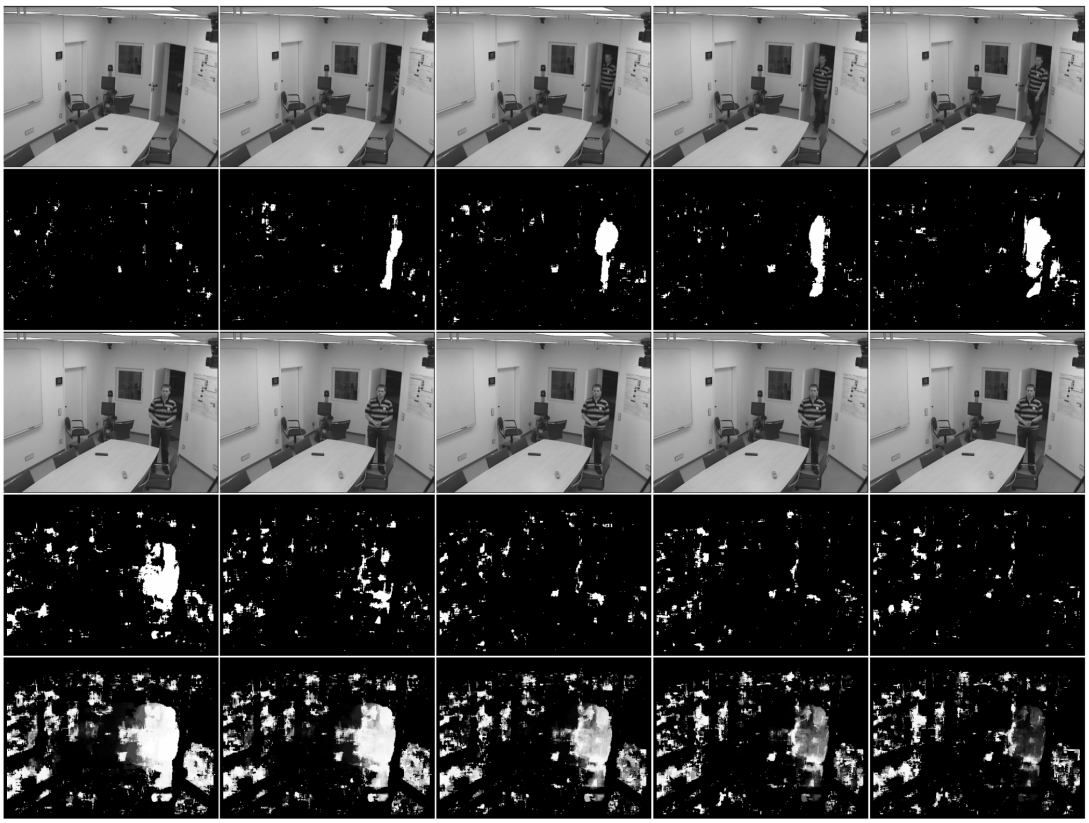

Abbildung 14: Beispiele für die Vordergrundsegmentierung mit einem HOG-basierten Modell unter Verwendung der gleichen Daten wie in Abbildung 13. (Parameter: HOG mit 2x2 Blöcken, je 1 Zelle, 8 Bins, 3 Auflösungsstufen, initiale Fenstergröße 12x12 Pixel, Skalierungsfaktor 1.5, Lernrate $\lambda = 0.1$, Schwellwert$= 0.27$, Verfallsfunktion Sigmoid, $\alpha_1 = 0.5$, $\alpha_2 = 8$.)

Ergebnisse zu kombinieren. Dies ist ratsam, weil in diesem Fall die unterschiedlichen Eigenschaften von Vorder- und Hintergrund unabhängig voneinander modelliert und repräsentiert werden. Die diskriminativen Eigenschaften der Modelle sind in diesem Fall i.d.R. besser.

Die einfachste Kombinationsmöglichkeit ist die Auswahl desjenigen Modells mit der höchsten assoziierten Wahrscheinlichkeit bzw. Güte (*Winner-takes-all*). Der Pixel wird dann derjenigen Klasse zugeordnet, die mit dem besten Modell assoziiert ist. Diese Vorgehensweise hat den grundlegenden Nachteil, dass die Verhältnisse der

einzelnen Modellgüten nicht betrachtet werden. Es ist jedoch intuitiv verständlich, dass eine Unterscheidung zwischen „sicheren“[4] und „unsicheren“[5] Entscheidungen wünschenswert ist.

Sei $\mathcal{V}$ ein Vordergrund-Modell und $\mathcal{H}$ ein Hintergrund-Modell. Die *normalisierte* Wahrscheinlichkeit, dass Pixel $b(x, y)$ Vordergrund ist, ergibt sich durch Anwendung der Bayes-Regel:

$$P(\mathcal{V}|b(x,y)) = \frac{P(b(x,y)|\mathcal{V})P(\mathcal{V})}{P(b(x,y)|\mathcal{V})P(\mathcal{V}) + P(b(x,y)|\mathcal{H})P(\mathcal{H})} \tag{6.11}$$

Dies erfordert die Verfügbarkeit der a-priori-Wahrscheinlichkeiten der Klassen $P(\mathcal{V}), P(\mathcal{H})$ sowie der klassenbedingten Wahrscheinlichkeiten, also eine vollständige probabilistische Modellierung. Die Verallgemeinerung hiervon für beliebige Gütewerte $G(...|b(x, y))$ ist das sog. *Softmax-Kriterium*:

$$\hat{P}(\mathcal{V}|b(x,y)) = \frac{G(\mathcal{V}|b(x,y))}{G(\mathcal{V}|b(x,y)) + G(\mathcal{H}|b(x,y))} \tag{6.12}$$

Beide Ausdrücke liefern eine kontinuierliche (Pseudo-) Wahrscheinlichkeit aus dem Intervall $[0, 1]$, die umso größer ist, je deutlicher die Entscheidung zugunsten des Vordergrund-Modelles ausfällt. Durch eine Schwellwertoperation auf diesen Wahrscheinlichkeiten können also unsichere Entscheidungen vermieden werden. Ein weiterer Vorteil ist, dass eine „harte“ binäre Entscheidung auf Pixelebene durch eine in fundierter Weise ermittelte Wahrscheinlichkeit ersetzt werden kann, was die Anwendung probabilistischer Methoden im weiteren Verlauf ermöglicht.

### 6.3.2 *Extraktion von Vordergrund-Regionen*

Die Vordergrund-Segmentierung liefert als Ergebnis zu jedem Zeitpunkt eine Maske in der Größe des Kamerabildes, in der die Werte der Pixel ihren jeweiligen Vordergrundwahrscheinlichkeiten entsprechen. Für eine effiziente Suchraumreduzierung müssen Regionen mit hoher Vordergrundwahrscheinlichkeit identifiziert und ihre Umrisse durch geometrische Primitive oder parametrische Hüllkurven beschrieben werden.

Für die beabsichtige Suchraumeinschränkung eines Personendetektors reichen umschließende Rechtecke aus. Ihre Berechnung sollte möglichst schnell und effizient sein. Weiterhin soll, um die Flexibilität möglichst wenig einzuschränken, keinerlei Vorwissen über Anzahl und Größe der gesuchten Regionen eingehen. Im Folgenden wird ein effizienter baumbasierter Algorithmus vorgestellt, der dies leistet.

4 ein Modell hat eine sehr viel höhere Güte als das andere
5 Alle Modelle haben ungefähr die gleiche Güte

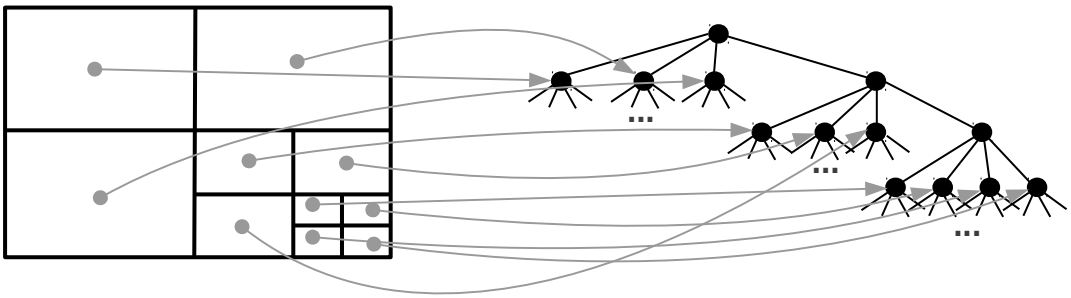

Abbildung 15: Schematischer Aufbau eines Quadtrees. Das Bild wird rekursiv in gleich große Viertel geteilt. Jede Ebene des Baumes unterhalb der Wurzel repräsentiert eine Ebene der Unterteilung und jeder Knoten des Baumes ist mit einer rechteckigen Region des Bildes assoziiert. Somit hat jeder Knoten genau vier Nachfolger.

Der vorgeschlagene Algorithmus ähnelt dem *Split and Merge* Segmentierungsalgorithmus [79] und basiert auf einer *Quadtree*-Zerlegung des Bildes. Hierbei wird ein Bild der Größe $w \times h$ in vier Bildbereiche der Größe $\frac{w}{2} \times \frac{h}{2}$ zerlegt. Diese werden wiederum rekursiv in ähnlicher Weise aufgeteilt, bis eine minimale Zellgröße erreicht ist (Abbildung 15). Jeder Knoten des resultierenden Baumes enthält die Koordinaten der Eckpunkte des Bildausschnittes, welchen er repräsentiert. Diese Struktur kann nun für eine effiziente Suche nach Vordergrundregionen genutzt werden, indem der Baum rekursiv nach Knoten durchsucht wird, deren durchschnittliche Vordergrundwahrscheinlichkeit hoch ist. Hierfür müssen Summen über rechteckige achsenparallele Bildbereiche berechnet werden. In Kapitel 5.2 wurde gezeigt, dass dies effizient mittels eines Integralbildes realisiert werden kann.

Der *Quadtree* benötigt und enthält keinerlei Bildinformationen mit Ausnahme der Bildgröße. Er kann somit für eine gegebene Bildgröße einmal konstruiert und beibehalten werden, solange die Größe sich nicht ändert. Für jedes neue Kamerabild wird das korrespondierende Integralbild berechnet. Anschließend wird der Suchbaum beginnend von der Wurzel rekursiv durchlaufen. Sobald in einem Zweig ein Knoten erreicht wird, dessen durchschnittlicher Vordergrundwert $\bar{m}$ über einem Schwellwert $\delta_m$ liegt und dessen Größe zwischen der gewünschten minimalen und maximalen Größe $s_{min}, s_{max}$ liegt, wird die Berechnung für diesen Zweig abgebrochen und die Koordinaten des entsprechenden Knotens werden gespeichert.

Das Resultat ist eine Liste $\Omega_R = \{\mathbf{r}_{R,i}, i = 1...n\}$, $\mathbf{r}_{R,i} = (x_i, y_i, w_i, h_i)$, mit Rechtecken unterschiedlicher Größe, die Vordergrundregionen umschließen (Abbildung 16). Dabei sind Lage, Form und Größe der Regionen beliebig. Die einzelnen Rechtecke

Abbildung 16: Beispiele für die Funktion der Quadtree-basierten Vordergrundextraktion. Erste und zweite Reihe: Eine Szene mit resultierenden Vordergrundkarten (adaptive HG-Subtraktion mit Vordergrundhistorie) und extrahierten Regionen (Rot: Rohe Ergebnisse; Grün: Umschließende Region nach Kombination). Trotz großer Lücken in der Vordergrundkarte werden beide Personen gut segmentiert. Dritte und vierte Reihe: Die Kamera schwenkt beim zweiten und dritten Bild nach rechts, weshalb das Hintergrundmodell versagt. Dies kann aufgrund der Größe der extrahierten Vordergrundregion detektiert werden. Nach Ende der Kamerabewegung adaptiert sich der Extraktionsalgorithmus wieder. (Parameter: $s_{min} = 4$, $s_{max} = 128$, $\delta_s = 0.5$)

werden in einem abschließenden Schritt zu umschließenden Rechtecken größerer zusammenhängender Regionen kombiniert. Dies geschieht durch einen einfachen Zusammenhangstest über die Eckpunkte aller Rechtecke in der Liste. Der Algorithmus ist sehr schnell und benötigt, abgesehen von den drei Parametern $\delta_m$, $s_{min}$ und $s_{max}$, keinerlei Vorwissen oder Annahmen über die zu findenden Regionen.

In der Praxis wird jedoch die Annahme getroffen, dass interessante Vordergrundregionen nur einen kleinen Teil des Bildes einnehmen. Dies ist insbesondere dann von Bedeutung, wenn aufgrund einer Kamerabewegung oder einer plötzlichen Änderung der Umgebungsbeleuchtung die adaptive Hintergrundmodellierung versagt. In einem

solchen Fall wird typischerweise für kurze Zeit das gesamte Bild oder ein großer Teil davon als Vordergrund segmentiert. Demzufolge ist eine Vordergrundregion, die annähernd die gleiche Größe hat wie das Kamerabild, ein Anzeichen für ein Versagen des Hintergrundmodelles. In diesem Fall wird das Hintergrundmodell reinitialisiert (Abbildung 16 unten).

### 6.3.3 *HOG-basierter MLP-Detektor*

Das Resultat der Vordergrundsegmentierung und Regionsextraktion ist eine Liste von Rechtecken um Regionen mit hoher mittlerer Vordergrundwahrscheinlichkeit. Diese können Personen enthalten, aber auch Objekte oder Szenebereiche, die sich bewegen oder durch die Modellierung schlecht erfasst wurden. Der nächste Schritt besteht in der Verifikation jeder Region in Hinsicht auf die Anwesenheit einer Person.

Die grundlegende Annahme, die hierbei getroffen wird, ist, dass Personen aufgrund fehlerhafter Segmentierung zwar aus mehreren nicht zusammenhängenden Regionen bestehen können[6], jedoch zumindest der Kopf oder Oberkörper einer Person innerhalb eines der umschließenden Rechtecke zu finden ist. Deshalb wird ein Detektor für Kopf-Schulter-Konturen eingesetzt. Dieser ist robuster gegenüber teilweiser Verdeckung der Person als ein Ganzkörperdetektor, beispielsweise wenn Beine oder Unterkörper durch Mobiliar verdeckt werden. Gegenüber den häufig eingesetzten Gesichtsdetektoren (vgl. Kapitel 4.4) bietet er den erheblichen Vorteil, dass die Detektionsleistung weniger stark von der relativen Orientierung der Person zur Kamera abhängt, weil die Form des Kopf-Schulter-Umrisses sich nur wenig ändert. Somit können auch Personen detektiert werden, welche der Kamera ihr Profil oder den Hinterkopf zuwenden. In solchen Fällen würde ein Gesichtsdetektor versagen.

Der Detektor benutzt HOG-Merkmale (vgl. Kapitel 5.3), die in einem gleitenden Fenster berechnet werden, dessen Mittelpunkt innerhalb der detektierten Vordergrundregionen liegen muss. Zur Klassifikation derartiger Merkmale werden, analog zur ursprünglichen Vorgehensweise in [37], häufig SVMs eingesetzt (z.B. [55, 102, 217]). Untersuchungen im Rahmen einer studentischen Projektgruppe [6] ergaben jedoch, dass sich mit MLP-Klassifikatoren (vgl. Kapitel 5.5) vergleichbare Ergebnisse bei größerer Effizienz erreichen lassen. Deshalb kommt in vorliegender Arbeit ein MLP als Klassifikator zum Einsatz.

Für die effiziente Berechnung der Merkmale wird ein Integralhistogramm (Kapitel 5.3.3) verwendet. Das gleitende Fenster wird in verschiedenen Größen angewendet,

6 Eine einfache Auswahl aufgrund heuristischer Regeln der Größe und Form von Regionen ist daher nicht robust anwendbar.

um Unabhängigkeit vom Abstand einer Person zur Kamera zu erreichen. Das im HOG-Deskriptor verwendete Gradientenhistogramm gewährleistet eine gewisse Invarianz gegenüber Translation und Skalierung. Deshalb kann der Skalierungsfaktor zwischen benachbarten Fenstergrößen verhältnismäßig groß gewählt werden und das Fenster muss nur alle $n$ Pixel mit $n > 1$ ausgewertet werden.

Der größte Nachteil des HOG-Detektors gegenüber beispielsweise dem flächenbasierten Viola-Jones Detektor [201] ist, dass der HOG-Deskriptor durch die Histogrammbildung in seinen Zellen nur schwache räumliche Zusammenhänge der Intensitätskanten codiert. Dies führt einerseits zwar zu einer guten Generalisierungsfähigkeit, andererseits aber zu einer vergleichsweise großen Anzahl von Fehldetektionen.

Die vorhergehende Vordergrundsegmentierung verringert die Anzahl an Fehldetektionen potentiell deutlich, weil große Bereiche des Bildes nicht betrachtet werden. Darüber hinaus kann für Detektoren, die mit gleitenden Fenstern arbeiten, folgende heuristische Annahme getroffen werden: Detektionen, die von Bildstrukturen stammen, deren Form der gesuchten Form sehr ähnlich ist, sind im Allgemeinen nicht isoliert. Aufgrund der Generalisierungsfähigkeiten von Merkmal und Klassifikator wird an benachbarten Suchpositionen und Skalierungsstufen eine Häufung von Detektionen auftreten. Eine korrekte Detektion wird also viele Nachbarn haben. Fehldetektionen, die an weniger ähnlichen Strukturen oder durch Bildrauschen auftreten, werden entsprechend wenige Nachbarn aufweisen. Die Güte einer Hypothese kann also durch die Anzahl benachbarter Detektionen abgeschätzt werden.

Zu diesem Zweck wird jeder Detektionskandidat durch einen Vektor $\mathbf{r}'_{D,i} = (x'_i, y'_i, w'_i)$ repräsentiert, mit den Pixelkoordinaten $x'_i, y'_i$ und der Suchfensterbreite $w'_i$. Alle Kandidaten werden anschließend per *Mean Shift* Clusteralgorithmus (vgl. Kapitel 5.4.2) zu Häufungsgebieten zusammengefasst. Diese werden dann durch den mittleren Parametervektor ihrer Mitglieder repräsentiert:

$$\mathbf{r}_{D,j} = \frac{1}{|\Psi_j|} \sum_{\mathbf{r}'_{D,i} \in \Psi_j} \mathbf{r}'_{D,i} \tag{6.13}$$

In obiger Formel bezeichnet $\mathbf{r}_{D,j}$ den Repräsentanten des j-ten Häufungsgebietes und $\Psi_j$ ist die Menge der zugeordneten Kandidaten.

Durch diesen Vorgang werden einerseits die „Wolken“ von Detektionskandidaten zu einzelnen Repräsentanten reduziert, was die Datenmenge für die Weiterverarbeitung verringert. Andererseits kann die Mächtigkeit $|\Psi_j|$ verwendet werden, um eine Anzahl guter Hypothesen auszuwählen und den Rest zu verwerfen. Abbildung 17 zeigt ein Beispiel. Das Resultat ist eine Liste von Detektionsrechtecken $\Omega_D = \{\mathbf{r}_{D,j}\}$.

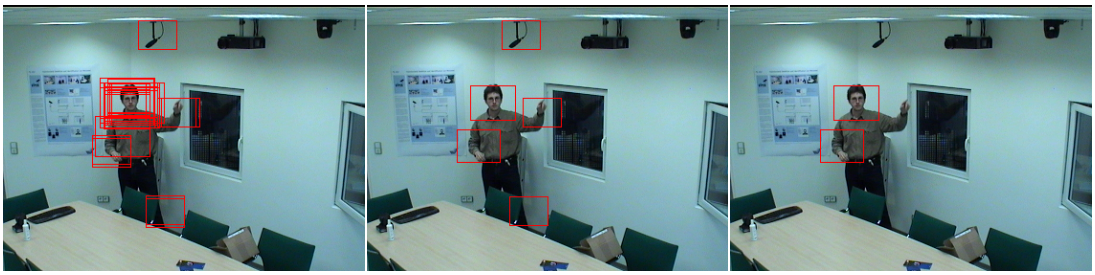

Abbildung 17: Beispiel für ein Detektionsergebnis des Kopf-Schulter Detektors (links), Cluster-Repräsentanten nach *Mean Shift* Clusterung (Mitte), Endergebnis nach Eliminierung aller Cluster mit $\Psi_j < 3$ (rechts).

## 6.4 PERSONENTRACKING

Der nächste Verarbeitungsschritt umfasst das Tracking einer detektierten Person. Dies dient einerseits zur weiteren Beschleunigung der Verarbeitung, andererseits zum Aufbau von online gelernten Vorder- und Hintergrundmodellen, um die Vordergrundsegmentierung zu verbessern. In der Literatur existiert eine Vielzahl visueller Trackingverfahren (vgl. z.B. [213]). In dieser Arbeit fiel die Wahl auf das in Kapitel 5.4.3 vorgestellte *Mean Shift* Trackingverfahren. Der Grund für diese Wahl ist einerseits die aus der Literatur bekannte gute Leistungsfähigkeit des Verfahrens (vgl. z.B. [66, 144, 212]) in Verbindung mit seiner sehr einfachen und effizienten Realisierung. Andererseits fügt es sich aus technischer Sicht nahtlos in das vorgeschlagene System ein: Das Tracking basiert auf Farbmodellen, wie sie in der Hintergrundmodellierung und auch im weiteren Verlauf für die Detektion von Hautfarbe verwendet werden. Zudem findet *Mean Shift* bereits an anderer Stelle Anwendung als Clustering-Verfahren.

Die Aufgabe des Trackers ist, ausgehend von der letzten bekannten Position ein Maximum in einer Vordergrund-Wahrscheinlichkeitskarte zu finden. Der Tracking-Algorithmus benutzt die Hypothese des *Mean Shift* Trackers, um den Suchbereich für den im vorherigen Kapitel vorgestellten Personendetektor weiter einzuschränken. Der Detektor wiederum verifiziert die Trackerhypothesen. Durch dieses Zusammenspiel werden auf effiziente Weise die Vorteile von Detektor und Tracker kombiniert: Der Tracker überbrückt Phasen, in denen der Detektor keine hinreichend guten Hypothesen liefert (z.B. durch zeitweise Verdeckung einer Person). Stehen gute Detektorhypothesen zur Verfügung, werden diese benutzt, um ein Trackerversagen zu detektieren. Abbildung 18 beschreibt das Personentracking. Die einzelnen Schritte werden im Folgenden erläutert.

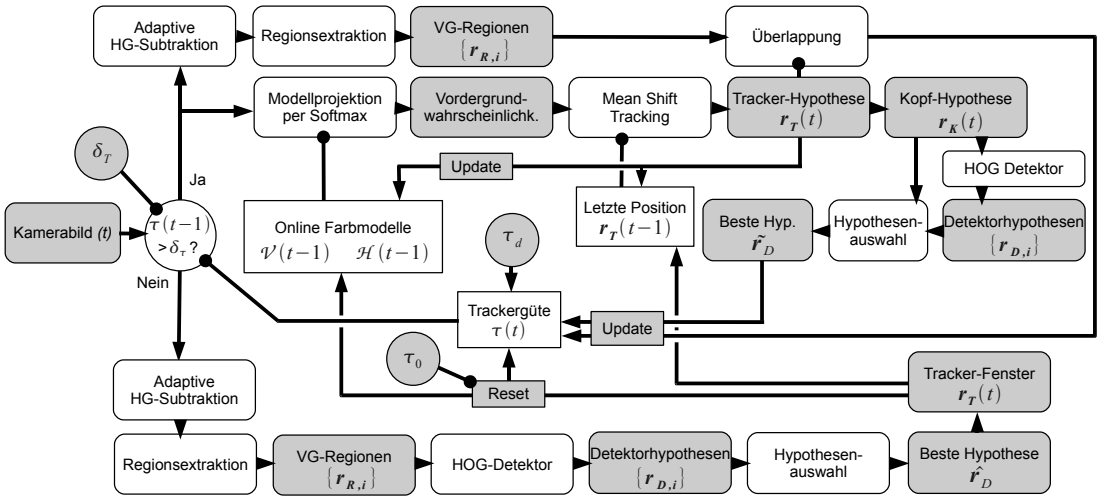


Abbildung 18: Grafische Darstellung des Personentrackings. Ausgefüllte gerundete Rechtecke sind Daten, gerundete Rechtecke Teilschritte des Algorithmus. Weiße Rechtecke bezeichnen Parameter und Modelle, ausgefüllte Rechtecke spezifizieren Operationen auf Daten. Ausgefüllte Kreise sind Konstanten. Pfeile bezeichnen den Daten- und Kontrollfluß, Verbindungen mit einem Kreis zeigen die Benutzung eines Datums im entsprechenden Teilschritt an. Zur Erklärung des Algorithmus siehe Text.

### 6.4.1 *Initialisierung des Trackers*

Der erste und entscheidende Schritt eines Trackingvorganges ist die Initialisierung des Trackers, d.h. die zu trackende Person muss zunächst lokalisiert werden. Anhand der Lokalisierung muss ein Modell der Person erstellt werden, welches dann in den folgenden Zeitschritten zum Tracking genutzt wird. Eine fehlerhafte Initialisierung führt demnach zu einem falschen Modell. Wenn im schlimmsten Fall eine falsche Lokalisierungshypothese zur Initialisierung benutzt wird, wird ein falsches Objekt oder ein Teil des Hintergrundes verfolgt.

Für die Initialisierung stehen die Ergebnisse des Kopf-Schulter-Detektors zur Verfügung. Aufgrund der zuvor erfolgten Suchraumeinschränkung sind dessen Hypothesen relativ verlässlich. Es besteht allerdings, wie bereits erwähnt, die Gefahr fehlerhafter Hypothesen, falls Objekte in der Szene eine Kopf-Schulter-ähnliche Form aufweisen. Dieser Gefahr wird mit der Annahme begegnet, dass Szeneobjekte statisch sind, während Personen sich häufig bewegen. Aus diesem Grund wird während der Initialisierungsphase die auf Bewegung reagierende adaptive Hintergrundsubtraktion

aus Kapitel 6.3.1 zur Hintergrundmodellierung genutzt. Sobald ein Bild mit gültigen Detektionen vorliegt, wird diejenige Hypothese $\hat{\mathbf{r}}_D$ mit der größten Anzahl an benachbarten Detektionen für die Initialisierung ausgewählt. Bei einem Trackerversagen wird zur Reinitialisierung auf die gleiche Weise verfahren.

Der *Mean Shift* Tracker hat den grundlegenden Nachteil, dass er versagt, wenn das zu trackende Objekt sich zwischen aufeinanderfolgenden Kamerabildern um mehr als seine eigene Größe bewegt. Aus diesem Grund sollte die zu verfolgende Region möglichst groß sein. Zudem wäre es wünschenswert, dass ihre Farbverteilung über eine große Variation des Blickwinkels stabil bleibt und das Verhältnis von Vordergrund- zu Hintergrundpixeln auch bei kleinen Lokalisierungsfehlern groß ist[7]. Die detektierte Kopf-Schulter-Region ist verhältnismäßig klein, enthält viele Hintergrundpixel und ihre Farbverteilung kann sich bei Rotation des Kopfes stark ändern (Gesicht/Haare). Aus diesen Gründen wird statt der Kopfregion der Oberkörper der Person verfolgt. Der Mittelpunkt $(x_T, y_T)$ und die Größe $(w_T, h_T)$ des Trackingfensters $\mathbf{r}_T$ wird anhand der Parameter der Kopf-Schulter Detektion $(\hat{x}_D, \hat{y}_D, \hat{w}_D, \hat{h}_D)$ wie folgt geschätzt:

$$\begin{aligned} \mathbf{r}_T &= (x_T, y_T, w_T, h_T) \\ x_T = \hat{x}_D - \mu_x \hat{w}_D, \quad y_T &= \hat{y}_D - \mu_y \hat{h}_D \\ w_T = \bar{w}_D, \quad h_T &= 2\bar{h}_D. \end{aligned} \tag{6.14}$$

Hierbei bezeichnen $\bar{w}_D$ und $\bar{h}_D$ die über ein zeitliches Fenster gemittelte Breite und Höhe des Kopf-Schulter Detektionsrechteckes[8]. Die Parameter $\mu_x, \mu_y$ bezeichnen den gleitenden Mittelwert des relativen Versatzes zwischen den Zentren des Detektions- und Trackerfensters. Sie werden zu Beginn mit heuristisch ermittelten Werten initialisiert. Zur Laufzeit können sie aus den Beobachtungen gelernt werden:

$$\begin{aligned} \hat{\mu}_x(t) &= \hat{\mu}_x(t-1) + \frac{(\hat{x}_D(t) - x_T(t))}{\hat{w}_D(t)} \\ \mu_x(t) &= \frac{\hat{\mu}_x(t)}{n_B(t)}. \end{aligned} \tag{6.15}$$

Hierbei ist $n_B(t)$ die Größe der bis zum Zeitpunkt $t$ beobachteten Stichprobe $\mathbf{B}(t)$. Für $\mu_y(t)$ wird analog vorgegangen.

7 Der Grund hierfür ist, dass das aus der Region gelernte Farbmodell möglichst gut zwischen Vorder- und Hintergrund differenzieren soll.

8 Für die erste Initialisierung ist keine zeitliche Historie vorhanden, weshalb direkt die Größe des Detektionsfensters verwendet wird.

### 6.4.2 *Lernen von online Farbmodellen*

Geht man von einer erfolgreichen Lokalisierung der zu trackenden Person aus, dann kann zur Laufzeit ein personalisiertes Modell der Farbverteilungen von Vorder- und Hintergrund gelernt werden. Dieses ist spezifischer und robuster – insbesondere auch gegen Kamerabewegungen – als die während der Initialisierung verwendete adaptive Hintergrundsubtraktion. Aus Effizienzgründen kommt hier ein Histogrammmodell zum Einsatz. Das Vordergrundmodell $\mathcal{V}(t)$ zum Zeitpunkt $t$ wird kumulativ aus den Farbwerten der Pixel innerhalb des Trackerfensters berechnet. Damit das Modell gegen leicht fehlerhafte Lokalisierungen des Trackerfensters robust ist, werden die Beiträge der Pixel mit einem monoton fallenden Kernel $\mathcal{K}$ gewichtet:

$$\mathcal{V}(t) = (v_i(t)), \quad v_i(t) = v_i(t-1) + \sum_{\mathbf{x}\in\mathbf{r}_T} \mathcal{K}(\mathbf{x} - \bar{\mathbf{r}}_T)\delta(f_h(b(\mathbf{x})) - i). \tag{6.16}$$

Hierbei ist $\bar{\mathbf{r}}_T$ der Mittelpunkt des Trackerfensters. Die Bandbreite des Kernels wird für jeden Zeitschritt an die Größe des Trackerfensters angepasst.

Das Hintergrundmodell $\mathcal{H}(t)$ verwendet keinen Kernel und wird ebenfalls kumulativ aus den Hintergrundpixeln berechnet. Zu diesem Zweck wird zu jedem Zeitschritt eine feste Anzahl von Pixeln zufällig ausgewählt, die außerhalb aller extrahierten Vordergrundregionen liegen. Die Vordergrund-Wahrscheinlichkeitskarte, die aus den so ermittelten Modellen per Softmax-Kriterium (6.12) berechnet wird, dient im Zeitschritt $t+1$ als Eingabe für den *Mean Shift* Trackingalgorithmus.

### 6.4.3 *Trackerunterstützte Auswahl von Kopfhypothesen*

Der Tracker verfolgt den Torso der Person. Für die weitere Verarbeitung wird aber die Position ihres Kopfes benötigt. Aufgrund der fehlerbehafteten Natur der automatischen Trackerinitialisierung und des Trackings selbst wäre eine Generierung von Kopfhypothesen allein basierend auf den Trackingergebnissen sehr fehleranfällig. Deshalb wird der Tracker nur unterstützend eingesetzt, um aus einer Menge von Detektorhypothesen die beste(n) auszuwählen.

Anhand der Position und Größe des Trackerfensters und der gelernten Parameter $\mu_x, \mu_y$ lässt sich analog zu (6.15) ein Kopf-Schulter-Detektionsfenster $\mathbf{r}_K = (\mathbf{x}_K, w_K, h_K)$ hypothetisieren. Der Suchbereich für den Detektor wird auf ein Fenster festgelegt, welches um den Mittelpunkt der Hypothese zentriert ist. Seine Größe wird gewählt gemäß

$$w_K = k\sigma_x w_H, \quad h_K = k\sigma_y h_H. \tag{6.17}$$

Hierbei ist $(w_H, h_H)$ die Größe der Hypothese, $k$ ist ein Skalierungsfaktor und $\sigma_x, \sigma_y$ sind die Standardabweichungen des Abstandes zwischen der besten Detektorhypothese $\hat{\mathbf{r}}_D = (\hat{x}_D, \hat{y}_D, \hat{w}_D)$ und der Trackerhypothese, bezogen auf die jeweilige Größe des Detektionsfensters. Die Standardabweichungen können zur Laufzeit aus den Beobachtungen gelernt werden. Aus Gründen der Robustheit werden ihre Werte nach unten beschränkt, so dass das resultierende Suchfenster nicht zu klein wird:

$$\sigma_x = max\left(\sigma_{x,min}, \sqrt{Var_B\left(\frac{\hat{x}_D - x_H}{w_H}\right)}\right). \tag{6.18}$$

Hierbei bezeichnet $Var_B(...)$ die Stichprobenvarianz über die bis zum Zeitpunkt $t$ beobachtete Stichprobe $\mathbf{B}$. Die Bestimmung von $\sigma_y$ erfolgt in gleicher Weise. Der resultierende Suchbereich ist typischerweise deutlich kleiner als der durch die Bewegungsdetektion ermittelte, weshalb das Tracking an dieser Stelle eine Beschleunigung der Suche bewirkt.

In der Folge wird der Detektor innerhalb des Suchfensters angewendet und seine Hypothesen werden wie oben beschrieben geclustert. Die anhand des Trackerfensters hypothetisierte Detektion wird mit der Clustergröße eins in die Hypothesenliste eingefügt. Die Güte $\gamma_j$ der j-ten Detektorhypothese $\mathbf{r}_{D,j}$ ergibt sich dann zu

$$\gamma_j = \frac{|\Psi_j|}{N} \cdot \mathcal{N}(\mathbf{x}_j - \mathbf{x}_K, \Sigma), \quad \Sigma = diag(\sigma_x^2, \sigma_y^2), \tag{6.19}$$

mit der Gesamtanzahl an Detektorhypothesen vor der Clusterung $N$, der Mächtigkeit des j-ten Clusters $|\Psi_j|$ und dem Mittelpunkt der Hypothese $\mathbf{x}_j$. Die Güte einer Detektion hängt somit von ihrer Clustergröße und ihrer Abweichung von der durch den Tracker generierten Hypothese ab. Die Hypothese mit der größten Güte liefert die getrackte Kopfposition $\tilde{\mathbf{r}}_D$. Es ist offensichtlich, dass die Trackerhypothese (mit $|\Psi_T| = 1$) nur dann ausgewählt wird, wenn die Detektorhypothesen entweder nicht verlässlich sind (d.h. jeweils nur sehr wenige Nachbarn haben), gar nicht existieren oder sehr weit von der hypothetisierten Position entfernt sind. Ansonsten wird den Hypothesen des Detektors vertraut.

### 6.4.4 *Detektion von Trackerversagen*

Ein zentrales Problem bei adaptiven Trackern ist die Detektion eines Trackerversagens. Das zum Tracking verwendete Modell selbst ist hierfür wenig aufschlußreich, weil es sich eben auch an Fehler adaptiert. Aus diesem Grund werden weiterhin der Kopf-Schulter Detektor sowie die Bewegungsdetektion in jedem neuen Kamerabild

angewendet, um die Trackerhypothesen zu verifizieren. Mit dem Tracker ist ein Gütewert $\beta$ assoziiert, der in jedem Zeitschritt entsprechend den Ergebnissen der Verifikation modifiziert wird. Fällt dieser unter einen Schwellwert $\delta_\beta$, so werden der Tracker und die Online-Farbmodelle neu initialisiert. Ein Tracker, der eine hohe Güte aufweist, wird somit auch dann für einige Zeit das Ziel weiter verfolgen, wenn seine Hypothese stark von denen der anderen Modalitäten abweicht. Bei einer niedrigen Güte wird der Tracker hingegen sehr schnell zurückgesetzt und es wird eher den Ergebnissen des Detektors vertraut.

Der Gütewert $\beta(t)$ wird bei jeder Initialisierung des Trackers auf einen festen Startwert $\beta_0$ gesetzt. Für jedes neue Bild wird der Gütewert von drei Parametern beeinflusst:

- Ein konstanter Verfallsfaktor $\beta_d$. Dieser sorgt dafür, dass die Güte des Trackers verfällt, falls seine Hypothesen über längere Zeit wegen fehlender Detektorhypothesen nicht verifiziert werden können (z.B. weil er eine Hintergrundregion verfolgt und die Person sich komplett außerhalb des Suchbereiches befindet).

- Ein Faktor $\beta_u(t)$, der aus der Überlappung des Trackerfensters mit durch die Bewegungsdetektion extrahierten Vordergrundregionen berechnet wird:

$$\beta_u(t) = \begin{cases} 0 & \text{falls} \quad \Omega_R = \emptyset \\ \beta_m \cdot \alpha \frac{\sum_k A(\mathbf{r}_R(t) \cap \mathbf{r}_T(t))}{A(\mathbf{r}_T(t))} - \beta_m & \text{sonst} \end{cases} \tag{6.20}$$

  Dieser integriert die Ergebnisse der Bewegungsdetektion in die Trackerverifikation. Hierbei bezeichnet $A(...)$ die Fläche eines Rechteckes und $\alpha$ ist ein Gewichtungsfaktor, der festlegt, wie viel Bedeutung der Bewegungsdetektion beigemessen wird.

- Ein Faktor $\beta_h(t)$, der von der Güte $\tilde{\gamma}(t)$ der ausgewählten besten Kopfhypothese $\tilde{\mathbf{r}}_D(t)$ abhängt:

$$\beta_h(t) = \begin{cases} -\beta_m & \text{falls} \quad \Omega_D = \{\} \\ 0 & \text{falls} \quad \tilde{\mathbf{r}}_D(t) = \mathbf{r}_K(t) \\ \beta_m(2 * \tilde{\gamma}(t) - 1) & \text{sonst} \end{cases} \tag{6.21}$$

  Somit wird die Trackergüte im Falle einer guten positionellen Übereinstimmung zwischen $\tilde{\mathbf{r}}_D(t)$ und $\mathbf{r}_K(t)$ (d.h. $\tilde{\gamma}(t) > 0.5$) erhöht, ansonsten verringert. Falls keine einzige Detektorhypothese vorliegt, ist die Trackerhypothese nicht verifiziert

und die Trackergüte wird um den maximal möglichen Wert verringert. Wird die trackergenerierte Hypothese als beste Hypothese gewählt, hat dies keinen Einfluß auf die Güte[9].

$\beta_m$ ist jeweils eine Konstante, mit der sich die Sensitivität der Güteänderung einstellen lässt: Je größer $\beta_m$ ist, umso schneller nimmt die Güte bei positiver Verifikation zu und umso schneller verfällt sie bei negativer Verifikation. Die Gesamtgüte nach jedem Zeitschritt ergibt sich dann zu

$$\beta(t) = min(1, max(0, \beta(t-1) + \beta_h(t) + \beta_u(t) - \beta_d)). \tag{6.22}$$

Abbildung 19 zeigt einen beispielhaften Verlauf der Trackergüte anhand einer Beispielsequenz. Einige Bilder dieser Sequenz zusammen mit den Trackingergebnissen und den aus den online Farbmodellen generierten Vordergrundkarten sind in Abbildung 20 zu sehen.

## 6.5 HANDDETEKTION

Die Bewegungstrajektorie einer Hand kann zur Klassifikation dynamischer Armgesten verwendet werden, und statische Zeichen können durch Handposturen repräsentiert werden. Demzufolge umfasst der nächste Schritt in der Einzelbildverarbeitung die Detektion von Hypothesen für die Positionen von Händen in den Kamerabildern. Dies ist aus verschiedenen Gründen ein schwieriges Problem. Zum Einen sind Hände artikulierte Objekte mit vielen Freiheitsgraden, deren Form und Erscheinungsbild sich sehr schnell sehr stark ändern kann. Eine Modellierung mit gestaltbasierten Modellen oder Umrissen erscheint somit nicht erfolgversprechend oder würde eine große Anzahl sehr flexibler (und somit fehleranfälliger) Modelle erfordern. Zum Anderen können Hände sich während dynamischer Gesten schnell bewegen und abrupt ihre Bewegungsrichtung ändern. Die dadurch entstehende Bewegungsunschärfe in den Kamerabildern erschwert wiederum eine gestaltbasierte Erkennung. Zudem ist das *Tracking* der Handkandidaten bei üblichen Bildwiederholraten problematisch, weil ihre Positionen in aufeinanderfolgenden Bildern sich drastisch ändern können und eine zuverlässige Vorhersage ihrer Bewegung mit einem deterministischen Bewegungsmodell sehr schwierig ist. Ansätze zum Handtracking (vgl. z.B. [12, 35, 50, 77, 132, 143]) gehen deshalb typischerweise von einem eingeschränkten Szenario oder relativ langsamen Bewegungen aus.

9 Sonst würde eine aus dem Modell generierte und somit per Definition gute Hypothese benutzt werden, um das selbe Modell zu bewerten.

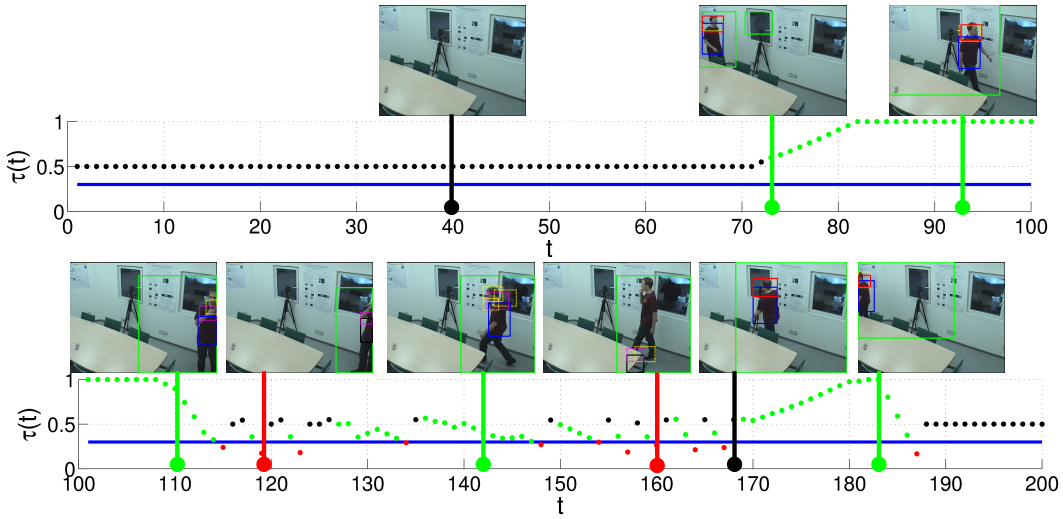


Abbildung 19: Verlauf des Tracker-Gütewertes über eine Sequenz von 200 Bildern. Schwarze Punkte: Kein Tracking, nur Detektor. Grüne Punkte: Tracker aktiv. Rote Punkte: Reinitialisierung. Blaue Linie: Schwellwert $\delta_\beta = 0.3$. Die Person betritt die Szene in Bild 71, wird detektiert und bis Bild 110 erfolgreich verfolgt. Danach verlässt sie die Szene am rechten Bildrand und bleibt dort teilweise verdeckt stehen. Bis ca. Bild 135 gibt es keine stabile Detektion, weshalb die Trackergüte niedrig bleibt. Bis Bild 158 betritt die Person die Szene wieder und bewegt sich sprunghaft vor- und rückwärts. Dieser schnellen Bewegung kann der Tracker schlecht folgen und verfolgt – auch aufgrund von Fehldetektionen des Detektors – zeitweise eine Hintergrundregion. Ab Bild 168 wird der Tracker wieder mit einer korrekten Detektion initialisiert und verfolgt die Person zuverlässig, bis sie die Szene verlässt. Das Verschwinden der Person wird registriert und der Tracker wird deaktiviert. ($\beta_0 = 0.5$, $\beta_m = 0.1$, $\beta_d = 0.1$)

In vorliegendem Szenario kommen weitere Schwierigkeiten hinzu: Durch die weitgehend uneingeschränkte Pose der Person in der Szene ergeben sich aufgrund des unbekannten Blickwinkels weitere Freiheitsgrade für die Gestalt der Hand. Zudem sind Hände vergleichsweise kleine Objekte, die in einem Kamerabild ggf. nur wenige Pixel groß sind. Zwar könnten die aktiven Kameras benutzt werden, um beispielsweise durch zoomen bessere Ansichten der Hände zu erhalten (z.B. zur Klassifikation von Handposturen), aber dafür müssen deren Positionen bekannt sein.

Im folgenden werden drei Möglichkeiten zur Handdetektion vorgestellt. Eine basiert auf *Keypoint*-Detektion und lokalen strukturellen Merkmalen, die zweite verwendet die

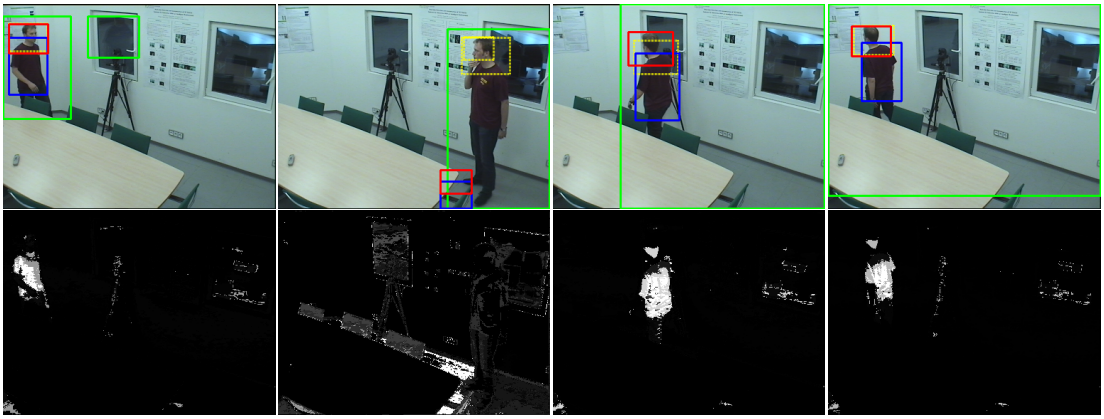

Abbildung 20: Beispiele für Trackingergebnisse. Bilder 73, 158, 170 und 176 der Sequenz (vgl. Abbildung 19). Grüne Rechtecke: Bewegungsregionen. Blaue Rechtecke: Trackerregion. Gelbe Rechtecke: Kopfhypothesen, die ausgewählte Hypothese ist rot markiert. Das erste Bild zeigt den Zustand kurz nach der ersten Initialisierung. Im zweiten Bild schlägt eine Reinitialisierung aufgrund einer falschen Detektorhypothese fehl. Dies wird bereits zwei Zeitschritte später erkannt. Auf den beiden letzten Bildern wurde der Tracker korrekt reinitialisiert und verfolgt wieder die Person. Die hypothetisierte Kopfposition (gelbes Rechteck) gleicht sich aufgrund des Lernens von $\mu_x, \mu_y$ der tatsächlichen Position an. Zweite Reihe: Die zugehörigen Vordergrundwahrscheinlichkeiten, auf denen der Tracker operiert. Sie werden mittels *Softmax*-Kriterium aus den im vorherigen Zeitschritt ermittelten Farbmodellen berechnet.

schon bekannte Bewegungsdetektion und ein Hautfarbmodell, und die dritte Variante kombiniert *Keypoint*-Deskriptoren mit Hautfarbe.

### 6.5.1 *Detektion mit SIFT*

Wenn die Modellierung einer Hand als Ganzes aufgrund ihrer Artikuliertheit nicht sinnvoll erscheint, dann ist es nahe liegend, stattdessen einzelne Teile einer Hand und deren örtliche Zusammenhänge zu modellieren (vgl. z.B. den *Elastic Bunch Graph* Ansatz in [196]). Auch hier stellt aber die Beweglichkeit der Hand und ihre vielen Freiheitsgrade ein Problem dar, weil die spatialen Beziehungen zwischen Teilen wenig eingeschränkt sind und einzelne Teile häufig aufgrund von Verdeckungen nicht sichtbar sind. Deshalb wird im Folgenden eine Hand als nur lose verbundene Kollektion charakteristischer Regionen betrachtet. Die Detektion dieser Regionen erfolgt anhand

von *Keypoints*, ihre Beschreibung entsprechend mit einem lokal invarianten Deskriptor. Die örtlichen Zusammenhänge zwischen den Regionen werden auf einfache Nachbarschaftsbeziehungen reduziert. Die Anwesenheit einer Hand wird somit durch das Vorhandensein einer gewissen Anzahl dieser charakteristischen Regionen in enger Nachbarschaft angezeigt.

Für die Detektion und Beschreibung von *Keypoints* wird der SIFT-Algorithmus verwendet. Der Ansatz zur Handetektion folgt weitgehend dem in Kapitel 5.1 beschriebenen Vorgehen, mit einer Ausnahme: Im originalen Algorithmus votieren die *Keypoint*-Deskriptoren per generalisierter Hough-Transformation gruppenweise für die Anwesenheit eines Objektes. Dieses Vorgehen funktioniert bei nichtartikulierten starren Objekten sehr gut, weil angenommen werden kann, dass die relativen Posen der *Keypoints* sich durch eine Posenänderung des Objektes nicht ändern. Befinden sich die *Keypoints* aber auf unterschiedlichen gegeneinander beweglichen Teilen eines artikulierten Objektes, wie im Falle der Handdetektion, ist diese grundlegende Annahme verletzt. Die Identifikation von Lokalisierungshypothesen wird aus diesem Grund durch ein hystereseähnliches Filterverfahren realisiert. Der gesamte Handdetektionsalgorithmus mit SIFT ist in Alg. 3 dargestellt.

In der Trainingsphase werden SIFT-Deskriptoren aus einer Menge von manuell annotierten Trainingsbildern extrahiert und jeweils eine Datenbank für Vorder- und Hintergrundbeispiele erstellt. Während der Erkennungsphase werden die Deskriptoren des Eingangsbildes zunächst mit den gespeicherten Beispielen abgeglichen, indem jeweils der nächste Nachbar – gemäß des euklidischen Abstandes der Deskriptoren – in der Vorder- und Hintergrunddatenbank gesucht wird. Zur effizienten NN-Suche kommt an dieser Stelle ein *kd-Suchbaum* [10] zum Einsatz. Das Verhältnis der zugehörigen Distanzen ist das Entscheidungskriterium für die Klassifikation. Dieses Vorgehen ist auch Teil des originalen SIFT-Algorithmus und liegt darin begründet, dass die Entscheidung für eine Klasse umso sicherer ist, je unterschiedlicher die beiden Distanzen sind. Es werden also – abhängig vom Klassifikationsschwellwert – nur „sichere“ Klassifikationsentscheidungen getroffen. Dennoch erhält man als Resultat i.d.R. eine hohe Anzahl von fälschlicherweise als Vordergrund klassifizierten *Keypoints*, weil der Vergleich von einzelnen Deskriptoren nicht sehr robust gegen Störungen ist. Die Vordergrundkandidaten werden deshalb anschließend anhand ihrer Nachbarschaftsbeziehungen gefiltert. Hierfür muss zunächst eine Nachbarschaft von *Keypoints* definiert werden. In vorliegendem Algorithmus wurden drei Möglichkeiten realisiert:

- Regionsbasiert mit fester Regionsgröße: Um den betrachteten *Keypoint* wird eine kreisförmige Region mit festem Radius gelegt. Alle anderen *Keypoints* innerhalb

**Algorithmus 3** Algorithmische Beschreibung der Handdetektion mittels SIFT. Siehe Text für Beschreibung. Verschiedene Möglichkeiten der Realisierung von *getKeypoint-List()* werden im Text erläutert.

**Erkennungsphase:**
**Eingabe:** Aktuelles Bild $\mathbf{B}(t)$, in Trainingsphase aus annotierten Beispielen generierte Exemplardatenbanken für Vordergrund $\Gamma^+$ und Hintergrund $\Gamma^-$.
Handkandidaten $\Upsilon^+ \leftarrow \emptyset$, Hintergrundkandidaten $\Upsilon^- \leftarrow \emptyset$.
Extrahiere Keypoints $\mathbf{x}_t^k$ und Deskriptoren $\mathbf{d}_t^k$.
**for** $(\forall \mathbf{d}_t^i)$ **do**
  Finde jeweils nächsten Nachbarn mit Distanz $\Delta^+$ in $\Gamma^+$ bzw. $\Delta^-$ in $\Gamma^-$.
  **if** $(\frac{\Delta^+}{\Delta^-} \leqslant \delta_d)$ **then**
    $\Upsilon^+ \leftarrow \Upsilon^+ \cup \mathbf{x}_t^i$.
  **else**
    $\Upsilon^- \leftarrow \Upsilon^- \cup \mathbf{x}_t^i$.
  **end if**
**end for**
filterCandidates($\Upsilon^+$, $\Upsilon^-$).
Ausgabe: Gefilterte Liste von Handkandidaten $\tilde{\Upsilon}^+$.

**function filterCandidates($\Upsilon^+$, $\Upsilon^-$)**
$\tilde{\Upsilon}^+ \leftarrow \emptyset$, $\Upsilon^* \leftarrow \emptyset$
**for** $(\mathbf{x}^i \in \Upsilon^+)$ **do**
  list = getKeypointList($\mathbf{x}_i$).
  Ermittle Anzahl Hand- und Hintergrundkandidaten $n^+$, $n^-$ in list.
  **if** $(n^+ \geqslant n^+_{min})$ **and** $(\frac{n^+}{n^-} \geqslant f_{min})$ **then**
    $\tilde{\Upsilon}^+ \leftarrow \tilde{\Upsilon}^+ \cup \mathbf{x}^i$.
  **else if** $(n^+ < n^+_{min})$ **and** $(\frac{n^+}{n^-} < f_{min})$ **then**
    Verwerfe $\mathbf{x}^i$.
  **else**
    $\Upsilon^* \leftarrow \Upsilon^* \cup \mathbf{x}^i$.
  **end if**
**end for**
**for** $(\mathbf{x}^j \in \Upsilon^*)$ **do**
  Finde $m$ nächste Nachbarn $\mathbf{x}_j^{1\ldots m} \in (\Upsilon^+ \cup \Upsilon^-)$.
  **if** $(\forall \mathbf{x}_j^{1\ldots m} : \mathbf{x}_j^k \in \tilde{\Upsilon}^+)$ **then**
    $\tilde{\Upsilon}^+ \leftarrow \tilde{\Upsilon}^+ \cup \mathbf{x}^j$.
  **end if**
**end for**

dieser Region sind benachbart. Hierbei wird angenommen, dass die Größe einer Handregion weitgehend konstant bleibt.

- Regionsbasiert mit skalenabhängiger Regionsgröße: Der Radius der Nachbarschaftsregion wird als ein Vielfaches der *Keypoint*-Skalierung gewählt. Ein *Keypoint*, der auf einer größeren Skalierung detektiert wurde, erhält somit auch eine größere Nachbarschaftsregion.

- k nächste Nachbarn: Anstatt eine feste Nachbarschaftsregion vorzugeben, werden die $k$ räumlich nächsten *Keypoints* als Nachbarn ausgewählt. Dieses Vorgehen trifft keine impliziten Annahmen über Regionsgröße und -form, verlässt sich stattdessen aber auf eine annähernd gleichbleibende *Keypoint*-Dichte.

Die auf diese Weise extrahierten Nachbarschaftslisten werden nun einem Hysteresähnlichen Test unterzogen. Hierfür wird zunächst die Anzahl $n^+$ von Vordergrundkandidaten und die Anzahl $n^-$ von Hintergrundkandidaten in der Nachbarschaft ermittelt. Wenn die Anzahl von Vordergrundkandidaten größer ist als ein Schwellwert $n^+_{min}$ **und** das Verhältnis $\frac{n^+}{n^-}$ größer ist als ein Schwellwert $f_{min}$, dann wird der betrachtete Kandidat als Vordergrund klassifiziert. Sind **beide** Kriterien verletzt, wird der Kandidat zurückgewiesen. Erfüllt er nur eines der beiden Kriterien, wird abschließend ein Zusammenhangstest über seine Nachbarn vorgenommen: Sind alle $m$ räumlich nächsten Nachbarn akzeptierte Vordergrund-*Keypoints*, wird der Kandidat akzeptiert, ansonsten verworfen. Dieses Filterverfahren bevorzugt kompakte Häufungsgebiete von Vordergrundkandidaten mit wenigen Hintergrundkandidaten dazwischen (vgl. Abbildung21). Diese Häufungsgebiete können in einem letzten Verarbeitungsschritt durch eine *Clusterung* zu einzelnen Hypothesen $\mathbf{r}_{H,i}$ zusammengefasst werden.

### 6.5.2 *Detektion mit Hautfarbe und Bewegung*

Eine Alternative zur strukturbasierten Handdetektion, die weit verbreitet ist und häufig erfolgreich eingesetzt wurde (vgl. Kapitel 4.4.1), ist die Verwendung eines Modelles der Hautfarbe. Der Vorteil dieser Vorgehensweise ist, dass Farbe ein pixelbasiertes Merkmal ist, für das die Form des betrachteten Objektes keine Rolle spielt. Sie ist also unabhängig von der Artikulation, Größe und Auflösung des Objektes. Eine Unabhängigkeit von Bewegungsunschärfe ist nur zum Teil gegeben, weil sich bei sehr schnellen Bewegungen während der Belichtungszeit einer Kamera die Farben von Vorder- und Hintergrund vermischen können.

In der Literatur wurden viele verschiedene Modellierungsarten für Hautfarbe vorgeschlagen (vgl. z.B. [87]), und sowohl statische wie auch dynamische bzw. online

Abbildung 21: Beispiele für die Handdetektion mit SIFT. Rote Punkte kennzeichnen zurückgewiesene *Keypoints*, grüne Punkte sind akzeptierte Vordergrundkandidaten. Selbst bei vergleichsweise schlechten Ergebnissen (rechtes Bild) befindet sich die überwiegende Mehrzahl der Fehldetektionen auf dem Vordergrund. Die meisten Fehldetektionen können durch andere Modalitäten, wie z.B. Hautfarbe, ausgeschlossen werden.

gelernte Modelle wurden erfolgreich eingesetzt (vgl. Kapitel 4.4.1). Die meisten Ansätze mit online Hautfarbmodellen verwenden für die Modellinitialisierung einen Gesichtsdetektor (vgl. z.B. [3, 132]). Nimmt man an, dass ein Gesicht korrekt detektiert wurde, so sind Pixel innerhalb der Gesichtsregion mit großer Wahrscheinlichkeit hautfarbig. Dieser Ansatz ist im vorliegenden Szenario nur bedingt anwendbar. Zwar könnten die Ergebnisse der Personendetektion genutzt werden, um Rückschlüsse auf die Position des Gesichtes zu ziehen, eine derartige Schätzung wäre aber ungenau und fehleranfällig. Zudem wird explizit nicht angenommen, dass die Person der Kamera immer frontal zugewandt ist. Folglich ist nicht notwendigerweise immer eine große Anzahl von Hautpixeln sichtbar. Deshalb wird eine Kombination aus einem statisch offline trainierten Modell mit einem online gelernten Histogrammmodell (vgl. Kapitel 6.3.1) und einem selektiven Adaptionsansatz verwendet.

Sei $P(\mathcal{V}|b(t,\mathbf{x}))$ die Hautfarbwahrscheinlichkeit des statisch trainierten Modelles für Pixel $b(\mathbf{x})$ zum Zeitpunkt $t$. Diese kann direkt als Gewichtung bei der Initialisierung des online Modelles verwendet werden:

$$\begin{aligned} \mathcal{V}(t) &= (v_i(t)) \\ v_i(t) &= v_i(t-1) + \sum_{\mathbf{x}\in\tilde{\mathbf{r}}_D(t)} (\mathcal{K}(\mathbf{x}-\tilde{\mathbf{x}}_D(t)) \cdot P(\mathcal{V}|b(t,\mathbf{x})) \cdot \delta(f_h(b(t,\mathbf{x}))-i)) \,. \end{aligned} \tag{6.23}$$

Der Kernel $\mathcal{K}(...)$ ist eine an die Größe der aktuellen Kopfhypothese $\tilde{\mathbf{r}}_D(t)$ angepasste Gewichtungsfunktion, zentriert am Mittelpunkt der Kopfhypothese $\tilde{\mathbf{x}}_D(t)$. Ähnlich wie beim Lernen des Farbmodelles für das Personentracking repräsentiert $\mathcal{K}(...)$ die durch die Kopfdetektion gegebenen Einschränkungen bezüglich der Lage der

hautfarbenen Region. In der Praxis hat sich hier ein Gauss'scher Kernel mit diagonaler Kovarianzmatrix als gut geeignet erwiesen. Die Varianzen werden so gewählt, dass die Standardabweichungen einem Sechstel der Höhe von $\tilde{\mathbf{r}}_D(t)$ entsprechen. Die Hautfarbwahrscheinlichkeit $P(\mathcal{S}|b(t,\mathbf{x}))$ ergibt sich dann aus $\mathcal{V}(t)$ zu

$$P(\mathcal{S}|b(t,\mathbf{x})) = \sum_i (\tilde{v}_i(t) \cdot \delta(f_h(b(t,\mathbf{x})) - i)), \tag{6.24}$$

mit den normierten Histogrammeinträgen $\tilde{v}_i(t)$.

Als Modellierungsart für das statische Hautfarbmodell wurden einerseits die bereits in Kapitel 6.3.1 eingeführten GMM untersucht, andererseits zum Vergleich der in der Literatur weit verbreitete sog. *Skin Locus* (vgl. z.B. [156, 186, 191]). Die Grundannahme hierbei ist, dass hautfarbene Pixel – auch von verschiedenen Ethnien und in unterschiedlichen Beleuchtungsszenarien – in einem geeigneten Farbraum ein einzelnes, eindeutig und eng begrenztes Häufungsgebiet bilden. Durch eine parametrische Beschreibung dieses Häufungsgebietes mit Hüllkurven erhält man einen sehr effizienten Klassifikator. Als Farbraum wird üblicherweise ein normalisierter chromatischer Farbraum, z.B. der normalisierte RG-Farbraum (nRG), benutzt. Anhand einer Trainingsstichprobe wird zunächst ein Histogramm der Farbverteilungen (vgl. Kapitel 6.3.1) mit relativ feiner Auflösung berechnet. Im Falle des nRG-Farbraumes ist dieses Histogramm zweidimensional (vgl. Abbildung 22). Durch Einstellen eines geeigneten Schwellwertes wird das Häufungsgebiet der Hautfarbpixel extrahiert, an das anschließend durch Polynomregression eine Hüllkurve – aus Effizienzgründen üblicherweise ein Polynom niedrigen Grades – angepasst wird. Zur Laufzeit ergibt sich die Klassifikation durch einen einfachen Test der Lage des Farbwertes relativ zur Hüllkurve. In diesem Fall ist $P(\mathcal{V}|b(t,\mathbf{x}))$ eine binäre Funktion.

Der größte Nachteil einer rein auf Hautfarbe basierenden Handdetektion ist, dass hautfarbene Objekte des Hintergrundes nicht von hautfarbenen Vordergrundobjekten unterschieden werden können. Wird ein hinreichend flexibles Hautfarbmodell benutzt, das unter verschiedenen Beleuchtungsbedingungen und für verschiedene Hauttypen funktioniert, ist mit einer großen Zahl an Fehldetektionen zu rechnen. Eine einfache Lösung für dieses Problem ist die Annahme, dass eine Hand ein bewegtes Objekt ist[10], während Hintergrundobjekte meistens statisch sind. Eine Vordergrundmaske $\mathbf{M}(\mathbf{x})$, die vorwiegend bewegte Objekte segmentiert, steht bereits aus der Hintergrundmodellierung bzw. der Personendetektion zur Verfügung. Diese kann – unter der Annahme statistischer Unabhängigkeit – ohne zusätzliche Kosten integriert werden:

$$P(\mathcal{S},\mathcal{M}|b(\mathbf{x})) = P(\mathcal{S}|b(\mathbf{x})) \cdot P(\mathcal{M}|b(\mathbf{x})). \tag{6.25}$$

10 Diese Annahme ist bei der Erkennung dynamischer Gesten offensichtlich erfüllt, weil eine relevante dynamische Geste gerade durch eine Bewegung definiert ist.

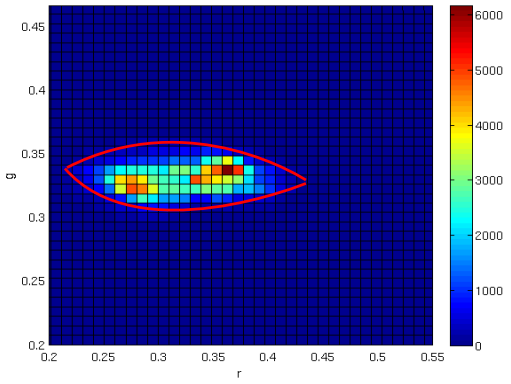


Abbildung 22: Prinzip des *Skin Locus*. Anhand eines Histogrammes der Farbwerte einer Trainingsstichprobe (hier im nRG-Farbraum) werden kompakte Häufungsgebiete der Hautfarbe durch parametrische Hüllkurven (rote Linien) approximiert.

Die Bewegungs-Pseudowahrscheinlichkeiten $P(\mathcal{M}|b(\mathbf{x}))$ entsprechen den Werten der Vordergrundmaske $\mathbf{M}(\mathbf{x})$. In Abbildung 23 ist der Ablauf grafisch dargestellt.

Die Positionen von Handhypothesen ergeben sich nun als Maxima der resultierenden Vordergrundkarte $\mathbf{V}(\mathbf{x}) = (P(\mathcal{S}, \mathcal{M}|b(\mathbf{x})))$. Hierfür wird innerhalb des Suchbereiches, der durch die umschließenden Rechtecke der Vordergrundregionen $\Omega_R$ gegeben ist, eine erschöpfende Suche durch ein gleitendes Fenster mit 50% Überlappung durchgeführt. Die Größe des gleitenden Fensters $(w, h)$ kann hierbei relativ zur Größe der Kopf-Schulter-Hypothese gewählt werden. Die untersuchten Regionen durchlaufen einen zweistufigen Auswahlprozess. Sei $\mathbf{r}_i = (x_i, y_i, w, h)$ das $i$-te Fenster mit dem Mittelpunkt $(x_i, y_i)$. Als vorläufige Handkandidaten $\hat{\Omega}_H$ werden zunächst alle Fenster betrachtet, deren mittlerer Vordergrundwert $\bar{V}_i$ über einem Schwellwert $\delta_V$ liegt:

$$\hat{\Omega}_H = \{\mathbf{r}_i | \bar{V}_i > \delta_V\}, \quad \bar{V}_i = \frac{1}{wh} \sum_{\mathbf{x} \in \mathbf{r}_i} \mathbf{V}(\mathbf{x}). \tag{6.26}$$

Die Berechnung der $\bar{V}_i$ ist effizient mittels eines Integralbildes möglich. Im zweiten Schritt wird die Kandidatenliste gemäß eines adaptiven Schwellwertes gefiltert:

$$\tilde{\Omega}_H = \{\mathbf{r}_i \in \hat{\Omega}_H | \bar{V}_i > \gamma_H \cdot \max_i(\bar{V}_i)\}. \tag{6.27}$$

Der Faktor $\gamma_H$ sorgt dafür, dass unabhängig vom absoluten maximalen Vordergrundwert immer eine gewisse Anzahl „guter“ Hypothesen generiert wird. Allerdings würde

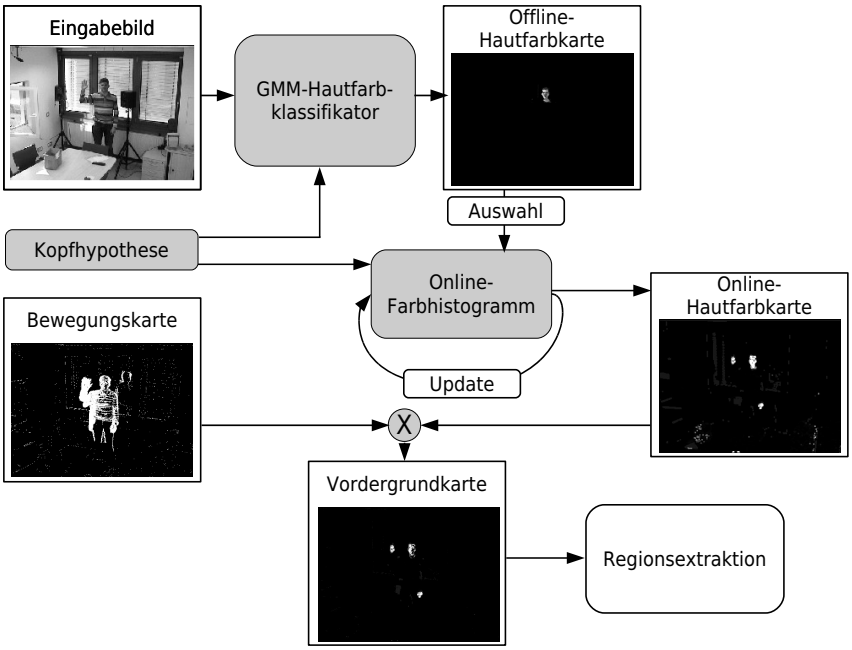


Abbildung 23: Schematische Darstellung der Handdetektion mittels Hautfarbe und Bewegung.

so immer mindestens eine Hypothese existieren, auch wenn diese sehr unplausibel wäre. Der erste Schwellwert $\delta_V$ stellt sicher, dass dies nicht geschieht, indem Suchfenster mit sehr niedrigem Vordergrundwert zuvor ausgeschlossen werden. Die resultierende Kandidatenliste wird – wiederum per *MeanShift* – geclustert, um die endgültige Liste der Handkandidaten $\Omega_H = \{\mathbf{r}_{H,i}, i = 1...n\}$ zu erhalten. Ein beispielhaftes Ergebnis ist in Abbildung 24 zu sehen. Durch den Beitrag der Bewegungskarte werden die hautfarbenen Bereiche auf dem Tisch und an der Wand entfernt, während anhand der Hautfarbkarte der bewegte Arm und die Randregionen des Körpers verworfen werden können. Die Einbeziehung der Bewegungsinformation sorgt jedoch auch dafür, dass die rechte, nicht bewegte Hand nicht detektiert wird, obwohl sie in der Hautfarbkarte gut repräsentiert ist.

Im Gegensatz zu vielen anderen Ansätzen zur Handtrajektorien-basierten Gestenerkennung (z.B. [12, 77]) wird keine perfekte Handdetektion angenommen. Die Liste $\Omega_H$ wird üblicherweise eine gewisse Anzahl falscher Hypothesen enthalten. Nicht plausible Hypothesen werden während der folgenden 3D-Kombination identifiziert und verworfen.

Abbildung 24: Beispielergebnis für die Handdetektion mit Hautfarbe und Bewegung. Obere Reihe von links nach rechts: Bewegungskarte, Hautfarbkarte des statischen Modelles, online Hautfarbkarte. Untere Reihe: Kombinierte Vordergrundkarte, Handkandidaten (rot) vor und nach der Clusterung. Gelbes Rechteck: Bewegungsregion. Grünes Rechteck: aktuelle Kopfhypothese.

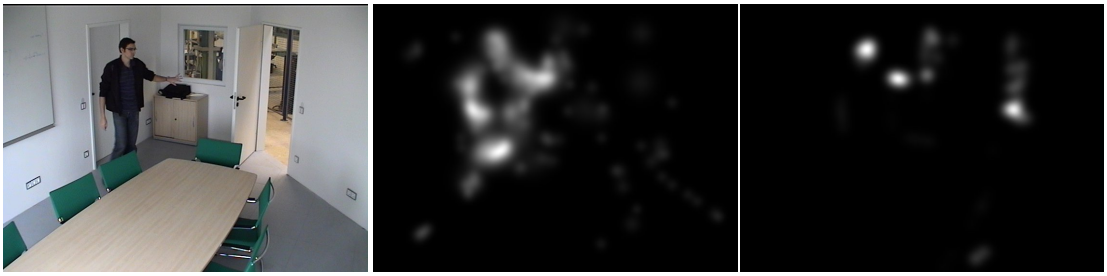

Abbildung 25: Beispiel für die verwendeten Salienzkarten bei der Handdetektion mit SIFT *Keypoints* und Hautfarbe. Von links nach rechts: Eingabebild, SIFT-Vordergrundwahrscheinlichkeit, Hautfarb-Vordergrundwahrscheinlichkeit.

### 6.5.3 *Kombination von SIFT und Hautfarbe*

Die größte Schwäche der in Kapitel 6.5.1 vorgestellten SIFT-basierten Handdetektion besteht in der üblicherweise großen Zahl von Fehldetektionen (vgl. Ergebnisse in

[162]). Deshalb ist es naheliegend, die Ergebnisse der *Keypoint*-Klassifikation mit denen einer Hautfarbdetektion zu kombinieren. Auf diese Weise können *Keypoints*, die nicht auf hautfarbenen Bereichen liegen, verworfen werden. Dieser Ansatz wurde in [149] und [161] ausführlich beschrieben und es wurden verschiedene Möglichkeiten für die Kombination der beiden Modalitäten untersucht:

- **Filterung der *Keypoints* mit der Hautfarbwahrscheinlichkeit:** Es werden nur diejenigen *Keypoints* betrachtet, die in Bildregionen mit hoher Hautfarbwahrscheinlichkeit liegen.
- **Kombinierter Deskriptor:** Innerhalb der durch die Skalierung des *Keypoints* vorgegebenen Region wird ein Histogramm der Hautfarbwerte berechnet. Dieses wird normalisiert und mit dem SIFT-Deskriptor konkateniert. Die Trainingsbeispiele müssen hierbei auf die gleiche Weise generiert werden. Die Klassifikation erfolgt dann wie oben beschrieben und betrachtet neben der Deskriptordistanz auch die Distanz der jeweiligen Hautfarbhistogramme.
- **Kombination von Salienzkarten:** Jeder Hautfarbpixel und jeder SIFT *Keypoint*-Kandidat wird als Mittelpunkt eines Gauss'schen *Blobs* angesehen, dessen Standardabweichung für die Hautfarbpixel fest vorgegeben ist und für die *Keypoints* proportional zu deren Skalierung gewählt wird. Die Überlagerung der *Blobs* ergibt für jede Modalität eine sog. Salienzkarte (vgl. Abbildung 9), deren Werte sich als Vordergrundwahrscheinlichkeiten interpretieren lassen. Diese Karten können in geeigneter Weise (z.B. Multiplikation oder gewichtete Summe) kombiniert und ihre Werte entweder zur Filterung der *Keypoints* oder direkt zur Extraktion von Handkandidaten (wie in Kapitel 6.5.2) verwendet werden.

Als Hautfarbmodell kommt – mit Ausnahme des kombinierten Deskriptors – ebenfalls ein statisch trainiertes GMM zum Einsatz. Die weitere Verarbeitung der Vordergrund-Kandidaten erfolgt jeweils mit dem in Kapitel 6.5.1 beschriebenen Filterprozess.

## 6.6 3D KOMBINATION

Im Anschluß an die Verarbeitung der einzelnen Bildströme müssen die Ergebnisse verschiedener Kameras zusammengeführt werden. Die Kombination der 2D-Punkthypothesen liefert eine Menge von 3D-Punkthypothesen, welche die aktuelle Konfiguration unabhängig vom Blickwinkel beschreiben. Aufgrund der unsynchronisierten, potentiell aktiven Kameras und der Unsicherheiten bei der Lokalisierung der 2D-Hypothesen ergeben sich hierbei einige Besonderheiten, die zu beachten sind. Im Folgenden werden die einzelnen Kombinationsschritte vorgestellt.

### 6.6.1 *Kamerakalibrierung*

#### *Grundkalibrierung*

Damit eine 3D-Kombination möglich ist, müssen zunächst die intrinsischen und extrinsischen Parameter der beteiligten Kameras bekannt sein, d.h. es muss eine Kalibrierung durchgeführt werden. Die Genauigkeit dieser bestimmt entscheidend die Genauigkeit der rekonstruierten 3D Punkte. Die Grundkalibrierung kann dabei einmalig offline erfolgen, solange die verwendeten Kameras und ihre Raumpositionen sich nicht ändern. Zwar existieren auch leistungsfähige Verfahren zur automatischen Selbstkalibration (z.B. [63, 74]), in vorliegender Arbeit wurde aber aufgrund der zu erwartenden höheren Genauigkeit eine MATLAB-Implementierung eines manuellen Kalibrierverfahren [15] eingesetzt. Es orientiert sich weitgehend an der Methode von Zhang [214] und arbeitet mit einem bekannten planaren Kalibriermuster (Schachbrett), welches der Kamera mehrfach in unterschiedlicher relativer Lage präsentiert wird. Die grundlegende Idee des Verfahrens wird im Folgenden kurz erläutert, für formale Details sei auf [214] verwiesen.

In Kapitel 2.9 wurde gezeigt, wie sich die Projektion eines Szenepunktes auf einen Bildpunkt mit Hilfe der extrinsischen und intrinsischen Kalibrationsmatrizen beschreiben lässt. Die Elemente dieser Matrizen sind die freien Parameter, die bestimmt werden müssen. Sind nun für ein gegebenes Bild die Koordinaten korrespondierender Punkte im Welt- und Kamerakoordinatensystem bekannt und liegen diese auf einer Ebene, so lässt sich eine *Homografie* berechnen. Das Kalibriermuster bildet eine solche Ebene, lässt eine einfache Ermittlung von Punktkorrespondenzen (die inneren Ecken des Schachbretts) zu und definiert gleichzeitig ein Weltkoordinatensystem mit bekannten Abmessungen. Also kann für jedes aufgenommene Bild des Musters eine Homografie berechnet werden. Diese liefert genau zwei Randbedingungen für die Schätzung der intrinsischen Parameter. Deshalb sind mehrere Aufnahmen des Kalibriermusters mit unterschiedlichen Homografien, d.h. unterschiedlicher relativer Lage des Musters, erforderlich. Die Kombination der Randbedingungen vieler verschiedener Homografien führt auf ein überbestimmtes Gleichungssystem, aus dem die intrinsischen Kameraparameter per kleinster-Quadrate-Regression bestimmt werden können. Aus der intrinsischen Matrix können die extrinsischen Parameter berechnet werden. Die Lösung wird anschließend durch iterative Minimierung des Rückprojektionsfehlers der Punkte des Kalibrationsmusters mit den ermittelten Projektionsmatrizen verfeinert. In diesen Optimierungsschritt kann eine gleichzeitige *Maximum Likelihood* Schätzung von Linsenverzerrungsparametern integriert werden.

Die so berechneten extrinsischen Kalibrationsmatrizen sind allerdings nur bildweise relativ zum jeweiligen Weltkoordinatensystem, d.h. zur Lage des Kalibrationsmusters, definiert. Um die Transformation für das tasächliche Weltkoordinatensystem zu berechnen, müsste die genaue Lage und Rotation der Kalibrationsebene in Weltkoordinaten jeweils bekannt sein. Es ist daher einfacher, die extrinsischen Parameter (Rotation und Translation) jeder Kamera einmalig durch manuelles Ausmessen zu bestimmen.

*Nachführung der Kalibrierung für aktive Kameras*

Bei den eingesetzten Schwenk-Neige-Zoom-Kameras können sich die Kameraparameter zur Laufzeit verändern. Schwenken und Neigen verändert die extrinsischen Parameter, während Zoomen die fokale Länge der Kamera verändert, also Einfluß auf die intrinsischen Parameter hat. Obige Grundkalibrierung beschreibt somit die Kameraprojektion in einem Nullzustand. Der tatsächliche Zustand zur Laufzeit kann von diesem abweichen und die Parameter müssen entsprechend nachgeführt werden.

Im Folgenden wird angenommen, dass die Schwenk- und Neigeachsen der Kamera sich in einer Ebene befinden. Diese Annahme ist für die verwendeten Kameras erfüllt. Weiterhin sei der Einfachheit halber der Nullzustand der Kamera zunächst so gewählt, dass ihr projektives Zentrum sich im Ursprung des Weltkoordinatensystems befindet und die Schwenk- und Neigeachsen jeweils mit einer Koordinatenachse des Weltkoordinatensystems übereinstimmen. Schwenken entspricht einer Rotation $\mathbf{R}_P$ mit dem Schwenkwinkel $\phi_P$ um die z-Achse, neigen einer weiteren Rotation $\mathbf{R}_T$ mit dem Neigewinkel $\phi_T$ um die x-Achse. Demzufolge überführt folgende Transformation einen Punkt $\mathbf{x}$ (gegeben in homogenen Koordinaten) in das neue (gedrehte) Koordinatensystem:

$$\begin{aligned}\hat{\mathbf{x}} &= \begin{pmatrix} \mathbf{R}_P^{-1}\mathbf{R}_T^{-1} & \mathbf{o}_3 \\ \mathbf{o}_3^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_3 & \mathbf{x}_Z \\ \mathbf{o}_3^T & 1 \end{pmatrix} \mathbf{x} \\ &= \mathbf{T}_{PT}\mathbf{x}\end{aligned} \tag{6.28}$$

Hierbei bezeichnet $\mathbf{x}_Z$ die Koordinaten des projektiven Zentrums relativ zum Schnittpunkt der Drehachsen und $\mathbf{o}$ ist der Nullvektor. Für die hier verwendeten Kameras kann mit guter Näherung angenommen werden, dass $\mathbf{x}_Z = \mathbf{o}$. Die neue extrinsische Kalibrationsmatrix ergibt sich demnach zu

$$\hat{\mathbf{K}}_e = \mathbf{T}_{PT}\mathbf{K}_e. \tag{6.29}$$

Die Schwenk- und Neigewinkel einer Kamerabewegung müssen für die Berechnung bekannt sein. Die verwendeten Kameras bieten die Möglichkeit, die aktuelle Position

der Drehachsen in Form von Stellmotorschritten $k_P, k_T$ auszulesen. Damit lassen sich die Winkel einfach durch lineare Interpolation bestimmen:

$$\phi_P = \phi_P^{min} + \frac{k_P - k_P^{min}}{k_P^{max} - k_P^{min}}(\phi_P^{max} - \phi_P^{min}). \tag{6.30}$$

Die Bestimmung von $\phi_T$ erfolgt analog. Dabei sind die minimalen und maximalen Winkel $\phi_P^{min}, \phi_P^{max}$ und Stellmotorwerte $k_P^{min}, k_P^{max}$ aus den Spezifikationen der Kamera bekannt.

Die Annahme eines linearen Zusammenhanges zwischen den Stellmotorschritten und dem entsprechenden Winkel hat sich in der Praxis als hinreichend gut erwiesen. Leider gilt dies nicht für die Abhängigkeit der fokalen Länge vom ausgelesenen Zoomwert. Diese hängen nicht linear voneinander ab, eine entsprechende Kennlinie muss demzufolge durch Interpolation zwischen mehreren Kalibrierpunkten bestimmt werden. Dies kann prinzipiell durch mehrmaliges Anwenden der obigen Grundkalibrierung bei unterschiedlichen Zoomstufen geschehen. Allerdings ist dieses Vorgehen sehr aufwändig und aufgrund des sehr großen Zoombereiches für hohe Zoomstufen nur noch schwer durchführbar. Deshalb wurde eine einfachere automatische Kalibriermethode für die fokale Länge realisiert.

Hierfür wird die Kamera so ausgerichtet, dass sie eine Ebene aufnimmt, die im Abstand $x$ ungefähr parallel zur Bildebene ist. Auf dieser Ebene befinden sich mehrere farbige Markierungen, die einfach zu detektieren sind[11]. Sei $d_M$ der Abstand zweier Marker und $\hat{d}_M(k_Z)$ der Abstand ihrer korrespondierenden Abbilder in der Bildebene für die Zoomstufe $k_Z$. Aufgrund des Strahlensatzes gilt

$$f(k_Z) = \frac{x\hat{d}_M(k_Z)}{d_M}. \tag{6.31}$$

Mit der bekannten minimalen Brennweite der Kamera $f_0$ ergibt sich die relative Änderung zu

$$\frac{f(k_Z)}{f_0} = \frac{x\hat{d}_M(k_Z)d_M}{x\hat{d}_M(0)d_M} = \frac{\hat{d}_M(k_Z)}{\hat{d}_M(0)}. \tag{6.32}$$

Die aktuelle Brennweite $f(k_Z)$ lässt sich also sehr einfach anhand der im Bild gemessenen Markerabstände bestimmen. Um gegenüber Messfehlern robust zu sein, werden mehrere Marker in einem Muster mit jeweils gleichen horizontalen und vertikalen Markerabständen platziert. $\hat{d}_M(k_Z)$ wird dann als Mittelwert der Mittelpunktabstände

11 In der Praxis wurden rote Magnetmarker an einer weißen Tafel benutzt

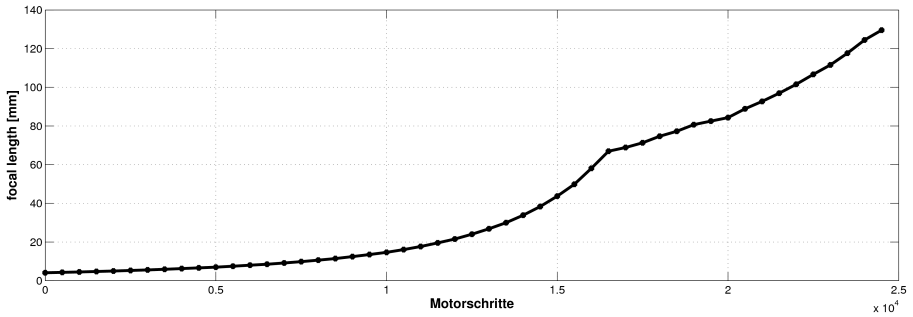


Abbildung 26: Automatisch ermittelte Messkurve für die Abhängigkeit der fokalen Länge einer Kamera von der eingestellten Zoomstufe (gemittelt über die entsprechenden Kurven für horizontale und vertikale Markerabstände). Die Knickstelle stellt den Übergang zum Digitalzoom dar.

aller benachbarten Marker über mehrere Messreihen berechnet. Zusätzlich wird vor der Berechnung des mittleren Abstandes eine Ausreißerelimination vorgenommen, indem alle Messungen verworfen werden, deren Differenz zum Median aller Messwerte für die jeweilige Zoomstufe größer ist als die Standardabweichung dieser Messwerte.

Obige Prozedur ist nicht nur unabhängig von $x$ und $d_M$, sondern auch von der Anzahl extrahierter Marker. Letzteres ist insbesondere bei starker Vergrößerung wichtig, weil nicht immer alle Marker im Kamerabild sichtbar sind. Abbildung 26 zeigt die so ermittelte Messkurve für eine der verwendeten Kameras. Diese Messkurve kann zur Nachführung der intrinsischen Kalibration verwendet werden[12].

### 6.6.2 *Verallgemeinerter Strahlenschnitt*

Für die 3D-Rekonstruktion eines Szenepunktes aus mindestens zwei Kameraansichten werden neben den intrinsischen und extrinsischen Kameraparametern *Punktkorrespondenzen* zwischen den Kameraansichten benötigt. D.h. um einen Szenepunkt rekonstruieren zu können, muss zunächst bekannt sein, auf welche Punkte er in den verschiedenen Kamerabildern abgebildet wird. Das zuverlässige Finden solcher Korrespondenzen stellt das Hauptproblem sämtlicher 3D-Rekonstruktionsalgorithmen dar. Im vorliegenden Fall ist dies aus zwei Gründen besonders problematisch:

12 Vernachlässigt man den Einfluß der Fokuseinstellung, hängt die intrinsische Kalibrierung nur von der fokalen Länge ab, da alle anderen intrinsischen Parameter für eine gegebene Kamera konstant sind.

- Es wird von einem uneingeschränkten Multikamerasystem ausgegangen. Daher zeichnen die Kameras potentiell sehr unterschiedliche Ansichten der Szene auf und es kann nicht angenommen werden, dass der gleiche Szenepunkt sich in verschiedenen Bildern mit einfachen bildbasierten Merkmalen finden lässt. Insbesondere können Szenepunkte fehlen oder verdeckt sein.

- Die Kameras sind unsynchronisiert. D.h. ihre Bildaufnahmezeiten können unterschiedlich sein, was bei bewegten Objekten – wie z.B. Personen – zu Problemen führt: Selbst wenn Punktkorrespondenzen auf einem bewegten Objekt gefunden werden, repräsentieren sie nicht mehr den selben Szenepunkt, weil die Szene sich verändert hat. Für eine Gestenerkennung werden aber gerade solche Punkte benötigt (die Rekonstruktion des statischen Hintergrundes ist für die Erkennungsaufgabe irrelevant).

Das Finden von dichten Punktkorrespondenzen und ihre Kombination zu einem (z.B.) volumetrischen 3D-Modell erscheint unter diesen Umständen nicht praktikabel. Aus diesem Grund werden nur einige wenige Punktkorrespondenzen betrachtet und eine Methode zur 3D-Rekonstruktion verwendet, die mit Fehlern in der Lokalisierung dieser Punkte und dem ggf. auftretenden zeitlichen Versatz umgehen kann.

Potentiell korrespondierende Punkte sind aus den vorherigen Verarbeitungsschritten bereits verfügbar: Die Kopf- und Handhypothesen, die in jedem Kamerabild lokalisiert wurden. Eine effiziente Methode zur 3D-Kombination solcher Punkthypothesen stellt der Strahlenschnitt dar. Gegeben seien $n$ Kameras $\mathcal{K}_i, i = 1...n$, mit ihren Projektionsmatrizen $\mathbf{K}_i$. In jedem Kamerabild seien $m$ Korrespondenzhypothesen $\mathbf{p}_{i,j}, j = 1...m$, bekannt (Punkte mit gleichen Indizes $j$ seien korrespondierend). Unter Verwendung der Projektionsmatrizen lässt sich für jeden Punkt ein Strahl $\hat{\mathbf{p}}_{i,j}$ berechnen:

$$\hat{\mathbf{p}}_{i,j} = \mathbf{k}_i + \lambda_i \mathbf{d}_{i,j}. \tag{6.33}$$

Hierbei ist $\mathbf{k}_i$ das projektive Zentrum von Kamera $\mathcal{K}_i$, $\lambda_i$ ein Skalierungsfaktor und $\mathbf{d}_{i,j}$ der Richtungsvektor des Strahls. Dieser kann mit Hilfe der Inversen der Kameraprojektionsmatrix berechnet werden:

$$\mathbf{d}_{i,j} = \tilde{\mathbf{K}}_i^{-1} \mathbf{p}_{i,j}. \tag{6.34}$$

Hierbei bezeichnet $\tilde{\mathbf{K}}_i$ die linke 3x3 Submatrix von $\mathbf{K}_i$.

Die Strahlen aller korrespondierenden Punkte schneiden sich theoretisch in einem Punkt der Szene. Dies ist der entsprechende rekonstruierte Szenepunkt. In der Praxis ist dies für gewöhnlich nicht der Fall. Aufgrund der approximativen Natur der Kalibrierung, der Diskretisierung der Bildebene sowie Fehlern bei der Lokalisierung

korrespondierender Punkte in den unterschiedlichen Kamerabildern ist davon auszugehen, dass die Strahlen windschief sind, d.h. dass kein bzw. kein einheitlicher Schnittpunkt existiert. Dies gilt insbesondere natürlich für nicht synchronisierte Kameras. Die Aufgabe besteht also darin, einen approximativen Schnittpunkt zu finden.

Deshalb wird nicht der Schnittpunkt zweier Strahlen berechnet, sondern derjenige Raumpunkt, an dem ihr Abstand minimal wird. Seien $\mathbf{p}_i$ und $\mathbf{p}_j$ zwei korrespondierende Punkte in den Kamerabildern $i$ und $j$, mit den entsprechenden Strahlen

$$\begin{aligned} \hat{\mathbf{p}}_i &= \mathbf{k}_i + \lambda_i \mathbf{v}_i, \\ \hat{\mathbf{p}}_j &= \mathbf{k}_j + \lambda_j \mathbf{v}_j. \end{aligned} \tag{6.35}$$

Die beiden Richtungsvektoren definieren eine Ebene $\mathcal{E} : \mathbf{n}^T\mathbf{x} - \gamma = 0$ mit $\mathbf{n} = \mathbf{v}_i \times \mathbf{v}_j$ und $\gamma = \mathbf{n}^T\mathbf{k}_i$, die $\hat{\mathbf{p}}_i$ enthält und parallel zu $\hat{\mathbf{p}}_j$ ist. Eine solche Ebene existiert immer, solange die Richtungsvektoren $\mathbf{v}_i$ und $\mathbf{v}_j$ nicht parallel sind. Der Abstand $d_{i,j}$ des Strahls $\hat{\mathbf{p}}_j$ von $\mathcal{E}$ ergibt sich durch Einsetzen in die Ebenengleichung zu

$$d_{i,j} = \mathbf{n}^T\mathbf{k}_j - \gamma. \tag{6.36}$$

Der gesuchte angenäherte Strahlenschnittpunkt liegt auf der kürzesten Verbindungsstrecke $\mathcal{S}$ zwischen $\hat{\mathbf{p}}_i$ und $\hat{\mathbf{p}}_j$. Diese hat die gleiche Richtung wie $\mathbf{n}$ und ihre Fußpunkte auf den Strahlen lassen sich wie folgt berechnen: Verschiebt man $\mathbf{k}_j$ entlang $\mathbf{n}$ um $-d_{i,j}$, so liegt der resultierende Strahl $\hat{\mathbf{q}}_j = \mathbf{k}_j - d_{i,j}\mathbf{n} + \lambda_j\mathbf{v}_j$ in der Ebene $\mathcal{E}$ und schneidet somit $\hat{\mathbf{p}}_i$. Sei $\mathbf{x}_s$ der Schnittpunkt der beiden Strahlen in $\mathcal{E}$. $\mathbf{x}_s$ ist der Fußpunkt von $\mathcal{S}$ auf $\hat{\mathbf{p}}_i$ und der Fußpunkt auf $\hat{\mathbf{p}}_j$ ergibt sich entsprechend durch Rückverschiebung um $d_j$ entlang $\mathbf{n}$. Der gesuchte approximierte Schnittpunkt $\tilde{\mathbf{x}}_s$ kann nun durch lineare Interpolation zwischen den Fußpunkten gefunden werden:

$$\tilde{\mathbf{x}}_s = \left(\frac{\alpha_i}{\alpha_i + \alpha_j}\right)\mathbf{x}_s + \left(\frac{\alpha_j}{\alpha_i + \alpha_j}\right)(\mathbf{x}_s + d_{j,i}\mathbf{n}) \tag{6.37}$$

Hierbei sind $\alpha_i$ und $\alpha_j$ Gewichtungsfaktoren, die beispielsweise die Zuverlässigkeit der Punktpositionen in den Kamerabildern repräsentieren können. Je höher die Zuverlässigkeit $\alpha_i$ von $\mathbf{p}_i$ im Vergleich zu $\alpha_j$ ist, desto näher liegt der interpolierte Schnittpunkt an $\hat{\mathbf{p}}_i$. Auf diese Weise lassen sich etwa Güte- oder Konfidenzwerte der einzelnen Punkthypothesen in die Schnittpunktschätzung integrieren. Wenn $\alpha_i$ und $\alpha_j$ den gleichen Wert haben, ist der resultierende Punkt der Mittelpunkt auf $\mathcal{S}$ zwischen den Fußpunkten von $\mathcal{S}$ auf $\hat{\mathbf{p}}_i$ und $\hat{\mathbf{p}}_j$. Ergebnis der 3D-Projektion ist je eine Liste mit Kopfhypothesen $\mathbf{h}_K = \{\mathbf{h}_{K,k}, k = 1...p\}$, $\mathbf{h}_{K,k} = (x_{K,k}, y_{K,k}, z_{K,k})$ und Handhypothesen $\mathbf{h}_H = \{\mathbf{h}_{H,l}, l = 1...q\}$, $\mathbf{h}_{H,l} = (x_{H,l}, y_{H,l}, z_{H,l})$.

### 6.6.3 *Kombination von Hypothesen mehrerer Kameras*

In einem Multikamerasystem mit mehr als zwei Kameras ergeben sich ggf. mehrere paarweise Kombinationsmöglichkeiten für den selben 3D Punkt. Aufgrund der erwähnten Unsicherheiten wird jede paarweise Kombination potentiell einen anderen approximierten Strahlenschnittpunkt liefern. In diesem Fall müssen die verschiedenen Schnittpunkt-Hypothesen in geeigneter Weise kombiniert werden. Die einfachste (aber auch fehleranfälligste) Möglichkeit stellt die Mittelwertbildung über die Koordinaten benachbarter Schnittpunkte dar. Diese kann um eine Ausreißerelimination erweitert oder durch ein Clusterverfahren ersetzt werden. Weiterhin ist es denkbar, bereits vor der Kombination aus den möglichen Kameraansichten die zwei am besten geeigneten anhand globaler Kriterien auszuwählen. Ansätze zu einer solchen Ansichtsauswahl finden sich z.B. in der Arbeit von Schauerte [170].

Prinzipiell kann jedoch jede Hypothese informationstragend sein. Anhand der bis dato verfügbaren Informationen lässt sich zudem kaum eine zuverlässige Rückweisung fehlerhafter Hypothesen vornehmen. Deshalb werden zunächst alle 3D-Punkthypothesen als gleichwertig betrachtet[13] und gemäß verschiedener Kriterien mit einem Gütewert versehen. Eine Rückweisung nicht plausibler Hypothesen muß im weiteren Verlauf während der Trajektorienaggregation erfolgen.

### 6.6.4 *Bewertung von 3D-Punkthypothesen*

Das og. Vorgehen zur 3D-Kombination von Punkthypothesen wird im Allgemeinen keine perfekten Ergebnisse liefern. Durch die notwendige Approximation des Schnittpunktes können 3D-Punkhypothesen entstehen, die keiner realen 3D-Merkmalspunktposition entsprechen. Zudem tritt bei allen 3D-Rekonstruktionsverfahren ein grundlegendes Problem auf: Die Position einer 2D-Hypothese im 3D Raum lässt sich lediglich auf die zugehörige projektive Linie einschränken, ist also nicht eindeutig. Demzufolge können für jede projektive Linie mehrere Schnittpunkte auftreten (sog. Schattenregionen oder -hypothesen, vgl. Abbildung 27) und es gibt zunächst keine Möglichkeit, festzustellen, welcher davon der richtige ist.

13 Hypothesen, die räumlich sehr nahe beieinanderliegen, können jedoch trotzdem gefahrlos durch ein geeignetes Verfahren – z.B. wiederum *Mean Shift* Clusterung in Verbindung mit einem Kernel geringer Bandbreite – zusammengefasst werden, weil angenommen werden kann, dass aufgrund der inhärenten Ungenauigkeit der Kombination rekonstruierte Punkte mit einem Abstand von wenigen cm dem gleichen Szenepunkt entsprechen. Dies reduziert den Suchraum für die weitere Verarbeitung und hat nur geringen Einfluß auf die Qualität des Ergebnisses.

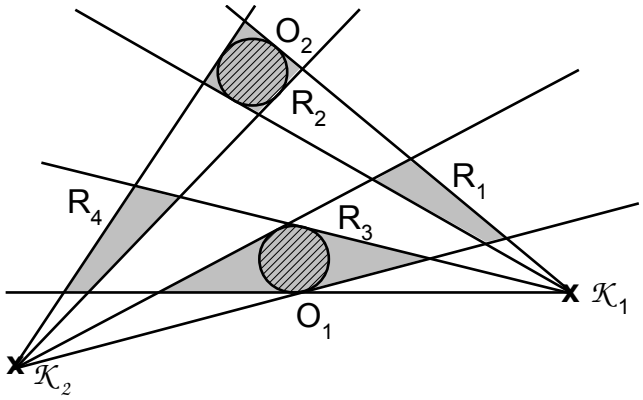


Abbildung 27: Schattenregionen: Die Objekte $O_1, O_2$ werden von beiden Kameras $\mathcal{K}_1, \mathcal{K}_2$ beobachtet. Es existieren vier mögliche Rekonstruktionen $R_1, R_2, R_3, R_4$. Im Falle von $O_1$ würde eine „harte" Verdeckungsmodellierung für $\mathcal{K}_1$ ($R_3$ verdeckt $R_4$) die richtige Region $R_4$ verwerfen, für $O_2$ jedoch die falsche.

Im Folgenden werden einige Möglichkeiten vorgestellt, 3D-Punkthypothesen anhand von Modellannahmen und Informationen aus vorhergehenden Arbeitsschritten zu bewerten, so dass eine Auswahl von 3D-Hypothesen gemäß definierter Kriterien erfolgen kann.

*Zeitliche Beschränkungen*

Ein offensichtliches Kriterium bei der Auswahl von Punktpaaren ist deren zeitliche Korrelation. Es wurde bereits mehrfach darauf hingewiesen, dass von unsynchronisierten Kameras ausgegangen wird. Daher sind die Hypothesen keinem festen Zeitpunkt $t$ zugeordnet, sondern es muss ein Zeitfenster $t \pm \Delta t$ betrachtet werden. In einem ersten Schritt können also alle Punkthypothesen aus Kamerabildern verworfen werden, die keine zeitliche Korrelation mit anderen Kamerahypothesen aufweisen. Ist die mittlere Bildwiederholrate $\bar{f}_B$ des Mustererkennungssystems bekannt[14], so beträgt der maximal mögliche zeitliche Versatz $\Delta t_{max}$ zwischen zwei Kamerabildern (ungefähr) $\frac{2}{\bar{f}_B}$. Dieser Wert – oder ein kleines Vielfaches davon – kann als Rückweisungsschwellwert verwendet werden.

Darüberhinaus gilt bei bewegten Objekten in der Szene, dass der Fehler, der bei der Kombination zweier Punkthypothesen auftritt, umso größer ist, je größer die

14 Diese kann beispielsweise auf einfache Weise während der Laufzeit ermittelt werden.

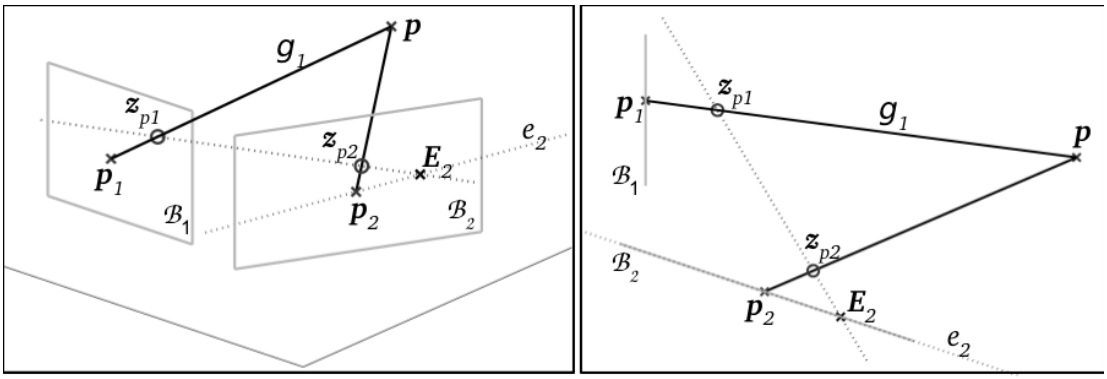


Abbildung 28: Beispiel zur Verdeutlichung der Beziehungen der Epipolargeometrie. Links: 3D-Ansicht. Rechts: Draufsicht.

Zeitdifferenz $\Delta t_{k,l}$ zwischen den Aufnahmezeitpunkten der Kameras $\mathcal{K}_k$ und $\mathcal{K}_l$ ist. Daher kann die Güte einer Kombination wie folgt probabilistisch modelliert werden:

$$P(\mathbf{h}_{i,j}^{k,l}|\Delta t_{k,l}) = \mathcal{N}(\Delta t_{k,l}, \sigma_t). \tag{6.38}$$

Hierbei bezeichnet $\mathbf{h}_{i,j}^{k,l}$ die Kombination der i-ten Hypothese von $\mathcal{K}_k$ mit der j-ten Hypothese von $\mathcal{K}_l$. Die Standardabweichung $\sigma_t$ repräsentiert Vorwissen über akzeptable Zeitunterschiede und kann z.B. so gewählt werden, dass $\sigma_t \propto \Delta t_{max}$.

*Geometrische Beziehungen*

Die 3D-Rekonstruktion erfolgt aufgrund paarweiser Kombination von 2D-Hypothesen. Im einfachsten Fall müssen dazu alle möglichen Paare von Punkten unterschiedlicher Kameras im Intervall $t \pm \Delta t$ betrachtet werden. Gewöhnlich ist aber nur ein Teil dieser Kombinationen geometrisch sinnvoll. Es ist daher naheliegend, bereits bei der Wahl der Hypothesenpaare eine Vorauswahl zu treffen. Eine Möglichkeit hierfür bietet die Epipolargeometrie (vgl. z.B. [51]).

Gegeben seien zwei Kameras $\mathcal{K}_1$ und $\mathcal{K}_2$ mit ihren Bildebenen $\mathcal{B}_1$ und $\mathcal{B}_2$ und ihren Projektionszentren $\mathbf{z}_{p1}$ und $\mathbf{z}_{p2}$. $\mathbf{p}$ sei ein dreidimensionaler Punkt der Szene und $\mathbf{p}_1$ bzw. $\mathbf{p}_2$ seien seine Projektionen auf die jeweiligen Bildebenen (Abbildung 28). Weiterhin seien die Projektionsmatrizen der Kameras $\mathbf{K}_1$ und $\mathbf{K}_2$ sowie die Bildkoordinaten von $\mathbf{p}_1$ bekannt. Die Halbgerade $g_1$ (d.h. die projektive Linie von $\mathbf{p}_1$) beginnt in $\mathbf{p}_1$ und geht durch $\mathbf{z}_{p1}$ und $\mathbf{P}$. Das Abbild von $g_1$ auf $\mathcal{B}_2$ ist ebenfalls eine Gerade $e_2$, die als Epipolarlinie von $\mathbf{p}_1$ in $\mathcal{B}_2$ bezeichnet wird. Sie verläuft durch

den Epipol $\mathbf{E}_2$. $\mathbf{E}_2$ ist der Schnittpunkt der Geraden durch die Projektionszentren der beiden Kameras mit $e_2$, also die Projektion von $\mathbf{z}_{p1}$ auf $\mathbf{B}_2$:

$$\mathbf{E}_2 = \mathbf{K}_2 \mathbf{z}_{p1} \tag{6.39}$$

Ein weiterer Punkt auf $e_2$ ist der Schnittpunkt von $g_1$ mit $\mathcal{B}_2$ im Unendlichen:

$$\tilde{\mathbf{p}_1} = \mathbf{K}_2 \tilde{\mathbf{K}}_1^{-1} \mathbf{p}_1 \tag{6.40}$$

Die Gleichung der gesuchten Epipolarlinie $e_2$ ergibt sich zu

$$\begin{aligned} \mathbf{L}_2^T \mathbf{x}_2 &= 0 \\ \mathbf{L}_2 = \mathbf{E}_2 \times \tilde{\mathbf{p}}_1 \quad &= \quad \left(\mathbf{K}_2 \mathbf{z}_{p1}\right) \times \left(\mathbf{K}_2 \tilde{\mathbf{K}}_1^{-1} \mathbf{p}_1\right) \\ &= \left(\mathbf{S}_{\mathbf{K}_2 \mathbf{z}_{p1}} \mathbf{K}_2 \tilde{\mathbf{K}}_1^{-1}\right) \mathbf{p}_1 = \mathbf{F} \mathbf{p}_1 \end{aligned} \tag{6.41}$$

mit der schiefsymmetrischen Kreuzproduktmatrix $\mathbf{S}$. $\mathbf{F}$ wird als *Fundamentalmatrix* bezeichnet und lässt sich bei bekannter Kamerakalibrierung direkt aus den Kalibrierungsmatrizen berechnen. Diese Beziehungen sind symmetrisch, d.h. sie gelten analog auch für die Bildebene $\mathcal{B}_1$.

Damit lässt sich für jeden Punkt in Bildebene $\mathcal{B}_1$ die entsprechende Epipolarlinie in $\mathcal{B}_2$ (und umgekehrt) berechnen. Alle möglichen korrespondierenden Punkte liegen im ungestörten Fall auf der Epipolarlinie bzw. im realen Fall in ihrer Nähe. Der Abstand $d_e$ einer Hypothese von der Epipolarlinie stellt also ein Maß für die Plausibilität der jeweiligen paarweisen Kombination dar und kann zur Vorauswahl verwendet werden.

In gleicher Weise kann auch der Abstand $d_{i,j}$ zwischen den Strahlen zweier Punkthypothesen aus (6.36) verwendet werden. Tatsächlich stellt die Betrachtung der Epipolargeometrie die Projektion dieser Situation auf eine der Kameraebenen dar. Die Abstände $d_e$ und $d_{i,j}$ sind für das gleiche Punktpaar jedoch i.d.R. nicht identisch, weil $d_{i,j}$ entlang der Richtung eines Normalenvektors $\mathbf{n}$ gemessen wird, der nicht notwendigerweise parallel zu einer der Bildebenen verläuft. Dennoch gilt auch hier die Aussage, dass eine 3D-Hypothese umso plausibler ist, je geringer der Abstand der beiden Strahlen ist, aus denen sie berechnet wurde.

Beide Varianten erfordern für jede Punkthypothese jeweils eine Multiplikation mit einer $3 \times 3$ Matrix sowie eine Abstandsberechnung in 2D (Epipolarlinie) bzw. 3D (Strahlenschnitt). Der Aufwand für die Plausibilitätsberechnung ist mit der Epipolargeometrie also geringfügig kleiner. Allerdings erfordert die Verwendung der Epipolargeometrie als zusätzlichen Schritt die Projektion und Kombination der resultierenden Hypothesenpaare, d.h. eine zusätzliche Berechnung des Strahlenschnitts auf einer Teilmenge der Hypothesen. Aus diesem Grund wird in dieser Arbeit der

Abstand $d_{i,j}$ des Strahlenschnitts als Auswahlkriterium verwendet. Das Kriterium wird wie folgt probabilistisch formuliert:

$$P(\mathbf{h}_{i,j}^{k,l}|d_{i,j}) = \mathcal{N}(d_{i,j}, \sigma_g). \tag{6.42}$$

Die Standardabweichung $\sigma_g$ muß abhängig vom erwarteten Kombinationsfehler gewählt werden, beispielsweise per *Maximum Likelihood* Schätzung aus einer annotierten Stichprobe.

*Körpermodellierung*

Wie in Kapitel 4 erläutert wurde, bietet eine Modellierung des menschlichen Körpers in Form von anthropologischen und kinematischen Beschränkungen eine Möglichkeit, fehlerhafte Hypothesen zu eliminieren. Steht im Idealfall ein Skelettmodell des Oberkörpers zur Verfügung, können Hypothesen, die unmögliche oder sehr unwahrscheinliche Körperkonfigurationen erfordern würden, verworfen werden. Es wurde allerdings bereits darauf hingewiesen, dass die akkurate Lokalisierung eines derartigen Modelles sehr aufwändig und fehleranfällig ist.

Anstelle eines kompletten Modelles werden deshalb an dieser Stelle zwei sehr einfach zu realisierende Kriterien eingeführt. Zum Einen lassen sich für die $z$-Koordinaten $z_H$ der Kopfhypothesen Erwartungswerte angeben, weil die Größe von Personen sich typischerweise in einem bestimmten Bereich bewegt. Dieser Bereich ist jedoch für Männer und Frauen sowie für verschiedene Ethnien unterschiedlich, so dass eine derartige Modellierung mit Vorsicht zu genießen ist. Deshalb wird an dieser Stelle auf eine statistische Modellierung verzichtet und es werden lediglich zwei Grenzwerte $z_{K,min} = 100\text{cm}, \quad z_{K,max} = 210\text{cm}$ benutzt, die sehr unwahrscheinliche Körpergrößen ausschließen[15]. Diese erlauben sowohl ungewöhnlich große Personen als auch Halbwüchsige bzw. Sitzende.

Zum Anderen besteht nach den Erkenntnissen der Anthropometrie bzw. der künstlerischen Proportionenlehre (vgl. z.B. [7], Kapitel 2) ein Zusammenhang zwischen der Körpergröße einer Person und ihrer Armlänge $l_A$: Die Spannweite der ausgestreckten Arme entspricht ungefähr der Körpergröße, die Länge eines Armes von der Schulter bis zu den Fingerspitzen somit etwas weniger als der halben Körpergröße. Weil die genaue Position der Schultern einer Person unbekannt ist, wird die Kopfhypothese als Referenzpunkt herangezogen und die maximale Armlänge entsprechend um eine Kopfgröße verlängert. Im Durchschnitt ist ein erwachsener Mensch zwischen sieben und acht Kopfgrößen groß. Somit ergibt sich der maximale Abstand zwischen Hand

15 Die Hypothesen entsprechen ungefähr dem Kopfmittelpunkt, so dass die realen Personengrößen etwas größer sind.

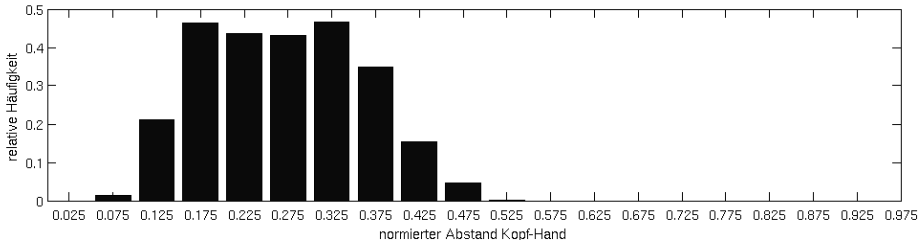


Abbildung 29: Beispielhaftes Histogramm der Verteilung des normierten Kopf-Hand-Abstandes. Die Werte der horizontalen Achse sind die Mittelpunkte der Histogrammbins, angegeben als einheitslose Bruchteile der Körpergröße. Die theoretische Annahme, dass die maximale Armlänge etwa der halben Körpergröße entspricht, wird für die hier betrachtete Stichprobe bestätigt.

und Kopf ungefähr zu $l_{A,max} \approx \frac{5}{8} z_H$. Eine Modellierung als Pseudowahrscheinlichkeit, die Unsicherheiten zulässt, ist beispielsweise mit einer Sigmoid-Funktion (vgl. Abbildung 12) möglich. Der Parameter $\alpha_2$ wird dabei zu $\alpha_2 = l_{A,max}$ gewählt, während $\alpha_1$ die Größe des „unscharfen" Bereiches angibt:

$$P\left(\mathbf{h}_{i,j}^{k} | l_A\right) = 1 - \frac{1}{1 + e^{-\alpha_1 (l_A - l_{A,max})}}. \tag{6.43}$$

Alternativ kann $p(\mathbf{h}_{i,j}^{k} | l_A)$ anhand einer Trainingsstichprobe gelernt werden. Als Beispiel zeigt Abbildung 29 ein Histogramm des normierten Kopf-Hand-Abstandes (bezogen auf $z_H$), das auf einem realistischen Armgesten-Datensatz (vgl. Kapitel 7.1.4) ermittelt wurde. Seine Form legt nahe, dass eine gute Approximation der Verteilungsdichte auch mit einer einzelnen Normalverteilung möglich wäre.

*Verdeckung*

Eine weitere Möglichkeit, fehlerhafte 3D-Hypothesen zu eliminieren, besteht in der Modellierung von Verdeckung. Geht man von einer idealen Situation aus, in der alle Punkthypothesen korrekt lokalisiert sind und keine Fehldetektionen existieren, dann lässt sich intuitiv folgende Heuristik formulieren: Liegen mehrere 3D-Hypothesen auf dem (bzw. nahe am) gleichen Strahl einer 2D-Punkthypothese, dann verdeckt diejenige mit der geringsten Distanz zur Bildebene alle anderen. Es kann also nur diese nächstgelegene Hypothese beobachtet werden und alle anderen müssen „Schattenregionen" sein. Eine Anwendung dieser Heuristik zur Elimination aller anderen Hypothesen ist gefährlich, wie anhand der in Abbildung 27 dargestellten Situation leicht nachvollziehbar ist. Sie kann jedoch in Form einer schwachen Beschränkung in die Güteberechnung einbezogen werden.

Seien $\mathbf{h}_{i,j}^k, j = 1...n$, 3D-Hypothesen, die auf dem Strahl der $i$-ten 2D-Hypothese von Kamera $\mathcal{K}_k$ liegen. Diese seien so gemäß ihres Abstandes $c_{i,j}^k$ zur Bildebene geordnet, dass $j = 1$ die nächste und $j = n$ die am weitesten entfernte Hypothese bezeichnet. Dann lässt sich mit

$$P\left(\mathbf{h}_{i,j}^k | c_{i,j}^k\right) = \alpha^{j-1}, \quad 0 < \alpha < 1 \tag{6.44}$$

auf einfache Art eine auf der Reihenfolge basierende Gewichtung realisieren, die unabhängig von den tatsächlichen Abständen $c_{i,j}^k$ ist. Sie repräsentiert die Annahme, dass Hypothesen umso unwahrscheinlicher werden, je mehr potentielle Verdeckungen existieren. Der Parameter $\alpha \in [0, 1]$ gibt die Stärke der Beschränkung an und sollte groß gewählt werden, um den Einfluß auf die Gesamtbewertung niedrig zu halten.

Eine wesentlich härtere und zuverlässigere Beschränkung ergibt sich für die Handhypothesen anhand der Kopfhypothesen: Geht man davon aus, dass sich unterhalb eines Kopfes der Körper der Person befindet, der eine gewisse Ausdehnung hat, dann können Hypothesen, die von einer bestimmten Kamera aus betrachtet hinter dem Körper liegen, nicht beobachtet worden sein. Sie können demzufolge für diese Kamera verworfen werden. Dies erfordert eine geeignete Modellierung der geometrischen Beziehungen.

Im Folgenden wird der Körper einer Person der Einfachheit halber durch einen Zylinder mit festem Radius $r_B$ und Länge $l_B$ modelliert. Geht man von den üblichen Proportionen eines Menschen aus (vgl. z.B. [7]), dann umfasst der Körper etwa $\frac{7}{8}$ der Körpergröße. $l_B$ wird demzufolge näherungsweise als $\frac{7}{8} z_K$ gewählt, mit der $z$-Koordinate der Kopfhypothese $\mathbf{h}_K = (x_K, y_K, z_K)$. Schneidet der zu einer Hypothese gehörige Strahl diesen Zylinder (Abbildung 30), d.h. ist sein kleinster Abstand $d_F$ von der Fußlinie $\mathcal{F}$ der Kopfhypothese kleiner als $r_B$ und liegt die Höhe $z_F$ des Punktes $P_F = (x_F, y_F, z_F)$ zwischen 0 und $l_B$, so können alle durch diesen Strahl erzeugten Hypothesen mit $c_{i,j}^k > c_F^k$ verworfen werden. Der kleinste Abstand $d_F$ und der Punkt $P_F$ lassen sich hierbei mit dem in Kapitel 6.6.2 vorgestellten verallgemeinerten Strahlenschnitt berechnen. In gleicher Weise können Informationen über die Szene einfließen, indem Hypothesen, die durch bekannte Szeneobjekte oder Wände verdeckt sind, ebenfalls verworfen werden.

## 6.7 TRAJEKTORIEN-BASIERTE GESTENKLASSIFIKATION

Der letzte Verarbeitungsschritt besteht in der Klassifikation von Armgesten anhand ihrer Trajektorie. Zu diesem Zweck müssen die Punkthypothesen aus unterschiedlichen Beobachtungszeitpunkten zunächst zu räumlich-zeitlichen Trajektorien kombiniert

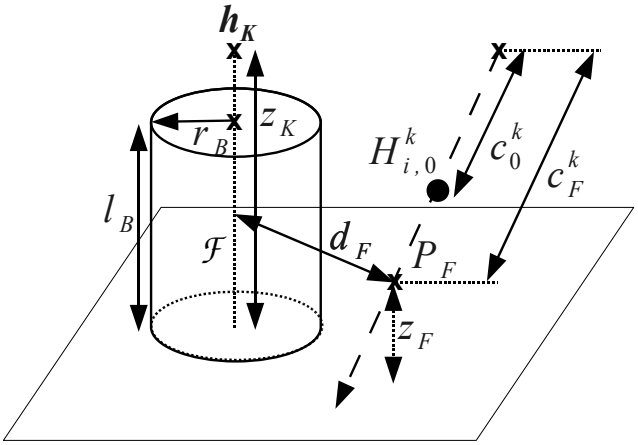


Abbildung 30: Verdeutlichung der Beziehungen bei der Berechnung von Verdeckungen durch den Körper der Person.

werden. Darauf folgen eine Normalisierung und eine Merkmalsextraktion. Die entstehenden Zeitreihen von Merkmalen werden abschließend in einem HMM-Framework klassifiziert. Die einzelnen Schritte werden im Folgenden ausführlich erläutert.

### 6.7.1 *Aggregation von Trajektorien*

Aus den vorherigen Verarbeitungsschritten resultiert zu jedem betrachteten Zeitpunkt und für jedes Kamerabild, eine Liste von Kopf- und Handhypothesen. Für die Klassifikation müssen diese einzelnen Hypothesen gemäß ihrer räumlichen und zeitlichen Beziehungen zu Trajektorien verbunden werden. Dies kann auf zwei Arten geschehen: Eine Möglichkeit besteht darin, für jede Kamera zunächst Trajektorienhypothesen in 2D zu generieren. Die Trajektorien unterschiedlicher Kameras müssen anschließend aneinander ausgerichtet und zu 3D-Trajektorien kombiniert werden.

Die zweite Möglichkeit ist, die 2D-Punkthypothesen zunächst paarweise – wie zuvor beschrieben – zu 3D-Hypothesen zu kombinieren, und die Trajektorienaggregation komplett in 3D vorzunehmen. Dieser Ansatz konnte aus Zeitgründen nicht mehr praktisch realisiert werden. Der Vollständigkeit halber werden lediglich einige Überlegungen vorgestellt, die auf ein dem *Beam Search* ähnliches Suchverfahren führen. Eine Validierung dieser Überlegungen muss diese Arbeit jedoch schuldig bleiben.

*Aggregation in 2D*

Die räumlich-zeitliche Aggregation der Kopfhypothesen ist im vorliegenden Szenario einfach: Zu jedem Zeitschritt existieren nur sehr wenige bzw. im normalen Fall eine einzige Hypothese. Weil zudem angenommen werden kann, dass die Person sich – im Vergleich zur Aufnahmegeschwindigkeit der Kamera – relativ langsam bewegt, kann bei mehreren Hypothesen die Zuordnung aufgrund der Euklidischen Distanzen der Detektionszentren getroffen werden. Zusätzlich kann als Einschränkung gelten, dass die Größe einer Person im Kamerabild – gemessen an der Größe des Kopf-Schulter-Detektionsrechteckes – sich nicht sprunghaft ändern kann. Demzufolge wird die Distanz $d_D(\mathbf{r}_D(t_1), \mathbf{r}_D(t_2))$ zweier Detektionen zu unterschiedlichen Zeitpunkten $t_1, t_2$ wie folgt definiert[16]:

$$\begin{aligned} d_D(\mathbf{r}_D(t_1), \mathbf{r}_D(t_2)) &= \|\mathbf{v}_D(t_1) - \mathbf{v}_D(t_2)\|, \\ \text{mit} \quad \mathbf{v}_D(t) &= (x_D(t), y_D(t), w_D(t))^T. \end{aligned} \tag{6.45}$$

Die Kopftrajektorie ergibt sich dann durch Zuordnung der Hypothese mit der geringsten Distanz.

Die Kombination von Handhypothesen stellt sich nicht so einfach dar: Üblicherweise werden zu jedem Kamerabild mehrere Handhypothesen existieren. Die räumliche Position einer Hand kann sich sehr schnell stark ändern und ist somit wenig aussagekräftig. Zudem steht im Gegensatz zur Kopfhypothese keine Größeninformation zur Verfügung. Es ist daher mit Mehrdeutigkeiten bei der Zuordnung zu rechnen. Die Distanz zweier Handhypothesen $d_H(\mathbf{r}_H(t_1), \mathbf{r}_H(t_2))$, wobei $\mathbf{r}_H(t_1)$ die letzte Handposition einer vorhandenen Trajektorie ist, wird wie folgt definiert:

$$\begin{aligned} d_H(\mathbf{r}_H(t_1), \mathbf{r}_H(t_2)) &= \|\mathbf{v}_H(t_1) - \mathbf{v}_H(t_2)\|, \\ \text{mit} \quad \mathbf{v}_H(t) &= (x_H(t), y_H(t), \alpha_H \cdot t)^T. \end{aligned} \tag{6.46}$$

Die Einbeziehung der Zeitdifferenz bevorzugt Zuordnungen von Hypothesen zu Trajektorien, deren letzte Punkte zeitlich benachbart sind. Weil der Beitrag des Zeitterms je nach Darstellung des Zeitstempels sowohl sehr klein (z.B. Differenz der laufenden Bildnummer) als auch sehr groß (z.B. Differenz absoluter Zeitstempel in Millisekunden) werden kann, wird ein Gewichtungsfaktor $\alpha_H$ eingeführt, mit dem sich der Einfluß des Zeitterms einstellen lässt.

Die Aggregation zu Trajektorien erfolgt dann wie in Alg. 4 dargestellt. Der Unsicherheit der Zuordnung von Hypothesen zu bestehenden Trajektorien wird Rechnung

16 Die Distanz zweier Hypothesen mit gleichem Beobachtungszeitpunkt ist per Definition unendlich.

**Algorithmus 4** Aggregation von 2D-Handhypothesen zu Trajektorien.

**Eingabe:** Menge an 2D Trajektorien $\mathcal{T}_i = \{\tau_{i,q}\}, \quad i = 1...n, q = 1...k_i$, Handhypothesen $\Omega_H = \{\mathbf{r}_{H,j}\}, \quad j = 1...m$.
**for** $i = 1...n$ **do**
  **for** $j = 1...m$ **do**
    $\mathbf{D}_{i,j} = d_H(\mathbf{r}_{H,j}, \tau_{i,k_i})$
  **end for**
**end for**
**for** $i = 1...n$ **do**
  Bestimme bis zu N beste Trajektorienfortsetzungen:
  **for** $\nu = 1...N$ **do**
    **if** $\min_j \mathbf{D}_{i,j} > \delta_T$ **then**
      Ende
    **else**
      $\tau^{\nu}_{i,k_i+1} = \mathbf{r}_{H,p} : p = \arg\min_j \mathbf{D}_{i,j}, \quad \mathbf{D}_{i,p} = \infty$
      $\mathcal{T}^{\nu}_i = \mathcal{T}_i \cup \tau^{\nu}_{i,k_i+1}$
    **end if**
  **end for**
**end for**
Generiere neue Trajektorienhypothesen für nicht zugewiesene Elemente aus $\Omega_H$.
Ausgabe: Menge an Trajektorienhypothesen $\mathcal{T}^{\nu}_i$.

getragen, indem für jede Trajektorie die N besten Hypothesen bestimmt werden, sofern deren Abstand innerhalb eines sinnvollen Schwellwertes $\delta_T$ liegt. Mit $N = 1$ ergibt sich ein einfacher gieriger Suchalgorithmus, der in jeder Iteration die jeweils global beste Zuordnung einer Punkthypothese zu einer Trajektorie auswählt. Diese Variante wurde für die experimentelle Evaluierung in [159] bzw. in Kapitel 7.6 verwendet. Sie funktioniert für die idealisierten und vorsegmentierten Daten, die dort betrachtet wurden, ausreichend gut.

Im realen kontinuierlichen Betrieb, d.h. bei Anwesenheit von Fehldetektionen und ohne gegebene Segmentierungsgrenzen, werden üblicherweise mehrere Alternativen verfolgt werden müssen, d.h. $N > 1$. Zusätzlich muss in jedem Zeitschritt eine Auswahl plausibler Trajektorienhypothesen erfolgen, weil obiger Algorithmus sonst nach kurzer Zeit zu einer kombinatorischen Explosion der Alternativen führen würde. Weil aus der 2D-Repräsentation keine zusätzlichen Informationen für eine Auswahl zu gewinnen sind, muss die Selektion sinnvoller Trajektorienhypothesen im Zuge der

3D-Kombination erfolgen. Potentielle Segmentgrenzen lassen sich z.B. auf einfache Weise durch die Detektion längerer Ruhephasen in den Trajektorienhypothesen finden.

*Ausrichtung und Interpolation von 2D-Trajektorienhypothesen*

Nach der 2D-Trajektorienaggregation existiert für jede Kamera eine Menge von Trajektorienhypothesen. Die Aufgabe besteht nun darin, zwischen diesen sinnvolle paarweise Kombinationen zu finden und gleichzeitig möglichst viele falsche Hypothesen zu verwerfen. Zu jedem Trajektorienpunkt ist der Zeitpunkt seiner Aufnahme bekannt. Der erste Schritt besteht also in der Suche nach zeitlich überlappenden Hypothesen. Paarweise Kombinationen, die keine zeitliche Überlappung aufweisen, können verworfen werden. Weiterhin kann die Anzahl möglicher Kombinationen eingeschränkt werden, falls aus vorherigen Zeitschritten bereits Zuordnungen bekannt sind. In diesem Fall werden nur die paarweisen Kombinationen der Alternativen der einander zugewiesenen Trajektorien betrachtet.

Seien $\mathcal{T}^k = \{(\tau_i^k, t_i^k), i = 1...n\}$, $\mathcal{T}^l = \{(\tau_j^l, t_j^l), j = 1...m\}$, zwei Trajektorien in den Kameras $\mathcal{K}^k, \mathcal{K}^l$ mit den Punktkoordinaten $\tau_i^k = (x_i^k, y_i^k)$ und den zugehörigen Zeitstempeln $t_i^k$. Zunächst wird durch Vergleich der Zeitstempel der Trajektorienstart- und Endpunkte die größte überlappende Teilsequenz ermittelt:

$$t_{start} = max(t_1^k, t_1^l), \quad t_{ende} = min(t_n^k, t_m^l). \tag{6.47}$$

Falls $t_{start} \geqslant t_{ende}$ ist, existiert keine zeitliche Überlappung für das gegeben Paar von Trajektorienhypothesen. Ansonsten wird jede der beiden Trajektorien an den durch die jeweils andere Trajektorie vorgegebenen Zeitpunkten linear interpoliert:

$$\forall \tau_i^k : t_i^k \in [t_{start}, t_{ende}] : \hat{\tau}_j^l = \tau_<^l + \frac{t_i^k - t_<^l}{t_>^l - t_<^l}(\tau_>^l - \tau_<^l). \tag{6.48}$$

Hierbei bezeichnen $\tau_<^l, \tau_>^l$ die jeweils zeitlich früher bzw. später zu $t_i^k$ benachbarten Trajektorienpunkte in $\mathcal{T}^l$. Analog wird für die Interpolation von $\mathcal{T}^k$ verfahren. Die beiden interpolierten Trajektorien $\hat{\mathcal{T}}^k, \hat{\mathcal{T}}^l$ sind gleich lang und ihre Elemente besitzen jeweils paarweise die gleichen Zeitstempel. Aufgrund der positionellen Unsicherheiten bei der Lokalisierung der Punkthypothesen weisen die auf diese Weise aggregierten 2D-Trajektorien üblicherweise erhebliche Variationen und Störungen auf. Deshalb werden sie abschließend mit einem Gauss'schen Fenster geglättet.

Die einander zugeordneten Punktepaare werden mit dem in Kapitel 6.6.2 beschriebenen Verfahren zu 3D Hypothesen kombiniert. Dies kann für bereits bestehende

Trajektorienzuordnungen inkrementell geschehen, indem nur neu hinzugekommene Punkte projiziert werden. Bei der Projektion kann in (6.37) für beobachtete 2D-Trajektorienpunkte ein höheres Gewicht $\alpha$ gewählt werden, als für interpolierte Punkte. Dies reflektiert die Tatsache, dass die lineare Interpolation insbesondere bei stark gekrümmten Trajektorien fehlerbehaftet ist.

Alle auf diese Weise berechneten 3D-Hypothesen werden gemäß der in Kapitel 6.6.4 eingeführten Kriterien bewertet und ggf. verworfen. Die Rekonstruktionsgüte eines Trajektorienpunktes ergibt sich unter der Annahme statistischer Unabhängigkeit zu[17] (die Indizes $k, l, i, j$ werden im Folgenden zu einem Index $q$ zusammengefasst, der über alle paarweisen Kombinationen aller Hypothesen und Kamerapaare iteriert)

$$P(\mathbf{h}_q|d_q, l_A, c_q) = P(\mathbf{h}_q|d_q) \cdot P(\mathbf{h}_q|l_A) \cdot P(\mathbf{h}_q|c_q). \tag{6.49}$$

Das Ziel ist nun, aus allen Zuordnungsalternativen für eine Trajektorie diejenige auszuwählen, welche die globale Rekonstruktionsgüte der Trajektorie maximiert. Weil die Güten der Trajektorienpunkte nach (6.49) voneinander unabhängig sind, kann die Gesamtgüte kumulativ berechnet werden. Somit ist in jedem Zeitschritt die Fortsetzungsalternative mit der höchsten lokalen Rekonstruktionsgüte auszuwählen.

Trajektorienhypothesen, denen über einen längeren Zeitraum keine weiteren Punkte zugeordnet werden, können als abgeschlossen betrachtet und gelöscht werden. Weiterhin können alle Trajektorienhypothesen, für die längere Zeit keine sinnvolle paarweise Kombination existiert (d.h. die keine zeitliche Überlappung mit anderen Hypothesen aufweisen oder deren Rekonstruktionsgüte zu gering ist) ebenfalls verworfen werden. Aus Stabilitätsgründen sollte der Beobachtungszeitraum in beiden Fällen mehrere Bildaufnahmezyklen betragen, um zu verhindern, dass z.B. einzelne Detektionsausfälle oder die zeitweise Verdeckung einer Hand in einer Kamera zur Rückweisung einer korrekten Trajektorienhypothese führen.

*Aggregation in 3D*

Obige Vorgehensweise nimmt die Zuordnung von Punkthypothesen zu 2D-Trajektorienhypothesen für jede Kamera unabhängig vor und identifiziert anschließend sinnvolle paarweise Kombinationen auf Trajektorienebene. Das hat zwei entscheidende Nachteile. Erstens werden die Möglichkeiten eines Mehrkamerasystems nur unzureichend genutzt. Insbesondere führen Detektionsausfälle in einem Kamerabild ggf. dazu, dass eine eigentlich gute Trajektorienhypothese nicht weiter verfolgt werden kann, obwohl die fehlenden Beobachtungen in einer anderen Kamera zur Verfügung stünden.

17 Die zeitlichen Beziehungen gemäß (6.38) sind hier nicht anwendbar, weil aufgrund der Interpolation einander zugeordnete Punkte immer identische Zeitstempel besitzen.

Zweitens geht in die Bewertung einer Trajektorienhypothese nur die lokale Rekonstruktionsgüte auf Ebene der 3D-Punkthypothesen ein, nicht jedoch die Gestalt der Trajektorie. Hierdurch könnten Fehldetektionen, die zufällig eine Rekonstruktion mit hoher Güte ergeben, selbst dann gegenüber korrekten Hypothesen bevorzugt werden, wenn die resultierende Trajektorie einen unplausiblen Verlauf aufweist.

Aus diesem Grund ist eine Vorgehensweise zur Trajektorienaggregation zu bevorzugen, die auf der Ebene der 3D-Punkthypothesen operiert. Weil die einzelnen Hypothesen in den Kamerabildern unsynchronisiert und ggf. mit variierender Bildrate auftreten, muss zunächst eine künstliche Taktung eingeführt werden, indem in definierten Abständen alle Hypothesen aller Kameras innerhalb eines gegebenen Zeitfensters gesammelt werden. Aus diesen werden durch paarweise Kombination per Strahlenschnitt 3D-Hypothesen berechnet und ggf. durch die in Kapitel 6.6.4 vorgestellten Kriterien gefiltert. Als Resultat erhält man zu jedem Zeitpunkt $t$ eine Menge von $n_t$ 3D-Punkthypothesen mit Rekonstruktionsgüten gemäß

$$\begin{aligned} \lambda_q^R(t) &= P(\mathbf{h}_q(t)|\Delta t_q(t), d_q(t), l_A, c_q(t)) \\ &= P(\mathbf{h}_q(t)|\Delta t_q(t)) \cdot P(\mathbf{h}_q(t)|d_q(t)) \cdot P(\mathbf{h}_q(t)|l_A) \cdot P(\mathbf{h}_q(t)|c_q(t)), \\ q &= 1...n_t. \end{aligned} \tag{6.50}$$

Für eine erschöpfende Suche nach der optimalen Trajektorie müssten in jedem Zeitschritt alle möglichen Kombinationen von Punkthypothesen mit allen möglichen Trajektorienhypothesen erstellt und weiterverfolgt werden. Aus offensichtlichen Gründen ist ein derartiges Vorgehen für reale Probleme nicht handhabbar. Ein einfaches Suchverfahren, dass die explizite Auswertung aller möglichen Kombinationen vermeidet, ist der sog. *Beam Search* Algorithmus (vgl. z.B. [57], Kapitel 10.2).

Die Grundidee besteht darin, in jedem Zeitschritt die aktuell – im Sinne ihrer globalen Pfadbewertung – beste Hypothese zu bestimmen. Sei $h^*(t)$ die zum Zeitpunkt $t$ beste Hypothese mit der Pfadbewertung $\tilde{\lambda}^*(t)$. Es werden dann nur solche Hypothesen $h_q(t)$ weiter verfolgt, für deren aktuelle Bewertung gilt $\tilde{\lambda}_q(t) \geqslant \beta\tilde{\lambda}^*(t)$. Der Wert des Parameters $\beta$ definiert dabei also gewissermaßen die Breite eines Suchstrahles. Alle anderen Alternativen werden zum frühest möglichen Zeitpunkt verworfen.

Hierbei sollte die Definition der Pfadbewertung so gewählt werden, dass eine inkrementelle Berechnung möglich ist. Wenn die lokale Bewertung $\lambda_q(t)$ eine normierte Wahrscheinlichkeit ist, ergibt sich die Pfadbewertung unter der Annahme statistischer Unabhängigkeit als Produkt aller lokalen Bewertungen bis zum Zeitpunkt $t$, die zu dem entsprechenden Pfad gehören. Im vorliegenden Fall steht als lokale Bewertung die Rekonstruktionsgüte $\lambda_q^R(t)$ zur Verfügung. Diese Bewertung beinhaltet jedoch

nur die Rekonstruktionsgüte einer einzelnen 3D-Hypothese und bewertet nicht den Trajektorienkontext.

Eine einfache Möglichkeit zur Einbeziehung des Trajektorienkontextes besteht wiederum in der Betrachtung der Euklidischen und zeitlichen Distanz analog zu (6.46). Die Überführung in eine normierte Pseudowahrscheinlichkeit ist wie in (6.42) durch eine probabilistische Modellierung mit einer Normalverteilung möglich.

Eine bessere Alternative besteht jedoch darin, die Gestalt bzw. den Verlauf „guter" Trajektorienfortsetzungen in einem probabilistischen Modell zu erfassen. Hierfür können in einem kleinen Fenster, das die aktuelle Fortsetzungshypothese und einige wenige Vorgängerpunkte der betrachteten Trajektorie enthält, gestaltbeschreibende Merkmale berechnet werden. Erste Versuche mit einem GMM und Merkmalen, die den in Kapitel 6.7.3 vorgestellten Nachbarschaftsmerkmalen ähneln, deuten an, dass eine derartige Modellierung möglich und sinnvoll ist (Abbildung 31). Die Pfadbewertung zum Zeitpunkt $t$ für eine gegebene Fortsetzungshypothese ergibt sich somit zu

$$\begin{aligned}\lambda_q(t) &= \lambda^F_{p,q}(t)^\alpha \cdot \lambda^R_q(t)^{1-\alpha} \\ &= P(\mathbf{h}_q(t)|\Delta t_q(t), d_q(t), l_A, c_q(t))^\alpha \cdot P(\mathbf{h}_q(t)|\mathcal{T}_p(t-1))^{1-\alpha},\end{aligned} \tag{6.51}$$

mit der Güte der Fortsetzungshypothese $\lambda^F_{p,q} = P(\mathbf{h}_q(t)|\mathcal{T}_p(t-1))$ mit Trajektorie $\mathcal{T}_p(t-1)$. Hierbei ist $\alpha$ ein Gewichtungsfaktor, über den sich der Einfluß der beiden Terme einstellen lässt.

Bei vorliegender Problemstellung ergeben sich jedoch zwei grundlegende Unterschiede zum einfachen *Beam Search* Algorithmus: Erstens ist insbesondere bei der Handdetektion mit Detektionsausfällen, also fehlenden Beobachtungen, zu rechnen. Diese müssen speziell behandelt werden, weil ansonsten eine einzige fehlende Beobachtung zur Rückweisung einer bis dahin sehr guten Trajektorienhypothese führen kann. Zweitens kann nicht angenommen werden, dass alle beobachteten Trajektorien zum jeweils gleichen Zeitpunkt beginnen und enden. Es muss also möglich sein, Hypothesen mit unterschiedlichen Startpunkten zu verfolgen. Das ist einerseits problematisch, weil die Pfadbewertungen längenabhängig und somit für Pfade unterschiedlicher Länge nicht vergleichbar sind. Andererseits muss ein Kriterium gefunden werden, anhand dessen neue Trajektorienstartpunkte hypothetisiert werden können.

Die Behandlung fehlender Beobachtungen kann durch Einführung eines *Dummys* in jedem Zeitschritt erfolgen, der stellvertretend für alle nicht beobachteten Hypothesen steht. Dieser muss ebenfalls eine lokale Bewertung erhalten, die relativ zu den anderen im aktuellen Zeitschritt beobachteten Bewertungen gewählt werden sollte (beispielsweise ein fester Bruchteil der besten oder schlechtesten 3D-Hypothese). Dabei ist es vermutlich sinnvoll, die *Dummy*-Bewertung so zu wählen, dass sie etwa in der

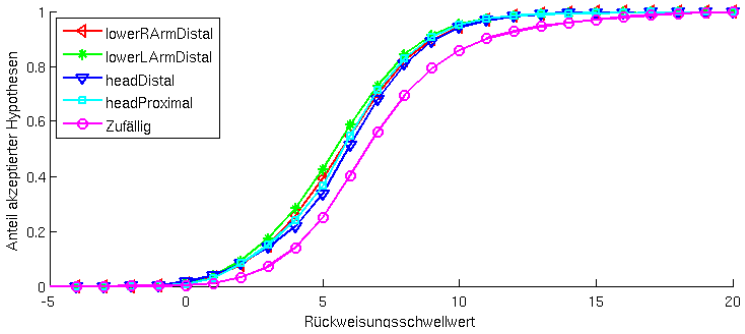


Abbildung 31: Rückweisungskurven bei Variation eines Rückweisungsschwellwertes über den Bewertungen (negative *Log-likelihoods*) eines GMM-Modelles (Codebuchgröße 30) für Trajektorien-Fortsetzungshypothesen (Fenstergröße drei). Die vier oberen Kurven stammen von realen 3D-Bewegungsmustern von Händen und Köpfen (HumanEVA *Motion Capture* Daten, siehe Kapitel 7.1.5). Die mit „Zufällig“ bezeichnete Kurve repräsentiert Trajektoriendaten, die aus realen Trajektorien durch Hinzufügen zufälliger Störungen erzeugt wurden. Die Tatsache, dass letztere Kurve unterhalb der anderen Kurven verläuft, zeigt, dass die fehlerhaften Trajektorienfortsetzungen im Mittel schlechtere Bewertungen erhalten. Eine derartige Modellierung eignet sich also für die vorgeschlagene Berechnung der Pfadfortsetzungsbewertungen.

Mitte des Suchstrahles liegt. Das spiegelt die Annahme wieder, dass die korrekte Trajektorienfortsetzung nicht beobachtet wurde, und diese eigentlich eine relativ gute Bewertung erhalten müsste. Das Suchverfahren kann dann normal fortgesetzt werden.

Hierbei taucht noch ein Problem bei der Berechnung der Gestaltbewertung einer Trajektorie auf, weil die Gestalt eines Trajektorienabschnittes mit *Dummy*-Hypothesen nicht definiert ist. Deren 3D-Position kann – beispielsweise mit der im folgenden Kapitel vorgestellten impulsbasierten Neuabtastung oder Spline-Interpolation – für die Merkmalsberechnung zwischen bekannten Trajektorienpunkten interpoliert werden.

Für das Einfügen neuer Trajektorienstartpunkte gilt: Die Wahrscheinlichkeit, dass eine Punkthypothese Startpunkt einer neuen Trajektorie ist, ist umso größer, je größer die lokale Rekonstruktionsgüte der Punkthypothese ist und je schlechter ihre Gestaltbewertung bei Kombination mit den bereits bestehenden Trajektorien ist. Demzufolge kann ein neuer Trajektorienstartpunkt aus der Menge der im aktuellen Zeitschritt verworfenen Punkthypothesen wie folgt gewählt werden: Gibt es in dieser Menge Hypothesen, deren lokale Rekonstruktionsgüte – bezogen auf die zum betrachteten

Zeitpunkt beste Hypothese – innerhalb des Suchstrahls[18] liegt, so sind diese mögliche Kandidaten für neue Startpunkte. Im weiteren Verlauf muss dann darauf geachtet werden, dass Pfadbewertungen von Pfaden unterschiedlicher Länge nicht vergleichbar sind. Deshalb müssen Trajektorienhypothesen mit ihrem Startpunkt bzw. ihrer Länge annotiert werden, so daß zu jedem Zeitschritt nur gleich lange Hypothesen verglichen werden. Es muss also für jede zu einem Zeitpunkt erzeugte Menge von Startpunkt-Hypothesen ein unabhängiger Suchprozess gestartet werden.

*Krümmungsadaptive Neuabtastung*

Die generierten 3D Trajektorien weisen aufgrund von Detektionsausfällen, variierenden Bildwiederholraten der Kameras und unterschiedlichen Ausführungsgeschwindigkeiten der Geste mitunter sehr unterschiedliche Punktabstände auf. Zudem entstehen durch die Interpolation und die Unsicherheiten bei der Projektion Lokalisierungsfehler, die zu Fehlern im Trajektorienverlauf führen. Dies erschwert die nachfolgende Klassifikation, weil diese Variationen unabhängig von der Art der Geste und somit für die Erkennung irrelevant sind. Wünschenswert wäre ein möglichst glatter Trajektorienverlauf mit gleichmäßiger Ortsauflösung. Aus diesem Grund werden die 3D Trajektorien einer sog. impulsbasierten Neuabtastung [207] unterzogen.

Die Grundidee besteht in der Einführung eines Impuls- oder Trägheitstermes, der die Orientierung des jeweils vorhergehenden Trajektorienabschnittes repräsentiert:

$$\begin{aligned} \hat{\tau}(0) &= \tau(0), \quad \mathbf{v}(0) = \frac{\tau(1)-\tau(0)}{\|\tau(1)-\tau(0)\|} \\ \mathbf{v}(t) &= \beta \frac{\tau_> - \hat{\tau}(t-1)}{\|\tau_> - \hat{\tau}(t-1)\|} + (1-\beta)\mathbf{v}(t-1) \\ \hat{\tau}(t) &= \hat{\tau}(t-1) + \mathbf{v}(t). \end{aligned} \tag{6.52}$$

Hierbei ist $\hat{\tau}(t)$ der Punkt der neuabgetasteten Trajektorie zum Zeitpunkt $t$ und $\tau_>$ ist der nächste Punkt der ursprünglichen Trajektorie, der noch nicht erreicht wurde. Der Glättungsfaktor $\beta$ legt fest, wie stark der Einfluß des Impulstermes ist. D.h. der Verschiebungsvektor $\mathbf{v}(t)$ vom aktuellen zum nächsten interpolierten Trajektorienpunkt ergibt sich als gewichtete Summe aus der tatsächlichen Richtung zum nächsten ursprünglichen Trajektorienpunkt $\tau_>$ und dem Verschiebungsvektor $\mathbf{v}(t-1)$ des vorhergehenden Zeitschrittes. Sprunghafte Richtungsänderungen werden demzufolge durch die schrittweise Interpolation der Richtung geglättet.

18 Die Suchstrahlbreite für diese Auswahl kann ggf. unabhängig von der globalen Suchstrahlbreite der Pfadbewertungen gewählt werden.

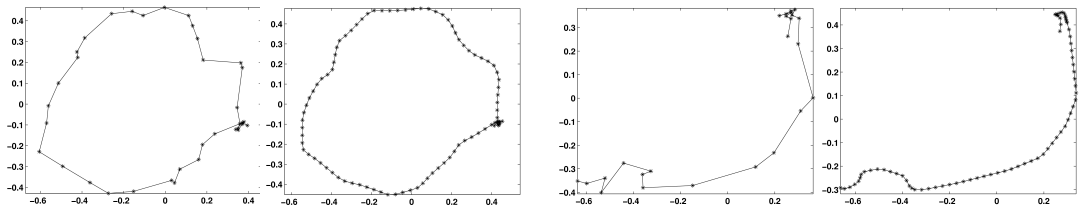

Abbildung 32: Beispiele für Gestentrajektorien und zugehörige Trajektorien nach der Neuabtastung.

Die Schrittweite der Abtastung ist immer kleiner oder gleich eins. Sie ist insbesondere in stark gekrümmten Trajektorienabschnitten kleiner als in geradlinigen [207]. Die Einführung des Impulsterms führt also implizit zu einer Anpassung der Abtastschrittweite an die Krümmung der Trajektorie. Das ist eine willkommene Eigenschaft, weil gekrümmte Bereiche, die potentiell mehr Information tragen, in feinerer Auflösung repräsentiert werden. Eine größere Schrittweite kann auf einfache Art erreicht werden, indem nur jeder $k$-te neuabgetastete Punkt zur Trajektorie hinzugefügt wird.

Abbildung 32 zeigt zwei Beispiele für die Ergebnisse der Neuabtastung. Neben der Erhöhung der Abtastrate ist deutlich zu erkennen, dass der verwendete Trägheitsterm zu einer Glättung des Trajektorienverlaufs führt.

Die optimale Wahl der Abtastungsparameter hängt stark von den betrachteten Daten ab. Insbesondere der Glättungseffekt kann kritisch sein, weil zwar einerseits Störungen vermindert, andererseits aber der Trajektorienverlauf verfälscht werden kann. Im weiteren Verlauf wird auf eine experimentelle Optimierung dieser Parameter verzichtet und ein Parametersatz verwendet, der auf einer Reihe typischer Trajektoriendaten gute Ergebnisse lieferte ($\beta = 0.9$, $k = 20$).

### 6.7.2 *Normalisierung*

Die Klassifikation einer Gestentrajektorie $\mathcal{T}$ anhand absoluter Punktkoordinaten ist in einem uneingeschränkten Szenario nicht erfolgversprechend, weil die verwendeten Merkmale von der absoluten Position der Person und ihrer Orientierung im Raum abhängen. Aus diesem Grund ist eine Normalisierung der Merkmale notwendig, um von den absoluten Punktkoordinaten zu abstrahieren. Hierfür existieren zwei verbreitete einfache Ansätze: Die Verwendung sog. *Delta-Merkmale* (vgl. z.B. [21, 132]) und die Transformation in ein personenzentrisches Koordinatensystem (vgl. z.B. [23, 132]). Darüber hinaus wurde im Rahmen dieser Arbeit eine weitere Methode entwickelt,

die auf der Projektion einer 3D-Gestentrajektorie auf die Hauptebene ihrer Bewegung beruht (im Folgenden als *Aktionsebene* bezeichnet).

*Normalisierung mit Delta-Merkmalen*

Die einfachste Möglichkeit, von den absoluten Werten eines Merkmales zu abstrahieren, besteht in der Betrachtung ihrer Veränderung über die Zeit. Hierfür kann in einem gleitenden Fenster die zeitliche Ableitung einer Folge von Merkmalsvektoren berechnet werden. Die Vereinfachung dieses Prinzips hin zur Betrachtung einfacher Differenzen zeitlich benachbarter Merkmalsvektoren $\mathbf{f}$ führt auf die sog. Delta-Merkmale:

$$\Delta\mathbf{f}_i = \mathbf{f}_i - \mathbf{f}_{i-1}, \quad i = 2...n. \tag{6.53}$$

Diese kodieren somit die relative Veränderung bezogen auf den vorhergehenden Zeitschritt. Dies ist offensichtlich für $\Delta\mathbf{f}_0$ nicht möglich, weshalb dieser Wert entweder auf Null bzw. den Wert von $\Delta\mathbf{f}_1$ gesetzt oder entfernt wird. Eine derartige Repräsentation ist zwar unabhängig von der absoluten Position, nicht jedoch von der globalen Orientierung, weil die Richtungen der Koordinatenachsen gleich bleiben.

*Normalisierung durch nutzerzentrisches Koordinatensystem*

Ein weiterer verbreiteter Ansatz zur Trajektoriennormalisierung definiert eine nutzerzentrisches Koordinatensystem, d.h. der Ursprung des Koordinatensystems wird durch eine definierte Stelle auf dem Körper des Nutzers vorgegeben. Im vorliegenden Szenario, bei dem die Orientierung des Nutzers im Raum unbekannt und uneingeschränkt ist, wird eine einfache Verschiebung des Koordinatenursprunges in den meisten Fällen fehlschlagen. Aus diesem Grund wird ein Polarkoordinatensystem gewählt, mit der Kopfposition der Person $\mathbf{h}_K$ als Ursprung. Jeder Punkt der polaren Trajektorie $\mathcal{T}^{pol}$ wird als Vektor $\tau_i^{pol} = (r_i, \sin\phi_i, \cos\phi_i)$ repräsentiert:

$$r_i = \frac{\|\tau_i - \mathbf{h}_K\|}{z_K}, \quad \phi_i = \arctan\left(\frac{\sqrt{(x_i^2 + y_i^2)}}{z_i}\right). \tag{6.54}$$

Hierbei ist $\phi_i$ der Polarwinkel zwischen der Vertikalen durch den Kopfmittelpunkt und der Verbindungslinie vom Kopfmittelpunkt zu $\tau_i$. Eine Größennormierung wird durch die Division durch die Kopfhöhe der Person $z_K$ erreicht, unter der Annahme, dass die Armlänge einer Person mit ihrer Körpergröße korreliert ist.

In obiger Repräsentation fehlt der Azimutwinkel, der gemäß der Orientierung der Person im Raum gewählt werden müsste. Da diese nicht bekannt ist, wird der

Azimutwinkel nicht betrachtet. Die Repräsentation ist somit unabhängig von der globalen Orientierung, auf Kosten eines Informationsverlustes.

*Normalisierung durch Projektion auf Aktionsebene*

Betrachtet man typische emblematische Armgesten und natürliche Armbewegungen, so fällt auf, dass sie häufig aus einfachen Bewegungen bestehen, bei denen hauptsächlich ein einziges Gelenk bewegt wird. Deshalb weisen diese Gesten eine inhärente Planarität auf. Die Hand bewegt sich annähernd in einer Ebene, die jedoch je nach Geste verschieden sein kann. Das legt nahe, dass die dreidimensionalen Trajektorien sich ohne großen Informationsverlust auf eine geeignete Ebene projizieren lassen.

Der Ansatz ist nun, die Bewegungsebene der Hand (im Folgenden als *Aktionsebene* bezeichnet) zu schätzen, so dass durch die Projektion einer Trajektorie auf ihre Aktionsebene eine Normalisierung erreicht wird. Ein ähnlicher Ansatz wurde in [157] verfolgt, um Schwenk- und Neigebewegungen einer aktiven Kamera auszugleichen. Im Gegensatz dazu wird im Folgenden davon ausgegangen, dass die Aktionsebene beliebig im Raum orientiert sein kann und nicht notwendigerweise mit einer der Kamera-Bildebenen übereinstimmt.

Inituitiv ist die Aktionsebene diejenige Ebene $\mathcal{E} : \mathbf{n}^T\mathbf{x} - \lambda = 0$, welche die durch die Trajektorienpunkte $\tau_i$ gegebene Punktewolke bestmöglich approximiert. Das führt auf ein multilineares kleinste-Quadrate Regressionsproblem (vgl. z.B. [119], Kapitel 3.5). Die Zielfunktion ergibt sich zu

$$f(\mathbf{n}) = \sum_{i_1}^{n} (n_x x_i + n_y y_i + n_z z_i - \lambda)^2 \to \text{Min}, \tag{6.55}$$

unter der Annahme dass $\|\mathbf{n}\| = 1$. Die Lösung dieses Problems ist wohlbekannt: Mit der zentrierten Stichprobenmatrix $\mathbf{M}$

$$\mathbf{M} = (x_i - \bar{x}, y_i - \bar{y}, z_i - \bar{z}), \quad \bar{\tau} = (\bar{x}, \bar{y}, \bar{z}) = \frac{1}{n}\sum_{i=1}^{n} \tau_i \tag{6.56}$$

ist die gesuchte Ebenennormale $\mathbf{n}$ durch den Eigenvektor mit dem kleinsten Eigenwert von $\mathbf{M}^T\mathbf{M}$ gegeben.

Die direkte Berechnung obiger Regression ist jedoch anfällig gegen fehlerhafte Trajektorienpunkte. Insbesondere können einzelne Ausreißer die Lösung stark beeinflussen. Aus diesem Grund wird vorher die *Konsensmenge* der Trajektorienpunkte mittels *Random Sample Concensus* (RANSAC [60]) gemäß Alg. 5 ermittelt. Die Aktionsebene ergibt sich durch Regression aus der Konsensmenge $\mathcal{C}_{\text{best}}$.

**Algorithmus 5** Ermittlung der Konsensmenge für die Ebenenschätzung mit RANSAC.

**Eingabe:** Trajektorienpunkte $\tau_i$, $i = 1...n$, Anzahl Iterationen $m$, Schwellwert $\delta$
$\mathcal{C}_{best} \leftarrow \emptyset$
**for** $k = 1...m$ **do**
  Konsensmenge $\mathcal{C}_k \leftarrow \emptyset$, Ausreißermenge $\mathcal{A}_k \leftarrow \emptyset$
  Wähle zufällig drei Punkte $\tau_o, \tau_p, \tau_q$.
  Berechne Ebene $\mathcal{E}_k : \mathbf{n}_k = \frac{(\tau_p - \tau_o) \times (\tau_q - \tau_o)}{\|(\tau_p - \tau_o) \times (\tau_q - \tau_o)\|}, \quad \lambda_k = \mathbf{n}_k^T \tau_o$
  **for** $i = 1...n$ **do**
    Berechne quadratische Distanz $d_{i,k} = (\mathbf{n}_k^T \tau_i - \lambda_k)^2$
  **end for**
  Berechne mittleren quadratischen Fehler $\bar{d}_{i,k} = \frac{1}{n} \sum_{i=1}^{n} d_{i,k}$
  **for** $i = 1...n$ **do**
    **if** $d_{i,k} > \delta \bar{d}_{i,k}$ **then**
      $\mathcal{A}_k = \mathcal{A}_k \cup \tau_i$
    **else**
      $\mathcal{C}_k = \mathcal{C}_k \cup \tau_i$
    **end if**
  **end for**
  **if** $\|\mathcal{C}_k\| > \|\mathcal{C}_{best}\|$ **then**
    $\mathcal{C}_{best} = \mathcal{C}_k, \quad \mathcal{E}_{best} = \mathcal{E}_k$
  **end if**
**end for**
**Ausgabe:** $\mathcal{E}_{best}, \quad \mathcal{C}_{best}$

Das Vorzeichen der resultierenden Ebenennormale $\mathbf{n}_{best}$ wird abschließend so gewählt, dass sie in Richtung der Kopfposition $\mathbf{h}_K$ zeigt. Anstelle des üblichen Auswahlkriteriums, nach dem diejenige Ebene mit der größten Konsensmenge ausgewählt wird, kann in obigem Algorithmus auch die Ebene mit minimalem mittleren quadratischen Fehler $\bar{d}_{i,k}$ gewählt werden. Die Resultate sind für beide Kriterien jedoch sehr ähnlich. Der Algorithmus benötigt offensichtlich mindestens drei Trajektorienpunkte, wodurch sich eine Latenz von drei Zeitschritten ergibt. Er lässt sich inkrementell unter Wiederverwendung der vorherigen Ergebnisse auf anlaufende Trajektorien anwenden.

Für bestimmte Gestentrajektorien, die im Wesentlichen aus einer geradlinigen Bewegung bestehen, also näherungsweise eindimensional sind, kann es vorkommen, dass keine eindeutig beste Ebene existiert. Das wird deutlich, wenn man von dem idealisierten Fall ausgeht, dass alle Trajektorienpunkte auf einer Geraden liegen: In diesem Fall haben alle Ebenen, welche diese Gerade enthalten, die gleiche Güte.

Deshalb kommt an dieser Stelle eine Heuristik zum Einsatz, welcher die Annahme zugrunde liegt, dass eine Kommandogeste üblicherweise an einen Adressaten in der Nähe gerichtet ist. Es ist somit unwahrscheinlich, dass die Aktionsebene einer Geste in Richtung Boden oder Decke ausgerichtet ist. Deshalb wird aus allen Ebenenkonfigurationen, für die $\|\mathcal{C}_k\| \leqslant \alpha\|\mathcal{C}_{best}\|$ gilt (d.h. deren Güte in Relation zur besten Lösung innerhalb eines durch $\alpha$ festgelegten Bereiches liegt), diejenige ausgewählt, für die $\cos\gamma = \mathbf{n}_k^T \cdot (0 \quad 0 \quad 1) = n_{z,k}$ (d.h. der Kosinus des Winkels zwischen der Ebenennormalen und der Vertikalen) minimal ist. Der Parameter $\alpha$ wird so gewählt, dass nur wenige gute Konfigurationen betrachtet werden (in vorliegender Arbeit ist üblicherweise $\alpha = 1.2$).

Bei der Projektion von $\mathcal{T}$ auf die so ermittelte Ebene muss ein geeignetes Koordinatensystem in $\mathcal{E}$ gewählt werden. Weil die Projektion der Normalisierung dient, sollten zudem Lage und Skalierung der projizierten Trajektorien normiert sein. Hierfür wurden zwei Möglichkeiten untersucht:

- **Größte Hauptkomponente:** Eine naheliegende Wahl ist, eine Koordinatenachse gemäß des Eigenvektors von $\mathbf{M}$ mit dem größten Eigenwert festzulegen. Die zweite Achse ergibt sich dann als deren Kreuzprodukt mit der Normalen. Dieses Vorgehen normiert die gesamte Lage der Trajektorie in der Ebene (Translation und Rotation), ist aber abhängig vom jeweils betrachteten Trajektorienausschnitt.

- **Parallel zur Grundebene:** Hier wird das Weltkoordinatensystem so in die Ebene rotiert, dass die y-Achse der Normalen entspricht und die x-Achse parallel zur Grundebene ist. Die neuen Koordinatenachsen entsprechen dann der x- und z-Achse des rotierten Koordinatensystems. Hierdurch bleibt die globale Orientierung der Trajektorie erhalten.

Als Koordinatenursprung wird die Projektion der Kopfposition – gemittelt über die Beobachtungslänge der Trajektorie – in die Aktionsebene verwendet. Sie stellt somit in gewisser Weise wiederum ein nutzerzentrisches Koordinatensystem dar. Abbildung 33 zeigt einige Beispiele nach zusätzlicher Lage- und Größennormalisierung. Es ist zu erkennen, dass Ausrichtung und allgemeine Form der Trajektorien gut durch die Projektion auf die Aktionsebene normiert werden. Die Größennormalisierung mit der mittleren Körpergröße kann Variationen in der Ausdehnung der Gesten nur bedingt ausgleichen. Das ist jedoch kaum überraschend, weil diese in viel größerem Maße von der konkreten Ausführung als von der Körpergröße abhängen.

Insbesondere für die beiden „Winken“ Gesten ist erkennbar, dass manche Trajektorien horizontal gespiegelt wiedergegeben werden. Der Grund hierfür ist einerseits die nicht eindeutige Wahl des Vorzeichens der Ebenennormalen, andererseits Unterschiede

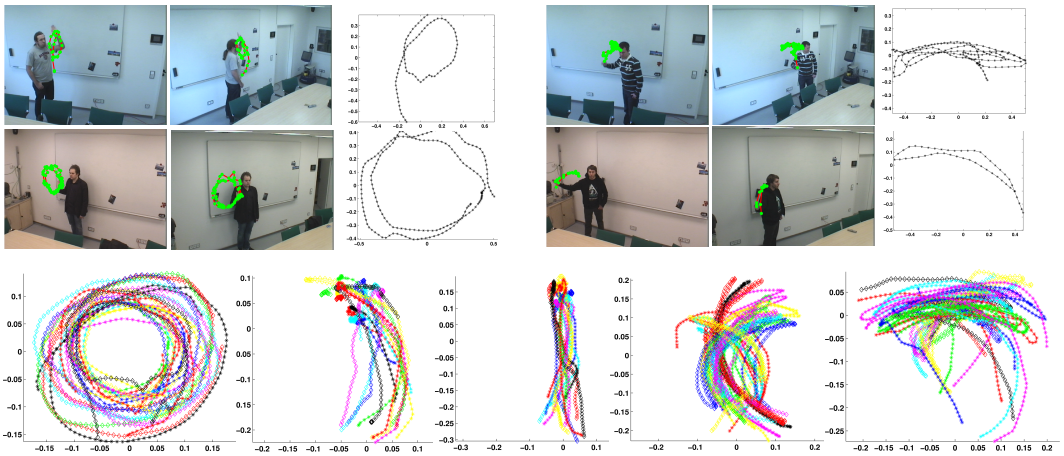

Abbildung 33: Beispiele für die Projektion von Gestentrajektorien auf die Aktionsebene. Die beiden oberen Reihen zeigen je zwei Beispiele einer „Kreis“ (links) und einer „horizontales Winken“ Geste (rechts). Zu sehen sind jeweils die Kamerabilder mit eingezeichneten 2D-Trajektorien und die normalisierte Repräsentation auf der Aktionsebene. Die untere Reihe zeigt übereinandergelegte Projektionen gleichartiger Gesten von verschiedenen Personen. Von links nach rechts: „Kreis“, „Stop“, „Hoch“, „vertikales Winken“ und „horizontales Winken“.

bei der Ausführung der Geste mit der rechten bzw. linken Hand. Ebenfalls ersichtlich ist, dass erhebliche Variationen in den Ausprägungen der Gesten erhalten bleiben. Dies liegt hauptsächlich an unterschiedlichen Start- und Endpunkten der hier gezeigten Trajektorien. Insbesondere beinhalten nicht alle Instanzen die Start- oder Endphase einer Geste, in der die Hand sich aus der bzw. in die Ruheposition bewegt. Weiterhin ist vor allem am Beispiel der „Stop“ Geste (untere Reihe, 2. v.l.) erkennbar, dass durch die Wahl des Koordinatensystems die globale Orientierung nicht normiert wird.

### 6.7.3 *Merkmale*

Die Analyse räumlich-zeitlicher Trajektorien von Hand- und/oder Gesichtspositionen wurde bereits erfolgreich für die Erkennung von Hand- und Armgesten eingesetzt. Campbell und Kollegen [21] untersuchten verschiedene Arten der Koordinatentransformation hinsichtlich ihrer Invarianz gegen Veränderungen des Blickwinkels. Die von ihnen vorgeschlagenen Repräsentationen stellen bis heute die vorherrschende Art von Merkmalen dar: Gewöhnlich werden die Koordinaten der Trajektorienpunkte in einem geeigneten invarianten Koordinatensystem direkt als Eingabe für einen Klassifikator

genutzt (vgl. Kapitel 4.6.4), ggf. in Verbindung mit formbeschreibenden Merkmalen der Hände (z.B. [105, 179]) oder optischem Fluß (z.B. [3, 23]). Gelegentlich werden die Punktkoordinaten vor der Klassifikation auf eine diskrete Symbolmenge abgebildet (z.B. [23, 45]).

Ähnliche Problemstellungen – die Klassifikation der Bewegung eines Punktes über die Zeit – existieren auch in anderen Bereichen der *Computer Vision*. Ein Beispiel hierfür ist die (online) Handschrifterkennung (vgl. z.B. [146] für einen guten Überblick über das Forschungsfeld), bei der die Trajektorie einer Stiftspitze verfolgt und klassifiziert wird. In diesem Bereich wurde eine Vielzahl alternativer trajektorienbasierter Merkmale vorgeschlagen. Beispiele hierfür sind Geschwindigkeit und Krümmung, zusammen mit gestaltbasierten Merkmalen kurzer Trajektoriensegmente [67], Hu Momente [36], Stiftdruck, Nachbarschaftsmerkmale, *Curliness* und Merkmale in Relation zur Grundlinie [59, 171]. Ebenso werden regelmäßig ansichtsbasierte Deskriptoren und strukturelle Merkmale wie Ober-, Unterlängen und Kreuzungspunkte verwendet (vgl. z.B.[67, 171]). Interessanterweise kamen derartige Merkmale in der Gestenerkennung – trotz der Ähnlichkeit der Aufgabenstellung – bisher kaum zur Anwendung. Das ist umso erstaunlicher, weil auch die verwendeten Klassifikationsmethoden sehr ähnlich sind: Aktuelle online Handschrifterkenner basieren entweder ebenfalls auf HMM (z.B. [171]) oder KNN (z.B. [67]). Aus diesem Grund liegt die Vermutung nahe, dass Merkmale aus dem Bereich der Handschrifterkennung ebenfalls gut zur Klassifikation von Gestentrajektorien geeignet sind. Diesem Ansatz folgend, wurden in vorliegender Arbeit verschiedene alternative Merkmalsrepräsentationen evaluiert.

Dabei gibt es nicht für alle der oben erwähnten Merkmale eine offensichtliche Analogie im Bereich der Gestenerkennung. So scheiden z.B. Merkmale, die auf dem Stiftdruck basieren, grundsätzlich aus. Handschrift-Merkmale, die sich auf die Grundlinie beziehen, sind nicht direkt auf Handtrajektorien übertragbar, weil im Gegensatz zu geschriebenem Text im Falle einer Geste nicht klar ist, was die Grundlinie sein soll bzw. ob diese existiert. Trotzdem bleiben viele verschiedene Merkmalsarten, die sich für die Gestenerkennung anpassen lassen und geeignet erscheinen.

Die untersuchten Merkmalsrepräsentationen sind größtenteils durch [67, 83] motiviert, mit einigen Anpassungen, um der geänderten Charakteristik der Daten gerecht zu werden. Dies umfasst insbesondere die Übertragung auf eine 3D-Trajektorie und Anpassungen bei Merkmalen, deren ursprüngliche Version relativ zu einer Grundlinie berechnet wird. Zur Erhöhung der Robustheit gegen Ausreißer und Störungen werden die Merkmale nicht punktweise, sondern in einem gleitenden Fenster berechnet. Sei $w$ die Fenstergröße und seien $\mathcal{T}_i = \{\tau_j, ...\tau_{j+w-1}, j = i * \lfloor \frac{w}{2} \rfloor\}$ Trajektorienpunkte innerhalb des $i$-ten Fensters (im folgenden werden der Einfachheit halber für 3D wie

auch 2D Repräsentationen auf der Aktionsebene die gleichen Bezeichner verwendet). Der Median des Fensters sei $\tau_i^m = \tau_{j+\lfloor \frac{w}{2} \rfloor}$. Folgende Merkmale wurden untersucht:

**Rohe Trajektorie:** Die mittlere Punkkoordinate des Fensters: $\bar{\mathcal{T}}_i = \frac{1}{w} \sum_k \tau_k, k = j...j+w-1$

**Normalisierte Trajektorie (3D):** $\hat{\mathcal{T}}_i = (\bar{\mathcal{T}}_i - \bar{\mathbf{h}}_K)/\bar{z}_K$, mit der mittleren 3D-Kopfposition $\bar{\mathbf{h}}_K$ und der mittleren Kopfhöhe $\bar{z}_K$. Der Sinn der Division durch $\bar{z}_K$ ist eine Normalisierung der Größe bzw. Ausdehnung der Gestentrajektorie unter der Annahme, dass diese von der Armlänge und diese wiederum von der Körpergröße abhängt.

**Normalisierte Trajektorie (2D):** $\hat{\mathcal{T}}_i = \frac{\bar{\mathcal{T}}_i - \bar{\mathcal{T}}}{\bar{z}_K}$. Die 2D-Trajektorie wird durch Subtraktion des Trajektorienmittels $\bar{\mathcal{T}}$ in Normlage gebracht und ihre Ausdehnung mit $\bar{z}_K$ normiert.

**Normalisierte polare Trajektorie (3D):** Dieses Merkmal entspricht der in (6.54) vorgestellten Polarrepräsentation, wobei $\tau_i$ durch $\bar{\mathcal{T}}_i$ und $\mathcal{H}_K$ durch $\bar{\mathcal{H}}_K$ ersetzt werden.

**Normalisierte polare Trajektorie (2D):** Analog zum 3D-Fall mit dem Unterschied, dass $r$ und $\phi$ relativ zum Koordinatenursprung berechnet werden.

**Geschwindigkeit:** Der mittlere Geschwindigkeitsvektor innerhalb des Fensters, d.h. $\mathbf{v}_i = \frac{1}{w} \sum_{k=j+1}^{j+w-1} \frac{\tau_k - \tau_{k-1}}{t_k - t_{k-1}}$, wobei $t_k$ der Zeitstempel von $\tau_k$ ist. Zusätzlich wird die mittlere absolute Geschwindigkeit $\bar{v}_i = \frac{1}{w} \sum_{k=j+1}^{j+w-1} \| \frac{\tau_k - \tau_{k-1}}{t_k - t_{k-1}} \|$ betrachtet. An dieser Stelle sei noch einmal daran erinnert, dass die Trajektorienpunkte nicht notwendigerweise gleichbleibende zeitliche Abstände aufweisen. Die Geschwindigkeit ist daher nicht identisch zur (skalierten) Delta-Trajektorie.

**Krümmung:** Die Krümmung ergibt sich als Sinus und Kosinus des Winkels zwischen den Vektoren $(\tau_i^m - \tau_j)$ und $(\tau_{j+w-1} - \tau_i^m)$.

**Nachbarschaft:** Diese Merkmale sollen die allgemeine Form eines Merkmalsfensters beschreiben. Sei $\mathbf{u}_i = \tau_{j+w-1} - \tau_j$ der Verbindungsvektor zwischen den Randpunkten des Fensters. Daraus ergibt sich der sog. Nachbarschaftsaspekt (engl. *vicinity aspect*) $\eta_{yx} = \frac{|u_{y,i}| - |u_{x,i}|}{|u_{y,i}| + |u_{x,i}|}$ für die 2D Trajektorie. Für die 3D Trajektorie umfasst dieses Merkmal die Werte $\eta_{yx}, \eta_{zx}, \eta_{zy}$. Weiterhin umfassen die Nachbarschaftsmerkmale die globale Orientierung, repräsentiert als Sinus

und Kosinus des Winkels zwischen $\mathbf{u}_i$ und der x-Achse (2D) bzw. der Projektion von $\mathbf{u}_i$ auf die Grundebene (3D), die sog. Welligkeit (engl. *Curliness*)[19] $l_i = \frac{\|\mathbf{u}_i\|}{\sum_{k=j+1}^{j+w-1} \|\tau_k - \tau_{k-1}\|}$ sowie den mittleren quadratischen Abstand der $\tau_k$ von $\mathbf{u}_i$ (in [83] als *Linearität* bezeichnet).

**Orientierungsänderung:** Für zwei aufeinanderfolgende Fenster $\mathcal{T}_i$ und $\mathcal{T}_k$ ergibt sich die Orientierungsänderung als Kosinus und Sinus des Winkels zwischen $\mathbf{u}_i$ und $\mathbf{u}_k$.

**Kopfabstand:** $\bar{d}_i^H = \frac{1}{wz_K} \sum_{k=j}^{j+w-1} \|\tau_k - \bar{\mathbf{h}}_K\|$. Dieses Merkmal kodiert eine schwache Repräsentation der relativen spatialen Lage der gestikulierenden Hand und des Kopfes.

Zusammengenommen ergeben alle vorgestellten Merkmale einen 20 (2D) bzw. 25 (3D) dimensionalen Merkmalsvektor. Durch Hinzunahme der Delta-Merkmale verdoppelt sich die Dimenisonalität.

### 6.7.4 *Detektion und Klassifikation mit HMM*

Für die Detektion des Auftretens einer Geste und die Klassifikation der Trajektorien kommen Hidden Markov Modelle zum Einsatz (vgl. Kapitel 5.6). Für jede Geste wird ein individuelles Modell trainiert. Dabei werden einfache lineare und sog. *Bakis* Topologien in Betracht gezogen (Abbildung 34 links). In einem linearen HMM sind – neben Selbstübergängen – nur Übergänge von einem Zustand in den direkten Folgezustand möglich. In einer Bakis-Topologie können Zustände übersprungen werden. Somit können Variationen aufgrund von Teilsequenzen, die nicht in jeder Instanz einer Geste beobachtet wurden, besser modelliert werden, allerdings auf Kosten einer größeren Modellkomplexität. Die Wahl der Modelllänge (d.h. die Anzahl der Zustände) erfolgt automatisch relativ zur minimalen Beobachtungslänge der jeweiligen Geste in der Trainingsstichprobe. Die Normalverteilungen des GMM werden mit diagonalen Kovarianzmatrizen modelliert. Das Training der Parameter erfolgt mit dem normalen Baum-Welch-Algorithmus.

In der Erkennungsphase werden die Gestenmodelle mittels *Viterbi Beam Search* ([57] Kapitel 10.2) dekodiert. Abbildung 34 (Mitte) zeigt die resultierende Modell- bzw.

19 Dieses Merkmal repräsentiert das Verhältnis zwischen der kürzesten Verbindungslinie der Randpunkte und der Trajektorienlänge. Je stärker gekrümmt der Verlauf der Trajektorie innerhalb des Fenster ist, umso kleiner wird $l_i$. Im Unterschied zu [67, 83] wird als Referenz nicht die längste Ausdehnung entlang einer der Koordinatenachsen gewählt, um eine Abhängigkeit von der globalen Lage zu vermeiden.

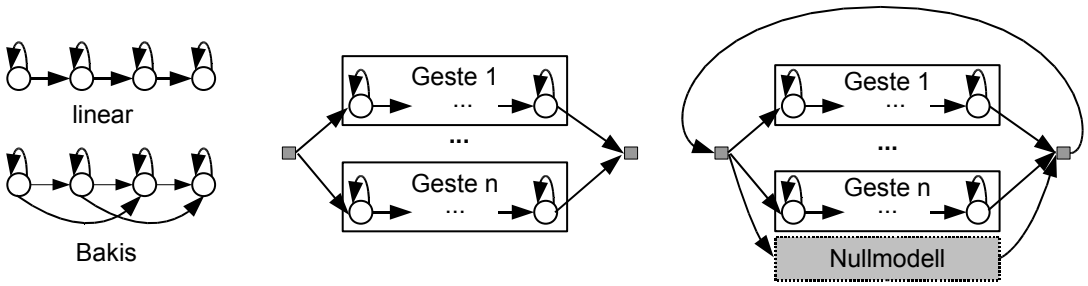


Abbildung 34: HMM Modellstruktur. Links: Lineare (oben) und Bakis (unten) Topologie. Mitte: Modellaufbau für Klassifikation. Rechts: Aufbau für Segmentierung/Detektion. Die globalen Start- und Endzustände (graue Quadrate) dienen nur der Übersichtlichkeit und sind nicht Teil des eigentlichen Modelles.

Verarbeitungsstruktur. Hier muss zwischen Klassifikations- und Detektionsaufgaben unterschieden werden: Bei der Klassifikation wird von segmentierten Daten ausgegangen. Jede Beobachtungssequenz entspricht somit exakt einer Gesteninstanz. Alle Modelle werden parallel dekodiert und dasjenige, welches zur optimalen Zustandssequenz $\mathbf{s}^*$ korrespondiert, definiert die erkannte Klasse.

Die Detektion beinhaltet die Klassifikation, operiert aber auf unsegmentierten Sequenzen, die eine unbekannte Anzahl Gesteninstanzen unterschiedlicher Art und Länge enthalten können. Zunächst muss also eine sinnvolle Segmentierung gefunden werden. Den Segmenten wird dann ein Klassenkennzeichen zugeordnet. Hierfür muss eine Rückkante eingeführt werden, welche die Endzustände der Modelle mit den Startzuständen verbindet (Abbildung 34 rechts). Es ergibt sich ein zyklischer Graph, der während der Dekodierung mehrfach durchlaufen werden kann und so die Aufteilung einer Sequenz in disjunkte Segmente ermöglicht. Tritt bei der Rückverfolgung des optimalen Viterbi-Pfades (d.h. bei der Ermittlung von $\mathbf{s}^*$) ein Übergang zwischen Start- und Endzuständen auf, so wird an dieser Stelle eine Segmentgrenze eingefügt. Die Klassifikation der Segmente ergibt sich wiederum aus dem während des jeweiligen Segmentes aktiven Modell.

Um auch mit Nichtgesten-Segmenten umgehen zu können, muss das Modell zusätzlich um ein Rückweisungskriterium erweitert werden. Hier wird ein Rückweisungsmodell (engl. *garbage* oder *background model*) verwendet, das ursprünglich für kontinuierliche Spracherkennung entwickelt wurde (siehe z.B. [5]). Dabei handelt es sich um ein zusätzliches HMM, das mit Nichtgesten-Daten trainiert wird. Während der Dekodierung tritt dieses dann als gleichberechtigtes Modell in Konkurrenz zu den

Gestenmodellen. Subsequenzen, die bei der Dekodierung dem Rückweisungsmodell zugewiesen wurden, werden aus der Erkennungshypothese entfernt.

Darüberhinaus wird als weiteres Rückweisungskriterium auf Einzelhypothesenbasis der sog. *log-odd scores* Ansatz (s. z.B. [44]) verwendet. Die Grundidee besteht darin, die Emissionswahrscheinlichkeiten der HMM-Zustände auf eine geeignete Hintergrundverteilung $P_H(\mathbf{x})$ zu normieren:

$$\hat{b}_j(\mathbf{x}) = \frac{b_j(\mathbf{x})}{P_H(\mathbf{x})}. \tag{6.57}$$

Hierbei ist $\hat{b}_j(\mathbf{x})$ die modifizierte Emissionswahrscheinlichkeit des HMM-Zustandes j für die Beobachtung $\mathbf{x}$. Für das hier verwendete semi-kontinuierliche HMM wird $P_H(\mathbf{x})$ gemäß der a-priori-Wahrscheinlichkeiten der GMM-Mischungskomponenten gewählt [148]. Werden die während der Dekodierung solcherart berechneten Wortbewertungen noch auf die Wortlänge normiert, ergibt sich ein vergleichbares Bewertungsmaß, anhand dessen mittels eines globalen Schwellwertes eine Rückweisung nicht plausibler Worthypothesen vorgenommen werden kann. Dies gilt unter der Annahme, dass falsch-positive Hypothesen durch die Gestenmodelle schlechter repräsentiert sind, als korrekte Gesteninstanzen, und somit üblicherweise schlechtere Bewertungen erhalten.

Das beschriebene Erkennungssystem wird mit der Open Source Toolbox *ESMERALDA* [58] realisiert. Diese bietet effiziente und flexible Implementierungen für Modellierung, Training und Dekodierung.

## 6.8 MODELLIERUNG DER UMGEBUNG

Für die Auswertung gestenbasierter Kommandos in einer intelligenten Umgebung wird eine Repräsentation der Umgebung benötigt. Zum Einen definiert ein Umgebungsmodell ein globales Referenzkoordinatensystem. Zum Anderen sind für die Auswertung einer Kommandogeste mitunter Kenntnisse über Objekte, andere Personen oder Positionen im Raum notwendig, um Adressaten oder Ziele identifizieren zu können. Beispiele hierfür sind Zeigegesten, deren Auswertung nur in Verbindung mit dem angezeigten Ziel möglich ist, oder Kommandogesten zur Steuerung eines bestimmten Gerätes. Um die Intention des Nutzers bestimmen zu können, muss in beiden Fällen bekannt sein, wo sich mögliche Ziele bzw. Interaktionspartner in der Umgebung befinden und welche Bedeutung diese haben bzw. welche Funktionalitäten sie anbieten können.

Wie im Vorfeld bereits erörtert wurde, ist die komplette Modellierung einer Umgebung und ihres Kontextes für sich alleine genommen eine sehr komplexe Aufgabe,

sofern nicht gravierende Vereinfachungen und Einschränkungen vorgenommen werden. Eine erschöpfende Behandlung dieses Problemes würde den Rahmen dieser Arbeit sprengen. Im Folgenden wird daher lediglich ein einfacher Modellierungsansatz vorgeschlagen, der von einem bekannten Umgebungsmodell und gegebenen Objekten ausgeht. Dieser ermöglicht einfache gestenbasierte Interaktion mit bekannten Objekten und Geräten und ist flexibel genug gehalten, um ggf. zur Laufzeit automatisch ergänzt und erweitert zu werden. Abschließend wird noch konkret auf die Modellierung von Zeigerichtungen im Raum eingegangen.

### 6.8.1 *Einfaches Umgebungsmodell*

Die realisierte Darstellung der Umgebung umfasst eine pfadbasierte Definition der äußeren Begrenzungen sowie eine Modellierung von Objekten, entweder als geometrische Primitive oder als unscharfe *Blobs*. Die unscharfe Modellierung erlaubt die Angabe ggf. überlappender Einzugsbereiche (vgl. [163]), so dass Objektreferenzen in probabilistischer Weise modelliert werden können. Die Implementierung umfasst zudem geometrische Schnitttests von Sichtstrahlen mit Objekten oder Begrenzungsflächen sowie Abstandsberechnungen zwischen Objektmittelpunkten und Strahlen, so dass z.B. die in Kapitel 6.6.4 vorgestellte Modellierung von Verdeckungen möglich ist.

Für jedes Objekt kann zudem eine Menge von Attributen angegeben werden, die dessen Funktionen und Interaktionsmöglichkeiten charakterisieren. Über diese Attribute kann eine Zuordnung von möglichen Kommandogesten zu Funktionen hergestellt werden. Sowohl diese Zuordnung, die Attribute eines Objektes als auch dessen Position und Ausmaße müssen in der vorhandenen Realisierung bekannt sein und werden durch eine einfache Beschreibungssprache definiert (Abbildung 35). Ansätze zur automatischen Erkennung und Modellierung unbekannter Objekte in der Umgebung werden z.B. in [169] aufgezeigt. Die dynamische Generierung von Objektattributen und -funktionalitäten könnte durch die vorhandene KNX-Gebäudeinstallation gelöst bzw. unterstützt werden, indem vorhandene Geräte sich bei einem Geräteserver registrieren und ihre unterstützen Funktionen bekannt machen. Ein Prototyp einer derartigen dynamischen Systemkonfiguration – allerdings basierend auf einem statisch hinterlegten Geräteverzeichnis – wurde z.B. in [208] realisiert.

Referenzen auf benutzbare Objekte können entweder durch explizites Zeigen oder durch zusätzliche Modalitäten (z.B. eine Spracherkennung) definiert werden. Auch die Orientierung der Aktionsebene einer Geste könnte ausgewertet werden, weil ein Nutzer sich üblicherweise seinem Interaktionspartner zuwendet. Inwiefern dieser Ansatz geeignet ist, wurde jedoch im Rahmen dieser Arbeit nicht mehr untersucht.

```
NODE:    0.0    0.0    0.0
NODE: 4650.0    0.0    0.0
NODE: 6790.0 2450.0    0.0
NODE: 6790.0 4030.0    0.0
NODE:    0.0 4030.0    0.0
ROOMHEIGHT:  2550.0
```

```
BEGIN OBJECT
 TYPE:   blob
 CENTER: 1580.0 1210.0  700.0
 SIZE:     50.0   50.0   50.0
 ID:     marker1
 BEGIN PROPERTIES
  pointable
 END
END
```

Abbildung 35: Beispiel für die einfache Konfigurationssprache zur Szenenbeschreibung (Ausschnitt der Beschreibung, die in [163] für die experimentelle Evaluierung verwendet wurde). Links: Pfadbasierte Beschreibung des Konferenzraumes der FINCA. Die Knoten werden in der gegebenen Reihenfolge verbunden. In Verbindung mit der angegebenen Raumhöhe können Begrenzungsflächen berechnet werden, die den Wänden entsprechen. Rechts: Beispiel für eine Objektdefinition als symmetrischer Gauss'scher *Blob* mit einer Standardabweichung von 50 mm in jede Koordinatenrichtung. Das Objekt hat die Eigenschaft *pointable*, was bedeutet, dass es Ziel einer Zeigegeste sein kann.

### 6.8.2 *Modellierung einer Zeigerichtung*

Besondere Beachtung verdient die Modellierung der angezeigten Richtung einer Zeigegeste. Durch die explizite Verfolgung der Zeigerichtung können Referenzen auf Objekte oder andere Personen definiert werden, die sich – im Gegensatz zur Zielidentifikation durch räumliche Nähe wie z.B. in [77] – zum Beobachtungszeitpunkt auch außerhalb des Sichtbereiches der Kameras befinden können. Die Möglichkeit, auf diese Weise Objekte zu identifizieren bzw. sogar unbekannte Objekte zu lernen, wurde z.B. in [169] (in 2D) untersucht, und die im Folgenden vorgestellte Modellierung wurde im Rahmen dieser Veröffentlichung mit Erfolg eingesetzt.

Ein verbreiteter Ansatz besteht darin, die Zeigerichtung als Sichtlinie von den Augen zur zeigenden Hand zu definieren (vgl. z.B. [70, 91, 176]). In [132] werden drei verschiedene Möglichkeiten (Sichtlinie, Unterarmorientierung und Kopforientierung) zur Richtungsmodellierung untersucht, von denen die Sichtlinie die höchste Genauigkeit erreichte. Intuitiv entspricht dies auch am Ehesten dem realen Zeigevorgang auf weit entfernte Ziele, bei dem das Ziel mit den Augen über die zeigende Hand bzw. den Zeigefinger angepeilt wird. Jedoch ist diese Art der Modellierung weniger geeignet, wenn auf Objekte in unmittelbarer Nähe gezeigt wird, weil in diesen Fällen eher die Orientierung des Zeigefingers oder des Unterarmes ausschlaggebend ist.

Tatsächlich ist eine Zeigerichtung immer mit Unsicherheit behaftet, wie z.B. in [98] diskutiert wird. Anhand der darin vorgestellten Experimente schließen die Autoren, dass eine menschliche Zeigegeste eine Winkelunsicherheit von ungefähr zehn Grad aufweist. Diese Unsicherheit sollte bei der expliziten Verfolgung einer Zeigegeste durch den Raum und der Bestimmung des angezeigten Zieles beachtet werden.

Diese Überlegungen führen zu der Idee, die Zeigerichtung als unscharfen Kegel zu modellieren. Als Grundlage wird das Sichtlinienmodell gewählt, weil es im vorliegenden Szenario wahrscheinlich ist, dass Referenzen auf weit entfernte Objekte häufiger auftreten, als auf räumlich sehr nahe Objekte. Eine alternative Behandlung „naher" Zeigegesten bleibt davon unberührt, wird hier jedoch nicht näher betrachtet. Im Folgenden wird zunächst der für [169] entwickelte zweidimensionale Ansatz erläutert. Anschließend wird die Erweiterung auf 3D vorgestellt.

*Modellierung in 2D*

Gemäß des Sichtlinienansatzes ist die initiale Zeigerichtung durch den Vektor vom Kopf- zum Handmittelpunkt gegeben. Der Ursprung befindet sich im Handmittelpunkt. Sei $\mathbf{r}_D = (x_D, y_D, w_D, h_D)$ eine Kopfhypothese und $\mathbf{r}_H = (x_H, y_H, w_H, h_H)$ eine Handhypothese. Der Richtungsvektor ergibt sich somit als

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \quad \mathbf{v} = (x_H - x_D, y_H - y_D). \tag{6.58}$$

Der Ursprung der Zeigerichtungsschätzung ist $\mathbf{o} = (x_H, y_H)$. Um den Vektor $\hat{\mathbf{v}}$ wird ein unscharfer Bereich definiert, der die inhärente Ungenauigkeit der Zeigegeste und die Unsicherheit der Hypothesenlokalisierung beinhaltet. Hierbei wird die Unsicherheit der Kopflokalisierung modelliert durch eine Normalverteilung

$$\mathcal{N}(\|(x - x_D, y - y_D)\|, \sigma_D), \quad \sigma_D = \bar{w}_D/8, \tag{6.59}$$

mit der über einen Beobachtungszeitraum gemittelten Breite des Kopfhypothesenrechteckes $\bar{w}_D$. Die Standardabweichung der Normalverteilung umfasst ein Achtel der mittleren Breite der Kopfhypothese. Diese relativ große Unsicherheit trägt der Tatsache Rechnung, dass die Schätzung der Augenposition als Mittelpunkt des Detektionsrechteckes sehr ungenau ist.

Weitere Unsicherheitsquellen sind die Variationen von $w_D$ und die Positionsunsicherheit der Kopf- und Handhypothesen. Diese werden als unabhängige normalverteilte Störungen mit den Standardabweichungen $\sigma_w$ und $\sigma_p$ modelliert. Die Standardabweichungen ergeben sich dabei als *Maximum Likelihood* Schätzungen über alle


Abbildung 36: Beispiel für die probabilistische Modellierung einer (2D) Zeigerichtung. Links: Eingabebild mit Kopf- und Handhypothese. Rechts: Darstellung von $p(\alpha(\mathbf{p})|\hat{\mathbf{v}})$ (helle Werte = hohe Wahrscheinlichkeit).

Hypothesen eines definierten Beobachtungszeitraumes $\Psi = \{t_j, j = i - n...i\}$, wobei $t_i$ der aktuelle Beobachtungszeitpunkt ist:

$$\sigma_w^2 = Var(w_D(t_j)), \quad \sigma_p^2 = Var(\arccos((sign(x_H - y_D), 0)^T \hat{\mathbf{v}}(t_j))). \tag{6.60}$$

Unter der Annahme, dass eine Zeigegeste eine Haltephase beinhaltet, kann von einer konstanten Handposition ausgegangen werden, wenn $|\Psi|$ klein genug ist.

$\sigma_p$ ist eine Standardabweichung in Winkeln, während $\sigma_w$ und $\sigma_D$ Standardabweichungen in Euklidischen Distanzen sind. Letztere lassen sich durch Division durch $r = \|\mathbf{v}\|$ näherungsweise in Winkelwerte überführen. Die gesamte Unsicherheit der Zeigerichtung ergibt sich durch Kombination der einzelnen Störeinflüsse zu

$$\begin{aligned} p(\alpha(\mathbf{p})|\hat{\mathbf{v}}) &= \mathcal{N}(\alpha(\hat{\mathbf{v}}), (\sigma_D + \sigma_w)/r + \sigma_p), \\ \alpha(\mathbf{x}) &= \arccos((sign(x_H - y_D), 0)^T (\mathbf{x} - \mathbf{o})). \end{aligned} \tag{6.61}$$

Die Wahrscheinlichkeit $P(\alpha(\mathbf{p})|\hat{\mathbf{v}})$, dass ein beliebiger Punkt $\mathbf{p}$ auf der Bildebene durch die aktuelle Zeigerichtungsschätzung referenziert wird, ergibt sich durch Anwendung der obigen Formel. Das Resultat ist in Abbildung 36 grafisch dargestellt.

*Erweiterung auf 3D*

Die vorgestellte Modellierung erlaubt die Verfolgung einer Zeigerichtung in eingeschränkten monokularen Szenarien. Jedoch ist die erreichbare Genauigkeit hierbei durch die fehlende Tiefeninformation begrenzt und dieser Ansatz ist für beliebige

Szenen so nicht anwendbar. Andererseits wurde in [163] demonstriert, dass die Verfolgung einer 3D-Zeigerichtung im Raum erfolgreich vorgenommen werden kann. In dieser Veröffentlichung wurden bekannte Objekte in der Szene als unscharfe *Blobs* dargestellt. Die Modellierung der positionellen Unsicherheit des Zeigezieles wurde somit auf das Ziel selbst verlagert, was problematisch ist, weil beispielsweise die Entfernung des Ziels von der zeigenden Person dabei nicht berücksichtigt wurde.

Deshalb wird obiger Ansatz zur expliziten Modellierung der Unsicherheit der Zeigerichtung auf 3D-Hypothesen übertragen. Somit sind $\hat{\mathbf{v}}$ und $\mathbf{o}$ nun dreidimensionale Vektoren. Die Schätzung von $\sigma_p$ kann analog zu (6.60) erfolgen:

$$\sigma_p^2 = Var(\arccos((sign(\bar{\mathbf{o}} - \bar{\mathbf{x}_D}), 0)^T \hat{\mathbf{v}}(t_j))). \tag{6.62}$$

Hierbei ist $\bar{\mathbf{o}}$ die mittlere Handposition und $\bar{\mathbf{x}_D}$ die mittlere Kopfposition. Die Standardabweichungen $\sigma_D$ und $\sigma_w$ wurden im zweidimensionalen Fall anhand der Größe des Kopf-Schulter-Detektionsrechteckes bestimmt. Eine analoge Vorgehensweise im Dreidimensionalen wäre möglich, indem die Detektionsrechtecke auf eine Ebene parallel zur jeweiligen Kameraebene durch die 3D-Kopfhypothese projiziert und über die Größen der projizierten Rechtecke aus allen Kameras gemittelt wird. Aus Gründen der Effizienz und Einfachheit wird hier jedoch ein anderer Ansatz gewählt.

In Kapitel 6.6.4 wurde bereits erwähnt, dass ein Zusammenhang zwischen der Körpergröße und der Kopfgröße einer Person existiert. Demnach kann die mittlere Kopfgröße $\bar{w}_D$ über den Beobachtungszeitraum abgeschätzt werden als $\bar{w}_D = \frac{\bar{z}_D}{7}$, mit der mittleren z-Koordinate der Kopfhypothese $\bar{z}_D$. Diese Größenschätzung kann nun analog zu (6.59) zur Abschätzung der Unsicherheit der Augenposition verwendet werden. Genauso ergibt sich dann $\sigma_w^2 = Var\left(\frac{z_D(t_j)}{7}\right)$, analog zu (6.60).

Um den Erkenntnissen in [98] Rechnung zu tragen, wird für den Wert von $\sigma_c = (\sigma_D + \sigma_w)/r + \sigma_p$ ein Minimum von drei Grad festgelegt. Damit umfassen $3\sigma_c$ einen Korridor von mindestens neun Grad. Der vorgeschlagene Ansatz beschreibt somit die inhärente Ungenauigkeit einer Zeigegeste sowie die Wahrscheinlichkeit, dass ein bestimmtes Szeneobjekt das Ziel dieser Geste ist, in einem integrierten probabilistischen Modell. Die Bestimmung des Sichtstrahles erlaubt die explizite Verfolgung der angezeigten Richtung durch die Szene, was beispielsweise zur Nachführung aktiver Kameras (vgl. [169]) genutzt werden kann. Zusätzlich ermöglicht die probabilistische Formulierung die Bewertung von Szeneobjekten gemäß ihrer Lage relativ zum Sichtstrahl. Damit kann eine sortierte Liste möglicher Objektreferenzen erstellt werden, so dass mehrere alternative Hypothesen verfolgt werden können. Die Effektivität dieses Vorgehens wurde in [169] für monokulare Szenen demonstriert.

# 7 EVALUIERUNG

Das folgende Kapitel evaluiert die realisierten Systembestandteile anhand realistischer Daten und Szenarien. Die Reihenfolge folgt dabei der in Kapitel 6 erarbeiteten Struktur. Begonnen wird mit der Bewertung der Hintergrundmodellierung, gefolgt von der Evaluierung der Personendetektion mit HOG-Deskriptoren. In diesem Zusammenhang wird auch aufgezeigt, inwiefern sich die Suchraumeinschränkung per Hintergrundmodellierung auf die Effizienz des Verfahrens auswirkt. Daran anschließend werden die vorgestellte Ansätze zum farbbasierten *Tracking* der detektierten Person und zur Detektion ihrer Hände betrachtet. Letzteres schließt die Evaluation der Hautfarbmodellierung ein. Das Kapitel schließt mit Ergebnissen der trajektorienbasierten Gestenerkennung mittels HMM.

## 7.1 DATENSÄTZE

Bevor mit der eigentlichen Evaluierung begonnen wird, werden zunächst die verwendeten Datenätze detailliert vorgestellt. Leider existieren nach Wissen des Autors keine geeigneten etablierten Standarddatensätze im Bereich der Multikamera 3D Gestenerkennung, so dass überwiegend auf proprietäre Daten zurückgegriffen werden muss, die in der FINCA am Institut für Roboterforschung der TU Dortmund aufgenommen wurden. Eine Ausnahme hierzu bildet der für das Training des HMM-Rückweisungsmodelles herangezogene HumanEVA-Datensatz [182]. Die eingesetzten Bewertungsmaße werden jeweils an der Stelle ihrer ersten Verwendung erläutert.

### 7.1.1 *FINCA Personen- und Handdetektions-Datensatz (FINCA-PH)*

Der Zweck dieses Datensatzes besteht in der Bewertung der 2D-Bildverarbeitungspipeline, also der Detektionsleistung der vorgestellten Verfahren zur Personenlokalisierung und Handdetektion in monokularen Bilddaten. Er enthält drei verschiedene Kameraansichten von zehn verschiedenen Personen. Pro Bild ist jeweils eine Person zu sehen, die sich in sehr unterschiedlichen Abständen von der Kamera und vor verschiedenartigen Hintergründen befindet (Abbildung 37). Die Testpersonen wurden gebeten mit beiden Armen beliebige Bewegungen zu vollführen. Die Personen sind dabei der Kamera zu-

Abbildung 37: Beispiele aus dem Datensatz FINCA-PH.

gewandt und gestikulieren mit geöffneten Händen, so dass in der Mehrzahl der Fälle die Handflächen sichtbar sind. Bezüglich Kleidung der Personen sowie Art, Position und Geschwindigkeit der ausgeführten Bewegungen gibt es keine Beschränkungen. Somit enthält der Datensatz auch viele Beispiele, in denen sich Hände in abgeschatteten Bereichen befinden oder Teile des Kopfes durch Hände oder Unterarme verdeckt werden. Der Datensatz enthält insgesamt 3110 RGB-Farbbilder in einer Auflösung von 378×278 Pixel. Köpfe und Hände wurden manuell mit umschließenden Rechtecken annotiert. Die genaue Zusammensetzung ist in Tabelle 1 angegeben.

| Person | m/w | Kamera 1 | Kamera 2 | Kamera 3 | gesamt |
|---|---|---|---|---|---|
| P0 | m | 108 | 155 | 109 | 372 |
| P1 | m | 146 | 151 | 127 | 424 |
| P2 | m | 140 | 94 | 145 | 379 |
| P3 | m | – | 126 | – | 126 |
| P4 | m | 98 | 104 | 120 | 322 |
| P5 | m | 118 | 94 | 95 | 307 |
| P6 | m | 95 | 86 | 101 | 282 |
| P7 | m | 113 | 57 | 104 | 274 |
| P8 | w | 108 | 84 | 123 | 315 |
| P9 | m | 120 | 96 | 93 | 309 |
| gesamt | – | 1046 | 1047 | 1017 | 3110 |

Tabelle 1: Zusammensetzung (Anzahl Bilder) des Datensatzes FINCA-PH.

Abbildung 38: Beispiele aus dem Datensatz FINCA-HF. Obere Reihe: Weißpunkt-Einstellung *indoor*. Untere Reihe: *outdoor*. Die dargestellten Beleuchtungssituationen sind von links nach rechts A, B, C, D.

### 7.1.2 *FINCA Hautfarbdatensatz (FINCA-HF)*

Dieser Datensatz enthält 357 innerhalb der FINCA aufgenommene Einzelbilder in einer Auflösung von 384x288 Pixeln, auf denen jeweils eine oder mehrere von 13 verschiedenen Personen zu sehen sind. Zusätzlich stehen 171 Bilder ohne Personen für das Training eines Hintergrundmodelles zur Verfügung. Dabei sind sehr unterschiedliche Blickwinkel aller im Raum vorhandenen Kameras enthalten, so dass nahezu alle Bereiche des Raumes als Hintergrund auftauchen (Abbildung 38). Die Bilder wurden an verschiedenen Tagen und Tageszeiten aufgenommen. Weiterhin umfassen die Daten vier verschiedene Beleuchtungsbedingungen:

**A:** Rolläden geöffnet, Deckenlichter eingeschaltet

**B:** Rolläden geöffnet, Deckenlichter ausgeschaltet

**C:** Rolläden geschlossen, Deckenlichter aus, Tafelstrahler auf 70%

**D:** Rolläden geschlossen, Deckenlichter auf 50%

Zusätzlich bieten die verwendeten Sony-Kameras zwei fest eingestellte Weißpunkte („indoor“ und „outdoor“), die sich in ihrer Farbdarstellung erheblich unterscheiden. Alle Beleuchtungskonfigurationen wurden mit beiden Einstellungen aufgenommen. Der Datensatz deckt somit eine große Bandbreite von Umgebungsbedingungen ab (vgl. Abbildung 38).

| Beleuchtung | A | P | B | P | C | P | D | P | gesamt |
|---|---|---|---|---|---|---|---|---|---|
| Vordergrund *indoor* | 34 | 0,1 | 39 | 1,2 | 23 | 2,3 | 49 | 2,4,5 | 145 |
| Vordergrund *outdoor* | 45 | 2,6 | 42 | 2,7 | 38 | 2,6 | 47 | 2,3,4,6 | 172 |
| Hintergrund *indoor* | 23 | – | 20 | – | 23 | – | 23 | – | 89 |
| Hintergrund *outdoor* | 22 | – | 21 | – | 19 | – | 20 | – | 82 |

Tabelle 2: Zusammensetzung (Anzahl Bilder) des Datensatzes FINCA-HF (Training). Die Spalten „P“geben jeweils die IDs der enthaltenen Personen an (nicht identisch mit den IDs aus Tabelle 1). Die Personen mit den IDs 8-12 bilden den Validierungsdatensatz.

Die Aufteilung in Trainings- und Validierungsdaten ist für alle im weiteren Verlauf präsentierten Experimente identisch. Tabelle 2 zeigt die Zusammensetzung der Trainingsdaten. Diese enthalten acht Personen. Die restlichen fünf Personen bilden den Validierungsdatensatz, der insgesamt 40 Bilder (16 *outdoor*, 24 *indoor*) umfasst. Die Validierungsdaten enthalten alle Beleuchtungsbedingungen, sind aber nicht entsprechend annotiert, so dass die Angabe einer genauen Aufteilung nicht möglich ist. In allen Bildern wurden sichtbare Hautpartien von Hand annotiert. Die Gesamtgröße der Hautfarb-Stichprobe beträgt 211698 (*indoor*) bzw. 251893 (*outdoor*) Pixel.

### 7.1.3 *FINCA Multikamera-Zeigeexperiment (FINCA-MZ)*

Dieser Datensatz diente zur Evaluierung einer frühen Version des vorgestellten Gestenerkennungssystems [163], bei der insbesondere die eigentliche Gestendetektion und -klassifikation auf HMM-Basis fehlte. Ziel war die Erkennung von Objektreferenzen anhand von Zeigegesten. Hierbei befanden sich sechs Markierungen im Abstand von ca. 40 cm auf einer Tischplatte. Sechs verschiedene Personen wurden gebeten, von zwei vordefinierten Raumpositionen aus in einer festen Reihenfolge mit ausgestrecktem Arm auf diese Markierungen zu zeigen. Dabei wurden mit je zwei Kameras unsynchronisierte Farbbildsequenzen in halber PAL-Auflösung (378×278 Pixel) aufgezeichnet (Abbildung 39). Der Datensatz enthält 5332 Bilder, die genaue Aufteilung auf die beteiligten Personen ist in Tabelle 3 gegeben.

Abbildung 39: Beispiele aus dem Datensatz FINCA-MZ. Zusammengehörige Bildpaare stehen jeweils übereinander.

### 7.1.4 *FINCA Gestendatensatz (FINCA-G)*

Ein weiterer innerhalb der FINCA aufgenommener Evaluierungsdatensatz umfasst Instanzen von neun verschiedenen emblematischen Armgesten. Diese wurden so ausgewählt, dass sie entweder Gesten entsprechen, die in normaler zwischenmenschlicher Kommunikation vorkommen, oder dass ihre Bedeutung intuitiv verständlich ist. Die Gesten sind im Einzelnen (vgl. Abbildung 40):

**Horizontales Winken:** Horizontale Winkbewegung um das Ellbogengelenk. Die Hand ist dabei geöffnet und befindet sich seitlich vom Kopf zwischen Kopf- und Schulterhöhe.

**Vertikales Winken:** Vertikale Winkbewegung mit geöffneter Hand parallel zum Körper.

**Aufwärts:** Anheben des Unterarms bis ungefähr auf Schulterhöhe. Zum Ende der Bewegung zeigt die geöffnete Handfläche nach oben. Danach wird die Handfläche durch Strecken der Schulter- und Ellbogengelenke aufwärts gedrückt.

**Abwärts:** Anheben des Unterarmes bis auf Schulterhöhe, wobei die geöffnete Hand nach unten gerichtet ist, gefolgt von einer vertikalen Abwärtsbewegung der Hand durch Strecken des Armes bis ungefähr auf Hüfthöhe.

**Kreis:** Wiederholte kreisende Bewegung mit geöffneter Handfläche auf Schulterhöhe seitlich des Kopfes.

| Person | m/w | Kamera 1 | Kamera 2 | Kamera 3 | Kamera 4 | gesamt |
|---|---|---|---|---|---|---|
| P0 | m | 277 | 262 | 240 | 271 | 1050 |
| P1 | m | 210 | 208 | 198 | 271 | 887 |
| P2 | m | 354 | 280 | 352 | 410 | 1396 |
| P3 | w | – | – | 246 | 311 | 557 |
| P4 | m | 182 | 160 | 172 | 212 | 726 |
| P5 | m | 147 | 131 | 189 | 249 | 716 |
| gesamt | – | 1170 | 1041 | 1397 | 1724 | 5332 |

Tabelle 3: Zusammensetzung (Anzahl Bilder) des Datensatzes FINCA-MZ. Die Kameras 1 und 2 bzw. 3 und 4 bildeten jeweils ein für die 3D-Rekonstruktion verwendetes Paar.

**Herkommen:** Ausstrecken des Armes in Richtung des Adressaten, danach wird die Hand in einer ausholenden bogenförmigen Bewegung in Richtung Schulter bewegt. Ggf. mehrere Wiederholungen.

**Weggehen:** Ausgehend von einer Ruheposition mit der Hand auf Hüfthöhe wird eine weit ausholende, schnelle bogenförmige Bewegung vorwärts und aufwärts in Richtung des Adressaten ausgeführt.

**Stop:** Aus der Ruheposition wird die Hand bis ungefähr auf Kopfhöhe gehoben, die geöffnete Handfläche zeigt nach vorne. Die Hand wird für kurze Zeit in dieser Position gehalten.

**Zeigen:** Eine Zeigegeste mit ausgestrecktem Arm und Zeigefinger, wie sie gewöhnlicherweise benutzt wird, um auf entfernte Objekte zu deuten. Diese Geste nimmt aus zwei Gründen eine Sonderstellung ein. Erstens handelt es sich bei ihr nicht um eine rein emblematische Geste, weil sie eine Referenz in die Umgebung definiert und nur unter Betrachtung des Umgebungskontextes interpretiert werden kann. Zweitens unterliegt die relative Lage und Orientierung relativ zum Körper wesentlich schwächeren Einschränkungen als bei den anderen Gesten: Es wurden keine Ziele definiert, die angezeigt werden mussten, sondern die Testpersonen konnten ihre Zeigeziele vollkommen frei wählen. Deshalb weisen die Trajektorien der Zeigegeste sehr große Variationen hinsichtlich ihrer globalen Orientierung im Raum auf.

Abbildung 40: Bildausschnitte aus dem Datensatz FINCA-G mit eingezeichneten Handtrajektorien. Von links nach rechts: Kreis (K), Herkommen (Her), Abwärts (Ab), Weggehen (Weg), Zeigen (Z), Stop (S), Aufwärts (Auf), Horizontales Winken (HW), Vertikales Winken (VW).

Der Datensatz enthält somit sowohl kurze, schnelle Bewegungen als auch kompliziertere repetitive Bewegungen. Einige Gesten (z.B. Herkommen, Weggehen) können sowohl einmalig als auch repetitiv sein.

Die Gesteninstanzen sind kurze Sequenzen aus Farbbildern mit einer Auflösung von 378×278 Pixeln. Sie wurden mit je zwei unsynchronisierten Kameras aufgenommen, deren relative Positionen immer gleich blieben. Die Bildrate beträgt nominell 20 Hz, unterliegt dabei jedoch starken Variationen, wie sie auch im realen Betrieb aufgrund unterschiedlicher Verarbeitungszeiten zu erwarten sind. In den meisten Fällen ist die tatsächliche Aufzeichnungsbildrate deutlich geringer und variiert auch während einer Sequenz. Die Ausrichtung der Kameras im Raum wurde mehrfach variiert.

Der Datensatz enthält 17 verschiedene Personen. Jede von ihnen führt mehrere Instanzen jeder beschriebenen Geste sowohl mit dem rechten als auch dem linken Arm aus. Den Testpersonen wurden die einzelnen Gesten einmal vorgeführt. Die Vorführung wurde auf Anfrage ggf. wiederholt. Es wurden keine präzisen Anweisungen hinsichtlich Ausführungsgeschwindigkeit, absoluten oder relativen Positionen oder sonstigen Ausprägungen der Gestentrajektorien gegeben. Die Testpersonen konnten sich frei im Sichtfeld der Kameras bewegen und auch ihre Orientierung relativ zu den Kameras frei wählen. Demnach ist der Datensatz schwierig und realistisch, weil die Gesteninstanzen unter verschiedenen Blickwinkeln und absoluten Raumpositionen beobachtet wurden und erhebliche Variationen in Erscheinungsbild und Ausführungsgeschwindigkeit aufweisen. Insgesamt umfasst dieser Datensatz 51217 Bilder und 800 Gesteninstanzen. Die genaue Zusammensetzung, aufgeschlüsselt nach Personen und Gestenart, ist in Tabelle 4 zu sehen.

Um eine Bewertung der Qualität der Trajektorienextraktion und -klassifikation zu ermöglichen, wurden die Positionen von Köpfen und Händen in den Bildern halbautomatisch annotiert. Zunächst wurden die beschriebenen Detektionsalgorithmen angewendet. Deren Hypothesen wurden anschließend manuell inspiziert und ggf.

| Geste | Zeigen | H.Winken | V.Winken | Aufw. | Abw. | Herk. | Wegg. | Kreis | Stop |
|---|---|---|---|---|---|---|---|---|---|
| #Instanzen | 86 | 89 | 79 | 88 | 92 | 92 | 96 | 95 | 83 |
| r/l | 44/42 | 43/46 | 40/39 | 44/44 | 46/46 | 46/46 | 47/49 | 46/49 | 41/42 |
| Person | | | | | | | | | |
| P0 (m) | 524/4/3 | 547/2/3 | 691/3/3 | 334/4/4 | 363/4/4 | 283/4/5 | 185/3/4 | 461/5/5 | 343/3/3 |
| P1 (m) | 190/2/2 | 309/2/2 | 405/2/2 | 217/2/2 | 300/2/2 | 205/2/2 | 171/3/3 | 301/2/2 | 147/2/2 |
| P2 (w) | 235/3/3 | 384/2/3 | 399/2/2 | 182/2/3 | 318/3/3 | 363/3/3 | 220/4/3 | 363/2/2 | 116/2/2 |
| P3 (m) | 228/2/2 | 370/2/2 | 425/2/2 | 252/3/3 | 335/3/3 | 306/2/2 | 204/3/3 | 418/2/2 | 178/2/2 |
| P4 (m) | 159/2/2 | 306/2/1 | 364/2/2 | 90/2/1 | 144/2/2 | 134/2/1 | 163/2/3 | 319/2/2 | 138/2/2 |
| P5 (m) | 199/2/2 | 239/2/2 | 288/2/2 | 166/2/2 | 178/2/2 | 205/2/2 | 128/2/2 | 254/2/3 | 181/2/2 |
| P6 (m) | 200/2/2 | 248/2/2 | 287/2/2 | 93/2/2 | 160/2/2 | 143/4/2 | 109/2/2 | 338/4/4 | 172/2/2 |
| P7 (m) | 426/2/2 | 342/2/3 | 535/2/2 | 284/2/2 | 248/2/2 | 234/2/2 | 215/2/2 | 420/2/3 | 235/2/2 |
| P8 (m) | 206/2/2 | 341/2/2 | 399/2/2 | 133/2/2 | 155/2/2 | 102/2/2 | 106/2/2 | 339/2/2 | 120/2/2 |
| P9 (m) | 243/3/2 | 299/3/3 | 452/2/3 | 316/3/3 | 296/3/2 | 201/3/3 | 216/3/3 | 411/2/3 | 213/2/3 |
| P10 (m) | 411/3/3 | 577/4/3 | 442/3/2 | 261/3/3 | 305/3/3 | 258/3/3 | 249/3/3 | 390/4/3 | 296/3/3 |
| P11 (m) | 472/3/3 | 521/3/3 | 605/2/2 | 309/3/3 | 365/3/3 | 265/3/3 | 327/3/3 | 717/3/3 | 362/3/3 |
| P12 (m) | 345/3/3 | 513/3/3 | 660/2/2 | 329/3/3 | 445/4/3 | 299/3/3 | 279/4/3 | 508/3/3 | 333/3/3 |
| P13 (w) | 337/2/2 | 729/3/2 | 588/2/2 | 254/2/2 | 286/2/2 | 157/2/2 | 163/2/2 | 503/2/2 | 362/2/2 |
| P14 (m) | 288/3/3 | 854/4/6 | 626/3/3 | 358/3/3 | 429/3/4 | 297/3/4 | 293/3/4 | 474/3/3 | 32373/3 |
| P15 (m) | 328/3/3 | 563/2/3 | 603/3/2 | 356/3/3 | 455/3/4 | 456/3/4 | 235/3/3 | 641/3/4 | 324/3/3 |
| P16 (m) | 639/3/3 | 800/3/3 | 734/4/4 | 549/3/3 | 403/3/3 | 505/3/3 | 304/3/4 | 612/3/3 | 382/3/3 |
| #Bilder | 5430 | 7942 | 8503 | 4483 | 5185 | 4413 | 3567 | 7469 | 4225 |

Tabelle 4: Übersicht über die Aufteilung des Datensatzes FINCA-G. Die oberen beiden Zeilen geben die Gesamtanzahl der Instanzen pro Geste und aufgeschlüsselt nach links- und rechtshändig an. Die folgende Tabelle schlüsselt die Zusammensetzung nach Personen auf. Die Einträge haben die Form „#Bilder/Instanzen rechtshändig/Instanzen linkshändig“. Die Anzahl der Bilder ist jeweils die Summe über alle Gesteninstanzen und beide Kameras.

korrigiert. Insbesondere wurden fehlende Detektionen ergänzt und Fehldetektionen entfernt. Detektionshypothesen wurden nur dann korrigiert, wenn ihre Abweichung von der korrekten Position sehr groß war, ansonsten wurden sie beibehalten. Damit wird durch die Annotation eine gute, aber keineswegs perfekte Lokalisierung der relevanten Körperteile vorgegeben, die kleinere Lokalisierungsfehler und Größenvariationen enthält. Weiterhin wurden die Gesteninstanzen dergestalt manuell segmentiert, dass sie erhebliche Variationen in Start- und Endpunkt sowie – bei repetitiven Gesten – Anzahl der Wiederholungen aufweisen.

### 7.1.5 *HumanEVA-I Motion Capture Daten*

Der *HumanEVA-I* Datensatz (siehe [182] für eine ausführliche Beschreibung) ist ursprünglich für die Evaluierung von Ansätzen zur Posenschätzung und zum Posentracking vorgesehen. Er besteht aus synchronisierten Videosequenzen (25Hz) von insgesamt sieben kalibrierten Kameras und zugehörigen Posenbeschreibungen in Form eines ikonischen Körpermodelles, die mit einem kommerziellen *Motion Capture* (MC) System mit hoher Datenrate (60 bzw. 120 Hz) erfasst wurden. Somit stehen hochaufgelöste, annotierte Trajektorien für verschiedene Körperpunkte zur Verfügung. Vier verschiedene Personen führen jeweils sechs verschiedene Aktionen (Laufen, Joggen, Boxen, Werfen/Fangen, Gestikulieren, Kombination aus allen vorherigen Aktionen) in mehreren Wiederholungen aus. Die Daten umfassen zusammenhängende lange Sequenzen (bis zu einer Minute). Aktionsinstanzen sind nicht segmentiert.

Aufgrund ihrer Beschaffenheit sind die enthaltenen Aktionen jedoch nicht für eine Evaluation des Gestenerkennungssystems geeignet. Die einzige Aktion, die Ähnlichkeit mit emblematischen Gesten aufweist, ist „Gestikulieren“: Hierbei führen die Testpersonen Bewegungen aus, die an Winken oder die weiter oben beschriebene „Herkommen“ Geste erinnern, jedoch in sehr undefinierter Weise. Allerdings entstammen die 3D-Punkttrajektorien, die der Datensatz zur Verfügung stellt, realistischen menschlichen Bewegungen und sind deshalb für das Training von Trajektorien- und Rückweisungsmodellen in der Klassifikation geeignet.

## 7.2 IMPLEMENTIERUNG UND HARDWARE

Die komplette vorgestellte 2D-Verarbeitungspipeline bis einschließlich der Handdetektion wurde in C++ umgesetzt. Die Implementierung ist *multithreaded*, ansonsten jedoch nicht speziell optimiert. Insbesondere wird keine Grafik- oder SIMD-Hardware zur Beschleunigung verwendet. Alle im Folgenden angegebenen Meßwerte zur Verar-

beitungsgeschwindigkeit wurden auf gleichartigen Rechnern (Intel Core2 Duo E8500 3,17 GHz, 2 GB RAM) ermittelt. Jedem der Experimente – mit Ausnahme der offline evaluierten HOG-Parameteroptimierung, Hautfarbendetektion und Trajektorienklassifikation – liegt eine initial identische Implementierung der Pipeline zugrunde, aus der für das jeweilige Experiment nicht benötigte Verarbeitungsschritte entfernt wurden. Somit sind die angegebenen Werte der Laufzeitmessungen vergleichbar. Jedoch muß darauf hingewiesen werden, dass sie Schwankungen unterliegen können, weil während der Laufzeit der Experimente kein exklusiver Zugriff auf die betroffenen Rechner bestand. Alle angegebenen Laufzeiten wurden ermittelt als Differenz der Systemzeitstempel, gemessen unmittelbar vor der Übergabe eines Bildes an die Pipeline bzw. unmittelbar nach Bereitstellung der Ergebnisse. Sie verstehen sich also als reine Bearbeitungszeit ohne Ein- und Ausgabeoperationen.

Die Trajektorienaggregation, 3D-Projektion, Normalisierung, Trajektoriennachbearbeitung und Merkmalsextraktion sind größtenteils als MATLAB-Code realisiert. Deshalb wird auf Laufzeitangaben verzichtet, weil die Messwerte nicht aussagekräftig für eine reale Implementierung wären. Alle Verarbeitungsschritte laufen jedoch üblicherweise schneller als Echtzeit. Das verwendete HMM-Klassifikationsframework *ESMERALDA* [58] ist in C geschrieben und hat sich als sehr effizient erwiesen.

## 7.3 PERSONENDETEKTION

Die Evaluierung der Personendetektion umfasst die Optimierung der Parameter des Kopf-Schulter-Detektors sowie die Bewertung des Einflusses der Suchraumeinschränkung mittels Bewegungsdetektion auf die Detektionsleistung und das Laufzeitverhalten. Hierfür kommen die Datensätze FINCA-PH, FINCA-MZ und FINCA-G zum Einsatz. Die Evaluierung erfolgt mittels dreifacher Kreuzvalidierung. Zu diesem Zweck werden die Bilddaten datensatzübergreifend in vier Teile eingeteilt (D1 bis D4). Diese Aufteilung geschieht anhand kompletter Personendatensätze, so dass die Kreuzvalidierungssätze disjunkt sind und jeweils unterschiedliche Personen beinhalten[1]. Das Training wird nun auf jeweils zweien der Kreuzvalidierungssätze durchgeführt, die Evaluation auf einem Dritten, und die abschließende Leistungsbewertung auf dem verbleibenden Vierten. Dies wird drei Mal mit unterschiedlicher Datenzusammensetzung wiederholt (KVal1 bis KVal3). Die Ergebnisse werden dann über alle Kreuzvalidierungsläufe gemittelt. Die genaue Aufteilung ist in Tabelle 5 gegeben.

1 Sinn dieser Maßnahme ist, dass die Ergebnisse somit als personenunabhängig interpretiert werden können.

| | Training | #Bilder | Validierung | #Bilder | Test | #Bilder |
|---|---|---|---|---|---|---|
| KVal1 | D1+D2 | 31433 | D4 | 15355 | D3 | 13662 |
| KVal2 | D2+D3 | 26422 | D1 | 18673 | D4 | 15355 |
| KVal3 | D3+D4 | 29017 | D2 | 12760 | D1 | 18673 |

Tabelle 5: Zusammensetzung der Kreuzvalidierungsdaten für die Parameteroptimierung des Personendetektors.

### 7.3.1 *Parameteroptimierung des Detektors*

Der HOG-basierte Personendetektor hat eine Vielzahl von Parametern, die seine Leistung erheblich beeinflussen. Ziel dieses Experimentes ist es, den Parameterraum innerhalb sinnvoller Grenzen nach „guten“ Parametern zu durchsuchen. In erster Linie ist der Aufbau des HOG-Deskriptors entscheidend für die Detektionsleistung. Hier können die Anzahl der Zellen pro Block, die Anzahl der HOG-Blöcke, die gegenseitige Überlappung der Blöcke, die Anzahl der Orientierungsbins in den Gradientenhistogrammen sowie die Art des verwendeten Kantenextraktionsalgorithmus variiert werden. Sinnvolle Grenzen für einige dieser Parameter ergeben sich aus folgenden Überlegungen:

- Der Deskriptor benötigt einerseits eine gewisse Mindestgröße (im Sinne der Anzahl seiner Zellen), damit die Vorteile dieser Modellierungsart zum Tragen kommen.
- Andererseits verliert ein zu großer bzw. zu komplexer Deskriptor seine Flexibilität und führt zu einer aufwändigeren Klassifikation, mithin also zu einer deutlichen Verringerung der Effizienz.
- Eine sehr große Anzahl von Zellen führt dazu, dass auf jede Zelle nur noch sehr wenige Pixel entfallen, was aus statistischen Gründen ungünstig ist.
- Die maximale sinnvolle Anzahl von Orientierungsbins ist klein, weil die verwendeten Kantenoperatoren aufgrund der kleinen Pixelnachbarschaften nur eine sehr begrenzte Winkelauflösung aufweisen.

Weiterhin hat die Topologie des verwendeten MLP-Klassifikators entscheidenden Einfluß auf die Klassifikationsleistung und Effizienz: Große Netzwerke führen zu einem potentiell mächtigeren Klassifikator, allerdings auch zu einer größeren Anzahl

freier Parameter und erhöhtem Rechenaufwand. Die Anzahl der Eingabeneuronen ist hierbei durch den Aufbau des HOG-Deskriptors vorgegeben. Als Ausgabekodierung bietet sich bei einem binären Klassifikationsproblem ein einzelnes Ausgabeneuron mit binärer Zielaktivierung an. Die weitere Topologie, also Anzahl und Größe der versteckten Schichten, kann beliebig gewählt werden. Eine Grenze ist jedoch aus statistischer Sicht durch die Anzahl verfügbarer Trainingsbeispiele gegeben: Für eine robuste Parameterschätzung werden pro freiem Parameter mehrere Trainingsbeispiele benötigt. Die Anzahl der Netzwerkgewichte sollte also nicht größer gewählt werden als die Größe der Trainingsstichprobe.

Die Vielzahl der möglichen Parameter macht eine erschöpfende experimentelle Untersuchung unmöglich. Die folgende Aufzählung gibt einen Überblick über die Parameterwerte, die gemäß den obigen Überlegungen und Erfahrungen mit vorherigen informellen Experimenten gewählt wurden. Für die Netzwerktopologien wurden drei- und vierschichtige Netze untersucht, deren Neuronenzahl entweder fest oder relativ zur Größe der jeweils vorherigen Schicht gewählt wurde.

- Gradientenberechnung: Einfacher $3 \times 1$-Operator (symmetrische Differenz) und *Sobel*-Operator. Auf die Verwendung des *Canny*-Algorithmus wurde aus Effizienzgründen verzichtet. Die Gradientenrichtungen werden im Wertebereich $[0, \pi[$ dargestellt.
- Anzahl Orientierungsbins: Vier (45° Schritte) oder acht (22.5° Schritte).
- Anzahl Deskriptorblöcke: Horizontal/vertikal 2/2, 3/2 und 3/3.
- Anzahl Zellen pro Block: Horizontal/vertikal 1/1, 2/1 und 2/2.
- Überlappung: Die Überlappung erfolgt immer derart, dass Blöcke sich um Vielfaches der Zellgröße überlappen. Die einzigen möglichen Einstellungen sind demzufolge Null (eine Zelle pro Block) und eins (zwei Zellen pro Block).
- Anzahl Neuronen in versteckten Schichten: 64/16, 32/8, 64, 32, $0.25 \cdot |\mathcal{N}^{k-1}|$, $\sqrt{|\mathcal{N}^{k-1}|}$, $2 \cdot \sqrt{|\mathcal{N}^{k-1}|}$, mit der Größe der vorherigen Netzwerkschicht $|\mathcal{N}^{k-1}|$.

Die Netzwerke wurden mit dem RPROP-Algorithmus (Resilient Backpropagation [165]) trainiert. Das Training wurde abgebrochen, wenn der Klassifikationsfehler über den Validierungsdaten ein stabiles Minimum erreichte. Als Aktivierungsfunktionen wurden Sigmoidfunktionen gewählt. Für Training und Modellierung kam die frei verfügbare *Fast Artificial Neural Network Library* [136] zum Einsatz.

Als Bewertungsmaß wird die *Equal Error Rate* (EER) der *Receiver Operating Characteristic* (ROC) Kurven (s. z.B. [52]) wie folgt ermittelt: Für einen gegebenen Entscheidungsschwellwert $\delta$ lässt sich durch einfaches Abzählen anhand der Annotationen der Evaluierungsdaten der Anteil korrekter Detektionen $\gamma_K$ (*Erkennungsrate*) und Fehldetektionen $\gamma_F$ (*Fehldetektionsrate*) ermitteln:

$$\gamma_K = \frac{|\Omega_D \cap \Omega_A|}{|\Omega_A|}, \quad \gamma_F = \frac{|\Omega_D \backslash \Omega_A|}{|\Omega|} \tag{7.1}$$

Hierbei ist $\Omega_D$ die Menge aller Detektorhypothesen, deren Ausgabeaktivierung größer als $\delta$ ist, $\Omega_A$ ist die Menge der annotierten Personen und $\Omega$ ist die Menge aller betrachteten Detektorfenster. Durch Variation des Wertes von $\delta$ ergibt sich die ROC-Kurve (vgl. Abbildung 42). Die EER ist definiert als der Wert von $\gamma_K$ an demjenigen Punkt der Kurve, für den $1 - \gamma_K = \gamma_F$ gilt. Dieses Maß ist deshalb zur Qualitätsbeurteilung eines Klassifikators geeignet, weil die Detektionsleistung umso besser ist, je größer im Diagramm die Fläche unter der Kurve ist. Die EER ist der Schnittpunkt der Kurve mit der Diagonalen $\gamma_K = 1 - \gamma_F$, demzufolge verläuft die Kurve (im Mittel) umso günstiger, je näher dieser Schnittpunkt am Punkt $(0, 1)$ liegt. Die EER hat jedoch keine Aussagekraft bezüglich der absolut besten erreichbaren Klassifikationsleistung.

Für die Berechnung der Erkennungsrate wurden die verwendeten Bilddaten von Hand annotiert (Abbildung 41). Die Annotation ist um die Schulterregion zentriert und besteht aus zwei Rechtecken mit gleichem Seitenverhältnis und gleichem Mittelpunkt. Das äußere Rechteck ist dreimal so groß wie das innere. Eine Detektorhypothese wird genau dann als korrekt angesehen, wenn sie komplett zwischen diesen beiden Rechtecken liegt, ansonsten wird sie als Fehldetektion gezählt. Eine Annotation wurde korrekt detektiert, wenn für sie mindestens eine korrekte Detektorhypothese existiert. Mehrfachdetektionen werden nur einmal gezählt, aber auch nicht als Fehler angesehen.

Die Gesamtanzahl der untersuchten Netzwerk-Konfigurationen beträgt ca. 500, so dass hier nicht alle Ergebnisse wiedergegeben werden können. Viele dieser Konfigurationen liefern zudem sehr ähnliche Resultate (die EER der besten 35 Netzwerke unterscheiden sich erst in der vierten oder fünften Nachkommastelle). Deshalb werden einige verschiedenartige gute (im Sinne ihrer EER) Konfigurationen exemplarisch untersucht. Die Detektoreinstellungen für die folgenden Experimente sind wie folgt: Kleinste Detektorfenstergröße $24 \times 18$, zehn Auflösungsstufen mit Skalierungsfaktor 1.3 (d.h. maximale Fenstergröße $255 \times 191$), Detektionsclustering mit einer minimal benötigten Clustergröße von zwei, Schrittweite des gleitenden Detektorfensters entspricht jeweils der halben horizontalen bzw. vertikalen Zellgröße.

Die absolut beste EER wurde mit dem Sobel-Kantenoperator und einem HOG-Deskriptor mit $3 \times 3$ Blöcken, $2 \times 2$ Zellen pro Block, 8 Histogrambins sowie keiner

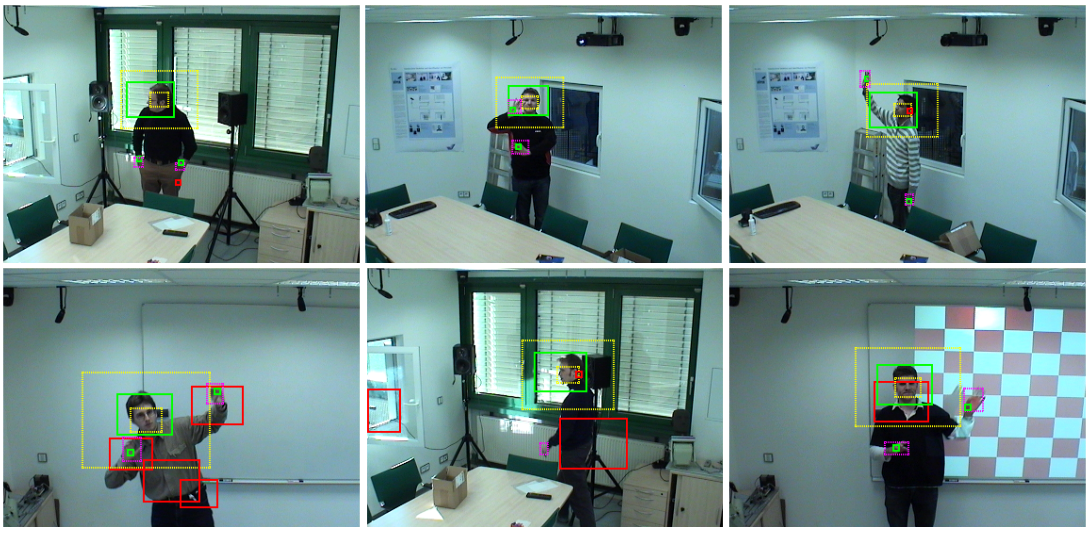

Abbildung 41: Beispiele für Annotationen der Bilddaten für die Evaluation der Kopf- und Handdetektion mit Beispielergebnissen. Gelbe gepunktete Rechtecke: Kopf-Schulter-Annotation. Violette gepunktete Rechtecke: Handannotation. Grüne Rechtecke: Akzeptierte (korrekte) Hypothesen. Rote Rechtecke: Zurückgewiesene Hypothesen (Fehldetektionen).

Überlappung zwischen den Blöcken erreicht (im Folgenden abgekürzt als B33-Z22-B8-U0) (Abbildung 42). Mit einer Netzwerktopologie von 288 Eingabeneuronen, einer versteckten Schicht mit 34 Neuronen und einem Ausgabeneuron beträgt die EER 0.9944. Gemittelt über alle untersuchten Topologien erreicht dieser Deskriptor eine EER von 0.9943 und ist auch damit der beste. Bei einem Klassifikationsschwellwert von $\delta = 0.9$ werden 95% der annotierten Personen in den Testdaten (insgesamt 47667) korrekt detektiert. Die mittlere Anzahl von Fehldetektionen pro Bild beträgt 8.9. Diese relativ hohe Anzahl an Fehldetektionen ist auf den großen untersuchten Skalierungsbereich sowie auf die eingestellte minimale Clustergröße zurückzuführen. Absolut gesehen ist der Anteil der Fehldetektionen sehr gering, weil pro Bild mehrere Tausend Detektionsfenster ausgewertet werden.

Interessanterweise erweist sich die – an sich sehr rauschanfällige – Kantenberechnung mit einfachen Pixeldifferenzen als ebenso gut, wie der robustere Sobel-Operator. Für einige Konfigurationen ist der Verlauf der ROC-Kurve im relevanten Bereich sogar deutlich günstiger (vgl. Abbildung 42, besonders auffällig für die Konfiguration B33-Z22-B8-U0). Diese Ergebnisse bestätigen die in [37] gemachten Beobachtungen, dass diese extrem einfache und effiziente Kantenberechnungsmethode ausreichend ist.

| Deskriptor | Topologie | KO | ER(%) | FD | EER | KO | ER(%) | FD | EER |
|---|---|---|---|---|---|---|---|---|---|
| B33-Z22-B8-U0 | 288-34-1 | PD | 98.2 | 8.6 | 0.9943 | Sobel | 95.0 | 8.9 | 0.9944 |
| B33-Z22-B8-U1 | 288-34-1 | PD | 94.5 | 5.5 | 0.9935 | Sobel | 96.4 | 6.8 | 0.9939 |
| B22-Z22-B4-U0 | 64-8-1 | PD | 90.0 | 11.2 | 0.9929 | Sobel | 90.6 | 11.6 | 0.9934 |
| B22-Z22-B8-U1 | 128-11-1 | PD | 90.9 | 4.7 | 0.9935 | Sobel | 94.0 | 7.4 | 0.9906 |
| B22-Z22-B8-U1 | 128-32-8-1 | PD | 94.9 | 4.7 | 0.9903 | Sobel | 92.6 | 5.5 | 0.9918 |
| B22-Z11-B8-U0 | 32-32-1 | PD | 66.2 | 3.75 | 0.9726 | | | | |

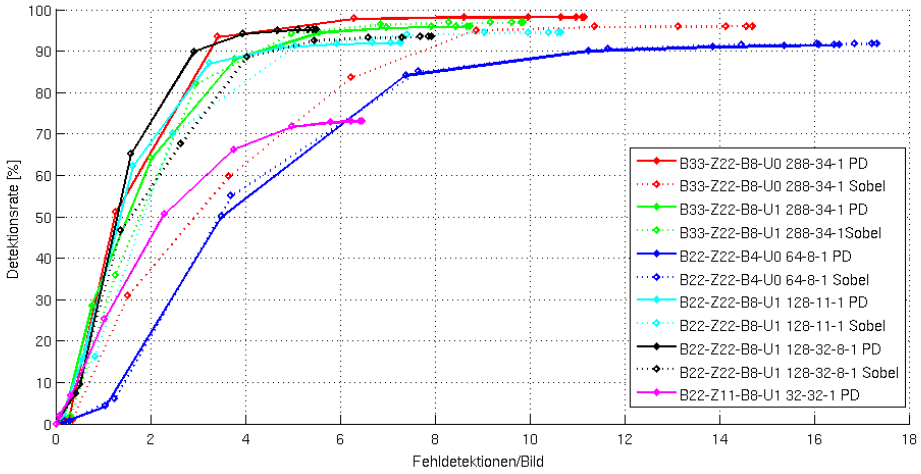


Abbildung 42: Exemplarische Ergebnisse für HOG-Personendetektoren mit verschiedenen Parametern. Oben: Absolute Zahlenwerte für einen Klassifikationsschwellert $\delta = 0.9$, gemittelt über die Testdaten der Kreuzvalidierung (KO: Kantenoperator; ER: Erkennungsrate; FD: Fehldetektionen pro Bild; EER: *Equal Error Rate.*). Die Spalte „Deskriptor“ gibt den Aufbau de HOG-Deskriptors wieder (horizontale/vertikale Blöcke, hor./ver. Zellen pro Block, Histogrammbins, Anzahl überlappender Zellen). PD steht für Pixeldifferenz, d.h. die einfachste Art der Gradientenberechnung. Unten: ROC-Kurven der obigen Versuche. Die Grafik zeigt zwecks besserer Erkennbarkeit nur den für den realen Einsatz relevanten Ausschnitt mit Schwellwerten zwischen 0.8 und 1.0.

Zum Vergleich sind zusätzlich die Ergebnisse mit einem kleineren Deskriptor (je zwei horizontale und vertikale Blöcke) und zwei unterschiedlichen Topologien (eine und zwei versteckte Netzwerkschichten) dargestellt. Während die Variante mit dem dreischichtigen Netzwerk geringfügig schlechtere Ergebnisse erzielt (vgl. ROC-Kurven

in Abbildung 42 unten), führt die Verwendung eines vierschichtigen MLP zwar zu einer geringeren maximalen Erkennungsrate, jedoch zu einem im relevanten Bereich günstigeren Verlauf der ROC-Kurve (schwarze Kurve). Zudem ist dieses Netz trotz der zusätzlichen Schicht aufgrund der kleineren Eingabeschicht deutlich kleiner (169 Neuronen und 4360 interneuronale Verbindungen statt 323/9826), was eine effizientere Auswertung zur Folge hat. In allen folgenden Experimenten wird deshalb, sofern nicht anders angegeben, dieser Klassifikator (B22-Z22-B8-U1, 128-32-8-1, PD) verwendet.

Insgesamt ist zu beobachten, dass die besten Ergebnisse mit MLP mit nur wenigen Neuronen in den versteckten Schichten erreicht werden. Das ist vorteilhaft, weil wie erwähnt ein kleineres Netz schneller ausgewertet werden kann. Eine gewisse Mindestkomplexität des HOG-Deskriptors wird jedoch benötigt: Ein Deskriptor mit nur einer Zelle pro Block führt zu im Vergleich sehr schlechten Resultaten. Exemplarisch wird dies anhand des Deskriptors B22-Z11-B8-U1 (magenta-farbene Kurve) deutlich. Genauso führt eine Verringerung der Binzahl des Histogramms (d.h. Verringerung der Winkelauflösung) zu einer Verschlechterung, erkennbar z.B. an den Resultaten des Deskriptors B22-Z22-B4-U0 mit halber Winkelauflösung. Das ist wenig überraschend, weil durch die Orientierung der Intensitätskanten im Bild die Form des betrachteten Objektes codiert wird und somit wichtige Informationen verloren gehen.

### 7.3.2 *Auswirkung der Hintergrund-Modellierung*

Die Evaluation der Hintergrundmodellierung als alleinstehendes Modul ist wenig sinnvoll. Eine pixelgenaue Annotation aller Vordergrundregionen in den Daten wäre sehr aufwändig und die Ergebnisse hätten für den praktischen Einsatz keine Relevanz, weil die Extraktion von Vordergrund-Regionen lediglich der Suchraumeinschränkung dient. Deshalb ist eine genaue Vordergrundmarkierung gar nicht notwenig. Eine Annotation mit umschließenden Rechtecken würde wiederum nur auf subjektiven Vorstellungen beruhen und Konfigurationen bevorzugen, die für das menschliche Empfinden „schöne“ Regionen als Ergebnis liefern. Diese müssen aber nicht notwendigerweise die beste Lösung für das Gesamtsystem sein. Aus diesem Grund wird die Hintergrund-Modellierung anhand ihrer Auswirkungen auf die Erkennungsrate des Personendetektors und die Verarbeitungsgeschwindigkeit bewertet.

Zu diesem Zweck wird der Datensatz KVal1 aus Tabelle 5 verwendet, d.h. der HOG-Detektor wurde auf den Teildatensätzen 1 und 2 trainiert und das Training mit Teildatensatz 4 validiert. Die Ergebnisse für die Hintergrundmodellierung wurden dann auf Teildatensatz 3 ermittelt. Dabei ist erwähnenswert, dass der Datensatz Bilder aus verschiedenen Szenarien enthält. D.h. es treten mehrmalig abrupte Szenen-

und Beleuchtungswechsel auf, bei denen das Hintergrundmodell versagen muss. Diese Stellen sind in keiner Weise gekennzeichnet, alle Ergebnisse beruhen auf einer ununterbrochenen sequentiellen Verarbeitung aller Bilder. Das Modell muss sich daher an diese abrupten Änderungen adaptieren und evtl. während der Adaptionsphase auftretende Fehler sind in den angegebenen Resultaten enthalten. Die Erkennungsrate und die Anzahl von Fehldetektionen werden auf die gleiche Weise wie in Kapitel 7.3.1 berechnet. Zusätzlich wird als Maß für die Verarbeitungsgeschwindigkeit die mittlere Anzahl verarbeiteter Bilder pro Sekunde (BPS) angegeben.

Die Parameter für das gleitende Detektorfenster wurden für dieses Experiment praxisnäher gewählt: In einem bekannten Szenario kann abgeschätzt werden, welche minimale und maximale Objektgröße auftreten kann. Deshalb kann der Suchraum für die Fenstergröße kleiner gewählt werden. Im vorliegenden Fall wurde eine minimale Fenstergröße von $40\times30$ Pixeln gewählt. Es werden vier Auflösungsstufen mit einem Skalierungsfaktor von 1.25 durchlaufen, die größtmögliche Fenstergröße beträgt also $78\times58$ Pixel. Die weiteren Einstellung wurden wie im vorherigen Experiment belassen, d.h. die Schrittweite des gleitenden Fensters entspricht wieder jeweils der halben Zellgröße des HOG-Deskriptors und Detektionscluster mit weniger als zwei Hypothesen werden verworfen.

Für die Hintergrundmodellierung wurde die in Kapitel 6 vorgestellte adaptive Hintergrundsubtraktion mit Verfallsfunktion gewählt. Die während dieser Arbeit gesammelten Erfahrungen haben gezeigt, dass die Ergebnisse, die mit dieser sehr einfachen und somit effizienten Herangehensweise erreicht werden können, absolut ausreichend sind. Der größte Einfluß auf das Ergebnis ist von den Parametern der Verfallsfunktion $\alpha_1, \alpha_2$ und der Lernrate $\lambda$ zu erwarten, weil sie das „Nachklingen" der Vordergrundregionen und die Adaptivität an Veränderungen bestimmen. Deshalb werden diese im Folgenden ausführlich evaluiert.

Der Parameter $\alpha_1$ der untersuchten Verfallsfunktionen (vgl. Kapitel 6.3.1) wurde so gewählt, dass der Funktionswert jeweils nach 2, 5, 10 und 15 Zeitschritten auf einen Wert kleiner oder gleich als 0.1 abfällt. Bei der Rampenfunktion wurde der Parameter $\alpha_2$ so gesetzt, dass nach jeweils der Hälfte der Zeitschritte (aufgerundet) der Übergang vom konstanten Teil zur Rampe erfolgt. Die Sigmoidfunktion wurde über $\alpha_2$ so eingestellt, dass sie nach der Hälfte der Zeitschritte ihren Wendepunkt erreicht. Weil diese Funktion symmetrisch ist, weist sie zum Zeitpunkt 0 nicht den Wert 1 auf. (Abbildung 43).

Für die nachfolgenden Experimente wurden Graustufenbilder und ein Schwellwert von $\delta = 10$ verwendet. Die Parameter des Quadtree-Regionenextraktionsalgorithmus wurden für alle Experimente konstant belassen ($s_{min} = 4, s_{max} = 32, \delta_m = 80$). Als Vergleichswert dienen die Ergebnisse eines identisch konfigurierten HOG-Detektors

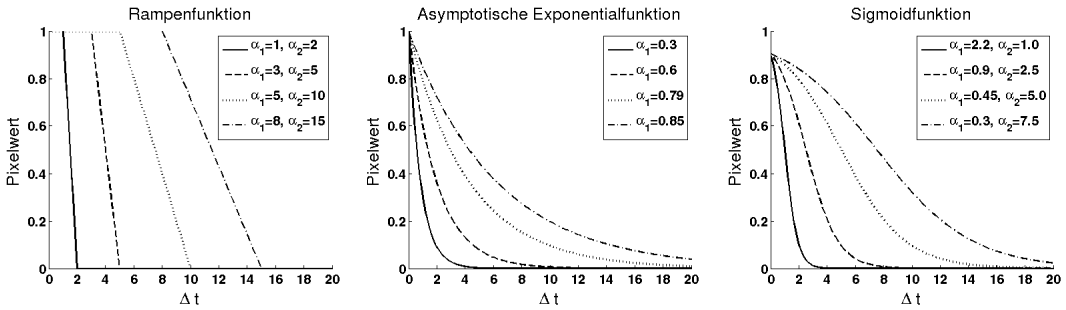


Abbildung 43: Überblick über die zur Evaluierung der Hintergrundmodellierung verwendeten Verfallsfunktionen und ihre Parameter.

bei erschöpfender Suche. Dieser erreicht eine Erkennungsrate von 98.01±0.25% (alle im Folgenden angegebenen Konfidenzintervalle beziehen sich auf ein Konfidenzniveau von 0.95) bei durchschnittlich 1.47 Fehldetektionen pro Bild und einer Verarbeitungsgeschwindigkeit von 10.9 BPS (91.69 ± 6.02 ms pro Bild). Die Erwartungen an das Hintergrundmodell sind, dass durch die Suchraumeinschränkung die Effizienz gesteigert und die Anzahl von Fehldetektionen reduziert wird, im Idealfall bei gleich bleibender Erkennungsrate. In Abbildung 44 sind die Resultate grafisch dargestellt, die vollständigen Tabellen mit allen Zahlenwerten finden sich in Anhang A.1.

Es ist klar erkennbar, dass sowohl eine Erhöhung von λ (d.h. sehr schnelle Adaption des Modelles) als auch eine Verringerung der Nachklingzeit zu einer Verschlechterung der Erkennungsrate führt. Das liegt daran, dass in beiden Fällen die Vordergrundkarte oft nur wenige zusammenhängende Regionen aufweist und entsprechend nur wenige sehr kleine Regionen extrahiert werden. Zwar sind kleine Regionen ausreichend, wenn sie so lokalisiert werden, dass sie den Bereich um den Mittelpunkt der Kopfregion enthalten. Allerdings ist das häufig nicht der Fall, weil geringe Bewegungen nur am Rand von Objekten detektiert werden können. Fehler treten hier insbesondere dann auf, wenn die Person sich wenig oder gar nicht bewegt. Sie geht dann sehr schnell in das Hintergrundmodell über und die kurze Nachklingzeit reicht nicht aus, um diese Ruhephasen zu überbrücken. Infolgedessen sinken auch die Anzahl der Fehldetektionen und die Verarbeitungszeit, weil der Suchraum für den Detektor sehr klein wird. Eine Erhöhung der Nachklingzeit führt auch bei einer sehr schnellen Adaption zu deutlich besseren Ergebnissen, hat aber nachteilige Auswirkungen auf die Verarbeitungsgeschwindigkeit, weil die extrahierten Vordergrundregionen nur langsam kleiner werden. Bei starker Bewegung im Bild kann es vorkommen, dass die Anwendung des Hintergrundmodelles sogar zu einer Verschlechterung des Laufzeitverhaltens führt, weil der Suchraum kaum eingeschränkt wird, das Hintergrundmodell aber zusätzliche

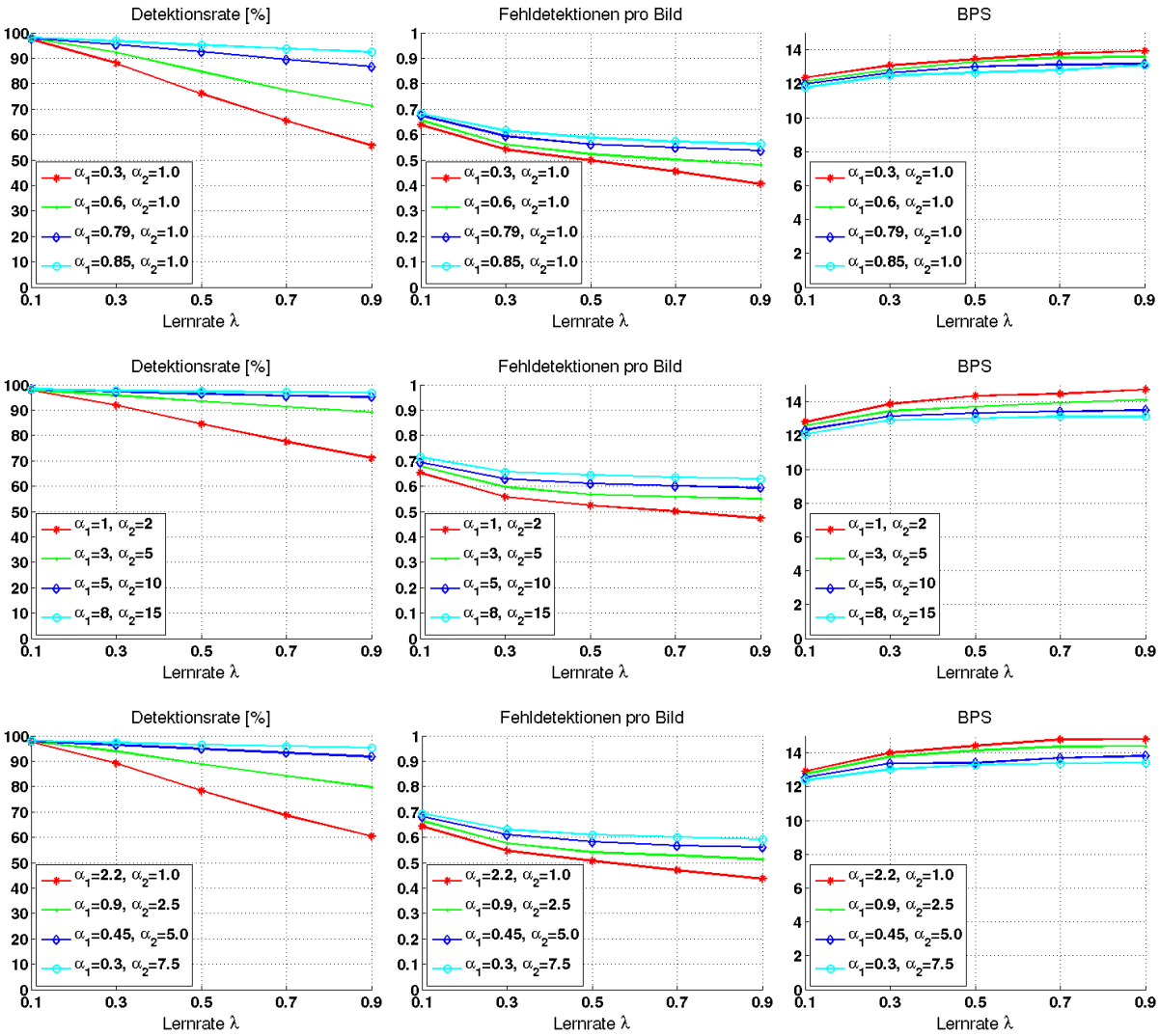


Abbildung 44: Ergebnisse der Personendetektion mit Hintergrundmodellierung in Abhängigkeit von Art und Parametrisierung der Verfallsfunktion. Von oben nach unten: Asymptotische Exponentialfunktion, Rampenfunktion, Sigmoid. Die linke Spalte zeigt die Erkennungsrate, die mittlere die durchschnittliche Anzahl von Fehldetektionen pro Bild und die Rechte die durchschnittliche Bildrate, jeweils in Abhängigkeit von den Funktionsparametern $\alpha_1, \alpha_2$ und der Lernrate $\lambda$. Die rote Kurve stellt jeweils die Parametrisierung mit der geringsten Nachklingdauer dar, die cyanfarbene diejenige mit der größten. Die genauen Zahlenwerte finden sich in Anhang A.1.

| Funktionstyp | $\lambda$ | $\alpha_1$ | $\alpha_2$ | Erk.rate [%] | FD/Bild | BPS |
|---|---|---|---|---|---|---|
| Exponential | 0.3 | 0.79 | 1.0 | 95.31±0.37 | 0.59 | 12.62 |
| Rampe | 0.3 | 3 | 5 | 95.81±0.35 | 0.60 | 13.43 |
| Sigmoid | 0.3 | 0.45 | 5.0 | 96.34±0.33 | 0.61 | 13.37 |

Tabelle 6: Gute Konfigurationen für die Verfallsfunktion der Hintergrundmodellierung.

Rechenlast generiert. In diesem Fall liefert die Kombination einer geringen Lernrate und einer Verfallsfunktion mit geringer Nachklingzeit bessere Ergebnisse.

Demzufolge ist ein möglichst guter Kompromiss wünschenswert, der einerseits eine gewisse Adaptivität des Modelles erhält, andererseits eine Stabilisierung der extrahierten Vordergrundregionen ermöglicht. Das Verhalten der verschiedenen Verfallsfunktionen ist hierbei grundsätzlich sehr ähnlich. Bei der asymptotischen Exponentialfunktion (Abbildung 44 oben) zeigt sich die größte Empfindlichkeit gegenüber Parameteränderungen. Durch ihr schnelles Abklingen lassen sich zufrieden stellende Ergebnisse nur mit geringen Lernraten und/oder großen Nachklingzeiten erzielen. Die Rampen- (Abbildung 44 Mitte) und Sigmoidfunktion (unten) sind im Vergleich weniger empfindlich. Beide Funktionstypen weisen anfangs für einen längeren Zeitraum einen hohen Funktionswert auf und fallen dann schnell gegen Null ab. Die Ergebnisse zeigen, dass diese Charakteristik für die Aufgabe besser geeignet ist.

Die absolut beste Erkennungsrate von 98.23±0.24%[2] wird für eine Rampenfunktion mit $\lambda = 0.1$, $\alpha_1 = 8$, $\alpha_2 = 15$ bei einer gleichzeitigen Fehlerrate von 0.71 Fehldetektionen pro Bild und einer Verarbeitungsgeschwindigkeit von 12.07 BPS erreicht. Für die reale Anwendung ist jedoch eher ein Parametersatz interessant, der einen guten Kompromiß aus Erkennungsrate, Fehldetektionsrate und Verarbeitungsgeschwindigkeit ergibt. Im weiteren Verlauf werden die in Tabelle 6 angegebenen exemplarischen Konfigurationen verwendet.

2 Theoretisch stellt der Vergleichswert der erschöpfenden Suche die obere Schranke für die Erkennungsrate und die Anzahl von Fehldetektionen dar. Praktisch treten bei beiden Werten gelegentlich höhere Werte auf. Der Grund dafür ist der veränderliche Startpunkt des Detektionsfensters abhängig von den extrahierten Vordergrundregionen in Verbindung mit der relativ großen Fensterschrittweite. Dies führt dazu, dass für jede Konfiguration der Hintergrundmodellierung der Detektor eine unterschiedliche Sicht auf die Daten hat.

| Funktionstyp | $\lambda$ | $\alpha_1$ | $\lambda_2$ | Erk.rate [%] | FD/Bild | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| Exponential | 0.3 | 0.79 | 1.0 | 91.56$\pm$0.48 | 0.06 | 13.09 | 76.39$\pm$9.95 |
| Rampe | 0.3 | 3 | 5 | 92.10$\pm$0.46 | 0.06 | 12.98 | 77.03$\pm$9.91 |
| Sigmoid | 0.3 | 0.45 | 5.0 | 92.83$\pm$0.44 | 0.06 | 12.77 | 78.31$\pm$10.10 |

Tabelle 7: Referenzwerte zur Einordnung der *Tracking*-Ergebnisse, ermittelt durch Auswahl der Detektorhypothese mit den meisten benachbarten Hypothesen.

### 7.3.3 *Auswirkung des Personentrackings*

In gleicher Weise wie die Hintergrundmodellierung wird der *Tracker* hinsichtlich seiner Auswirkungen auf die Ergebnisse der Personendetektion bewertet. Zu diesem Zweck wird auf dem gleichen Datensatz wie bisher mit den im vorherigen Kapitel ermittelten guten Parametersätzen für die Hintergrundmodellierung evaluiert. Dabei werden die Parameter $\beta_0$ und $\beta_m$ des *Trackers* variiert. Der Parameter $\beta_d$ wird jeweils identisch zu $\beta_m$ gewählt, und der Gewichtungsfaktor $\alpha$ in Formel (6.20) wird auf 1.5 gesetzt. Für $\beta_0$ werden die Werte 0.0, 0.2, 0.4, 0.6, 0.8 und 1.0 betrachtet. Der Wert von $\beta_m$ wird zwischen 0.1 und 0.5 in Schritten von 0.1 variiert. Zusätzlich werden für die Repräsentation des *Tracker*-Zielmodelles die Farbräume RGB und HSV betrachtet.

Zu beachten ist, dass der *Tracker* in jedem bearbeiteten Bild eine Kopf-Schulter-Hypothese auswählt und die Übrigen verwirft. Um die Ergebnisse mit denjenigen ohne *Tracking* vergleichen zu können, wird das Detektionsexperiment für die betrachteten Hintergrundmodellierungen wiederholt. Dabei werden die Detektorhypothesen dergestalt nachbearbeitet, dass jeweils nur diejenige Hypothese behalten wird, welche die meisten benachbarten Hypothesen aufweist (Tabelle 7). Dieses Vorgehen liefert somit ebenfalls maximal eine Detektionshypothese pro Bild, wodurch die im Vergleich sehr niedrige Fehldetektionsrate zustande kommt. Die korrekten Erkennungsraten sind im Vergleich zu Tabelle 6 etwas niedriger, liegen aber bei allen Varianten über 90%. Das zeigt, dass die Größe der Detektionscluster ein geeignetes Auswahl- und Qualitätsmerkmal darstellt.

Die Ergebnisse für die in Tabelle 6 aufgeführten Hintergrundmodellkonfiguration mit sigmoider Verfallsfunktion sind in Abbildung 45 gegeben. Für die anderen betrachteten Konfigurationen des Hintergrundmodelles ergeben sich sehr ähnliche Ergebnisse. Alle Zahlenwerte finden sich in Anhang A.2.

Es ist erkennbar, dass eine Erhöhung des Wertes von $\beta_0$ tendenziell zu einer schlechteren Detektionsrate und einer höheren Fehldetektionsrate führt (diese beiden Kenn-

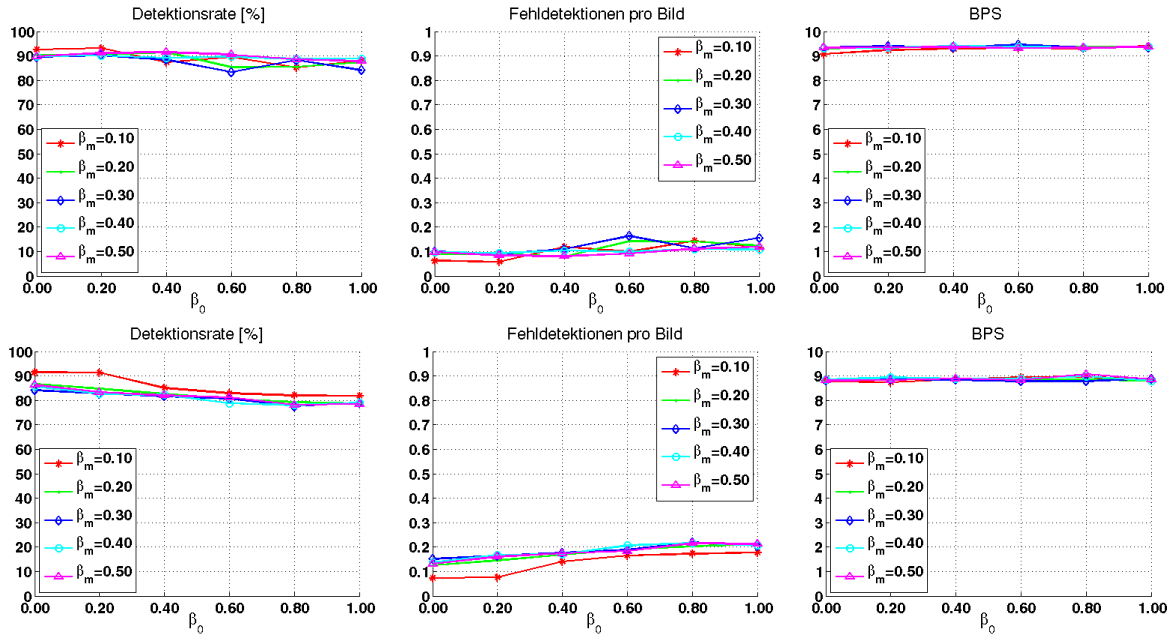


Abbildung 45: Erkennungsrate (links), Fehldetektionsrate (mitte) und mittlere Bildrate (rechts) bei Verwendung des *Trackers* mit verschiedenen Parametrisierungen und einer Hintergrundmodellierung mit sigmoider Verfallsfunktion. Oben: Berechnung des *Tracking*-Zielmodelles im RGB-Farbraum. Unten: HSV-Farbraum. Die Zahlenwerte sind in Anhang A.2 gegeben.

zahlen sind in vorliegendem Experiment stark korreliert). Eine mögliche Erklärung hierfür ist, dass es bei einer fehlerhaften Initialisierung des *Trackers* für große $\beta_0$ deutlich länger dauert, bis der Fehler detektiert und der *Tracker* neu initialisiert wird. Es ist daher ratsam, dem *Tracker* anfangs eine geringe Güte zuzuweisen. Dies führt dazu, dass seinen Ergebnissen erst dann vertraut wird, wenn sie längere Zeit mit den Ergebnissen des Personendetektors konsistent sind.

Die Variation von $\beta_m$ ergibt hingegen kein klar erkennbares Verhaltensmuster. Dieser Parameter hat starken Einfluß darauf, wann ein *Tracker*versagen detektiert wird. Hohe Parameterwerte führen zu einer schnelleren Anpassung der *Tracker*güte, was einerseits zu einer schnellen Rückweisung des aktuellen *Trackers* führen kann, andererseits aber bedeutet, dass einer *Tracker*hypothese bereits nach wenigen konsistenten Ergebnissen vertraut wird. Das führt zu einem insgesamt wenig stabilen Verhalten. Niedrige Werte von $\beta_m$, d.h. langsame Veränderung des *Tracker*-Gütewertes, in Verbindung mit niedrigen Werten von $\beta_0$ führen zu den insgesamt besten Ergebnissen.

Der RGB-Farbraum erwies sich als besser geeignet, als der HSV-Farbraum: Die maximal erreichte Detektionsrate ist geringfügig höher, der Einfluß von Parameteränderungen ist weniger stark. Eine mögliche Ursache liegt in der hohen Rauschanfälligkeit der H-Komponente bei dunklen Farben und geringer Farbsättigung: Der *Tracker* verfolgt den Oberkörperbereich einer Person, d.h. deren Kleidung. Im vorliegenden Datensatz überwiegen gedeckte Kleidungsfarben (vgl. Abbildung 41). Deshalb ist die Modellierung im RGB-Farbraum hier stabiler, solange die Beleuchtungsbedingungen annähernd konstant bleiben.

Die Verarbeitungsgeschwindigkeit zeigt kaum Abhängigkeit von den *Tracking*parametern und liegt üblicherweise zwischen 8 und 9.5 BPS. Bei Verwendung des HSV-Farbraumes und ansonsten identischer Parametrisierung ist die Verarbeitung etwas langsamer, weil die Farbraum-Konvertierung des Eingabebildes einen zusätzlichen Rechenaufwand bedeutet.

Insgesamt hat die Verwendung des *Trackers* nicht die gewünschte Wirkung: Die besten Detektions- und Fehldetektionsraten, die in den Experimenten erreicht wurden (Tabelle 8), sind vergleichbar mit den in Tabelle 7 angegebenen Resultaten, die ohne *Tracking* ermittelt wurden. Weil außerdem die erhoffte Erhöhung der Effizienz nicht eintritt – der zusätzliche Rechenaufwand zur Berechnung und Lokalisierung des *Tracking*modelles übersteigt den Effizienzgewinn durch die zusätzliche Suchraumeinschränkung – ist für die hier verwendeten Daten der Einsatz des *Trackers* nicht gewinnbringend. Hierbei ist allerdings anzumerken, dass die betrachteten Bildsequenzen nur selten Situationen aufweisen, in denen die erwarteten Vorteile des *Trackers* zum Tragen kommen können. Beispielsweise stehen die Personen selten völlig still, so dass kaum Ausfallzeiten des Hintergrundmodelles überbrückt werden müssen. Die abrupten Szenenwechsel führen außerdem zwangsläufig zu Fehlern, weil der *Tracker* bei einem solchen Übergang versagen muss.

## 7.4 OFFLINE HAUTFARBMODELLIERUNG

Beim offline Hautfarbmodell handelt es sich um ein statisches Modell der Farbverteilung, das zur Laufzeit nicht adaptiert wird. Seine Aufgabe besteht in erster Linie darin, eine robustere Initialisierung und Schätzung des personalisierten adaptiven online Farbmodelles zu erreichen. Es muss deshalb für große Variationen der Beleuchtungsbedingungen zuverlässige Ergebnisse liefern. Im Folgenden werden zwei Ansätze zur statischen Modellierung – die in Kapitel 6.3.1 vorgestellten GMM und *skin locus* – evaluiert und hinsichtlich ihrer Anwendbarkeit bewertet. Hierfür kommt der Datensatz FINCA-HF zum Einsatz.

| HG-Modell | $\beta_0$ | $\beta_m$ | Farbraum | Erk.rate [%] | FD/Bild | BPS | Zeit/Bild $\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| Exp | 0.0 | 0.4 | RGB | 92.30±0.46 | 0.07 | 8.97 | 111.52±10.20 |
| Exp | 0.0 | 0.1 | HSV | 90.68±0.5 | 0.08 | 8.25 | 121.15±14.18 |
| Rampe | 0.0 | 0.1 | RGB | 92.82±0.45 | 0.06 | 9.03 | 110.72±11.03 |
| Rampe | 0.2 | 0.1 | HSV | 91.08±0.49 | 0.07 | 8.68 | 115.20±11.29 |
| Sigmoid | 0.2 | 0.1 | RGB | 93.24±0.43 | 0.06 | 9.24 | 108.27±10.41 |
| Sigmoid | 0.0 | 0.1 | HSV | 91.50±0.48 | 0.7 | 8.78 | 113.83±10.77 |

Tabelle 8: Jeweils bestes *Tracking*-Ergebnis für verschiedene Hintergrundmodelle.

### 7.4.1 *Gauss'sche Mischverteilung*

Im Rahmen der Evaluierung des GMM-Modelles wurden die Farbräume RGB, nRG, LAB und HSV untersucht. Zunächst wurden die Trainingsdaten in die entsprechenden Farbräume überführt. Anschließend wurden die annotierten Vordergrundpixel aller Beleuchtungsbedingungen für die jeweilige Weißabgleich-Einstellung der Kamera für das Training des Vordergrundmodelles verwendet, während ein gleichartiges Hintergrundmodell auf allen Pixeln der Hintergrunddaten trainiert wurde. Hierfür wurde der EM-Trainingsalgorithmus mit jeweils 20 Iterationen benutzt. Die Anzahl der Normalverteilungen in der Mischverteilung wurde von fünf bis 50 in Schritten von fünf variiert. Zusätzlich wurde jeweils die in der Literatur verbreitete Modellierungsart mit nur einer Normalverteilung untersucht. Für die Klassifikation wurden die Vorder- und Hintergrundmodelle mit jeweils gleicher Anzahl von Mischungskomponenten verwendet. Die Klassifikationsentscheidung wird anhand der pixelweisen a-posteriori-Wahrscheinlichkeit $P(\mathcal{V}|b(x,y))$ des kombinierten Modelles getroffen. Dabei erfolgt die Bewertung der Detektionsleistung analog zu Kapitel 7.3.1 anhand der EER der ROC-Kurven. $\Omega_D$ in (7.1) ist in diesem Falle definiert als die Anzahl der Pixel $b(x,y)$, für die gilt $P(\mathcal{V}|b(x,y)) \geqslant \delta$. $\Omega_A$ ist die Menge der annotierten Vordergrundpixel und $\Omega$ ist die Menge aller Hintergrundpixel des Evaluierungsdatensatzes.

Die besten Ergebnisse für jeden Farbraum sind in Tabelle 9 dargestellt. Eine Tabelle mit allen Einzelergebnissen findet sich in Anhang A.3. Das insgesamt beste Ergebnis wird für den RGB-Farbraum mit der *indoor*-Einstellung und fünf Mischungskomponenten erreicht. Hier beträgt die EER 0.90, d.h. bei einer Detektionsrate von 90% werden 10% der Hintergrundpixel fehlerhaft als Vordergrund detektiert. Würde die Hautfarbdetektion sich alleine auf das statische Modell verlassen, wäre die Anzahl

| | outdoor | | indoor | | kombiniert | |
|---|---|---|---|---|---|---|
| Farbraum | EER | #Mix | EER | #Mix | EER | #Mix |
| RGB | 0.86 | 40 | **0.90** | **5** | 0.78 | 1 |
| HSV | 0.82 | 35 | 0.863 | 10 | - | - |
| nRG | 0.76 | 45 | 0.83 | 5 | - | - |
| LAB | 0.67 | 45 | 0.73 | 25 | - | - |

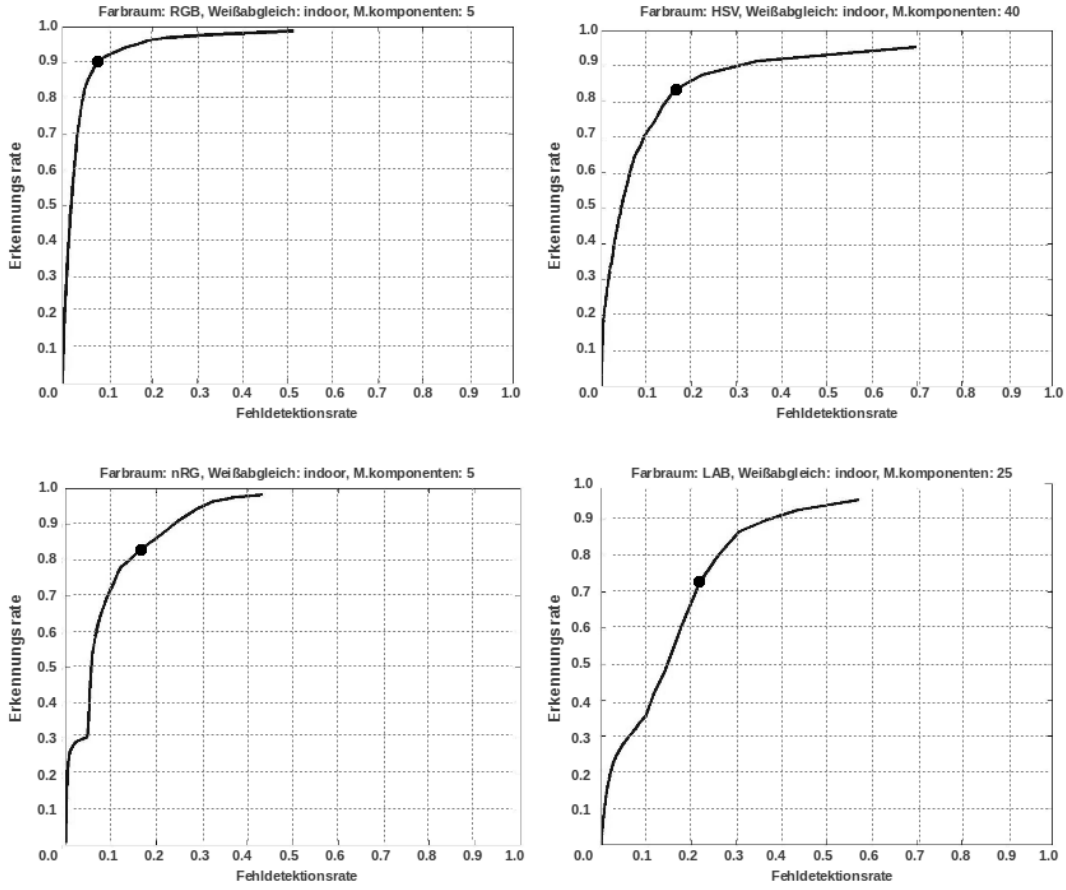


Tabelle 9: Evaluationsergebnisse der statischen Hautfarbmodellierung mit GMM. Oben: Beste Ergebnisse nach Farbraum und Weißabgleich. Angegeben ist jeweils die EER und die Anzahl Mischungskomponenten des besten Modells. Für den besten Farbraum (RGB) wurde außerdem ein kombiniertes Modell mit Daten für beide Weißabgleich-Einstellungen erstellt. Unten: Zugehörige ROC-Kurven. Von links oben nach rechts unten: RGB, HSV, nRG, LAB, alle *indoor*. Der EER-Punkt ist markiert.

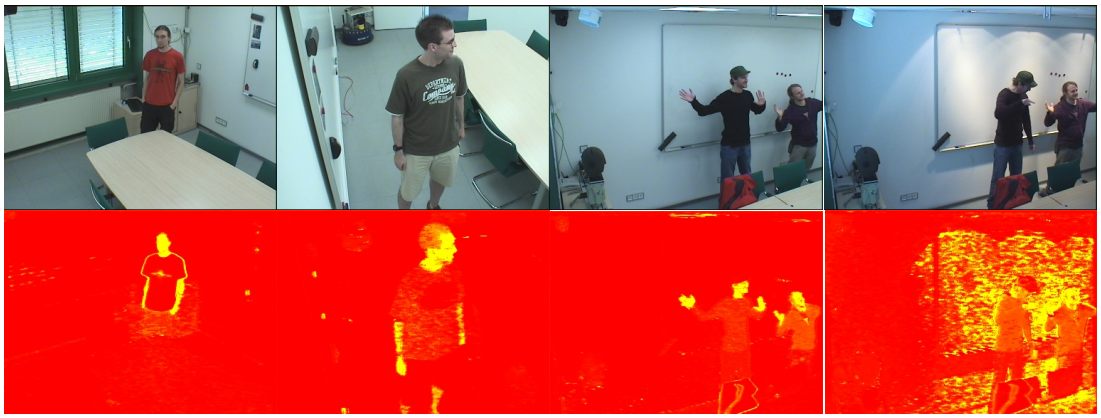

Abbildung 46: Einige Ergebnisse der Hautfarbdetektion mit dem besten GMM-Modell (RGB, *indoor*, 5 Mischungskomponenten). Oben: Originalbilder. Unten: Hautfarb-Wahrscheinlichkeitskarten (gelb = hohe Wahrscheinlichkeit). Die rechte Spalte zeigt eines der schlechtesten Ergebnisse und verdeutlicht den Einfluß der Umgebungsbeleuchtung: Das gelbliche Licht der Tafelstrahler führt dazu, dass die eigentlich weiße Tafel zu großen Teilen als Hautfarbe klassifiziert wird.

von Fehldetektionen also groß und der Detektor insgesamt nicht robust genug (vgl. die Beispielergebnisse in Abbildung 46). Die Aufgabe des offline Modelles ist jedoch in erster Linie die Auswahl von Kandidaten für das Training des online Hautfarbmodelles. Daher ist eine gute Detektionsleistung wichtiger als eine geringe Anzahl Fehldetektionen. Viele Fehldetektionen können durch die Einbeziehung der Ergebnisse des Personendetektors, wie in Kapitel 6.5.2 beschrieben, zurückgewiesen werden.

Auffällig ist, dass die Ergebnisse im RGB-Farbraum besser sind, als diejenigen, die in anderen Farbräumen erzielt wurden. Das widerspricht der häufig in der Literatur zur Hautfarbdetektion zu findenden Annahme, dass perzeptionsorientierte oder chromatische Farbräume (d.h. Farbräume, in denen Farb- und Helligkeitskomponenten getrennt sind) prinzipiell besser geeignet sind (vgl. z.B. [87]) und unterstützt die Beobachtungen in [181]. Insbesondere HSV und LAB sind bei dunklen Farben, gedämpfter Beleuchtung oder Abschattungen sehr rauschanfällig, weil die Farbanteile dann nicht zuverläsig ermittelt werden können. Das ist in einem realistischen Innenraumszenario von Nachteil. Zudem hat die gewählte Modellierung mit GMM den konzeptionellen Vorteil, dass theoretisch jede Verteilung über dem Merkmalsraum mit beliebiger Genauigkeit nachgebildet werden kann. Daher ist die einzige Voraussetzung für eine gute Modellierung, dass die Hautfarbpixel im gewählten Farbraum kompakte Häufungsgebiete bilden. Lage und Form dieser Häufungsgebiete können nahezu beliebig

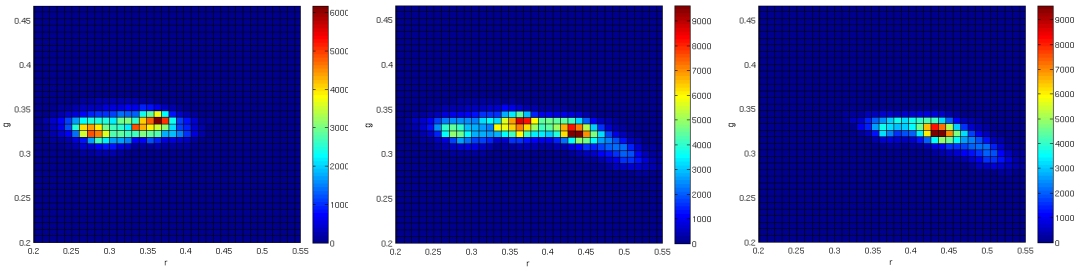

Abbildung 47: Grafische Darstellung der Histogramme der Farbhäufigkeit für die Hautfarbdetektion mittels *skin locus*. Von links nach rechts: *indoor*, kombiniert, *outdoor*.

sein. Dadurch hängt die Modellierung kaum von der Charakteristik des gewählten Farbraumes ab. Informelle Experimente, die hier nicht wiedergegeben werden, ergaben zudem, dass das Weglassen eines oder mehrerer Farbkanäle – beispielsweise der Helligkeitskomponente V bei HSV – zu wesentlich schlechteren Ergebnissen führt. Auch das bestätigt wiederum einige Beobachtungen in [181].

Der Grund für die deutlich unterschiedlichen Ergebnisse für die beiden Weißpunkt-Einstellung wird durch einen visuellen Vergleich ersichtlich (vgl. Abbildung 38): Die Einstellung *indoor* resultiert in einer bläulichen Farbtemperatur, während *outdoor* einen Gelb- bzw. Rotstich aufweist. Es ist daher nicht verwunderlich dass die resultierenden Farbverteilungen sehr verschieden sind. Dies ist auch eine mögliche Erklärung dafür, dass mit dem kombinierten Modell über beide Modi schlechtere Ergebnisse erzielt wurden, als mit den Einzelmodellen: Durch die unterschiedlichen Farbtemperaturen decken die Hautfarbpixel einen großen Bereich des Farbraumes ab. Somit kann keine gute Klassentrennung mehr erreicht werden.

### 7.4.2 *Skin Locus*

Als Alternative zur GMM-basierten Modellierung wurde der *Skin Locus* untersucht. Als Basis dient ein Histogramm der nRG Farbwerte mit $100 \times 100$ Bins. Die Grundannahme, dass die Hautfarbpixel ein kompaktes Häufungsgebiet bilden, ist mit den verwendeten Trainingsdaten erfüllt (Abbildung 47). Die Grenzen dieses Häufungsgebietes werden mit einem Polynom vierten Grades approximiert. Die Klassifikation besteht dann in einem einfachen Lagetest und liefert eine rein binäre Entscheidung. Damit ist die Berechnung von ROC-Kurven in der gleichen Weise wie zuvor nicht möglich.

Zum Zwecke einer einheitlichen Darstellung der Ergebnisse wird deshalb wie folgt vorgegangen: Der Schwellwert $\delta$ wird auf die Häufigkeitswerte in den Histogrammbins

angewendet. Nur Bins mit einem Wert größer als $\delta$ werden zur Berechnung der Begrenzungspolynome verwendet. Damit ergibt sich durch Variation von $\delta$ eine ROC-ähnliche Kurve. Der Unterschied ist, dass hier jeder Messpunkt auf einem anderen Klassifikator (d.h. einem anderen Begrenzungspolynom) beruht, während bei einer ROC-Kurve der Klassifikator stets derselbe ist. Die Bezeichnung „ROC“ wird daher im Folgenden ausdrücklich vermieden.

Als Bewertungsmaß dient wiederum die in gleicher Weise wie zuvor ermittelte EER der Ergebniskurven. (Abbildung 48). Die Ergebnisse sind insgesamt deutlich schlechter als diejenigen, die mit GMM erreicht wurden. Die EER betragen 0.73 für *indoor*, 0.80 für *outdoor* und 0.75 für die Kombination beider Weißabgleichseinstellungen. Hohe Erkennungsraten sind nur mit einer sehr großen Anzahl von Fehldetektionen erreichbar (vgl. Abbildung 48 unten mit Abbildung 46). Damit ermöglicht der *Skin Locus* zwar eine sehr effiziente Klassifikation, der Effizienzgewinn geht aber zu stark zu Lasten der Ergebnisqualität. Für die Anwendung im Gestenerkennungssystem ist daher die Modellierung mittels GMM zu bevorzugen.

## 7.5 HANDDETEKTION

Die Güte des Handdetektors ist für den vorgestellten Ansatz zur Gestenklassifikation von großer Bedeutung. Weil einige wenige Fehldetektionen bzw. fehlerhafte 3D-Kombinationshypothesen im Zweifelsfalle weniger schlimm sind, als häufige Detektionsausfälle, wird in der folgenden Evaluierung besonderes Augenmerk auf eine möglichst hohe Detektionsrate gelegt. In diesem Zusammenhang stellte sich in informellen Experimenten schnell heraus, dass der SIFT-basierte Handdetektor (vgl. Kapitel 6.5.1) in dieser Hinsicht ungeeignet ist. Hierfür gibt es eine einfache Erklärung: Die *Keypoint*-Detektion im Skalenraum benötigt kontraststarke Bilddaten, um zuverlässig *Keypoints* zu extrahieren. Bereits bei geringfügig dämmriger Beleuchtung oder leichter Bewegungsunschärfe nahm die Anzahl gefundener *Keypoint*-Kandidaten dramatisch ab, so dass keine zuverlässige Detektion mehr gewährleistet war. Angesichts der mitunter sehr geringen Auflösung der Hände in den Bilddaten und der häufig auftretenden Bewegungsunschärfe erwiesen sich auch die kantenbasierten SIFT-Deskriptoren als ungeeignet. Die vielversprechenden Ergebnisse in [162] wurden somit auf Daten erreicht, die sich als nicht realistisch für das Zielszenario herausstellten. Die Kombination der SIFT-basierten Detektion mit einer Hautfarbdetektion führte zwar zu insgesamt stabileren Ergebnissen [149], diese wurden jedoch auf denselben Daten ermittelt. Daher kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass auch dieser Ansatz auf realistischeren Daten versagt – im schlimmsten Falle redu-

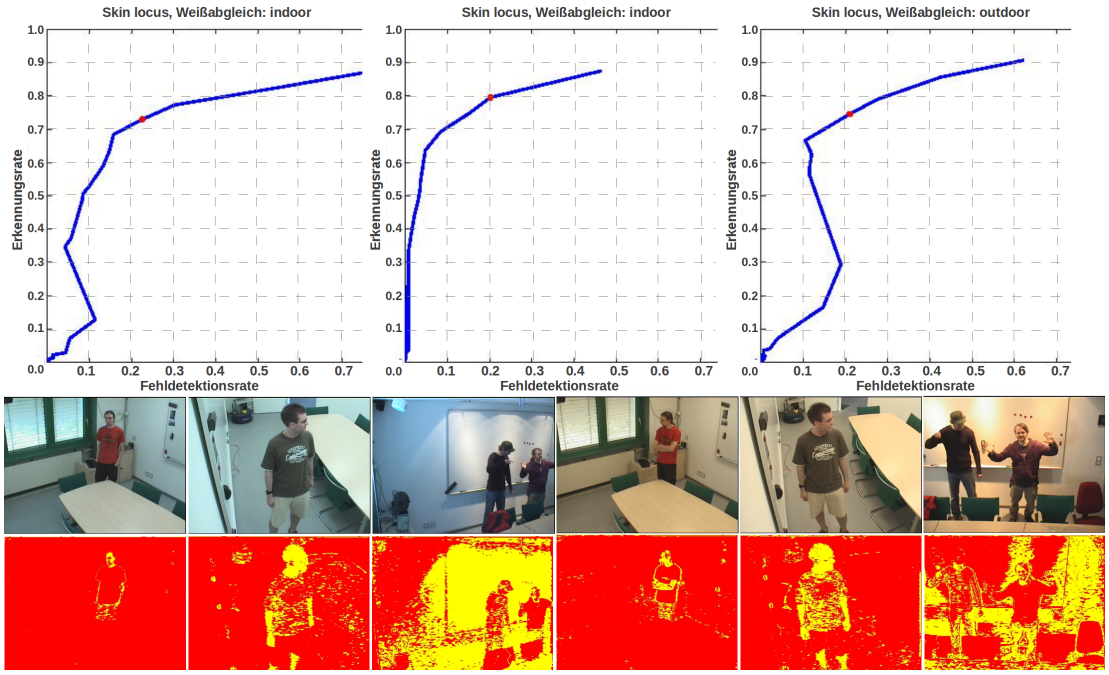


Abbildung 48: Oben: Ergebniskurven für die Hautfarbdetektion mittels *skin locus*. Von links nach rechts: *indoor*, *outdoor*, kombiniert. Der EER-Punkt ist rot markiert. Unten: Beispielergebnisse mit Originalbildern (obere Reihe) und zugehörigen Klassifikationsergebnissen (untere Reihe). Die linken drei Bildpaare entsprechen den Bildern in Abbildung 46 (Einstellung *indoor*) und dienen zum Vergleich der Qualität. Die restlichen Ergebnisse wurden mit dem besten Modell (Einstellung *outdoor*) ermittelt ($\delta = 300$).

ziert sich die Handdetektion hierbei auf eine reine statische Hautfarbmodellierung. Zudem ist die Anwendung der SIFT-basierten Verfahren aufgrund der aufwändigen *Keypoint*-Extraktion und der NN-Suche in üblicherweise sehr großen Datenbanken zeitaufwändig, so dass die realisierten Implementierungen weit davon entfernt sind, echtzeitfähig zu sein. Aus diesem Grund werden diese Ansätze an dieser Stelle nicht detailliert betrachtet, sondern es sei lediglich auf die Evaluationsergebnisse in den erwähnten Publikationen verwiesen.

Demzufolge konzentriert sich die Evaluierung auf die Detektion mit Hautfarbe und Bewegung. Deren Detektionsleistung hängt zum Einen von den Parametern der Hintergrundmodellierung ab, zum Anderen von den Extraktionsparametern $\delta_V$ und $\gamma_H$ (vgl. Kapitel 6.5.2). Um den Parametersuchraum einzuschränken, werden für die Hintergrundmodellierung lediglich die drei vorgeschlagenen guten Konfigurationen

aus Tabelle 6 betrachtet. Die beiden anderen Parameter werden jeweils in einem groben Raster ($\delta_V = \{50, 70, 90, 110, 130, 150\}$[3], $\gamma_H$={0.5, 0.6, 0.7, 0.8, 0.9}.) variiert. Für die Hautfarbmodellierung werden die Farbräume HSV (Online-Modell) bzw. RGB (Offline-Modell) verwendet. Bei letzterem Modell handelt es sich um das in Kapitel 7.4.1 ermittelte beste GMM (Weißabgleich *indoor*, fünf Mischungskomponenten). Die Größe des gleitenden Extraktionsfensters sowie die Kernel-Bandbreite für die Clusterung der Hypothesen wurde auf ein Achtel der Breite des jeweiligen Kopf-Schulter Detektionsrechteckes festgelegt.

Die Evaluierung erfolgt mit zwei verschiedenen Datensätzen. Zunächst wird der Einfluß der Extraktionsparameter auf dem Datensatz FINCA-PH evaluiert. Dieser Datensatz ist relativ einfach in dem Sinne, dass die Personen ebenso wie die Handflächen meistens der Kamera zugewandt sind und überwiegend langsame Bewegungen vollführt werden. Trotzdem umfasst er eine Reihe verschiedener Blickwinkel und Szenarien mit anspruchsvollen Hintergründen (vgl. Abbildung 41), so dass die Handdetektion keineswegs trivial ist. Die Personen befinden sich zudem in einigen Fällen relativ weit von der Kamera entfernt, so dass die Handregionen in den aufgenommenen Bildern mitunter sehr klein sind. Insgesamt enthält der Datensatz 5960 annotierte Hände in 3109 Bildern.

Der zweite Evaluationsdatensatz umfasst einen Teil des Datensatzes FINCA-G mit vier Personen (P0, P1, P10, P11) und 13116 Bildern. Hierbei ist zu beachten, dass dort meistens nur eine Hand aktiv ist, während die andere sich in Ruheposition befindet. Es ist also zu erwarten, dass die inaktive Hand in den meisten Fällen aufgrund der fehlenden Bewegung nicht detektiert werden kann. Deshalb wurde die Annotation der Hände dahingehend geändert, dass die jeweils inaktive Hand speziell gekennzeichnet ist. Sie wird dann während der Evaluation nicht beachtet, d.h. weder korrekte noch fehlende Detektionen auf inaktiven Händen werden gezählt (auch nicht als falsch positive Detektionen). Der Datensatz enthält somit noch 13158 annotierte Hände. Die Personen befinden sich im Allgemeinen etwas näher an der Kamera als in FINCA-PH und die Handregionen sind somit größer. Jedoch treten schnelle Handbewegungen mit entsprechender Bewegungsunschärfe auf und die Personen sind häufig nicht frontal zur Kamera orientiert.

Als Qualitätsmaße werden wiederum die Detektionsrate und die mittlere Anzahl Fehldetektionen pro Bild betrachtet. Hierbei gilt eine Hand als korrekt detektiert, wenn sich der Mittelpunkt der Hypothese innerhalb des Annotationsrechteckes befindet. Weiterhin wird die mittlere Verarbeitungsrate in Bildern pro Sekunde (BPS) angegeben.

3 Bezogen auf den maximalen mittleren Grauwert einer Region von 255. Diese Werte entsprechen mittleren Vordergrund-Pseudowahrscheinlichkeiten von $\{0.20, 0.27, 0.35, 0.43, 0.51, 0.59\}$.

Im Folgenden kommen zwei einfache Heuristiken zur Fehlervermeidung zum Einsatz, die sich im realen Betrieb als sinnvoll herausgestellt haben: Zum Einen wird die in Kapitel 6.3.2 angesprochene Heuristik zur Detektion eines Versagens der Hintergrundmodellierung verwendet, indem die Verarbeitung eines Bildes abgebrochen wird, falls die Vordergrund-ROI mindestens 90% des Bildes einnimmt. Dies geschieht typischerweise bei abrupten Übergängen zwischen unterschiedlichen Szenen. In diesem Fall werden das Hintergrundmodell und das online Hautfarbmodell reinitialisiert.

Zum Anderen wird die Verarbeitung abgebrochen, falls eine sehr große Anzahl von Handkandidaten (>250) in einem Bild gefunden wurde. Dies weist auf ein degeneriertes Hautfarbmodell hin, z.B. aufgrund vieler falscher Kopfdetektionen in kurzer Zeit. Folglich wird in einem solchen Fall das online Hautfarbmodell zurückgesetzt.

### 7.5.1 *Online-Modell*

Zunächst wird die Detektionsleistung ohne Verwendung des statischen Hautfarbmodelles bewertet. Das bedeutet, dass die Hautfarbmodellierung ausschließlich online durch Extraktion von Pixeln innerhalb der Kopf-Schulter Regionen erstellt wird. Somit führen fehlerhafte Kopfhypothesen, insbesondere in der Initialisierungsphase, zu Fehlern im Farbmodell. Abbildung 49 (oben) zeigt exemplarisch das Ergebnis für den Datensatz FINCA-PH bei Verwendung des Hintergrundmodelles mit der Sigmoidfunktion als Verfallsfunktion. Die Resultate für die anderen Hintergrund-Modellierungsarten unterscheiden sich nur unwesentlich. Alle Zahlenwerte sind in Anhang A.4 gegeben.

Die beste Detektionsleistung von 86.78±0.88% wird für og. Hintergrundmodellierung bei $\delta_V = 90, \quad \lambda_H = 0.5$ erreicht. Werden nur die Ergebnisse mit korrekter Kopflokalisierung betrachtet, beträgt die Detektionsrate 87.78±0.90%. Dabei treten im Mittel 1.74 Fehldetektionen pro Bild auf, die mittlere Verarbeitungsgeschwindigkeit beträgt 7.91 BPS. Die anderen Varianten der Hintergrundmodellierung erreichen bei identischer Parametrisierung vergleichbare Ergebnisse von 86.19±0.90% bei 1.53 Fehldetektionen (asymptotische Exponentialfunktion) bzw. 86.36±0.89% bei 1.79 Fehldetektionen (Rampenfunktion).

Der Einfluß von $\delta_V$ und $\gamma_H$ ist dabei grundsätzlich so wie erwartet: Bei beiden Parametern führt eine Erhöhung des Wertes zu einer Verringerung sowohl der Detektions- als auch der Fehldetektionsrate. Die Tatsache, dass die Detektionsraten für kleine Werte von $\delta_V$ zunächst niedriger sind als das Maximum, liegt in der oben erwähnten Rückweisungsheuristik begründet: Für sehr niedrige $\delta_V$ tritt häufiger der Fall auf,

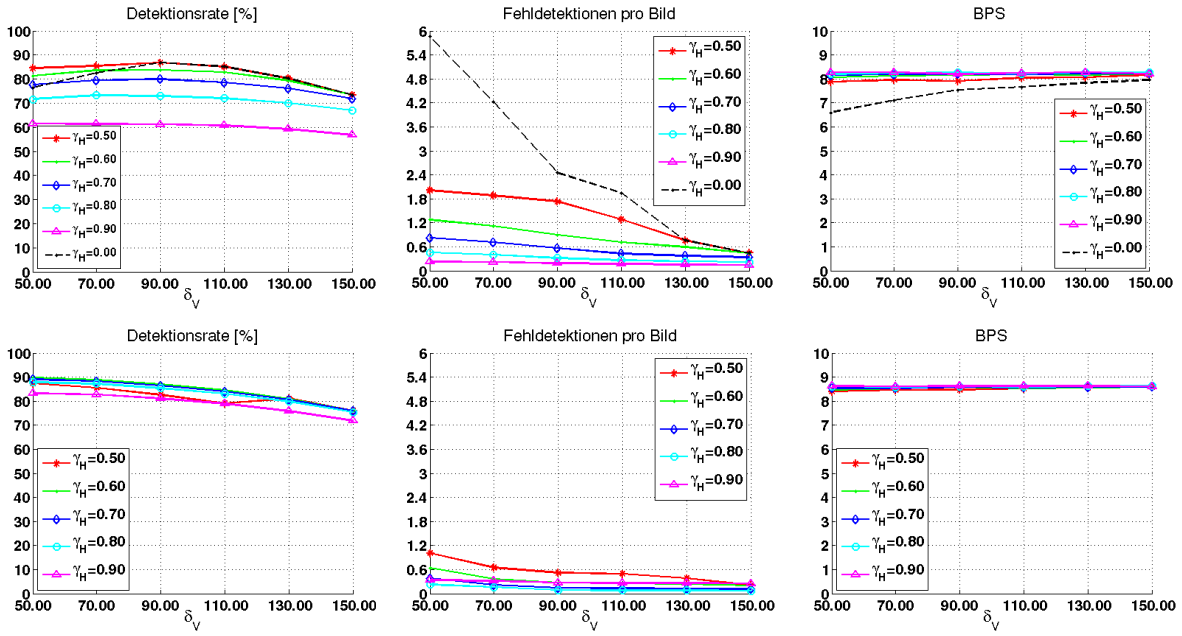


Abbildung 49: Ergebnisse der Handdetektion auf den Datensätzen FINCA-PH (oben) und FINCA-G (unten) unter Verwendung des Hintergrundmodelles mit Sigmoidfunktion und des online Hautfarbmodelles. Von links nach rechts: Detektionsrate, mittlere Anzahl Fehldetektionen pro Bild, mittlere Verarbeitungsgeschwindigkeit in Bildern pro Sekunde.

dass ein Bild aufgrund zu vieler Handkandidaten verworfen wird, was zu einer global niedrigeren Detektionsrate führt.

Aus den Diagrammen ist ersichtlich, dass der relative Rückweisungsschwellwert $\gamma_H$ sehr starke Auswirkungen sowohl auf die Detektions- als auch auf die Fehldetektionsrate hat. Insbesondere kommt es im betrachteten Innenraumszenario häufig vor, dass eine Hand von der Deckenbeleuchtung angestrahlt wird, während die andere sich im Schatten des Körpers befindet. In einem solchen Fall kann die Hautfarbwahrscheinlichkeit der beiden Hände stark unterschiedlich sein. Eine zu restriktive Wahl von $\gamma_H$ führt dann dazu, dass eine der Hände immer verworfen wird.

Dennoch hat dieser Parameter seine Berechtigung. Um dies zu verdeutlichen, sind in Abbildung 49 (oben) zusätzlich die Messkurven für $\gamma_H = 0.0$ angegeben. Dies führt für die meisten Werte von $\delta_V$ zu einer weitaus höheren Anzahl von Fehldetektionen, ohne eine Verbesserung der maximal erreichten Detektionsrate zu bewirken. Zusätzlich ist ein deutlicher Einfluß auf die Verarbeitungsgeschwindigkeit zu beobachten. Der

Hauptgrund hierfür liegt im *Clustering* der Handhypothesen, das ineffizient wird, wenn sehr viele Hypothesen vorliegen.

Der Einfluß des initialen Auswahlschwellwertes $\delta_V$ ist weniger stark und leichter kontrollierbar. Für den betrachteten Datensatz ist das Verhalten insofern günstig, als bis zu einem Wert von ca. $\delta_V = 110$ der Verlust bei der Detektionsrate gering ist, während die Fehldetektionsrate deutlich verringert wird. Durch geeignete Parameterwahl lassen sich somit Konfigurationen finden, bei denen bei über 80% Detektionsrate im Mittel weniger als eine Fehldetektion pro Bild auftritt.

Diese Ergebnisse zeigen einerseits, dass das gelernte online Hautfarbmodell in Verbindung mit der Hintergrundmodellierung in realistischen Szenarien für die Detektion von hautfarbenen Vordergrundregionen geeignet ist. Andererseits zeigt sich der Vorteil der zweistufigen Hypothesenfilterung, weil durch die richtige Wahl des zweiten relativen Schwellwertes $\gamma_H$ viele Fehldetektionen verworfen werden können, ohne die Detektionsrate negativ zu beeinflussen.

Abbildung 49 (unten) zeigt die Resultate auf FINCA-G mit identischem Hintergrundmodell. Eine Variation der Parameterwerte führt im Wesentlichen zu einem ähnlichen Verhalten wie im vorherigen Experiment, bei allerdings insgesamt höheren Detektions- und niedrigeren Fehldetektionsraten. Diese Resultate zeigen, dass hautfarbene und nicht hautfarbene Regionen in diesem Experiment sehr gut durch das Modell separiert werden. Der Grund hierfür liegt in der Charakteristik des Datensatzes (größere Gesichts- und Handregionen). Aufgrund der eingangs erwähnten Tatsache, dass in den meisten Bildern nur eine einzige Hand annotiert ist, fällt der Einfluß von $\gamma_H$ im Vergleich zum vorherigen Experiment deutlich geringer aus.

Die beste Detektionsrate liegt bei $89.70 \pm 0.53\%$ ($\delta_V = 50, \quad \gamma_H = 0.6$) bei im Mittel 0.64 Fehldetektionen pro Bild. Werden nur die Ergebnisse der Bilder mit korrekter Kopfdetektion betrachtet, beträgt die Handdetektionsrate bei einigen Parameterkombinationen über 90%. Angesichts des realistischen Aufnahmeszenarios ist das ein sehr gutes Ergebnis.

Die mittlere Verarbeitungszeit liegt bei den meisten Parameterkombinationen für alle Hintergrundmodelle zwischen 7.5 und 9 Bildern pro Sekunde. Die Berechnung der pixelweisen Hautfarbwahrscheinlichkeiten und die Extraktion der Hypothesen im gleitenden Fenster ließe sich sehr einfach durch Verwendung von SIMD-Erweiterungen oder Auslagerung auf Grafikhardware parallelisieren und somit beschleunigen.

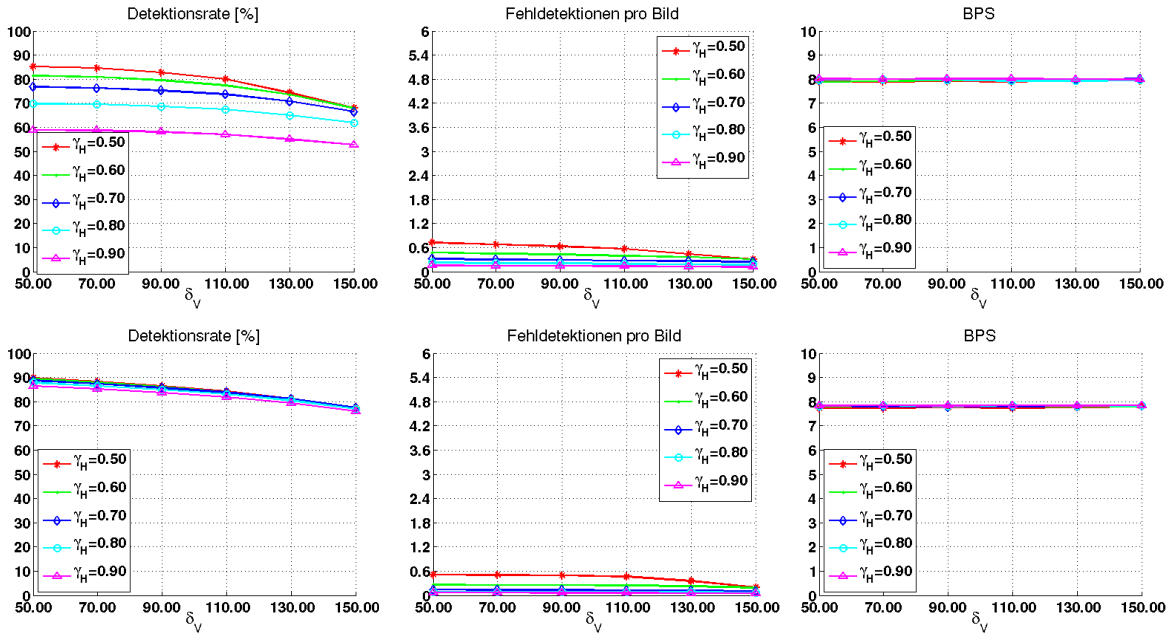


Abbildung 50: Ergebnisse der Handdetektion auf den Datensätzen FINCA-PH (oben) und FINCA-G (unten) unter Verwendung des Hintergrundmodelles mit Sigmoidfunktion und der Kombination von statischem und online Hautfarbmodell. Von links nach rechts: Detektionsrate, mittlere Anzahl Fehldetektionen pro Bild, mittlere Verarbeitungsgeschwindigkeit in Bildern pro Sekunde.

### 7.5.2 *Kombination*

Im folgenden Experiment wird nun zusätzlich das statische Hautfarbmodell verwendet. Hierbei werden die Hautfarbpixel für das Lernen des online Modelles wie in Kapitel 6.5.2 erläutert mit ihrer Bewertung durch das statische GMM gewichtet. Weil auf diese Weise implizit geeignete Pixel ausgewählt werden, entspricht dieses Vorgehen einem selektiv adaptiven Ansatz, im Gegensatz zur blinden Adaption im vorherigen Experiment. Für den Datensatz FINCA-G wurde das statische Modell dabei gemäß des jeweils verwendeten Weißabgleich-Modus bei der Aufnahme gewählt. Deshalb kommen die beiden in Tabelle 9 aufgeführten besten Modelle im RGB-Farbraum für die Modi *indoor* und *outdoor* zum Einsatz.

In Abbildung 50 sind die Ergebnisse für die gleiche Hintergrundmodellierung wie zuvor (Sigmoidfunktion) dargestellt. Die Zahlenwerte aller Experimente finden sich in Anhang A.5.

| HG-Modell | $\delta_V$ | $\gamma_H$ | Datensatz | Modellierung | Erk.rate [%] | FD/Bild | BPS |
|---|---|---|---|---|---|---|---|
| Exp | 90 | 0.5 | FINCA-PH | ON | 86.19±0.90 | 1.53 | 7.67 |
| Exp | 50 | 0.5 | FINCA-PH | OFF+ON | 85.02±0.93 | 0.66 | 7.45 |
| Exp | 50 | 0.6 | FINCA-G | ON | 87.19±0.58 | 0.61 | 7.99 |
| Exp | 50 | 0.5 | FINCA-G | OFF+ON | 87.49±0.58 | 0.49 | 7.53 |
| Sigmoid | 90 | 0.5 | FINCA-PH | ON | 86.78±0.88 | 1.74 | 7.91 |
| Sigmoid | 50 | 0.5 | FINCA-PH | OFF+ON | 85.22±0.92 | 0.73 | 7.90 |
| Sigmoid | 50 | 0.6 | FINCA-G | ON | 89.70±0.53 | 0.64 | 8.47 |
| Sigmoid | 50 | 0.5 | FINCA-G | OFF+ON | 89.85±0.53 | 0.51 | 7.72 |
| Rampe | 90 | 0.5 | FINCA-PH | ON | 86.36±0.89 | 1.79 | 8.01 |
| Rampe | 50 | 0.5 | FINCA-PH | OFF+ON | 84.92±0.93 | 0.74 | 7.62 |
| Rampe | 50 | 0.7 | FINCA-G | ON | 87.43±0.58 | 0.46 | 8.51 |
| Rampe | 50 | 0.6 | FINCA-G | OFF+ON | 87.95±0.57 | 0.27 | 7.74 |

Tabelle 10: Beste Handdetektions-Ergebnisse für verschiedene Hintergrundmodelle sowie Hautfarb-Modellierung nur mit online Modell (ON) bzw. zusäzlichem statischen GMM (OFF+ON).

Die selektive Auswahl führt aufgrund der Anwendung des zusätzlichen Modelles auf das Eingabebild zu einer leicht verringerten Verarbeitungsgeschwindigkeit, die bei den meisten Experimenten zwischen 7 und 8 BPS liegt. Während der Einfluss auf die maximal erreichbaren Detektionsraten (vgl. Tabelle 10) vernachlässigbar ist, zeigt sich eine sehr deutliche Verringerung der mittleren Anzahl von Fehldetektionen im Vergleich zu den Experimenten ohne statisches Modell. Aufgrund der zusätzlichen Gewichtung durch das offline Model haben nicht hautfarbene Pixel nur einen sehr geringen Einfluß auf das online Modell. Das ist insbesondere bei fehlerhaften Kopf-Schulter Hypothesen sowie Hinterkopf- bzw. Profilansichten von Vorteil, weil das online Modell sich in diesen Fällen nicht an die im Suchbereich vorhandenen falschen Farben adaptiert.

Dieser Vorteil ist jedoch offensichtlich nur dann gegeben, wenn das verwendete statische Farbmodell zum gegebenen Szenario passt. Es besteht somit die Gefahr, dass bei starken Schwankungen der Umgebungsbeleuchtung oder Änderungen der Aufnahmebedingungen (z.B. Änderung des Weißabgleichs der Kamera) die Hautfarbmodellierung versagt. Dennoch zeigen die Ergebnisse, dass eine robuste Detektion von

| | FINCA-G Train | FINCA-G Val | HumanEVA Train | HumanEVA Val |
|---|---|---|---|---|
| KVal1 | 0-7, 11-16 | 8-10 | 1,2 | 3 |
| KVal2 | 0-4, 8-16 | 5-7 | 1,3 | 2 |
| KVal3 | 0-1, 5-16 | 2-4 | 2,3 | 1 |
| KVal4 | 2-15 | 0-1, 16 | 1,2 | 3 |
| KVal5 | 0-12, 16 | 13-15 | 1,3 | 2 |

Tabelle 11: Zusammensetzung der Kreuzvalidierungs-Datensätze für die Evaluierung der Gestenerkennung. Die Zahlen bezeichnen die Nummer der jeweiligen Person im Datensatz.

Händen mit dem vorgestellten Ansatz in realistischen Innenraumszenarien möglich ist (vgl. Ergebnisse in Abbildung 41). Die sehr geringe mittlere Anzahl von Fehldetektionen bedeutet außerdem, dass während der Trajektorienaggregation nur sehr wenige falsche 3D-Punkhypothesen auftreten. Somit ist einerseits eine gute Genauigkeit zu erwarten, andererseits hat dies auch positive Auswirkungen auf die Effizienz, weil nur wenige alternative Hypothesen verfolgt werden müssen.

## 7.6 GESTENERKENNUNG

Die Evaluierung der trajektorienbasierten Gestenerkennung mittels HMM besteht aus zwei Teilen. Zunächst wird die Eignung der vorgeschlagenen Merkmale untersucht. Zu diesem Zweck wird ein Klassifikationsexperiment auf den segmentierten Gesteninstanzen von Datensatz FINCA-G durchgeführt. Anschließend wird die Fähigkeit zur automatischen Segmentierung unter Verwendung eines Rückweisungsmodelles auf längeren Beobachtungsfolgen getestet. Hierfür kommen zusätzlich Trajektoriendaten aus dem *HumanEVA*-Datensatz zum Einsatz. Die Auswertung erfolgt anhand einer fünffachen Kreuzvalidierung. Dabei werden jeweils 14 Personen aus FINCA-G für das Training und die verbliebenen drei für die Validierung verwendet. Für die *HumanEVA* Daten (7.1.5) stehen nur drei Personen zur Verfügung (die vierte dient nur Testzwecken und die zugehörigen *Motion Capture* Trajektoriendaten sind nicht öffentlich verfügbar). In Tabelle 11 ist die genaue Aufteilung in die Kreuzvalidierungssätze angegeben. Die Auswertung erfolgt also in personenunabhängiger Weise auf 15 der 17 Personen aus 7.1.4 und – im Falle des Segmentierungsexperimentes – auf allen Personen aus 7.1.5.

### 7.6.1 *Klassifikationsexperiment*

Der HMM-Klassifikator wird wie in Kapitel 6.7.4 beschrieben trainiert. Dabei wird je ein Modell pro Gestenart erstellt. Links- und rechtshändige Instanzen derselben Geste werden nicht unterschieden und durch dasselbe Modell repräsentiert. Die 2D-Gestentrajektorien wurden mit dem in Kapitel 6.7.1 beschriebenen Verfahren aus den Punkthypothesen aggregiert, per Strahlenschnitt zu 3D-Trajektorien kombiniert und der impulsbasierten Neuabtastung unterzogen.

Für die Merkmalsextraktion wurden verschiedene Größen des gleitenden Fensters betrachtet. Fenstergrößen kleiner als fünf ergeben für die Berechnung der Nachbarschafts-Merkmale keinen Sinn. Bei großen Fenstergrößen wird einerseits der Glättungseffekt so stark, dass wichtige Charakteristiken der Trajektorien verloren gehen, andererseits wird die Anzahl der Merkmalspunkte mit zunehmender Fenstergröße immer kleiner, wodurch weniger Daten für das Training der Gestenmodelle zur Verfügung stehen. Aus diesen Gründen wurden Fenstergrößen von fünf, sieben und neun Trajektorienpunkten gewählt, jeweils mit 50% Überlappung.

Die Anzahl der Komponenten des GMM und die Anzahl der Zustände (Länge) der Modelle wurden ebenfalls variiert. Jedes Gestenmodell wird mit einer Zustandsanzahl initialisiert, die proportional zur Sequenzlänge der kürzesten beobachteten Trainingsinstanz der jeweiligen Geste ist[4]. Die konkrete Länge eines HMM ergibt sich somit aus der minimalen Beobachtungslänge multipliziert mit einem einstellbaren Skalierungsfaktor, der für ein gegebenes Experiment global für alle Modelle gewählt wird. Untersucht wurden Codebücher mit 150, 200 und 250 Dichten sowie Längenskalierungsfaktoren von 1.0, 0.75 und 0.5.

#### *Ergebnisse für Einzelmerkmale*

Auf eine detaillierte Aufstellung der Resultate für alle Parameterkombinationen wird aufgrund der enormen Datenmenge an dieser Stelle verzichtet, eine ausführlichere Übersicht findet sich in Anhang A.6. Die besten Ergebnisse, aufgeschlüsselt nach Merkmalsart und Trajektorienrepräsentation, sind in Tabelle 12 dargestellt. Hierbei bezeichnet „3D“ Merkmale, die aus den 3D-Trajektorien berechnet wurden, „2D“ bezieht sich auf Merkmale, die aus der Projektion auf die Aktionsebene berechnet wurden, und Δ3D bzw. Δ2D sind die jeweiligen Delta-Merkmale. Das Koordinatensystem in der Aktionsebene wurde gemäß des in Kapitel 6.7.2 als „parallel zur Grundlinie“

4 Diese heuristische Längeninitialisierung ist eine Besonderheit des verwendeten HMM Toolkits *ESMERALDA*.

| Merkmal | 3D | 2D | Δ3D | Δ2D |
|---|---|---|---|---|
| Rohe Trajektorie | 56.1 (3.7) | 76.2 (3.2) | **84.0 (2.7)** | 78.0 (3.1) |
| Normalisierte Trajektorie | 78.5 (3.1) | **79.8 (3.0)** | 49.0 (3.7) | 76.2 (3.2) |
| Norm. polare Trajektorie | 80.3 (3.0) | 76.7 (3.1) | 77.5 (3.1) | **81.3 (2.9)** |
| Krümmung | 39.6 (3.6) | 39.8 (3.6) | 43.2 (3.7) | 42.7 (3.7) |
| Kopfabstand | 59.7 (3.6) | 61.2 (3.6) | 27.7 (3.3) | 25.9 (3.3) |
| Orientierungsänderung | 42.5 (3.7) | 40.5 (3.6) | 45.5 (3.7) | 46.1 (3.7) |
| Geschwindigkeit | **80.8 (2.9)** | 71.3 (3.4) | 75.5 (3.2) | 68.7 (3.4) |
| Nachbarschaft | 71.6 (3.3) | 66.7 (3.5) | 67.1 (3.5) | 56.8 (3.7) |

Tabelle 12: Klassifikationsergebnisse (% korrekt klassifizierte Gesten) für einzelne Merkmalstypen (links) und entsprechende Delta-Merkmale (rechts). Die besten Ergebnisse für jeden Trajektorientyp sind in Fettdruck dargestellt. Konfidenzintervalle (±) sind in Klammern angegeben.

bezeichneten Vorgehens gewählt. Frühere Experimente [158] mit anderer Wahl des Koordinatensystems ergaben deutlich schlechtere Ergebnisse.

Im Falle der 3D-Trajektorien wurde die beste Klassifikationsleistung von 84.0 ±2.7% (alle hier und im Folgenden angegebenen Konfidenzintervalle beziehen sich auf ein Signifikanzniveau von 95%) mit den Delta-Merkmalen der rohen Trajektorie erzielt. Das zugehörige beste Modell hat eine Bakis-Topologie, ein Codebuch mit 150 Dichten und einen Skalierungsfaktor für die Modelllänge von 0.75. Die Fenstergröße für die Merkmalsextraktion betrug fünf. Die mittleren Modelllängen für die besten Gestenmodelle sind in Tabelle 13 dargestellt. Mit identischen Parametern beträgt die Detektionsleistung für die gleichartigen 2D Merkmale 78%. Das beste Ergebnis für die 2D Merkmale ist 81.3 ±2.9% mit den Delta-Merkmalen der normalisierten polaren Trajektorie, einer linearen Modelltopologie und einer Fenstergröße von neun. Dieses Modell hat eine Codebuchgröße von 200 und einen Skalierungsfaktor von 0.75. Die Anzahl der Zustände ist für dieses Modell etwa um den Faktor vier kleiner (vgl. Tabelle 13). Das liegt zum Einen an der Modellart – Bakis-Modelle werden in ESMERALDA jeweils automatisch in doppelter Länge initialisiert, um das Überspringen von Zuständen korrekt zu repräsentieren – zum Anderen am größeren Merkmalsextraktionsfenster, das zu kürzeren Trajektorien führt.

Neben den normalisierten Trajektorienrepräsentationen lieferten auch das Geschwindigkeitsprofil und die Nachbarschaftsmerkmale sowohl für 3D als auch 2D brauch-

| Geste | h. Winken | v. Winken | Aufw. | Abw. | Kreis | Herk. | Wegg. | Stop | Zeigen |
|---|---|---|---|---|---|---|---|---|---|
| 3D | 27 | 54 | 11 | 18 | 37 | 12 | 12 | 16 | 19 |
| 2D | 7 | 14 | 3 | 5 | 10 | 3 | 3 | 4 | 5 |

Tabelle 13: Mittlere Modelllängen der besten Gestenmodelle.

| Geste | Fehler | Geste | Fehler | Geste | Fehler |
|---|---|---|---|---|---|
| Kreis | 17.1 | Herkommen | 13.4 | Abwärts | 14.8 |
| Weggehen | 10.6 | Zeigen | 8.7 | Stop | 8.0 |
| Aufwärts | 8.0 | h. Winken | 21.8 | v. Winken | 29.7 |

Tabelle 14: Mittlerer Rekonstruktionsfehler pro Trajektorienpunkt in mm für die Projektion auf die Aktionsebene.

bare Ergebnisse. Interessanterweise scheint selbst das sehr einfache eindimensionale Kopfabstand-Merkmal in der Lage zu sein, einige Charakteristika der Gesten zu kodieren. Dies weist darauf hin, dass selbst sehr schwache Informationen über relative Positionen von Körperteilen einen Beitrag zur erfolgreichen Klassifikation leisten können. Die Ergebnisse in [17], wo relative Körperteilpositionen mit einfachen linguistischen Merkmalen kodiert und für die Erkennung von Zeichensprache-Zeichen eingesetzt werden, stützen diese Vermutung.

Die Projektion der 3D Trajektorien auf ihre Aktionsebene bedeutet einen Informationsverlust. Deshalb ist für die 2D Merkmale mit einem Verlust an Genauigkeit zu rechnen. Die Ergebnisse zeigen allerdings, dass dieser Verlust klein ist. Somit scheint die Annahme, dass typische Gesten eine inhärente planare Natur aufweisen, gerechtfertigt zu sein. Sie wird zusätzlich durch die geringen mittleren Rekonstruktionsfehler bei der Projektion (Tabelle 14) gestützt. Die Wahl des Koordinatensystems in der Aktionsebene scheint jedoch kritisch für die erreichbare Genauigkeit zu sein (vgl. Ergebnisse in [158]). Insbesondere hat sich die Wahl der Hautpkomponenten der projizierten Trajektorie als Koordinatenachsen in diesem Zusammenhang als ungeeignet erwiesen. Ein möglicher Grund ist, dass hierdurch die globale Orientierung der Trajektorie verloren geht.

Die Klassifikationsleistung mit der rohen 2D-Trajektorie kommt mit 76.2 $\pm$3.2% dem besten Ergebnis nahe. Dies zeigt, dass die Aktionsebene in der Tat dazu geeignet ist, von der globalen Position und Orientierung einer Person zu abstrahieren. Wie wichtig diese Abstraktion für das vorliegende realistische Szenario ist, zeigt das Ergebnis der

rohen 3D-Trajektorie, das mit nur 56.1 $\pm$3.7% signifikant schlechter ist. Trotzdem ist dieses Resultat besser als erwartet, was darauf schließen lässt, dass der Datensatz eine Tendenz in Richtung bestimmter globaler Posen aufweist. Erkennbar ist dies auch daran, dass die Delta-Merkmale der rohen Trajektorie das beste Ergebnis liefern, denn diese sind nicht invariant gegenüber der globalen Orientierung.

Eine Inspektion der Daten ergab, dass tatsächlich die meisten der Testpersonen dazu neigten, sich einer der Kameras annähernd frontal zuzuwenden, obwohl sie nicht dazu aufgefordert wurden. Das ist jedoch kaum verwunderlich: Die Kameras stellen in diesem Fall den Adressaten oder Interaktionspartner dar. Menschen wenden sich während einer Interaktion intuitiv ihrem Interaktionspartner zu. Deshalb scheint das Problem der Ansichtsinvarianz nicht so gravierend zu sein, wie man annehmen könnte, sofern der Nutzer sich bewusst ist, wo sich die Sensorik in seiner Umgebung befindet[5]. Anders gesagt scheinen selbst einfache Sensoren als Avatare oder Interface-Agenten einer Umgebungsintelligenz fungieren zu können.

Abschließend wurde zum Vergleich eine weitere Versuchsreihe durchgeführt, die kein gleitendes Fenster bei der Merkmalsextraktion verwendet, sondern die Merkmale direkt aus den interpolierten Trajektorien berechnet. Die Ergebnisse sind in Anhang A.6 unter Fenstergröße eins aufgeführt. Die Resultate sind tendenziell geringfügig schlechter im Vergleich zur gefensterten Merkmalsextraktion, jedoch sind die Unterschiede nicht signifikant. Eine Ausnahme bilden die Nachbarschaftsmerkmale, die hier in einem Fenster der Größe fünf um den aktuellen Trajektorienpunkt herum berechnet wurden, sowie die Delta-Merkmale der normierten Trajektorien. Während die Ergebnisse mit Nachbarschaftsmerkmalen für den ungefensterten Fall etwas besser sind, sind die Ergebnisse mit den Delta-Trajektorien zum Teil dramatisch schlechter und liegen teilweise nur geringfügig über dem Niveau zufälligen Ratens (11.1%). Diese Merkmale weisen – hauptsächlich aufgrund der Normierung mit der Körpergröße der Person – einen im Vergleich sehr geringen Dynamikumfang auf und sind daher sehr anfällig gegen Störungen und Lokalisierungsfehler. Demzufolge wirkt sich die fehlende Glättung durch die Fensterung hier besonders stark aus.

*Ergebnisse mit PCA-transformierten Merkmalen*

Obige Resultate zeigen, dass mit einigen der vorgeschlagenen alternativen Trajektorienmerkmale gute Ergebnisse erzielt werden können. Es ist allerdings nicht zu erwarten, dass eine einfache Konkatenation mehrerer Merkmale zu einer deutlichen Verbesserung führt, weil alle Merkmale aus der gleichen Ursprungstrajektorie berech-

5 Unter dem *Disappearing Computer*-Paradigma bleibt dieses Problem bestehen, denn es schließt ein, dass alle Repräsentanten der Umgebungsintelligenz für den Nutzer unsichtbar sind.

net wurden. Daher kann angenommen werden, dass sie stark korrelliert sind. Diese Vermutung wird durch die Experimente in [158] bestätigt, bei denen verschiedene Merkmalskombinationen untersucht wurden und keine Verbesserung der Klassifikationsleistung erreicht werden konnte. Zusätzlich führt eine einfache Merkmalskonkatenation zu einer höheren Dimensionalität des Merkmalsraums und einer erhöhten Modellkomplexität. Ein Nachteil statistischer Klassifikatoren wie HMM ist, dass sie typischerweise große Trainingsdatenmengen benötigen. Deshalb kann die Erhöhung der Merkmalsdimensionalität bei schlechter Datenlage sogar zu einer schlechteren Modellqualität führen.

Aus diesen Gründen wird hier ein anderer Ansatz verfolgt, um die Vorteile verschiedener Merkmalsrepräsentationen zu kombinieren und dennoch die Dimensionalität klein zu halten: Alle Merkmale (inklusive Delta-Merkmale) einer Trainingsstichprobe werden zunächst auf Mittelwert Null und Varianz eins normiert, um ihren unterschiedlichen Dynamiken Rechnung zu tragen. Danach wird eine Hauptkomponentenanalyse (vgl. Kapitel 2.4) angewendet. Die resultierenden Hauptkomponenten (HK) sind dekorrelliert und umfassen Beiträge aller Merkmalstypen. Dies wird im gleichen fünffachen Kreuzvalidierungsschema wie im vorherigen Experiment wiederholt. Die Transformationen für jeden Kreuzvalidierungssatz wurden dabei nur auf den jeweiligen Trainingsdaten berechnet und global auf alle Daten des Satzes angewendet.

Anschließend wurden HMM-Klassifikatoren mit unterschiedlichen Anzahlen von Hauptkomponenten und den gleichen Parameterkombinationen wie zuvor trainiert. In Tabelle 15 ist dargestellt, welcher Anteil der Datenvarianz durch die jeweilige Hauptkomponentenzahl repräsentiert wird. Die Klassifikationsergebnisse (Tabelle 16) zeigen tatsächlich eine Verbesserung. Bemerkenswerterweise zeigen sowohl die Wahl der Größe des Fensters bei der Merkmalsextraktion als auch der Modellparameter nur sehr geringen Einfluss (eine Übersicht über alle Ergebnisse ist in Anhang A.7 gegeben). Die einzige Ausnahme bildet der Längenskalierungsfaktor. Wird dieser auf 1.0 gesetzt, d.h. die Modelllänge entspricht der Länge der kürzesten Beobachtung der jeweiligen Gestenklasse, so führt das in den meisten Fällen zu einer deutlichen Verringerung der Klassifikationsleistung. Eine mögliche Erklärung hierfür ist, dass durch die größere Anzahl freier Modellparameter die Trainingsdatenmenge nicht mehr ausreichend ist.

Davon abgesehen wurden mit den meisten Parameterkombinationen vergleichbare Klassifikationsergebnisse von 88% bis 89% erreicht. Insbesondere wurde das global beste Ergebnis von 89.8 $\pm$2.3% mit zwei substantiell unterschiedlichen Parametersätzen erzielt: Das eine Modell hat eine Bakis Topologie und verwendet ein Codebuch der Größe 250 und einen Skalierungsfaktor von 0.75. Das andere Modell ist linear und verwendet ein Codebuch mit 150 Dichten und einen Skalierungsfaktor von 0.5. In beiden Fällen wurden eine Merkmalsfenstergröße von neun und die ersten sieben

| #HK | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 12 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3D | 12.5 | 22.6 | 31.6 | 39.3 | 46.0 | 51.2 | 56.1 | 60.5 | 64.3 | 68.0 | 73.5 | 80.7 |
| 2D | 12.8 | 23.1 | 32.8 | 41.5 | 47.7 | 53.7 | 58.6 | 62.7 | 66.7 | 70.5 | 76.6 | 84.0 |
| 2D+3D | 12.0 | 21.9 | 30.7 | 37.2 | 42.4 | 46.9 | 50.9 | 54.5 | 57.5 | 60.5 | 66.0 | 73.4 |

Tabelle 15: Repräsentierte Datenvarianz in % für verschiedene Anzahlen von HK, gemittelt über die Kreuzvalidierungssätze und Merkmalsfenstergrößen. Die maximale Standardabweichung der dargestellten Werte beträgt weniger als 0.08, was zeigt, dass die durch die Merkmalsextraktion und Zusammensetzung der Kreuzvalidierungssätze entstehenden Variationen nur sehr geringen Einfluß haben.

| #HK | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 12 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3D | 56.1 | 75.9 | 81.6 | 85.3 | 86.3 | **88.3** | 87.8 | 87.2 | 86.7 | 86.0 | 86.7 | 86.7 |
| 2D | 45.7 | 70.6 | 82.0 | 86.0 | 85.6 | **86.6** | 84.6 | 85.0 | 85.3 | 85.6 | 84.6 | 84.4 |
| 2D+3D | 55.2 | 76.7 | 81.7 | 86.2 | 86.6 | 87.8 | **89.8** | 89.2 | 88.6 | 88.5 | 88.2 | 86.5 |

Tabelle 16: Beste Klassifikationsergebnisse für PCA Merkmale in %. Konfidenzintervalle für die hervorgehobenen Werte sind im Text angegeben.

Hauptkomponenten des kombinierten (2D+3D) Merkmalssatzes verwendet. Diese Verbesserung ist signifikant.

Bei alleiniger Verwendung der 2D und 3D Merkmale wurden maximale Klassifikationsraten von 86.6 $\pm$2.5% bzw. 88.3 $\pm$2.4% erreicht. Obwohl die Verbesserungen gegenüber den besten Einzelmerkmalen in diesen Fällen nicht signifikant sind, zeigt sich doch eine deutliche Tendenz, die darauf hinweist, dass die Kombination verschiedenartiger Merkmale via PCA zu einer Verbesserung der Klassifikationsleistung beitragen kann.

Interessanterweise werden bereits mit sehr wenigen HK gute Ergebnisse erzielt. Die Hinzunahme von mehr als vier HK führte in allen Fällen zu keiner signifikanten Verbesserungen der Ergebnisse. Das ist vorteilhaft, weil die Modellkomplexität – im Sinne der freien Parameter des GMM-Codebuches – gering gehalten werden kann. Werden mehr HK hinzugefügt, führt das ab einem bestimmten Punkt zu einer Verschlechterung der Klassifikationsleistung. Das kann einerseits wiederum in der relativ geringen Trainingsdatenmenge und der erhöhten Modellkomplexität begründet

sein. Andererseits ist denkbar, dass die Hinzunahme weiterer Merkmalsdimensionen nur noch irrelevante Variabilität in die Klassifikationsaufgabe einbringt.

Abschließend zeigt Abbildung 51 die Verwechslungsmatrizen der besten Modelle mit PCA-Merkmalen. Die Verwechslungsmuster sind bei allen Modellen ähnlich und weisen Symmetrien auf: „Zeigen“ wird häufig fälschlicherweise als „Weggehen“ oder „Stop“ klassifiziert. Die erste Verwechslung tritt möglicherweise auf, weil beide Gesten eine Bewegung vom Körper weg ausführen und der Arm bei der Ausführung einer Zeigegeste eine ähnliche bogenförmige Bewegung beschreibt, wie bei „Weggehen“. Die Verwechslung mit „Stop“ geschieht vermutlich, weil beide Gesten kurze Hebebewegungen mit einer ausgeprägten Haltephase am Schluß beinhalten. Die „Aufwärts“ Geste wird ebenfalls gelegentlich mit „Stop“ verwechselt. Das ist nicht verwunderlich, weil ihre erste Phase fast identisch mit der „Stop“ Bewegung ist und erst die zweite Phase (das „nach oben drücken“) beide Gesten voneinander unterscheidet. Ist die zweite Phase also schwach ausgeprägt, kann leicht eine Verwechslung auftreten. Ein weiterer relativ häufig auftretender Fehler ist die Verwechslung von „Herkommen“ und „Stop“. Beide beinhalten eine Bewegung der Hand zur Schulter und sind in diesem Sinne ähnlich. Die gelegentlich auftretende Verwechslung von „Herkommen“ und den beiden „Winken“ Gesten erscheint jedoch merkwürdig. Möglicherweise ist ein Grund dafür, dass die Winkbewegungen ebenfalls bogenförmige Bewegungen der Hand in Richtung Schulter aufweisen. Die Beobachtung, dass diese Fehler weitgehend symmetrisch auftreten, weist darauf hin, dass tatsächlich Ähnlichkeiten zwischen den Gesten die Ursache sind.

### 7.6.2 *Segmentierungsexperiment*

In diesem Experiment soll die Fähigkeit der gewählten Modellierung zur Segmentierung längerer Sequenzen mit mehreren Gesteninstanzen untersucht werden. Die Aufgabe ist also, die Eingabesequenz in sinnvolle Subsequenzen zu segmentieren und den Segmenten jeweils ein Klassenkennzeichen zuzuweisen bzw. sie als unbekannt zurückzuweisen. Als Rückweisungskriterium kommt hierbei ein Nullmodell zum Einsatz, wie in Kapitel 6.7.4 beschrieben. In diesem Zusammenhang wird untersucht, ob ein solches Rückweisungsmodell mit domänenfremden Daten erstellt werden kann.

#### *Training eines Rückweisungsmodelles mit domänenfremden Daten*

Für das Training eines Rückweisungsmodelles wird eine große Stichprobe repräsentativer negativer Trainingsbeispiele benötigt. Derartige Daten sind oft nicht verfügbar bzw. ihre Aufnahme ist mit großem Aufwand verbunden. Daher wird an dieser Stelle

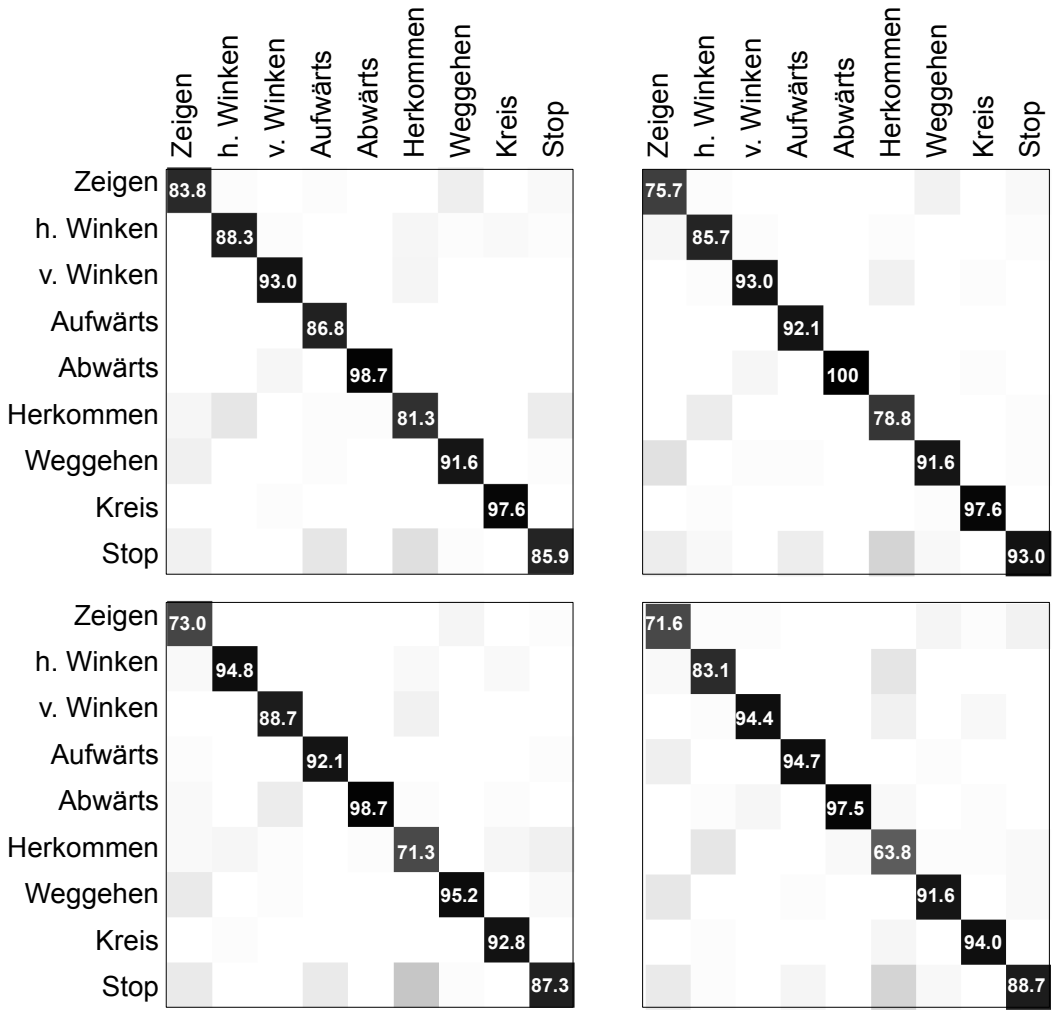


Abbildung 51: Verwechslungsmatrizen für die besten Modelle mit PCA-Merkmalen. Oben: Die beiden besten Modelle mit kombinierte Merkmalen (links: Linear, 150/0.5; rechts: Bakis, 250/0.75). Unten: Beste Ergebnisse für 3D (links) und 2D (rechts). Die Angaben sind in % von 0 (weiß) bis 100% (schwarz).

untersucht, ob es möglich ist, ein Nullmodell mit domänenfremden Daten, d.h. Daten, die in einer anderen experimentellen Umgebung und unter anderen Bedingungen aufgezeichnet wurden, zu trainieren. Hierbei sind jedoch einige Dinge zu beachten. Erstens sollten die Daten aus offensichtlichen Gründen ähnliche Charakteristiken wie die Daten aufweisen, die dem Modell zur Laufzeit präsentiert werden. In vorliegendem Fall heißt das, dass sie üblichen räumlich-zeitlichen Bewegungsmustern menschlicher Körperteile entsprechen sollten. Zweitens sollten alle Daten die gleiche Vorverarbeitung durchlaufen. Und drittens ist die Verwendung globaler Punktkoordinaten in diesem Fall nicht möglich, weil diese von den globalen Koordinatensystemen der jeweiligen Datendomäne abhängen, die sich stark unterscheiden können.

Die hier verwendeten Daten sind die *Motion Capture* Trajektorien des HumanEVA-Datensatzes 7.1.5. Es handelt sich also um Trajektorien menschlicher Bewegungen. Somit erfüllen diese Daten das erste zu beachtende Kriterium. Für die folgenden Experimente werden nur diejenigen Trajektorien benutzt, die üblicherweise deutliche Bewegung aufweisen. Dabei handelt es sich um die Trajektorien der Handgelenke, der Ellenbogen und der Füße. Diese werden auf 20 Hz unterabgetastet, damit ihre Datenrate ungefähr derjenigen der Gestendaten entspricht. Anschließend werden sie durch Subtraktion der jeweils zugehörigen Kopftrajektorie in ein nutzerzentrisches Koordinatensystem überführt. Danach durchlaufen sie dieselbe impulsbasierte Neuabtastung und Merkmalsextraktion im gleitenden Fenster, wie die Gestendaten. Aus den oben genannten Gründen wird die rohe Trajektorie – d.h. die globalen Punktkoordinaten – aus dem Merkmalssatz entfernt.

Damit ergibt sich eine geänderte Merkmalsrepräsentation nach Anwendung der PCA. Aus diesem Grund wurde das Klassifikationsexperiment aus Kapitel 7.6.1 mit dem reduzierten Merkmalssatz wiederholt, um die Auswirkungen zu untersuchen. Es stellte sich heraus, dass die Ergebnisse fast identisch sind (Tabelle 17). Das ist nicht überraschend, denn es war zu erwarten, dass die verschiedenen Trajektorienrepräsentationen in hohem Maße redundant sind.

Mit den *Motion Capture* Trajektorien wird nun das Rückweisungsmodell auf die gleiche Weise trainiert, wie die Gestenmodelle. Der einzige Unterschied besteht in der Initialisierung der Modelllänge. Weil nur wenige Trainingstrajektorien zur Verfügung stehen, die jedoch im Vergleich zu den Gestentrajektorien sehr lang sind, ist eine Initialisierung relativ zur Beobachtungslänge nicht ratsam. Deshalb wird die Modelllänge manuell auf kleine Werte gesetzt, deren Wahl im Folgenden ebenfalls evaluiert wird. Beim Training kommt die identische Aufteilung in Kreuzvalidierungssätze wie zuvor zur Anwendung (vgl. Tabelle 11), erweitert um die Rückweisungsdaten.

| # HK | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|
| 3D | 85.7 (2.6) | **88.0 (2.4)** | 87.9 (2.4) | 86.9 (2.5) | 87.5 (2.5) |
| 2D | **86.3 (2.6)** | 85.5 (2.6) | 84.3 (2.7) | 85.7 (2.6) | 85.9 (2.6) |
| 2D+3D | 85.0 (2.7) | 87.3 (2.5) | **89.8 (2.3)** | 88.3 (2.4) | 88.9 (2.3) |
| 3D | 85.6 (2.6) | 86.9 (2.5) | **87.8 (2.4)** | 85.7 (2.6) | 85.9 (2.6) |
| 2D | 83.0 (2.8) | 83.6 (2.8) | 82.6 (2.8) | **83.7 (2.7)** | 81.8 (2.9) |
| 2D+3D | 84.7 (2.7) | **87.6 (2.5)** | 87.5 (2.5) | 87.5 (2.5) | 87.3 (2.5) |

Tabelle 17: Beste Klassifikationsergebnisse für PCA-Merkmale ohne globale Position in %. Obere Tabelle: Lineares HMM, Codebuch mit 150 Dichten, Längenskalierung 0.5. Untere Tabelle: Bakis-Topologie, 250 Dichten, Skalierung 0.75. Die Werte in Klammern geben die Konfidenzintervalle an.

*Konstruktion der Validierungssequenzen*

Für die Evaluierung der Segmentierungsleistung werden längere Sequenzen mit Gesten- und Hintergrunddaten benötigt. Diese werden wie folgt erzeugt: Zunächst wird für jede Gesteninstanz des Validierungsdatensatzes eine gültige Fortsetzung in den *Motion Capture* Hintergrundtrajektorien gesucht. Der Übergang sollte glatt sein, weil ansonsten die Merkmalsrepräsentation an den Segmentgrenzen abrupte Sprünge aufweisen würde. Damit wäre das Segmentierungsproblem trivial. Deshalb werden nur Trajektorienfortsetzungen als gültig erachtet, bei denen der euklidische Abstand der Nahtpunkte nicht größer ist als der mittlere Punktabstand der jeweiligen Gesteninstanz zuzüglich der Standardabweichung.

Danach wird eine Gesteninstanz zufällig als Startpunkt ausgewählt. Diese wird dann beidseitig fortgeführt, indem weitere Gesteninstanzen ausgewählt werden, die passende Fortsetzungen aufweisen. Dabei wird zwischen den Gesteninstanzen jeweils der Abschnitt der Hintergrundtrajektorie eingefügt, der durch die Fortsetzungspunkte begrenzt wird. Während der Auswahl werden Gestentypen bevorzugt, die zuvor weniger oft als der Durchschnitt ausgewählt wurden. Auf diese Weise wird eine ungefähre Gleichverteilung der in den Validierungssequenzen enthaltenen Gestenklassen erreicht (vgl. Tabelle 18).

Das Ergebnis ist eine alternierende Folge von Gesten- und Hintergrundsequenzen unterschiedlicher Länge und mit glatten Übergängen. Dabei wird sichergestellt, dass in einer Evaluierungssequenz nicht zweimal dieselbe Gesteninstanz verwendet wird,

| | | | | | |
|---|---|---|---|---|---|
| Zeigen | 402 | H. winken | 479 | V. winken | 525 |
| Herkommen | 505 | Weggehen | 500 | Aufwärts | 424 |
| Abwärts | 531 | Kreis | 524 | Stop | 481 |

Tabelle 18: Verteilung der Gestenklassen (Anzahl Instanzen) in den Validierungssequenzen.

um Kreisschlüsse zu vermeiden. Der Vorgang wird abgebrochen, wenn die Sequenz maximal neun Gesteninstanzen enthält oder keine gültige Fortsetzung mehr gefunden wird. Abschließend werden wiederum zufällig Hintergrundsequenzen am Beginn oder Ende der Evaluierungssequenz eingefügt und die Zeitstempel der Trajektorienpunkte werden wie folgt chronologisch verschoben und interpoliert:

Sei $\mathcal{T}_< = \{\tau_0 ... \tau_j, t_0 ... t_j\}$ die aktuelle Sequenz mit den 3D-Punktkoordinaten $\tau_i$ und assoziierten Zeitstempeln $t_i$, an die ein Trajektoriensegment $\mathcal{T}_>$ zeitlich folgend angehängt werden soll. Dabei sei $\mathcal{T}_> = \{\tau_{j+1} ... \tau_{j+m}, t_{j+1} ... t_{j+m}\}$. Bei der Auswahl der Fortsetzung wurde bereits darauf geachtet, dass die euklidischen Abstände der Punkte $\tau_j$ und $\tau_{j+1}$ nicht zu groß sind. Dies gilt jedoch nicht für die Zeitstempel. Um die Interpolation durchzuführen, wird zunächst gefordert, dass die Trajektoriengeschwindigkeit an der Nahtstelle sich nicht sprunghaft ändert. Daher wird der Median der Geschwindigkeiten $\bar{v}_N$ in einem Fenster berechnet, das jeweils fünf Punkte beiderseits der Nahtstelle umfasst. Die neuen Zeitstempel $\hat{t}_i$ ergeben sich dann aus

$$\begin{aligned} \hat{t}_i &= t_N + t_i - t_{j+1}, \quad i = j+1 ... j+m, \\ t_N &= \frac{\|\tau_{j+1} - \tau_j\|}{\bar{v}_N}. \end{aligned} \tag{7.2}$$

Weil die einzelnen Teile einer Evaluierungssequenz von verschiedenen Personen stammen können, muss zudem die Kopfhöhe, die bei der anschließenden Merkmalsberechnung zur Normalisierung benötigt wird, interpoliert werden. Hierfür wird für jede Gesteninstanz die Trajektorie der Kopfpunkte beibehalten und der jeweilige Median der Kopfhöhe berechnet. Für eine dazwischen liegende Hintergrundinstanz wird die Kopfhöhe nun linear zwischen den beiden Medianwerten interpoliert. Dies führt zwar ggf. zu einer leichten Verfälschung der Merkmalsdarstellung der Hintergrundinstanzen, vermeidet jedoch sprunghafte Merkmalsänderungen an den Nahtstellen.

Aufgrund der zufälligen Natur des obigen Auswahlprozesses und der Tatsache, dass manche Trajektorien keine zulässigen Fortsetzungen aufweisen, kann nicht sichergestellt werden, dass alle Gesten- und Hintergrundinstanzen für den Validierungssatz

verwendet werden. Insgesamt wurden 122 von 694 Gesteninstanzen und 50 von 196 Hintergrundtrajektorien nicht verwendet.

Für jedes der fünf Kreuzvalidierungssets wurden auf diese Weise 200 Validierungssequenzen erzeugt. Somit erfolgt die Evaluierung insgesamt auf 1000 Sequenzen mit 4371 Gesteninstanzen. Die mittlere Punktanzahl pro Trajektorie beträgt 220 vor und 548 nach der Neuabtastung. Gesteninstanzen sind dabei immer durch ein Hintergrundsegment variabler Länge voneinander getrennt. Insgesamt enthält der Datensatz 3860 Hintergrundsegmente mit einer mittleren Länge von 26 (Minimum: 6, Maximum: 51) Punkten vor der Neuabtastung. Der Übergang zwischen Gesten und Hintergrundsegmenten erfolgt nicht an einer definierten Ruheposition. Die Validierungssequenzen können mit jeder beliebigen Klasse – inklusive Hintergrund – beginnen oder enden.

*Ergebnisse*

Das Segmentierungsexperiment wurde mit folgenden Parametern durchgeführt: Die Anzahl der Dichten des GMM-Codebuches wurde auf 1024 erhöht, weil nun eine größere Datenmenge zur Verfügung steht und bei ausreichender Datenlage ein größeres Codebuch üblicherweise zu einer verbesserten Modellierungsqualität führt. Für die Initialisierung wurde der Skalierungsfaktor des besten Modelles aus Tabelle 17 (0.5) gewählt. Die Größe des gleitenden Fensters bei der Merkmalsextraktion ist für die Segmentierung ein kritischer Parameter, weil der Beobachtungshorizont großer Fenster verschiedene Gesteninstanzen und Klassen umfassen kann und zudem die absolute Länge der Sequenz beeinflusst wird. Deshalb werden weiterhin alle drei vorher betrachteten Fenstergrößen – fünf, sieben und neun – evaluiert. Als Merkmale werden die ersten sieben HK der 3D-Trajektorienmerkmale verwendet.

Zusätzlich muss die Wahl der Anzahl der Zustände im Rückweisungsmodell evaluiert werden. Die Parameterwahl basiert auf der Überlegung, dass das Rückweisungsmodell aus Gründen der Flexibilität nicht länger sein sollte, als ein typisches Gestenmodell. Deshalb wurde die maximale Zustandszahl gemäß der mittleren Modelllänge der (linearen) Gestenmodelle für die jeweilige Fenstergröße gewählt. Die kleinste mögliche Zustandszahl ist eins, und zwei weitere Modellängen wurden so gewählt, dass sie ungefähr gleichmäßig zwischen den beiden Grenzwerten verteilt sind. Das führt auf die in Tabelle 19 gegebenen Zustandsanzahlen.

Die Dekodierung wurde wiederum mit *Viterbi Beam Search* durchgeführt. Dabei sind alle Modelle gleichberechtigt und treten während der Dekodierung in Konkurrenz zueinander. Es wird also kein Vorwissen über die Problemstruktur in die Modellstruktur eingebracht. Anschließend werden alle Segmente, die dem Rückweisungsmodell

| Fenstergröße | 5 | 7 | 9 |
|---|---|---|---|
| Zustände im Rückweisungsmodell | 1,3,6,8 | 1,2,3,5 | 1,2,3,4 |

Tabelle 19: Übersicht über evaluierte Modelllängen für das Rückweisungsmodell.

zugewiesen wurden, verworfen und zusätzlich die in Kapitel 6.7.4 vorgestellte *log-odd scores* basierte Rückweisung angewendet.

Als Qualitätsmaße werden der Anteil korrekter Detektionen (K), Vertauschungen (V), Einfügungen (E) und Löschungen (L) ermittelt. Diese Maße werden regelmäßig in Segmentierungs- und *Spotting*-Aufgaben angewendet (vgl. z.B. [50, 86, 176]). Eine Gestenhypothese und Annotation werden als koinzident angesehen, wenn die Länge ihrer Überlappung mindestens 50% der Länge der Annotation entspricht. Weisen sie das gleiche Klassenkennzeichen auf, handelt es sich um eine korrekte Detektion, anderenfalls um eine Vertauschung. Gesteninstanzen, für die keine koinzidente Annotation existiert, sind Einfügungen. Annotationen, für die keine koinzidente Hypothese gefunden wird, sind Löschungen. Um ein skalares Entscheidungkriterium zu erhalten, wird der $F_1$-Wert [200] berechnet, der sich als das harmonische Mittel von *Precision* (P) und *Recall* (R) ergibt:

$$F_1 = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{7.3}$$

Hierbei ist *Recall* definiert als die Anzahl korrekter Detektionen (K) dividiert durch die Anzahl von Annotationen (K+V+L), wohingegen *Precision* gegeben ist als K dividiert durch die Anzahl aller Hypothesen (K+V+E).

Durch Variation des Rückweisungsschwellwertes ergeben sich die in Abbildung 52 gezeigten Kurven für die gemäß des $F_1$ Wertes beste Konfiguration ($F_1$=67.4, P=64.5 ±1.4%, R=70.5 ±1.4%). Das zugehörige Modell hat eine Bakis-Topologie, das Rückweisungsmodell hat drei Zustände und die Merkmalsfenstergröße ist fünf. Aus dem K/V/L/E-Graphen wird ersichtlich, dass die häufigsten Fehler Einfügungen, also Fehldetektionen sind. Die verwendeten Daten sind schwierig, weil sie einerseits beliebige glatte Übergänge zwischen Gesten- und Nichtgesteninstanzen ohne definierte Ruhepunkte enthalten. Andererseits sind in den Hintergrunddaten Trajektorien enthalten, die ebenfalls von Handbewegungen stammen und durchaus Ähnlichkeiten mit den betrachteten Gesten aufweisen können (vgl. z.B. die „Gestures“ Aktion des *HumanEVA* Datensatzes). Deshalb ist die große Anzahl von Fehldetektionen nicht überraschend. Der höchste absolute *Recall*-Wert – welcher der Detektionsrate entspricht – beträgt 78.5 ±1.2% (P=50.1 ±1.2%).

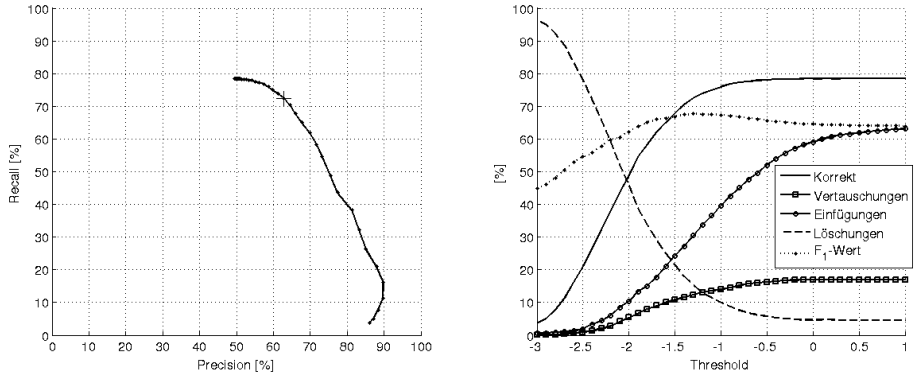


Abbildung 52: Segmentierungsexperiment: *Precision-Recall* Kurve (links) und K/V/L/E Plot (rechts) für das beste Modell, erzeugt durch Variation des *log-odd scores* Rückweisungsschwellwertes. Die Gütewerte sind negative logarithmische Wahrscheinlichkeiten, d.h. kleine Werte entsprechen einer besseren Güte. Der zum besten $F_1$-Wert gehörende Punkt ist in der linken Kurve markiert.

Keine klare Aussage lässt sich für den Einfluss der Zustandsanzahl im Rückweisungsmodell treffen. Ein Modell mit wenigen Zuständen führt generell zu einer leicht höheren Anzahl von Fehldetektionen und somit zu einem geringeren *Precision* Wert. Die Unterschiede sind jedoch vernachlässigbar und werden größtenteils durch eine geeignete Wahl des Rückweisungsschwellwertes ausgeglichen. Genauso sind die Unterschiede zwischen Modellen mit linearer und Bakis-Topologie nicht signifikant (vgl. vollständige Ergebnisse in Anhang A.8).

Es gibt jedoch einen signifikanten negativen Einfluß bei einer Erhöhung der Merkmalsfenstergröße. Mit einer Fenstergröße von neun, identischen Modellparametern wie oben und einem Rückweisungsmodell mit zwei Zuständen (was im Sinne der relativen Länge ungefähr dem obigen Modell mit drei Zuständen entspricht) fällt der $F_1$ Wert auf 62.4 (P=60.1 $\pm$1.4%, R=64.8 $\pm$1.4%) und die maximal erreichbare Detektionsrate auf 70.1 $\pm$1.4%. Experimente mit anderen Parametersätzen (Anhang A.8) bestätigen, dass die Ergebnisse im Mittel erheblich schlechter sind.

Ein möglicher Grund besteht wieder in der Datenlage: Aufgrund der gestiegenen Parameteranzahl des GMM-Codebuches und der aus dem größeren Fenster resultierenden kürzeren Trajektorienlänge ist es denkbar, dass die Datenmenge nicht mehr ausreichend für eine robuste Modellierung ist. Tatsächlich führte eine Wiederholung des Experimentes mit den gleichen Parametern, aber einer Codebuchgröße von 500 zu einer Verbesserung ($F_1$=66.4 (P=65.8 $\pm$1.4%, R=67.0 $\pm$1.4%), höchster *Recall*-Wert 74.5

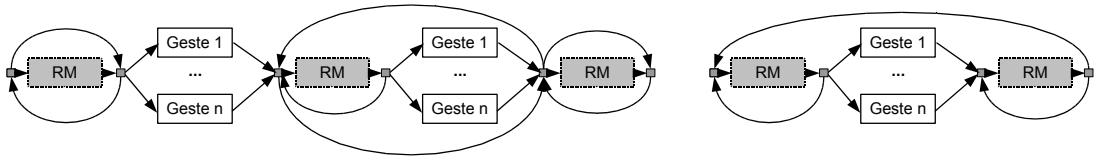


Abbildung 53: HMM-Modellstruktur und Verwendung von Vorwissen. Links: eHMM1; Rechts: eHMM2. RM steht für Rückweisungsmodell.

±1.3%). Diese Verbesserung ist zwar signifikant, jedoch bleiben die *Recall*-Werte stets signifikant schlechter, als für die kleinere Fenstergröße.

Eine interessante Frage ist, ob sich die Ergebnisse durch Einbeziehung von Vorwissen über die Problemstruktur verbessern lassen. Um dies zu untersuchen, wurden zwei HMM mit eingeschränkterer Modellstruktur verwendet (vgl. Abbildung 53). Das erste (eHMM1) beinhaltet nur schwache Einschränkungen, indem gefordert wird, dass jede Sequenz eine optionale Hintergrundinstanz zu Beginn und Ende hat und sich dazwischen eine alternierende Folge aus jeweils einer Geste gefolgt von mindestens einer Hintergrundinstanz befindet. Das entspricht exakt der tatsächlichen Struktur der Evaluierungssequenzen. Das zweite Modell (eHMM2) erzwingt, dass eine Gesteninstanz immer von mindestens einer Hintergrundinstanz auf beiden Seiten begrenzt wird. Das ist für die verwendeten Sequenzen nicht immer der Fall, also verwendet dieses Modell eine im Vergleich weniger flexible Struktur.

Die Verwendung des Modelles eHMM1 hatte keinen Einfluß auf die Segmentierungsergebnisse. Alle Werte sind nahezu identisch, weshalb sie hier nicht gesondert aufgeführt werden. Weil die Modellstruktur sehr flexibel gehalten ist und praktisch jederzeit jede beliebige Kombination von Gesten- und Nichtgesteninstanzen erlaubt, ist dieses Modell weitgehend äquivalent zum uneingeschränkten Fall.

Die Einführung stärkerer strukturelle Beschränkungen in eHMM2 führt hingegen zu einer signifikanten Abnahme der Anzahl von Einfügungen (Abbildung 54) und somit im Mittel zu einem höheren *Precision*-Wert, hat jedoch einen negativen Einfluß auf die erreichbare Detektionsrate. Der höchste mit diesem Modell und der gleichen Parametrisierung wie zuvor erreichte *Recall*-Wert beträgt 70.6 ±1.3% (P=68.9 ±1.4%), der maximale $F_1$-Wert erhöht sich leicht auf 70.0 (P=69.9 ±1.4%, R=70.0 ±1.4%). Die starre Modellstruktur beschränkt die möglichen Übergänge zwischen Teilmodellen, was der Grund für die geringere Anzahl von Fehldetektionen ist. Jedoch erzwingt sie zusätzlich die Ausrichtung der Sequenz an einer vorgegebenen Struktur, die nicht immer der tatsächlichen Struktur entspricht. Insbesondere wird erzwungen, dass Gesteninstanzen immer von mindestens zwei Nichtgesteninstanzen getrennt sind.

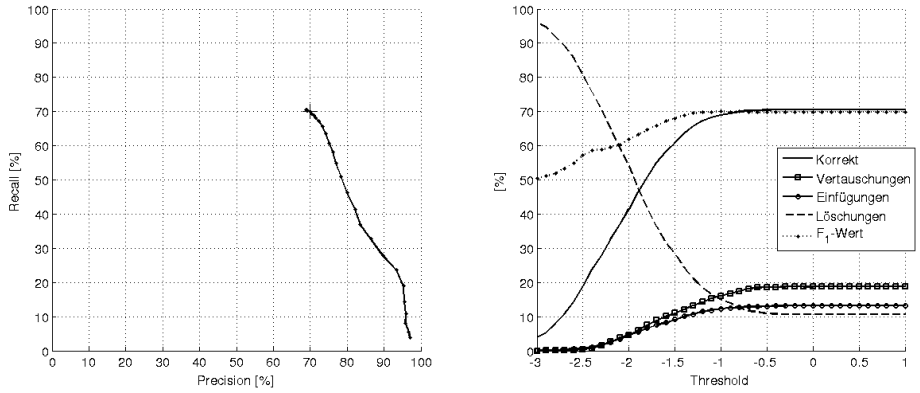


Abbildung 54: Segmentierungsexperiment: *Precision-Recall* Kurve (links) und K/V/L/E Plot (rechts) für Modell eHMM2.

Das führt in erster Linie bei sehr kurzen Gesteninstanzen zu Problemen, weil diese übersprungen werden können.

Tatsächlich wird der negative Einfluß auf die Detektionsrate umso größer, je mehr Zustände das Rückweisungsmodell hat. Im Gegensatz zum strukturell uneingeschränkten Modell, bei dem der Einfluß der Rückweisungsmodelllänge vernachlässigbar war, verschlechtern sich die Ergebnisse dramatisch, wenn zusätzliche Zustände eingefügt werden (Tabelle 20). Experimente mit anderen Parametersätzen (vgl. Anhang A.8, Tabelle 46) bestätigen dies. Obwohl mit einer guten Parameterwahl leichte Verbesserungen (im Sinne des $F_1$-Wertes) erzielt werden konnten, scheint die Einführung starker struktureller Vorgaben im Allgemeinen also nicht empfehlenswert zu sein.

| Rückweisungsmodelllänge | 1 | 3 | 6 | 8 |
|---|---|---|---|---|
| $F_1$-Wert (*Precision, Recall*) | 68.2 (66.3, 70.2) | 70.0 (69.9, 70.0) | 60.9 (72.3, 52.7) | 52.5 (71.2, 41.6) |
| max. *Recall* | 74.7 | 70.6 | 52.9 | 42.1 |
| $F_1$-Wert (*Precision, Recall*) | 65.8 (62.5, 69.4) | 67.4 (64.5, 70.5) | 64.4 (60.8, 68.5) | 65.6 (59.9, 72.4) |
| max. *Recall* | 76.9 | 78.5 | 76.0 | 76.9 |

Tabelle 20: Einfluß der Zustandsanzahl im Rückweisungsmodell. Oben: Modell eHMM2. Unten: Uneingeschränkte parallele Dekodierung zum Vergleich. Alle Werte sind in %. Die Konfidenzintervalle bewegen sich zwischen 1.3 und 1.8, sind aber aus Platz- und Lesbarkeitsgründen nicht einzeln angegeben.

# 8 FAZIT

## 8.1 ZUSAMMENFASSUNG

Diese Arbeit hatte das Ziel, anhand von Methoden der *Computer Vision*, der Mustererkennung und des maschinellen Lernens eine berührungslose und nutzerunabhängige visuelle Klassifikation bestimmter Armgesten anhand ihrer räumlich-zeitlichen Bewegungsmuster zu realisieren. Das Anwendungsszenario war hierbei ein intelligenter Konferenzraum der mit mehreren handelsüblichen aktiven Kameras ausgerüstet war.

Dieses Szenario stellte aus drei Gründen eine besondere Herausforderung dar: Für eine möglichst intuitive Interaktion war es erstens notwendig, die Erkennung so zu realisieren, dass sie unabhängig von der Position und Orientierung des Nutzers im Raum ist. Somit wurden vereinfachende Annahmen bezüglich der relativen Positionen von Nutzer und Kamera von Anfang an ausgeschlossen. Zweitens stellte die Verwendung eines nicht synchronisierten Multikamerasystems eine Neuerung dar, die dazu führte, dass während der 3D-Rekonstruktion der Hypothesen aus verschiedenen Kamerabildern besonderes Augenmerk auf den Umgang mit dem auftretenden zeitlichen Versatz gelegt werden musste. Dies hatte indirekt auch Folgen für die Klassifikationsaufgabe, weil in den rekonstruierten 3D-Trajektorien mit entsprechenden Ungenauigkeiten zu rechnen war. Drittens wurde explizit von aktiven Kameras ausgegangen, deren extrinsische und intrinsische Parameter sich zur Laufzeit ändern können. Hierdurch ergab sich einerseits die Notwendigkeit der Nachführung der Kalibration zur Laufzeit, andererseits mussten die verwendeten *Computer Vision* Methoden so ausgelegt werden, dass sie sich an derartige Änderungen adaptieren können.

Zusätzlich wurde die Reaktivität der verwendeten Verfahren als wichtiges Kriterium identifiziert, weil die Akzeptanz einer gestenbasierten Mensch-Maschine-Schnittstelle durch große Latenzen gefährdet wäre. Bei der Auswahl und Konzeption geeigneter Verfahren wurden daher immer auch deren Komplexität und inhärente Latenzen berücksichtigt. Das führte insbesondere zu einer parallelen Verarbeitungsstruktur, in der die verschiedenen Kameradatenströme getrennt verarbeitet und die Einzelergebnisse anschließend kombiniert wurden.

Das realisierte Gestenerkennungssystem umfasst die Verarbeitungsschritte Personendetektion, Personentracking, Handdetektion, Aggregation und 3D-Kombination

der 2D-Hypothesen unterschiedlicher Kameras, sowie trajektorienbasierte Gestenklassifikation. Zusätzlich wurde auf einfache Möglichkeiten zur Repräsentation des Umgebungskontextes eingegangen.

Für die Personendetektion wurde ein auf HOG-Merkmalen und einem MLP-Klassifikator basierender Detektor für Kopf-Schulter-Regionen realisiert. Zum Zwecke der Suchraumeinschränkung wurden verschiedene Möglichkeiten der adaptiven Hintergrundmodellierung untersucht.

Die Detektionsergebnisse dienen als Eingabe für einen adaptiven histogrammbasierten *Mean Shift Tracker*. Hierbei wurde gezeigt, wie durch eine enge Wechselwirkung mit dem Detektor eine Detektion von *Tracker*versagen möglich ist.

Anschließend wurden verschiedene Möglichkeiten zur Detektion von Händen in Innenraumszenarien untersucht. Insbesondere wurde eine Methode entwickelt, die ein zur Laufzeit gelerntes adaptives personalisiertes Hautfarbmodell mit einem statisch trainierten Modell in einem selektiven Adaptionsansatz kombiniert, um eine hohe Robustheit gegenüber Änderungen der Umgebungsbedingungen zu erreichen.

Die 2D-Punkthypothesen für Kopf- und Handpositionen wurden dann zu 3D-Hypothesen kombiniert. Dabei wurde ein Rekonstruktionsverfahren realisiert, dass mit dem zeitlichen Versatz der unsynchronisierten Ergebnisse der einzelnen Bildströme umgehen kann. Zur Bewertung der resultierenden 3D-Punkthypothesen und zur Rückweisung fehlerhafter Hypothesen wurden verschiedene probabilistisch modellierte Maße entwickelt. Weiterhin wurden zwei verschiedene Ansätze für die anschließende Aggregation zu räumlich-zeitlichen Trajektorien erarbeitet.

Abschließend wurde ein Klassifikationsansatz für dreidimensionale Gestentrajektorien auf der Basis von Hidden Markov Modellen vorgestellt. In diesem Zusammenhang wurden verschiedene alternative Merkmalsrepräsentationen und Normalisierungsansätze untersucht. Für den speziellen Fall einer Zeigegeste wurde außerdem eine probabilistische Modellierung als unscharfer Kegel eingeführt, welche die explizite Verfolgung einer 3D-Zeigerichtung durch die Szene ermöglicht.

Die vorgestellten Ansätze wurden anhand realistischer Datensätze ausführlich evaluiert. Dabei konnten sowohl für die Personendetektion als auch für die Handdetektion sehr gute Ergebnisse erzielt werden. Auch der Klassifikationsansatz mittels HMM erwies sich als sehr leistungsfähig und erreichte Klassifikationsraten von annähernd 90% für neun verschiedene Gesten. Die Ergebnisse für die Segmentierung kontinuierlicher Beobachtungssequenzen lassen indes Raum für Verbesserungen. Ein Grund hierfür ist die Nichtverfügbarkeit adäquater Daten zum Training eines Rückweisungsmodelles.

Abschließend kann gesagt werden, dass die Forschungsziele weitestgehend erreicht und sehr ermutigende Ergebnisse erzielt wurden. Es wurde gezeigt, dass die Kombination einfacher, effizienter Methoden der *Computer Vision* die Realisierung zuverlässiger

Erkennungssysteme in realistischen und wenig eingeschränkten Innenraumszenarien erlaubt. Die prototypische C++ Implementierung der 2D-Verarbeitungspipeline erreicht auf handelsüblicher Hardware bereits Verarbeitungsgeschwindigkeiten von etwa acht Bildern pro Sekunde. Das lässt vermuten, dass mit einer auf Effizienz optimierten Umsetzung die notwendige Reaktivität erreicht werden kann. Die MATLAB Implementierung der 3D-Verarbeitung und die Realisierung des Klassifikationsframeworks erfolgte offline und konnte somit nicht integriert getestet werden. Weil dieser Verarbeitungsteil aber nur noch auf einigen wenigen Punkthypothesen operiert, sollte eine effiziente Umsetzung möglich sein.

## 8.2 AUSBLICK

Für die Zukunft muss in erster Linie eine vollständige, integrierte Umsetzung der erarbeiteten Methoden zu einem lauffähigen Prototypen erfolgen. Hierbei sind vor allem im Bereich der 3D-Kombination und Trajektorienaggregation noch einige Probleme zu lösen, die in dieser Arbeit nur am Rande behandelt werden konnten. Insbesondere ist eine algorithmische Umsetzung der theoretischen Überlegungen zur 3D-Trajektorienaggregation notwendig. Die Anwendbarkeit und Akzeptanz einer derartigen gestenbasierten Mensch-Maschine-Schnittstelle muss dann im Rahmen einer ausführlichen Nutzerstudie evaluiert werden.

Diese Arbeit beschränkte sich auf die Verarbeitung von Bilddaten. Selbstverständlich stehen innerhalb einer intelligenten Umgebung weitere Sensormodalitäten – beispielsweise Audiodaten oder Bewegungssensoren – zur Verfügung, deren Integration zu einer erhöhten Robustheit führen kann. So kann z.B. eine Lokalisierung mit Audiodaten helfen, fehlerhafte Lokalisierungshypothesen zu verwerfen, und eine Analyse gestenbegleitender Sprachkommandos kann Fehldetektionen und Mehrdeutigkeiten vermeiden. Die Integration verschiedener Modalitäten muss das endgültige Ziel sein, um ein wirklich robustes System zu erreichen.

Weiterhin wurde die Modellierung und Einbeziehung von Kontextwissen – insbesondere des Umgebungskontextes – nur am Rande betrachtet und beschränkte sich auf eine sehr einfache Szenenbeschreibung und die Modellierung einer Zeigerichtung. Dieser einfache Ansatz war jedoch bereits ausreichend, um angezeigte Referenzen auf Objekte in der Szene zuverlässig zu identifizieren. An dieser Stelle muss zusätzlicher Forschungsaufwand betrieben werden, um die Möglichkeiten, die sich durch Auswertung des Umgebungskontextes ergeben, in größerem Maße nutzen zu können. Insbesondere muss der Schritt von vordefinierten Szene- und Aktionsmodellen hin zu

automatisch gelernten, flexibel erweiterbaren Modellen erfolgen. In diesem Rahmen findet sich noch eine große Vielfalt spannender und ungelöster Forschungsthemen.

Im Laufe des Jahres 2010 wurde zudem durch Projekte wie *Microsoft Natal/Kinect* [121] und *LightSpace* [209] oder *Intel OASIS* [82] die Entwicklung der Videosensorik, insbesondere von Kameras mit integrierter Tiefenmessung, rapide vorangetrieben. Derartige Kameras werden in naher Zukunft in weitaus besserer Qualität und zu konkurrenzfähigen Preisen verfügbar sein. Viele schwierige Probleme der *Computer Vision* – z.B. *Tracking* eines 3D-Körpermodelles in Echtzeit – lassen sich mit dieser Art von Sensorik einfacher und effizienter lösen. Solche „Tiefenkameras“ eröffnen somit viele neue Möglichkeiten im Bereich der Mensch-Maschine-Interaktion, die sie auch für den Einsatz in vorliegendem Gestenerkennungs-Szenario attraktiv machen.

# A

# ANHANG

## A.1 ERGEBNISTABELLEN HINTERGRUNDMODELLIERUNG

| | $\lambda$ | $\alpha_1$ | $\alpha_2$ | Det.rate [%] | FD/Bild | BPS | Zeit/Bild $\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| Referenz | – | – | – | 98.01 | 1.47 | 10.91 | 91.69±6.02 |
| Exp | 0.1 | 0.3 | 1.0 | 97.38±0.28 | 0.64 | 12.33 | 81.07±12.62 |
| Exp | 0.3 | 0.3 | 1.0 | 87.99±0.56 | 0.54 | 13.08 | 76.48±10.78 |
| Exp | 0.5 | 0.3 | 1.0 | 76.05±0.72 | 0.50 | 13.45 | 74.37±10.22 |
| Exp | 0.7 | 0.3 | 1.0 | 65.36±0.80 | 0.45 | 13.75 | 72.72±9.68 |
| Exp | 0.9 | 0.3 | 1.0 | 55.59±0.83 | 0.41 | 13.93 | 71.79±9.32 |
| Exp | 0.1 | 0.6 | 1.0 | 97.77±0.26 | 0.66 | 12.11 | 82.57±12.22 |
| Exp | 0.3 | 0.6 | 1.0 | 92.17±0.46 | 0.56 | 12.81 | 78.04±11.21 |
| Exp | 0.5 | 0.6 | 1.0 | 84.69±0.61 | 0.52 | 13.27 | 75.37±10.28 |
| Exp | 0.7 | 0.6 | 1.0 | 77.47±0.71 | 0.50 | 13.54 | 73.87±10.03 |
| Exp | 0.9 | 0.6 | 1.0 | 71.19±0.77 | 0.48 | 13.57 | 73.68±9.75 |
| Exp | 0.1 | 0.79 | 1.0 | 97.83±0.26 | 0.67 | 11.98 | 83.46±12.31 |
| Exp | 0.3 | 0.79 | 1.0 | 95.31±0.37 | 0.59 | 12.62 | 79.22±10.73 |
| Exp | 0.5 | 0.79 | 1.0 | 92.51±0.45 | 0.56 | 13.01 | 76.87±10.58 |
| Exp | 0.7 | 0.79 | 1.0 | 89.38±0.53 | 0.55 | 13.11 | 76.29±10.45 |
| Exp | 0.9 | 0.79 | 1.0 | 86.73±0.58 | 0.54 | 13.17 | 75.93±10.45 |
| Exp | 0.1 | 0.85 | 1.0 | 98.02±0.25 | 0.68 | 11.77 | 84.98±12.62 |
| Exp | 0.3 | 0.85 | 1.0 | 96.64±0.32 | 0.61 | 12.47 | 80.19±10.90 |
| Exp | 0.5 | 0.85 | 1.0 | 95.12±0.37 | 0.59 | 12.66 | 78.99±10.68 |
| Exp | 0.7 | 0.85 | 1.0 | 93.71±0.42 | 0.57 | 12.79 | 78.20±10.67 |
| Exp | 0.9 | 0.85 | 1.0 | 92.40±0.46 | 0.56 | 13.09 | 76.39±10.83 |

Tabelle 21: Ergebnisse der Evaluation des Hintergrundmodelles mit einer asymptotischen Exponentialfunktion als Verfallsfunktion. Verwendeter Farbraum: Graustufen.

| | $\lambda$ | $\alpha_1$ | $\alpha_2$ | Det.rate [%] | FD/Bild | BPS | Zeit/Bild $\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| Referenz | – | – | – | 98.01 | 1.47 | 10.91 | 91.69±6.02 |
| Sigmoid | 0.1 | 0.3 | 7.5 | 98.06±0.25 | 0.69 | 12.37 | 80.84±11.80 |
| Sigmoid | 0.3 | 0.3 | 7.5 | 97.22±0.29 | 0.63 | 13.02 | 76.81±9.95 |
| Sigmoid | 0.5 | 0.3 | 7.5 | 96.56±0.32 | 0.61 | 13.28 | 75.31±9.70 |
| Sigmoid | 0.7 | 0.3 | 7.5 | 95.87±0.35 | 0.60 | 13.37 | 74.79±9.74 |
| Sigmoid | 0.9 | 0.3 | 7.5 | 95.23±0.37 | 0.59 | 13.41 | 74.55±9.66 |
| Sigmoid | 0.1 | 0.45 | 5.0 | 97.95±0.25 | 0.68 | 12.54 | 79.78±11.25 |
| Sigmoid | 0.3 | 0.45 | 5.0 | 96.34±0.33 | 0.61 | 13.37 | 74.78±9.65 |
| Sigmoid | 0.5 | 0.45 | 5.0 | 94.81±0.39 | 0.58 | 13.40 | 74.64±9.40 |
| Sigmoid | 0.7 | 0.45 | 5.0 | 93.22±0.43 | 0.57 | 13.68 | 73.10±9.41 |
| Sigmoid | 0.9 | 0.45 | 5.0 | 91.77±0.47 | 0.56 | 13.82 | 72.37±9.47 |
| Sigmoid | 0.1 | 0.9 | 2.5 | 97.82±0.26 | 0.66 | 12.75 | 78.45±11.19 |
| Sigmoid | 0.3 | 0.9 | 2.5 | 93.89±0.41 | 0.58 | 13.76 | 72.69±9.23 |
| Sigmoid | 0.5 | 0.9 | 2.5 | 88.77±0.54 | 0.54 | 14.14 | 70.74±9.45 |
| Sigmoid | 0.7 | 0.9 | 2.5 | 84.10±0.62 | 0.53 | 14.37 | 69.59±9.02 |
| Sigmoid | 0.9 | 0.9 | 2.5 | 79.76±0.68 | 0.51 | 14.38 | 69.52±8.91 |
| Sigmoid | 0.1 | 2.2 | 1.0 | 97.46±0.28 | 0.64 | 12.89 | 77.59±11.10 |
| Sigmoid | 0.3 | 2.2 | 1.0 | 89.16±0.53 | 0.55 | 13.99 | 71.48±9.29 |
| Sigmoid | 0.5 | 2.2 | 1.0 | 78.29±0.70 | 0.51 | 14.40 | 69.44±8.75 |
| Sigmoid | 0.7 | 2.2 | 1.0 | 68.65±0.78 | 0.47 | 14.78 | 67.65±8.21 |
| Sigmoid | 0.9 | 2.2 | 1.0 | 60.46±0.82 | 0.44 | 14.81 | 67.53±7.92 |

Tabelle 22: Ergebnisse der Evaluation des Hintergrundmodelles mit einer Sigmoidfunktion als Verfallsfunktion. Verwendeter Farbraum: Graustufen.

| | $\lambda$ | $\alpha_1$ | $\alpha_2$ | Det.rate [%] | FD/Bild | BPS | Zeit/Bild $\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| Referenz | – | – | – | 98.01 | 1.47 | 10.91 | 91.69±6.02 |
| Rampe | 0.1 | 1 | 2 | 97.74±0.26 | 0.65 | 12.79 | 78.17±10.87 |
| Rampe | 0.3 | 1 | 2 | 91.95±0.47 | 0.56 | 13.86 | 72.16±9.35 |
| Rampe | 0.5 | 1 | 2 | 84.52±0.62 | 0.52 | 14.33 | 69.78±8.99 |
| Rampe | 0.7 | 1 | 2 | 77.54±0.71 | 0.50 | 14.45 | 69.23±8.76 |
| Rampe | 0.9 | 1 | 2 | 71.03±0.77 | 0.47 | 14.71 | 67.98±8.39 |
| Rampe | 0.1 | 3 | 5 | 97.88±0.26 | 0.68 | 12.59 | 79.45±11.29 |
| Rampe | 0.3 | 3 | 5 | 95.81±0.35 | 0.60 | 13.43 | 74.47±9.49 |
| Rampe | 0.5 | 3 | 5 | 93.46±0.43 | 0.57 | 13.69 | 73.04±9.88 |
| Rampe | 0.7 | 3 | 5 | 91.25±0.49 | 0.56 | 13.93 | 71.78±9.25 |
| Rampe | 0.9 | 3 | 5 | 89.16±0.53 | 0.55 | 14.10 | 70.91±9.05 |
| Rampe | 0.1 | 5 | 10 | 98.16±0.24 | 0.69 | 12.31 | 81.22±11.56 |
| Rampe | 0.3 | 5 | 10 | 97.14±0.29 | 0.63 | 13.14 | 76.11±9.45 |
| Rampe | 0.5 | 5 | 10 | 96.40±0.33 | 0.61 | 13.31 | 75.10±9.38 |
| Rampe | 0.7 | 5 | 10 | 95.65±0.36 | 0.60 | 13.41 | 74.58±9.49 |
| Rampe | 0.9 | 5 | 10 | 95.10±0.38 | 0.59 | 13.49 | 74.14±9.49 |
| Rampe | 0.1 | 8 | 15 | 98.23±0.24 | 0.71 | 12.07 | 82.84±12.41 |
| Rampe | 0.3 | 8 | 15 | 97.69±0.27 | 0.66 | 12.90 | 77.55±9.76 |
| Rampe | 0.5 | 8 | 15 | 97.36±0.28 | 0.64 | 13.00 | 76.91±9.65 |
| Rampe | 0.7 | 8 | 15 | 97.07±0.30 | 0.63 | 13.11 | 76.28±9.58 |
| Rampe | 0.9 | 8 | 15 | 96.86±0.31 | 0.63 | 13.12 | 76.24±9.55 |

Tabelle 23: Ergebnisse der Evaluation des Hintergrundmodelles mit einer Rampenfunktion als Verfallsfunktion. Verwendeter Farbraum: Graustufen.

## A.2 ERGEBNISTABELLEN PERSONENTRACKING

| $\beta_0$ | $\beta_m$ | Det.rate [%] | | FD/Bild | | BPS | | Zeit/Bild $\pm\sigma$ [ms] | |
|---|---|---|---|---|---|---|---|---|---|
| Referenz | | 91.56±0.48 | | 0.06 | | 13.09 | | 76.39±9.95 | |
| Farbraum | | RGB | HSV | RGB | HSV | RGB | HSV | RGB | HSV |
| 0.0 | 0.1 | 92.11±0.46 | 90.68±0.50 | 0.06 | 0.08 | 8.44 | 8.25 | 118.44±17.72 | 121.15±14.18 |
| 0.0 | 0.2 | 90.91±0.49 | 87.69±0.56 | 0.08 | 0.11 | 8.90 | 8.27 | 112.35±12.61 | 120.98±12.75 |
| 0.0 | 0.3 | 90.80±0.50 | 85.28±0.60 | 0.08 | 0.14 | 8.90 | 8.23 | 112.41±12.24 | 121.49±12.43 |
| 0.0 | 0.4 | 92.30±0.46 | 87.11±0.57 | 0.07 | 0.12 | 8.97 | 8.14 | 111.52±10.20 | 122.90±12.71 |
| 0.0 | 0.5 | 90.59±0.50 | 87.14±0.57 | 0.09 | 0.12 | 8.94 | 8.17 | 111.82±10.35 | 122.39±13.18 |
| 0.2 | 0.1 | 92.05±0.47 | 89.58±0.52 | 0.06 | 0.09 | 8.98 | 8.42 | 111.33±12.37 | 118.82±12.38 |
| 0.2 | 0.2 | 90.44±0.50 | 87.09±0.57 | 0.09 | 0.12 | 8.86 | 8.57 | 112.86±11.57 | 116.67±11.21 |
| 0.2 | 0.3 | 90.12±0.51 | 84.57±0.62 | 0.09 | 0.15 | 8.78 | 8.47 | 113.91±14.14 | 118.10±10.83 |
| 0.2 | 0.4 | 91.44±0.48 | 85.44±0.60 | 0.08 | 0.14 | 8.86 | 8.44 | 112.81±12.22 | 118.50±10.93 |
| 0.2 | 0.5 | 90.54±0.50 | 85.88±0.59 | 0.09 | 0.13 | 8.72 | 8.42 | 114.73±15.27 | 118.79±10.84 |
| 0.4 | 0.1 | 87.51±0.57 | 84.94±0.61 | 0.11 | 0.14 | 8.83 | 8.38 | 113.26±13.05 | 119.30±12.29 |
| 0.4 | 0.2 | 90.05±0.51 | 84.68±0.61 | 0.09 | 0.14 | 8.82 | 8.39 | 113.37±12.89 | 119.15±11.47 |
| 0.4 | 0.3 | 89.79±0.52 | 82.50±0.65 | 0.09 | 0.17 | 8.99 | 8.51 | 111.25±10.62 | 117.52±11.17 |
| 0.4 | 0.4 | 90.86±0.50 | 83.51±0.63 | 0.08 | 0.16 | 8.96 | 8.43 | 111.56±11.17 | 118.58±11.05 |
| 0.4 | 0.5 | 90.61±0.50 | 83.55±0.63 | 0.09 | 0.16 | 9.00 | 8.53 | 111.05±10.79 | 117.19±10.48 |
| 0.6 | 0.1 | 87.58±0.56 | 82.98±0.64 | 0.11 | 0.16 | 8.91 | 8.42 | 112.23±12.49 | 118.72±11.54 |
| 0.6 | 0.2 | 90.39±0.51 | 83.19±0.64 | 0.09 | 0.16 | 8.94 | 8.45 | 111.90±11.69 | 118.30±11.62 |
| 0.6 | 0.3 | 84.42±0.62 | 81.06±0.67 | 0.15 | 0.18 | 9.01 | 8.51 | 111.00±11.26 | 117.50±10.96 |
| 0.6 | 0.4 | 88.55±0.55 | 80.20±0.68 | 0.11 | 0.19 | 9.03 | 8.48 | 110.79±10.59 | 117.95±10.51 |
| 0.6 | 0.5 | 89.82±0.52 | 80.67±0.67 | 0.10 | 0.19 | 9.06 | 8.50 | 110.33±9.81 | 117.71±10.73 |
| 0.8 | 0.1 | 87.11±0.57 | 84.46±0.62 | 0.12 | 0.15 | 9.04 | 8.43 | 110.67±11.20 | 118.67±11.42 |
| 0.8 | 0.2 | 88.29±0.55 | 81.76±0.66 | 0.11 | 0.17 | 9.02 | 8.44 | 110.81±10.30 | 118.46±10.99 |
| 0.8 | 0.3 | 89.08±0.53 | 79.94±0.68 | 0.10 | 0.19 | 8.96 | 8.47 | 111.61±10.53 | 118.02±11.66 |
| 0.8 | 0.4 | 89.87±0.52 | 79.47±0.69 | 0.10 | 0.20 | 9.07 | 8.54 | 110.30±9.94 | 117.12±10.56 |
| 0.8 | 0.5 | 90.81±0.50 | 81.83±0.66 | 0.09 | 0.18 | 9.04 | 8.54 | 110.67±9.86 | 117.09±10.43 |
| 1.0 | 0.1 | 87.52±0.57 | 82.70±0.64 | 0.12 | 0.16 | 9.08 | 8.55 | 110.14±10.70 | 116.91±10.97 |
| 1.0 | 0.2 | 86.94±0.58 | 80.17±0.68 | 0.12 | 0.19 | 9.00 | 8.42 | 111.15±10.30 | 118.81±10.87 |
| 1.0 | 0.3 | 90.04±0.51 | 79.41±0.69 | 0.09 | 0.20 | 9.02 | 8.39 | 110.91±10.23 | 119.20±10.93 |
| 1.0 | 0.4 | 89.31±0.53 | 79.44±0.69 | 0.10 | 0.20 | 9.02 | 8.55 | 110.83±9.99 | 117.02±10.56 |
| 1.0 | 0.5 | 90.33±0.51 | 80.61±0.67 | 0.09 | 0.19 | 9.13 | 8.58 | 109.47±9.10 | 116.49±9.98 |

Tabelle 24: Ergebnisse der Evaluation des Trackers. Hintergrundmodell mit Exponentialfunktion ($\lambda = 0.3$, $\alpha_1 = 0.79$) als Verfallsfunktion.

| $\beta_0$ | $\beta_m$ | Det.rate [%] | | FD/Bild | | BPS | | Zeit/Bild $\pm\sigma$ [ms] | |
|---|---|---|---|---|---|---|---|---|---|
| Referenz | | 92.83±0.44 | | 0.06 | | 12.77 | | 78.31±10.10 | |
| Farbraum | | RGB | HSV | RGB | HSV | RGB | HSV | RGB | HSV |
| 0.0 | 0.1 | 92.62±0.45 | 91.50±0.48 | 0.06 | 0.07 | 9.08 | 8.78 | 110.19±10.97 | 113.83±10.77 |
| 0.0 | 0.2 | 90.38±0.51 | 86.73±0.58 | 0.09 | 0.13 | 9.29 | 8.79 | 107.61±9.61 | 113.75±9.81 |
| 0.0 | 0.3 | 89.54±0.52 | 84.20±0.62 | 0.10 | 0.15 | 9.35 | 8.81 | 106.91±9.29 | 113.53±9.46 |
| 0.0 | 0.4 | 89.82±0.52 | 85.67±0.60 | 0.10 | 0.14 | 9.31 | 8.84 | 107.40±8.85 | 113.13±9.22 |
| 0.0 | 0.5 | 89.79±0.52 | 86.35±0.59 | 0.10 | 0.13 | 9.33 | 8.81 | 107.16±8.89 | 113.57±9.59 |
| 0.2 | 0.1 | 93.24±0.43 | 91.30±0.48 | 0.06 | 0.08 | 9.24 | 8.74 | 108.27±10.41 | 114.45±10.97 |
| 0.2 | 0.2 | 90.41±0.51 | 84.78±0.61 | 0.09 | 0.14 | 9.34 | 8.87 | 107.02±9.23 | 112.74±9.45 |
| 0.2 | 0.3 | 90.25±0.51 | 82.76±0.64 | 0.09 | 0.17 | 9.39 | 8.86 | 106.45±9.24 | 112.85±9.28 |
| 0.2 | 0.4 | 90.16±0.51 | 82.67±0.64 | 0.09 | 0.17 | 9.33 | 8.94 | 107.23±8.92 | 111.89±9.45 |
| 0.2 | 0.5 | 91.22±0.49 | 83.40±0.63 | 0.08 | 0.16 | 9.35 | 8.82 | 106.97±8.82 | 113.34±9.18 |
| 0.4 | 0.1 | 87.28±0.57 | 85.10±0.61 | 0.12 | 0.14 | 9.29 | 8.86 | 107.67±10.78 | 112.92±11.08 |
| 0.4 | 0.2 | 91.52±0.48 | 82.53±0.65 | 0.08 | 0.17 | 9.36 | 8.85 | 106.81±9.18 | 112.98±9.75 |
| 0.4 | 0.3 | 88.44±0.55 | 81.71±0.66 | 0.11 | 0.18 | 9.35 | 8.85 | 106.96±9.15 | 113.00±9.39 |
| 0.4 | 0.4 | 89.22±0.53 | 82.36±0.65 | 0.10 | 0.17 | 9.40 | 8.90 | 106.36±8.82 | 112.42±9.93 |
| 0.4 | 0.5 | 91.44±0.48 | 81.96±0.65 | 0.08 | 0.17 | 9.37 | 8.89 | 106.75±8.78 | 112.48±9.12 |
| 0.6 | 0.1 | 89.47±0.53 | 82.84±0.64 | 0.10 | 0.17 | 9.32 | 8.95 | 107.31±9.60 | 111.79±10.25 |
| 0.6 | 0.2 | 85.39±0.60 | 80.59±0.67 | 0.14 | 0.19 | 9.35 | 8.85 | 106.93±9.43 | 113.00±9.66 |
| 0.6 | 0.3 | 83.30±0.64 | 80.62±0.67 | 0.16 | 0.19 | 9.45 | 8.77 | 105.82±9.03 | 113.98±9.45 |
| 0.6 | 0.4 | 89.69±0.52 | 78.83±0.69 | 0.10 | 0.21 | 9.37 | 8.89 | 106.75±8.86 | 112.47±9.01 |
| 0.6 | 0.5 | 90.49±0.50 | 81.22±0.66 | 0.09 | 0.18 | 9.32 | 8.83 | 107.26±8.92 | 113.31±9.11 |
| 0.8 | 0.1 | 85.22±0.61 | 82.05±0.65 | 0.14 | 0.17 | 9.30 | 8.96 | 107.55±9.83 | 111.62±9.42 |
| 0.8 | 0.2 | 85.58±0.60 | 79.19±0.69 | 0.14 | 0.20 | 9.39 | 8.85 | 106.54±9.03 | 113.01±9.42 |
| 0.8 | 0.3 | 88.31±0.55 | 77.53±0.71 | 0.11 | 0.22 | 9.33 | 8.78 | 107.15±8.96 | 113.96±9.54 |
| 0.8 | 0.4 | 88.70±0.54 | 77.96±0.70 | 0.11 | 0.22 | 9.35 | 8.98 | 106.90±8.97 | 111.33±9.19 |
| 0.8 | 0.5 | 88.57±0.54 | 78.18±0.70 | 0.11 | 0.21 | 9.34 | 9.08 | 107.07±8.80 | 110.18±8.90 |
| 1.0 | 0.1 | 87.91±0.56 | 81.83±0.66 | 0.12 | 0.18 | 9.40 | 8.83 | 106.34±9.21 | 113.22±9.53 |
| 1.0 | 0.2 | 87.31±0.57 | 78.57±0.70 | 0.12 | 0.21 | 9.34 | 8.80 | 107.04±8.96 | 113.59±9.32 |
| 1.0 | 0.3 | 84.06±0.62 | 79.09±0.69 | 0.16 | 0.20 | 9.35 | 8.89 | 107.00±9.05 | 112.44±9.38 |
| 1.0 | 0.4 | 88.82±0.54 | 78.96±0.69 | 0.11 | 0.21 | 9.37 | 8.80 | 106.67±8.83 | 113.59±9.06 |
| 1.0 | 0.5 | 87.84±0.56 | 78.43±0.70 | 0.12 | 0.21 | 9.36 | 8.86 | 106.78±8.82 | 112.89±8.67 |

Tabelle 25: Ergebnisse der Evaluation des Trackers. Hintergrundmodell mit Sigmoidfunktion ($\lambda = 0.3$, $\alpha_1 = 0.45$, $\alpha_2 = 5.0$) als Verfallsfunktion.

| $\beta_0$ | $\beta_m$ | Det.rate [%] | | FD/Bild | | BPS | | Zeit/Bild $\pm\sigma$ [ms] | |
|---|---|---|---|---|---|---|---|---|---|
| Referenz | | 92.10±0.46 | | 0.06 | | 12.98 | | 77.03±9.91 | |
| Farbraum | | RGB | HSV | RGB | HSV | RGB | HSV | RGB | HSV |
| 0.0 | 0.1 | 92.82±0.45 | 90.81±0.50 | 0.06 | 0.08 | 9.03 | 8.69 | 110.72±11.03 | 115.09±10.99 |
| 0.0 | 0.2 | 91.20±0.49 | 87.15±0.57 | 0.08 | 0.12 | 9.09 | 8.70 | 110.00±9.74 | 114.88±9.92 |
| 0.0 | 0.3 | 89.34±0.53 | 85.39±0.60 | 0.10 | 0.14 | 9.25 | 8.78 | 108.12±9.58 | 113.85±9.93 |
| 0.0 | 0.4 | 92.08±0.46 | 84.68±0.61 | 0.07 | 0.15 | 9.16 | 8.86 | 109.17±8.96 | 112.80±9.21 |
| 0.0 | 0.5 | 91.33±0.48 | 84.65±0.61 | 0.08 | 0.15 | 9.26 | 8.76 | 108.01±8.85 | 114.10±9.51 |
| 0.2 | 0.1 | 92.53±0.45 | 91.08±0.49 | 0.06 | 0.07 | 9.11 | 8.68 | 109.79±10.55 | 115.20±11.29 |
| 0.2 | 0.2 | 90.66±0.50 | 84.98±0.61 | 0.08 | 0.14 | 9.22 | 8.82 | 108.51±9.64 | 113.42±9.41 |
| 0.2 | 0.3 | 90.06±0.51 | 82.45±0.65 | 0.09 | 0.17 | 9.16 | 8.78 | 109.17±9.22 | 113.85±9.50 |
| 0.2 | 0.4 | 91.25±0.49 | 84.26±0.62 | 0.08 | 0.15 | 9.22 | 8.79 | 108.47±8.70 | 113.77±9.24 |
| 0.2 | 0.5 | 91.11±0.49 | 83.42±0.63 | 0.08 | 0.16 | 9.15 | 8.77 | 109.25±8.83 | 114.07±9.27 |
| 0.4 | 0.1 | 88.08±0.55 | 85.72±0.60 | 0.11 | 0.13 | 9.12 | 8.80 | 109.67±11.05 | 113.58±11.30 |
| 0.4 | 0.2 | 90.43±0.51 | 83.09±0.64 | 0.09 | 0.16 | 9.16 | 8.78 | 109.18±9.46 | 113.89±10.34 |
| 0.4 | 0.3 | 90.25±0.51 | 82.56±0.65 | 0.09 | 0.17 | 9.18 | 8.79 | 108.88±9.32 | 113.76±9.69 |
| 0.4 | 0.4 | 92.03±0.47 | 83.23±0.64 | 0.07 | 0.16 | 9.13 | 8.88 | 109.47±8.86 | 112.60±9.68 |
| 0.4 | 0.5 | 90.82±0.50 | 82.20±0.65 | 0.09 | 0.17 | 9.19 | 8.82 | 108.78±8.91 | 113.37±9.41 |
| 0.6 | 0.1 | 88.10±0.55 | 83.06±0.64 | 0.11 | 0.16 | 9.21 | 8.70 | 108.52±10.31 | 114.90±10.72 |
| 0.6 | 0.2 | 85.52±0.60 | 78.91±0.69 | 0.14 | 0.20 | 9.19 | 8.85 | 108.85±9.31 | 113.05±9.64 |
| 0.6 | 0.3 | 84.11±0.62 | 82.13±0.65 | 0.15 | 0.17 | 9.17 | 8.81 | 109.01±9.21 | 113.53±9.57 |
| 0.6 | 0.4 | 92.21±0.46 | 81.98±0.65 | 0.07 | 0.17 | 9.21 | 8.86 | 108.59±9.12 | 112.90±9.23 |
| 0.6 | 0.5 | 89.33±0.53 | 82.66±0.64 | 0.10 | 0.17 | 9.16 | 8.97 | 109.14±9.46 | 111.51±9.07 |
| 0.8 | 0.1 | 88.16±0.55 | 84.05±0.62 | 0.11 | 0.15 | 9.20 | 8.77 | 108.75±9.62 | 113.99±10.07 |
| 0.8 | 0.2 | 85.69±0.60 | 79.52±0.69 | 0.14 | 0.20 | 9.22 | 8.63 | 108.48±9.22 | 115.89±9.55 |
| 0.8 | 0.3 | 85.69±0.60 | 80.50±0.67 | 0.14 | 0.19 | 9.19 | 8.84 | 108.87±9.52 | 113.11±10.02 |
| 0.8 | 0.4 | 91.81±0.47 | 80.65±0.67 | 0.08 | 0.19 | 9.18 | 8.81 | 108.89±8.66 | 113.53±9.23 |
| 0.8 | 0.5 | 90.56±0.50 | 80.90±0.67 | 0.09 | 0.19 | 9.23 | 8.72 | 108.36±8.86 | 114.65±9.01 |
| 1.0 | 0.1 | 87.47±0.57 | 81.32±0.66 | 0.12 | 0.18 | 9.21 | 8.75 | 108.63±9.56 | 114.27±9.71 |
| 1.0 | 0.2 | 84.86±0.61 | 79.09±0.69 | 0.15 | 0.20 | 9.20 | 8.82 | 108.69±9.27 | 113.37±9.08 |
| 1.0 | 0.3 | 87.53±0.56 | 81.66±0.66 | 0.12 | 0.18 | 9.21 | 8.85 | 108.60±9.36 | 113.02±9.36 |
| 1.0 | 0.4 | 90.80±0.50 | 80.33±0.68 | 0.09 | 0.19 | 9.16 | 8.76 | 109.12±8.89 | 114.22±9.33 |
| 1.0 | 0.5 | 91.25±0.49 | 81.45±0.66 | 0.08 | 0.18 | 9.31 | 8.81 | 107.40±9.12 | 113.49±9.25 |

Tabelle 26: Ergebnisse der Evaluation des Trackers. Hintergrundmodell mit Rampenfunktion ($\lambda = 0.3$, $\alpha_1 = 3$, $\alpha_2 = 5$) als Verfallsfunktion.

## A.3 ERGEBNISTABELLEN HAUTFARBMODELLIERUNG MIT GMM

| | RGB | | | HSV | | LAB | | nRG | |
|---|---|---|---|---|---|---|---|---|---|
| #Mix | indoor | outdoor | komb. | ind. | outd. | ind. | outd. | ind. | outd. |
| 1 | 0.867 | 0.754 | **0.780** | 0.325 | 0.104 | 0.703 | 0.546 | 0.721 | 0.732 |
| 5 | **0.904** | 0.846 | - | 0.840 | 0.732 | 0.672 | 0.648 | **0.829** | 0.743 |
| 10 | **0.904** | **0.848** | 0.685 | **0.860** | 0.743 | 0.660 | 0.649 | 0.822 | 0.700 |
| 15 | 0.880 | 0.830 | - | 0.835 | 0.801 | 0.660 | 0.665 | 0.809 | 0.745 |
| 20 | 0.886 | 0.838 | 0.566 | 0.813 | 0.785 | 0.685 | 0.658 | 0.824 | 0.738 |
| 25 | 0.872 | 0.851 | - | 0.805 | 0.784 | **0.726** | 0.656 | 0.820 | 0.753 |
| 30 | 0.880 | 0.837 | 0.634 | 0.843 | 0.794 | 0.694 | 0.625 | 0.795 | 0.740 |
| 35 | 0.868 | 0.835 | - | 0.852 | **0.822** | 0.678 | 0.663 | 0.794 | 0.741 |
| 40 | 0.885 | 0.855 | 0.676 | 0.814 | 0.792 | 0.691 | 0.669 | 0.794 | 0.748 |
| 45 | 0.872 | 0.852 | - | 0.852 | 0.790 | 0.699 | **0.670** | 0.793 | **0.756** |
| 50 | 0.877 | 0.820 | 0.695 | 0.855 | 0.803 | 0.686 | 0.655 | 0.796 | 0.704 |

Tabelle 27: Hautfarb-Detektionsergebnisse (*Equal Error Rate*) für alle Versuchsreihen. Das jeweils beste Ergebnis für jede Kombination aus Farbraum und Weißabgleichseinstellung ist in Fettdruck hervorgehoben.

## A.4 ERGEBNISTABELLEN HANDDETEKTION MIT ONLINE MODELL

### A.4.1 *Datensatz FINCA-PH*

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 84.97±0.93 | 86.59±0.93 | 1.95 | 1.78 | 7.41 | 134.98±26.14 |
| 50 | 0.6 | 81.54±1.01 | 83.28±1.02 | 1.30 | 1.17 | 7.64 | 130.88±24.11 |
| 50 | 0.7 | 77.62±1.08 | 79.38±1.10 | 0.79 | 0.69 | 7.33 | 136.45±29.50 |
| 50 | 0.8 | 70.74±1.17 | 72.38±1.21 | 0.40 | 0.31 | 7.72 | 129.60±18.78 |
| 50 | 0.9 | 60.44±1.25 | 62.12±1.30 | 0.20 | 0.14 | 7.82 | 127.84±16.46 |
| 70 | 0.5 | 84.95±0.93 | 86.49±0.94 | 1.82 | 1.65 | 7.37 | 135.64±26.08 |
| 70 | 0.6 | 83.20±0.97 | 84.72±0.98 | 1.08 | 0.92 | 7.67 | 130.44±22.19 |
| 70 | 0.7 | 79.85±1.04 | 81.43±1.06 | 0.74 | 0.59 | 7.35 | 136.14±21.07 |
| 70 | 0.8 | 72.01±1.15 | 73.48±1.19 | 0.36 | 0.26 | 7.57 | 132.13±19.27 |
| 70 | 0.9 | 60.34±1.25 | 62.01±1.30 | 0.19 | 0.13 | 7.65 | 130.80±17.85 |
| 90 | 0.5 | 86.19±0.90 | 87.31±0.91 | 1.53 | 1.30 | 7.67 | 130.31±21.13 |
| 90 | 0.6 | 83.17±0.97 | 84.42±0.99 | 0.81 | 0.62 | 7.79 | 128.39±18.67 |
| 90 | 0.7 | 78.89±1.05 | 80.32±1.08 | 0.51 | 0.37 | 7.82 | 127.96±16.82 |
| 90 | 0.8 | 71.71±1.16 | 73.19±1.20 | 0.29 | 0.19 | 7.83 | 127.73±16.91 |
| 90 | 0.9 | 60.23±1.25 | 61.90±1.30 | 0.17 | 0.11 | 7.86 | 127.25±16.38 |
| 110 | 0.5 | 83.94±0.95 | 84.81±0.98 | 1.10 | 0.83 | 7.68 | 130.20±20.51 |
| 110 | 0.6 | 81.66±1.00 | 82.63±1.03 | 0.64 | 0.43 | 7.75 | 128.95±17.96 |
| 110 | 0.7 | 77.28±1.08 | 78.58±1.11 | 0.38 | 0.24 | 7.85 | 127.37±17.19 |
| 110 | 0.8 | 70.57±1.17 | 71.95±1.21 | 0.23 | 0.13 | 7.73 | 129.38±18.22 |
| 110 | 0.9 | 59.46±1.25 | 61.07±1.31 | 0.14 | 0.08 | 7.82 | 127.85±17.39 |
| 130 | 0.5 | 78.62±1.06 | 79.25±1.10 | 0.64 | 0.38 | 7.74 | 129.18±19.23 |
| 130 | 0.6 | 77.79±1.07 | 78.53±1.11 | 0.52 | 0.31 | 7.69 | 129.99±19.40 |
| 130 | 0.7 | 74.48±1.12 | 75.59±1.16 | 0.33 | 0.19 | 7.77 | 128.66±18.43 |
| 130 | 0.8 | 68.26±1.19 | 69.49±1.24 | 0.21 | 0.10 | 7.93 | 126.14±16.36 |
| 130 | 0.9 | 57.70±1.26 | 59.20±1.32 | 0.13 | 0.06 | 7.89 | 126.67±17.15 |
| 150 | 0.5 | 71.58±1.16 | 71.99±1.21 | 0.39 | 0.19 | 7.85 | 127.36±17.53 |
| 150 | 0.6 | 71.58±1.16 | 71.99±1.21 | 0.39 | 0.19 | 7.87 | 127.07±17.51 |
| 150 | 0.7 | 70.02±1.18 | 70.73±1.23 | 0.29 | 0.15 | 7.83 | 127.64±17.01 |
| 150 | 0.8 | 65.12±1.22 | 66.08±1.27 | 0.19 | 0.09 | 7.92 | 126.19±16.42 |
| 150 | 0.9 | 55.35±1.27 | 56.65±1.32 | 0.12 | 0.06 | 7.91 | 126.47±15.49 |

Tabelle 28: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-PH. Hintergrundmodell: Asymptotische Exponentialfunktion, $\alpha_1 = 0.79$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild±σ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 84.51±0.94 | 85.97±0.95 | 2.01 | 1.80 | 7.88 | 126.92±24.39 |
| 50 | 0.6 | 81.26±1.01 | 82.82±1.03 | 1.27 | 1.10 | 8.06 | 124.08±21.22 |
| 50 | 0.7 | 77.70±1.07 | 79.38±1.10 | 0.82 | 0.68 | 8.15 | 122.77±20.35 |
| 50 | 0.8 | 71.64±1.16 | 73.33±1.20 | 0.45 | 0.35 | 8.20 | 121.89±17.16 |
| 50 | 0.9 | 61.44±1.24 | 63.04±1.30 | 0.23 | 0.15 | 8.28 | 120.78±15.70 |
| 70 | 0.5 | 85.40±0.92 | 86.75±0.93 | 1.88 | 1.69 | 7.96 | 125.66±23.38 |
| 70 | 0.6 | 83.54±0.96 | 84.92±0.98 | 1.12 | 0.92 | 8.13 | 123.03±19.22 |
| 70 | 0.7 | 79.56±1.04 | 81.03±1.07 | 0.71 | 0.55 | 8.18 | 122.22±17.39 |
| 70 | 0.8 | 73.22±1.14 | 74.70±1.18 | 0.40 | 0.28 | 8.24 | 121.30±15.79 |
| 70 | 0.9 | 61.34±1.24 | 62.93±1.30 | 0.21 | 0.14 | 8.28 | 120.84±15.27 |
| 90 | 0.5 | 86.78±0.88 | 87.78±0.90 | 1.74 | 1.45 | 7.91 | 126.43±20.89 |
| 90 | 0.6 | 83.79±0.96 | 84.93±0.98 | 0.90 | 0.69 | 8.16 | 122.62±17.25 |
| 90 | 0.7 | 79.93±1.04 | 81.29±1.06 | 0.56 | 0.40 | 8.20 | 122.02±16.15 |
| 90 | 0.8 | 72.94±1.14 | 74.40±1.18 | 0.32 | 0.21 | 8.25 | 121.19±15.53 |
| 90 | 0.9 | 61.22±1.24 | 62.80±1.30 | 0.19 | 0.11 | 8.20 | 121.91±15.10 |
| 110 | 0.5 | 85.10±0.93 | 85.90±0.95 | 1.28 | 0.97 | 8.04 | 124.35±20.94 |
| 110 | 0.6 | 82.84±0.98 | 83.77±1.01 | 0.72 | 0.49 | 8.18 | 122.32±17.13 |
| 110 | 0.7 | 78.66±1.06 | 79.90±1.09 | 0.43 | 0.26 | 8.20 | 121.96±15.98 |
| 110 | 0.8 | 72.03±1.15 | 73.42±1.19 | 0.26 | 0.15 | 8.22 | 121.70±15.80 |
| 110 | 0.9 | 60.70±1.25 | 62.24±1.30 | 0.17 | 0.09 | 8.24 | 121.33±15.09 |
| 130 | 0.5 | 80.29±1.03 | 80.81±1.07 | 0.76 | 0.45 | 8.05 | 124.16±19.07 |
| 130 | 0.6 | 79.35±1.05 | 80.01±1.09 | 0.59 | 0.36 | 8.17 | 122.39±17.28 |
| 130 | 0.7 | 76.12±1.10 | 77.16±1.14 | 0.37 | 0.20 | 8.23 | 121.54±16.03 |
| 130 | 0.8 | 70.07±1.18 | 71.29±1.22 | 0.23 | 0.11 | 8.27 | 120.96±15.32 |
| 130 | 0.9 | 59.28±1.25 | 60.71±1.31 | 0.15 | 0.07 | 8.27 | 120.86±15.28 |
| 150 | 0.5 | 73.44±1.14 | 73.76±1.19 | 0.44 | 0.22 | 8.16 | 122.51±16.58 |
| 150 | 0.6 | 73.42±1.14 | 73.76±1.19 | 0.44 | 0.22 | 8.21 | 121.74±16.71 |
| 150 | 0.7 | 71.90±1.16 | 72.59±1.21 | 0.33 | 0.16 | 8.22 | 121.60±15.93 |
| 150 | 0.8 | 67.03±1.20 | 68.04±1.26 | 0.21 | 0.10 | 8.28 | 120.84±15.32 |
| 150 | 0.9 | 56.80±1.26 | 58.04±1.32 | 0.14 | 0.06 | 8.22 | 121.71±14.89 |

Tabelle 29: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-PH. Hintergrundmodell: Sigmoid, $\alpha_1 = 0.45$, $\alpha_2 = 5.0$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 83.78±0.96 | 85.40±0.97 | 1.90 | 1.70 | 7.80 | 128.26±23.79 |
| 50 | 0.6 | 80.79±1.02 | 82.31±1.04 | 1.19 | 1.01 | 7.29 | 137.17±26.27 |
| 50 | 0.7 | 78.02±1.07 | 79.76±1.09 | 0.72 | 0.57 | 7.74 | 129.19±19.68 |
| 50 | 0.8 | 71.12±1.16 | 72.77±1.20 | 0.41 | 0.32 | 7.71 | 129.72±19.34 |
| 50 | 0.9 | 60.54±1.25 | 62.27±1.30 | 0.25 | 0.18 | 7.59 | 131.74±18.92 |
| 70 | 0.5 | 84.82±0.93 | 86.29±0.94 | 1.85 | 1.65 | 7.58 | 131.85±25.63 |
| 70 | 0.6 | 83.15±0.97 | 84.55±0.99 | 1.09 | 0.88 | 8.15 | 122.63±18.14 |
| 70 | 0.7 | 79.26±1.05 | 80.76±1.07 | 0.71 | 0.55 | 8.21 | 121.78±16.60 |
| 70 | 0.8 | 73.52±1.14 | 74.79±1.18 | 0.45 | 0.33 | 8.26 | 121.09±16.27 |
| 70 | 0.9 | 61.51±1.24 | 63.07±1.30 | 0.23 | 0.15 | 8.32 | 120.23±14.88 |
| 90 | 0.5 | 86.36±0.89 | 87.52±0.91 | 1.79 | 1.53 | 8.01 | 124.80±21.11 |
| 90 | 0.6 | 83.34±0.97 | 84.61±0.99 | 0.95 | 0.74 | 8.04 | 124.40±18.18 |
| 90 | 0.7 | 79.45±1.04 | 80.76±1.07 | 0.63 | 0.47 | 8.21 | 121.85±15.72 |
| 90 | 0.8 | 72.53±1.15 | 73.96±1.19 | 0.34 | 0.23 | 8.23 | 121.55±15.10 |
| 90 | 0.9 | 61.48±1.24 | 63.03±1.30 | 0.20 | 0.13 | 8.25 | 121.21±15.23 |
| 110 | 0.5 | 84.82±0.93 | 85.78±0.96 | 1.33 | 1.02 | 8.07 | 123.93±19.73 |
| 110 | 0.6 | 82.63±0.98 | 83.66±1.01 | 0.74 | 0.51 | 8.18 | 122.24±16.63 |
| 110 | 0.7 | 78.46±1.06 | 79.68±1.09 | 0.45 | 0.28 | 8.23 | 121.58±15.36 |
| 110 | 0.8 | 71.96±1.15 | 73.35±1.20 | 0.27 | 0.16 | 7.99 | 125.17±14.46 |
| 110 | 0.9 | 61.14±1.24 | 62.68±1.30 | 0.17 | 0.10 | 8.30 | 120.52±14.92 |
| 130 | 0.5 | 80.67±1.02 | 81.35±1.06 | 0.80 | 0.49 | 8.13 | 122.93±18.36 |
| 130 | 0.6 | 79.51±1.04 | 80.27±1.08 | 0.61 | 0.37 | 8.18 | 122.22±16.97 |
| 130 | 0.7 | 76.04±1.10 | 77.05±1.14 | 0.39 | 0.21 | 8.26 | 121.08±15.15 |
| 130 | 0.8 | 70.05±1.18 | 71.27±1.22 | 0.24 | 0.12 | 8.29 | 120.66±15.09 |
| 130 | 0.9 | 59.65±1.25 | 61.07±1.31 | 0.15 | 0.08 | 8.29 | 120.66±14.81 |
| 150 | 0.5 | 74.03±1.13 | 74.48±1.18 | 0.48 | 0.24 | 8.19 | 122.11±16.64 |
| 150 | 0.6 | 73.99±1.13 | 74.44±1.18 | 0.47 | 0.24 | 8.19 | 122.11±16.65 |
| 150 | 0.7 | 72.33±1.15 | 73.03±1.20 | 0.35 | 0.18 | 8.26 | 121.04±15.36 |
| 150 | 0.8 | 67.32±1.20 | 68.31±1.25 | 0.22 | 0.10 | 8.29 | 120.60±14.95 |
| 150 | 0.9 | 57.45±1.26 | 58.68±1.32 | 0.14 | 0.07 | 8.29 | 120.66±14.72 |

Tabelle 30: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-PH. Hintergrundmodell: Rampe, $\alpha_1 = 3$, $\alpha_2 = 5$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

### A.4.2 *Datensatz FINCA-G*

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 85.10±0.62 | 90.74±0.53 | 0.97 | 0.98 | 8.09 | 123.63±20.01 |
| 50 | 0.6 | 87.19±0.58 | 92.93±0.47 | 0.61 | 0.60 | 7.99 | 125.11±19.31 |
| 50 | 0.7 | 86.66±0.59 | 92.38±0.49 | 0.36 | 0.35 | 8.24 | 121.42±16.44 |
| 50 | 0.8 | 83.80±0.64 | 89.29±0.57 | 0.70 | 0.71 | 8.22 | 121.61±15.93 |
| 50 | 0.9 | 80.85±0.68 | 86.25±0.63 | 0.33 | 0.33 | 8.32 | 120.20±14.92 |
| 70 | 0.5 | 82.86±0.65 | 88.35±0.59 | 0.66 | 0.66 | 8.18 | 122.27±15.78 |
| 70 | 0.6 | 86.02±0.60 | 91.68±0.51 | 0.39 | 0.37 | 8.26 | 121.11±15.51 |
| 70 | 0.7 | 85.64±0.61 | 91.30±0.52 | 0.24 | 0.23 | 8.26 | 121.00±15.28 |
| 70 | 0.8 | 82.85±0.65 | 88.29±0.59 | 0.66 | 0.67 | 8.25 | 121.28±15.81 |
| 70 | 0.9 | 80.04±0.69 | 85.41±0.64 | 0.32 | 0.32 | 8.34 | 119.91±14.87 |
| 90 | 0.5 | 80.12±0.69 | 85.48±0.64 | 0.50 | 0.49 | 8.20 | 121.95±16.29 |
| 90 | 0.6 | 84.43±0.63 | 90.06±0.55 | 0.26 | 0.25 | 8.27 | 120.94±15.32 |
| 90 | 0.7 | 84.02±0.64 | 89.65±0.56 | 0.14 | 0.13 | 8.27 | 120.88±15.28 |
| 90 | 0.8 | 81.20±0.68 | 86.61±0.62 | 0.58 | 0.59 | 8.27 | 120.95±15.63 |
| 90 | 0.9 | 78.42±0.71 | 83.76±0.67 | 0.27 | 0.27 | 8.33 | 120.02±14.77 |
| 110 | 0.5 | 76.50±0.73 | 81.54±0.71 | 0.45 | 0.45 | 8.20 | 121.93±15.82 |
| 110 | 0.6 | 81.72±0.67 | 87.18±0.61 | 0.24 | 0.23 | 8.29 | 120.56±15.38 |
| 110 | 0.7 | 81.33±0.67 | 86.77±0.62 | 0.13 | 0.12 | 8.30 | 120.46±15.03 |
| 110 | 0.8 | 78.70±0.71 | 83.98±0.67 | 0.57 | 0.58 | 8.28 | 120.74±15.58 |
| 110 | 0.9 | 76.00±0.74 | 81.21±0.71 | 0.26 | 0.26 | 8.30 | 120.49±15.12 |
| 130 | 0.5 | 78.03±0.72 | 83.32±0.68 | 0.35 | 0.34 | 8.25 | 121.22±15.56 |
| 130 | 0.6 | 77.95±0.72 | 83.21±0.68 | 0.22 | 0.21 | 8.41 | 118.84±15.65 |
| 130 | 0.7 | 77.67±0.72 | 82.93±0.69 | 0.12 | 0.11 | 7.68 | 130.14±27.07 |
| 130 | 0.8 | 75.35±0.74 | 80.45±0.72 | 0.56 | 0.57 | 8.15 | 122.75±17.71 |
| 130 | 0.9 | 72.76±0.77 | 77.78±0.75 | 0.25 | 0.25 | 8.31 | 120.32±14.73 |
| 150 | 0.5 | 73.13±0.76 | 78.11±0.75 | 0.19 | 0.18 | 8.21 | 121.77±16.78 |
| 150 | 0.6 | 73.12±0.76 | 78.11±0.75 | 0.19 | 0.18 | 8.00 | 124.98±17.63 |
| 150 | 0.7 | 73.01±0.77 | 77.98±0.75 | 0.11 | 0.10 | 6.77 | 147.79±28.47 |
| 150 | 0.8 | 71.21±0.78 | 76.05±0.77 | 0.55 | 0.57 | 8.32 | 120.23±17.08 |
| 150 | 0.9 | 68.77±0.80 | 73.56±0.80 | 0.25 | 0.25 | 8.37 | 119.45±14.74 |

Tabelle 31: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-G. Hintergrundmodell: Asymptotische Exponentialfunktion, $\alpha_1 = 0.79$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild±σ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 87.40±0.58 | 91.02±0.52 | 1.01 | 1.00 | 8.39 | 119.12±18.50 |
| 50 | 0.6 | 89.70±0.53 | 93.43±0.45 | 0.64 | 0.63 | 8.47 | 118.06±16.84 |
| 50 | 0.7 | 88.97±0.55 | 92.65±0.47 | 0.38 | 0.37 | 8.53 | 117.20±14.34 |
| 50 | 0.8 | 87.85±0.57 | 91.45±0.51 | 0.22 | 0.21 | 8.61 | 116.15±12.39 |
| 50 | 0.9 | 83.22±0.65 | 86.70±0.61 | 0.33 | 0.33 | 8.63 | 115.93±11.90 |
| 70 | 0.5 | 85.50±0.61 | 88.99±0.56 | 0.65 | 0.64 | 8.46 | 118.15±12.99 |
| 70 | 0.6 | 88.79±0.55 | 92.47±0.48 | 0.37 | 0.35 | 8.54 | 117.09±12.37 |
| 70 | 0.7 | 88.27±0.56 | 91.92±0.49 | 0.22 | 0.20 | 8.54 | 117.09±12.17 |
| 70 | 0.8 | 87.15±0.58 | 90.72±0.53 | 0.16 | 0.15 | 8.58 | 116.60±12.12 |
| 70 | 0.9 | 82.70±0.66 | 86.15±0.62 | 0.32 | 0.31 | 8.61 | 116.21±12.26 |
| 90 | 0.5 | 82.64±0.66 | 86.03±0.62 | 0.52 | 0.50 | 8.47 | 118.06±12.98 |
| 90 | 0.6 | 86.96±0.59 | 90.58±0.53 | 0.27 | 0.25 | 8.58 | 116.60±12.26 |
| 90 | 0.7 | 86.43±0.60 | 90.01±0.54 | 0.14 | 0.12 | 8.59 | 116.36±12.00 |
| 90 | 0.8 | 85.37±0.61 | 88.87±0.57 | 0.09 | 0.08 | 8.61 | 116.10±11.85 |
| 90 | 0.9 | 81.04±0.68 | 84.47±0.65 | 0.27 | 0.26 | 8.63 | 115.85±11.87 |
| 110 | 0.5 | 79.08±0.70 | 82.36±0.68 | 0.49 | 0.48 | 8.52 | 117.34±12.85 |
| 110 | 0.6 | 84.43±0.63 | 88.00±0.59 | 0.26 | 0.24 | 8.53 | 117.21±12.34 |
| 110 | 0.7 | 83.98±0.64 | 87.52±0.60 | 0.13 | 0.12 | 8.57 | 116.64±12.07 |
| 110 | 0.8 | 83.07±0.65 | 86.54±0.61 | 0.08 | 0.07 | 8.60 | 116.29±12.09 |
| 110 | 0.9 | 78.85±0.71 | 82.26±0.68 | 0.26 | 0.26 | 8.63 | 115.92±11.76 |
| 130 | 0.5 | 80.99±0.68 | 84.42±0.65 | 0.38 | 0.37 | 8.55 | 117.02±12.90 |
| 130 | 0.6 | 80.86±0.68 | 84.28±0.65 | 0.24 | 0.22 | 8.55 | 116.97±12.45 |
| 130 | 0.7 | 80.57±0.68 | 83.98±0.66 | 0.12 | 0.11 | 8.58 | 116.62±12.38 |
| 130 | 0.8 | 79.84±0.69 | 83.19±0.67 | 0.08 | 0.07 | 8.62 | 115.97±11.89 |
| 130 | 0.9 | 75.81±0.74 | 79.12±0.73 | 0.26 | 0.26 | 8.62 | 115.94±11.88 |
| 150 | 0.5 | 76.05±0.74 | 79.30±0.72 | 0.21 | 0.19 | 8.58 | 116.56±12.18 |
| 150 | 0.6 | 76.04±0.74 | 79.28±0.72 | 0.20 | 0.19 | 8.58 | 116.51±12.27 |
| 150 | 0.7 | 75.89±0.74 | 79.13±0.73 | 0.12 | 0.10 | 8.60 | 116.28±12.06 |
| 150 | 0.8 | 75.36±0.74 | 78.57±0.73 | 0.07 | 0.06 | 8.63 | 115.94±11.90 |
| 150 | 0.9 | 71.81±0.78 | 74.96±0.77 | 0.26 | 0.25 | 8.61 | 116.09±11.74 |

Tabelle 32: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-G. Hintergrundmodell: Sigmoidfunktion, $\alpha_1 = 0.45$, $\alpha_2 = 5.0$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 85.98±0.60 | 90.70±0.53 | 1.09 | 1.10 | 8.35 | 119.79±19.38 |
| 50 | 0.6 | 85.77±0.61 | 90.42±0.54 | 0.68 | 0.67 | 8.45 | 118.36±17.79 |
| 50 | 0.7 | 87.43±0.58 | 92.14±0.49 | 0.46 | 0.45 | 8.51 | 117.50±16.73 |
| 50 | 0.8 | 86.37±0.60 | 91.00±0.52 | 0.27 | 0.25 | 8.57 | 116.64±14.11 |
| 50 | 0.9 | 81.90±0.67 | 86.39±0.62 | 0.38 | 0.37 | 8.59 | 116.45±13.23 |
| 70 | 0.5 | 84.25±0.63 | 88.88±0.57 | 0.75 | 0.75 | 8.48 | 117.97±14.50 |
| 70 | 0.6 | 83.96±0.64 | 88.53±0.58 | 0.44 | 0.43 | 8.53 | 117.30±14.19 |
| 70 | 0.7 | 86.69±0.59 | 91.41±0.51 | 0.29 | 0.27 | 8.53 | 117.20±13.43 |
| 70 | 0.8 | 85.73±0.61 | 90.36±0.54 | 0.20 | 0.19 | 8.56 | 116.75±13.47 |
| 70 | 0.9 | 81.37±0.67 | 85.87±0.63 | 0.36 | 0.35 | 8.59 | 116.41±13.19 |
| 90 | 0.5 | 81.83±0.67 | 86.29±0.62 | 0.53 | 0.52 | 8.50 | 117.71±13.94 |
| 90 | 0.6 | 81.49±0.67 | 85.92±0.63 | 0.27 | 0.25 | 8.56 | 116.77±13.38 |
| 90 | 0.7 | 85.23±0.62 | 89.91±0.55 | 0.15 | 0.13 | 8.58 | 116.61±13.22 |
| 90 | 0.8 | 84.25±0.63 | 88.85±0.57 | 0.09 | 0.08 | 8.60 | 116.22±12.93 |
| 90 | 0.9 | 79.96±0.69 | 84.41±0.66 | 0.28 | 0.28 | 8.56 | 116.84±13.07 |
| 110 | 0.5 | 78.33±0.71 | 82.61±0.69 | 0.51 | 0.50 | 8.50 | 117.67±13.99 |
| 110 | 0.6 | 78.13±0.71 | 82.41±0.69 | 0.26 | 0.24 | 8.53 | 117.23±13.58 |
| 110 | 0.7 | 82.89±0.65 | 87.49±0.60 | 0.14 | 0.13 | 8.58 | 116.62±13.21 |
| 110 | 0.8 | 82.01±0.67 | 86.53±0.62 | 0.08 | 0.07 | 8.48 | 117.93±13.18 |
| 110 | 0.9 | 77.94±0.72 | 82.35±0.69 | 0.28 | 0.28 | 8.53 | 117.26±14.41 |
| 130 | 0.5 | 74.18±0.75 | 78.25±0.74 | 0.41 | 0.40 | 8.50 | 117.64±13.75 |
| 130 | 0.6 | 74.14±0.76 | 78.19±0.74 | 0.24 | 0.23 | 8.54 | 117.13±13.54 |
| 130 | 0.7 | 79.73±0.70 | 84.16±0.66 | 0.13 | 0.12 | 8.60 | 116.33±13.09 |
| 130 | 0.8 | 79.00±0.70 | 83.36±0.67 | 0.08 | 0.07 | 8.60 | 116.21±13.06 |
| 130 | 0.9 | 75.00±0.75 | 79.26±0.73 | 0.28 | 0.27 | 8.57 | 116.74±13.61 |
| 150 | 0.5 | 75.82±0.74 | 80.17±0.72 | 0.23 | 0.21 | 8.01 | 124.89±19.99 |
| 150 | 0.6 | 75.82±0.74 | 80.16±0.72 | 0.22 | 0.21 | 8.54 | 117.10±13.67 |
| 150 | 0.7 | 75.66±0.74 | 79.99±0.72 | 0.12 | 0.11 | 8.58 | 116.59±13.24 |
| 150 | 0.8 | 75.09±0.75 | 79.35±0.73 | 0.07 | 0.06 | 8.56 | 116.80±13.64 |
| 150 | 0.9 | 71.58±0.78 | 75.69±0.77 | 0.27 | 0.27 | 8.51 | 117.45±14.56 |

Tabelle 33: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-G. Hintergrundmodell: Rampenfunktion, $\alpha_1 = 3$, $\alpha_2 = 5$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

## A.5 ERGEBNISTABELLEN HANDDETEKTION MIT KOMBINIERTEM MODELL

### A.5.1 *Datensatz FINCA-PH*

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 85.02±0.93 | 86.16±0.95 | 0.66 | 0.44 | 7.45 | 134.27±19.99 |
| 50 | 0.6 | 80.54±1.02 | 81.89±1.05 | 0.44 | 0.26 | 7.40 | 135.15±21.55 |
| 50 | 0.7 | 75.97±1.10 | 77.44±1.13 | 0.30 | 0.17 | 7.18 | 139.22±24.66 |
| 50 | 0.8 | 68.94±1.19 | 70.53±1.23 | 0.20 | 0.11 | 7.47 | 133.80±20.57 |
| 50 | 0.9 | 58.27±1.26 | 59.96±1.31 | 0.14 | 0.08 | 7.57 | 132.06±18.14 |
| 70 | 0.5 | 84.19±0.95 | 85.27±0.97 | 0.61 | 0.39 | 7.47 | 133.87±18.59 |
| 70 | 0.6 | 80.03±1.03 | 81.34±1.06 | 0.42 | 0.24 | 7.47 | 133.79±19.23 |
| 70 | 0.7 | 75.57±1.11 | 77.00±1.14 | 0.28 | 0.16 | 7.60 | 131.54±17.21 |
| 70 | 0.8 | 68.64±1.19 | 70.20±1.23 | 0.19 | 0.10 | 7.31 | 136.80±18.11 |
| 70 | 0.9 | 58.05±1.26 | 59.72±1.31 | 0.13 | 0.07 | 7.63 | 131.03±17.15 |
| 90 | 0.5 | 81.98±1.00 | 82.93±1.03 | 0.57 | 0.35 | 7.52 | 132.97±19.58 |
| 90 | 0.6 | 78.42±1.06 | 79.60±1.09 | 0.39 | 0.22 | 7.55 | 132.48±18.81 |
| 90 | 0.7 | 74.28±1.13 | 75.61±1.16 | 0.27 | 0.15 | 7.55 | 132.38±17.40 |
| 90 | 0.8 | 67.62±1.20 | 69.11±1.24 | 0.18 | 0.09 | 7.61 | 131.35±17.14 |
| 90 | 0.9 | 57.16±1.26 | 58.78±1.32 | 0.12 | 0.06 | 7.53 | 132.84±18.02 |
| 110 | 0.5 | 78.56±1.06 | 79.27±1.10 | 0.50 | 0.28 | 7.49 | 133.43±18.82 |
| 110 | 0.6 | 76.02±1.10 | 77.01±1.14 | 0.36 | 0.19 | 7.55 | 132.52±17.68 |
| 110 | 0.7 | 72.43±1.15 | 73.61±1.19 | 0.25 | 0.13 | 7.64 | 130.95±16.82 |
| 110 | 0.8 | 66.01±1.21 | 67.37±1.26 | 0.17 | 0.08 | 7.63 | 131.10±17.04 |
| 110 | 0.9 | 55.91±1.26 | 57.43±1.32 | 0.11 | 0.05 | 7.56 | 132.23±17.16 |
| 130 | 0.5 | 72.75±1.14 | 73.30±1.19 | 0.38 | 0.18 | 7.50 | 133.36±19.40 |
| 130 | 0.6 | 71.91±1.15 | 72.63±1.20 | 0.33 | 0.16 | 7.40 | 135.17±19.95 |
| 130 | 0.7 | 69.26±1.18 | 70.21±1.23 | 0.24 | 0.11 | 7.66 | 130.50±16.94 |
| 130 | 0.8 | 63.47±1.23 | 64.65±1.28 | 0.16 | 0.07 | 7.66 | 130.47±18.05 |
| 130 | 0.9 | 53.67±1.27 | 55.04±1.33 | 0.11 | 0.05 | 7.75 | 129.00±16.44 |
| 150 | 0.5 | 65.86±1.21 | 66.24±1.27 | 0.28 | 0.11 | 7.72 | 129.46±16.90 |
| 150 | 0.6 | 65.84±1.21 | 66.24±1.27 | 0.28 | 0.11 | 7.73 | 129.35±17.08 |
| 150 | 0.7 | 64.73±1.22 | 65.34±1.28 | 0.22 | 0.09 | 7.71 | 129.63±16.63 |
| 150 | 0.8 | 60.20±1.25 | 61.12±1.31 | 0.15 | 0.06 | 7.67 | 130.45±16.03 |
| 150 | 0.9 | 51.11±1.27 | 52.27±1.33 | 0.10 | 0.04 | 7.69 | 130.12±15.55 |

Tabelle 34: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-PH. Hintergrundmodell: Asymptotische Exponentialfunktion, $\alpha_1 = 0.79$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild±σ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 85.22±0.92 | 86.23±0.94 | 0.73 | 0.48 | 7.90 | 126.53±17.35 |
| 50 | 0.6 | 81.29±1.01 | 82.51±1.04 | 0.48 | 0.28 | 7.88 | 126.96±17.35 |
| 50 | 0.7 | 76.69±1.09 | 78.12±1.12 | 0.32 | 0.18 | 7.97 | 125.39±15.93 |
| 50 | 0.8 | 69.78±1.18 | 71.28±1.22 | 0.22 | 0.13 | 7.98 | 125.39±15.98 |
| 50 | 0.9 | 58.86±1.25 | 60.43±1.31 | 0.15 | 0.09 | 8.01 | 124.77±15.38 |
| 70 | 0.5 | 84.56±0.94 | 85.51±0.96 | 0.67 | 0.43 | 7.90 | 126.62±17.35 |
| 70 | 0.6 | 80.87±1.02 | 82.05±1.05 | 0.45 | 0.26 | 7.91 | 126.47±16.25 |
| 70 | 0.7 | 76.38±1.10 | 77.77±1.13 | 0.30 | 0.16 | 8.00 | 124.93±15.82 |
| 70 | 0.8 | 69.51±1.18 | 70.98±1.22 | 0.21 | 0.11 | 8.00 | 124.99±15.48 |
| 70 | 0.9 | 58.66±1.26 | 60.21±1.31 | 0.15 | 0.08 | 8.00 | 125.03±15.62 |
| 90 | 0.5 | 82.80±0.98 | 83.62±1.01 | 0.63 | 0.38 | 7.92 | 126.26±17.52 |
| 90 | 0.6 | 79.48±1.04 | 80.53±1.08 | 0.42 | 0.23 | 7.95 | 125.77±16.24 |
| 90 | 0.7 | 75.29±1.11 | 76.59±1.15 | 0.29 | 0.15 | 7.99 | 125.15±15.99 |
| 90 | 0.8 | 68.69±1.19 | 70.09±1.23 | 0.20 | 0.10 | 7.97 | 125.54±15.58 |
| 90 | 0.9 | 57.99±1.26 | 59.49±1.32 | 0.14 | 0.07 | 8.02 | 124.72±15.49 |
| 110 | 0.5 | 80.02±1.03 | 80.62±1.08 | 0.56 | 0.32 | 7.90 | 126.64±17.50 |
| 110 | 0.6 | 77.45±1.08 | 78.33±1.12 | 0.39 | 0.20 | 7.97 | 125.50±16.26 |
| 110 | 0.7 | 73.67±1.13 | 74.85±1.17 | 0.28 | 0.13 | 7.99 | 125.20±15.68 |
| 110 | 0.8 | 67.33±1.20 | 68.63±1.25 | 0.19 | 0.09 | 7.91 | 126.35±15.49 |
| 110 | 0.9 | 57.00±1.26 | 58.43±1.32 | 0.13 | 0.06 | 8.02 | 124.70±15.52 |
| 130 | 0.5 | 74.35±1.12 | 74.77±1.18 | 0.44 | 0.22 | 7.94 | 125.87±16.90 |
| 130 | 0.6 | 73.49±1.14 | 74.11±1.18 | 0.37 | 0.18 | 7.93 | 126.12±16.35 |
| 130 | 0.7 | 70.72±1.17 | 71.68±1.22 | 0.26 | 0.12 | 7.97 | 125.42±15.80 |
| 130 | 0.8 | 64.97±1.22 | 66.09±1.27 | 0.18 | 0.08 | 7.93 | 126.18±15.65 |
| 130 | 0.9 | 55.05±1.27 | 56.34±1.33 | 0.12 | 0.05 | 7.99 | 125.09±15.43 |
| 150 | 0.5 | 67.95±1.20 | 68.24±1.25 | 0.31 | 0.12 | 7.94 | 125.93±16.18 |
| 150 | 0.6 | 67.92±1.20 | 68.24±1.25 | 0.31 | 0.12 | 7.94 | 126.01±16.21 |
| 150 | 0.7 | 66.54±1.21 | 67.17±1.26 | 0.24 | 0.10 | 8.01 | 124.90±15.67 |
| 150 | 0.8 | 61.81±1.24 | 62.69±1.30 | 0.17 | 0.07 | 7.95 | 125.81±15.34 |
| 150 | 0.9 | 52.62±1.27 | 53.71±1.33 | 0.12 | 0.05 | 7.98 | 125.36±15.52 |

Tabelle 35: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-PH. Hintergrundmodell: Sigmoidfunktion, $\alpha_1 = 0.45$, $\alpha_2 = 5.0$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 84.92±0.93 | 86.07±0.95 | 0.74 | 0.50 | 7.62 | 131.29±18.65 |
| 50 | 0.6 | 81.14±1.01 | 82.39±1.04 | 0.48 | 0.29 | 7.18 | 139.19±18.57 |
| 50 | 0.7 | 76.54±1.09 | 77.98±1.13 | 0.34 | 0.19 | 7.58 | 132.01±17.37 |
| 50 | 0.8 | 69.24±1.18 | 70.70±1.23 | 0.23 | 0.13 | 7.66 | 130.50±17.54 |
| 50 | 0.9 | 59.03±1.25 | 60.60±1.31 | 0.15 | 0.09 | 7.62 | 131.28±17.63 |
| 70 | 0.5 | 84.23±0.95 | 85.33±0.97 | 0.68 | 0.44 | 7.95 | 125.74±16.91 |
| 70 | 0.6 | 80.70±1.02 | 81.92±1.05 | 0.45 | 0.26 | 8.01 | 124.92±15.76 |
| 70 | 0.7 | 76.29±1.10 | 77.70±1.13 | 0.32 | 0.18 | 7.99 | 125.14±15.40 |
| 70 | 0.8 | 69.08±1.19 | 70.51±1.23 | 0.22 | 0.12 | 8.01 | 124.84±14.98 |
| 70 | 0.9 | 58.88±1.25 | 60.44±1.31 | 0.14 | 0.08 | 8.04 | 124.34±14.96 |
| 90 | 0.5 | 82.57±0.98 | 83.53±1.01 | 0.64 | 0.40 | 7.89 | 126.75±17.17 |
| 90 | 0.6 | 79.50±1.04 | 80.61±1.08 | 0.43 | 0.24 | 7.94 | 126.01±16.17 |
| 90 | 0.7 | 75.30±1.11 | 76.63±1.15 | 0.31 | 0.17 | 8.03 | 124.59±15.09 |
| 90 | 0.8 | 68.37±1.19 | 69.75±1.24 | 0.21 | 0.11 | 8.03 | 124.56±15.07 |
| 90 | 0.9 | 58.31±1.26 | 59.83±1.31 | 0.14 | 0.07 | 8.00 | 124.96±15.34 |
| 110 | 0.5 | 79.71±1.04 | 80.46±1.08 | 0.58 | 0.34 | 7.95 | 125.84±16.77 |
| 110 | 0.6 | 77.32±1.08 | 78.24±1.12 | 0.40 | 0.22 | 7.95 | 125.77±16.35 |
| 110 | 0.7 | 73.69±1.13 | 74.88±1.17 | 0.30 | 0.15 | 7.99 | 125.18±15.22 |
| 110 | 0.8 | 67.03±1.20 | 68.31±1.25 | 0.20 | 0.10 | 8.03 | 124.59±15.34 |
| 110 | 0.9 | 57.28±1.26 | 58.73±1.32 | 0.13 | 0.07 | 8.01 | 124.87±14.86 |
| 130 | 0.5 | 74.82±1.12 | 75.37±1.17 | 0.46 | 0.23 | 7.91 | 126.38±16.53 |
| 130 | 0.6 | 73.96±1.13 | 74.62±1.18 | 0.37 | 0.18 | 7.98 | 125.29±15.88 |
| 130 | 0.7 | 71.04±1.16 | 72.01±1.21 | 0.28 | 0.13 | 8.02 | 124.62±15.38 |
| 130 | 0.8 | 64.92±1.22 | 66.01±1.27 | 0.19 | 0.09 | 8.05 | 124.25±14.85 |
| 130 | 0.9 | 55.50±1.26 | 56.79±1.33 | 0.12 | 0.06 | 8.04 | 124.34±14.88 |
| 150 | 0.5 | 68.78±1.19 | 69.20±1.24 | 0.33 | 0.15 | 7.96 | 125.64±15.71 |
| 150 | 0.6 | 68.78±1.19 | 69.20±1.24 | 0.33 | 0.15 | 7.93 | 126.06±16.06 |
| 150 | 0.7 | 67.23±1.20 | 67.92±1.26 | 0.26 | 0.12 | 8.01 | 124.80±15.15 |
| 150 | 0.8 | 62.25±1.24 | 63.16±1.30 | 0.17 | 0.08 | 7.93 | 126.12±16.36 |
| 150 | 0.9 | 53.36±1.27 | 54.49±1.33 | 0.12 | 0.05 | 8.05 | 124.27±14.86 |

Tabelle 36: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-PH. Hintergrundmodell: Rampenfunktion, $\alpha_1 = 3$, $\alpha_2 = 5$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

### A.5.2 *Datensatz FINCA-G*

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 87.49±0.58 | 93.25±0.46 | 0.49 | 0.48 | 7.53 | 132.80±18.99 |
| 50 | 0.6 | 86.97±0.59 | 92.68±0.48 | 0.26 | 0.24 | 7.53 | 132.74±18.65 |
| 50 | 0.7 | 86.51±0.59 | 92.17±0.50 | 0.14 | 0.13 | 7.59 | 131.70±18.01 |
| 50 | 0.8 | 85.72±0.61 | 91.34±0.52 | 0.09 | 0.08 | 7.59 | 131.68±17.89 |
| 50 | 0.9 | 84.37±0.63 | 89.96±0.55 | 0.07 | 0.06 | 7.62 | 131.22±17.80 |
| 70 | 0.5 | 85.79±0.61 | 91.46±0.51 | 0.48 | 0.47 | 7.47 | 133.93±19.91 |
| 70 | 0.6 | 85.35±0.61 | 90.98±0.53 | 0.25 | 0.24 | 7.57 | 132.18±18.25 |
| 70 | 0.7 | 84.94±0.62 | 90.52±0.54 | 0.14 | 0.12 | 7.59 | 131.79±17.92 |
| 70 | 0.8 | 84.24±0.63 | 89.79±0.56 | 0.09 | 0.08 | 7.59 | 131.76±17.85 |
| 70 | 0.9 | 82.93±0.65 | 88.45±0.59 | 0.07 | 0.06 | 7.62 | 131.28±17.85 |
| 90 | 0.5 | 83.73±0.64 | 89.29±0.57 | 0.46 | 0.46 | 7.53 | 132.78±18.77 |
| 90 | 0.6 | 83.45±0.64 | 88.98±0.57 | 0.24 | 0.23 | 7.55 | 132.41±18.20 |
| 90 | 0.7 | 83.10±0.65 | 88.59±0.58 | 0.13 | 0.12 | 7.60 | 131.61±17.99 |
| 90 | 0.8 | 82.50±0.66 | 87.96±0.60 | 0.09 | 0.08 | 7.58 | 131.89±17.96 |
| 90 | 0.9 | 81.24±0.68 | 86.67±0.62 | 0.06 | 0.06 | 7.60 | 131.62±18.04 |
| 110 | 0.5 | 81.44±0.67 | 86.85±0.62 | 0.43 | 0.43 | 7.47 | 133.88±19.43 |
| 110 | 0.6 | 81.32±0.68 | 86.73±0.62 | 0.23 | 0.22 | 7.57 | 132.12±18.17 |
| 110 | 0.7 | 81.04±0.68 | 86.41±0.63 | 0.12 | 0.11 | 7.61 | 131.45±18.00 |
| 110 | 0.8 | 80.49±0.69 | 85.84±0.64 | 0.08 | 0.07 | 7.53 | 132.88±18.82 |
| 110 | 0.9 | 79.30±0.70 | 84.61±0.66 | 0.06 | 0.05 | 7.59 | 131.70±17.74 |
| 130 | 0.5 | 78.21±0.71 | 83.50±0.68 | 0.32 | 0.32 | 7.64 | 130.86±18.38 |
| 130 | 0.6 | 78.08±0.72 | 83.37±0.68 | 0.21 | 0.20 | 7.42 | 134.82±21.94 |
| 130 | 0.7 | 77.93±0.72 | 83.18±0.68 | 0.11 | 0.11 | 7.63 | 131.00±19.23 |
| 130 | 0.8 | 77.50±0.72 | 82.75±0.69 | 0.07 | 0.07 | 7.61 | 131.48±17.85 |
| 130 | 0.9 | 76.39±0.73 | 81.58±0.71 | 0.05 | 0.05 | 7.49 | 133.42±19.21 |
| 150 | 0.5 | 74.29±0.75 | 79.36±0.74 | 0.18 | 0.17 | 7.36 | 135.95±21.74 |
| 150 | 0.6 | 74.29±0.75 | 79.37±0.74 | 0.17 | 0.17 | 6.53 | 153.18±28.24 |
| 150 | 0.7 | 74.24±0.75 | 79.28±0.74 | 0.11 | 0.10 | 6.44 | 155.28±32.44 |
| 150 | 0.8 | 73.95±0.76 | 78.99±0.74 | 0.07 | 0.06 | 7.60 | 131.65±17.94 |
| 150 | 0.9 | 72.89±0.77 | 77.88±0.75 | 0.05 | 0.04 | 7.62 | 131.30±17.88 |

Tabelle 37: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-G. Hintergrundmodell: Asymptotische Exponentialfunktion, $\alpha_1 = 0.79$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild±σ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 89.85±0.53 | 93.54±0.45 | 0.51 | 0.50 | 7.72 | 129.52±16.21 |
| 50 | 0.6 | 89.28±0.54 | 92.96±0.47 | 0.26 | 0.24 | 7.78 | 128.54±15.42 |
| 50 | 0.7 | 88.77±0.55 | 92.41±0.48 | 0.14 | 0.13 | 7.82 | 127.88±15.02 |
| 50 | 0.8 | 87.82±0.57 | 91.39±0.51 | 0.09 | 0.08 | 7.82 | 127.87±14.94 |
| 50 | 0.9 | 86.48±0.59 | 89.99±0.54 | 0.07 | 0.06 | 7.83 | 127.73±15.16 |
| 70 | 0.5 | 88.32±0.56 | 91.94±0.49 | 0.50 | 0.49 | 7.72 | 129.55±16.26 |
| 70 | 0.6 | 87.81±0.57 | 91.42±0.51 | 0.26 | 0.24 | 7.79 | 128.37±15.50 |
| 70 | 0.7 | 87.38±0.58 | 90.96±0.52 | 0.14 | 0.12 | 7.80 | 128.22±15.09 |
| 70 | 0.8 | 86.50±0.59 | 90.01±0.54 | 0.09 | 0.08 | 7.85 | 127.45±14.73 |
| 70 | 0.9 | 85.20±0.62 | 88.66±0.57 | 0.07 | 0.06 | 7.84 | 127.56±14.83 |
| 90 | 0.5 | 86.43±0.60 | 89.99±0.54 | 0.49 | 0.48 | 7.75 | 129.07±16.14 |
| 90 | 0.6 | 86.01±0.60 | 89.56±0.55 | 0.25 | 0.23 | 7.78 | 128.57±15.31 |
| 90 | 0.7 | 85.71±0.61 | 89.24±0.56 | 0.13 | 0.12 | 7.79 | 128.39±15.11 |
| 90 | 0.8 | 84.91±0.62 | 88.38±0.58 | 0.09 | 0.07 | 7.80 | 128.19±14.93 |
| 90 | 0.9 | 83.67±0.64 | 87.09±0.60 | 0.06 | 0.06 | 7.83 | 127.68±14.86 |
| 110 | 0.5 | 84.16±0.63 | 87.66±0.59 | 0.46 | 0.45 | 7.74 | 129.21±16.19 |
| 110 | 0.6 | 83.93±0.64 | 87.43±0.60 | 0.24 | 0.23 | 7.80 | 128.28±15.27 |
| 110 | 0.7 | 83.75±0.64 | 87.22±0.60 | 0.13 | 0.11 | 7.80 | 128.23±15.24 |
| 110 | 0.8 | 83.05±0.65 | 86.46±0.62 | 0.08 | 0.07 | 7.84 | 127.53±14.87 |
| 110 | 0.9 | 81.90±0.67 | 85.26±0.64 | 0.06 | 0.05 | 7.85 | 127.43±14.68 |
| 130 | 0.5 | 81.28±0.68 | 84.70±0.65 | 0.36 | 0.34 | 7.76 | 128.80±15.80 |
| 130 | 0.6 | 81.11±0.68 | 84.54±0.65 | 0.22 | 0.21 | 7.79 | 128.29±15.32 |
| 130 | 0.7 | 81.05±0.68 | 84.45±0.65 | 0.12 | 0.11 | 7.81 | 128.05±15.22 |
| 130 | 0.8 | 80.44±0.69 | 83.79±0.66 | 0.07 | 0.07 | 7.82 | 127.86±14.85 |
| 130 | 0.9 | 79.31±0.70 | 82.62±0.68 | 0.05 | 0.05 | 7.83 | 127.75±14.84 |
| 150 | 0.5 | 77.35±0.72 | 80.65±0.71 | 0.20 | 0.18 | 7.80 | 128.13±15.36 |
| 150 | 0.6 | 77.34±0.72 | 80.65±0.71 | 0.19 | 0.17 | 7.78 | 128.46±15.65 |
| 150 | 0.7 | 77.34±0.72 | 80.63±0.71 | 0.11 | 0.10 | 7.82 | 127.80±15.07 |
| 150 | 0.8 | 76.93±0.73 | 80.19±0.71 | 0.07 | 0.06 | 7.83 | 127.66±14.76 |
| 150 | 0.9 | 75.98±0.74 | 79.20±0.73 | 0.05 | 0.04 | 7.85 | 127.36±14.64 |

Tabelle 38: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-G. Hintergrundmodell: Sigmoidfunktion, $\alpha_1 = 0.45$, $\alpha_2 = 5.0$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

| $\delta_V$ | $\gamma_H$ | Det.rate [%] | | FD/Bild | | BPS | Zeit/Bild$\pm\sigma$ [ms] |
|---|---|---|---|---|---|---|---|
| 50 | 0.5 | 87.92±0.57 | 92.73±0.48 | 0.53 | 0.52 | 7.72 | 129.52±17.56 |
| 50 | 0.6 | 87.95±0.57 | 92.73±0.48 | 0.27 | 0.26 | 7.74 | 129.24±17.04 |
| 50 | 0.7 | 87.46±0.58 | 92.17±0.49 | 0.15 | 0.13 | 7.80 | 128.14±16.36 |
| 50 | 0.8 | 86.64±0.59 | 91.29±0.52 | 0.09 | 0.08 | 7.80 | 128.21±16.10 |
| 50 | 0.9 | 85.16±0.62 | 89.74±0.55 | 0.07 | 0.06 | 7.81 | 128.09±16.24 |
| 70 | 0.5 | 86.44±0.60 | 91.17±0.52 | 0.52 | 0.51 | 7.70 | 129.85±17.72 |
| 70 | 0.6 | 86.51±0.59 | 91.21±0.52 | 0.26 | 0.25 | 7.76 | 128.83±16.83 |
| 70 | 0.7 | 86.08±0.60 | 90.74±0.53 | 0.14 | 0.13 | 7.80 | 128.22±16.37 |
| 70 | 0.8 | 85.31±0.62 | 89.92±0.55 | 0.09 | 0.07 | 7.82 | 127.92±16.21 |
| 70 | 0.9 | 83.86±0.64 | 88.39±0.58 | 0.06 | 0.06 | 7.81 | 128.03±16.08 |
| 90 | 0.5 | 84.60±0.63 | 89.26±0.56 | 0.50 | 0.50 | 7.71 | 129.69±17.43 |
| 90 | 0.6 | 84.85±0.62 | 89.49±0.56 | 0.26 | 0.24 | 7.77 | 128.78±16.63 |
| 90 | 0.7 | 84.47±0.63 | 89.07±0.57 | 0.14 | 0.12 | 7.78 | 128.61±16.44 |
| 90 | 0.8 | 83.78±0.64 | 88.33±0.58 | 0.08 | 0.07 | 7.81 | 128.09±16.21 |
| 90 | 0.9 | 82.37±0.66 | 86.85±0.61 | 0.06 | 0.05 | 7.81 | 128.01±16.48 |
| 110 | 0.5 | 82.32±0.66 | 86.91±0.61 | 0.48 | 0.48 | 7.65 | 130.71±17.78 |
| 110 | 0.6 | 82.75±0.66 | 87.34±0.60 | 0.25 | 0.24 | 7.77 | 128.67±16.65 |
| 110 | 0.7 | 82.48±0.66 | 87.04±0.61 | 0.13 | 0.12 | 7.78 | 128.46±16.62 |
| 110 | 0.8 | 81.90±0.67 | 86.43±0.62 | 0.08 | 0.07 | 7.82 | 127.83±16.09 |
| 110 | 0.9 | 80.56±0.69 | 84.99±0.65 | 0.06 | 0.05 | 7.83 | 127.74±16.08 |
| 130 | 0.5 | 79.52±0.70 | 83.95±0.66 | 0.39 | 0.38 | 7.71 | 129.74±17.13 |
| 130 | 0.6 | 80.14±0.69 | 84.62±0.65 | 0.23 | 0.22 | 7.74 | 129.17±16.46 |
| 130 | 0.7 | 80.00±0.69 | 84.44±0.66 | 0.12 | 0.11 | 7.79 | 128.30±16.49 |
| 130 | 0.8 | 79.52±0.70 | 83.92±0.66 | 0.07 | 0.06 | 7.79 | 128.42±16.57 |
| 130 | 0.9 | 78.20±0.71 | 82.52±0.69 | 0.05 | 0.05 | 7.77 | 128.70±16.44 |
| 150 | 0.5 | 76.76±0.73 | 81.13±0.71 | 0.21 | 0.20 | 7.73 | 129.31±17.78 |
| 150 | 0.6 | 76.74±0.73 | 81.12±0.71 | 0.20 | 0.19 | 7.78 | 128.54±16.67 |
| 150 | 0.7 | 76.80±0.73 | 81.14±0.71 | 0.11 | 0.10 | 7.78 | 128.56±16.59 |
| 150 | 0.8 | 76.42±0.73 | 80.74±0.71 | 0.07 | 0.06 | 7.79 | 128.29±16.29 |
| 150 | 0.9 | 75.25±0.74 | 79.47±0.73 | 0.05 | 0.04 | 7.53 | 132.78±18.82 |

Tabelle 39: Ergebnisse der Evaluation der Handdetektion, Datensatz FINCA-G. Hintergrundmodell: Rampenfunktion, $\alpha_1 = 3$, $\alpha_2 = 5$, $\lambda = 0.3$. Der erste angegebene Wert bei Detektions- und Fehldetektionsrate ist jeweils das globale Ergebnis. Der jeweils zweite Wert ist das Ergebnis ohne Kopfdetektionsfehler.

## A.6 TRAJEKTORIEN-KLASSIFIKATIONSEXPERIMENT – ERGEBNISSE FÜR EINZELMERKMALE

| | Topologie | Linear | | | | Bakis | | | |
|---|---|---|---|---|---|---|---|---|---|
| Merkmal | Fenstergröße | 1 | 5 | 7 | 9 | 1 | 5 | 7 | 9 |
| Rohe Trajektorie | | 73.3 | 75.1 | 75.6 | 75.4 | 72.5 | 76.2 | 74.9 | 74.5 |
| Δ Rohe Trajektorie | | 77.1 | 78.0 | 76.7 | 76.7 | 77.7 | 78.0 | 76.7 | 75.4 |
| Geschwindigkeit | | 68.9 | 71.2 | 70.6 | 69.9 | 70.2 | 70.2 | 71.3 | 69.9 |
| Δ Geschwindigkeit | | 69.2 | 66.7 | 66 | 68.4 | 68.9 | 64.4 | 65.1 | 68.7 |
| Norm. polare Traj. | | 72.0 | 75.8 | 75.9 | 76.1 | 71.8 | 75.5 | **76.7** | 76.4 |
| Δ Norm. polare Traj. | | 34.6 | 74.4 | 79.8 | **81.3** | 30.4 | 75.6 | 80.3 | 80.3 |
| Norm. Trajektorie | | 76.4 | 79.3 | 79.0 | 79.0 | 78.2 | 78.8 | 78.9 | 79.8 |
| Δ Norm. Trajektorie | | 12.4 | 65.9 | 72.9 | 74.2 | 13.1 | 73.5 | 76.2 | 75.4 |
| Nachbarschaft | | 68.4 | 65.0 | 65.0 | 66.7 | 68.3 | 64.1 | 65.4 | 65.1 |
| Δ Nachbarschaft | | 64.7 | 56.3 | 49.0 | 48.3 | 59.9 | 56.8 | 50.1 | 49.6 |
| Kopfabstand | | 56.5 | 57.1 | 55.5 | 57.8 | 60.5 | 58.9 | 61.2 | 60.5 |
| Δ Kopfabstand | | 11.5 | 12.7 | 21.3 | 24.2 | 13.1 | 14.0 | 20.6 | 25.9 |
| Orient.änderung | | 26.9 | 39.9 | 38.9 | 39.2 | 30.4 | 39.9 | 36.7 | 40.5 |
| Δ Orient.änderung | | 37.9 | 42.9 | 39.6 | 43.4 | 41.9 | 41.5 | 43.1 | 46.1 |
| Krümmung | | 29.7 | 35.3 | 35 | 36.6 | 38.5 | 31.8 | 36.6 | 39.8 |
| Δ Krümmung | | 42.9 | 40.1 | 39.8 | 40.1 | 46.3 | 41.4 | 42.7 | 42.1 |

Tabelle 40: Klassifikationsergebnisse für Einzelmerkmale (2D) für verschiedene Modelltopologien und Fenstergrößen bei der Merkmalsextraktion. Werte in %.

| | Topologie | Linear | | | | Bakis | | | |
|---|---|---|---|---|---|---|---|---|---|
| Merkmal | Fenstergröße | 1 | 5 | 7 | 9 | 1 | 5 | 7 | 9 |
| Rohe Trajektorie | | 56.3 | 52.7 | 54.8 | 54.2 | 53.2 | 55.6 | 56.1 | 55.8 |
| Δ Rohe Trajektorie | | 79.1 | 82.7 | 82.1 | 83.3 | 81.1 | **84.0** | 83.4 | 83.0 |
| Geschwindigkeit | | 80.3 | 80.3 | **80.8** | 79.8 | 79.7 | 79.3 | 80.4 | 79.8 |
| Δ Geschwindigkeit | | 77.8 | 74.9 | 74.2 | 75.5 | 76.9 | 72.5 | 74.4 | 74.9 |
| Norm. polare Traj. | | 76.8 | 79.7 | 79.4 | 79.0 | 77.1 | 80.3 | 80.0 | 79.5 |
| Δ norm. polare Traj. | | 31.7 | 52.3 | 68.3 | 76.8 | 34.3 | 54.9 | 68.2 | 77.5 |
| Norm. Trajektorie | | 75.4 | 77.7 | 77.8 | 76.7 | 76.9 | 77.7 | 78.5 | 77.5 |
| Δ norm. Trajektorie | | 11.5 | 13.0 | 27.4 | 37.8 | 13.1 | 15.4 | 27.4 | 49.0 |
| Nachbarschaft | | 70.3 | 69.6 | 70.5 | 71.3 | 70.7 | 70.2 | 71.3 | 71.6 |
| Δ Nachbarschaft | | 71.0 | 67.1 | 63.1 | 62.1 | 70.2 | 65.7 | 61.5 | 61.4 |
| Kopfabstand | | 59.9 | 55.8 | 56.5 | 55.6 | 58.9 | 59.1 | 59.7 | 59.5 |
| Δ Kopfabstand | | 12.2 | 12.1 | 20.3 | 25.4 | 13.1 | 14.0 | 19.9 | 27.7 |
| Orient.änderung | | 36.5 | 40.8 | 40.6 | 38.2 | 36.5 | 37.2 | 39.6 | 42.5 |
| Δ Orient.änderung | | 40.2 | 45.1 | 40.3 | 45.2 | 42.6 | 43.8 | 42.1 | 45.5 |
| Krümmung | | 36.3 | 37.8 | 38.6 | 36.6 | 40.9 | 34.6 | 38.6 | 39.6 |
| Δ Krümmung | | 49.1 | 41.5 | 40.3 | 43.2 | 48.3 | 41.5 | 39.9 | 42.4 |

Tabelle 41: Klassifikationsergebnisse für Einzelmerkmale (3D) für verschiedene Modelltopologien und Fenstergrößen bei der Merkmalsextraktion. Werte in %.

## A.7 TRAJEKTORIEN-KLASSIFIKATIONSEXPERIMENT – ERGEBNISSE FÜR HKA-MERKMALE

| | Topologie | Linear | | | Bakis | | |
|---|---|---|---|---|---|---|---|
| #HK | Fenstergröße | 5 | 7 | 9 | 5 | 7 | 9 |
| 1 | | 44.2 | 41.8 | 38.8 | 45.7 | 44.1 | 43.9 |
| 2 | | 70.6 | 68.6 | 68.0 | 69.0 | 68.9 | 68.4 |
| 3 | | 78.4 | 81.1 | 82.0 | 79.1 | 80.8 | 80.8 |
| 4 | | 84.1 | 83.7 | 86.0 | 85.0 | 84.1 | 85.4 |
| 5 | | 83.6 | 83.7 | 85.6 | **85.2** | 84.1 | 84.9 |
| 6 | | 83.1 | 84.1 | **86.6** | 84.1 | 84.6 | **86.2** |
| 7 | | **84.4** | 84.0 | 84.6 | 83.9 | 84.1 | 84.0 |
| 8 | | 83.9 | 84.6 | 85.0 | 84.1 | 85.0 | 83.9 |
| 9 | | 84.0 | 84.9 | 85.0 | 84.6 | **85.2** | 85.3 |
| 10 | | 84.0 | **85.0** | 85.6 | **85.2** | 84.6 | 84.0 |
| 12 | | 84.0 | 84.0 | 83.6 | 84.4 | 84.6 | 82.9 |
| 15 | | 83.1 | 82.9 | 80.7 | 84.4 | 83.6 | 80.8 |

Tabelle 42: Klassifikationsergebnisse (in %) für 2D HKA-Merkmale und verschiedene Topologien bzw. Fenstergrößen bei der Merkmalsextraktion in Abhängigkeit von der Anzahl verwendeter Hauptkomponenten. Angegeben sind jeweils die besten Ergebnisse aller Experimente mit unterschiedlicher Modelllänge und Anzahl Mixturkomponenten.

| | Topologie | Linear | | | Bakis | | |
|---|---|---|---|---|---|---|---|
| #HK | Fenstergröße | 5 | 7 | 9 | 5 | 7 | 9 |
| 1 | | 56.1 | 52.9 | 51.2 | 54.9 | 55.3 | 54.2 |
| 2 | | 70.6 | 73.8 | 75.9 | 68.9 | 74.2 | 74.9 |
| 3 | | 79.4 | 79.1 | 81.6 | 77.4 | 80.4 | 80.8 |
| 4 | | 82.1 | 83.6 | 85.3 | 81.3 | 83.0 | 84.7 |
| 5 | | 83.3 | 84.9 | 86.3 | 83.0 | 85.2 | 85.7 |
| 6 | | **85.9** | **86.9** | **88.3** | 85.9 | 86.9 | 86.9 |
| 7 | | 85.2 | 85.9 | 86.5 | 85.6 | **87.8** | **86.6** |
| 8 | | 85.2 | 85.4 | 87.0 | 85.2 | 87.2 | **87.2** |
| 9 | | 84.3 | 85.9 | 86.5 | 85.4 | 85.9 | 86.7 |
| 10 | | 84.3 | 85.3 | 85.4 | 84.7 | 84.4 | 86.0 |
| 12 | | 85.7 | 86.7 | 84.9 | 86.3 | 86.0 | 86.0 |
| 15 | | 85.6 | 86.0 | 84.1 | **86.7** | 85.7 | 83.4 |

Tabelle 43: Klassifikationsergebnisse (in %) für 3D HKA-Merkmale und verschiedene Topologien bzw. Fenstergrößen bei der Merkmalsextraktion in Abhängigkeit von der Anzahl verwendeter Hauptkomponenten. Angegeben sind jeweils die besten Ergebnisse aller Experimente mit unterschiedlicher Modelllänge und Anzahl Mixturkomponenten.

| | Topologie | Linear | | | Bakis | | |
|---|---|---|---|---|---|---|---|
| #HK | Fenstergröße | 5 | 7 | 9 | 5 | 7 | 9 |
| 1 | | 51.0 | 48.3 | 46.3 | 55.2 | 51.7 | 48.7 |
| 2 | | 68.2 | 74.4 | 76.7 | 69.6 | 74.5 | 76.7 |
| 3 | | 77.8 | 80.7 | 81.7 | 78.0 | 80.1 | 81.4 |
| 4 | | 83.1 | 86.2 | 85.9 | 82.9 | 84.4 | 85.7 |
| 5 | | 83.7 | 85.2 | 86.6 | 84.7 | 85.2 | 86.3 |
| 6 | | 86.0 | 86.0 | 87.8 | 86.3 | 86.2 | 86.9 |
| 7 | | 87.5 | **89.0** | **89.8** | 88.0 | 88.3 | **89.8** |
| 8 | | **88.2** | 88.2 | 88.6 | **89.2** | **88.9** | 89.3 |
| 9 | | 86.6 | 87.9 | 88.6 | 87.5 | 88.6 | 88.3 |
| 10 | | 87.9 | 88.2 | 88.5 | 87.0 | 88.3 | 88.2 |
| 12 | | 87.0 | 88.2 | 86.5 | 87.2 | 87.5 | 87.3 |
| 15 | | 85.7 | 85.2 | 82.9 | 86.5 | 85.3 | 83.4 |

Tabelle 44: Klassifikationsergebnisse (in %) für kombinierte HKA-Merkmale und verschiedene Topologien bzw. Fenstergrößen bei der Merkmalsextraktion in Abhängigkeit von der Anzahl verwendeter Hauptkomponenten. Angegeben sind jeweils die besten Ergebnisse aller Experimente mit unterschiedlicher Modelllänge und Anzahl Mixturkomponenten.

## A.8 ERGEBNISSE DES TRAJEKTORIEN-SEGMENTIERUNGSEXPERIMENTS

| Linear, Fenstergröße 5 | | | | |
|---|---|---|---|---|
| Rückweisungsmodelllänge | 1 | 3 | 6 | 8 |
| $F_1$-Wert (*Precision, Recall*) | **66.6 (64.5, 68.8)** | 65.3 (62.0, 69.1) | 65.5 (62.0, 69.5) | 65.4 (63.9, 67.0) |
| max. *Recall* (*Precision*) | 76.1 (46.7) | 75.6 (46.7) | **77.1 (46.5)** | 75.8 (48.0) |
| Bakis, Fenstergröße 5 | | | | |
| $F_1$-Wert (*Precision, Recall*) | 65.8 (62.5, 69.4) | **67.4 (64.5, 70.5)** | 64.4 (60.8, 68.5) | 65.6 (59.9, 72.4) |
| max. *Recall* (*Precision*) | 76.9 (47.1) | **78.5 (50.1)** | 76.0 (48.2) | 76.9 (49.1) |
| Linear, Fenstergröße 7 | | | | |
| Rückweisungsmodelllänge | 1 | 2 | 3 | 5 |
| $F_1$-Wert (*Precision, Recall*) | 64.8 (65.5, 64.1) | 66.0 (64.7, 67.3) | **66.2 (69.7, 63.1)** | 64.1 (63.6, 64.5) |
| max. *Recall* (*Precision*) | 73.1 (48.6) | **74.5 (50.3)** | 74.2 (50.4) | 72.6 (48.0) |
| Bakis, Fenstergröße 7 | | | | |
| $F_1$-Wert (*Precision, Recall*) | **66.1 (66.7, 65.5)** | 64.4 (59.7, 69.9) | 64.4 (64.5, 64.3) | 64.1 (63.8, 64.4) |
| max. *Recall* (*Precision*) | **74.6 (47.9)** | 73.5 (49.8) | 73.4 (48.1) | 72.8 (46.9) |
| Linear, Fenstergröße 9 | | | | |
| Rückweisungsmodelllänge | 1 | 2 | 3 | 4 |
| $F_1$-Wert (*Precision, Recall*) | **63.5 (63.5, 63.5)** | 60.6 (60.0, 61.2) | 63.2 (63.5, 62.8) | 63.2 (64.1, 62.3) |
| max. *Recall* (*Precision*) | **71.6 (50.5)** | 66.2 (48.6) | 68.6 (50.8) | 68.6 (49.7) |
| Bakis, Fenstergröße 9 | | | | |
| $F_1$-Wert (*Precision, Recall*) | 61.3 (58.9, 63.9) | 62.4 (60.1, 64.8) | 61.6 (64.3, 59.1) | **62.9 (66.2, 59.9)** |
| max. *Recall* (*Precision*) | 67.7 (49.1) | **70.1 (49.6)** | 67.9 (48.0) | 69.9 (48.4) |
| Linear, Fenstergröße 9, Codebuchgröße 500 | | | | |
| Rückweisungsmodelllänge | 1 | 2 | 3 | 4 |
| $F_1$-Wert (*Precision, Recall*) | 66.6 (65.0, 68.2) | 66.8 (68.4, 65.3) | 67.1 (71.3, 63.3) | **67.5 (71.2, 64.1)** |
| max. *Recall* (*Precision*) | **74.6 (50.7)** | 74.4 (51.7) | 74.4 (51.2) | 73.7 (51.1) |
| Bakis, Fenstergröße 9, Codebuchgröße 500 | | | | |
| $F_1$-Wert (*Precision, Recall*) | 64.2 (65.2, 63.3) | **66.4 (65.8, 67.0)** | 65.5 (65.4, 65.6) | 64.1 (64.5, 63.8) |
| max. *Recall* (*Precision*) | 71.6 (48.6) | **74.5 (50.8)** | 72.4 (50.0) | 71.3 (49.3) |

Tabelle 45: Ergebnisse für verschiedene Längen des Rückweisungsmodells und unterschiedliche Fenstergrößen. Alle Angaben in %. Die Konfidenzintervalle liegen zwischen 1.2 und 2.0, sind jedoch aus Platz- und Lesbarkeitsgründen nicht einzeln angegeben.

| Linear, Fenstergröße 5 | | | | |
|---|---|---|---|---|
| Rückweisungsmodelllänge | 1 | 3 | 6 | 8 |
| $F_1$-Wert (*Precision, Recall*) | **68.3 (68.7, 67.9)** | 65.6 (69.3, 62.3) | 58.6 (74.1, 48.5) | 49.6 (69.6, 38.5) |
| max. *Recall* (*Precision*) | **72.7 (56.8)** | 65.0 (64.8) | 49.6 (70.9) | 38.8 (67.7) |
| Bakis, Fenstergröße 5 | | | | |
| $F_1$-Wert (*Precision, Recall*) | 68.2 (66.3, 70.2) | **70.0 (69.9, 70.0)** | 60.9 (72.3, 52.7) | 52.5 (71.2, 41.6) |
| max. *Recall* (*Precision*) | **74.7 (57.4)** | 70.6 (68.9) | 52.9 (71.8) | 42.1 (68.4) |
| Linear, Fenstergröße 7 | | | | |
| Rückweisungsmodelllänge | 1 | 2 | 3 | 5 |
| $F_1$-Wert (*Precision, Recall*) | **64.8 (71.4, 59.3)** | 64.0 (70.5, 58.6) | 61.1 (72.8, 52.6) | 49.5 (70.7, 38.1) |
| max. *Recall* (*Precision*) | **65.6 (61.2)** | 61.1 (65.0) | 54.5 (67.5) | 39.1 (64.8) |
| Bakis, Fenstergröße 7 | | | | |
| $F_1$-Wert (*Precision, Recall*) | **66.6 (69.3, 64.1)** | 65.6 (68.0, 63.3) | 64.1 (72.7, 57.4) | 49.3 (73.7, 37.1) |
| max. *Recall* (*Precision*) | **69.3 (60.9)** | 65.5 (64.8) | 59.7 (67.8) | 38.0 (68.2) |
| Linear, Fenstergröße 9 | | | | |
| Rückweisungsmodelllänge | 1 | 2 | 3 | 4 |
| $F_1$-Wert (*Precision, Recall*) | **61.6 (65.8, 57.9)** | 52.5 (64.6, 44.2) | 52.5 (69.3, 42.3) | 45.7 (66.1, 34.9) |
| max. *Recall* (*Precision*) | **59.0 (62.4)** | 45.4 (60.6) | 43.1 (65.9) | 35.3 (63.0) |
| Bakis, Fenstergröße 9 | | | | |
| $F_1$-Wert (*Precision, Recall*) | **60.0 (66.0, 55.0)** | 55.9 (71.1, 46.1) | 53.2 (71.8, 42.3) | 50.2 (68.4, 39.7) |
| max. *Recall* (*Precision*) | **58.3 (58.7)** | 49.0 (62.6) | 44.4 (64.4) | 39.9 (67.4) |

Tabelle 46: Ergebnisse für verschiedene Längen des Rückweisungsmodells und unterschiedliche Fenstergrößen bei Verwendung des Modelles eHMM2. Alle Angaben in %.

# LITERATURVERZEICHNIS


[1] Agarwal, A. ; Triggs, B.: Recovering 3D human pose from monocular images. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), S. 44–58 (Zitiert auf Seiten 35, 39, 42, 48, 50 und 52.)

[2] Alon, J. ; Athitsos, V. ; Yuan, Q. ; Sclaroff, S.: Simultaneous Localization and Recognition of Dynamic Hand Gestures. In: *Proc. IEEE Workshop on Motion and Video Computing* Bd. 2. Breckenridge, CO, USA, 2005, S. 254–260 (Zitiert auf Seiten 37, 39, 45 und 51.)

[3] Alon, J. ; Athitsos, V. ; Yuan, Q. ; Sclaroff, S.: A unified framework for gesture recognition and spatiotemporal gesture segmentation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2009), Nr. 9, S. 1685–1699 (Zitiert auf Seiten 37, 39, 40, 45, 51, 54, 120 und 154.)

[4] Ardoe, H. ; Berthilsson, R.: Adaptive Background Estimation Using Intensity Independent Features. In: *Proc. British Machine Vision Conf.* Edinburgh, UK, 2006 (Zitiert auf Seiten 17 und 18.)

[5] Asadi, A. ; Schwartz, R. ; Makhoul, J.: Automatic detection of new words in a large vocabulary continuous speech recognition system. In: *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*. Albuquerque, NM, USA, 1990, S. 125–128 (Zitiert auf Seite 157.)

[6] Bagdat, F. ; Bahne, M. ; Cernat, R. ; Häussler, T. ; Kandelberg, A. ; Knobloch, F. ; Neugebauer, O. ; Niegl, C. ; Plinge, A. ; Schmitz, N. ; Wiggerich, T. ; Wilking, T.: Endbericht der Projektgruppe 525 PARTYBOT: Entwicklung einer Aufmerksamkeitsarchitektur für einen mobilen Roboter auf der Basis von OSGi. / TU Dortmund. 2009. – Forschungsbericht (Zitiert auf Seite 106.)

[7] Bammes, G.: *Die Gestalt des Menschen – ein Handbuch der Anatomie für Künstler*. Verlag Otto Maier GmbH, Ravensburg, 1974 (Zitiert auf Seiten 136 und 138.)

[8] Bay, H. ; Tuytelaars, T. ; van Gool, L.: SURF: Speeded up robust features. In: *Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science* Bd. 3951. Springer, 2006, S. 404–417 (Zitiert auf Seite 64.)

[9] Beis, J. ; Lowe, D. G.: Shape indexing using approximate nearest-neighbor search in high dimensional spaces. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. San Juan, Puerto Rico, 1997, S. 1000–1006 (Zitiert auf Seite 64.)

[10] Bentley, J. L.: Multidimensional binary search trees used for associative searching. In: *Communications of the ACM* 18 (1975), Nr. 9, S. 509–517 (Zitiert auf Seite 117.)

[11] Bhaskar, H. ; Mihaylova, L. ; Maskell, S.: Human Body Party Tracking Using Pictorial Structures and a Genetic Algorithm. In: *Proc. IEEE Int. Conf. on Intelligent Systems*. Varna, Bulgarien, 2008 (Zitiert auf Seiten 35, 39, 40, 41, 46 und 51.)

[12] Bhuyan, M. K. ; Bora, P. K. ; Ghosh, D.: Trajectory Guided Recognition of Hand Gestures having only Global Motions. In: *Int. Journal of Computer Sciences* 3 (2008), Nr. 4, S. 222–233 (Zitiert auf Seiten 37, 42, 45, 50, 51, 114 und 123.)

[13] Bishop, C. M.: *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, NY, 2006 (Zitiert auf Seite 49.)

[14] Bobick, A. F. ; Davis, J. W.: The Recognition of Human Movement Using Temporal Templates. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 3, S. 257–267 (Zitiert auf Seiten 36, 39, 42, 44, 47 und 97.)

[15] Bouguet, J.-Y.: *Camera calibration toolbox for Matlab*. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. – aufgerufen am 21.10.2010 (Zitiert auf Seite 126.)

[16] Bowden, R. ; Mitchell, T. A. ; Sarhadi, M.: Non-linear statistical models for the 3D reconstruction of human pose and motion from monocular image sequences. In: *Image and Vision Computing* 18 (2000), S. 729–737 (Zitiert auf Seiten 35, 39, 41, 47 und 48.)

[17] Bowden, R. ; Windridge, D. ; Kadir, T. ; Zisserman, A. ; Brady, M.: A linguistic feature vector for the visual interpretation of sign language. In: *Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science* Bd. 3021. Springer, 2004, S. 390–401 (Zitiert auf Seiten 37, 39, 41, 44, 45, 47, 50, 52 und 203.)

[18] Bérard, F.: The magic table: Computer-vision based augmentation of a whiteboard for creative meetings. In: *Proc. IEEE ICCV Workshop on Projector-Camera Systems*. Nizza, Frankreich, 2003 (Zitiert auf Seiten 37, 39 und 45.)

[19] Caillette, F. ; Galata, A. ; Howard, T.: Real-time 3-D human body tracking using learnt models of behaviour. In: *Computer Vision and Image Understanding* 109 (2008), Nr. 2, S. 112–125 (Zitiert auf Seiten 36, 39, 42, 45, 46, 49 und 50.)

[20] Calinon, S. ; Billard, A.: Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM. In: *Proc. Int. Conf. on Machine Learning*. Bonn, Deutschland, 2005, S. 105–112 (Zitiert auf Seiten 38, 39, 45 und 50.)

[21] Campbell, L. W. ; Becker, D. A. ; Azarbayejani, A. ; Bobick, A. F. ; Pentland, A.: Invariant features for 3-D gesture recognition. In: *2nd Int. Conf. on Automatic Face and Gesture Recognition*. Killington, VT, USA, 1996, S. 157–162 (Zitiert auf Seiten 148 und 153.)

[22] Camurri, A. ; Canepa, C. ; Ghisio, S. ; Volpe, G.: Automatic classification of expressive hand gestures on tangible acoustic interfaces according to Laban's theory of effort. In: *Gesture-Based Human-Computer Interaction and Simulation, Lecture Notes in Computer Science* Bd. 5085. Springer, 2009, S. 151–162 (Zitiert auf Seite 50.)

[23] Caridakis, G. ; Karpouzis, K. ; Drosopoulos, A. ; Kollias, S.: SOMM: Self organizing Markov map for gesture recognition. In: *Pattern Recognition Letters* 31 (2010), S. 52–59 (Zitiert auf Seiten 37, 39, 45, 47, 50, 148 und 154.)

[24] Chen, C. ; Fan, G.: Hybrid body representation for integrated pose recognition, localization and segmentation. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 35, 39, 40, 41, 47 und 49.)

[25] Chen, F.-S. ; Fu, C.-M. ; Huang, C.-L.: Hand gesture recognition using a real-time tracking method and hidden Markov models. In: *Image and Vision Computing* 21 (2003), Nr. 8, S. 745–758 (Zitiert auf Seiten 37, 39, 42, 45 und 50.)

[26] Chien, C.-Y. ; C.-L., Huang ; Fu, C.-M.: A Vision-based real-time pointing arm gesture tracking and recognition system. In: *Proc. IEEE Int. Conf. on Multimedia and Expo*. Beijing, China, 2007, S. 983–986 (Zitiert auf Seiten 38, 39 und 56.)

[27] Cipolla, R. ; Hadfield, P. A. ; Hollinghurst, N. J.: Uncalibrated stereo vision with pointing for a man-machine interface. In: *Proc. of IAPR Workshop on Machine Vision Applications*. Kawasaki, Japan, 1994, S. 163–166 (Zitiert auf Seiten 38, 40, 41 und 47.)

[28] Cohen, I. ; Li, H.: Inference of human postures by classification of 3D human body shape. In: *Proc. IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures*. Nizza, Frankreich, 2003 (Zitiert auf Seiten 36, 39, 42, 47 und 50.)

[29] Cohen, I. ; Medioni, G. ; Gu, H.: Inference of 3D human body posture from multiple cameras for vision-based user interfaces. In: *Proc. 5th World Multi-Conference on Systemics Cybernetics and Informatics*. Orlando, FL, USA, 2001 (Zitiert auf Seiten 36, 39, 41 und 48.)

[30] Comaniciu, D. ; Meer, P.: Mean Shift: A robust approach toward feature space analysis. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), Nr. 5, S. 603–619 (Zitiert auf Seiten 72, 74 und 75.)

[31] Comaniciu, D. ; Ramesh, V. ; Meer, P.: The variable bandwidth Mean Shift and data-driven scale selection. In: *Proc. Int. Conf. on Computer Vision*. Vancouver, Kanada, 2001, S. 438–445 (Zitiert auf Seite 74.)

[32] Comaniciu, D. ; Ramesh, V. ; Meer, P.: Kernel-based object tracking. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), Nr. 2, S. 564–575 (Zitiert auf Seiten 75, 76, 77 und 78.)

[33] Corso, J. J. ; Ye, G. ; Burschka, D. ; Hager, G. D.: A practical paradigm and platform for video-based human-computer interaction. In: *IEEE Computer* 41 (2008), Nr. 5, S. 48–55 (Zitiert auf Seiten 38, 39, 42, 44, 50 und 57.)

[34] Corso, J. J. ; Ye, G. ; Hager, G. D.: Analysis of composite gestures with a coherent probabilistic graphical model. In: *Virtual Reality* 8 (2005), Nr. 4, S. 242–252 (Zitiert auf Seiten 38, 42, 45 und 47.)

[35] Dadgostar, F. ; Sarrafzadeh, A.: A component-based architecture for vision-based gesture recognition. In: *Proc. Image and Vision Computing New Zealand Conf.* Dunedin, Neuseeland, 2005 (Zitiert auf Seiten 37, 39, 42, 45, 50 und 114.)

[36] Daifallah, K. ; Zarka, N. ; Jamous, H.: Recognition-based segmentation algorithm for on-line arabic handwriting. In: *Proc. Int. Conf. on Document Analysis and Recognition*. Barcelona, Spanien, 2009, S. 886–890 (Zitiert auf Seite 154.)

[37] Dalal, N. ; Triggs, B.: Histograms of oriented gradients for human detection. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. San Diego, CA, USA, 2005, S. 886–893 (Zitiert auf Seiten 66, 68, 71, 106 und 178.)

[38] Daubney, B. ; Gibson, D. ; Campbell, N.: Real-time pose estimation of articulated objects using low-level motion. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 35, 42, 46 und 49.)

[39] Daubney, B. ; Xie, X.: Estimating 3D human pose from single images using iterative refinement of the prior. In: *Proc. Int. Conf. on Pattern Recognition*. Istanbul, Türkei, 2010 (Zitiert auf Seiten 35, 39, 43, 46 und 49.)

[40] Deans, M. ; Kunz, C. ; Sargent, R. ; Park, E. ; Pedersen, L.: Combined feature based and shape based visual tracker for robot navigation. In: *Proc. IEEE Aerospace Conf.* Big Sky, MT, USA, 2005 (Zitiert auf Seite 64.)

[41] Delponte, E. ; Isgro, F. ; Odone, F. ; Verri, A.: SVD-Matching using SIFT features. In: *Graphical Models* 68 (2006), Nr. 5-6, S. 415–431 (Zitiert auf Seite 64.)

[42] Dollàr, P. ; Rabaud, V. ; Cottrell, G. ; Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: *Proc. IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. Beijing, China, 2005, S. 65–72 (Zitiert auf Seiten 36, 40, 43, 47 und 52.)

[43] Duda, R. O. ; Hart, P. E. ; Stork, D. G.: *Pattern Classification*. John Wiley & Sons, Inc., 2001 (Zitiert auf Seiten 13, 14, 77, 80, 83, 84, 85 und 86.)

[44] Durbin, R. ; Eddy, S. ; Krogh, A. ; Mitchison, G.: *Biological sequence analysis*. Cambridge University Press, 1998 (Zitiert auf Seite 158.)

[45] Ehrenmann, M. ; Luetticke, T. ; Dillmann, R.: Directing a mobile robot with dynamic gestures. In: *Proc. IEEE Int. Conf. on Robotics and Automation*. Seoul, Korea, 2001, S. 2596–2601 (Zitiert auf Seiten 37, 39, 45, 50 und 154.)

[46] Eisenstein, Jacob ; Davis, Randall: Visual and linguistic information in gesture classification. In: *Proc. Int. Conf. on Multimodal Interfaces*. State College, PA, USA, 2004, S. 113–120 (Zitiert auf Seite 24.)

[47] Ekman, P. ; Friesen, W.V.: The repertoire of nonverbial behavior: Categories, origins, usage and coding. In: *Semiotica* 1 (1969), S. 49–98 (Zitiert auf Seite 24.)

[48] Elgammal, A. ; Harwood, D. ; Davis, L. S.: Non-parametric model for background subtraction. In: *Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science* Bd. 1843. Springer, 2000 (Zitiert auf Seite 18.)

[49] Elmezain, M. ; Al-Hamadi, A. ; Appenrodt, J. ; Michaelis, B.: A hidden Markov model-based continuous gesture recognition system for hand motion trajectory. In: *Proc. Int. Conf. on Pattern Recognition*. Tampa, FL, USA, 2008 (Zitiert auf Seiten 38, 39 und 50.)

[50] Elmezain, M. ; Al-Hamadi, A. ; Michaelis, B.: A robust method for hand gesture segmentation and recognition using forward spotting scheme in conditional random fields. In: *Proc. Int. Conf. on Pattern Recognition*. Istanbul, Türkei, 2010 (Zitiert auf Seiten 114 und 213.)

[51] Faugeras, O. ; Anagnostopoulos, P (Hrsg.) ; Snowden, J. (Hrsg.): *Three-dimensional computer vision - a geometric viewpoint*. Cambridge, Ma. : MIT Press, 1993 (Zitiert auf Seiten 20, 45 und 134.)

[52] Fawcett, T.: An introduction to ROC analysis. In: *Pattern Recognition Letters* 27 (2006), S. 861–874 (Zitiert auf Seite 177.)

[53] Felzenszwalb, P. ; McAllester, D. ; Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seite 71.)

[54] Felzenszwalb, P. F. ; Huttenlocher, D. P.: Pictorial structures for object recognition. In: *Int. Journal of Computer Vision* 61 (2005), Nr. 1, S. 55–79 (Zitiert auf Seiten 35, 39, 40, 41, 46, 47 und 49.)

[55] Ferrari, V. ; Marin-Jimenez, M. ; Zisserman, A.: Progressive search space reduction for human pose estimation. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 35, 40, 42, 43, 44, 46, 49, 71 und 106.)

[56] Filipovych, R. ; Ribeiro, E.: Learning human motion models from unsegmented video. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 36, 40, 42, 43, 47 und 49.)

[57] Fink, G. A.: *Markov models for pattern recognition*. Springer, 2008 (Zitiert auf Seiten 15, 86, 88, 89, 100, 144 und 156.)

[58] Fink, G. A. ; Plötz, T.: Developing pattern recognition systems based on Markov models: The ESMERALDA framework. In: *Pattern Recognition and Image Analysis* 18 (2008), Nr. 2, S. 207–215 (Zitiert auf Seiten 158 und 174.)

[59] Fink, G. A. ; Wienecke, M. ; Sagerer, G.: Video-based on-line handwriting recognition. In: *Proc. Int. Conf. on Document Analysis and Recognition*. Seattle, WA, USA, 2001, S. 226–230 (Zitiert auf Seite 154.)

[60] Fischler, M. A. ; Bolles, R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381–395 (Zitiert auf Seite 150.)

[61] Fukunaga, K. ; Hostetler, L. D.: The estimation of the gradient of a density function, with applications in pattern recognition. In: *IEEE Transactions on Information Theory* 21 (1975), S. 32–40 (Zitiert auf Seite 72.)

[62] Fusier, F. ; Valentin, V. ; Brémond, F. ; Thonnat, M. ; Borg, M. ; Thirde, D. ; Ferryman, J.: Video understanding for complex activity recognition. In: *Machine Vision and Applications* 18(3) (2007), S. 167–188 (Zitiert auf Seite 35.)

[63] Gao, Y. ; Radha, H.: A multistage camera self-calibration algorithm. In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*. Montreal, Kanada, 2004 (Zitiert auf Seite 126.)

[64] Gonzales, R. C. ; Woods, R. E. ; McDonald, M. (Hrsg.) ; Dworkin, A. (Hrsg.) ; Opaluch, W. (Hrsg.) ; Disanno, S. (Hrsg.) ; Kernan, R. (Hrsg.): *Digital image processing*. Pearson Education Inc., 2008 (Zitiert auf Seiten 11 und 64.)

[65] Gorelick, L. ; Blank, M. ; Shechtman, E. ; Irani, M. ; Basri, R.: Actions as space-time shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007), Nr. 12, S. 2247–2253 (Zitiert auf Seiten 36, 39, 42, 43 und 50.)

[66] Gorry, B. ; Chen, Z. ; Hammond, K. ; Wallace, A. ; Michaelson, G.: Using mean-shift tracking algorithms for real-time tracking of moving images on an autonomous vehicle testbed platform. In: *World Academy of Science, Engineering and Technology* 34 (2007), S. 209–214 (Zitiert auf Seite 108.)

[67] Graves, A. ; Liwicki, M. ; Fernandez, S. ; Bertolami, R. ; Bunke, H. ; Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2009), Nr. 5, S. 855–868 (Zitiert auf Seiten 154 und 156.)

[68] Gross, H.-M. ; Mueller, S. ; Richarz, J. ; Martin, C. ; Scheidig, A.: Probabilistic multi-modal people tracker and monocular pointing pose estimator for

visual instruction of mobile robot assistants. In: *Proc. IEEE World Congress on Computational Intelligence, Int. Joint Conf. on Neural Networks*. Vancouver, Kanada, 2006, S. 8325–8333 (Zitiert auf Seite 38.)

[69] Grundmann, M. ; Meier, F. ; Essa, I.: 3D shape context and distance transform for action recognition. In: *Proc. Int. Conf. on Pattern Recognition*. Tampa, FL, USA, 2008 (Zitiert auf Seiten 36, 39, 42, 43 und 47.)

[70] Guan, Y.: Stereo vision based video real-time 3D pointing gesture recognition. In: *Proc. IET Conf. on Wireless, Mobile and Sensor Networks*. Shanghai, China, 2007, S. 355–358 (Zitiert auf Seite 160.)

[71] Guan, Y. ; Zhang, M.: Real-time 3D pointing gesture recognition for natural HCI. In: *Proc. World Congress on Intelligent Control and Automation*. Chongqing, China, 2008, S. 2433–2436 (Zitiert auf Seiten 38, 39 und 41.)

[72] Gupta, A. ; Chen, T. ; Chen, F. ; Kimber, D. ; Davis, L. S.: Context and observation driven latent variable model for human pose estimation. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 35, 39, 42, 48 und 49.)

[73] Hartley, R. ; Zisserman, A.: *Multiple view geometry in computer vision*. Cambridge University Press, 2004 (Zitiert auf Seiten 18 und 19.)

[74] Hemayed, E. E.: A survey of camera self-calibration. In: *Proc. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. Miami, FL, USA, 2003 (Zitiert auf Seite 126.)

[75] Heymann, S. ; Müller, K. ; Smolic, A. ; Fröhlich, B. ; Wiegand, T.: SIFT implementation and optimization for general-purpose GPU. In: *Proc. 15th Int. Conf. on Computer Graphics, Visualization and Computer Vision*. Lissabon, Portugal, 2007 (Zitiert auf Seite 64.)

[76] Hilliges, O. ; Izadi, S. ; Wilson, A. D. ; Hodges, S. ; Garcia-Mendoza, a. ; Butz, A.: Interactions in the air: adding further depth to interactive tabletops. In: *Proc. ACM Symposium on user interface software and technology*. Victoria, Kanada, 2009 (Zitiert auf Seite 57.)

[77] Hofemann, N. ; Fritsch, J. ; Sagerer, G.: Recognition of deictic gestures with context. In: *Proc. 26th DAGM Symposium, Lecture Notes in Computer Science* Bd. 3175. Springer, 2004, S. 334–341 (Zitiert auf Seiten 37, 38, 39, 45, 114, 123 und 160.)

[78] Hofmann, M. ; Gavrila, D. M.: Single-frame 3D human pose recovery from multiple views. In: *Proc. 31st DAGM Symposium, Lecture Notes in Computer Science* Bd. 5748. Springer, 2009, S. 71–80 (Zitiert auf Seiten 36, 39, 41, 47, 48, 50 und 51.)

[79] Horowitz, S. L. ; Pavlidis, T.: Picture segmentation by a tree traversal algorithm. In: *Journal of the ACM* 23 (1976), Nr. 2, S. 368–388 (Zitiert auf Seite 104.)

[80] Hu, W. ; Tan, L. T.and W. T.and Wang ; Maybank, S.: A survey on visual surveillance on object motion and behaviors. In: *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews* 34 (2004), S. 334–352 (Zitiert auf Seite 35.)

[81] Ikizler, N. ; Duygulu, P.: Human action recognition using distribution of oriented rectangular patches. In: *Proc. 2nd Workshop on Human Motion Understanding, Modeling, Capture and Animation, Lecture Notes in Computer Science* Bd. 4814. Springer, 2007, S. 271–284 (Zitiert auf Seiten 36, 39, 42, 43, 44, 47, 50, 51, 52 und 71.)

[82] Intel Corporation: *Object-aware situated interactive system (OASIS)*. `http://software.intel.com/en-us/videos/oasis-object-aware-situated-interactive-system/`. – aufgerufen am 06.04.2011 (Zitiert auf Seite 220.)

[83] Jaeger, S. ; Manke, S. ; Reichert, J. ; Waibel, A.: Online handwriting recognition: The NPen++ recognizer. In: *Int. Journal on Document Analysis and Recognition* 3 (2001), S. 169–180 (Zitiert auf Seiten 154 und 156.)

[84] Jiang, H. ; Martin, D. R.: Finding actions using shape flows. In: Forsyth, D. (Hrsg.) ; Torr, P. (Hrsg.) ; Zisserman, A. (Hrsg.): *Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science* Bd. 5303. Springer, 2008, S. 278–292 (Zitiert auf Seiten 36, 41, 42, 45, 47 und 51.)

[85] Jojic, N. ; Brumitt, B. ; Meyers, B. ; Harris, S. ; Huang, T.: Detecting and estimating pointing gestures in dense disparity maps. In: *Proc. IEEE Int. Conf. on Face and Gesture Recognition*. Grenoble, Frankreich, 2000 (Zitiert auf Seiten 38, 39, 41, 49 und 51.)

[86] Junker, H. ; Amft, O. ; Lukowicz, P. ; Tröster, G.: Gesture spotting with body-worn inertial sensors to detect user activities. In: *Pattern Recognition* 41 (2008), S. 2010–2024 (Zitiert auf Seiten 33 und 213.)

[87] Kakumanu, P. ; Makrogiannis, S. ; Bourbakis, N.: A survey of skin-color modeling and detection methods. In: *Pattern Recognition* 40 (2007), S. 1106–1122 (Zitiert auf Seiten 119 und 190.)

[88] Kaâniche, M. B. ; Brémond, F.: Gesture recognition by learning local motion signatures. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. San Francisco, CA, USA, 2010 (Zitiert auf Seiten 36, 39, 40, 43, 44 und 50.)

[89] Ke, Y. ; Sukthangar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Washington, DC, USA, 2004 (Zitiert auf Seite 64.)

[90] Kehl, R. ; Bray, M. ; van Gool, L.: Full body tracking from multiple views using stochastic sampling. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. San Diego, CA, USA, 2005, S. 129–136 (Zitiert auf Seiten 36, 39, 42 und 46.)

[91] Kehl, R. ; Van Gool, L.: Real-time pointing gesture recognition for an immersive environment. In: *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*. Southampton, UK, 2004, S. 577–582 (Zitiert auf Seiten 38, 39, 41, 45, 49 und 160.)

[92] Kendon, A.: Current issues in the study of gestures. In: *The Biological Foundation of Gestures. Motor and Semiotic Aspects.* Lawrence Erlbaum Assoc., 1986, S. 23–47 (Zitiert auf Seite 24.)

[93] Kim, D. ; Song, J. ; Kim, D.: Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs. In: *Pattern Recognition* 40 (2007), S. 3012–3026 (Zitiert auf Seiten 38, 39, 44, 48, 50 und 55.)

[94] Kiratiratanapruk, K. ; Dubey, P. ; Siddhichai, S.: A gradient-based foreground detection technique for object tracking in a traffic monitoring system. In: *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*. Genua, Italien, 2005, S. 377–381 (Zitiert auf Seite 18.)

[95] Kirishima, T. ; Sato, K. ; Chihara, K.: Real-time gesture recognition by learning and selective control of visual interest points. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), March, Nr. 3, S. 351–364 (Zitiert auf Seiten 37, 39, 42, 44, 47 und 50.)

[96] KNX Association: *KNX Standard*. `http://www.knx.org/knx-standard/introduction/`. Version: 2010. – aufgerufen am 21.09.2010 (Zitiert auf Seite 93.)

[97] KOPF, S. ; HAENSELMANN, T. ; EFFELSBERG, W.: Shape-based posture and gesture recognition in videos. In: *Storage and Retrieval Methods and Applications for Multimedia* Bd. 5682, 2005 (Proc. of SPIE), S. 114–124 (Zitiert auf Seiten 35, 37, 39, 42, 47 und 50.)

[98] KRANSTEDT, A. ; LÜCKING, A. ; PFEIFFER, T. ; RIESER, H. ; WACHSMUTH, I.: Deixis: How to determine demonstrated objects using a pointing cone. In: *Gesture in Human-Computer Interaction and Simulation, 6th Int. Gesture Workshop, Lecture Notes in Computer Science* Bd. 3881. Springer, 2006 (Zitiert auf Seiten 161 und 163.)

[99] KUNCHEVA, L. I.: *Combining pattern classifiers*. Hoboken, New Jersey : John Wiley & Sons, Inc., 2004 (Zitiert auf Seite 77.)

[100] LAPTEV, I.: On space-time interest points. In: *Int. Journal of Computer Vision* 64 (2005), Nr. 2, S. 107–123 (Zitiert auf Seiten 36, 40, 42, 43, 47, 51 und 52.)

[101] LAPTEV, I.: Improving object detection with boosted histograms. In: *Image and Vision Computing* 27 (2009), April, Nr. 5, S. 535–544 (Zitiert auf Seite 71.)

[102] LAPTEV, I. ; MARSZALEK, M. ; SCHMID, C. ; ROZENFELD, B.: Learning realistic human actions from movies. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008, S. 1–8 (Zitiert auf Seiten 36, 40, 43, 47, 50, 52, 71 und 106.)

[103] LI, H. ; GREENSPAN, M.: Multi-scale gesture recognition from time-varying contours. In: *Proc. Int. Conf. on Computer Vision* Bd. 1. Beijing, China, 2005, S. 236–243 (Zitiert auf Seiten 37, 39, 42, 44, 51 und 53.)

[104] LI, Zhe ; HOFEMANN, Nils ; FRITSCH, Jannik ; SAGERER, Gerhard: Hierarchical modeling and recognition of manipulative gesture. In: *Proc. ICCV Workshop on Modeling People and Human Interaction*. Beijing, China, 2005 (Zitiert auf Seiten 37, 39, 45 und 50.)

[105] LICSAR, A. ; SZIRANYI, T.: Hand gesture recognition in camera-projector system. In: *Computer vision in human-computer interaction: ECCV 2004 workshop on HCI, Lecture Notes in Computer Science* Bd. 3058. Springer, 2004, S. 83–93 (Zitiert auf Seiten 37, 39, 42, 44, 47, 50 und 154.)

[106] LINDEBERG, T.: Scale-space theory: A basic tool for analysing structures at different scales. In: *Journal of Applied Statistics* 21 (1994), Nr. 2, S. 224–270 (Zitiert auf Seiten 61 und 62.)

[107] Liu, J. ; Hubbold, R.: Automatic camera calibration and scene reconstruction with scale-invariant features. In: *Advances in Visual Computing, Lecture Notes in Computer Science* Bd. 4291. Springer, 2006, S. 558–568 (Zitiert auf Seite 64.)

[108] Liu, J. ; Saad, A. ; Shah, M.: Recognizing human actions using multiple features. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 36, 39, 40, 42, 43 und 52.)

[109] Lockton, R. ; Fitzgibbon, A. W.: Real-time gesture recognition using deterministic boosting. In: *Proc. British Machine Vision Conf.* Cardiff, UK, 2002, S. 817–826 (Zitiert auf Seiten 37, 39, 40, 47 und 50.)

[110] Lowe, D. G.: Object recognition from local scale-invariant features. In: *Proc. Int. Conf. on Computer Vision*. Kerkyra, Griechenland, 1999, S. 1150–1157 (Zitiert auf Seite 64.)

[111] Lowe, D. G.: Distinctive image features from scale-invariant keypoints. In: *Int. Journal of Computer Vision* 60 (2004), Nr. 2, S. 91–110 (Zitiert auf Seiten 61, 62 und 64.)

[112] Luo, X. ; Berendsen, B. ; Tan, R. T. ; Veltkamp, R. C.: Human pose estimation for multiple persons based on volume reconstruction. In: *Proc. Int. Conf. on Pattern Recognition*. Istanbul, Türkei, 2010 (Zitiert auf Seiten 36, 39, 41 und 46.)

[113] Malgireddy, M. R. ; Corso, J. J. ; Setlur, S. ; Govindaraju, V. ; Mandalapu, D.: A framework for hand gesture recognition and spotting using sub-gesture modeling. In: *Proc. Int. Conf. on Pattern Recognition*. Istanbul, Türkei, 2010 (Zitiert auf Seiten 37, 41, 45, 50 und 54.)

[114] Marszalek, M. ; Laptev, I. ; Schmid, C.: Actions in context. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Miami, FL, USA, 2009 (Zitiert auf Seiten 31 und 37.)

[115] Martin, J. ; Crowley, J. L.: An appearance-based approach to gesture-recognition. In: *Proc. 9th Int. Conf. on Image Analysis and Processing, Lecture Notes in Computer Science* Bd. 1311. Springer, 1997, S. 340–347 (Zitiert auf Seiten 39, 42, 44, 47 und 50.)

[116] McCulloch, W. S. ; Pitts, W.: A logical calculus of the ideas immanent in nervous activity. In: *Bulletin of Mathematical Biophysics* 5 (1943), S. 115–133 (Zitiert auf Seite 81.)

[117] McKenna, S. J. ; Raja, Y. ; Gong, S.: Tracking colour objects using adaptive mixture models. In: *Image and Vision Computing* 17 (1999), S. 225–231 (Zitiert auf Seite 100.)

[118] Medioni, G. ; Cohen, I. ; Bremond, F. ; Hongeng, S. ; Nevatia, R.: Event detection and analysis from video streams. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001), S. 873–889 (Zitiert auf Seite 35.)

[119] Meyberg, K. ; Vachenauer, P.: *Höhere Mathematik*. Bd. 1. Springer, 1999 (Zitiert auf Seite 150.)

[120] Micilotta, A. S. ; Ong, E.-J. ; Bowden, R.: Real-time upper body detection and 3D pose estimation in monoscopic images. In: *Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science* Bd. 3953. Graz, Austria : Springer, 2006, S. 139–150 (Zitiert auf Seiten 35, 39, 40, 41, 47 und 50.)

[121] Microsoft Corporation: *Projekt Natal*. `http://www.xbox.com/de-DE/news-features/news/Controller-free-gaming-htm`. Version: 2010. – abgerufen am 06.10.2010 (Zitiert auf Seiten 53 und 220.)

[122] Mikic, E. ; Trivedi, M. M. ; Hunter, E. ; Cosman, P. C.: Human body model acquisition and motion capture using voxel data. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Kauai, HI, USA, 2001, S. 104–118 (Zitiert auf Seiten 36, 39, 41 und 48.)

[123] Mikolajczyk, K. ; Schmid, C.: A performance evaluation of local descriptors. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), Nr. 10, S. 1615–1630 (Zitiert auf Seite 14.)

[124] Mikolajczyk, K. ; Tuytelaars, T. ; Schmid, C. ; Zisserman, A. ; Matas, J. ; Schaffalitzky, F. ; Kadir, T. ; van Gool, L.: A comparison of affine region detectors. In: *Int. Journal of Computer Vision* 65 (2005), Nr. 1/2, S. 43–72 (Zitiert auf Seite 14.)

[125] Mitra, S. ; Acharya, T.: Gesture recognition: A survey. In: *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews* 37 (2007), S. 311–324 (Zitiert auf Seite 34.)

[126] Mittal, A. ; Zhao, L. ; Davis, L. S.: Human body pose estimation using silhouette shape analysis. In: *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*. Miami, FL, USA, 2003, S. 263–270 (Zitiert auf Seiten 36, 39, 41, 42, 48 und 49.)

[127] Moeslund, T.B. ; Granum, E.: A survey of computer vision-based human motion capture. In: *Computer Vision and Image Understanding* 81(3) (2001), S. 231–268 (Zitiert auf Seite 33.)

[128] Moeslund, T.B. ; Hilton, A. ; Krüger, V.: A survey of advances in vision-based human motion capture and analysis. In: *Computer Vision and Image Understanding* 104 (2006), S. 90–126 (Zitiert auf Seiten 33 und 34.)

[129] Moore, D. J. ; Essa, I. A. ; Hayes, M. H.: Exploiting human actions and object context for recognition tasks. In: *Proc. Int. Conf. on Computer Vision*. Kerkyra, Griechenland, 1999 (Zitiert auf Seite 31.)

[130] Natarajan, P. ; Nevatia, R.: View and scale invariant action recognition using multiview shape-flow models. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 36, 41, 44, 45, 47 und 50.)

[131] Navaratnam, R. ; Thayananthan, A. ; Torr, P. H. S. ; Cipolla, R.: Hierarchical part-based human body pose estimation. In: *Proc. British Machine Vision Conf.* London, UK, 2005 (Zitiert auf Seiten 35, 40, 42, 44, 46 und 51.)

[132] Nickel, K. ; Stiefelhagen, R.: Visual recognition of pointing gestures for human-robot interaction. In: *Image and Vision Computing* 25 (2007), Nr. 12, S. 1875–1884 (Zitiert auf Seiten 38, 39, 40, 45, 47, 50, 56, 114, 120, 148 und 160.)

[133] Niemann, H.: *Klassifikation von Mustern*. 2003. – http://www5.informatik.uni-erlangen.de/fileadmin/Persons/NiemannHeinrich/klassifikation-von-mustern/moolinks.html (Zitiert auf Seiten 9, 10 und 12.)

[134] Ning, H. ; Xu, W. ; Gong, S. ; Huang, T.: Discriminative learning of visual words for 3D human pose estimation. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 35, 40, 43, 44, 48, 49 und 52.)

[135] Ning, H. ; Xu, W. ; Gong, S. ; Huang, T.: Latent pose estimator for continuous action recognition. In: *Proc. European Conf. on Computer Vision*. Marseille, Frankreich, 2008, S. 419–433 (Zitiert auf Seiten 36, 39, 40, 43, 44, 50 und 52.)

[136] Nissen, S.: Implementation of a fast artificial neural network library (FANN) / Universität Kopenhagen (DIKU). 2003. – Forschungsbericht (Zitiert auf Seite 176.)

[137] Noelker, C. ; Ritter, H.: Illumination independent recognition of deictic arm postures. In: *Proc. 24th Annual Conf. of the IEEE Industrial Electronics Society*. Aachen, Germany, 1998, S. 206–2011 (Zitiert auf Seiten 38, 42 und 50.)

[138] Noriega, P. ; Bernier, O.: Real time illumination invariant background subtraction using local kernel histograms. In: *Proc. Int. Conf. on Computer Vision Theory and Applications*. Setúbal, Portugal, 2006 (Zitiert auf Seite 18.)

[139] Nöth, Winfried: *Handbuch der Semiotik*. J.B. Metzler, Stuttgart, 2000 (Zitiert auf Seiten 23 und 24.)

[140] Oikonomopoulos, A. ; Patras, I. ; Pantic, M.: Spatiotemporal salient points for visual recognition of human actions. In: *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 36 (2006), June, Nr. 3, S. 710–719 (Zitiert auf Seiten 36, 40, 43 und 50.)

[141] Ong, E.-J. ; Bowden, R.: A boosted classifier tree for hand shape detection. In: *Proc. 6th IEEE Int. Conf. on Automatic Face and Gesture Recognition*. Seoul, Korea, 2004, S. 889–894 (Zitiert auf Seiten 37, 39, 40, 42, 47 und 50.)

[142] Ong, S.C.W. ; Ranganath, S.: Automatic sign language analysis: A survey and the future beyond lexical meaning. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), Nr. 6, S. 873–891 (Zitiert auf Seite 34.)

[143] Park, C.-B. ; Roh, M.-C. ; Lee, S.-W.: Real-time 3D pointing gesture recognition in mobile space. In: *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*. Amsterdam, Niederlande, 2008, S. 1–6 (Zitiert auf Seiten 38, 39, 40, 45, 50 und 114.)

[144] Peng, N. S. ; Yang, J. ; Liu, Z.: Mean shift blob tracking with kernel histogram filtering and hypothesis testing. In: *Pattern Recognition Letters* 26 (2005), S. 605–614 (Zitiert auf Seite 108.)

[145] Pla, F. ; Ribeiro, P. ; Santos-Victor, J. ; Bernardino, A.: Extracting motion features for visual human activity representation. In: *Proc. 2nd Iberian Conf. on Pattern Recognition and Image Analysis*. Estoril, Portugal, 2005, S. 537–544 (Zitiert auf Seiten 36, 40 und 45.)

[146] Plamondon, R. ; Srihari, S. N.: On-line and off-line handwriting recognition: A comprehensive survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), S. 63–84 (Zitiert auf Seite 154.)

[147] Pless, R.: Spatio-temporal background models for outdoor surveillance. In: *EURASIP Journal on Applied Signal Processing* 2005 (2005), S. 2281–2291 (Zitiert auf Seite 18.)

[148] Plötz, T. ; Fink, G. A.: An efficient method for making un-supervised adaptation of HMM-based speech recognition systems robust against out-of-domain data. In: *Proc. 4th Int. Workshop on Natural Language Processing and Cognitive Science*. Funchal, Portugal, 2007 (Zitiert auf Seite 158.)

[149] Plötz, T. ; Richarz, J. ; Fink, G. A.: Robust hand detection in still video images using a combination of salient regions and color cues for interaction with an intelligent environment. In: *Pattern Recognition and Image Analysis* 18 (2008), Nr. 3, S. 417–430 (Zitiert auf Seiten 125 und 192.)

[150] Poppe, R.: A survey on vision-based human action recognition. In: *Image and Vision Computing* 28 (2010), S. 976–990 (Zitiert auf Seite 33.)

[151] Qu, H. ; Wang, L. ; Leckie, C.: Action recognition using space-time shape difference images. In: *Proc. Int. Conf. on Pattern Recognition*. Istanbul, Türkei, 2010 (Zitiert auf Seiten 36, 41, 44, 47 und 52.)

[152] Radke, R. J. ; Andra, S. ; Al-Kofahi, O. ; Roysam, B.: Image change detection algorithms: A systematic survey. In: *IEEE Transactions on Image Processing* 14 (2005), Nr. 3, S. 294–307 (Zitiert auf Seite 16.)

[153] Rahman, M. M. ; Ishikawa, S.: Appearance-based representation and recognition of human motions. In: *Proc. IEEE Int. Conf. on Robotics and Automation*, 2003 (1410–1415), S. Taipei, Taiwan (Zitiert auf Seiten 36, 42, 45 und 51.)

[154] Ramanan, D.: Learning to parse images of articulated bodies. In: *Advances in Neural Information Processing Systems* Bd. 19. Cambridge, MA, USA, 2006 (Zitiert auf Seiten 35, 40, 42 und 47.)

[155] Rapantzikos, K. ; Avrithis, Y. ; Kollias, S.: Dense saliency-based spatiotemporal feature points for action recognition. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Miami, FL, USA, 2009, S. 1454–1461 (Zitiert auf Seiten 36, 41, 42, 43, 47 und 52.)

[156] Raza Ali, M. ; Morris, T.: Skin locus based skin detection for gesture recognition. In: *Proc. British Machine Vision Conf. Postgraduate Workshop*. Aberystwyth, UK, 2010 (Zitiert auf Seite 121.)

[157] Rett, J. ; Dias, J.: Gesture recognition using a marionette model and dynamic Bayesian networks (DBNs). In: *Proc. Int. Conf. on Image Analysis and Recognition*. Póvoa de Varzim, Portugal, 2006, S. 69–80 (Zitiert auf Seite 150.)

[158] Richarz, J. ; Fink, G. A.: Feature representations for the recognition of 3D emblematic gestures. In: *Proc. Int. Workshop on Human Behavior Understanding, in conjunction with 20th Int. Conf. on Pattern Recognition, Lecture Notes in Computer Science*. Springer, 2010 (6219), S. 193–211 (Zitiert auf Seiten 202, 203 und 205.)

[159] Richarz, J. ; Fink, G. A.: Visual recognition of 3D emblematic gestures in an HMM framework. In: *Journal of Ambient Intelligence and Smart Environments, Thematic Issue on Computer Vision for Ambient Intelligence* 3 (2011), Nr. 3, S. 193–211 (Zitiert auf Seite 141.)

[160] Richarz, J. ; Martin, C. ; Scheidig, A. ; Gross, H.-M.: There you go! - Estimating pointing gestures in monocular images for mobile robot control. In: *Proc. Int. Symposium on Robot and Human Interactive Communication*. Hatfield, UK, 2006, S. 546–551 (Zitiert auf Seite 31.)

[161] Richarz, J. ; Plötz, T. ; Fink, G. A.: Integration of structural and color cues for robust hand detection in video images. In: *Proc. 7th Open German/Russian Workshop on Pattern Recognition and Image Understanding*. Ettlingen, Deutschland, 2007 (Zitiert auf Seite 125.)

[162] Richarz, J. ; Plötz, T. ; Fink, G. A.: Detecting hands in video images using scale invariant local descriptors. In: *Proc. IASTED Int. Conf. on Visualization, Imaging and Image Processing*. Palma de Mallorca, Spanien, 2007, S. 259–264 (Zitiert auf Seiten 125 und 192.)

[163] Richarz, J. ; Plötz, T. ; Fink, G. A.: Real-time detection and interpretation of 3D deictic gestures for interaction with an intelligent environment. In: *Proc. Int. Conf. on Pattern Recognition*. Tampa, FL, USA, 2008 (Zitiert auf Seiten 159, 160, 163 und 168.)

[164] Richarz, J. ; Scheidig, A. ; Martin, C. ; Mueller, S. ; Gross, H.-M.: A monocular pointing pose estimator for gestural instruction of a mobile robot. In: *International Journal of Advanced Robotic Systems* 4 (2007), Nr. 1, S. 139–150 (Zitiert auf Seiten 37, 38, 40, 42 und 50.)

[165] Riedmiller, M. ; Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: *Proc. IEEE Int. Conf. on Neural Networks*. San Francisco, CA, USA, 1993, S. 586–591 (Zitiert auf Seite 176.)

[166] Rittscher, J. ; Kato, J. ; Joga, S. ; Blake, A.: A probabilistic background model for tracking. In: *Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science* Bd. 1843. Springer, 2000, S. 336–350 (Zitiert auf Seite 18.)

[167] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. In: *Psychological Review* 65 (1958), Nr. 6, S. 386–408 (Zitiert auf Seite 82.)

[168] Satpathy, A. ; Eng, H.-L. ; Jiang, X.: Difference of Gaussian edge-texture based background modeling for dynamic traffic conditions. In: Bebis, G. (Hrsg.): *Proc. Int. Symposium on Visual Computing, Lecture Notes in Computer Science* Bd. 1. Springer, 2008, S. 406–417 (Zitiert auf Seite 18.)

[169] Schauerte, B. ; Richarz, J. ; Fink, G. A.: Saliency-based identification and recognition of pointed-at objects. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Taipei, Taiwan, 2010 (Zitiert auf Seiten 31, 38, 159, 160, 161 und 163.)

[170] Schauerte, B. ; Richarz, J. ; Plötz, T. ; Thurau, C. ; Fink, G. A.: Multi-modal and multi-camera attention in smart environments. In: *Proc. Int. Conf. on Multimodal Interfaces and Workshop on Machine Learning for Multi-Modal Interaction (ICMI-MLMI)*. Boston, MA, USA, 2009, S. 261–268 (Zitiert auf Seiten 91 und 132.)

[171] Schenk, J. ; Kaiser, M. ; Rigoll, G.: Selecting features in on-line handwritten whiteboard note recognition: SFS or SFFS? In: *Proc. Int. Conf. on Document Analysis and Recognition*. Barcelona, Spanien, 2009, S. 1251–1254 (Zitiert auf Seite 154.)

[172] Schindler, K. ; van Gool, L.: Action snippets: How many frames does human action recognition require? In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 36, 40, 42, 44, 45, 47, 50 und 52.)

[173] Schindler, K. ; van Gool, L.: Combining densely sampled form and motion for human action recognition. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 36, 40, 42, 44, 45, 50 und 52.)

[174] Schüldt, C. ; Laptev, I. ; Caputo, B.: Recognizing human actions: A local SVM approach. In: *Proc. Int. Conf. on Pattern Recognition*. Cambridge, UK, 2004, S. 32–36 (Zitiert auf Seite 36.)

[175] SCHMIDT, J. ; FRITSCH, J. ; KWOLEK, B.: Kernel particle filter for real-time 3D body tracking in monocular color images. In: *Proc.7th Int. Conf. on Automatic Face and Gesture Recognition*. Southampton, UK, 2006, S. 567–572 (Zitiert auf Seiten 35, 42, 45 und 46.)

[176] SCHMIDT, J. ; HOFEMANN, N. ; HAASCH, A. ; FRITSCH, J. ; SAGERER, G.: Interacting with a mobile robot: evaluating gestural object references. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Nizza, Frankreich, 2008 (Zitiert auf Seiten 37, 38, 39, 42, 45, 46, 50, 160 und 213.)

[177] SCHOELKOPF, B. ; SMOLA, A. J. ; DIETTERICH, Thomas (Hrsg.): *Learning with kernels*. Cambridge, London : MIT Press, 2002 (Zitiert auf Seite 72.)

[178] SHAMAIE, A. ; SUTHERLAND, A.: Accurate recognition of large number of hand gestures. In: *Proc. 2nd Iranian Conf. on Machine Vision and Image Processing*, 2003 (Zitiert auf Seiten 37, 39, 45 und 50.)

[179] SHAMAIE, A. ; SUTHERLAND, A.: Bayesian fusion of hidden Markov models for understanding bimanual movements. In: *Proc. Int. Conf. on Automatic Face and Gesture Recognition*. Seoul, Korea, 2004 (Zitiert auf Seiten 37, 45 und 154.)

[180] SHEN, Y. ; FOROOSH, H.: View-invariant action recognition using fundamental ratios. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 36, 44, 47, 50 und 51.)

[181] SHIN, M. C. ; CHANG, K. I. ; TSAP, L. V.: Does colorspace transformation make any difference on skin detection? In: *Proc. 6th IEEE Workshop on Applications of Computer Vision*. Orlando, FL, USA, 2002, S. 275–279 (Zitiert auf Seiten 190 und 191.)

[182] SIGAL, L. ; BLACK, M. J.: HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion / Brown University. Providence, RI, 2006 (CS-06-08). – Forschungsbericht (Zitiert auf Seiten 37, 165 und 173.)

[183] SIGAL, L. ; BLACK, M. J.: Predicting 3D people from 2D pictures. In: *Proc. Int. Conf. on Articulated Motion and Deformable Objects*. Port d'Andratx, Spanien, 2006 (Zitiert auf Seiten 35, 40, 41, 46 und 49.)

[184] SONKA, M. ; HLAVAC, V. ; BOYLE, R. ; GOWANS, H. (Hrsg.): *Image processing, analysis, and machine vision*. Third Edition. Thomson Learning, 2008 (Zitiert auf Seiten 41 und 45.)

[185] Sony Corporation: *EVI-D70/D70P color video camera technical manual*, 2003 (Zitiert auf Seite 93.)

[186] Soriano, M. ; Martinkauppi, B. ; Huovinen, S. ; Laaksonen, M.: Skin detection in video under changing illumination conditions. In: *Proc. Int. Conf. on Pattern Recognition*. Barcelona, Spanien, 2000 (Zitiert auf Seite 121.)

[187] Speckmann, E.-J. ; Wittkowski, W.: *Bau und Funktion des menschlichen Körpers*. Urban & Fischer, 1998 (Zitiert auf Seite 80.)

[188] Starner, T. ; Leibe, B. ; Minnen, D. ; Westyn, T. ; Hurst, A. ; Weeks, J.: The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3D reconstruction for augmented desks. In: *Machine Vision and Applications* 14 (2003), S. 59–71 (Zitiert auf Seite 57.)

[189] Stauffer, C. ; Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Los Angeles, CA, USA, 1998, S. 2246–2252 (Zitiert auf Seiten 17 und 18.)

[190] Stenger, B. ; Thayananthan, A. ; Torr, P. H. S. ; Cipolla, R.: Hand pose estimation using hierarchical detection. In: *Computer Vision in Human-Computer Interaction, Lecture Notes in Computer Science* Bd. 3058. Springer, 2004, S. 105–116 (Zitiert auf Seiten 37, 39, 41, 42, 47 und 50.)

[191] Störring, M. ; Andersen, H. ; Granum, E.: Skin color detection under changing lighting conditions. In: *Proc. 7th Symposium on Intelligent Robotics Systems*, 1999, S. 187–195 (Zitiert auf Seite 121.)

[192] Tamimi, H. ; Andreasson, H. ; Treptow, A. ; Duckett, T. ; Zell, A.: Localization of mobile robots with omnidirectional vision using particle filter and iterative SIFT. In: *Robotics and Autonomous Systems* 54 (2006), Nr. 9, S. 758–765 (Zitiert auf Seite 64.)

[193] Tavakkoli, A. ; Nicolescu, M. ; Bebis, G. ; Nicolescu, M.: Efficient background modeling through incremental support vector data description. In: *Proc. Int. Conf. on Pattern Recognition*. Tampa, FL, USA, 2008 (Zitiert auf Seite 18.)

[194] Thurau, C. ; Hlaváč, V.: Pose primitive based human action recognition in videos or still images. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 36, 40, 43, 44, 47, 52 und 71.)

[195] TRAN, D. ; SOROKIN, A. ; FORSYTH, D.: Human activity recognition with metric learning. In: *Proc. European Conf. on Computer Vision, Lecture Notes in Computer Science* Bd. 5302. Springer, 2008, S. 548–561 (Zitiert auf Seiten 36, 39, 42, 43, 44, 45, 47 und 50.)

[196] TRIESCH, J. ; VON DER MALSBURG, C.: A system for person-independent hand posture recognition against complex backgrounds. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 12, S. 1449–1453 (Zitiert auf Seiten 37, 40, 42, 47, 51 und 116.)

[197] TURAGA, P. ; CHELLAPPA, R. ; SUBRAHMANIAN, V. S. ; UDREA, O.: Machine recognition of human activities: A survey. In: *IEEE Transactions on Circuits and Systems for Video Technology* 18 (2008), Nr. 11, S. 1473–1488 (Zitiert auf Seiten 34 und 36.)

[198] TURK, M. ; PENTLAND, A.: Eigenfaces for recognition. In: *Journal of Cognitive Neuroscience* 3 (1991), Nr. 1, S. 71–86 (Zitiert auf Seite 14.)

[199] URTASUN, R. ; DARRELL, T.: Sparse probabilistic regression for activity-independent human pose inference. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008 (Zitiert auf Seiten 35, 39, 41, 48 und 49.)

[200] VAN RIJSBERGEN, C. J.: *Information retrieval*. Butterworths, 1975 (Zitiert auf Seite 213.)

[201] VIOLA, P. ; JONES, M.: Robust real-time object detection. In: *Proc. 2nd Int. Workshop on Statistical and Computational Theories of Vision-Modeling, Learning, Computing, and Sampling*. Vancouver, Kanada, 2001 (Zitiert auf Seiten 50, 66, 70 und 107.)

[202] VON HARDENBERG, C. ; BÉRARD, F.: Bare-hand human-computer interaction. In: *Proc. Workshop on Perceptive User Interfaces* Bd. 15. Orlando, FL, USA, 2001 (ACM Int. Conf. Proc. Series), S. 1–8 (Zitiert auf Seiten 37, 42 und 49.)

[203] WANG, Q. ; CHEN, X. ; ZHANG, L.-G. ; WANG, C. ; GAO, W.: Viewpoint invariant sign language recognition. In: *Computer Vision and Image Understanding* 108 (2007), S. 87–97 (Zitiert auf Seiten 37, 45 und 51.)

[204] WANG, S. B. ; QUATTONI, A. ; MORENCY, L.-P. ; DEMIRDJIAN, D.: Hidden conditional random fields for gesture recognition. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. New York, NY, USA, 2006 (Zitiert auf Seiten 38, 39, 45, 49, 50 und 55.)

[205] Weinland, D. ; Boyer, E. ; Ronfard, R.: Action recognition from arbitrary views using 3D exemplars. In: *Proc. Int. Conf. on Computer Vision*. Rio de Janeiro, Brasilien, 2007, S. 1–7 (Zitiert auf Seiten 36, 39, 41, 44, 47 und 50.)

[206] Weinland, D. ; Ronfard, R. ; Boyer, E.: Free viewpoint action recognition using motion history volumes. In: *Computer Vision and Image Understanding* 104(2) (2006), S. 249–257 (Zitiert auf Seiten 36, 37, 39, 42, 44, 47 und 50.)

[207] Wienecke, M.: *Videobasierte Handschrifterkennung*, Universität Bielefeld, Diss., 2003 (Zitiert auf Seiten 147 und 148.)

[208] Wilking, T.: *Entwicklung eines ubiquitären Interfaces für intelligente Umgebungen*, TU Dortmund, Fakultät für Informatik, Diplomarbeit, 2010 (Zitiert auf Seite 159.)

[209] Wilson, A. D. ; Benko, H.: Combining multiple depth cameras and projectors for interactions on, above, and between surfaces. In: *Proc. ACM Symposium on User Interface Software and Technology*. New York, NY, USA, 2010 (Zitiert auf Seiten 3, 53, 57 und 220.)

[210] Wobbrock, J. O. ; Wilson, A. D. ; Li, Y.: Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In: *Proc. ACM Symposium on User Interface Software and Technology*. Newport, RI, USA, 2007 (Zitiert auf Seite 57.)

[211] Wu, C. ; Aghajan, H.: Model-based human posture estimation for gesture analysis in an ppportunistic fusion smart camera network. In: *Proc. Int. Conf. Advanced Video and Signal based Surveillance*. London, UK, 2007, S. 453–458 (Zitiert auf Seiten 36, 39, 42, 48 und 51.)

[212] Yang, C. ; Duraiswami, R. ; Davis, L.: Efficient mean-shift tracking via a new similarity measure. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. San Diego, CA, USA, 2005 (Zitiert auf Seite 108.)

[213] Yilmaz, A. ; Javed, O. ; Shah, M.: Object tracking: A survey. In: *ACM Computing Surveys* 38 (2006), Nr. 4. – Article 13 (Zitiert auf Seiten 15 und 108.)

[214] Zhang, Z.: A flexible new technique for camera calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 11, S. 1330–1334 (Zitiert auf Seite 126.)

[215] ZHAO, Z. ; ELGAMMAL, A.: Human activity recognition from frame's spatiotemporal representation. In: *Proc. Int. Conf. on Pattern Recognition*. Tampa, FL, USA, 2008 (Zitiert auf Seiten 36, 40, 43, 44, 47 und 52.)

[216] ZHU, Q. ; AVIDAN, S. ; CHENG, K.-T.: Learning a sparse, corner-based representation for time-varying background modelling. In: *Proc. Int. Conf. on Computer Vision*. Beijing, China, 2005, S. 678–685 (Zitiert auf Seite 18.)

[217] ZHU, Q. ; SHAI, A. ; MEI, C. Y. ; KWANG, T. C.: Fast human detection using a cascade of histograms of oriented gradients. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. New York, NY, USA, 2006, S. 1491–1498 (Zitiert auf Seiten 70, 71 und 106.)