Erkennung menschlicher Aktivitäten zur Belehrung von Robotern

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Martin Lösch

aus Landau/Pfalz

Tag der mündlichen Prüfung:	22. Oktober 2012
Erster Gutachter:	Prof. Dr.-Ing. Rüdiger Dillmann
Zweiter Gutachter:	Prof. Dr.-Ing. Gerhard Sagerer

Inhaltsverzeichnis

1. Einführung

1.1. Motivation

Der Gedanke, einen stillen und stets zu Dienste stehenden Helfer zu haben, bewegt die Menschen schon seit langer Zeit, sei es als Flaschengeister in den Märchen von „Tausendundeiner Nacht“, der Legende vom Prager Golem, der Sage von den Heinzelmännchen von Köln oder als eine der vielen anderen Ideen, die in Sagen und Märchen Eingang gefunden haben. In neuerer Zeit wurden diese sehr vagen, mit Aberglauben und Magie versetzten Ideen immer stärker durch die Idee eines mechanisch-technischen Dieners ersetzt, seien es die mechanischen Figuren eines Leonardo da Vinci oder die Roboter in den Geschichten eines Isaac Asimov [Asimov, 2004]. Durch die filmische Darstellung mit immer besseren Spezialeffekten wurden diese Vorstellungen in jüngerer Zeit immer konkreter und erweckten Erwartungen, von denen die Realität der Roboterforschung leider noch immer weit entfernt ist. Menschenähnliches Verhalten, Wahrnehmung und Handeln von Robotern wie dargestellt bei *C3PO* in *Star Wars*, *Data* in *Star Trek* oder den Robotern in *I, Robot* ist noch immer eine unerreichte Vision.

Trotzdem gibt es natürlich auch große Fortschritte bei real existierenden Systemen. Sie manifestieren sich unter anderem auch dadurch, dass das Interesse an Servicerobotern und allgemein robotischen Systemen für den Haushalt in den letzten Jahren massiv zugenommen hat. Zwei herausragende Meilensteine diesbezüglich sind die Verfügbarkeit von autonom arbeitenden Staubsaugerrobotern wie dem *Roomba*® der Firma *iRobot*®[1], und die Entwicklung des Serviceroboters *PR 2* der Firma *Willow Garage*[2], die in Abb. 1.1 gezeigt sind. Alleine iRobot® hat laut eigener Aussage bis 2010 über 6 Millionen Heimroboter verkauft [Angle , CEO bei iRobot®]. Folgerichtig konzentrieren sich viele aktuelle Forschungsanstrengungen auf die Bereitstellung von Technologien, die autonome Serviceroboter für den Alltag in normalen Haushalten tauglich machen sollen.

Unterstützt werden Forschungsarbeiten in diesem Bereich durch Fortschritte in anderen Technologiebereichen, insbesondere auch in der Sensortechnologie. Gestützt auch von Entwicklungen für die neueste Generation von Spielekonsolen sind mittlerweile günstige und hochauflösende Tiefensensoren verfügbar, deren Leistungsparameter sie für die Verwendung zur Er-

[1] `http://www.irobot.com/de/`

[2] `http://www.willowgarage.com/`

(a)

(b)

Abb. 1.1.: Beispiel für verbreitete oder weithin bekannt gemachte Roboter: (a) Roomba® der Firma iRobot®, Bildquelle: [guzugi , Wikipedia]. (b) PR2 der Firma Willow Garage, Bildquelle: [Vollmer].

kennung menschlicher Bewegungen prädestinieren. Unter diesen Voraussetzungen gibt es zwei Anwendungen für die Erkennung von menschlichen Aktivitäten, die die vorliegende Arbeit motivieren: Die Interaktion zwischen Mensch und Roboter einerseits, und die natürliche Programmierung von Robotern nach dem *Programmieren durch Vormachen (PdV)*-Ansatz andererseits.

Zum Einen ist in natürlichen Umgebungen die Interaktion mit dem Menschen eine wichtige Fähigkeit für autonome Roboter. Neben dem Verstehen von sprachlichen Äußerungen ist dabei ein Verständnis für die Handlungen von Menschen ein zentraler Kommunikationskanal für Haushaltsroboter, dessen Nutzung wichtig für die Akzeptanz solcher Systeme sein wird. Für eine autonome Entscheidungsfindung – beispielsweise ob Hilfe angeboten werden soll, oder ob Störungen unterlassen werden sollten – stellt die Interpretation von beobachteten menschlichen Bewegungen einen natürlichen, oft sogar den einzigen verfügbaren Informationskanal dar, da in den beschriebenen Situationen nicht mit auf den Roboter gerichteten Sprachäußerungen gerechnet werden kann.

Zum Anderen stellt die aktuell aufkommende Verwendung von Robotern in natürlichen Umgebungen und im Umgang mit Menschen, wobei die Roboter auch autonom gewisse Entscheidungen treffen sollen, neue Anforderungen an die Programmierung. Klassische Programmiertechniken stoßen hier schnell an ihre Grenzen, weswegen der Ansatz des *Programmierens durch Vormachen (PdV)* entwickelt wurde. Bei diesem Paradigma wird eine Handlung, die ein Roboter durchführen soll, von einem Menschen in natürlicher Art und Weise demonstriert, so wie er sie auch einem anderen Menschen zeigen würde. Um aus einer beobachteten Demonstration Handlungswissen für den Roboter zu synthetisieren, ist insbesondere auch eine Analyse und Interpretation der vom Menschen durchgeführten Handlungen notwendig. Diese Aktivitätserkennung weist umso mehr Analogien mit der Erkennung in der Interaktion auf, wenn das PdV Handlungswissen über komplexe Missionen generieren soll

Zusammengefasst stellt die Erkennung und Interpretation von menschlichen Handlungen einen wichtigen Schritt hin zu autonomen Robotern da. Die von einem solchen System bereitgestellten Ergebnisse können sowohl für die Programmierung von Robotern, als auch zur Unterstützung bei der Kommunikation und in der autonomen Entscheidungsfindung genutzt werden.

1.2. Zielsetzung und Beitrag der Arbeit

Es existieren zwei grundlegend verschiedene Ansätze zur Erkennung von menschlichen Bewegungen und Aktivitäten. Folgend dem einen Ansatz wird zunächst eine Modellbildungsphase durchgeführt, in der ein Modell des beobachteten Menschen aufgebaut wird. Die Erkennung von Aktivitäten erfolgt anschließend auf der Sequenz von Modelldaten. Der andere Ansatz arbeitet direkt auf den Sensorwahrnehmungen, indem die Beobachtungen und Änderungen der Sensordaten direkt interpretiert werden.

Die vorliegende Arbeit verfolgt den ersten Ansatz und leistet Beiträge in diesem Bereich, um die Verwendung eines Aktivitätserkennungssystems in PdV-Systemen und als Informationskanal für Systeme zur autonomen Entscheidungsfindung von Robotern möglich zu machen. Die in den folgenden Kapiteln präsentierte Arbeit zeigt Lösungen für die folgenden Forschungsfragen, die die Leitgedanken der Hauptbeiträge dieser Arbeit darstellen. Der dabei zentral verwendete Begriff des *Erkenners* wird in den folgenden Kapiteln näher erläutert, hier genüge die Idee eines Modell für die Erkennung einer einzelnen Aktivität, beispielsweise könnte man sich ein neuronales Netz vorstellen.

Forschungsfrage 1 Kann ein Aktivitätserkennungssystem so realisiert werden, dass es gleichzeitig

- mit verschiedenen Sensoren eingesetzt werden kann,
- trainierte Erkenner für einzelne Aktivitäten zwischen verschiedenen Plattformen übertragen werden können, und
- Erkenner für verschiedene Aktivitäten in beliebiger Kombination eingesetzt werden können?

Zur Lösung wurde ein am Lehrstuhl entwickeltes Trackingsystem erweitert um die Nutzung von Hintergrundwissen zur Modellnachführung. Darauf aufbauend wurde ein dreistufiges System zur Aktivitätserkennung konzeptioniert und implementiert. Die Hauptidee dabei ist die Verwendung von aussagekräftigen, möglichst sensor-unabhängigen Merkmalen, die auf verschiedenen Plattformen bestimmt werden können. Darüberhinaus werden für jede Aktivität eigene

Erkenner trainiert, die systematisch kombiniert werden können unter zusätzlichem Einsatz von Hintergrundwissen.

Forschungsfrage 2a Kann ein Aktivitätserkennungssystem so gestaltet werden, dass ein einfaches und schnelles Trainieren neuer Aktivitäten möglich ist?

Forschungsfrage 2b Kann dieser Trainingsprozess sogar so weit vereinfacht werden, dass die Erkennung auch von technisch-versierten, aber nicht speziell geschulten Benutzern durch das Einlernen neuer Aktivitäten erweitert werden kann?

Durch eine weitgehende Automatisierung der Prozesskette wurde eine Lösung dieser Fragen erreicht. Beginnend mit einer automatischen Suche nach geeigneten Merkmalen in unbekannten Domänen, über die von Hintergrundwissen gestützte Auswahl relevanter Merkmale für spezifische Aktivitäten, bis hin zum Training neuer Erkenner ergibt sich ein auch für Nicht-Experten erweiterbares Aktivitätserkennungssystem.

1.3. Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich neben dem aktuellen Kapitel in 8 Abschnitte, deren Inhalt sich wie folgt zusammensetzt:

Kapitel 2 diskutiert den Stand der Forschung in den von dieser Arbeit berührten Bereichen der Forschung, namentlich die Beobachtung von menschlichen Bewegungen und die Erkennung und Interpretation von Bewegungen und komplexeren Aktivitäten.

Kapitel 3 beschreibt die wichtigsten Grundlagen, auf denen die Forschungsbeiträge dieser Arbeit aufbauen. Die Inhalte sind das in dieser Arbeit genutzte und weiterentwickelte Trackingsystem *VooDoo*, Verfahren zur Auswahl relevanter Merkmale und im Rahmen der Arbeit eingesetzte Klassifikatoren, *Hidden Markov Modelle (HMMs)* und *Support Vector Maschinen (SVMs)*.

Kapitel 4 formalisiert die behandelte Problemstellung mit den für eine Lösung nötigen Eigenschaften und Randbedingungen. Anschließend wird ein diesen Randbedingungen genügendes Lösungskonzept präsentiert, das auf abstrakter Ebene das Zusammenspiel der einzelnen Komponenten und die Einbindung von Hintergrundwissen in den Prozess beschreibt. Die Details der Realisierung dieses Konzeptes werden in den folgenden Kapiteln dargestellt.

Kapitel 5 beschreibt die Forschungsarbeiten im Bereich der Menschbeobachtung (und damit der Datenakquisition) als Grundlage für die Erkennung von Aktivitäten. Die präsentierten Beiträge sind hier ein Verfahren zur Initialisierung eines Menschmodells für das Tracking und die Realisierung von Gelenkwinkelbegrenzungen, ein zentrales Element um korrekte Körperstellungen zu erzwingen, die als Grundlage für die darauf aufbauende Erkennung dienen können.

Kapitel 6 beschreibt die Details der eigentlichen Aktivitätserkennung, mit Beiträgen zur Repräsentation und automatischen Generierung von Merkmalen, der Auswahl relevanter Merkmale für einzelne Aktivitäten, und der Struktur der Erkennungskomponenten, um Aktivitäten unterschiedlicher Komplexität robust erkennen zu können.

Kapitel 7 präsentiert die Evaluation der Neuerungen des Trackingsystems und der drei Teilprozessketten der Aktivitätserkennung: Die Erschließung neuer Domänen durch die automatische Exploration von Merkmalen, das Trainieren neuer Aktivitäten, und die Erkennung von verschiedenen Aktivitäten.

Kapitel 8 fasst die Ausführungen noch einmal zusammen und diskutiert mögliche Erweiterungen und Weiterentwicklungen.

1.4. Notation

An dieser Stelle seien einige Anmerkungen zur verwendeten Notation in mathematischen Ausdrücken erlaubt:

- Vektoren werden standardmäßig in Fettschrift dargestellt, z.B. $\mathbf{v}$, $\mathbf{n_0}$.
- In einigen Fällen wird für Vektoren die Schreibweise als Verbindung zweier Punkte bevorzugt, und durch einen Pfeil über den beiden Punkten deutlich gemacht, z.B. $\overrightarrow{P_1P_2}$, $\overrightarrow{AB}$.
- Variablen für selbst-definierte Werte werden mit Großbuchstaben der folgenden Form bezeichnet: $\mathcal{A}, \mathcal{B}, \ldots$.
  Gegebenenfalls wird eine Variable zur genaueren Bezeichnung um tiefgestellte Indizes ergänzt, beispielsweise $\mathcal{K}_T$.
- Variablen für Messwerte, Zwischenergebnisse etc. werden standardmäßig durch Kleinbuchstaben in Kursivschrift, eventuell ergänzt um tiefgestellte Indizes, repräsentiert: a, b, h, b_{BB}.
  Wenn die Verwendung von Großbuchstaben in der gleichen Kursivschrift für mehr Klarheit sorgt (beispielsweise U für einen Kreisumfang), wird letzteren der Vorzug gegeben, auch hier sind eventuell zusätzliche Indizes zur genaueren Spezifizierung mit angegeben: A, U, U_K.
- Faktoren, Schwellwerte etc. werden mit kleinen griechischen Buchstaben bezeichnet, die gegebenenfalls durch Indizes genauer spezifiziert werden können, z.B. α, η, θ_{min}.

2. Stand der Forschung

Die vorliegende Arbeit liefert Beiträge in der Beobachtung von menschlichen Bewegungen und der Erkennung der in diesen Bewegungen ausgedrückten Aktivitäten. In diesem Kapitel wird der Stand der Technik in diesen beiden Forschungsgebieten zusammenfassend dargestellt. Zunächst gibt Abschnitt 2.1 einen Überblick über unterschiedliche Ansätze und Lösungen für die Beobachtung von menschlichen Bewegungen. Anschließend präsentiert der folgende Abschnitt 2.2 wichtige Ansätze zur Erkennung von Aktivitäten.

2.1. Beobachtung von Menschen

Die Beobachtung menschlicher Bewegungen durch das Nachführen (engl. *tracking*) eines Modells ist ein sehr aktives Forschungsgebiet, dessen zahlreiche Anwendungsmöglichkeiten von der Überwachung öffentlicher Plätze über die Robotik bis hin zur Spieleindustrie reichen. Ausgehend von diesen diversen Anwendungsgebieten wurden Trackingverfahren mit unterschiedlichen Eigenschaften entwickelt, um den jeweiligen Anforderungen gerecht zu werden.

Die Artikel von Moeslund et. al. [Moeslund and Granum, 2001], Wang et al. [Wang et al., 2003] und Ji et al. [Ji and Liu, 2010] geben einen Überblicke über Ansätze zur kamerabasierte Bewegungsverfolgung. Die folgende Darstellung des Stands der Personenbeobachtung ist geordnet nach der verwendeten Sensorik (Sensortypen, Anzahl und Anordnung), da so auf natürliche Art verschiedene Anwendungen für derartige Systeme unterschieden werden. Die in der vorliegenden Arbeit angestrebte Verwendung des Systems auf robotischen Systemen stellt damit nur ein mögliches Anwendungsszenario dar. Ein anderes, häufig genutztes Ordnungskriterium für Systeme zur Personenbeobachtung stellen die eingesetzten Familien von Trackingverfahren dar. Für die hier angestrebte Darstellung ist dieser Ansatz aber nicht so gut geeignet, da die eingesetzten Verfahren nur bedingt mit den möglichen Anwendungsszenarien korrelieren. Tracking ist ein typisches Einsatzgebiet für Kameras, aber auch Laser-basierte Tiefensensoren oder Beschleunigungssensoren werden mitunter verwendet. In der Robotik werden meist nur auf einem Roboter angebrachte Sensoren (*Onboard-Sensorik*) eingesetzt, während in anderen Anwendungsgebieten der Einsatz verteilter Sensoren (z.B. im Rahmen eines *Smart Rooms*) möglich ist. Im Bereich von detaillierten Bewegungsuntersuchungen z.B. im Rahmen der Sportwissenschaften wird auch auf den Einsatz von Sensoren zurückgegriffen, die direkt

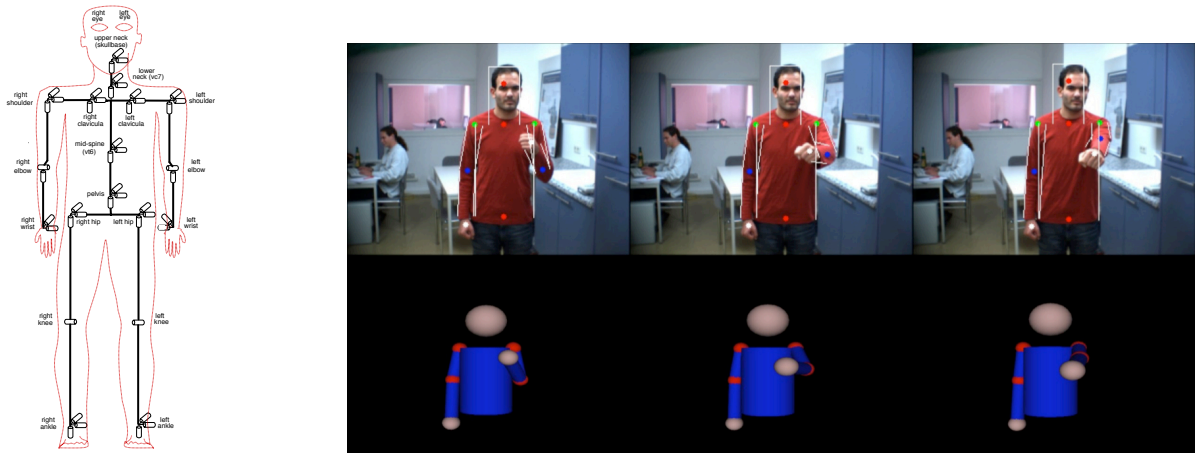
auf dem menschlichen Körper befestigt werden. Die beiden folgenden Abschnitte diskutieren wichtige Ansätze basierend auf der Onboard-Sensorik von Robotern in Abschnitt 2.1.1, andere kamerabasierte Ansätze in Abschnitt 2.1.2 und mittels am menschlichen Körper applizierter Sensoren in Abschnitt 2.1.3.

2.1.1. Ansätze basierend auf Roboter-geeigneter Sensorik

Auf einem autonomen Roboter werden ganz spezielle Anforderungen an ein Trackingsystem gestellt, da sowohl die Sensorik als auch die Rechenkapazität starken Einschränkungen unterliegen, andererseits aber ein Tracking in Echtzeit benötigt wird, damit das System einsetzbar ist. In einem Roboter kann nicht eine beliebige Anzahl Sensoren verbaut werden, dadurch ist der Sichtbereich beschränkt und ähnelt durch Sensorposition und Sensortyp dem des Menschen. Aufgrund des verfügbaren Raums und der aus einer Batterie stammenden Energie ist auch die Rechenkapazität in den meisten Fällen nicht mit einem aktuellen Standard-PC vergleichbar.

Tracking auf Videokamera-Bildern

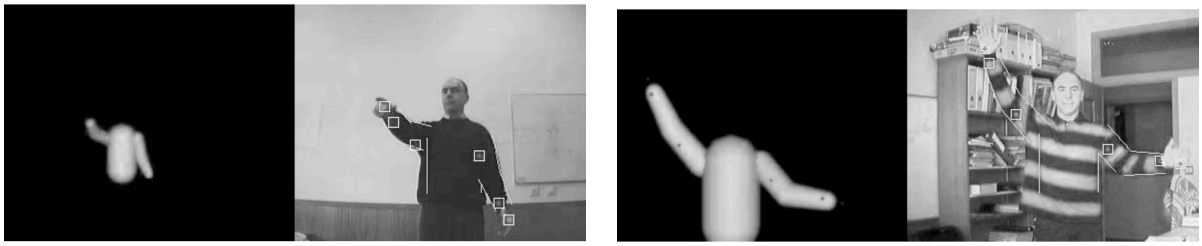
Die übliche Wahl bezüglich des eingesetzten Sensors waren lange Zeit klassische Farb-Videokameras, entweder einzeln (Mono-Kamera-Aufbau) oder als Paar (dem menschlichen Augenpaar nachgebildet) in einem Stereo-Kamera-Aufbau.



(a) (b)

Abb. 2.1.: Details zum Trackingsystem von Azad: (a) Das verwendete Mensch-Modell, die sogenannte *Master Motor Map*. (b) Beispiel für die Ergebnisse des markerlosen Trackingsystems. Bildquelle: [Azad et al., 2007].

Das von Azad in [Azad et al., 2004, 2007; Azad, 2009] im Rahmen des EU-Projekts *PACO-PLUS*[1] und des DFG-Sonderforschungsbereichs *SFB-588 Humanoide Roboter*[2] entwickelte Trackingsystem wird auf dem Roboter ARMAR am Karlsruher Institut für Technologie eingesetzt. Als Datenquelle werden Stereo-Farbbilder eines aus zwei Kameras aufgebauten, aktiven Sensorkopfes genutzt. Das entwickelte Verfahren basiert auf einemPartikelfilter. Ein Partikelfilter schätzt eine Verteilung durch eine gewählte Anzahl von Partikeln, von denen jedes eine Hypothese darstellt. Im vorliegenden Fall eines Körpertrackings stellt jedes Partikel eine mögliche Modellkonfiguration dar. In jedem Schritt werden neue Partikel geschätzt und gewichtet abhängig von ihrer Qualität bezüglich der aktuellen Messungen. Das System wird hauptsächlich genutzt für das Verfolgen des Oberkörpers, aber der Ansatz wird auch für das Tracking des vollständigen Körpers inklusive Beinen eingesetzt. Das verwendete Modell des Menschen ist in Abb. 2.1(a) dargestellt. Das Trackingproblem wird hierarchisch aufgeteilt in mehrere Teilprobleme, die mittels einzelner Partikelfilter gelöst werden. Zur Bewertung der Partikel werden in Trackingsystemen verschiedene Hinweise (engl. *cue*) genutzt, im System von Azad dienen hierzu Vergleiche der Kanten im Bild mit den von den Partikeln vorhergesagten (engl. *edge cue*) und der Vergleich der Tiefe im Stereobild mit der aus dem Partikel bestimmten Tiefe (engl. *distance cue*). Die Initialisierung erfolgt in einem gesonderten Schritt oder manuell, die möglichen Konfigurationen (repräsentiert durch die Partikel) sind von vornherein nur auf anatomisch mögliche Konfigurationen beschränkt. Bei der Verfolgung nur des Oberkörpers läuft das System mit 15 Hz.



(a) (b)

Abb. 2.2.: Ergebnisse des Trackingsystem von Menezes et. al. in verschiedenen Umgebungen: (a) Einfach strukturierter Hintergrund mit hohem Kontrast zur verfolgten Person. (b) Unordentlicher Hintergrund. Bildquelle: [Menezes et al., 2006].

Ein ebenfalls auf einem Partikelfilter-Ansatz basierendes System wurde am LAAS in Toulouse von Menezes et. al. entwickelt [Menezes et al., 2005, 2006] für die Verwendung mit einer einzelnen, auf einem Roboter angebrachten Farbkamera. Als Hinweise für die Gewichtung der

[1] Homepage des Projekts: `http://www.paco-plus.org/`

[2] Homepage des Projekt: `http://www.sfb588.uni-karlsruhe.de/`

Partikel werden hier Kanten, Bewegungen, lokale Farbverteilungen aus den Kamerabildern sowie die Betrachtung der Stabilität von Konfigurationen und Selbstkollisionen genutzt. Modelliert werden beide Arme, aber weder Kopf noch Unterkörper, und das Tracking erreicht dabei eine Geschwindigkeit von 1 Hz. Die Initialisierung wird manuell vorgenommen, Einschränkungen der Beweglichkeit sind in das Bewertungsmaß für die Partikel integriert. Beispielhafte Ergebnisbilder bei unterschiedlichen Hintergrund werden in Abb. 2.2 gezeigt.

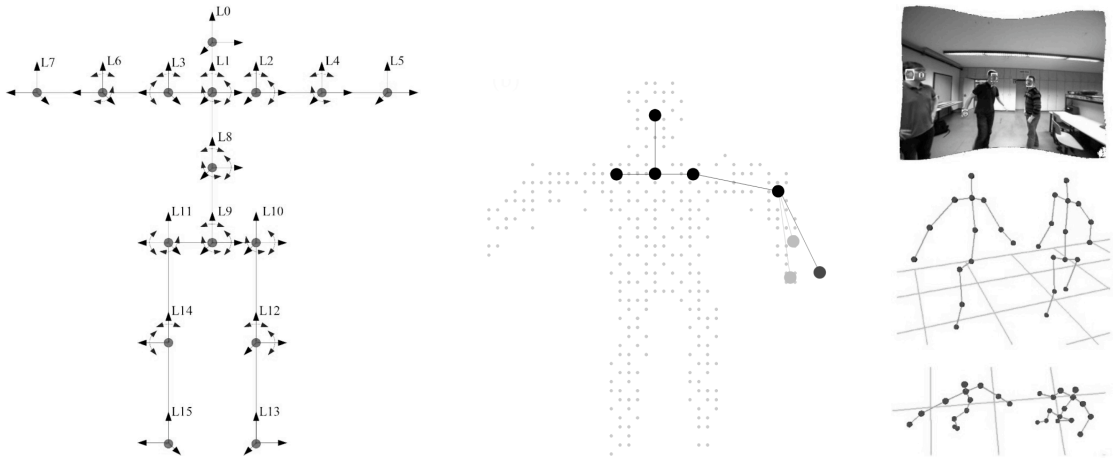
Eine ähnliche Zielsetzung, die Verfolgung von Gesten, liegt auch den Arbeiten von Nickel et. al. [Nickel et al., 2004; Nickel, 2008] und Mühlbauer et. al [Mühlbauer et al., 2008] zugrunde. Die Arbeit von Nickel im Rahmen des *SFB-588 Humanoide Roboter* nutzt ein Stereo-Kamera-Paar auf dem Roboter ARMAR als Sensoren. Aufbauend auf einer Hautfarbenerkennung wird ein Multihypothesen-Tracking für die Position der Hände und des Kopfes genutzt. Zusätzlich wird ein auf neuronalen Netzen beruhendes Verfahren zur Schätzung der Kopfneigung eingesetzt. Die Arbeit von Mühlbauer in München dient der Mensch-Roboter-Interaktion des *Autonomous City Explorer* ACE[3]. ACE ist ein autonomer Roboter, der in unstrukturieren Stadtumgebungen navigieren soll, indem Passanten um Hilfe und Weghinweise gebeten werden. Das System in der veröffentlichten Form ist kein Verfolgen der Bewegungen im eigentlichen Sinn, sondern bestimmt fortwährend (analog einer Initialisierung) die Pose von beobachteten Menschen (Abb. 2.3(a) zeigt das dabei verwendete Menschmodell) in aus Stereokamerabildern gewonnen Punktwolken, wobei das System auch die Detektion von mehreren Personen ermöglicht. Dazu werden die Punkte anhand von Farbmerkmalen geclustert, in die 27 vorher bestimmte typische Posen eingepasst und bewertet werden. Abb. 2.4 zeigt eine Auswahl der verwendeten typischen Posen, Abb. 2.3(b) zeigt die Adaption eines einzelnen Körperteils. Das System ermöglicht eine Geschwindigkeit von 2 Hz, das Ergebnis eines Erkennungsschritts zeigt Abb. 2.3(c).

Tracking auf Tiefenkamera-Punktwolken

In Erweiterung der Ansätze, die auf dem Einsatz von Farbkameras basieren, beschäftigen sich aktuelle Forschungen vermehrt mit dem Einsatz von Tiefenkameras anstelle von Mono- oder Stereo-Kameraköpfen, seit solche Sensoren zur Erzeugung dichter Punktwolken kommerziell verfügbar sind. Abb. 2.5 zeigt zwei typische Vertreter solcher aktiver Sensoren. Der in Abb. 2.5(a) gezeigt Sensor ist ein *Swissranger^{TM}SR4000* der Firma Mesa Imaging AG[4], ein *Time-of-Flight (ToF)*-Sensor, der moduliertes Infrarotlicht aussendet und die Phasenverschie-

[3]Homepage des Projekts: `http://www.lsr.ei.tum.de/research/research-areas/robotics/ace-the-autonomous-city-explorer-project/`

[4]Homepage der Firma: `http://www.mesa-imaging.ch`



(a) (b) (c)

Abb. 2.3.: Details zum Trackingsystem von Mühlbauer: (a) Verwendetes Mensch-Modell. (b) Einzelschritt der Einpassung einer Modellkonfiguration in die Punktwolken-Cluster. (c) Resultat einer Modelleinpassung mit zwei Personen im Bild. Bildquelle: [Mühlbauer et al., 2008].

Abb. 2.4.: Details zum Trackingsystem von Mühlbauer: Auswahl der verwendeten typischen Posen. Bildquelle: [Mühlbauer et al., 2008].

bung der reflektierten Signale misst, um daraus die Entfernung des reflektierenden Punktes vom Kamerachip zu berechnen. Ein anderes Messprinzip kommt bei dem in Abb. 2.5(b) abgebildeten Kinect-Sensor der Firma Microsoft[5] zum Einsatz. Mittels eines integrierten Infrarot-Projektors wird ein für das menschliche Auge unsichtbares Gittermuster projiziert, mit dessen Hilfe eine 3D-Punktwolke berechnet wird. Die aus diesen Sensoren resultierenden Punktwolken weisen meist deutlich andere Eigenschaften auf als Punktwolken, die aus klassischer Stereorekonstruktion mittels im Bild gefundener Punktkorrespondenzen berechnet werden. Insbesondere sind die Punktwolken auch in gleichartig strukturierten Umgebungen dicht und ohne Lücken, beispielsweise beim „Blick“ auf eine weiße Wand.

Für die Verwendung von Punktwolken zum Zwecke der Personenbeobachtung wurden – getrieben von den deutlichen Unterschieden in den Ausgangsdaten im Vergleich zu Bildern – in den letzten Jahren neue Tracking-Ansätze entwickelt, die besonders für die Verwendung auf Robotern geeignet sind.

Am IAIM Dillmann in Karlsruhe entwickelten Steffen Knoop et al. [Knoop et al., 2005, 2006a,b, 2009] ein System, das einen auf dem *Iterative Closest Point*-Algorithmus basierenden Tracking-Ansatz realisiert. Abb. 2.6(a) zeigt das verwendete Menschmodell, bestehend aus 10 hierarchisch verbundenen, verallgemeinerten Zylindern. Die Neuartigkeit des Ansatzes resultiert aus der entwickelten Modellierung für Gelenke und Gelenkwinkel-Beschränkungen, die die aus früheren ICP-basierten Systemen bekannten Probleme bei der Aufrechterhaltung des Zusammenhalts zwischen den einzelnen Körperteilen umgeht. Gelenke werden durch eine geringe Anzahl künstlicher Punktkorrespondenzen modelliert, die in Analogie zu Gummibändern dafür sorgen, dass die durch sie verbundenen Körperteile mittels des ICP-Algorithmus' aufeinander zu bewegt werden. Dadurch wird eine Geschwindigkeit von $20 - 25\,\mathrm{Hz}$ beim Tracking einer Person erreicht.

Als Sensoren kamen hauptsächlich die von der Schweizer Firma Mesa Imaging (früher: CSEM) entwickelten *SwissRanger*^{TM}-Tiefenkameras in den Versionen 2 und 3000 zum Einsatz, aber auch die Fusion mit aus 2D-Kamerabildern extrahierten Merkmalen wurde behandelt, Abb. 2.6(b) zeigt ein beispielhaftes Ergebnis bei der Fusion von 3D-Punktwolken mit durch Hautfarbensegmentierung gefundenen 2D-Merkmalen aus[Knoop et al., 2006b]. Dieses System ist eine der verwendeten Datenquellen für die vorliegende Arbeit, und wird in Kapitel 3.1 im Detail vorgestellt.

Der oben schon erwähnte Kinect-Sensor (Abb. 2.5(b)) wird bei Verwendung mit der zugehörigen Spielekonsole mit einem Trackingsystem ausgeliefert, von dem einige Details veröffentlicht wurden. In [Shotton et al., 2011] werden zur Initialisierung und Unterstützung des

[5] Webseite des Sensors: `http://www.xbox.com/de-DE/kinect`

(a) (b)

Abb. 2.5.: Zwei aktuelle Vertreter für Sensoren zur Gewinnung von Punktwolken: (a) SwissRanger^{TM}SR4000 der Firma Mesa Imaging AG. (b) Kinect-Kamera der Firma Primesense.

(a) (b)

Abb. 2.6.: Details zum Trackingsystem von Knoop: (a) Verwendetes Mensch-Modell. (b) Trackingergebnis bei Fusion von Punktwolken und aus 2D-Daten gewonnenen Merkmalen. Bildquelle: [Knoop, 2007].

Trackings einzelne Körperteile in den vom Sensor gelieferten Tiefenbildern analog einem Objekterkennungsproblem detektiert. Auf Merkmalen der Form $f_\theta(I,\mathbf{x}) = d_I\left(\mathbf{x}+\frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x}+\frac{\mathbf{v}}{d_I(\mathbf{x})}\right)$ mit Offsets $\theta = (\mathbf{u},\mathbf{v})$ (Abb. 2.7 zeigt solche Merkmale) einer Trainingsdatenmenge von rund 500.000 Frames repräsentativer Bewegungen werden Klassifikatoren (vom Typ *randomized decision forests*) trainiert. Die Gesamtperformance ist durch die Geschwindigkeit des Sensors auf 30 Hz beschränkt, die der Detektion alleine wird mit 200 Hz angegeben.

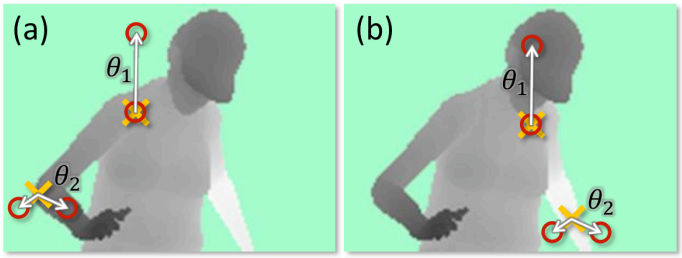


Abb. 2.7.: Zwei Merkmale für die Körperteildetektion von Shotton an zwei unterschiedlichen Positionen angewandt (in (a) und (b)). Die Merkmale sind definiert durch die jeweiligen Offsets θ_1 und θ_2. Bildquelle: [Shotton et al., 2011].

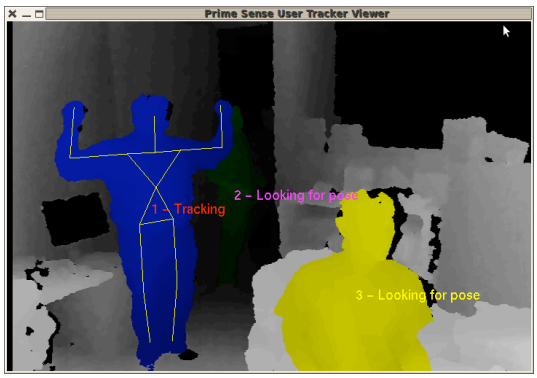


Abb. 2.8.: Beispielhafte Trackingsszene mit NiTE-System: Eine Person (blau) in der zur Initialisierung genutzten Pose, zwei Personen (*A* gelb im Vordergrund, *B* grün im Hintergrund) die (noch) nicht getrackt werden.

Ein frei verfügbares Trackingsystem ausgerichtet auf den Kinect-Sensor wird von der Firma PrimeSense (die ursprünglichen Entwickler der Kinect-Technologie) unter dem Namen *NiTE*[6] angeboten, implementiert gegen die ebenfalls frei verfügbare Schnittstellen-Definition der gemeinnützigen OpenNI-Organisation[7], zu deren Mitgliedern PrimeSense gehört. Details zur Arbeitsweise des Trackings sind bisher nicht veröffentlicht, aber das System zeigt qualitativ ähnliche Ergebnisse wie das offizielle System von Microsoft bezüglich der Robustheit gegenüber

[6] Homepage: `http://www.primesense.com/nite`
[7] Webseite der Organisation: `http://openni.org/`

Ausreißern, und läuft auf Standard-PCs mit bis zu 30 Hz. Darüberhinaus existiert eine erprobte Anbindung der Software an die *Robot Operating System (ROS)*-Middleware[8] unter der Modulbezeichnung `openni_tracker`. In Abb. 2.8 wird eine beispielhafte Szene des NiTE-Trackings gezeigt, mit einer Person in der zur Initialisierung genutzten Pose und zwei weiteren Personen im Bild, die (noch) nicht getrackt werden.

2.1.2. Andere kamerabasierte Trackingverfahren

Die bisher beschriebenen Ansätze wurden für den Einsatz mit Sensordaten aus menschenähnlichen Blickwinkeln entwickelt, beispielsweise auf Robotern. Wenn andererseits mehr Freiheit bei der Anbringung der Kameras und ihrer Anzahl besteht, können deutlich bessere Ergebnisse erzielt werden. Für hochgenaue Beobachtungen wie beispielsweise in den Sportwissenschaften kommen häufig verteile Kameranetze zum Einsatz, die den Vorteil haben, dass Verdeckungen leichter umgangen werden können. Darüberhinaus sind in vielen Bereichen die Laufzeitanforderungen etwas gelockert im Vergleich zu Robotikanwendungen, da hier keine direkte Interaktion zwischen dem beobachteten Menschen und dem Beobachtungssystem stattfindet.

Kakadiaris et al. beschreiben in [Kakadiaris and Metaxas, 1998] ein Verfahren, bei dem drei paarweise orthogonal angebrachte Kameras eingesetzt werden (Abb. 2.9 zeigt die Anordnung der Kameras). Das Tracking basiert auf der Analyse der Silhouette des beobachteten Menschen, das hier durch Hintergrundsubtraktion (engl. *background subtraction*) gewonnen wird. Es wird kein vormodelliertes Menschmodell benutzt, sondern in einer Initialisierungsphase führt die Person eine Reihe von Bewegungen durch, auf deren Grundlage dynamisch ein auf diese Person abgestimmtes Menschmodell bestimmt wird. Das Modell wird mittels des spziell dafür entwickelten *Human Body Part Decomposition*-Algorithmus (HBPDA) berechnet. Bezüglich der Geschwindigkeit des Verfahrens werden keine Angaben gemacht.

Deutscher et. al. entwickeln in [Deutscher et al., 2000; Deutscher and Reid, 2005] ein auf einem Partikelfilter basierendes Trackingsystem, das ebenfalls drei Kameras einsetzt. Allerdings ist die Anordnung der Kameras hier freier als bei dem System von Kakadiaris. Abb. 2.10(b) zeigt die zeitgleiche Ansicht der drei Kameras mit überblendetem Modell. Das verwendete Menschmodell ist in Abb. 2.10(a) dargestellt, es baut auf einer kinematischen Kette mit 17 Segmenten ein Zylinder-Modell auf. Ein speziell angepasster *Annealed Particle Filter* wird verwendet für das Tracking, zur Gewichtung der Partikel werden als Merkmale Kanten und die Silhouette der Person aus den Kameraansichten extrahiert. Das System benötigt etwa 15 s zur Verarbeitung eines Frames (was einer Geschwindigkeit von etwa 0,067 Hz entspricht). Einer der Gründe für die höhere Rechenzeit im Vergleich zu den vorher vorgestellen Partikelfilter-

[8] Homepage: `http://www.ros.org`

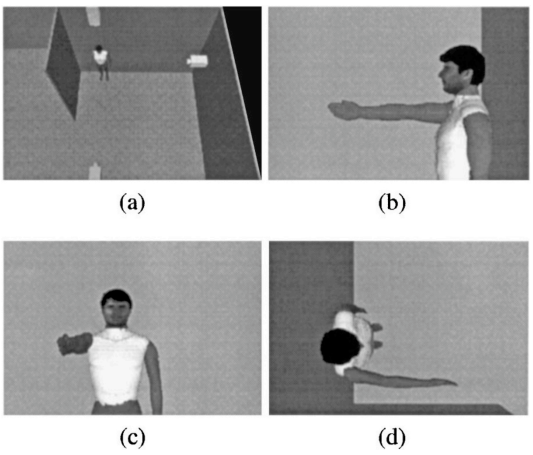
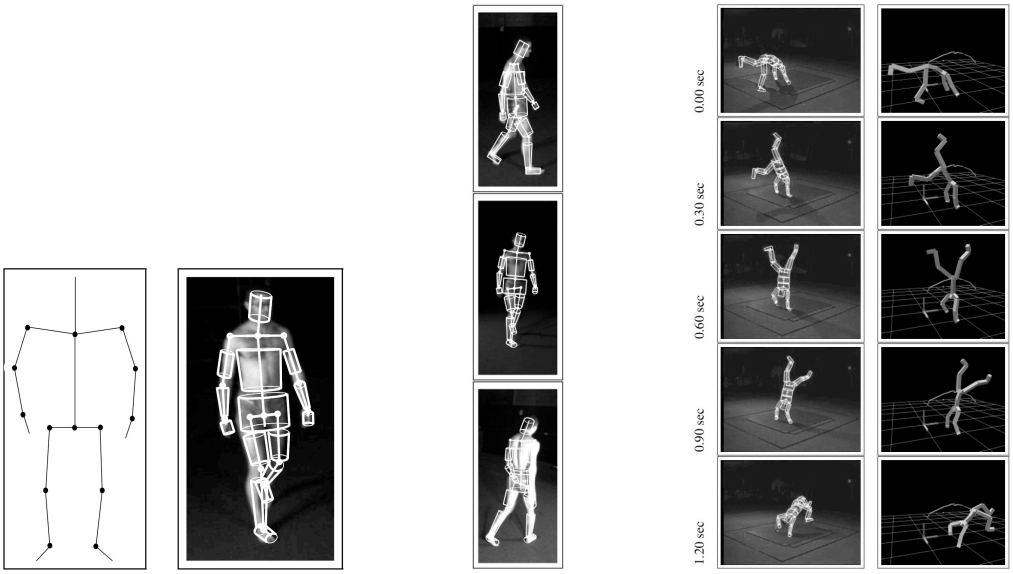


Abb. 2.9.: Darstellung der Positionierung der Kameras beim System von Kakadiaris et al.: (a) Paarweise orthogonale Position der Kameras. (b) Bild der Kamera an der Seite. (c) Bild der Kamera vorne. (d) Bild der Kamera oben. Bildquelle: [Kakadiaris and Metaxas, 1998].

Ansätzen ist die Verwendung von mehreren Kameras und die dadurch erheblich größere zu verarbeitenden Datenmenge. Eine beispielhafte Trackingszene einer Person, die einen Handstand zeigt, ist in Abb. 2.10(c) dargestellt.

Ebenfalls eine Bayes'scher Filter-Ansatz wird von Vondrak et al. verfolgt [Vondrak et al., 2008]. Allerdings wird die Prädiktion der Modellkonfiguration für den nächsten Zeitschritt unterstützt durch die Verwendung einer physikalischen Simulation der Körperdynamik. Das verfolgte Körpermodell besteht aus 13 Teilen mit 31 Freiheitsgraden. Das resultierende System wurde verglichen mit einem Standard-Partikelfilter und einem Annealed Particle Filter, und zeigt dabei bessere Ergebnisse auf Kosten einer höheren Laufzeit. Die Geschwindigkeit ist angegeben pro Partikel, bei den eingesetzten 250 Partikeln in der durchgeführten Evaluation führt das zu einer Geschwindigkeit von etwa $15 - 17\,\text{Hz}$. Die Evaluation erfolgt auf Datensätzen aufgezeichnet mit drei respektive vier Kameras.

Im Gegensatz zu diesen Partikelfilter-basierten Ansätzen verwendet die Arbeit von Feldmann et al. [Feldmann et al., 2010] das System von Knoop (aus [Knoop, 2007]) für das Einpassen des Modells. Entsprechend verwenden auch beide Systeme das gleiche Menschmodell (dargestellt in Abb. 2.6(a)). Der Unterschied besteht in der Akquisition der verwendeten Daten, die bei Feldmann nicht aus einem auf einem Roboter angebrachten 3D-Sensor stammen, sondern aus einem Kamera-Netz generiert werden. Dabei wird eine Extraktion der Silhouette auf allen Kamerabildern vorgenommen, und aus den Schnitten der Silhouette eine Voxel-Karte generiert, aus der wiederum die für das Tracking benötigten 3D-Punkte berechnet werden. Das System wurde sowohl offline mit acht Kameras mit einer Aufnahmegeschwindigkeit von $50\,\text{Hz}$ getestet, als auch online mit drei Kameras mit einer Aufnahmegeschwindigkeit von $25\,\text{Hz}$. In letzterem



(a) (b) (c)

Abb. 2.10.: Details zum Tracking von Deutscher et. al.: (a) Verwendetes Mensch-Modell (links: kinematische Kette, rechts: resultierendes Zylindermodell). (b) Ansichten der drei verwendeten Kameras auf die gleiche Szene. (c) Beispiel einer Trackingsequenz beim Beobachten einer Person, die einen Handstand macht. Bildquelle: [Deutscher and Reid, 2005].

Fall wird eine Geschwindigkeit des Gesamtsystems von 13 Hz erreicht. Im Gegensatz zum System von Knoop treten (durch die Sicht von allen Seiten) weniger Verdeckungen auf, und damit steigt die Trackingqualität, insbesondere bei Bewegungen, die bei Sicht aus nur einer Richtung nicht gut beobachtbar sind.

Andere von auf Roboter-Hardware basierenden Beobachtungssystemen zu unterscheidende Ansätze verwenden einzelne oder mehrere Videokameras aus größerer Entfernung, um eine allgemeinere Beobachtung von mehreren Menschen durchzuführen. Die Arbeit von Park et al. [Park and Aggarwal, 2002] verwendet eine einzelne Videokamera mit geringer Auflösung (320×240 Pixel) in einer dreistufigen Prozesskette. Im ersten Schritt wird die Silhouette der beobachteten Person(en) isoliert, und mittels auf den ersten Frames der Beobachtung trainierter *Gaussscher Mixturen* (engl. *gaussian mixture model (GMM)*) werden die einzelnen Pixel klassifiziert. Im nächsten Schritt werden die Pixel zu Regionen zusammengefasst (unter Einsatz von *Markov Random Fields (MRFs)* und zusätzlicher Heuristiken). Schließlich erfolgt eine Zuordnung der Regionen zu den Körperteilen eines vereinfachten Menschmodells (siehe Abb. 2.11(a)), wobei hierarchisch zunächst eine Zuordnung zu Kopf, Oberkörper und Unterleib mittels geometrischer Heuristiken und anschließend (farbbasiert) eine Zuordnung zu detallierteren Körperteilen durchgeführt wird. Der gesamte Prozess und das Verfolgen der Körperteile

über aufeinanderfolgende Frames läuft mit einer Geschwindigkeit von 15 Hz, Abb. 2.11(b) zeigt zwei beispielhafte Erkennungen (links das Kamerabild, rechts die Silhouette mit farbig markierten Körperteilen). Allerdings wird im Gegensatz zu den bisher vorgestellten Verfahren nicht die Konfiguration der einzelnen Körperteile verfolgt.

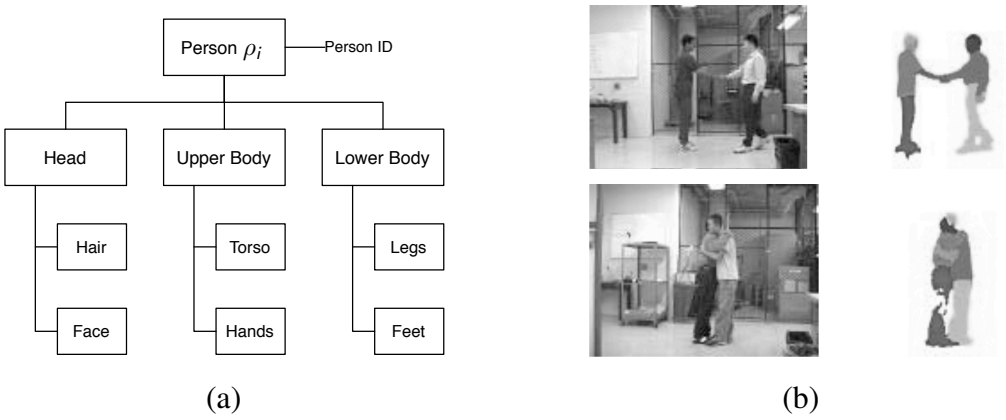


Abb. 2.11.: Details des Trackingverfahrens von Park et al.: (a) Verwendetes Menschmodell. (b) Zwei beispielhafte Trackingergebnisse (links beobachtetes Kamerabild, rechts Körperteile in der Silhouette farbig markiert). Bildquelle: [Park and Aggarwal, 2002].

Ein weitgehend automatisiertes System wird von Mittal et al. [Mittal et al., 2003] vorgeschlagen. Sowohl die Initialisierung des Systems als auch die Wahl vieler Parameter, die in anderen Systemen vorgegeben, heuristisch gewählt oder gelernt werden müssen, werden hier ohne Parametrisierung auf den Daten direkt durchgeführt. Zunächst werden die Silhouetten von Personen im Kamerabild erkannt und getrennt (durch Pixel-weise Bayes-Klassifikation mit Hinweisen aus Farbmodellen und eventuellem Wissen über die Position der Personen). Anschließend werden die Silhouetten automatisch in verschiedene Regionen unterteilt, die durch Rückprojektion einzelne Körperteile in 3D liefern. Schließlich werden Hypothesen über die Identität und den Zusammenhang dieser Körperteile gebildet und bezüglich ihrer Wahrscheinlichkeit bewertet. Das System kann mit einer einzelnen oder mehreren Kameras verwendet werden (es gibt Ergebnisse für den Einsatz mit 1, 2 und 5 Kameras), beim Einsatz einer einzelnen Kamera läuft das System mit einer Geschwindigkeit von $0,1$ Hz, Abb. 2.12(a) zeigt ein beispielhaftes Ergebnis für diesen Fall.

Parameswaran et al. beschäftigen sich in [Parameswaran and Chellappa, 2004] mit einem Teilproblem bei der Verfolgung von Bewegungen mittels einer Monokamera, der Posenrekonstruktion. Mit einem 2D-Bild und einer Menge definierter Körperpunkte in diesem Bild als Eingabe wird eine 3D-Pose der Person bestimmt, in einem im Torso zentrierten Koordinatensystem. Das genutzte Körpermodell besteht aus 14 Gelenken, deren Konfiguration die Pose abbilden (in Abb. 2.12(b) ist rechts das Modell abgebildet).

(a) (b)

Abb. 2.12.: (a) Beispielergebnis des Trackings von Mittal et al., Bildquelle: [Mittal et al., 2003]. (b) Körpermodell und beispielhaftes Ergebnis der Posenrekonstruktion nach Parameswaran et al., Bildquelle: [Parameswaran and Chellappa, 2004].

Ein lernbasierter Ansatz zur Posenerkennung in Sequenzen von Monokamerabildern wird von Agarwal et al. in [Agarwal and Triggs, 2006] vorgestellt. Auf den extrahierten Silhouetten von Personen werden unterschiedliche Regressionsmodelle, unter anderem *Relevance Vector Machines (RVMs)*, trainiert, deren Ausgabe ein Vektor von 55 Werten ist, die das verwendete Menschmodell darstellen. Die 55 Werte setzen sich zusammen aus den je Freiheitsgraden für jedes der 18 modellierten Gelenke und 1 Winkel für die globale Orientierung. Die eigentliche Regression läuft in Echtzeit, die Extraktion der Silhouette wird offline durchgeführt. Die Dauer des Lernvorgangs ist abhängig vom eingesetzten Regressor, und liegt zwischen 2 und 20 Minuten. Ein beispielhaftes Ergebnis der Posenrekonstruktion ist in Abb. 2.13(a) dargestellt.

Fontmarty et al. stellen in [Fontmarty et al., 2007, 2008] ein Trackingsystem vor, das zwei mit größerem räumlichem Blick angebrachte Monokameras verwendet (siehe Abb. 2.13(b) für ein Beispiel der Kamerasicht). Das verwendete Menschmodell modelliert den Oberkörper des Menschen mit 14 Freiheitsgraden, und verwendet für das Einpassen des Modells auf die Beobachtung eine Kombination des *ICONDENSATION*-Algorithmus und eines *Annealed Partikelfilters* (bezeichnet als *I-Annealed particle filter*), mit verschiedenen aus den Kamerabildern gewonnenen Hinweisen wie Kanten, Farbmodellen der einzelnen Körperteile u.ä.. Das System erreicht eine Geschwindigkeit zwischen 1 Hz und 4 Hz, abhängig von der genauen Wahl der genutzten Merkmale.

2.1.3. Tracking mittels applizierter Sensoren

In gewissen Anwendungsbereichen ist die Anbringung von Sensoren an das zu beobachtende Objekt oder die zu beobachtende Person gängige Praxis, die Gründe für den Einsatz solcher spezialisierter Systeme sind vielfältig. Die wichtigsten sind die dadurch erreichbare höhere Ge-

(a) (b)

Abb. 2.13.: (a) Beispielergebnis des Trackings von Agarwal et al., Bildquelle: [Agarwal and Triggs, 2006]. (b) Beispielergebnis des Trackings von Fontmarty et al., Bildquelle: [Fontmarty et al., 2008].

nauigkeit der Beobachtung (beispielsweise in der Medizin und den Sportwissenschaften eine wichtige Anforderung), die Möglichkeit zur Messung von Größen mit den spezialisierten Sensoren, die durch äußere Beobachtung nur indirekt erschlossen werden können (wie beispielsweise Beschleunigungen), sowie die Eigenschaft, dass solche Sensoren häufig die Beobachtung einer Größe ohne Verdeckungen ermöglichen. Letzteres ist eine Eigenschaft, die beispielsweise bei der Beobachtung von Händen und Objektmanipulationen eine große Rolle spielt.

(a) (b)

Abb. 2.14.: Zwei typische Datenhandschuhe: (a) Datenhandschuh *CyberGlove II* mit 22 Freiheitsgraden. (b) Datenhandschuh *5DT Data Glove 5 Ultra* mit 5 Freiheitsgraden.

Zur Unterstützung von kamerabasierten Trackingverfahren werden häufig spezielle Marker eingesetzt, die an wichtigen Punkten des Körper der Versuchsperson angebracht werden. Das von der Firma *Vicon Motion Systems*[9] entwickelte Vicon-System verbindet spezielle, Infrarot-reflektierende Marker mit Kameras, die Infrarot-Emitter integrieren. Die Marker können durch ihre reflektive Eigenschaften in den Kamerabildern sehr einfach segmentiert und hochgenau verfolgt werden. Auf Software-Seite wird eine Integration der von den verschiedenen Kame-

[9]Firmen-Webseite: `http://www.vicon.com/`

ras beobachteten Marker durchgeführt, mit dem Resultat einer Menge von 3D-Punkten, deren Positionen genauer bestimmt werden können als bei handelsüblichen 3D-Sensoren. Für das Verfolgen von Bewegungen wird ein Modell verwendet, auf dem die Positionen der angebrachten Marker ebenfalls entsprechend markiert sind. Durch Optimierungsverfahren wird dann die Konfiguration des Modells bestimmt, die für den kleinsten Fehler bezüglich der beobachteten Marker sorgt. Fehler resultieren einerseits aus den üblichen Fehlern von Kameras (durch Rauschen, Auflösung und Digitalisierung), andererseits aber auch aus Bewegungen der Marker, die auf der Haut oder Kleidung angebracht sind. Durch die elastischen Eigenschaften des menschlichen Körpers sind die relativen Positionen der Marker zueinander nicht unbedingt fix, sondern können sich in einem gewissen Rahmen ändern. Die Kameras sind in verschiedenen Ausführungen verfügbar, mit Auflösungen von 1 bis 16 Megapixeln sowie Frameraten von 120 Hz bis 1000 Hz.

Noch einen Schritt weiter geht das *Impulse Motion Capture System* der Firma PhaseSpace Inc.[10]. Hier werden aktive LED-Marker eingesetzt, die eine eindeutige Identifikation erlauben und damit das Verwechseln von Signalen unterschiedlicher Marker verhindern können. Das System bietet eine Geschwindigkeit von bis zu 480 Hz bei einer Auflösung der Kameras von 12 Megapixel. Wie auch beim Vicon-System müssen mindestens zwei Kameras eingesetzt werden, um eine Triangulation für die Bestimmung der 3D-Position der Marker durchzuführen. In der Praxis werden für die meisten Zwecke aber erheblich mehr Kameras benötigt, um Verdeckungen zu vermeiden, obwohl das *Impulse*-System durch die eindeutig identifizierbaren Marker weniger anfällig ist für kurzzeitige Verdeckungen einzelner Marker.

Als Alternative zur Verfolgung eines Körpermodells werden teilweise die Daten von am Körper angebrachten Beschleunigungssensoren verwendet, beispielsweise bei [Mäntyjärvi et al., 2001] und bei [Huynh and Schiele, 2005]. Abhängig von der Anzahl der Anzahl und der Anbringung der Sensoren können die Bewegungen unterschiedlicher Körperteile unterschiedlich genau wahrgenommen werden.

Im Gegensatz zu dieser relativen Beobachtung der Bewegungen besteht auch die Möglichkeit, mit einem magnetfeld-basierten Trackingsystem direkt die Absolutbewegung mittels am Körper angebrachter Sensoren zu messen (im Ergebnis ähnlich Verwendung von mit Kameras beobachteten Markern, aber ohne die Verdeckungs-Problematik). Solche Systeme sind kommerziell verfügbar, beispielsweise in Form des *Flock of Birds*-Systems der Firma *Ascension Technology Corporation*[11]. Die Funktionsweise dieser Systeme basiert auf dem Aussenden eines pulsierenden Magnetfeldes, das von den entsprechenden Sensoren empfangen und ausge-

[10] Firmen-Webseite: `http://www.phasespace.com/`

[11] Firmen-Webseite: `http://www.ascension-tech.com/`

wertet werden kann. Nach einer Kalibrierung der Sensoren kann die relative Lage (also Position und Orientierung) des Sensors zum Sender in hoher Frequenz (125 Hz beim oben genannten Sensor) bestimmt werden.

Auf einer anderen Detailebene, speziell in Systemen zum *Programmieren durch Vormachen (PdV*, werden detaillierte Beobachtungen der Hände benötigt. Kameras sind in diesem Anwendungsbereich sehr fehleranfällig, da Finger sehr kleine Strukturen darstellen, die noch dazu schon bei einfachen Objektmanipulationen Verdeckungen zum Opfer fallen können. Letzteres ist auch der Grund dafür, dass sogar sehr robuste Kamera-Varianten mit Markern große Probleme haben. Daher werden in diesem Bereich häufig sogenannte *Datenhandschuhe* eingesetzt, die direkt die Fingergelenkwinkel messen. In Datenhandschuhen sind *Dehnmessstreifen* eingenäht, bimetallische Streifen, die bei Krümmung eine messbare Widerstandsänderung aufweisen, aus der die Auslenkung der einzelnen Gelenke bestimmt werden kann.

Es gibt verschiedene kommerziell vertriebene Produkte dieser Art, die sich grob in Handschuhe mit feiner Auflösung und solche mit grober Auflösung unterteilen lassen. Handschuhe mit feiner Auflösung haben 14 und mehr Dehnmessstreifen eingearbeitet, während in Handschuhen mit grober Auflösung meist 5 Dehnmessstreifen (einer pro Finger) verarbeitet sind. Ein Beispiel für die erste Klasse ist der Datenhandschuh *CyberGlove II* der Firma *CyberGlove Systems*[12] (siehe Abb. 2.14(a)) mit 22 Freiheitsgraden, die die Handkonfiguration beschreiben. Ein Beispiel für die zweite Klasse ist der *5DT Data Glove 5 Ultra* der Firma *Fifth Dimension Technologies (5DT)*[13] (siehe Abb. 2.14(b)) mit fünf Dehnmessstreifen, je einem auf jedem Fingerrücken.

Vor allem die feinauflösende Variante wird für verschiedene Untersuchungen verwendet, beispielsweise für Arbeiten im Bereich des Programmierens-durch-Vormachen am HIS Dillmann in [Ehrenmann et al., 2003; Pardowitz, 2007; Jäkel et al., 2010], zur Erkennung von statischen und dynamischen Griffen bei der Handhabung von Objekten.

2.1.4. Bewertung

Die vorgestellten Trackingssysteme bieten einen repräsentativen Überblick über die große Bandbreite der Ansätze, die zur Verfolgung von Personen zur Verfügung stehen. Deutlich erkennbar ist dabei, dass für eine genauere und robustere Beobachtung einerseits mehr Rechenaufwand benötigt wird, und andererseits auch meist mehr Sensoren eingesetzt werden müssen. Im Gegensatz dazu stehen Verfahren, die auch auf Robotern und ähnlichen autonomen Systemen zum Einsatz kommen können. Hier muss aufgrund der beschränkten Sensorik und Rechenkapazität

[12] Firmen-Webseite: `http://www.cyberglovesystems.com/`
[13] Firmen-Webseite: `http://www.5dt.com/`

Tab. 2.1.: Vergleich der verschiedenen Klassen von Verfahren zur Beobachtung von menschlichen Bewegungen. Die Bewertungskriterien Geschwindigkeit und Qualität der Beobachtung sind in die diskreten Werte *niedrig*, *mittel* und *hoch* unterteilt.

Ansätze	**Beobachtungsziel**	**Sensoren**	**Geschwindigkeit**	**Qualität**
Für Einsatz auf Robotern geeignete Systeme	Ganzkörper oder Oberkörper	Monokamera / Stereokamera / Tiefenbildkamera	hoch	niedrig
Kameranetzwerke	Ganzkörper	mehrere verteilte Kameras	niedrig	mittel
Marker-basierte Ansätze	Ganzkörper oder beliebige Teile	mehrere Kameras, Marker	hoch	hoch
Beschleunigungssensoren	Geschwindigkeit einzelner Körperteile	Beschleunigungssensoren an Körper befestigt	hoch	mittel
Datenhandschuhe	Hände und Finger	Datenhandschuhe mit Dehnmessstreifen	hoch	hoch

ein Kompromiss gefunden werden zwischen besserer Qualität der Ergebnisse und vertretbarem Aufwand. Allerdings ist hier auch deutlich erkennbar, dass die vor allem in jüngster Zeit entwickelten neuen Sensoren (speziell zur direkten Wahrnehmung von 3D-Informationen) und schnellere Rechner große Fortschritte ermöglicht haben. Tabelle 2.1 fasst die Unterschiede zwischen diesen unterschiedlichen Anwendungsklassen zusammen.

Für eine besonders präzise Beobachtung der Bewegungen einer Person ist der Einsatz von Marker-basierten Trackingsystemen im Moment die einzige mögliche Wahl. Aus diesem Grund kommen Systeme wie das weit verbreitete Vicon-System auch als Gold-Standard zur Evaluation von anderen Trackingverfahren zum Einsatz. Die Qualität der Personenbeobachtung auf mobilen Robotern hat große Fortschritte gemacht in den letzten Jahren, aber die Einschränkungen durch den festgelegten Blickwinkel und die beschränkten Rechenressourcen begrenzen den erreichbaren Fortschritt in diesem Anwendungsfall. Ein wichtiger Aspekt für die praktische Verwendung ist dabei auch die automatische Detektion von Personen zur Initialisierung des Trackings, für die noch keine einheitlich akzeptierte Lösung in Verwendung ist.

Zur Beobachtung einzelner Details, speziell der Hände, ist der Einsatz entsprechender Spezialsensoren wie Datenhandschuhe zur Zeit noch alternativlos, da aufgrund der gerade bei den Händen schnell auftretenden Verdeckungen eine auch nur grobe Beobachtung der Bewegungen kaum möglich ist, wenn nur die Sicht von außen genutzt werden kann.

2.2. Interpretation menschlicher Bewegungen

Die Erkennung menschlicher Bewegungen stellt eine wichtige Fähigkeit für Service-Roboter dar, die im menschlichen Umfeld proaktiv handeln sollen. Aber auch in anderen Anwendungsgebieten, man denke hier beispielsweise an die Überwachung von öffentlichen Bereichen, ist die reine Beobachtung von Bewegungen nicht der Hauptzweck, sondern die zweckdienliche Interpreation der beobachteten Bewegung stellt die Hauptaufgabe dar. Ansätze zur Erkennung von menschlichen Aktivitäten können nach verschiedenen Gesichtspunkten kategorisiert werden. Drei häufig verwendete Kriterien sind die folgenden:

Anwendungsgebiet Das Anwendungsgebiet ist entscheidend für die Art der zu erkennenden Aktivitäten verantwortlich, und kann mitentscheidend sein für die Sensorik, die eingesetzt werden kann.

Datenherkunft Die Herkunft der Daten ist sowohl abhängig von den verwendeten Sensoren, als auch von eventuell eingesetzten Methoden zur Modellbildung (beispielsweise Trackingverfahren).

Klassifikationsverfahren Unterschiedliche Klassifikationsverfahren resultieren in unterschiedlichen Stärken und Schwächen bei der Erkennung verschiedener Aktivitäten.

Die folgende Darstellung des Standes aktueller Forschungsarbeiten auf dem Gebiet der Aktivitätserkennung gruppiert die Arbeiten anhand des Anwendungsgebietes. Die Beschreibung der einzelnen Ansätze ist dann nach untergeordneten Kriterien geordnet, zunächst nach dem Gesichtspunkt, ob die Aktivitätserkennung auf rohen Sensordaten erfolgt oder ob zunächst aus den Sensordaten ein abstraktes Modell bestimmt, dann schließlich nach den zugrundeliegenden Sensoren, falls es in dem Anwendungsfeld hier unterschiedliche Ansätze gibt.

Im Folgenden sei eine *menschliche Aktivität* informell definiert als eine in einem überschaubaren Zeitraum beobachtbare Handlung eines Menschen, der im weitesten Sinn eine Bedeutung (beispielsweise durch eine eindeutige und allgemeinverständliche Bezeichnung) zugeordnet werden kann. Eine formale Definition wird in Abschnitt 4.1.2 gegeben.

Forschung zur Erkennung menschlicher Aktivitäten wird im Rahmen unterschiedlicher Anwendungsbereiche durchgeführt, die ihrerseits starken Einfluss auf die Art und Ausprägung der Erkennung haben. Einerseits werden die für die Erkennung interessanten Aktivitäten durch die Anwendung bestimmt. Beispielsweise sind die beim Überwachen einer Flugzeughalle interessanten Aktivitäten (eine Beispielaktivität: KOFFER ABSTELLEN) in den meisten Fällen nicht identisch mit denen, die bei der Beobachtung und automatischen Aufzeichnung einer Besprechung (eine Beispielaktivität: NOTIZEN MACHEN) interessant sind. Andererseits werden die

Arbeitsbedingungen hinsichtlich Sensorik, Zeitanforderungen etc. von den Rahmenbedingungen der Anwendung bestimmt. Beispielsweise sind für Überwachungsaufgaben (eine Beispielaktivität: VERFOLGT ANDERE PERSON) meist ganze Netzwerke von Kameras verfügbar, die aber nur aus einer gewissen Distanz wahrnehmen können, während andererseits Serviceroboter (eine Beispielaktivität: ZEIGEN) zwar auch eine Wahrnehmung aus großer Nähe durchführen können, dafür aber auf die Beobachtung aus einer einzigen Perspektive beschränkt sind.

Weitere überblickshafte Darstellungen mit teilweise anderen Schwerpunkten finden sich in [Aggarwal and Cai, 1997], [Pavlovic et al., 1997], [Cedras and Shah, 1999], [Wu et al., 1999], [Wang et al., 2003] und [Hu et al., 2004].

2.2.1. Mensch-Maschine-Interaktion und Robotik

Anwendungen im Bereich der *Mensch-Maschine-/Mensch-Roboter-Interaktion (MMI/MRI)* basieren meist auf einer detaillierten Beobachtung des Menschen in dem Sinne, dass einzelne Körperteile beobachtet und unterschieden werden müssen. Oft gibt es grosse Unterscheide bei der eingesetzten Sensorik im Vergleich zwischen MMI und MRI. Während bei ersterer noch umfangreiche Sensorik eingesetzt werden kann wie z.B. Kamera-Netzwerke, sind Anwendungen auf Robotern starken Einschränkungen unterworfen, sowohl bezüglich der verfügbaren Sensorik (aufgrund von beschränktem Raum und beschränkter Energie), als auch bezüglich der verfügbaren Berechnungskapazität. Auch ist die Sensorik oft heterogen, um durch die unterschiedlichen Vor- und Nachteile der verschiedenen Sensoren in jeder Situation durch Multisensor-Fusion möglichst gute Daten zu erhalten.

Davis et al. stellen in [Davis and Bobick, 1997] ein System zur Erkennung von Bewegungen vor, das die Handlungen einer beobachteten Person direkt anhand der resultierenden Bewegungen in den Bildern der Videosequenz analysiert. Aus der Bestimmung der bewegten Bereiche in Einzelbildern über eine Sequenz von Kamerabildern wird ein sogenanntes *Motion History Image (MHI)* aufgebaut, das als Vorlage (engl. *template*) für die repräsentierte Aktivität dient. Abb.2.15(a) zeigt ein beispielhaftes MHI. Der Vergleich von neuen Aufnahmen erfolgt über 7 Momente des Bildes. Aufgrund der Form des Templates ist dieser Ansatz sehr stark abhängig vom Sensor, insbesondere der eingenommenen Perspektive. Daher werden Trainingsdaten aus unterschiedlichen Blickwinkeln in 30°-Schritten genutzt. Die Repräsentation ist darüberhinaus nur schwierig einsetzbar für Aktivitäten, die von feinen Bewegungen bestimmt sind. Das System wurde getestet mit verschiedenen Aerobic-Bewegungen.

In [Madabhushi and Aggarwal, 1999, 2000] entwickelt Madabhushi einen Ansatz zur Unterscheidung von Aktivitäten basierend auf der Bewegung des Kopfes der beobachteten Person. Als Merkmale dienen die Bewegungen des Kopfes in x- und y-Richtung (wobei zwischen

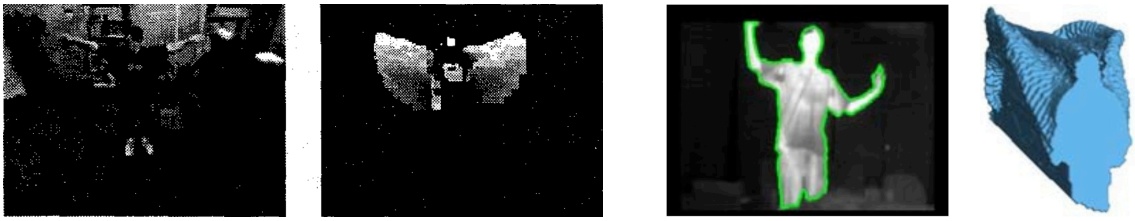
Front- und Seitenansicht unterschieden wird), die unter Betrachtung der letzten $5-9$ Werte mittels eines Nächster Nachbar-Klassifikators einer bekannten Aktivität zugeordnert werden. Die Erkennung beschränkt sich auf 12 Aktivitäten, die sich in der Art der Kopfbewegung stark genug unterscheiden, so dass eine Zuordnung einer beobachteten Bewegung noch möglich ist. Dabei geht das System von einer geschlossenen Welt aus, das bedeutet bei jeder Bewegung wird davon ausgegangen, dass sie zu genau einer der bekannten Klassen gehört.

Stiefelhagen et al. stellen in [Stiefelhagen et al., 2004] ein System zur natürlichen Interaktion zwischen Mensch und Roboter vor, das Spracherkennung, Kopfhaltung und Gestenerkennung kombiniert. Als Sensor für die Erkennung von Kopfhaltung und Gesten wird eine Farbstereokamera genutzt. Kopfposen werden mittels dreischichtiger neuronaler Netze erkannt, denen Farb- und Tiefendaten der Kamera als Eingabe dient. Gesten werden in drei Phasen unterteilt, die von dedizierten HMMs erkannt werden. Beides dient hauptsächlich der Unterstützung der Dialoge.

Speziell mit Zeigegesten beschäftigen sich Hofemann et al. [Hofemann et al., 2004], wobei hier ein adaptierter CONDENSATION-Algorithmus zur Erkennung der Bewegungen zum Einsatz kommt. Für verschiedene Handbewegungen, repräsentiert als Trajektorien von Geschwindigkeit und Richtungsänderung, werden Modelle angelegt, auf die mittels des Partikelfilters die beobachtete Bewegung möglichst gut angepasst wird, um die ähnlichste zu bestimmen. Darüberhinaus können als zusätzliche Information die Resultate der eingesetzten Objekterkennung eingespeist werden. Das System ist intendiert für die Verwendung in Dialogsituationen, in denen durch die Repräsentation der Handbewegungen eine gewisse Blickwinkel-Unabhängigkeit erreicht wird. Als Eingabedaten für das System dienen Monokamerabilder niedriger Auflösung, die mit 15 Hz aufgenommen werden, und damit die Maximalgeschwindigkeit des Systems bestimmen.

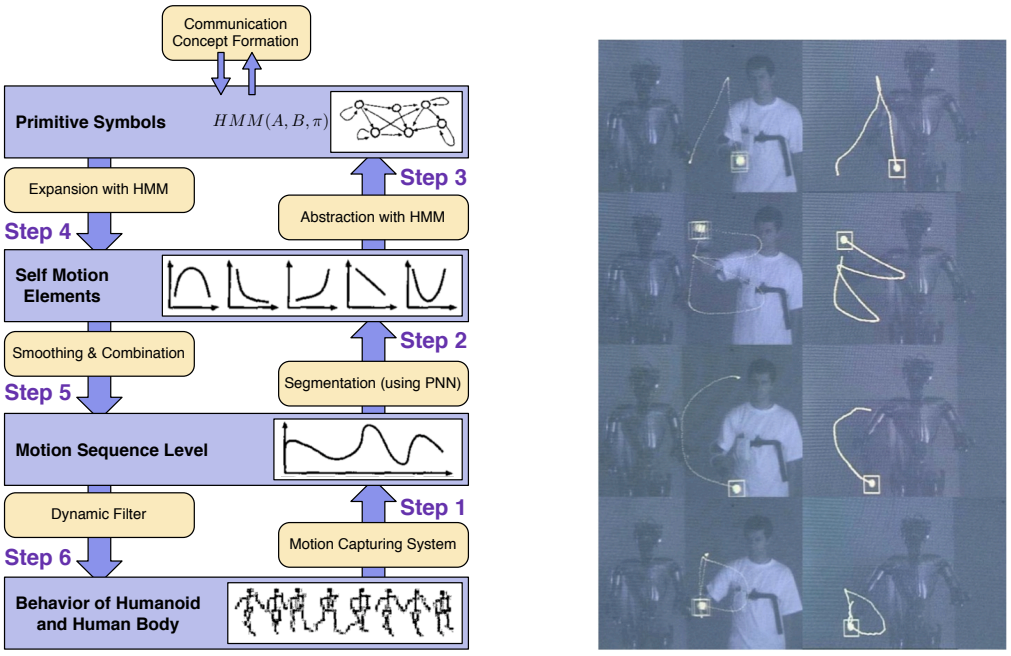
Nehaniv et al. untersuchen in [Nehaniv et al., 2005; Nehaniv, 2005; Otero et al., 2006] die Eigenschaften und unterschiedlichen Ausprägungen von Gesten, die bei der Interaktion von Mensch und Roboter und speziell bei der Demonstration von Handlungen für den Roboter auftreten können. Dabei werden fünf Arten von Gesten unterschieden bezüglich ihrer Bedeutung für die Interaktion. Das vorgestellte System zur Erkennung der Gesten verwendet dreischichtige neuronale Netze zur Klassifikation. Als Eingabe der Klassifikatoren dienen verschiedene Merkmale, die aus getrackten Menschmodellen extrahiert werden.

Einen anderen Ansatz verfolgen Becher et al. in [Becher et al., 2006]. Hier wird die Modellierung und Erkennung von Objekten und von menschlichen Aktionen eng gekoppelt. Die verwendeten Objektmodelle bestehen aus einer Menge von Merkmalen, die wiederum durch verschiedene Attribute repräsentiert sind. Um die Verbindung zwischen Aktionen und Objekten darzustellen, enthalten die ein Objekt beschreibenden Merkmale (zusätzlich zu den sie be-



(a) (b) (c)

Abb. 2.15.: (a) Beispielsszene mit Repräsentation als MHI im Aktivitätserkennungssystem von Davis et al., Bildquelle: [Davis and Bobick, 1997]. (b) Mittels Schwellwert auf Thermobild extrahierte Silhouette eines Menschen, Bildquelle: [Rusu et al., 2008]. (c) Aus einer Sequenz von Silhouetten generierte *space-time shape*, Bildquelle: [Rusu et al., 2008].



(a) (b)

Abb. 2.16.: Weitere Details zu Systemen, die Aktivitätserkennung im Rahmen von Imitiation Learning durchführen: (a) Nachgebildete Mimesis-Schleife im System zur Erkennung und Nachahmung von Bewegungen bei Inamura et al. (Diagramm analog zu [Inamura et al., 2001]). (b) Beispiel für die Erkennung und Reproduktion von Gesten bei Calinon et al., Bildquelle: [Calinon and Billard, 2004].

schreibenden Attributen) auch Verweise auf Aktionen, die mit dem Objekt durchgeführt werden können, dadurch dass es das fragliche Merkmal besitzt. Zusätzlich können diese Aktionen nur unter bestimmten Bedingungen (beschrieben anhand der Attribute) ausführbar sein oder die Attribute in bestimmter Weise verändern. Bewegungen andererseits sind als HMMs modelliert, die auf Bewegungssequenzen trainiert werden, die mit einem Vicon-Trackingsystem aufgezeichnet wurden. Die verknüpften Objekt- und Bewegungsmodelle können dann entweder genutzt werden, um die Erkennung von Objekten durch erkannte Bewegungen zu unterstützen, oder um umgekehrt Objekt-spezifische Bewegungen zu generieren für die Ausführung auf einem Roboter.

Ein auf der Arbeit von Yilmaz [Yilmaz and Shah, 2005] und Gorelick [Gorelick et al., 2007] (siehe auch Abschnitt 2.2.2) aufbauendes System zur Aktivitätserkennung wird von Rusu et al. in [Rusu et al., 2008] vorgeschlagen. Als Grundlage für die Erkennung dient hier die Silhouette eines beobachteten Menschen, die entweder durch Hintergrundsubtraktion oder durch das Anwenden eines Schwellwerts auf die Bilder einer Thermokamera (ein Beispiel für letzteres ist in Abb. 2.15(b) zu sehen) gewonnen werden kann. Eine Sequenz solcher Silhouetten wird aneinandergefügt und bildet damit ein sogenanntes *space-time shape*, siehe Abb. 2.15(c) für ein Beispiel. Die resultierende 3D-Form wird normalisiert, um Ausführungen der gleichen Aktion mit unterschiedlichen Geschwindigkeiten einheitlich behandeln zu können. Anschließend wird als Repräsentation der Form ein Histogramm von Merkmalswerten genutzt, die aus einzelnen Punktpaaren berechnet werden, wie es in [Rusu et al., 2007] schon erfolgreich zur Repräsentation und Erkennung von Objekten zum Einsatz kommt. Auf dieser Repräsentationsform werden SVMs trainiert, die bei den Evaluationen sehr gute Resultate zeigen. Durch die Verwendung von Silhouetten als Basis der Erkennung sind die resultierenden Erkenner abhängig von der genutzten Kameraperspektive. Daher sind in dem betrachteten Szenario die Kameras fest in der Umgebung angebracht.

Ähnlich zur Verwendung in der direkten Unterstützung bei der Interaktion zwischen Mensch und Roboter stellt die Erkennung menschlicher Aktivitäten bei der Programmierung von proaktivem Verhalten eines Roboters eine wichtige Informationsquelle dar, und ist ebenso in der Wahrnehmung ihrer Umwelt auf die Robotersensorik beschränkt. Ein Beispiel für diese Verwendung ist bei Schmidt-Rohr et al. zu finden [Schmidt-Rohr et al., 2008, 2010], der das in dieser Arbeit entwickelte System als eine Datenquelle für *Partially Observable Markov Decision Process (POMDP)*-Modelle verwendet, die proaktiv Handlungen des Roboters anstoßen können.

Weitere Verwendung finden Aktivitätserkennungssysteme in der Robotik zur Unterstützung von Imitation Learning-Ansätzen. Inamura et al. beschäftigen sich in [Inamura et al., 2001]

mit dem Problem, wie ein humanoider Roboter Bewegungen von Menschen lernen und imitieren kann. Dazu werden zunächst einfache Bewegungen (bezeichnet als *self motion elements*) als primitive Symbole genutzt, die mit Hilfe von probabilistischen neuronalen Netzen aus beobachteten Bewegungen segmentiert und erkannt werden. Auf diesen Symbolketten werden Hidden Markov Modelle trainiert, die dann einerseits zur Erkennung von beobachteten Bewegungssequenzen, andererseits zur Erzeugung von Ganzkörperbewegungen genutzt werden. Abb. 2.16(a) zeigt den Ablauf dieses Prozesses.

Den Ansatz von Inamura bauen Calinon et al. in [Calinon and Billard, 2004, 2005] weiter aus, indem das Alphabet primitiver Symbole (in Verbindung mit diskreten HMMs) ersetzt wird. Statt dieser Symbole werden Schlüsselpunkte der beobachteten Trajektorie (lokale Minima und Maxima) identifiziert, und für das Trainieren von kontinuierlichen Links-Rechts-HMMs genutzt. Die betrachteten Trajektorien bestehen dabei aus den Winkeln von Schulter und Ellbogen (beobachtet mittels an Schulter, Ober- und Unterarm befestigter *Xsens*-Bewegungssensoren, die mit einer Geschwindigkeit von 100 Hz die absolute Orientierung der Sensoren messen) und der Position der Hände (die mittels auf der Hand befestiger Marker und eines Stereokamerasystems mit 15 Hz beobachtet werden). Abb. 2.16(b) zeigt einige Beispiele für die Erkennung und Reproduktion von mit der Hand in die Luft gezeichneten Buchstaben mit diesem System.

Die Arbeiten von Kulić et al. in [Kulić et al., 2007, 2009a,b; Kulić and Nakamura, 2010] umfassen von der Erkennung von Aktivitäten über das automatische Lernen von Bewegungsprimitiven bis hin zur Generierung von der menschlichen Bewegung ähnlichen Bewegungsmustern viele der oben angesprochenen Verwendungsbereiche der Aktivitätserkennung. Mit einem Mensch-Modell mit in älteren Arbeiten 20 Freiheitsgraden, in aktuelleren Arbeiten 40 Freiheitsgraden, das mittels eines Marker-basierten Trackings gewonnen wird, werden HMMs und *Factorial Hidden Markov Models (FHMMs)* zur Erkennung (und Generierung) von Bewegungsprimitiven trainiert[14]. Mit Clustering-Verfahren können diese Modelle in einen *Bewegungsprimitiv-Graphen* (engl. *motion primitive graph*) eingetragen werden. Als übergeordnete Struktur werden *Growing Hidden Markov Models (GHMMs)* auf dem Graphen trainiert, die die größeren Bewegungsstrukturen, gewissermaßen das Zusammenspiel der Bewegungsprimitive, modellieren.

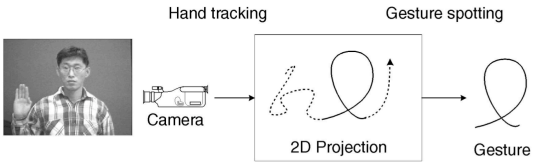
Unter dem etwas allgemeiner gefassten Begriff der Mensch-Maschine-Interaktion kommen einige Ansätze zu den bisher schon vorgestellten, die insbesondere auch die verwendete Sensorik und damit die zugrundeliegenden Daten erweitern. Einige typische Systeme dieser Art seien daher an dieser Stelle genannt.

[14] FHMMs erlauben eine verteilte Zustandsrepräsentation, bei der mehrere Zustände parallel zur Ausgabe des Modells beitragen können.

Als prototypische Anwendung entwickeln Lee et al. in [Lee and Kim, 1999] ein System zur Gestenerkennung, mit dem eine Powerpoint-Präsentation gesteuert werden kann. Dazu werden im Bild einer frontal beobachtenden Kamera Gesicht und rechte Hand segmentiert und die Trajektorie der Hand genutzt, um Links-Rechts-HMMs für eine fixe Menge von 10 Gesten (siehe Abb. 2.17(a)) zu trainieren. Abb. 2.17(b) zeigt die Anwendung mit einer Beispielgeste. Das System zeigt im definierten, beschränkten Rahmen (einfacher Hintergrund, nur Kopf und Hand im Bild) sehr gute Erkennungsraten in der Größenordnung von 93%, allerdings nicht in Echtzeit auf der genutzten Hardware. Ein weiteres Problem neben der Geschwindigkeit ist die Verzögerung bei der Erkennung der Gesten, da das System bis zur Erkennung der nächsten Geste wartet, bevor die vorherige als sicher erkannt ausgegeben wird.

Gesture	Command	Command Description
	Last	Go to last slide.
	First	Go to first slide.
	Next	Go to next slide.
	Previous	Go to previous slide.
	Start	Start presentation.
	Bye	Quit PowerPoint™.
	White	Fill screen with white color.
	Black	Fill Screen with black color.
	Hidden	Show hidden slide.
	Stop	End presentation.

(a)



(b)

Abb. 2.17.: Details zur Gestenerkennung von Lee et al.: (a) Katalog der genutzten und erkannten Gesten. (b) Vereinfachter Ablauf der Erkennung. Bildquelle: [Lee and Kim, 1999].

Niitsuma et al. schlagen in [Niitsuma et al., 2008] ein System vor, das zur Unterstützung der Interaktion von Benutzern mit einer intelligenten Umgebung (engl. *intelligent space*, in der Arbeit als *iSpace* abgekürzt) dient[15]. Als Datenquelle werden hierbei RFID-Transponder und Beschleunigungssensoren befestigt an den Objekten der Umgebung und ein Ultraschall-basiertes Lokalisierungssystem für die Hände verwendet. Die verwendeten Sensoren arbeiten mit einer Geschwindigkeit von 25 Hz, eine freie, spontane Interaktion zwischen Benutzer und System ist aber durch die Art der Sensorik nicht möglich.

Ein System zur Erkennung von Gesten, das zur Unterstützung der Interaktion von Menschen mit virtuellen Umgebungen, Robotern und ähnliche Anwendungen gedacht ist, wurde von Portillo-Rodriguez et al. in [Portillo-Rodriguez et al., 2008] vorgestellt. Gesten werden als Sequenzen von Zuständen des beobachteten Menschmodells definiert, die mittels *endlicher*

[15] Als *intelligente Umgebung* werden allgemein Bereiche bezeichnet, in denen Informations- und Kommunikationstechnologien unsichtbar in Gegenstände und die Umgebung integriert sind, um die Interaktion des Menschen mit ebendieser Umgebung zu unterstützen.

Automaten modelliert werden. Als Eingabe für die Automaten dienen die Zustände des Modells, die mit *Probabilistischen Neuronalen Netzen (PNNs)* erkannt werden. Ein großer Vorteil von PNNs ist an dieser Stelle, dass sie schnell und mit einer relativ kleinen Menge von Daten trainiert werden können. Als Daten dienen dabei mit einem VICON-System beobachtete Trajektorien eines Oberkörpermodells mit 13 Freiheitsgraden. Das ganze System läuft mit einer Geschwindigkeit von 50 Hz in der Evaluation, für die 7 komplexe Gesten genutzt wurden, die aus indischen Tanzfiguren ausgewählt wurden.

2.2.2. Überwachung und Videoanalyse

Im Bereich der Überwachung und Videoanalyse ist das Ziel nicht das Unterstützen der direkten Interaktion zwischen Mensch und Maschine, sondern eine (oft sogar zeitversetzte) Aufzeichnung und Analyse der beobachteten Handlungen des Menschen. Die Daten beschränken sich dabei in den meisten Fällen auf einen Strom von 2D-Videodaten, die je nach Anwendung (beispielsweise bei der automatischen Annotation von Sportereignissen wie Fussballspielen) auch aus großer Entfernung aufgenommen sein können.

Eine zeitnahe Interpretation führen sowohl Ansätze zur automatischen Überwachung, wie sie in öffentlichen Einrichtungen wie beispielsweise Flughäfen von Interesse sind, als auch Smartroom-Anwendungen, in denen Personen z.B. in Meetings oder Vorlesungen von einem automatischen System beobachtet werden, aus. Das Ziel ist dabei eine der Situation angemessene Unterstützung anbieten zu können.

Ein Ansatz für die Erkennung von Aktivitäten bei der Überwachung von öffentlichen Plätzen wie Parkplätzen oder Einkaufszentren wird von W. Niu et al. in [Niu et al., 2004] untersucht. Auf den Bildern von verteilten Kameras werden die Bewegungen von Personen verfolgt, und Aktivitäten wie *A folgt B* oder *A interagiert mit B* sollen erkannt werden. Der gewählte Ansatz verwendet dabei statistische Merkmale, die aus den Trajektorien der Personen berechnet werden können, wie relative Position und Geschwindigkeit zwischen Personenpaaren. Mit diesen Merkmalen trainierte SVMs mit Gausskernel zeigen gute Ergebnisse zwischen 80% und 100% Erkennungsrate in Experimenten mit realen Daten. Das System kann in Echtzeit genutzt werden, wenn die Auflösung der Kamerabilder nicht zu hoch gewählt ist.

Hongeng et al. beschäftigen sich in [Hongeng et al., 2000] mit der Erkennung von Aktivitäten in der automatisierten Überwachung am Beispiel von Parkplätzen. Dazu wird ein hierarchisches System System entwickelt, das auf unterster Ebene Bildmerkmale in Kamerabildern verfolgt wie Position und Form. Diese Merkmale dienen als Eingabe für die Bestimmung von Eigenschaften mobiler Entitäten (Menschen, Autos etc.) wie Textur, komplexe Formbeschreibungen und ähnliche entitätsspezifische Eigenschaften. Auf der nächsten Ebene, dem sogenannten *sce-*

nario level, werden zunächst einfache Aktivitäten mittels Bayes-Klassifikatoren bestimmt, die Erkennung von komplexeren Aktivitäten erfolgt wiederum durch die Verknüpfung mehrerer einfacher Aktivitäten in einer Graphenstruktur, die ähnlich einem vereinfachten HMM definiert ist.

Zur Verwirklichung von intelligenten Umgebungen haben Demirdjian et al. sich in [Demirdjian et al., 2002] mit der Erstellung von Aktivitätskärten (engl. *activity maps*) beschäftigt. Das Ziel dabei ist es, eine 2D-Karte der Umgebung aufzubauen, in der Zonen, in denen gleiche Aktivitäten durchgeführt werden, verzeichnet sind. Solche Aktivitätskarten können verwendet werden, um den räumlichen Kontext von Benutzern für weiterführende Unterstützung durch die intelligente Umgebung festzulegen. Zur Erkennung von Regionen für die Karte werden räumlich-zeitliche Merkmale wie Position und Bewegung (Geschwindigkeiten) mit einem Ballungsverfahren geclustert. Die dafür benötigten Daten werden von einem Stereokameranetz geliefert, das den Raum von oben mit 12 Hz beobachtet und von allen Personen im Raum Position und Größe (Höhe über dem Boden) liefert.

Auf Anwendungen in ähnlichen Szenarien zielt die Arbeit von Wojek et al. in [Wojek et al., 2006], in der in Büros angebrachte Mikrofone und Kameras genutzt werden, um HMMs zur Erkennung von typischen Aktivitäten in Büros zu trainieren und zu verwenden. Die betrachtete Aktivitätsmenge ist dabei fest und relativ klein gewählt (NIEMAND IM BÜRO, SCHREIBTISCHARBEIT, TELEFONIEREN, MEETING).

Ebenfalls HMM-basierte Ansätze für Szenarien in Büros und zuhause werden von Duong et al. und von Nuria et al. eingesetzt. Duong untersucht in [Duong et al., 2005] als spezielle Ausprägung *Switching Hidden Semi-Markov Models (S-HSMMs)* für die Erkennung von alltäglichen Aktivitäten zuhause (engl. *activities of the daily living (ADL)*). Als Eingabe dient das Ergebnis eines auf vier Videokameras basierenden Trackings, das die Position von Personen in einer diskreten Zelleneinteilung des beobachteten Raumes wiedergibt. Dabei kann das System auf unsegmentierten, ungelabelten Daten trainiert in einem relativ einfachen Szenario mit 6 verschiedenen Aktivitäten gute Ergebnisse liefern. In [Nuria et al., 2002] verwendet Nuria *Layered Hidden Markov Models (LHMMs)* zur Modellierung von typischen Büroaktivitäten ähnlich der Arbeit von Wojek. Der Ansatz, HMMs in mehreren Schichten einzusetzen, sorgt für eine abnehmende zeitliche Granularität und zunehmende Sensorunabhängigkeit auf höheren Schichten. Dadurch soll eine größere Unabhängigkeit von der exakten Umgebung erreicht werden, in denen die trainierte Erkennung eingesetzt wird. Als Sensordaten dienen dem System neben einer Videokamera zwei Mikrofone sowie ein 5-Sekunden-Puffer von Aktivitäten der Tastatur und Maus.

In der Gruppe von J. Crowley wurden in Grenoble verschiedene Untersuchung im Bereich Videoanalyse und Smartroom-Technologien durchgeführt, unter anderem im Rahmen des EU-Projektes *CHIL*[16]. Allgemein wird hier auf verteilte Sensorik mit vollständiger Raumübersicht gebaut. Auch der Begriff des *Kontextes* und mögliche Implementierungen dieses Konzeptes im Rechner sind wichtige Bausteine dieser Systeme.

Die Grundlagen für weitere Arbeiten zur Erkennung von Aktivitäten und Situationen auf kontextabhängige Weise werden in [Crowley, 2002, 2005; Coutaz et al., 2005] dargelegt. Neben formalen Definitionen für die wichtigen Begriffe wie *Kontext, Situation, Rolle, Relationen* werden auch abstrakte Architekturen für die benötigten Prozessketten zur Verarbeitung der Perzeption und zur Interpretation der Daten eingeführt, und eine Methodik für die Entwicklung von entsprechenden Systemen vorgeschlagen. Exemplarisch wird die Verwendung des Ansatzes für die automatisierte Aufzeichnung von Audio- und Videodaten von Vorlesungen mit mehreren Kameras verwendet (mit Aktivitäten wie AUF TAFEL ZEIGEN, FRAGEN, ANTWORTEN, ...).

In [Brdiczka et al., 2005, 2006a,b] wird eine Definition, Repräsentation und die Akquisition von Kontext für intelligente Umgebungen (als Beispiele werden Besprechungsräume und Hörsäle genannt) entwickelt, das zur Unterscheidung von problematischen Sensorwahrnehmungen und zur Unterstützung von Detektions- und Erkennungsalgorithmen eingesetzt werden kann. Kontext wird definiert als Zustand der Umgebung, in der die interessierenden Aktivitäten stattfinden. Als Repräsentation werden Zustandsautomaten genutzt, die sowohl in einer deterministischen Ausprägung als Petrinetze, als auch in einer probabilistischen Ausprägung als HMMs genutzt werden können, indem mittels einer allgemein definierten Abbildung in diese beiden Formalismen eine Instanziierung vorgenommen wird, siehe Abb. 2.18 und 2.19 für Beispiele beider Varianten. Um bei vorhandener Erkennung von Entitäten und ihrer Rollen und Relationen zueinander die Situationsmodelle zu adaptieren, werden Entscheidungsbäume genutzt, die in einem überwachten Szenario in der Lage sind, Situationsgraphen durch aufteilen von Situationsknoten zu verfeinern, wenn in einer gegebenen Situation deutlich unterschiedliche Reaktionen des Systems erforderlich sind.

Bei der Analyse von aufgezeichneten Videos, um sie beispielsweise für eine automatische Suche zu annotieren, erfolgt die Interpretation der beobachteten Handlungen nachträglich. Durch die weite Vernetzung und die schnell wachsenden Datenmengen im Internet ist diese Anwendung von großem praktischem Interesse und entsprechend Gegenstand aktiver Forschungen.

Yilmaz et al. in [Yilmaz and Shah, 2005], Blank et al. in [Blank et al., 2005] und [Gorelick et al., 2007] entwickelten unabhängig voneinander eine Darstellung von Aktivitäten, die durch Hintergrundsubtraktion gewonnene Silhouetten zu einer dreidimensionalen Darstellung einer

[16] Projekt-Homepage: `http://chil.server.de`

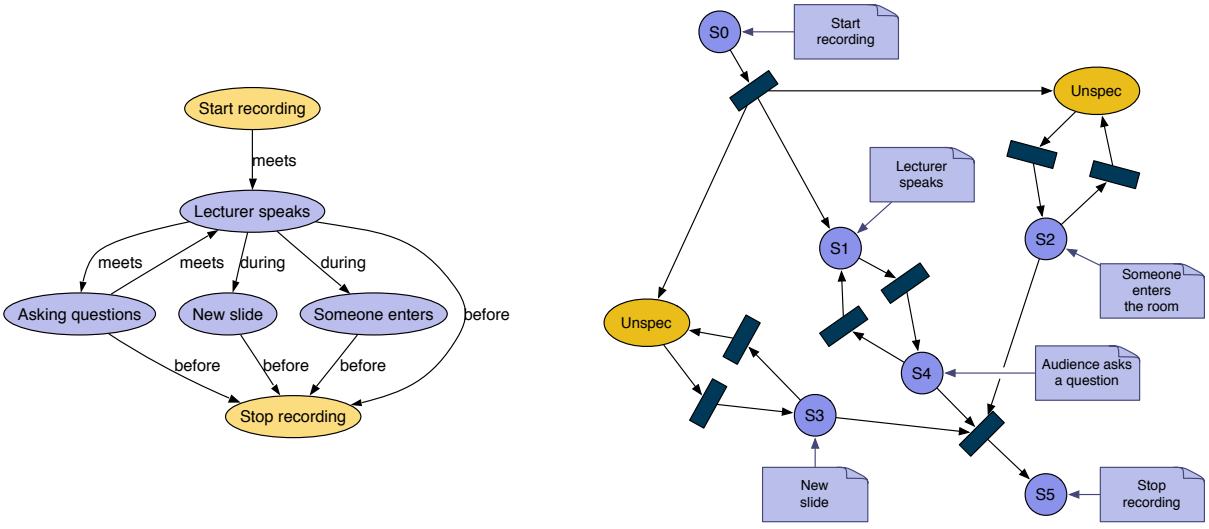


Abb. 2.18.: Beispiel für Realisierung von Situationsmodellen bei Brdiczka et al.: Deterministisches Situationsnetzwerk für ein „intelligentes Kameramann-System“ (links), realisiert als Petrinetz (rechts). (Abbildung analog zu [Brdiczka et al., 2006a]).

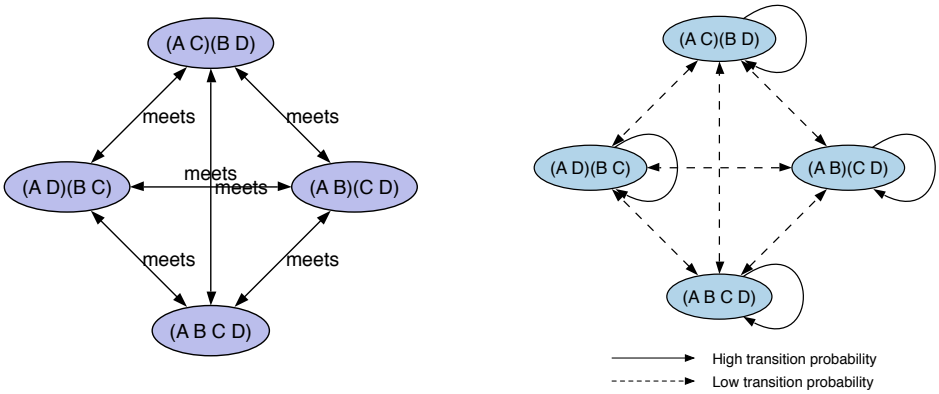
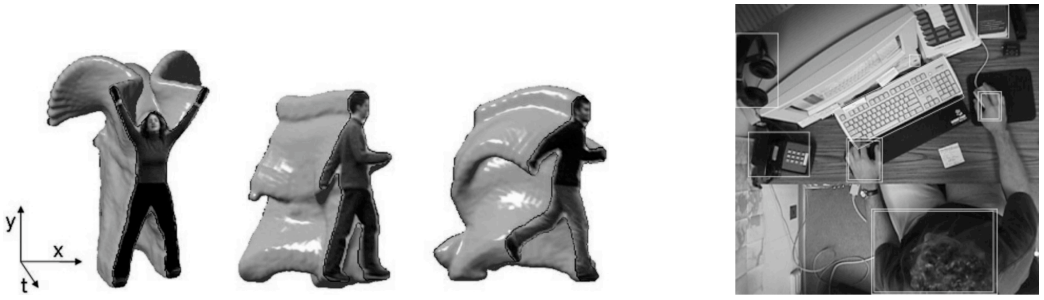


Abb. 2.19.: Beispiel für Realisierungen von Situationsmodellen bei Brdiczka et al.: Probabilistisches Situationsmodell für ein Treffen von 4 Personen mit unterschiedlichen Gesprächspaarungen (links), realisiert als HMM (rechts). (Abbildung analog zu [Brdiczka et al., 2006a]).

Aktivität verknüpfen (bei Yilmaz als *spatio-temporal volumes (SVT)* und bei Blank als *space-time shape* bezeichnet), siehe Abb. 2.20 für ein Beispiel dieser Darstellung. Die Verwendung dieser Darstellung bei den Gruppen unterscheidet sich aber in der Extraktion von Merkmalen daraus. Während bei Yilmaz die Volumen aus differentialgeometrischer Sicht betrachtet werden, um verschiedene interessante Punkte der Struktur (wie beispielsweise Sattelpunkte) zu bestimmen, die zusammengenommen die für die Erkennung genutzte Repräsentation einer Aktivität (bezeichnet als *action sketch*) darstellen. Die Erkennung einer Aktivität erfolgt dann durch das Vergleichen der Aktivität mit der als Action Sketch dargestellten aktuellen Sequenz, und Auswahl der ähnlichsten. Im Gegensatz dazu werden bei Blank die einzelnen Punkte des Volumens durch die Lösungen von Poisson-Gleichungen bewertet, die zur Identifikation bestimmter interessanter Punkte der Volumen (beispielsweise hervorstehende Punkte) aufgestellt werden. Die Erkennung auf diesen Merkmalen wird dann mittels eines nächster Nachbar-Klassifikators durchgeführt. Der grundsätzliche Ansatz ist auch für Videos niedriger Auflösung verwendbar, und bietet sogar eine eingeschränkte Unabhängigkeit vom Blickwinkel der Kamera.



(a) (b)

Abb. 2.20.: (a) Beispiel für die bei Yilmaz und bei Blank verwendete Darstellung von Aktivitäten als geschichtete Silhouetten, Bildquelle: [Gorelick et al., 2007]. (b) Sichtfeld der Überkopf-Kamera bei Moore et al., Bildquelle: [Moore et al., 1998].

Die Verbindung zwischen Aktionen und Objekten und die Möglichkeiten, die Erkennung beider sich gegenseitig unterstützen zu lassen wurde von Moore et al. untersucht [Moore et al., 1998, 1999]. Die Idee dabei ist es, auch den Kontext beobachteter Entitäten (Objekte, Personen) mit in die Beobachtung einzubeziehen und Relationen zwischen den Beobachtungen herzustellen. Aufbauend auf einer einzelnen Kamera, die senkrecht über einem Schreibtisch angebracht ist, werden HMMs zur Modellierung der Handbewegungen und Bayesmodelle zur Modellierung und Erkennung der Objekte verwendet. Dabei ist eine Hierarchisierung in Betracht gezogen, so dass eine Aktivität wie SCHREIBEN aus einer Menge von Aktionsmodellen wie ZEICHNEN, LÖSCHEN und STIFT BEWEGEN zusammengesetzt werden kann. In weiterführenden Untersuchungen in [Moore and Essa, 2002] wird ein Ansatz entwickelt, komplexes

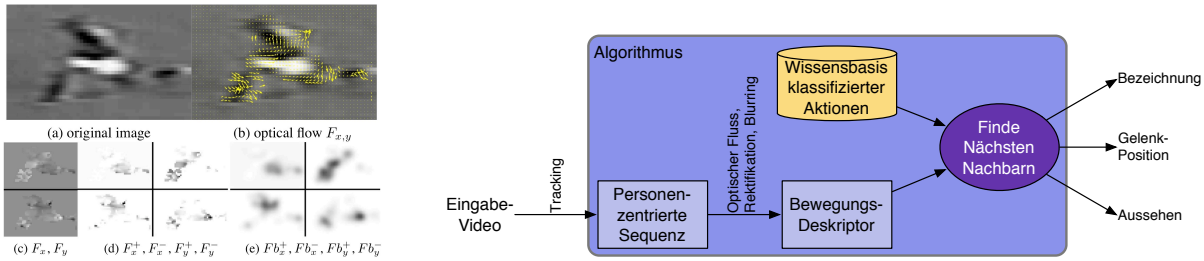
Multitasking-Verhalten von mehreren Personen mittels stochastischer Kontext-freier Grammatiken zu repräsentieren. Dieses System wird am Beispiel des Kartenspiels *Black Jack* experimentell evaluiert.

Efros et al. beschäftigen sich in [Efros et al., 2003] mit der Analyse und Annotation von Videodaten für Aufzeichnungen aus mittlerer Entfernung, d.h. mit dem Fall, dass Personen im Kamerabild eine Größe von etwa 30 Pixeln haben. In den meisten Fällen ist diese Auflösung zu gering für die Verwendung eines echten Modell-bildenden Trackingverfahrens, andererseits ist aber noch mehr Information enthalten als bei der Wahl eines noch weiteren Blickwinkels, bei dem nur noch die Gesamtbewegung von Menschen beobachtet werden kann. Als Repräsentation für die Bewegungen wird der optische Fluss des als Person segmentierten Bereichs berechnet, in positive und negative x- und y-Komponente aufgeteilt, und entschärft (s. Abb. 2.21(a) für die Zwischenergebnisse dieser Schritte). Die resultierenden Bewegungs-Deskriptoren werden mittels eines Nächster Nachbar-Ansatzes mit einer Wissensbasis klassifizierter Aktionen verglichen, siehe Abb. 2.21(b) für eine grafische Darstellung dieses Ablaufs.

2.2.3. Anwendungsunabhängige Ansätze

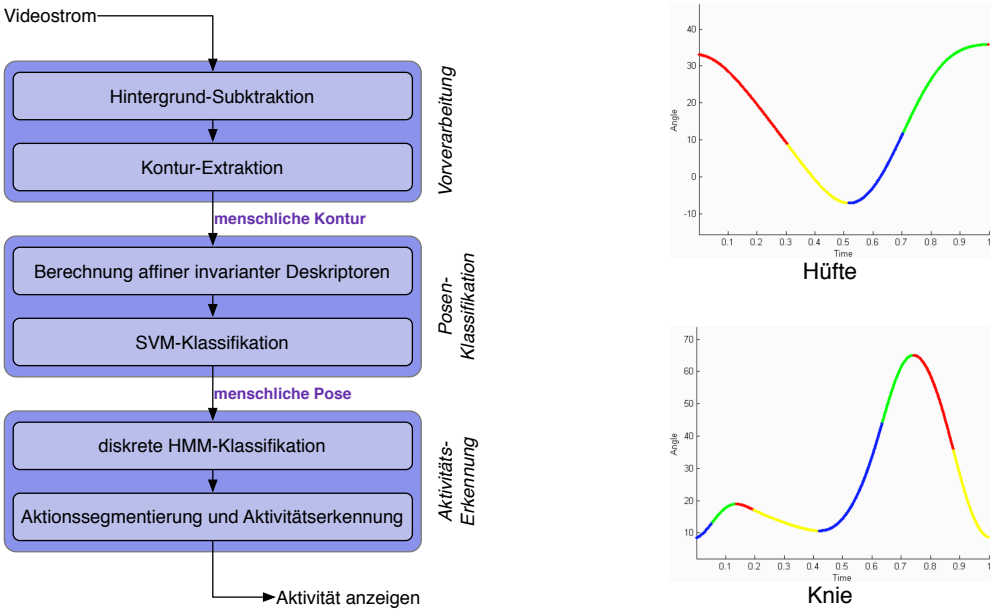
Die im Folgenden vorgestellten Arbeiten im Bereich der Erkennung von Aktivitäten zielen nicht auf eine bestimmte Anwendung wie die obigen Arbeiten. Stattdessen steht hier ein tieferes Verständnis der betrachteten Aktivitäten hinsichtlich Repräsentation und Aufbau im Zentrum des Interesses.

Ali und Agarwal untersuchen in [Ali and Aggarwal, 2001], wie Aktivitäten aus einer Seitenansicht des Menschen segmentiert und erkannt werden können. Dazu wird die Silhouette der Person segmentiert, und zu einem Skelett ausgedünnt. Auf dem resultierenden Strichmännchen werden die Beugewinkel von Hüften und Knien bestimmt. Zur Beobachtung dient dabei eine CCD-Kamera mit festem Blickwinkel und einer Aufnahmegeschwindigkeit von 15 Hz. Sowohl für die Erkennung von Segmentgrenzen zwischen verschiedenen Aktivitäten, als auch für die Erkennung von segmentierten Bewegungssequenzen werden nächster Nachbar-Klassifikatoren eingesetzt. Für die Segmentierung dienen dabei einzelne Frames als Trainingsdaten, während für die Erkennung der Merkmalsvektor als Sequenz von Gelenkwinkeln gegeben ist. Zum Ausgleich zwischen verschieden langen Sequenzen wird dabei der kürzere Vektor (Trainingsdaten oder Anfrage) auf die Länge des anderen Vektors interpoliert. Die untersuchten Aktivitäten sind GEHEN, SITZEN, AUFSTEHEN (aus Sitzen), VORBEUGEN, HOCHKOMMEN (aus Hocke), HOCKEN, SICH ERHEBEN. Das System läuft in Echtzeit im Rahmen der Sensorgeschwindigkeit.



(a) (b)

Abb. 2.21.: Details zum System von Efros et al.: (a) Zwischenergebnisse bei Berechnung von Bewegungs-Deskriptoren (*motion descriptors*), Bildquelle: [Efros et al., 2003]. (b) Verarbeitungskette zur Erkennung von Aktivitäten (Abb. nach [Efros et al., 2003]).



(a) (b)

Abb. 2.22.: Details zu den Arbeiten von Kellokumpu und Aloimonos: (a) Ablauf der Aktivitätserkennung nach Kellokumpu et al., (Diagramm analog zu [Kellokumpu et al., 2005]). (b) Zwei Beispiele für Segmentierungen und Symbolisierungen mit Kinemen aus HAL von Aloimonos et al.. Oben: Flexion/Extension der Hüfte, unten: Flexion/Extension des Knies. Bildquelle: [Guerra-Filho and Aloimonos, 2006a].

Einen Ansatz zur Erkennung von Aktivitäten ohne explizite Bildung eines Menschmodells untersuchen Kellokumpu et al. in [Kellokumpu et al., 2005]. Die Idee dabei ist ein Ansichtsbasiertes Verfahren, das zunächst (nach Extraktion der Kontur einer Person) die Pose mittels eines SVM-Klassifikators in eine diskrete Pose überführt, anhand von aus der Kontur berechneter affiner, invarianter Fourier-Deskriptoren. Die resultierenden Posen dienen dann als Eingabe für diskrete HMMs, die die eigentlichen Aktivitäten erkennen. Der Ablauf ist in Abb. 2.22(a) als Diagramm grafisch dargestellt. Die gesamte Verarbeitung eines Bildes dauert in dem System unter 10 ms, sodass eine Verwendung in Echtzeit möglich ist. Die betrachteten 15 Aktivitäten beinhalten beispielsweise WINKEN, BÜCKEN, FUSS HEBEN.

Sowohl in der Arbeit von Green et a. [Green and Guan, 2004a,b] als auch bei Husz et al. [Husz et al., 2007] wird eine Erkennung von komplexen Aktivitäten aufgebaut auf der Erkennung primitiver Einheiten, die bei Green *Dyneme* genannt werden, bei Husz dagegen *Aktionsprimitiv* (engl. *action primitive*). Während diese Basiseinheiten bei Green manuell definiert werden, wenn eine Aktivität mit den vorhandenen nicht beschrieben werden kann, wird bei Husz ein Clustering zur automatischen Generierung der entsprechenden Modelle (im Folgenden *APMs* für Aktionsprimitiv-Modell) vorgenommen. Allerdings müssen die Modelle anschließend noch manuell mit Beschreibungen versehen werden. Die Klassifikation bei Husz kombiniert in einem Bayesschen Ansatz die Ähnlichkeit der aktuellen Bewegung mit den bekannten APMs und der Auftrittswahrscheinlichkeit der APMs in den bekannten Aktivititäen. Im Gegensatz dazu kombiniert Green zunächst Dynemfolgen zu Skills (etwas größeren Einheiten von Bewegungen), die mit statistischen Auftrittswahrscheinlichkeiten von Paaren und Tripeln ähnlich der Spracherkennung gewichtet werden können. Anschließend erfolgt die Erkennung von Aktivitäten als Sequenzen von Skills. Zur Modellierung aller dieser einzeln Modelle werden HMMs eingesetzt. Beide verwenden einfache, aus einem dreidimensionalen Körpermodell extrahierbare Parameter (Gelenkwinkel, Positionen und Orientierungen, falls nötig Ableitungen dieser Werte) als Merkmale für das Training der benötigten Modelle.

Mit der Frage, wie ansichtsunabhängig Aktivitäten in Kamerabildern erkannt werden können beschäftigen sich F. Niu und Abdel-Mottaleb in [Niu and Abdel-Mottaleb, 2004]. Hierzu werden für jede Aktivität mehrere HMM-Modelle trainiert, auf Daten, die aus verscheidenen Blickwinkeln aufgezeichnet wurden. Als Eingabe für die Klassifikatoren dienen 208 Merkmale, die aus der im Kamerabild segmentierten Person berechnet werden können, und die Bewegung und die Form der Silhouette beschreiben. Für die Beschreibung der Bewegung wird die Silhouette in 64 Blöcke aufgeteilt, und in jedem Block der durchschnittliche optische Fluss berechnet (aufgeteilt in x- und y-Komponente liefert das 128 Merkmale). Für die Beschreibung der Form werden mittels eines *Eigenshape*-Ansatzes (im Wesentlich eine Hauptkomponentenanalyse auf

den konkatenierten Zeilen der Silhouette) 90 weitere Merkmale bestimmt. Das System wurde offline evaluiert, an mit 30 Hz aufgezeichneten Bildsequenzen niedriger Auflösung.

Aloimonos und Guerra-Filho entwickeln in [Guerra-Filho and Aloimonos, 2006a,b] unter der Bezeichnung *Human Activity Language (HAL)* einen Ansatz analog zur Spracherkennung, der ein linguistisches Rahmenwerk zur Modellierung, Erkennung und Rekonstruktion von menschlichen Aktivitäten darstellt. Das Analogon zur Phonologie, der Untersuchung der Bedeutung einzelner Laute in gesprochener Sprache, wird hier als *Kinetologie* (engl. *kinetology*) bezeichnet. Dabei werden Basiseinheiten von Bewegungen (*Kineteme*) gesucht, aus denen größere Bewegungen und mithin Aktivitäten aufgebaut sind. Als Eingabe werden allgemein Gelenkwinkel genutzt, ohne dass ihre Herkunft von Belang ist. Getrennt für jeden Gelenkwinkel wird jedem Zeitpunkt einer von vier Zuständen (als Kombination von positiver/negativer Geschwindigkeit und positiver/negativer Beschleunigung), bezeichnet mit **B**, **G**, **R**, **Y** zugeordnet, und Folgen gleicher Zustände zu einem Segment zusammengefasst, siehe Abb. 2.22(b) für zwei Beispiele einer solchen Segmentierung und Symbolisierung von kurzen Bewegungssequenzen in einzelnen Gelenken. Die Morphologie beschäftigt sich mit der Zusammensetzung der Grundeinheiten zu Worten, die in der HAL einzelne Aktivitäten (zusammengesetzt aus einer Folge von Kinetemen) darstellen. Das Wissen über den Aufbau dieser Wörter wird durch ein kombiniertes Lernen von sequentieller Grammatik (für einzelne Gelenke) und paralleler Grammatik (für mehrere Gelenke) gewonnen. Für die Repräsentation noch größerer Einheiten wird die Syntax untersucht, die definiert, wie Wörter zu Aktionssequenzen kombiniert werden können. Die Grundform eines Satzes besteht dabei aus einem Subjekt (Angabe des betrachteten Körperteils) mit einem Adjektiv (zur Angabe der Ausgangsstellung) und einem Verb (zur Angabe der durchgeführten Bewegung) mit einem Adverb (das eventuelle Variationen in der Ausführung näher beschreibt). Diese Grundsyntax kann entsprechend erweitert werden zu einer sequentiellen (mehrere Sätze) und parallelen (Betrachtung mehrerer gleichzeitig aktiver Gelenke) Syntax.

2.2.4. Bewertung

Die vorgestellten Arbeiten zur Erkennung von Aktivitäten zeigen deutlich die grosse Bandbreite von Anwendungen, für die solche Systeme eingesetzt werden können. Daraus resultieren unterschiedliche Anforderungen an die Systeme, aber auch Unterschiede in den Lösungsansätzen. Im folgenden soll ein Vergleich der Verfahren bezüglich zentraler Eigenschaften vorgenommen werden, die entsprechend der Diskussion wünschenswerter Eigenschaften für ein Aktivitätserkennungssystem in Kapitel 4 ausgewählt wurden:

Tab. 2.2.: Vergleich von Ansätzen zur Erkennung von Aktivitäten in verschiedenen Anwendungsgebieten. Betrachtet wird die Qualität der Systeme in den drei Kategorien *Geschwindigkeit & Verwendbarkeit*, *Erweiterbarkeit & Aufwand des Trainings* sowie *Einsatzbreite & Sensor-Spezifizität*.

Anwendung	**Geschwindigkeit & Verwendbarkeit**	**Erweiterbarkeit & Aufwand des Trainings**	**Einsatzbreite & Sensor-Spezifität**
Mensch-Roboter-Interaktion	⊕⊕	⊖	⊖
Mensch-Maschine-Interaktion	⊕⊕	⊙	⊙
proaktive Roboter	⊕	⊖	⊖
Imitation Learning	⊙	⊖	⊖⊖
intelligente Umgebungen	⊙	⊙	⊙
automatische Überwachung	⊖	⊙	⊕
Videoanalyse/-annotation	⊖⊖	⊕	⊕
unabhängige Ansätze	⊖	⊕	⊕⊕

Geschwindigkeit und Verwendbarkeit Diese Kategorie bewertet einerseits die Geschwindigkeit, mit der die Erkennung durchgeführt werden kann, andererseits die Eigenschaft, ob eine Aktivität erst vollständig beobachtet worden sein muss, bevor ihre Erkennung möglich ist. Grob kann hier in der Praxis unterschieden werden zwischen Verfahren, die mit mindestens 15 Hz laufen können (in Publikationen üblicherweise als Echtzeit bezeichnet, im Gegensatz zur *harten Echtzeitfähigkeit*, wie sie beispielsweise in der Regelungstechnik gefordert wird, und die eine feste, vorhersagbare Dauer von Operationen fordert), und eine instantane Erkennung der aktuellen Aktivität in jedem Frame erlauben; solche Verfahren, die zwar zeitnah eine erkannte Aktivität ausgeben können, aber das Ende einer Sequenz abwarten müssen, bevor eine Erkennung möglich ist. Und schließlich Verfahren, deren Berechnungszeit höher ist, sodass eine Verwendung für Systeme mit direkter Interaktion nicht sinnvoll ist (oft als *offline*-Verfahren bezeichnet).
Als schlecht werden hier Systeme bewertet, die sowohl eine sehr hohe Laufzeit für die Berechnung benötigen, als auch jeweils vollständige Aktionen zur Verfügung haben müssen. Als sehr gut werden im Gegensatz dazu Systeme bewertet, der Laufzeit eine Verwendung in Echtzeit erlaubt, und die auch schon teilweise erfasste Aktionssequenzen erkennen können.

Erweiterbarkeit & Aufwand des Trainings Die Erweiterbarkeit bewertet, inwieweit das System zur Erkennung neuer Aktivitäten erweitert werden kann (im Gegensatz zum Aspekt der Einsatzbreite werden dabei zusätzliche Aktivitäten in einem schon betrachteten Einsatzbereich erwogen). Der Aufwand des Trainings stellt in diesem Zusammenhang den zu betreibenden Aufwand dar, wenn ein zusätzlicher neuer Erkenner für das System eingelernt werden soll.
Als schlecht werden hier Systeme bewertet, die auf die Erkennung einer kleinen, festen Anzahl von Aktivitäten beschränkt sind. Als sehr gut werden dagegen Systeme bewertet, deren Konzept die Erkennung vieler Aktivitäten erlaubt, und bei denen das Einlernen solcher zusätzlicher Aktivitäten schnell und ohne großes Expertenwissen möglich ist.

Einsatzbreite & Sensor-Spezifität Die Einsatzbreite bewertet die Möglichkeit, das System für deutlich unterschiedliche Zwecke einsetzen zu können. Ein Beispiel für diesen Fall wäre die Verwendung eines Systems für die Erkennung von Gesten in einer Anwendung zur Erkennung von verschiedenen Gangarten. Die Sensor-Spezifität gibt an, wie stark die Erkennung von der Verwendung eines spezifischen Sensors abhängt, oder ob beispielsweise bei einem kamerabasierten System auch andere Kameras mit anderer Auflösung verwendet werden können, ohne dass die Erkennung neu trainiert werden muss.

Als schlecht werden hier Systeme bewertet, die nur zusammen mit einem bestimmten Sensor(-Modell) eingesetzt werden können, als sehr gut Systeme, die mit vielen, auch deutlich unterschiedlichen Sensoren für unterschiedliche Zwecke genutzt werden können.

In Tabelle 2.2 sind die oben genutzten groben Klassen von Anwendungsgebieten für Aktivitätserkennungssysteme gegenübergestellt und in ihrer Tendenz in diese Kategorien eingeordnet, bewertet mit einer fünfstufigen Skala ($\ominus\ominus / \ominus / \odot / \oplus / \oplus\oplus$) (wobei $\ominus\ominus$ die schlechteste, und $\oplus\oplus$ die beste Bewertung ist). Dabei gibt die Einordnung nur den Trend der jeweiligen Arbeiten wieder, spezifische Einzelsysteme können in einzelnen Kategorien auch abweichende Eigenschaften aufweisen.

Wie der Vergleich zeigt, lassen sich deutliche Stärken und Schwächen in den verschiedenen Anwendungsgebieten erkennen. Typische Ansätze im Bereich der Robotik arbeiten mit relativ hoher Geschwindigkeit, sind aber meist relativ stark auf spezifische Einsatzzwecke und die verwendete Sensorik zugeschnitten. Umgekehrt sind Systeme, die für eine einfache Erweiterbarkeit und eine große Einsatzbreite konzipiert wurden, meist nicht auf die in der Robotik benötigte Geschwindigkeit bei der Berechnung der Ergebnisse ausgelegt. Auch ist in solchen Systemen die Abhängigkeit von einem spezifischen Sensor zwar geringer als bei anderen Systemen; im Gegenzug sind aber meist mehr und andere Sensoren erforderlich, als es autonome Robotik-Anwendungen erlauben, beispielsweise Kameranetzwerke oder vom Benutzer getragene Sensoren.

Wie im nächsten Kapitel gezeigt wird, ist allerdings gerade eine Kombination dieser Eigenschaften wünschenswert für zukünftige Serviceroboter, um die allgemeine Wiederverwendung von erlerntem Wissen und damit einen möglichen Schritt aus Forschungslaboren in reale Anwendungsszenarien zu ermöglichen.

3. Basistechnologien und Verfahren des Maschinellen Lernens zur Aktivitätserkennung

Die vorliegende Arbeit liefert Beiträge in den Bereichen der Beobachtung und der Interpretation menschlicher Bewegungen. Im Folgenden werden die für das Verständnis nötigen Grundlagen und Schreibweisen dargestellt. Abschnitt 3.1 beschreibt das als Hauptdatenquelle verwendete Trackingsystem. Abschnitt 3.2 beschreibt Grundlagen des Merkmalsauswahl-Problems, Abschnitt 3.3 führt das Konzept von Klassifikatoren ein und präsentiert einige gebräuchliche Ansätze.

3.1. Trackingsystem *VooDoo*

Das Software-System *VooDoo* wurde entwickelt zum Tracking von menschlichen Bewegungen. Begonnen wurde die Entwicklung von Steffen Knoop et. al. [Knoop et al., 2005, 2006a,b] mit dem Ziel eines echtzeitfähigen Trackingsystems, das die Fusion der Daten von verschiedenen Sensoren erlaubt. Das System wird als Hauptdatenquelle für die Beobachtung menschlicher Bewegungen genutzt. Im Folgenden werden die wichtigsten Grundlagen für das Verständnis der Arbeitsweise des Trackings, dessen Leistungswerte und Randbedingungen beschrieben.

3.1.1. Überblick

Als algorithmische Grundlage des Trackings dient ein erweiterter *Iterative Closest Points (ICP)*-Algorithmus, der speziell angepasst wurde, um drei Aspekten Rechnung zu tragen:

1. Für das Tracking von menschlichen Körpern werden aus mehreren Gliedern bestehende Modelle benötigt. Der in Abschnitt 3.1.2 beschriebene Grundalgorithmus kann nur einzelne, starre Objekte behandeln. Um ein artikuliertes Körpertracking zur ermöglichen, werden künstliche Messpunkte genutzt, die für den Zusammenhalt der Gliedmaßen sorgen.

2. Da das System für die Verwendung auf autonomen Robotern entwickelt wurde, musste von einem eingeschränkten Blickwinkel (aus einer einzigen Blickrichtung) und daher von eingeschränkten Sensordaten ausgegangen werden. Daher sollte das System in der

Lage sein, zum Ausgleich alle verfügbaren Sensordaten zur Verbesserung der Ergebnisse einzusetzen.

3. Um den Rechenaufwand des Systems zu verringern, dass das Tracking in auch auf einem in einem autonomen Roboter verfügbaren Rechnern in Echtzeit genutzt werden kann, wurden einige Anpassungen bei den verwendeten Modellen vorgenommen. Insbesondere wird die Modell-Repräsentation auf deformierte Zylinder als Modell-Glieder beschränkt.

Im nächsten Abschnitt wird zunächst der ICP-Basisalgorithmus beschrieben, bevor die Struktur des erweiterten ICP-Algorithmus im darauf folgenden Abschnitt 3.1.3 im Detail dargestellt wird. Die Erweiterung für das Finden von Punktkorrespondenzen wird in Abschnitt 3.1.4, die ICP-kompatible Gelenkmodellierung in Abschnitt 3.1.5 beschrieben, im anschließenden Abschnitt 3.1.6 werden die sich eröffnenden Möglichkeiten zur Multisensor-Fusion beschrieben.

3.1.2. ICP-Algorithmus

Das Tracking basiert auf dem *Iterative Closest Points (ICP)*-Algorithmus nach Besl [Besl and McKay, 1992], der grundsätzlich dazu dient, den Abstand zweier Punktwolken zueinander zu minimieren. Ein typisches Einsatzgebiet für diesen Algorithmus ist die Registrierung von Daten aus mehreren Aufnahmen, wie es beispielsweise beim SLAM-Problem (*Simultaneous Localization and Mapping*) auftritt. Der Grundalgorithmus ist konzeptuell relativ einfach und durch die iterative Struktur kann ein günstiger Kompromiss zwischen schneller Ausführung und genauen Ergebnissen für viele Anwendungen gefunden werden, insbesondere auch für Echtzeit-Anwendungen.

Die Darstellung des ICP im Folgenden ist angepasst an die Verwendung für ein Trackingsystem und orientiert sich an [Demirdjian, 2003] und [Knoop, 2007]. Die folgenden Erklärungen gehen von der Annahme aus, dass ein (dreidimensionales) Modell M an eine Messung $\mathbf{S}$ (gegeben als Menge von 3D-Punkten) angepasst werden soll. Wiederholt (*iterative*) werden Korrespondenzen zwischen Messdaten und Modell hergestellt (*closest points*), und die Lage des Modells wird durch eine Transformation so angepasst, dass der Abstand zwischen den korrespondierenden Punkten minimiert wird, siehe Algorithmus 3.1.2 für eine algorithmische Beschreibung des Verfahrens.

Die Bestimmung von Punktkorrespondenzen kann abhängig von der Anwendung unterschiedliche Formen annehmen. Der Standard-Ansatz besteht im Finden der namensgebenden *Nächsten Punkte*, d.h. für jeden Punkt P_i der Messungen wird der nächste Punkt P_i' (mit minimaler euklidischer Distanz $d(P_i, P_i')$) des Modells gesucht. Es existieren aber auch Anwendungen, die

Algorithmus 3.1 Allgemeiner ICP-Algorithmus.

Eingabe: Modell M_{alt}, Messung **S** (geg. als Punktwolke), Abbruchschwellwert ε

Ausgabe: eingepasstes Modell M_{neu}

1: **repeat**
2: Bestimme Punktkorrespondenzen **K** zwischen M_{alt} und **S**
3: Transformation $T \leftarrow$ optimale Transformation zur Minimierung der Abstände in **K**
4: $M_{neu} \leftarrow T(M_{alt})$
5: $Fehler_{neu} \leftarrow Abstand(M_{neu}, \mathbf{S})$
6: **until** $Fehler_{neu} < \varepsilon$

mit einer festen Menge von ausgezeichneten Modellpunkten und Messungen arbeiten und auf diesen ein Matching durchführen.

Eine optimale Transformation, die eine vollständige Abbildung aller korrespondierenden Punkte aufeinander durchführt, ist im Allgemeinen aufgrund von Messfehlern und durch die Art der Durchführung der Zuordnung von Korrespondenzpunkten nicht möglich. Stattdessen wird eine Transformation T bestimmt, die die Summe der Abstandsquadrate zwischen den korrespondierenden Punkten minimiert. Der Ansatz nach Besl bestimmt nacheinander Translation und Rotation, für Details speziell zur Bestimmung der Rotation sei auf [Horn, 1987] verwiesen.

In der Praxis hat dieser allgemeine Algorithmus allerdings Probleme. Zum Einen ist es durch Sensorrauschen und die nur begrenzte Genauigkeit der Computermathematik normalerweise nicht möglich, das Modell so weit zu optimieren, dass der Fehler auf Null reduziert wird. Auf semantischer Ebene besteht das Problem, dass die bestimmten Punktkorrespondenzen nicht notwendigerweise die tatsächlichen Punktzuordnungen darstellen. Abhängig von der Form des Modells und den verfügbaren Sensordaten ist es möglich, dass trotz eines auf Null reduzierten Fehlers das Modell nicht die korrekte Lage erreicht, sondern in einem lokalen Minimum verharrt. Ein weiteres Problem für die Verwendung in einem Trackingverfahren ist schließlich, dass das Verfahren nur für statische, zusammenhängende Objekte entwickelt wurde. Für das Verfolgen menschlicher Bewegungen ist ein solches Modell meist nicht ausreichend, da der menschliche Körper aus mehreren beweglichen Teilen besteht. Ein typisches Modell mit ausreichendem Detailgrad für heutige Sensoren besteht aus 10 Körperteilen (Torso, Kopf, 2 Ober- und Unterarme sowie 2 Ober- und Unterschenkel). Der ICP-Grundalgorithmus ist nicht in der Lage, einen bestimmten Abstand zwischen den Körperteilen zu erhalten, gleichzeitig aber Bewegungen um freie Bewegungsachsen zu erlauben.

3.1.3. Angepasster ICP-Algorithmus

Der allgemeine ICP-Algorithmus wurde durch eine vorgeschaltete Pipeline mit zusätzlichen Cache-Speichern erweitert, die sequentiell gefilterte und verdichtete Informationen aus dem jeweils vorhergehenden Verarbeitungsschritt bzw. Cache enthalten. Durch diese Struktur ist die Einspeisung von zusätzlichen Informationen unterschiedlichen Detailgrades möglich, was die Grundvoraussetzung sowohl für die Fusion unterschiedlicher Sensordaten, als auch für die Einbindung von mehrgliedrigen Modellen ist. Abb. 3.1 zeigt die detaillierte, nummerierte Struktur, die im Folgenden beschrieben wird.

Als unbeschränkteste Daten werden freie 3D-Punktmessungen in einem Cache gesammelt (1). Unter Verwendung der momentanen Modellkonfiguration werden die Punkte durch einen umhüllenden Quader (engl. *Bounding Box*) um das Modell gefiltert (2). Die resultierende Datenmenge wird mit Messungen ergänzt, deren Zugehörigkeit zum Körper bekannt ist (3). Anschließend wird eine entsprechende Filterung auf der Ebene einzelner Körperteile durchgeführt (4), und die resultierenden Daten um Messungen, deren Zugehörigkeit zu einem bestimmten Körperteil bekannt ist, ergänzt (5). Abhängig von der verwendeten Sensorik sind die Daten zu diesem Zeitpunkt meist zu umfangreich für eine schnelle Verarbeitung, daher wird im letzten Schritt der Pipeline eine Ausdünnung der Punktemenge vorgenommen (6) (engl. *Downsampling*). Ab dieser Stelle beginnt der eigentliche ICP-Durchlauf.Mit den verbliebenen Punkten wird eine Bestimmung der nächsten Punkte auf dem jeweiligen Modellzylinder vorgenommen (siehe Abschnitt 3.1.5), mit dem Resultat einer Liste von Punktkorrespondenzen zwischen Messungen und Modellpunkten (7). Diese Korrespondenzen werden fusioniert mit direkt gemessenen Korrespondenzen (8), Korrespondenzen die aus 2D-Merkmalen generiert werden (9) (siehe Abschnitt 3.1.6 für Details) und Korrespondenzen, die aus der Modellierung von Gelenkwinkel-Beschränkungen generiert werden (10) (siehe Abschnitt 3.1.5). Unter Verwendung der berechneten Punktkorrespondenzen wird mittels *Kleinste Quadrate-Optimierung* eine Transformationen für jedes Modellglied bestimmt, die eine Verringerung der Abweichung von den Messungen erreicht (11). Damit ist eine ICP-Iteration abgeschlossen, und mittels eines Vergleichs der resultierenden Distanz zwischen Modell und Messungen (Änderung der Punktkorrespondenz-Abstände) und einem Schwellwert e wird entschieden, ob eine weitere Iteration durchlaufen wird (12).

Der Aufwand des Tracking-Algorithmus' ist proportional zur Anzahl der benötigten ICP-Schritte (diese Anzahl wiederum ist abhängig von der Größe der nachzuführenden Bewegung), und der Aufwand der einzelnen ICP-Schritte skaliert linear mit der Anzahl der Messpunkte auf dem Modell [Knoop et al., 2006b; Knoop, 2007]. In der Praxis ist das Verfahren auf aktuellen

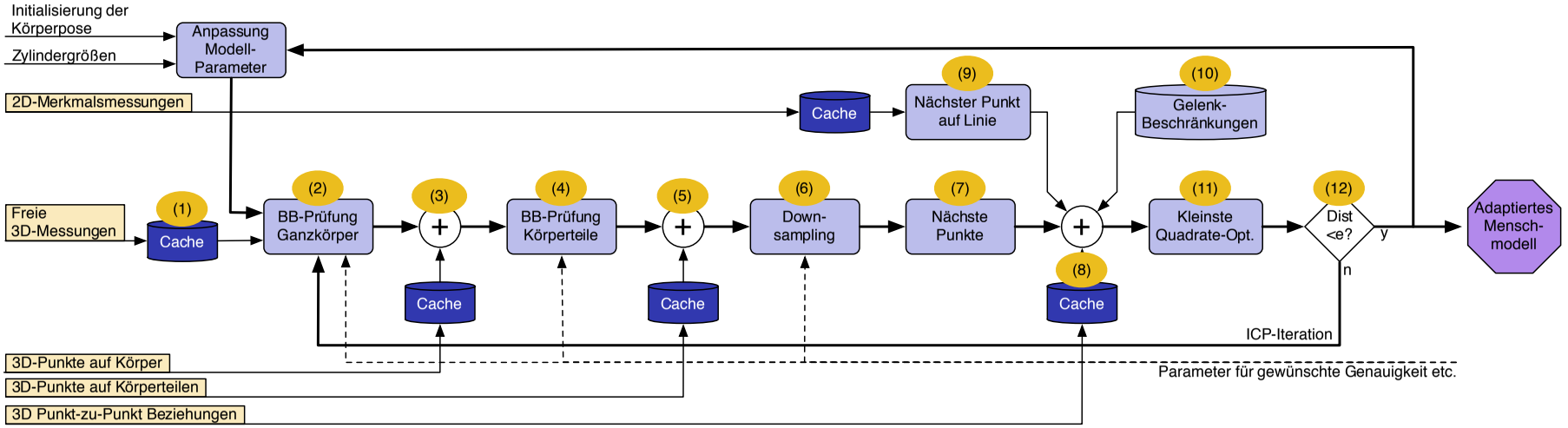


Abb. 3.1.: Ablauf und Cache-Struktur des für *VooDoo* angepassten ICP-Trackings. Die verschiedenen Sensordaten (links, hell orange markiert) sind durch Caches (dunkelblau markiert) von der Verarbeitungskette getrennt. Die wichtigsten Prozessschritte (auf orangem Feld nummeriert) werden im Haupttext näher erläutert: (1) Cache f. freie 3D-Messungen. (2) Filterung mittels Boundingbox um ganzen Körper. (3) Fusion von 3D-Punkten, die von getracktem Objekt stammen. (4) Zuordnung der Punkte zu Körperteilen mittels Boundingboxen. (5) Für jedes Körperteil Fusion von Messungen, die von diesem Körperteil stammen. (6) Ausdünnung (engl. downsampling) der Messungen. (7) Berechnung von Punktkorrespondenzen. (8) Fusion mit direkt gemessenen Punktkorrespondenzen. (9) Berechnung von Punktkorrespondenzen aus 2D-Messungen. (10) Bestimmung zusätzlicher Punktkorrespondenzen aus Gelenkwinkelbeschränkungen. (11) Modelltransformation mittels Kleinste Quadrate-Optimierung bestimmten. (12) Bewertung der erreichten Verbesserung, ggf. erneute Iteration.

Standard-PCs schnell genug, um eine Geschwindigkeit zwischen 20Hz und 25Hz zu erreichen, was ungefähr der erreichbaren Geschwindigkeit des verwendeten Tiefensensors entspricht.

3.1.4. Berechnung von Punktkorrespondenzen

Die Bestimmung von Punktkorrespondenzen zwischen Messungen und Modell sind ein zentraler Teil des ICP-Algorithmus, wie schon der Name ausdrückt. Für das *VooDoo*-System wurde diese Komponente angepasst, um statt einer zweiten Punktwolke eine parametrische Modellbeschreibung nutzen zu können. Als Modelle werden verallgemeinerte Zylinder genutzt, auf denen die nächsten Punkte zu den Messungen gesucht werden. Dadurch werden einerseits die Modelle nicht auf eine feste Menge von Punkten des Modells eingeschränkt, sondern die wirklich nächsten Punkte können genutzt werden. Andererseits ist eine effiziente Berechnung der Korrespondenzen möglich, die bei der Verwendung von Quadern oder menschenähnlicheren, aber unsymmetrischen Modellen nicht garantiert werden kann.

Algorithmus 3.2 Berechnung der nächsten Punkte (engl. *closest points*). (Kommentare erscheinen in geschweiften Klammern.)

Eingabe: Zylinder Z, Menge von Messungen $\mathbf{M}$ in Boundingbox von Z

Ausgabe: Menge von Korrespondenzen $\mathbf{K}$

1: **for all** $P \in \mathbf{M}$ **do**
2: { *Berechnung von* $h_Z(\cdot)$ *erfolgt mittels Gl. 3.1* }
3: **if** ($h_Z(P) >$ Länge von Z) $\wedge$ (Z ist kein Endzylinder) **then**
4: Verwerfe P
5: **else**
6: $p_{r_{b,a}}$, $p_{r_{b,b}} \leftarrow$ Projektion von P auf Zylinder-Basisvektoren
7: $P' \leftarrow$ näherungsweise Darstellung von P in Koordinatensystem von Z { Gl. 3.2 und Gl. 3.3 }
8: $P_O \leftarrow$ nächster Punkt zu P auf Z unter Verwendung von P' { mittels Gl. 3.6 }
9: **end if**
10: Füge (P, P_O) als neue Korrespondenz zu $\mathbf{K}$ hinzu
11: **end for**

Um für einen gegebenen Messpunkt P den nächsten Punkt P_O auf der Oberfläche des Zylinders Z zu bestimmen, werden mit den Bezeichnungen aus Abb. 3.2 die in Alg. 3.1.4 beschriebenen Schritte durchgeführt. Für jede Messung, die nicht zu einem Endzylinder (ein Zylinder, der an einem Ende mit keinem anderen Zylinder verbunden ist) gehört, wird geprüft, ob sie "neben" dem Zylinder liegt (durch Vergleich der mittels Gl. 3.1 ermittelten Höhe im Zylinderkoordina-

tensystem). Anschließend wird zunächst eine Näherung P'_O für P_O berechnet, siehe Gl. 3.2 – 3.5.

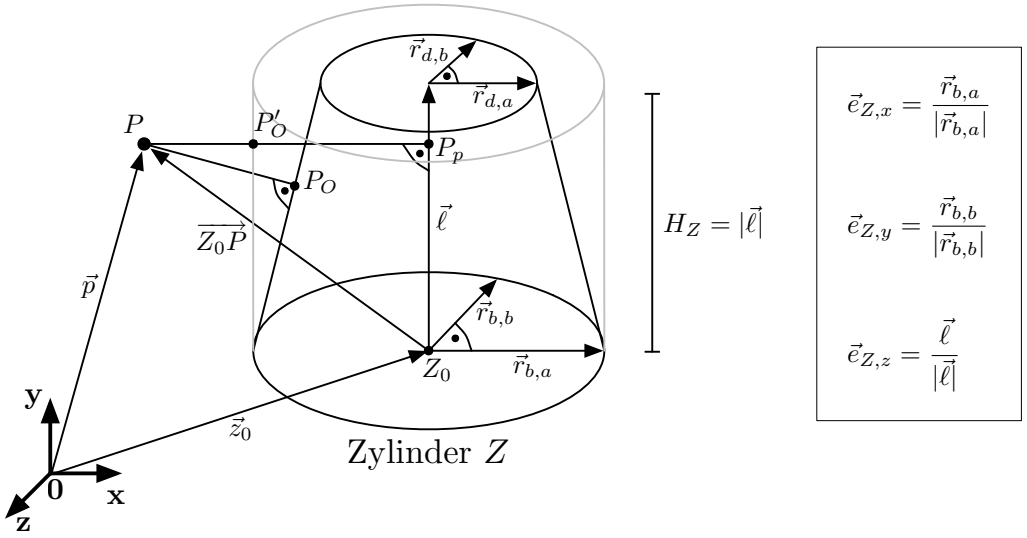


Abb. 3.2.: Berechnung des nächsten Punktes P_O zu P auf Zylinder Z, Abbildung nach [Knoop, 2007].

$$h_Z(P) = |\overline{Z_0P_p}| = \overrightarrow{Z_0P} \bullet \vec{e}_{Z,z} \qquad [3.1]$$

$$p_{Z_x} = \overrightarrow{Z_0P} \bullet \vec{e}_{Z,x} \qquad [3.2]$$

$$p_{Z_y} = \overrightarrow{Z_0P} \bullet \vec{e}_{Z,y} \qquad [3.3]$$

$$p'_y = \sqrt{\frac{1}{\left(\frac{p_{Z_x}/p_{Z_y}}{|\vec{r}_{b,a}|}\right)^2 + \frac{1}{|\vec{r}_{b,b}|^2}}} \qquad [3.4]$$

$$P'_O = \begin{pmatrix} p'_{O,x} \\ p'_{O,y} \\ p'_{O,z} \end{pmatrix} = \begin{pmatrix} p'_y \cdot \vec{e}_{Z,x} \\ p'_y \\ \min(H_Z, h_Z(P)) \end{pmatrix} \qquad [3.5]$$

Bei der Berechnung von P'_O wird dabei die Annahme $h_Z(P) \approx h_Z(P_O)$ getroffen, die für $(r_{b\cdot} - r_{d\cdot}) \ll H_Z$ angenommen werden kann. Diese Näherung dient hautpsächlich der Beschleunigung des Verfahrens durch Vermeiden besonders rechenintensiver Operationen. Ausgehend von P'_O wird dann der nächste Punkt P_O zu P berechnet, indem die aus dem unterschiedlichen Durchmesser von Ober- und Unterseite des Zylinders resultierende Steigung des Zylindermantels mittels Gl. 3.6 kompensiert wird. Das zu verwendende Vorzeichen in der Gleichung hängt vom Quadranten ab, in dem P liegt. Weitere Details sind in [Knoop, 2007] zu finden.

$$P_O = \begin{pmatrix} \pm p'_{O,x} + \frac{|\vec{r}_{d,a} - \vec{r}_{b,a}|}{h_Z} \\ \pm p'_{O,y} + \frac{|\vec{r}_{d,b} - \vec{r}_{b,b}|}{h_Z} \\ p'_{O,z} \end{pmatrix} \quad [3.6]$$

3.1.5. Modellierung von Gelenken

Das VooDoo-Tracking verwendet ein aus 10 verallgemeinerten Zylindern bestehendes Körpermodell, wie es in Abb. 3.3(a) gezeigt ist. Die Entscheidung für diese Modellierung des menschlichen Körpers resultiert aus verschiedenen Gründen. Die Granularität des Modells (10 Elemente, keine Modellierung der Hände oder Füße) resultiert aus den Eigenschaften der verfügbaren Sensoren, beispielsweise der Auflösung der eingesetzten Tiefenbildkamera. Argumente für die Wahl von Zylindern gegenüber z.B. Quadern sind einerseits die bessere Nachahmung der Form des menschlichen Körpers, andererseits die vorne beschriebene, einfachere Berechnung der nächste-Punkte-Korrelationen (die im ICP-Algorithmus benötigt werden).

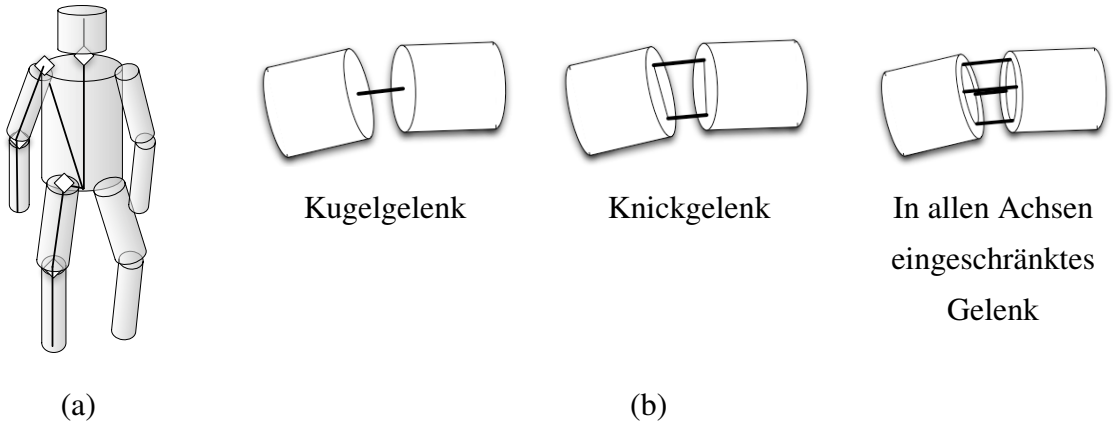


Abb. 3.3.: In VooDoo eingesetzte Modelle: (a) Beispielhaftes Menschmodell, bestehend aus 10 verallgemeinerten Zylindern. (b) Modellierung verschiedener Gelenktypen durch künstliche Korrespondenzen.

Auch die nicht aus der Abbildung ersichtliche Modellierung von Gelenken ist ein wichtiger Bestandteil des Körpermodells. Die Idee dahinter ist die Verwendung von künstlichen Punktkorrespondenzen, die analog zu Gummibändern wirken. Die Analogie basiert auf der folgenden Interpretation des ICP: Man kann die Arbeitsweise des ICP so interpretieren, dass auf jeden Korrespondenzpunkt des Modells eine Kraft wirkt, die ihn zu dem korrespondierenden Messpunkt „zieht“. Die resultierende Transformation entsteht dann durch die Überlagerung dieser Zugkräfte. Dieser Idee folgend werden zur Modellierung von Gelenken künstliche Korrespondenzen eingefügt, indem jeweils künstliche Messungen auf dem einen Glied eingefügt werden,

deren Korrespondenz auf dem anderen Glied definiert wird, und umgekehrt. Als Resultat wird eine Anziehungskraft zwischen beiden Gliedmaßen erzeugt, und somit wird ein „Auseinanderdriften“ der Gliedmaßen verhindert.

Neben dem Sicherstellen des Zusammenhalts der einzelnen Gliedmaßen bietet die Gelenkmodellierung darüberhinaus noch die Möglichkeit, Einschränkungen der Beweglichkeit in den einzelnen Gelenken zu modellieren. Dazu werden die künstlichen Punktkorrespondenzen an bestimmten Stellen relativ zu den involvierten Zylindern eingefügt und eventuell auch mehr als eine einfache Korrespondenz genutzt. Abb. 3.3(b) zeigt die drei auf diese Art realisierten Gelenkmodelle. Es besteht die Möglichkeit, keine Einschränkungen an die Bewegung zu stellen, die Bewegung hauptsächlich in einer Bewegungsachse zu erlauben (Scharniergelenk) oder allgemein die Beweglichkeit in allen Bewegungsachsen zwar zu erlauben, aber einzuschränken.

3.1.6. Möglichkeiten zur Multisensor-Fusion

Das *VooDoo*-System wurde mit besonderem Blick auf die Möglichkeit zur Fusion unterschiedlicher Sensoren entwickelt. Durch die kaskadierte Struktur der unterschiedlichen Caches können Sensordaten unterschiedlicher Qualität im passenden Schritt der Verarbeitungspipeline zu den schon genutzten Daten hinzugefügt werden. Zur Erläuterung einige Beispiele, wie Daten in unterschiedlichen Stufen genutzt werden können:

- Punktwolken von beliebigen 3D-Sensoren (Laserscanner, ToF-Kameras, Microsoft Kinect, ...) werden in den ersten Cache für *Freie 3D-Messungen* eingefügt.
- Durch die Kombination von Stereokamera und Marker können direkt 3D-Messungen bestimmter Körperpunkte gemacht werden, die dann in den Cache *3D Punkt-zu-Punkt Beziehungen* eingefügt werden.
- In 2D-Farbbildern können verschiedene Punkte mit üblichen Bildverarbeitungsmethoden getrackt werden, die durch Rückprojektion eine Gerade liefern. Die resultierenden Kombinationen aus Gerade und zugeordnetem Modellpunkt können in den Cache *Closest point on line* eingefügt werden.

Dieses Konzept zur Sensor-Fusion wurde am HIS mit unterschiedlichen Sensor-Kombinationen erfolgreich in der Praxis getestet. Die dabei zum Einsatz gekommenen Sensoren waren eine ToF-Kamera *SwissRanger*, eine Stereokamera vom Typ *SVS*, ein Laserscanner der Firma Sick, und ein Magnetfeld-Tracker *Flock-of-Birds* angebracht auf dem Handrücken von *CyberGlove*-Datenhandschuhen.

Die Fusion mit 2D-Informationen, die mit klassischen Bildverarbeitungsmethoden aus Farbkamera-Daten gewonnen werden, wurde ebenfalls getestet, siehe [Knoop et al., 2006a]. Die dabei genutzten Informationen sind die Position des Kopfes gewonnen mittels eines Moduls zur Gesichtsdetektion, und die Position der Hände gewonnen aus einem Tracking der Hautfarbe. Durch Rückprojektion wird aus den 2D-Positionen eine dreidimensionale Gerade bestimmt, auf der sich die entsprechenden Punkte befinden müssen, und in den Cache für 2D-Merkmalsmessungen eingefügt. Die Nutzung dieser zusätzlichen Merkmale stabilisiert insbesondere die Position der Unterarm-Zylinder des Modells.

3.2. Merkmalsauswahl

Ein wichtiger Teilschritt in vielen maschinellen Lernproblemen ist die Auswahl geeigneter Merkmale, um beispielsweise eine Klassifikation durchzuführen. Im Folgenden wird eine kurze Einführung in die Problematik und Lösungsansätze gegeben, um den Rahmen für die in Abschnitt 6.2.1 präsentierten Arbeiten abzustecken. Eine ausführlichere Einführung in das Thema ist in [Guyon and Elisseeff, 2003] zu finden, eine tiefe Behandlung des Themas gibt es in [Liu and Motoda, 2008].

3.2.1. Überblick

Eine häufige Aufgabenstellung im Bereich des Maschinellen Lernens ist die Auswahl geeigneter Merkmale und Merkmalsteilmengen für eine gegebene Aufgabenstellung. Dabei werden im Detail Ziele verfolgt, die sich in den Randbedingungen bezüglich der Ursache und der gewünschten Ergebnisse unterscheiden lassen. Die Standard-Aufgabenstellungen sind dabei die folgenden:

(I) Nur rohe Sensordaten sind verfügbar. Was geeignete Merkmale für die Aufgabenstellung sind, ist noch vollständig unbekannt.

(II) Aus den Sensoraufnahmen der interessierenden Phänomene werden Merkmale extrahiert, aber es ist unbekannt, ob diese Merkmale geeignet sind oder ob es besser geeignete Merkmale gibt.

(III) Aus den Sensordaten wird eine große Menge von Merkmalen berechnet. Allerdings ist die Berechnung aller Merkmale in jedem Zeitschritt zu aufwändig für die Anwendung. Deshalb soll eine kleinere Teilmenge verwendet werden, die nur die relevantesten Merkmale der Obermenge enthält.

(IV) Es sind zu wenige und/oder nur verrauschte Trainingsdaten verfügbar. Daher soll die Lernaufgabe auf die wirklich relevanten, die Aufgabe treffend beschreibenden Merkmale beschränkt werden.

Die ersten beiden Punkte (I) und (II) werden meist unter dem Stichwort *Merkmalskonstruktion* (engl. *feature construction*) zusammengefasst. Dabei geht es um die Fragestellung, wie aus einer gegebenen Daten- oder Merkmalsmenge neue, besser geeignete Merkmale konstruiert werden können. Typische Lösungsverfahren sind *Hauptkomponentenanalyse (engl. Principal Component Analysis PCA)* und *Lineare Diskriminantenanalyse (engl. Linear Discriminant Analysis LDA)* [Duda et al., 2001, Seite 114ff]. Auch Untersuchungen zu nichtlinearen Kombinationen und Transformationen von Merkmalen wurden durchgeführt, beispielsweise bei [Piramuthu and Sikora, 2009].

Bei den beiden Punkten (III) und (IV) hingegen geht es um die Auswahl einer relevanten Merkmalsteilmenge aus einer gegebenen (größeren) Menge von Merkmalen (engl. *relevant feature subset selection*). Üblicherweise wird dabei einer von zwei Grundansätzen verfolgt. Sogenannte *Filter-Verfahren* versuchen, algorithmisch zu einer Relevanzbewertung für jedes Merkmal zu kommen, um dann anhand dieses Relevanzwertes eine Auswahl der geeigneten Merkmalsteilmenge zu treffen. Bei den sogenannten *Wrapper-Verfahren* hingegen werden Klassifikatoren eingesetzt, um den konkreten Nutzen einer ganzen Merkmalsteilmenge zu bewerten.

Die folgenden Abschnitte beschreiben diese beiden Ansätze im Detail mit Verweisen auf konkrete Verfahren, und vergleichen schließlich die Vor- und Nachteile beider Ansätze in Abschnitt 3.2.4.

3.2.2. Filter-Verfahren

Filter-Verfahren basieren im Allgemeinen auf der Idee der *Information* oder des *Informationsgehalts (engl. information content)*, wie sie erstmals in [Shannon, 1948] formalisiert wurde. Der Grundgedanke von Filter-Verfahren ist es dabei, jedem Merkmal einen Informationsgehalt zuzuschreiben, der umso größer ist, je mehr das Merkmal zur Lösung der betrachteten Problemstellung (Klassifikation, Regression, ...) beitragen kann.

Der Ablauf der Verwendung ist in Abb. 3.4 gezeigt. Der Filter erhält Daten von allen Merkmalen und führt eine Bewertung der Merkmale durch, deren Details sich zwischen den verschiedenen Filter-Varianten unterscheiden. Anschließend werden die Merkmale mit den höchsten Bewertungen gewählt, entweder bis eine gewählte Anzahl von Merkmalen gewählt ist, oder bis alle Merkmale, deren Bewertung über einem gewählten Schwellwert liegt, ausgewählt wurden.

Abb. 3.4.: Schematische Darstellung des Filter-Ansatzes zur Merkmalsauswahl.

Verschiedene Algorithmen, die diesem Ansatz folgen, unterscheiden sich vor allem in der Berechnung der Bewertung der Merkmale. Grundsätzlich resultieren Probleme daraus, dass der Informationsgehalt nur empirisch anhand der vorhandenen Trainingsdaten bestimmt werden kann, und es darüber hinaus noch Abhängigkeiten zwischen verschiedenen Merkmale geben kann, die dazu führen, dass beispielsweise die Kombination aus zwei scheinbar sehr relevanten Merkmalen nicht nennenswert mehr Information enthält als nur eines der beiden Merkmale alleine.

Definitionen

Unabhängig von einer exakten numerischen Bewertung können Merkmale bezüglich ihrer Relevanz kategorisiert werden. Meist wird zwischen *stark relevanten* und *schwach relevanten* Merkmalen unterschieden, die folgenden Definitionen orientieren sich an [John et al., 1994].

Definition 3.1 (Starke Relevanz)**.** Sei X_i ein Merkmal und S_i die Menge der Merkmale ohne X_i, und s_i eine Wertzuweisung zu allen Merkmalen S_i, Y eine Aussage (z.B. Klassifikation). Dann ist X_i *stark relevant* genau dann wenn es x_i, y, s_i gibt mit $p(X_i = x_i, S_i = s_i) > 0$, so dass

$$p(Y = y|S_i = s_i, X_i = x_i) \neq p(Y = y|S_i = s_i)$$

Die in Def. 3.1 definierte *starke Relevanz* besagt also, dass ein Merkmal X_i die Wahrscheinlichkeitsverteilung für die in Frage stehende Prädiktion beeinflusst, abhängig davon ob man es betrachtet oder nicht.

Definition 3.2 (schwache Relevanz)**.** Ein Merkmal X_i ist *schwach relevant* genau dann wenn es nicht stark relevant ist und es eine Menge von Merkmalen $S'_i \subset S_i$ gibt, für die es $x_{i,}, y, s'_i$ gibt mit $p(X_i = x_i, S'_i = s'_i) > 0$, so dass

$$p(Y = y|S'_i = s'_i, X_i = x_i) \neq p(Y = y|S'_i = s'_i)$$

Die in Def. 3.2 definierte *schwache Relevanz* besagt also, dass es für ein Merkmal X_i eine Teilmenge der übrigen Merkmale gibt, die bei Verwendung für die Prädiktion nicht die gleiche Verteilung liefert wie bei der zusätzlichen Verwendung von X_i.

Definition 3.3 (Relevanz)**.** Ein Merkmal ist *relevant*, wenn es entweder stark oder schwach relevant ist, ansonsten ist es *irrelevant*.

Die Definitionen 3.1-3.3 bilden die Grundlage für Filter-basierte Merkmalsauswahlverfahren, obwohl sie kein direktes Berechnungsverfahren an die Hand geben, das für eine Bewertung von Merkmale genutzt werden kann. Auch ziehen diese Definitionen nicht in Betracht, dass einzelne Merkmale nur in Kombination mit bestimmten anderen Merkmalen eine Information beitragen können. Verschiedene konkrete Verfahren (s.u.) bieten hier Lösungsansätze.

Verfahren

Es existieren verschiedene Algorithmen, die den allgemeinen Filter-Ansatz realisieren, und sich meist hauptsächlich in der für die Berechnung der Bewertung verwendeten Formel unterscheiden. Im Folgenden werden vier repräsentative Verfahren vorgestellt, für eine Übersicht über andere Verfahren und weitere Details sei auf [Liu and Motoda, 2008] verwiesen.

Relief & Relief-F Der in [Kira and Rendell, 1992] eingeführte und in [Kononenko, 1994] erweiterte *Relief*- bzw. *Relief-F*-Algorithmus ist ein *Monte Carlo*-Algorithmus, der auf die Bestimmung aller stark relevanten und schwach relevanten Merkmale abzielt. Die Grundidee des Algorithmus lässt sich dabei wie folgt zusammenfassen: Die Bewertung der Merkmale erfolgt durch eine Gewichtung aller Merkmale, die iterativ verfeinert wird durch die Betrachtung des Abstandes zufällig gewählter Trainingsinstanzen zur nächsten Instanz der gleichen Klasse (bezeichnet als *nearest hit*) und zur nächsten Instanz der anderen Klasse (bezeichnet als *nearest miss*). Der genaue Ablauf des Grundalgorithmus (für den 2 Klassen-Fall) ist in Alg. 3.3 dargestellt, und verwendet die in Gl. 3.7 definierte, auf einzelne Attribute beschränkte Abstandsfunktion. Die Auswahl einer Teilmenge relevanter Merkmale erfolgt dann durch Anwendung eines Schwellwertes auf die Gewichte der Attribute.

$$\text{diff}(A,\mathbf{x},\mathbf{y}) := \begin{cases} x_A - y_A & \text{für kontinuierliches Attribut } A \\ \delta_{x_A y_A} & \text{für nominales Attribut } A \quad \text{(mit Kronecker-Delta } \delta_{ij}) \end{cases} \qquad [3.7]$$

Variationen des Relief-Algorithmus ergeben sich durch die Verwendung alternativer diff-Funktionen, die die Behandlung von Instanzen mit fehlenden Attributwerten oder den Einsatz für Multiklassenprobleme erlauben.

Algorithmus 3.3 *Relief*-Algorithmus nach [Kononenko, 1994].

Eingabe: Trainingsdaten **D** mit $|A|$ Attributen, Anzahl der Durchläufe M

Ausgabe: Gewichte aller Attribute als Vektor **W**

1: Initialisiere **W** mit 0
2: **for** $1 \ldots M$ **do**
3: Wähle zufällige Instanz i
4: $h \leftarrow$ nächster Hit zu i
5: $m \leftarrow$ nächster Miss zu i
6: **for** $a = 1, \ldots, |A|$ **do**
7: $\mathbf{W}[a] \leftarrow \mathbf{W}[a] - \frac{\text{diff}(a,i,h)}{M} + \frac{\text{diff}(a,i,m)}{M}$
8: **end for**
9: **end for**

Correlation-Based Feature Subset Selection (CBFSS) Dieses in [Hall, 2000] beschriebene Verfahren bewertet nicht nur den Informationsgehalt einzelner Merkmale, sondern zieht auch die Korrelationen zwischen den Merkmalen in einer Merkmalsteilmenge in Betracht. Ziel des Verfahrens ist das Finden einer günstigen Merkmalsmenge mittels einer heuristischen Suche. Als zentrales Element dient die als Heuristik genutzte Funktion Merit, die eine Merkmalsteilmenge **T** bewertet. Die Funktion ist in Gl. 3.8 definiert, wobei n die Anzahl der Merkmale in Teilmenge **T** repräsentiert, $\overline{r_{KM}}$ die durchschnittliche Merkmal-Klassen-Korrelation und $\overline{r_{MM}}$ die durchschnittliche Merkmals-Merkmals-Korrelation.

$$\text{Merit}(\mathbf{T}) = \frac{n\overline{r_{KM}}}{\sqrt{n + n(n-1)\overline{r_{MM}}}} \qquad [3.8]$$

Für die Berechnung der Korrelation von zwei Merkmalen bzw. Merkmal und Klasse (über die gemittelt wird zur Berechnung von $\overline{r_{KM}}$ bzw. $\overline{r_{MM}}$) können verschiedene Maße eingesetzt werden, beispielhaft wird hier die *symmetrische Unsicherheit* (engl.*symmetrical uncertainty*) *SU* (definiert in Gl. 3.9) genutzt, die auf den Definitionen von *Informationsgewinn* (engl. *information gain*) *IG* bzw. *Entropie* (engl. *entropy*) *H* aufbaut, siehe Gl. 3.10-3.12.

$$SU(X,Y) = 2 \cdot \frac{IG(X|Y)}{H(X) + H(Y)} \qquad [3.9]$$

$$\begin{aligned} IG(X|Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \qquad [3.10]$$

$$H(Y) = -\sum_{y \in Y} p(y) \log_2(p(y)) \quad [3.11]$$

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2(p(y|x)) \quad [3.12]$$

Im Verbund mit der Heuristik können verschiedene Verfahren zur Suche im Raum aller Merkmalsteilmengen genutzt werden, in [Hall, 2000] werden insbesondere drei Suchstrategien diskutiert: Vorwärtswahl (engl. *forward selection*) (Greedy-Ansatz, ausgehend von einer leeren Startmenge werden schrittweise Merkmale hinzugefügt, die jeweils bestbewertete Menge dient als Ausgangsmenge für den nächsten Schritt), Rückwärtseliminierung (engl. *backward elimination*) (Greedy-Ansatz, ausgehend von der vollen Merkmalsmenge werden schrittweise Merkmale entfernt, solange sich die Bewertung nicht verschlechtert), Bestensuche (engl. *best first search*) (entweder von einer leeren oder der vollen Startmenge beginnend Expandierung der jeweils bestbewerteten Teilmenge und Bewertung der so entstandenen Teilmengen).

Mutual Information Feature Selection (MIFS) Ein auf der *Transinformation* (engl. *mutual information*) *MI* als Bewertungsmaß beruhendes Verfahren zur Merkmalsauswahl wird in [Benoudjit et al., 2004][Peng et al., 2005][Rossi et al., 2006][François et al., 2007] eingeführt, die Notation im Folgenden orientiert sich an der Darstellung in [Verleysen et al., 2009]. Gl. 3.13 definiert die Transinformation, zur Berechnung in realen Anwendungen werden meist leichter berechenbare Schätzverfahren eingesetzt (siehe [Kraskov et al., 2004]).

$$\begin{aligned} MI(X,Y) &= H(X) + H(Y) - H(X,Y) \\ &= \iint \mu_{X,Y}(x,y) \log \frac{\mu_{X,Y}(x,y)}{\mu_X(x)\mu_Y(y)} \, \mathrm{d}x\,\mathrm{y} \end{aligned} \quad [3.13]$$

Um die bei größeren Merkmalsmengen nicht zu bewältigende Bewertung aller möglichen Teilmengen zu umgehen, werden heuristische Verfahren zur Auswahl der jeweils nächsten zu bewertenden Merkmalsteilmengen eingesetzt, analog zu den im vorherigen Abschnitt dargestellten. Die oben angegebenen Publikationen greifen dabei alle auf Vorwärtswahl-ähnliche Greedy-Algorithmen zurück.

Fast Correlation-Based Filter (FCBF) Das in [Yu and Liu, 2003] und [Yu and Liu, 2004] eingeführte Verfahren wird in Abschnitt 6.3.2 ausführlich beschrieben, daher wird im Folgenden nur die Idee des Verfahrens skizziert.

Als relevant gelten im FCBF Merkmale, die *prädominant* für die Vorhersage des Klassenkonzeptes sind. Als prädominant wird ein Merkmal M bezeichnet, das gemessen mittels der symmetrischen Unsicherheit (aus Gl. 3.9) stärker mit der Klasse korreliert ist als ein vorbestimmter Schwellwert θ_{SU} und stärker als alle Korrelationen zwischen M und irgendeinem anderen Merkmal (d.h. M sagt die Klasse besser vorher als M von einem anderen Merkmal vorhergesagt werden kann). Um darauf aufbauend die relevanten Merkmale auszuwählen, werden alle Merkmale mit einer Mindest-Korrelation zur Klasse in einer geordneten Liste gesammelt und fortlaufend alle *redundant gleichgestellten* Merkmale (Merkmale, deren Korrelation zur Klasse über dem Schwellwert liegt, die aber eine noch größere Korrelation zueinander aufweisen, werden als redundant-gleichgestellt bezeichnet) entfernt. Die übrigen Merkmale stellen dann die relevanten Merkmale dar. Dieser Ablauf ist in Alg. 3.4 noch einmal zusammengefasst.

Algorithmus 3.4 Überblick Fast Correlation-based Filter-Algorithmus zur automatischen Auswahl eines relevanten Merkmalsteilmenge nach [Yu and Liu, 2003].

Eingabe: Trainingsdaten **D**, Schwellwert θ_{SU}

Ausgabe: Teilmenge $\mathcal{M}_{rel}$ relevanter Merkmale

1: Berechne absteigend geordnete Liste L von Merkmalen deren Korrelation zur Klasse einen Mindestwert aufweist ($SU(M,C) > \theta_{SU}$).
2: **repeat**
3: Wähle von Anfang der Liste beginnend das nächste noch nicht betrachtete Merkmal M.
4: Entferne alle Merkmale im Restteil der Liste, denen gegenüber M prädominant ist.
5: **until** Ende von L erreicht
6: Gib verbleibende Liste L als $\mathcal{M}_{rel}$ zurück

Bewertung

Filter-basierte Ansätze zur Merkmalsauswahl haben Vorteile im Bereich der Effizienz und der Schnelligkeit der Auswahl, und sind dadurch auch für große Datensätze anwendbar, bei denen Wrapper-basierte Verfahren nicht mehr sinnvoll einsetzbar sind. [Guyon and Elisseeff, 2003] berichten außerdem über eine große Robustheit gegenüber Overfitting, sodass Filter-Algorithmen auch für kleine Trainingsdatensätze gut geeignet sind.

Dagegen stehen Nachteile bei der Verwendung auf Daten mit unterschiedlichen Wertedomänen, beispielsweise ist oft eine spezielle Behandlung von diskreten gegenüber kontinuierlichen Daten notwendig. Auch die Erkennung von redundanten Merkmalen ist ohne weiteres, d.h. bei der Verwendung einfacher Informationsmaße, nur in geringem Maß oder überhaupt nicht

möglich (beispielsweise beim Relief-Algorithmus). So kann es Situationen geben, in denen besonders nützliche Merkmale von Filtern als irrelevant aussortiert werden.

3.2.3. Wrapper-Verfahren

Wrapper-Verfahren basieren auf der Idee, den später verwendeten Klassifikator direkt zur Auswahl relevanter Merkmale einzusetzen. Der Vorteil gegenüber Filter-Ansätzen liegt vor allem in der erwünschten, impliziten Beachtung der Eigenheiten des Klassifikators. Während Filter-Ansätze in gewissen Situationen Merkmale auswählen können, die zwar sehr relevant sind, aber durch die Eigenschaften des verwendeten Klassifikators (z.B. nur lineare Trennung) nicht genutzt werden können, dient hier der verwendete Klassifikator selbst als Kriterium für die Auswahl.

Als Bewertung einer Merkmalsmenge dient die Qualität der Ergebnisse, die mit dieser Menge bei der Verwendung zum Training eines Prädiktors erreicht werden können. Die Auswahl arbeitet dann iterativ, indem eine oder mehrere zu evaluierende Merkmalsteilmengen generiert werden, anschließend je ein Prädiktor mit jeder dieser Merkmalsmengen trainiert wird, und die Leistung des Prädiktors auf einer Testmenge evaluiert wird. Diese Leistung wird zur Bewertung der Merkmalsteilmengen genutzt, anschließend werden je nach Qualität der erreichten Ergebnisse entweder neue Merkmalsmengen generiert, oder die beste Merkmalsteilmenge wird als Ergebnis ausgegeben. Abb. 3.5 zeigt diesen Ablauf im Überblick.

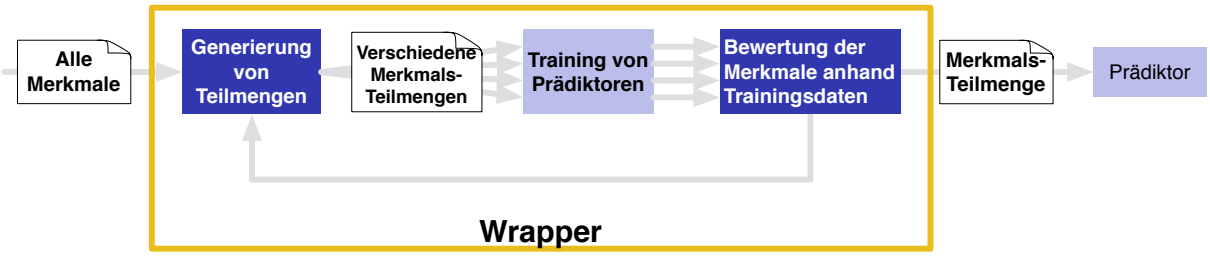


Abb. 3.5.: Schematische Darstellung des Wrapper-Ansatzes zur Merkmalsauswahl.

Durch den allgemeinen Aufbau des Verfahrens können prinzipiell beliebige Klassifikatoren mit dem Wrapper-Ansatz kombiniert werden. Einige Beispiele eingesetzter Klassifikatoren sind Naive Bayes-Klassifikatoren bei [Kohavi and John, 1997], Entscheidungsbäume bei [Kohavi and John, 1997][Blum and Langley, 1997], Bayes-Netze bei [Inza et al., 2000] sowie SVMs bei [Weston et al., 2001].

Ein wichtiger Aspekt ist die Zusammenstellung der Merkmalsteilmengen, die evaluiert werden sollen. Neben Brute Force-Verfahren (nur geeignet bei sehr kleinen Merkmalsmengen)

kommen hier häufig verschiedene Heuristiken zum Einsatz, beispielsweise Vorwärtssuchen, Rückwärtseliminierung, Genetische Algorithmen und andere.

Der Wrapper-Ansatz hat den großen Vorteil, speziell für das verwendete Lernverfahren geeignete Merkmale zu finden, woraus in vielen Fällen eine bessere Performanz folgt. Wrapper bewerten immer die gesamte jeweils bewertete Merkmalskombination, dadurch ist die Wahrscheinlichkeit, nützliche Merkmale zu übersehen, sehr gering (wenn die eingesetzte Heuristik zur Auswahl der zu bewertenden Merkmale gut genug den Raum aller Merkmalsteilmengen exploriert). Durch den Einsatz geeigneter Klassifikatoren stellt auch die Verwendung für diskrete und kontinuierliche Merkmale kein Problem dar. Allerdings gibt es auch Nachteile, von denen zuvorderst die oft sehr lange Laufzeit genannt werden muss, da jede Bewertung einer Merkmalsmenge das vollständige Trainieren und Evaluieren erfordert. Auch besteht abhängig vom verwendeten Klassifikator die Gefahr, dass es bei kleinen Trainingsdatensätzen zu einem Overfitting des Klassifikators und der gut bewerteten Merkmalsmenge kommt.

Nicht unerwähnt bleiben soll schließlich die Möglichkeit, die Merkmalsauswahl integriert in einem Klassifikator durchzuführen (engl. *embedded feature selection*). Ein Beispiel für einen Klassifikator, der dazu in der Lage ist, sind *Support Vector Machines (SVMs)*. Durch den Einsatz von Kernels führen SVMs implizit eine Transformation der Eingaben durch, die als Konstruktion bzw. Auswahl geeigneter Merkmale interpretiert werden kann.

3.2.4. Vergleich und Bewertung von Filtern und Wrappern

Filter- und Wrapper-Ansätze sind komplementär zueinander bezüglich ihrer Vorteile und Nachteile. Filter sind im Allgemeinen schneller, da eine direkte Bewertung der Merkmale ohne Trainieren und Testen eines Klassifikators erfolgt. Zudem ist die Verwendung oder das Testen unterschiedlicher Klassifikatoren sehr einfach, da die Merkmale nach allgemeinen Relevanz-Gesichtspunkten gewählt wurden. Andererseits können durch die Nichtbeachtung des Zielklassifikators Merkmale gewählt werden, die im realen Einsatz keine guten Ergebnisse zeigen.

Umgekehrt sind Wrapper zwar langsamer, dafür sind aber die Ergebnisse durch die Abstimmung auf den eingesetzten Klassifikator häufig besser. Allerdings besteht bei Wrappern eine stärkere Gefahr des Overfitting, da die Merkmalsauswahl sehr stark von den Ergebnissen des Klassifikators auf den Trainings- und Testdaten abhängt. Dadurch sind als Ergebnis Merkmalsmengen möglich, die speziell auf diese Datensätze hin optimiert sind.

3.3. Klassifikatoren

Ein Klassifikator ist ein mathematisches Modell, das die Zuordnung eines durch *Merkmale* beschriebenen Objekts zu einer oder mehrerer Klassen durchführt. Im entwickelten System zur Erkennung von Aktivitäten werden an verschiedenen Stellen unterschiedliche Anforderungen an die benötigten Mustererkennungsysteme gestellt. Die folgenden Abschnitte fassen die notwendigen Informationen zu den eingesetzten Klassifikatoren zusammen.

3.3.1. Support Vector Machine

Eine *Support Vector Machine (SVM)* (die Übersetzung „Stützvektormaschine“ existiert, ist aber nicht gebräuchlich) ist ein Verfahren, das eine trennende Hyperebene in einer Menge von Trainingsdaten findet. Eingeführt wurde es in der heute üblichen Form 1995 von Vapnik und Cortes [Cortes and Vapnik, 1995]. Die Hyperebene wird definiert durch die sogenannten *Stützvektoren* (engl. *Support Vectors*), die dem Verfahren auch seinen Namen geben. Die Trennhyperebene wird gemäß Gl. 3.14 durch den Normalenvektor $\mathbf{w}$ und den sogenannten *Bias* b repräsentiert, die durch die Stützvektoren $\mathbf{x}_i$ mit Gl. 3.15 bestimmt sind. Stützvektoren sind diejenigen Lernbeispiele, die am nächsten an der Trennhyperebene (und damit an der anderen Klasse) liegen.

$$\mathbf{w} \cdot \mathbf{x} - b = 0 \qquad [3.14]$$

$$\forall \mathbf{x}_i : \ |\mathbf{w} \cdot \mathbf{x}_i - b| = 1 \qquad [3.15]$$

Zur Lösung wird das Problem mittels Lagrange-Multiplikatoren dargestellt und in eine duale Form überführt (siehe Gl. 3.16 mit Bedingungen 3.17 und 3.18), die eine einfachere Lösung erlaubt. Auch eine verallgemeinerte Form des Problems, erweitert mit sogenannten *Schlupfvariablen* (engl. *slack variables*), ist auf diese Art lösbar. Damit können auch Trennhyperebenen für nicht linear trennbare Datensätze gefunden werden. Dieses Problem tritt beispielsweise durch fehlerhaft klassifizierte Lernbeispiele (resultierend aus Sensorrauschen oder menschlichen Fehlern beim Vorklassifizieren der Daten) auf.

$$\max_{\alpha} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \, \mathbf{x}_i \cdot \mathbf{x}_j \right) \qquad (\alpha_1, \ldots, \alpha_N \text{ Lagrange-Multiplikatoren}) \qquad [3.16]$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \qquad [3.17]$$

$$\alpha_i \geq 0 \qquad \forall i = 1, \ldots, m \qquad [3.18]$$

Zwei wichtige charakteristische Eigenschaften sind der in ihnen realisierte induktive Bias und die Möglichkeit, sogenannte *Kernel* einzusetzen, um auch nicht-lineare Problemstellungen behandeln zu können. Der induktive Bias liegt dabei in der Strategie zur Wahl der Trennhyperebene, die so gewählt wird, dass der Abstand zu beiden Klassen (engl. *margin*) maximiert wird, um dadurch eine optimale Generalisierung zu erreichen.

In der Praxis genügen aber auch in dieser Art optimal gewählte lineare Trennhyperebenen oft nicht zur Lösung gestellter Klassifikationsaufgaben, auch wenn Schlupfvariablen eingesetzt werden, da viele Probleme nicht linear trennbar sind. Für deutlich nicht-lineare Probleme genügt die Verallgemeinerung durch Schlupfvariablen nicht. Um auch solche Problemstellungen behandeln zu können, wurde der SVM-Ansatz erweitert. Die eingesetzte Lösungsstrategie ist es, die Daten zu transformieren, sodass das resultierende Problem linear trennbar ist (üblicherweise werden die Daten in geeigneter Weise in einen höherdimensionalen Raum projiziert). An dieser Stelle greift der sogenannte *Kernel-Trick*. Das einer SVM zugrundeliegende mathematische Optimierungsproblem lässt sich wie oben beschrieben in einer Form darstellen, in der die Trainingsdaten nur noch innerhalb von Skalarprodukten auftreten (siehe Gl. 3.16). Diese Form lässt sich dann erweitern, indem die Skalarprodukte durch positiv definite *Kernelfunktionen* ersetzt werden. Die Kernelfunktionen ermöglichen dann ein implizites Rechnen mit den Daten in einem (meist höherdimensionalen) anderen Raum, und damit die korrekte Trennung auch nicht-linear trennbarer Daten. Eine Kernelfunktion ist eine Abbildung $K : X \times X \rightarrow \mathsf{R}$ mit der in Gl. 3.19 gegebenen Eigenschaft. Dabei ist $\phi : X \rightarrow F$ eine Abbildung, die in einen Skalarvektorraum $(F, \langle \cdot, \cdot \rangle)$ abbildet.

$$K(x,y) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \quad \text{mit } \mathbf{x}, \mathbf{y} \in X \tag{3.19}$$

Einige gebräuchliche Kernel-Funktionen sind in Tab. 3.3.1 aufgelistet. Unterschiedliche Resultaten ergeben sich durch die implizit durchgeführte Transformation bei der Verwendung der verschiedenen Kernel. Beispielsweise realisiert der RBF-Kernel eine implizite Transformation in einen unendlichdimensionalen Hilbert-Raum. Darüberhinaus wurden in den letzten Jahren aber auch zahlreiche Kernelfunktionen für symbolische Daten wie Texte und Graphen definiert, die neue Anwendungsfelder für die SVM-Methodologie erschlossen haben.

Ausführliche Informationen zu SVMs, Kernel-basierten Methoden und viele Informationen zu verschiedenen Kerneln sind in den beiden Referenzwerken [Schölkopf and Smola, 2001][Shawe-Taylor and Cristianini, 2004] zu finden.

Tab. 3.1.: Typische Kernel-Funktionen, die häufig mit SVMs verwendet werden.

Lineare Kernel	$K(\mathbf{x},\mathbf{y}) = \langle \mathbf{x},\mathbf{y} \rangle$
Polynomielle Kernel	$K(\mathbf{x},\mathbf{y}) = \langle \mathbf{x},\mathbf{y} \rangle^d$
RBF-Kernel	$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$
Sigmoid-Kernel	$K(\mathbf{x},\mathbf{y}) = \tanh(\kappa(\mathbf{x}\cdot\mathbf{y}) + \theta)$

3.3.2. Hidden Markov Model

Die 1989 von Rabiner in [Rabiner, 1989] präsentierten *Hidden Markov Models (HMMs)* modellieren einen Markov-Prozess mit nicht beobachtbaren (*hidden*) Zuständen. Während in regulären Markov-Modellen der aktuelle Zustand direkt beobachtet werden kann, ist das bei HMMs nicht der Fall. Hier können nur Ausgaben der Zustände beobachtet werden. HMMs werden insbesondere zur Erkennung von zeitlichen Mustern eingesetzt, beispielsweise in der Spracherkennung und der Handschrifterkennung. Die folgende Darstellung und die verwendeten Schreibweisen orientieren sich an [Rabiner, 1989].

Ein HMM kann definiert werden als 5-Tupel (N, M, A, B, π) mit den folgenden Elementen, wobei für eine klarere Darstellung $\lambda = (A, B, \pi)$ die Modellparameter zusammenfasst:

N Anzahl der Zustände $S = \{S_1, \ldots, S_N\}$.

M Größe des Ausgabealphabet $V = \{v_1, \ldots, v_M\}$.

A Wahrscheinlichkeitsverteilung der Zustandsübergänge gegeben als $A = \{a_{ij}\}$ mit a_{ij} die Wahrscheinlichkeit des Übergangs von Zustand S_i in Zustand S_j.

B Ausgabewahrscheinlichkeiten gegeben als $B = \{b_i(k)\}$, wobei gilt:

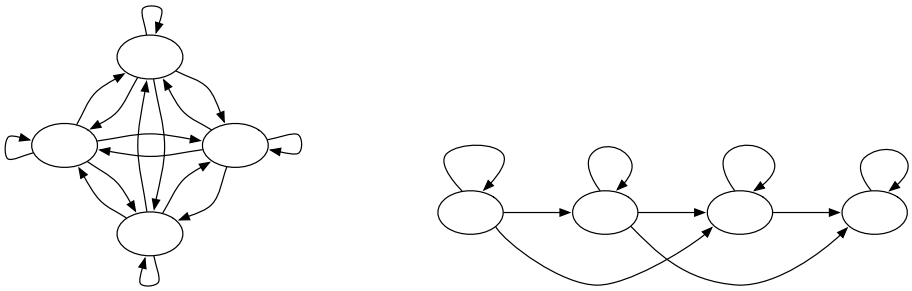
$$b_i(k) = P(\text{Ausgabe von } v_k \text{ zu Zeitpunkt } t | q_t = S_i)$$

π Initiale Zustandsverteilung gegeben als $\pi = \{\pi_i\}$ wobei gilt:

$$\pi_i = P(q_1 = S_i)$$

HMMs können als Graphen dargestellt werden, mit Knoten als Repräsentation von Zuständen und Kanten zwischen Zuständen, deren Übergangswahrscheinlichkeit größer 0 ist. Abb. 3.6 zeigt zwei Beispiele für die Darstellung von HMMs als Graphen anhand zweier typischer Ausprägungen von HMMs. In Abb. 3.6(a) ist ein *ergodisches* HMM dargestellt, bei dem von jedem

Zustand ein Übergang in jeden anderen Zustand möglich ist (d.h. $\forall i, j : a_{ij} > 0$), in Abb. 3.6(b) ist ein *Links-Rechts-HMM* dargestellt, bei dem Übergänge von einem Zustand nur in einen Zustand „weiter rechts“ möglich ist (d.h. $i > j \Rightarrow a_{ij} = 0$).



(a) (b)

Abb. 3.6.: Beispiele für die Darstellung von HMMs als Graphen anhand zweier unterschiedlicher HMM-Typen. (a) Ergodisches HMM. (b) Links-Rechts-HMM.

Eine Folge von Beobachtungen kann mittels eines gegebenen Modells erzeugt werden durch Wahl eines von π erlaubten Startzustandes q_1, und abwechselnde Wahl einer von B zugelassenen Ausgabe und eines von A erlaubten Zustandsübergangs.

Für HMMs werden drei Hauptprobleme betrachtet, die gelöst werden müssen, um diese Modelle in realen Anwendungen einsetzen zu können.

Problem 1: Berechnung der Erzeugungswahrscheinlichkeit $P(O|\lambda)$

> Eine effiziente Berechnung der Wahrscheinlichkeit $P(O|\lambda)$, dass eine gegebene Beobachtungsfolge $O = O_1 O_2 \ldots O_T$ von einem gegebenen HMM, definiert durch $\lambda = (A, B, \pi)$, erzeugt wurde.

Problem 2: Berechnung der wahrscheinlichsten Zustandsfolge Q

> Gegeben eine Folge von Ausgaben $O = O_1 O_2 \ldots O_T$, was ist in einem gegebenen HMM $\lambda = (A, B, \pi)$ die „beste“ Zustandsfolge $Q = q_1 q_2 \ldots q_T$ in dem Sinn, dass sie die Ausgabe mit der größten Wahrscheinlichkeit erzeugt hat.

Problem 3: Bestimmung von Modellparametern

> Gegeben eine oder mehrere Beobachtungsfolgen O, wie müssen die Modellparameter $\lambda = (A, B, \pi)$ angepasst werden, um $P(O|\lambda)$ zu maximieren.

Für die Lösung der drei Probleme sind effiziente Lösungsverfahren aus der Literatur bekannt, teilweise existieren verschiedene Alternativen zu Lösung. Zur Berechnung der Wahrscheinlichkeit, dass eine Beobachtung O von einem Modell λ erzeugt wurde (Problem 1), dient

der auf dynamischer Programmierung beruhende *Forward-Algorithmus*. Für die Berechnung der wahrscheinlichsten Zustandsfolge Q in einem Modell λ, die eine Beobachtungssequenz O erzeugt hat (Problem 2), dient der *Viterbi-Algorithmus*, der ebenfalls auf dynamischer Programmierung basiert. Zur Bestimmung geeigneter Modellparameter λ, um das Modell für die Erzeugung eines oder mehrerer gegebener Beobachtungsfolgen zu maximieren (Problem 3), wird der *Baum-Welch-Algorithmus* eingesetzt. Diese Variante eines *Expectation Maximization*-Algorithmus' nutzt den erwähnten Forward-Algorithmus und den umgekehrt arbeitenden *Backward-Algorithmus* zur Berechnung von benötigten Zwischenergebnissen. Alle diese Algorithmen sind in [Rabiner, 1989] ausführlich vorgestellt.

Die Verwendung mit mehrdimensionalen Eingabedaten erfordert eine Erweiterung der HMM-Modelle. Dazu wurden verschiedene Formalismen vorgeschlagen, die unterschiedlichen Kompromisse zwischen der Erweiterung der Modellmächtigkeit und der dadurch entstehenden erhöhten Komplexität eingehen. Wichtige derartige Erweiterungen sind *Coupled Hidden Markov Model (CHMM)* nach [Brand, 1996][Brand et al., 1997], *Independently Coupled Hidden Markov Model (ICHMM)* nach [Darmanjian et al., 2006] und *Linked Hidden Markov Model (LHMM)* nach [Saul and Jordan, 1995][Basu, 2003].

4. Ansatz zur automatischen Erkennung & Interpretation von Handlungen des Menschen

In diesem Kapitel wird die Problemstellung der vorliegenden Arbeit herausgearbeitet und formalisiert. Anschließend wird das Lösungskonzept vorgestellt, das entwickelt wurde, und dessen Details und Realisierung in den folgenden Kapiteln vorgestellt werden.

4.1. Problemstellung und Begriffsdefinitionen

Die vorliegende Arbeit beschäftigt sich mit dem Problem der Aktivitätserkennung, dessen Relevanz für die Robotik schon in der Einführung dargelegt wurde. Wie aus Kapitel 2, das den Stand der Forschung in diesem Feld analysiert, hervorgeht, gibt es keine allgemein akzeptierte Definition dieses Begriffes. Daher werden im Folgenden zunächst die wichtigsten Begrifflichkeiten eindeutig definiert, bevor mit ihrer Hilfe die Problemstellung formalisiert wird.

4.1.1. Anwendungsdomäne

Eine Erkennung von Aktivitäten kann in verschiedenem Rahmen und zu unterschiedlichen Zwecken erfolgen, wie die verschiedenen in Kapitel 2 beschriebenen Arbeiten zeigen. Das im Rahmen dieser Arbeit entwickelte Konzept zielt auf eine Verwendung im Robotik-Kontext in einem System zur Umsetzung des *Programmieren durch Vormachen*-Programmierparadigmas. Das betrachtete Einsatzgebiet, aus dem Aktivitäten für die spätere Evaluation gewählt wurden, ist ein Küchen-/Cafeteria-Szenario.

In solchen Szenarien sind einerseits allgemeine Gesten zur Kommandierung und Interaktion von Interesse (wie Zeigegesten, Winken, etc.), andererseits Szenario-spezifische Handlungen (wie Gegenstände nehmen und abstellen, etwas eingießen, trinken, etc.). Dabei ist abhängig von der verwendeten Sensorik und der betrachteten Abstraktionsebene im PdV-Prozess Betrachtung von Handlungen unterschiedlichen Detailgrades notwendig und möglich, beispielsweise eine Unterscheidung unterschiedlicher Griffe für detaillierte Lernprozesse, für das Lernen auf einer symbolischeren Ebene nur die Erkennung eines Griffes (ohne weitere Details).

4.1.2. Begriffsdefinitionen

Als Grundeinheit für die Erkennung von Aktivitäten dienen einzelne Bewegungen und Posen von Menschen. Die folgende Definition legt fest, was im Rahmen dieser Arbeit unter diesen umgangssprachlichen Begriffen verstanden wird:

Definition 4.1 (Bewegungen)**.** Als *Bewegungen* im engeren Sinn werden von außen beobachtbare Bewegungen und Posen von Menschen bezeichnet, d.h. Denkprozesse, Organbewegungen und ähnliche Vorgänge im Körper werden hiermit ausgeschlossen. Formal wird eine *Bewegung* als Zeitreihe (b_t) mit $b_t \in K^T, t \in T$ von beobachtbaren Modellparametern (K bezeichnet den Raum aller Modellparameter) definiert, die eine menschliche Pose oder Bewegung beschreiben.

Diese Definition stellt keine Bedingungen an die zu verwendenden Modellparameter oder ein bestimmtes Modell als Grundlage der Zeitreihe, solange die Pose eines Menschen damit beschrieben werden kann. Auch eine Zeitreihe von Geschwindigkeitsvektoren mit einer definierten Anfangspose des beschriebenen Körpers erfüllt diese Definition. Insbesondere kann diese Darstellung aus verschiedenen Beobachtungen gewonnen werden. Eine Aktivität kann sich in verschiedenen Bewegungen ausprägen:

Definition 4.2 (Aktivitäten)**.** Eine *menschliche Aktivität* $\mathcal{A}$ ist eine unscharfe Menge von Ausschnitten von Bewegungen im Sinne von Definition 4.1, d.h. es existiert eine Zugehörigkeitsfunktion $\mu_{\mathcal{A}} : (b_t) \rightarrow [0,1]$, die endlichen Bewegungen $(b_t)_{t\in\{t_1,\ldots,t_2\}}$ eine Zugehörigkeit zur Aktivität zuordnet, abhängig davon wie charakteristisch die Bewegung für die Aktivität ist. Die Elemente von $\mathcal{A}$ müssen nicht alle die gleiche Länge haben, d.h. t_1, t_2 sind nicht fix für alle betrachteten Bewegungen.

Diese Definition fordert nicht, dass verschiedene Aktivitäten disjunkt sind. Es kann Bewegungen oder Posen geben, die für verschiedene Aktivitäten charakteristisch sind. In solchen Fällen kann eine Unterscheidung durch zusätzliche Nutzung von *Kontext* und *Hintergrundwissen* erfolgen, oder unmöglich sein. Die folgende Definition ist angelehnt an die Kontext-Definition in [Dey, 2001]:

Definition 4.3 (Kontext)**.** Der *Kontext einer Aktivität* umfasst alle Informationen, die zur Charakterisierung von Situationen und Entitäten, die für die Aktivität relevant sind, genutzt werden können. Eine Entität kann hierbei eine Person, ein Ort oder ein Objekt sein, einschließlich der Person, die in die Aktivität involviert ist.

Beispiele für Information, die der Kontext umfasst, sind der Ort, an dem eine Aktivität durchgeführt wird, Art der Objekte, die manipuliert werden, Alter und Geschlecht der agierenden

Person. Kontext ist nicht immer durch Sensoren erfassbar, beispielsweise das Alter einer Person. Solche Information muss als Hintergrundwissen zur Verfügung gestellt werden, wenn sie genutzt werden soll.

Definition 4.4 (Hintergrundwissen)**.** Als *Hintergrundwissen* für die Aktivitätserkennung werden Informationen bezeichnet, die nicht aus den Eingabedaten des Systems abgeleitet werden können, sondern die aus einer persistenten Wissensbasis oder interaktiv vom Benutzer abgefragt werden.

Definition 4.5 (Aktivitätserkennung)**.** Als *Aktivitätserkennung* wird im weiteren Verlauf dieser Arbeit die Zuordnung von Beobachtungen zu einer oder mehreren Aktivitäten bezeichnet. Im Sinne Definitionen 4.1–4.4 wird damit eine Abbildung aus dem Raum aller Bewegungszeitreihen K^T auf die Zugehörigkeitswahrscheinlichkeit zur betrachteten Aktivität realisiert:

$$\begin{aligned} K^T &\rightarrow [0,1] \\ b_t &\mapsto P_{\mathcal{A}}(b_t) := P(\text{passend gewähltes Suffix von } b_t \text{ gehört zu Aktivität } \mathcal{A}) \end{aligned}$$

Abhängig von der Realisierung dieser Abbildung kann das Ergebnis unterschiedlich gut die ideale Abbildung annähern. Allerdings sind auch Menschen nicht unbedingt perfekt hinsichtlich dieser Aufgabe, wie sich beispielsweise beim widersprüchlichen Labeln von Trainingsdaten durch verschiedene Personen zeigt.

Erkannte Aktivitäten dienen als Eingabeinformationen für darauf aufbauende Systeme, die auf einer abstrakteren Ebene diese Informationen nutzen. Unterschiedliche weiterverwendende Systeme erfordern auch die Erkennung unterschiedlicher Arten von Aktivitäten. Zum Einen sollen Aktivitäten gelernt werden können, die als Eingabe für das Lernen von Handlungswissen mittels des Programmieren-durch-Vormachen-Paradigmas dienen können. Zum Zweiten ist Verständnis für menschliche Aktivitäten eine wichtige, unterstützende Information für die Interaktion zwischen Mensch und Roboter. Und schließlich ist Wissen über menschliche Aktivitäten notwendig, um autonomes, proaktives Handeln von Robotern zu ermöglichen.

4.1.3. Formalisierung

Die Problemstellung der Arbeit besteht in der Entwicklung eines Systems zur Aktivitätserkennung im Sinne von Definition 4.5, das eine Verwendung in drei Anwendungsbereichen in der Robotik erlaubt:

- Eingabe für das Lernen von Handlungswissen mittels Programmieren-durch-Vormachen
- Unterstützende Information für Mensch-Roboter-Interaktion

- Zusätzlicher Eingabekanal für autonomes Entscheiden

Die Aufgabenstellung erfordert also eine Lösung, die eine Bandbreite unterschiedlicher Anwendungen unterstützt. Das bedingt einen Satz von Eigenschaften, die bei einer Lösung vorhanden sein müssen, damit sie auch wirklich den angestrebten Zweck erfüllen kann. Die sich ergebenden Eigenschaften sind (in alphabetischer Reihenfolge):

Eigenschaften gültiger Lösungen

Erweiterbarkeit Als Erweiterbarkeit wird die Eigenschaft des System bezeichnet, die es erlaubt, auf einfache und weitgehend auf manuelle Eingaben verzichtende Art und Weise die Erkennung neuer Aktivitäten zu trainieren.

Einfach im hier geforderten Sinn bedeutet, dass auch ein technisch versierter, aber nicht speziell mit dem System vertraute Benutzer selbständig die Erkennung neuer Aktivitäten eintrainieren kann in einem zeitlichen Rahmen der in der Größenordnung weniger Stunden liegt.

Kombinierbarkeit Erkenner für einzelne Aktivitäten sollen gemeinsam und in nicht explizit eintrainierten Kombinationen verwendet werden können, da zum Lernzeitpunkt nicht alle möglichen Verwendungszwecke schon bekannt sind.

Kombinierbar im hier geforderten Sinn bedeutet, dass zwei gegebene Erkenner E_1 und E_2, die beide mit der verfügbaren Sensorik einzeln genutzt werden können, auch gemeinsam eingesetzt werden können, und konsistente Resultate liefern.

Sensorabstraktion Die Erkennung soll soweit als möglich von der als Datenquelle dienenden Sensorik abstrahiert werden, um die in der Praxis häufig vorkommenden Änderungen an der Systemkonfiguration (Kamera-Austausch, Änderung des Blickwinkels, ...) von der Erkennung zu trennen und transparent durchführen zu können.

Die *Sensorabstraktion* im hier geforderten Sinn erfordert die Verwendung geeigneter Referenzmodelle, auf die die konkreten Sensordaten abstrahiert und abgebildet werden können.

Skalierbarkeit Das System muss in der Lage sein, sowohl Aktivitäten des ganzen Körpers als auch feiner aufgelöste Aktivitäten einzelner Körperteile erkennen zu können, wenn die verfügbaren Datenquellen die Beobachtung entsprechend benötigter feiner Details erlauben.

Skalierbarkeit im hier geforderten Sinn bedeutet, dass das System keine impliziten Annahmen über die Art der zu erkennenden Aktivitäten trifft, sondern für die Erkennung auch deutlich unterschiedlicher Aktivitäten genutzt werden kann, wenn die Perzeption im dafür benötigten Detailgrad verfügbar ist.

Übertragbarkeit Das erlernte Wissen über die Erkennung von Aktivitäten soll übertragbar sein zwischen verschiedenen Trägerplattformen, das bedeutet eine auf Roboter *A* eingelernte Aktivität soll auch auf Roboter *B* erkennbar sein durch eine einfache Übertragung des Erkenners, ohne dass ein Neulernen der Aktivität nötig ist.

Übertragbarkeit im hier geforderten Sinn stellt keine Anforderungen an den Vorgang der Übertragung zwischen den Systemen *A* und *B*. Das was übertragen wird ist darüberhinaus nicht die Fähigkeit, die erkannte Aktion auszuführen (im Sinne eines *Imitation Learning*-Ansatzes), sondern nur die Übertragung der Fähigkeit des Erkennens der betreffenden Aktivität.

Diese Eigenschaften sind nicht vollständig unabhängig, so unterstützt die Eigenschaft der Sensorabstraktion auch die mögliche Übertragbarkeit des Gelernten zwischen verschiedenen Systemen. Trotzdem sind diese beiden Eigenschaften nicht äquivalent die eine subsumiert von der anderen, daher werden beide gefordert.

Anforderungen an gültige Lösungen

Über die oben geforderten Eigenschaften hinaus, die eine Lösung der Aufgabenstellung haben muss, müssen Lösungen aus der geplanten Verwendung folgenden Randbedingungen genügen, die messbare Eigenschaften der Lösung quantifizieren:

Laufzeit Ganz besonders die Verwendung in der Mensch-Roboter-Interaktion, aber auch das proaktive Entscheiden benötigt eine gewisse Mindestgeschwindigkeit der Erkennung. Sie erfordern zwar keine harte Echtzeitfähigkeit, da solche Systeme im Allgemeinen mit dem Fehlen vereinzelter Erkennungsergebnisse gut umgehen können. Für ein reaktives System ist – aus Erfahrungswerten – dennoch eine Geschwindigkeit von mindestens 15 Hz nötig.

Format Als Ergebnisse sind einfache boolesche Variablen (WINKEN: wahr oder falsch) nicht ausreichend für die auf den Ergebnissen aufbauenden Systemen. Stattdessen wird für jede Aktivität ein Plausibilität im Intervall $[0, 1]$ erwartet, um in Systemen, die unter Beachtung von Unschärfe arbeiten, integriert werden zu können.

4.2. Konzept

Das Konzept zur Lösung der Problemstellung baut auf einer Prozesskette aus, wie sie allgemein für Erkennungsaufgaben (Klassifikation, Regression) als abstrakte Architektur genutzt wird (Abbildung 4.1). In der *Datenakquisition* werden die Daten aufgezeichnet, anhand derer eine Erkennungsaufgabe gelöst werden soll. In der Merkmalsextraktion werden aus den Rohdaten Merkmale extrahiert, die ein geeignetes Format haben, um die verwendeten Erkenner einsetzen zu können. Da aus den Rohdaten meist sehr viele Merkmale extrahiert werden können, auch solche, die unabhängig von Erkennungsaufgabe sind, werden in der *Merkmalsauswahl* die für die Aufgabe relevanten Merkmale ausgewählt (mit dem Ziel eines robusteren Systems). Die gewählten Merkmale werden dann in der *Klassifikation* zum Training bzw. zur Erkennung genutzt.

Um den speziellen Randbedingungen definiert in 4.1.3 zu genügen, wurde eine Erweiterung der Standard-Architektur entworfen, die diese Architektur um die Verwendung zusätzlicher Informationen und Verarbeitungsmöglichkeiten ergänzt, wie in Abbildung 4.2 gezeigt. Die Prozesskette wird um die Möglichkeit zur Einbindung von Hintergrundwissen erweitert, das in den einzelnen Komponenten der Prozesskette genutzt wird, um besser an die spezifische Aufgabenstellung angepasste Ergebnisse zu erzielen. Zusätzlich wird eine Rückkoppelung der Erkennungsresultate des Systems in die Merkmalsextraktion und in die Merkmalsauswahl ermöglicht.

Die Datenakquisition ist die Komponente, die auch unabhängig von der eigentlichen Aktivitätserkennung genutzt werden kann. Umgekehrt soll es auch auch möglich sein, eine andere als die im Folgenden verwendeten Datenakquisitions-Komponenten mit dem Rest des Systems einzusetzen (Eigenschaft der Sensorabstraktion). Im Rahmen dieser Arbeit wurden aber auch Untersuchungen am Beispiel des Trackingsystems *VooDoo* vorgenommen zur Möglichkeit, die Datenakquisition durch die Verwendung von Hintergrundwissen zu verbessern. Als Hintergrundwissen werden hier anatomische Informationen über den menschlichen Körper eingesetzt, um die eine initiale Modellerkennung und ein genaueres Tracking durch das Einhalten von Bewegungsgrenzen des menschlichen Körpers zu ermöglichen.

Die Nutzung von Merkmalen dient auch im erweiterten Konzept zur Abstraktion von den in der Datenakquisition verwendeten realen Sensoren auf abstrakte, allgemeiner verwendbare Modelle. Die Merkmalsextraktion nutzt dabei zusätzliches Hintergrundwissen, um Domänenspezifisch günstige Merkmale zu definieren, die aus den verfügbaren Daten extrahiert werden können.

Die Merkmalsauswahl erfüllt den oben schon genannten Zweck, die Erkennung auf eine kleinere Auswahl von relevanten Merkmalen zu beschränken, um den Einfluss von Rauschen

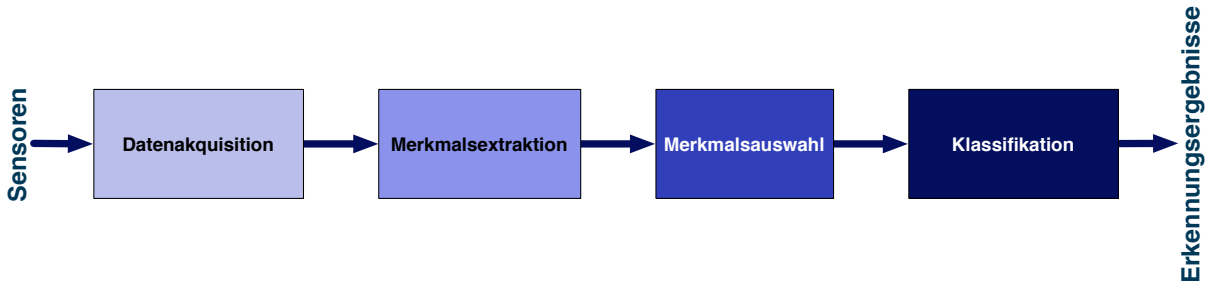


Abb. 4.1.: Typische Architektur der Prozesskette eines Erkennungssystems.

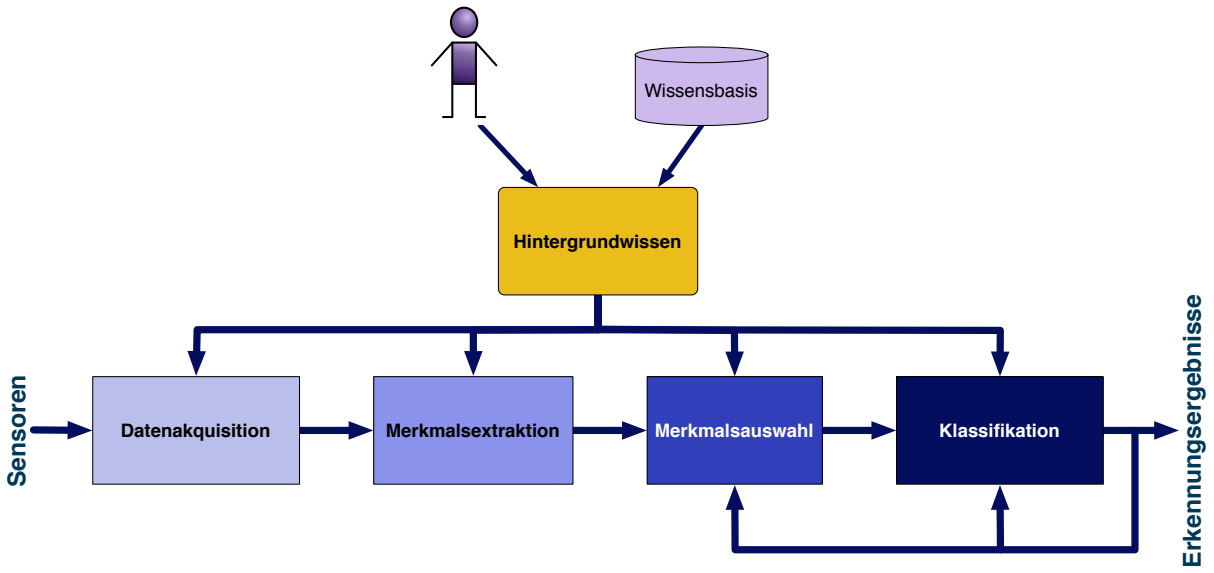


Abb. 4.2.: Referenzarchitektur mit erweiterter Prozesskette für die Erkennung von Aktivitäten. Hintergrundwissen, das entweder aus Interaktion direkt mit dem Benutzer oder aus einer Wissensbasis stammt, wird als zusätzliche Informationsquellen genutzt, darüberhinaus gibt es eine Rückführung der Klassifikationsergebnisse in die Datenakquisition, die Merkmalsauswahl und die Klassifikation.

in den Daten zu verringern und eine gute Erkennung auch bei Verfügbarkeit von nur kleinen Trainingsdatenmengen zu ermöglichen. Darüberhinaus kann die Merkmalsauswahl von Hintergrundwissen und Erkennungsresultaten auf den Trainingsdaten profitieren, um bei der Auswahl Aktivitäts-spezifischer Merkmale die Robustheit gegen Ausreißer und Rauschen noch weiter zu verbessern.

Die Klassifikation kann Hintergrundwissen und Klassifikationsergebnisse aus vorherigen Zeitschritten im Trainingsprozess nutzen, um die Auswahl Aktivitäts-spezifisch geeigneter Klassifikatoren und die Nachbearbeitung der Erkennungsergebnisse zu ermöglichen, wenn mehrere Erkenner für verschiedene Aktivitäten genutzt werden.

Eine weitergehende Automatisierung der Prozesskette als üblich in derartigen Systemen wurde realisiert, um den Anforderungen der Erweiterbarkeit, die auch von Nicht-Experten durchführbar sein soll, zu genügen. Dadurch kann das System autonom Prozessschritte parametrisieren und ausführen, die sonst häufig noch Benutzereingriffe erfordern würden.

Hintergrundwissen wird wie oben angedeutet in allen Prozessschritten zur Verbesserung der Ergebnisse verwendet. Das Hintergrundwissen kann dabei aus zwei Quellen stammen, entweder direkt vom Menschen durch interaktive Eingriffsmöglichkeiten, oder aus einer persistenten Wissensbasis. Die persistente Speicherung von interaktiv erlangtem Wissen, um damit zukünftig Interaktion durch Zugriffe auf die Wissensbasis zu ersetzen, ist ebenfalls denkbar. In den einzelnen Komponenten werden unterschiedliche Arten von Wissen eingesetzt, deren Details in den jeweiligen Abschnitten in Kapitel 5 und Kapitel 6 diskutiert werden.

Die Ergebnisse der Erkennung werden in zurückgeschleift in die Komponenten der *Merkmalsauswahl* und der *Klassifikation*, um dort das in den Erkennern vorhandene Wissen über Aktivitäten einzusetzen. Durch das Einlernen neuer Aktivitäten werden damit auch weitere Informationen für die vorherigen Prozessschritte beim Einlernen weiterer Aktivitäten verfügbar.

5. Erweiterte Verfahren zur Beobachtung menschlicher Bewegungen

Der Zweck der Datenakquisition im Rahmen der Aktivitätserkennung ist die Beobachtung von menschlichen Bewegungen. Dieses Kapitel beschreibt die Untersuchungen, die zur Verbesserung der Perzeption durch den Einsatz von Hintergrundwissen durchgeführt wurden, um für eine detaillierte Erkennung von Aktivitäten die benötigten, genauen und feinaufgelösten Daten bereitstellen zu können. In Abschnitt 5.1 wird ein Überblick über die Gesamtheit dieser Arbeiten gegeben und in das in Abschnitt 3.1 vorgestellte Trackingsystem *VooDoo* eingeordnet. Die folgenden Abschnitte stellen dann die einzelnen Teilaspekte der Verbesserungen im Detail vor.

5.1. Überblick über Verbesserungen der Personenbeobachtung

Der Stand des *VooDoo*-Trackingsystems, der Ausgangspunkt der folgenden Arbeiten war (wie dargestellt in Kapitel 3.1), liefert zwar gute Resultate für eine reine Beobachtung von Bewegungen. Für die Verwendung in einem System zur Erkennung von Aktivitäten, das in realen Roboterexperimenten genutzt werden soll, gelten aber erhöhte Anforderungen was die Konsistenz beobachteter Gelenkwinkel und -bewegungen angeht, und Fehlstellungen, deren vereinzeltes Auftreten bei der reinen Beobachtung toleriert werden kann, stellen ein zu lösendes Problem dar. Die durchgeführten Arbeiten dienen daher der Verbesserung des Tracking in Bezug auf Robustheit und Qualität, um es für den Einsatz in den angestrebten Roboterszenarien verwendbar zu machen.

Um ohne manuellen Eingriff in Interaktion mit dem System zu treten, wurde eine automatische Initialisierung von Körpermodellen für Personen im Sichtbereich entwickelt, die in Abschnitt 5.2 beschrieben wird. Das Tracking selbst zeigt Schwächen in bestimmten Situationen und bei bestimmten Bewegungen. Das Hauptproblem stellt das Fehlen von Begrenzungen für die Gelenkwinkel eines beobachteten Modells dar, die durchgesetzt werden können. Zur Behebung dieses Problems dient die Entwicklung einer Modellierung von Gelenkwinkelbegrenzungen und der darauf basierenden, in das Tracking integrierten Komponente. Die Details dieses Ansatzes werden in Abschnitt 5.3 beschrieben.

Abb. 5.1 zeigt die Einordnung der Arbeiten in das VooDoo-Trackingsystem (orange markiert), speziell das Verhältnis zu den existierenden Modulen. Die automatische Modellinitiali-

sierung dient als Alternative zur manuellen Modellinitialisierung, die Gelenkwinkelbegrenzungen ergänzen die schon existierenden Gelenkwinkelbeschränkungen der Modelle.

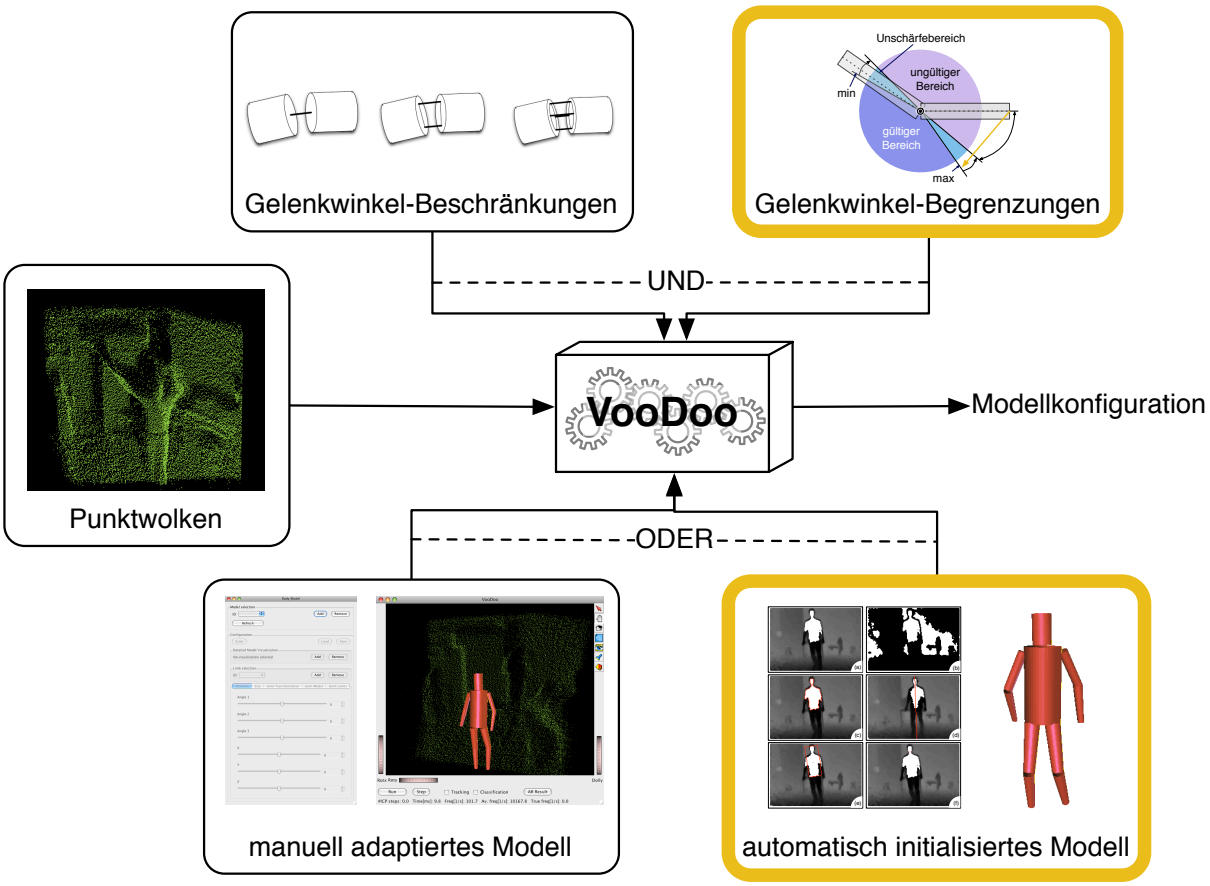


Abb. 5.1.: Einordnung der durchgeführten Forschungsarbeiten zur Personenbeobachtung in das VooDoo-Trackingsystem, markiert in orange. Das System verwendet Punktwolken als Eingabe, das Modell kann entweder manuell adaptiert oder automatisch initialisiert (neu) werden. Für die Trackingschritte werden noch Gelenkwinkel-Beschränkungen und Gelenkwinkel-Begrenzungen (neu) ergänzt. Als Resultat liefert das System die aktuelle Modellkonfiguration zurück.

5.2. Automatische Modell-Initialisierung

Ein zentraler Punkt für ein erfolgreiches Tracking stellt die korrekte Modell-Initialisierung dar. Nur wenn ein Modell in korrekter Größe und Position initialisiert wird, kann mit guten Ergebnisse für die weitere Nachführung gerechnet werden. Dazu wurde ein Verfahren entwickelt, das anhand von Tiefenbildern und Wissen über die menschlichen Körperproportionen eine initiale Bestimmung der Größe und Position vornehmen kann, indem Silhouetten gesucht und bezüg-

lich ihrer Wahrscheinlichkeit, eine menschliche Silhouette zu sein, evaluiert werden [Lösch et al., 2009].

5.2.1. Eingesetzte Daten

Als Daten für den Einsatz des Initialisierungs-Algorithmus werden 3D-Punktwolken der Szene benötigt, im Gegensatz zu anderen Verfahren, die auf 2D-Bilddaten arbeiten wie beispielsweise [Mittal et al., 2003; Parameswaran and Chellappa, 2004; Agarwal and Triggs, 2006].

Daten der benötigten Form können von einer Reihe unterschiedlicher Sensorsysteme bereitgestellt werden, allerdings hat die Wahl des Sensors starken Einfluss auf die Echtzeitfähigkeit des Gesamtsystems. Einerseits stellt die Geschwindigkeit des Sensors eine natürliche Obergrenze für die Systemgeschwindigkeit dar, andererseits erhöht eine wachsende Punktemenge auch die Verarbeitungszeit. Daher muss ein guter Mittelweg zwischen zu niedriger Auflösung für gute Ergebnisse und zu hoher Auflösung für Echtzeitfähigkeit gefunden werden.

Als Standard-Sensor wird im Folgenden eine SwissRanger^{TM} SR3000-Tiefenkamera der Firma Mesa Imaging[1] eingesetzt, die Punktwolken mit einer Auflösung von 176×144 Pixel mit einer Frequenz von ungefähr 25Hz aufnehmen kann.

5.2.2. Initialisierungs-Algorithmus

Die Initialisierung läuft in zwei Stufen ab. In der ersten Stufe werden menschenähnliche Silhouetten im Sichtbereich der Kamera gesucht unter Einsatz klassischer Bildverarbeitungsmethoden. In der anschließenden zweiten Stufe werden die aus der ersten Stufe resultierenden Hypothesen verifiziert und gegebenenfalls noch adaptiert, unter Verwendung der verfügbaren 3D-Daten und sorgfältig gewählter Annahmen über Beschränkungen der Welt und möglicher Systemkonfigurationen.

Stufe 1: Mensch-Hypothesen aus Daten extrahieren

Die Extraktion von Hypothesen erfolgt in drei Schritten: Berechnung von Tiefenbildern, Extraktion von Konturen, und schließlich die Evaluation von Entscheidungskriterien auf den gefundenen Konturen.

Grundlage für das entwickelte Verfahren sind Tiefenbilder. Abhängig von der verwendeten Sensorik gibt es unterschiedliche Wege, wie das benötigte Tiefenbild bereitgestellt werden kann. Für den verwenden SwissRanger Tiefensensor, der 3D-Punktwolken mit Koordinatenzentrum im Sensorchip der Kamera liefert, genügt es die z-Koordinate der Punkte zu verwenden.

[1] `http://www.mesa-imaging.ch/`

Wenn der Koordinatenursprung an anderer Stelle liegt, ist eine Projektion der Punkte auf eine entsprechende, parallel zur Bildebene der Kamera positionierte, Ebene nötig. Abb. 5.2 zeigt ein beispielhaftes Tiefenbild, generiert aus einer mit einem SwissRanger gewonnenen Punktwolke.

Abb. 5.2.: Tiefenbild einer Szene, in der ein Mensch im Vordergrund steht. Erhalten als z-Koordinate einer Aufnahme mit einem Swissranger 3000-Sensor.

Das Bild wird segmentiert aufgrund der Tiefeninformation, in Schritten von 1 m, mit der zusätzlichen Randbedingung, dass ein Mensch vollständig sichtbar sein sollte, sodass jeweils pro Bildframe 4 unterschiedliche Tiefenbilder(-ausschnitte) generiert werden. Jedes dieser Bilder wird binarisiert, und die entstehenden Bildblobs werden anhand verschiedener Kritierien daraufhin untersucht, ob es sich bei den jeweiligen Silhouetten um die Silhouette eines Menschen handelt oder nicht. Abb. 5.3(a) zeigt einen markierten Kandidaten für eine menschliche (Teil-)Silhouette.

Als Kriterien werden einfache Metriken eingesetzt, von denen jede eine Einschränkung an die vorliegende Kontur prüft, die aus bekannten Eigenschaften über die menschliche Figur hergeleitet wurden. Die Verwendung einfacher, wohlverstandener Einschränkungen bietet verschiedene Vorteile gegenüber komplexer, weniger gut verstandener Metriken. Neben der einfacheren Berechnung und der größeren Robustheit gegenüber Fehlern ist auch eine größere Verständlichkeit der Metriken für den Benutzer (und damit die einfachere Möglichkeit, notwendige Anpassungen direkt vornehmen zu können) gegeben.

Die Kriterien sind als Kaskade von Prüfungen angelegt, damit nicht-menschliche Silhouetten möglichst schnell aussortiert werden können (ähnlich dem bei Viola und Jones [Viola and Jones, 2001] beschriebenen Kaskaden-Ansatz). In Versuchen mit verschiedenen Kriterien haben sich die folgenden 4 Kriterien (K1) – (K4) als sinnvoll und ausreichend für die robuste Erkennung von Menschen herauskristallisiert, und bieten einen guten Ausgleich zwischen Flexibilität in der Erkennung verschiedener Posen und Robustheit gegenüber Fehlerkennungen:

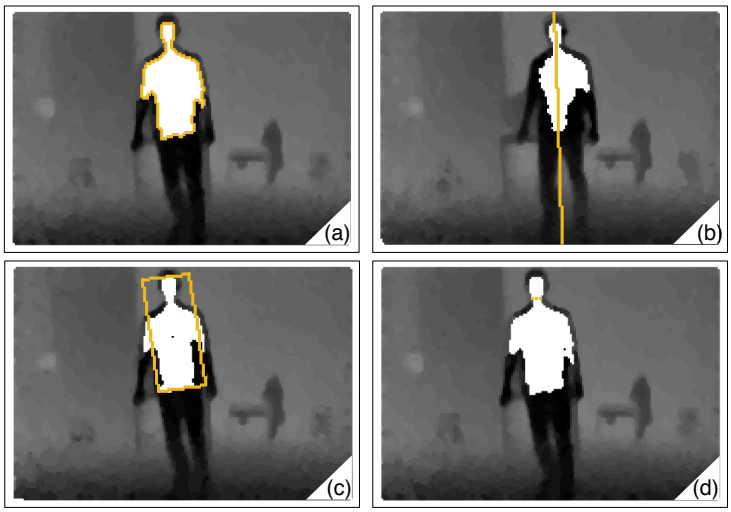


Abb. 5.3.: Ergebnisbilder für bestimmte Schritte in Stufe 1 der Modellinitialisierung: (a) extrahierte Kontur des Menschen, (b) Hauptachse der extrahierten Kontur, (c) Boundingbox die die wichtigen Teile der Kontur enthält, (d) Hals und zugehöriger Referenzpunkt (Punkt in der Mitte) geschätzt aus Kontur.

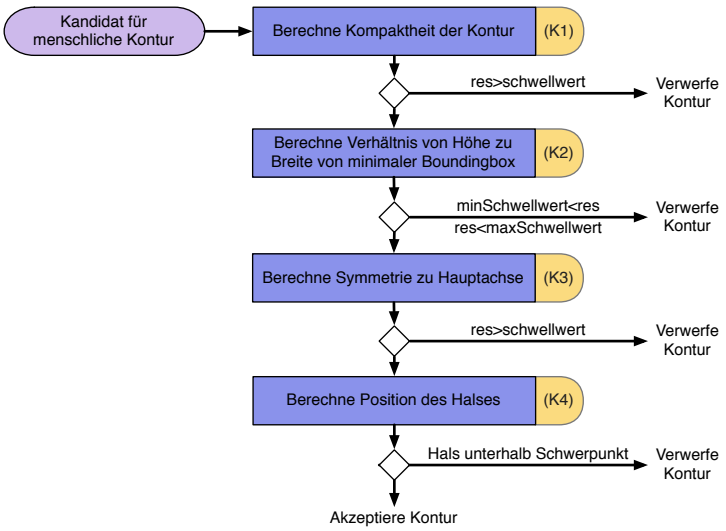


Abb. 5.4.: Schematische Darstellung der Prüfungs-Kaskade, die zur Evaluation von extrahierten Konturen benutzt wird, um die Wahrscheinlichkeit zu bestimmen, mit der die Kontur von einem Menschen stammt. Der Ansatz ist analog einem zusammengesetzten Klassifikator, der mittels Boosting trainiert wird. Das Ergebnis jeder Stufe der Kaskade wird mit einem Schwellwert verglichen, und bei fehlgeschlagenem Test wird die Kontur als Kandidat verworfen.

(K1) Kompaktheit $\mathcal{K}$ der Kontur

(K2) Verhältnis $\mathcal{V}$ von Höhe und Breite einer minimalen Boundingbox um die Kontur

(K3) Symmetrie $\mathcal{S}$ einer Kontur zur Hauptachse

(K4) Position $\mathcal{P}$ des Halses

Die Kaskade der Prüfungen ist hierarchisch angeordnet, um die benötigte Rechenzeit möglichst gering zu halten. Die Einschränkung mit dem besten Verhältnis von Aufwand zu Erkennungsrate wird als erste angewandt und so weiter. Die Reihenfolge der Prüfungen wurde empirisch bestimmt, mit dem in Abb. 5.4 gezeigten Ergebnis.

Die Berechung der Kompaktheit $\mathcal{K}$ für Kriterium (K1) erfolgt mittels Gleichung 5.1, unter Verwendung von U_K als Umfang der Kontur und A_K als Fläche der Kontur K.

$$\mathcal{K} = \mathcal{K}_K = \frac{U_K^2}{A_K} \qquad [5.1]$$

Für die Berechnung der übrigen Kriterien (K2) – (K4) wird die Hauptachse HA_K der Kontur (sowie der Richtungsvektor $\mathbf{HA}_K$ der Hauptachse) genutzt, wie sie in Abb. 5.3(b) eingezeichnet ist, und das minimale, die Silhouette umschließende Rechteck (engl. *bounding box*) benötigt. Die Hauptachse wird mittels einer *Hauptachsenanalyse* (engl. *principal component analysis, PCA*) bestimmt. Im Fall einer menschlichen Kontur sollte die berechnete Achse mit der Hauptachse des Menschen korrespondieren, und annähernd vertikal zum Boden stehen. Die Hauptachse wird auch genutzt, um Teile der Konturen, die eine zu große Distanz von der Hauptachse aufweisen, zu entfernen. Dadurch wird ein negativer Einfluss von solchen Ausreißern auf die folgenden Berechnungen (beispielsweise die Größe der Boundingbox) verringert. Zur Bestimmung einer minimalen Boundingbox, wie sie in Abb. 5.3(c) eingezeichnet ist, wird ein *Scanline-Algorithmus* eingesetzt, der anhand der Hauptachse nach Tangenten der Kontur sucht. Die gefundenen Tangenten müssen entweder parallel oder orthogonal zur Hauptachse sein, und den Rand der Kontur markieren. Das Resultat des Algorithmus' ist dann ein minimales Rechteck, das die Kontur umschreibt.

Kriterium (K2) wird durch Betrachtung des Verhältnisses $\mathcal{V}$ zwischen Breite b_{BB} und Höhe h_{BB} der Boundingbox bestimmt, wie in Gleichung 5.2 angegeben. Dieses Kriterium ist stark abhängig vom Blickwinkel des eingesetzten Sensors: Wenn eine Person im Sensorbild vollständig sichtbar ist, wird der Verhältnis sehr klein sein, aber wachsen, wenn sich die Person dem Sensor nähert und dabei schrittweise Körperteile aus dem Sichtbereich herausragen.

$$\mathcal{V} = \mathcal{V}_{BB} = \frac{b_{BB}}{h_{BB}} \qquad [5.2]$$

Um unabhängig von der Sensorposition gute Resultate zu erzielen, kann entweder eine sehr große Spanne zwischen oberem und unterem Schwellwert gewählt werden (was zu Problemen in Bezug auf die Qualität der Erkennung von Menschen führt), oder die Schwellwerte müssen anhand der Entfernung der Kontur zum Sensor und vom Öffnungswinkel der Linse adaptiert werden.

Das nächste ausgewertete Kriterium (K3) ist die Bewertung der Symmetrie der Kontur. Die Annahme der Symmetrie einer menschlichen Kontur K ist sinnvoll für Menschen, die dem Sensor zugewandt sind, ohne den Körper verdreht zu haben. Da Symmetrie in diesem Sinn (relativ zur Hauptachse einer Kontur, in Bezug auf den Blickwinkel eines Sensor, der sich in ähnlicher Position wie der menschliche Kopf befindet) auch eine Eigenschaft vieler Gegenstände wie Möbel u.ä. ist, muss dieses Kriterium auf jeden Fall mit weiteren Kriterien kombiniert werden. In der Literatur sind viele verschiedene Methoden zur Berechnung von Symmetrien bekannt [Kovesi, 1997]. Aufgrund der Zeitanforderungen in der vorliegenden Anwendung für Mensch-Roboter-Interaktionen wurde auf einen Ansatz zurückgegriffen, der möglichst schnell und unter Verwendung nur einfacher Berechnungen arbeitet.

Die Kontur wird in eine binäre Bitmap eingetragen, um die Differenz der Pixel u auf der linken Seite der Hauptachse werden und der Pixel v auf der rechten Seite der Hauptachse zu berechnen (in den folgenden Gleichungen werden die Variablen u_d bzw. v_d verwendet als Platzhalter für ein Pixel links bzw. rechts der Hauptachse mit dem Abstand d zur Hauptachse der Kontur). Die verbleibenden Werte werden aufsummiert und bilden das Maß für die Symmetrie $\mathcal{S}$ der Kontur. Je kleiner der errechnete Wert, desto symmetrischer ist die Kontur. Für die Berechnung (siehe Gleichung 5.3) werden die Pixel dabei gemäß ihrem Abstand d zur Hauptachse gewichtet mit einem Gewichtsfaktor w_d. Das verwendete Gewicht nimmt linear mit der Entfernung ab gemäß Gleichung 5.4, wodurch insbesondere auch der Einfluss der Position der Arme (die die beweglichsten Körperteile darstellen) verringert wird. Das Gewicht wird normalisiert mit der maximalen Höhe der Kontur (was der Höhe h_{BB} der Boundingbox BB entspricht).

$$\mathcal{S} = \mathcal{S}_K = \left| \sum_{u_d} w_d u_d - \sum_{v_d} w_d v_d \right| \qquad [5.3]$$

$$w_d = 1 - \frac{d}{\frac{h_{BB}}{2} + 1} = \frac{h_{BB} + 2 - 2d}{h_{BB} + 2} \qquad [5.4]$$

Die Position des Halses (definiert durch das Punktepaar $(\mathcal{P}_1, \mathcal{P}_2)$ der Punkte am Rand des Halses links und rechts des Kehlkopfes) ist als Kriterium (K4) das Menschen-spezifischste der vier eingesetzten Kriterien. Die Eigenschaft des Halses in Bezug auf die gesamte Silhouette, dass der Hals der schmalste Teil der Silhouette am Oberkörper des Menschen ist, macht dieses

Kriterium relativ unempfindlich gegenüber Änderungen der Sensorposition und Rauschen in den Daten. Ein Beispiel für eine erkannte Halsposition wird in Abb. 5.3(d) gezeigt.

Zur Bestimmung des Halses wird die untersuchte Kontur abgeleitet. Dazu müsste die Punktbasierte Darstellung eigentlich in eine Spline-Darstellung überführt werden (siehe [i Capo et al., 2006] für Details zu diesem Ansatz). Zur Beschleunigung wird allerdings ein etwas veränderter Ansatz verfolgt, der auf der Differenz zwischen benachbarten Konturpunkten basiert. Diese Differenz kann je nach Vorzeichen als positive oder negative Ableitung in dem Punkt der Kontur interpretiert werden. Ein Durchlauf über alle Punkte im Uhrzeigersinn liefert die Bedingung, dass zum Hals gehörende Punkte zwischen Punkten mit positiver und negativer Ableitung liegen müssen. Bei einem Paar von Kandidatenpunkten (p_1, p_2) müssen die beiden Punkte auf unterschiedlichen Seiten Hauptachse HA_K liegen, siehe Gleichung 5.5. Dabei steht $\mathcal{B}_K$ für den Schwerpunkt der Kontur K. Das Paar $(\mathcal{P}_1, \mathcal{P}_2)$ wird als das Kandidatenpaar mit der minimalen Entfernung gewählt (siehe Gl. 5.6). Mit dieser Methode kann die Positions des Halses sehr schnell und mit ausreichender Genauigkeit berechnet werden.

$$\operatorname{sgn}((\mathcal{P}_1 - \mathcal{B}_K) \bullet \mathbf{s}) \neq \operatorname{sgn}((\mathcal{P}_2 - \mathcal{B}_K) \bullet \mathbf{s}) \qquad \text{mit} \quad \mathbf{s} \perp HA_K \tag{5.5}$$

$$\mathcal{P}_1, \mathcal{P}_2 = \underset{p_1, p_2 \text{ Kandidaten}}{\arg\min} |p_1 - p_2| \tag{5.6}$$

Kriterium (K4) prüft zwei Bedingungen an die Position des Halses. Erstens muss die verbindende Linie zwischen den gegenüberliegenden Punkten $\mathcal{P}_1$ und $\mathcal{P}_2$ annähernd vertikal zur Hauptachse HA_K des Körpers liegen. Das wird geprüft, indem der Wert $\mathcal{N}_{orth}$ mit einem klein gewählten Schwellwert verglichen wird (siehe Gl. 5.7 und 5.8). Zweitens muss die Verbindung von Schwerpunkt $\mathcal{B}_K$ und Hals $\mathcal{P} = \frac{\mathcal{P}_1 + \mathcal{P}_2}{2}$ annähernd parallel zur Hauptachse verlaufen, was durch den Vergleich von Wert $\mathcal{N}_{par}$ mit einem Schwellwert geprüft wird (siehe Gl. 5.9 und 5.10).

$$\overline{\mathcal{P}_1\mathcal{P}_2} \perp HA_K \;\Leftrightarrow\; (\mathcal{P}_1 - \mathcal{P}_2) \bullet \mathbf{HA}_K = 0 \tag{5.7}$$

$$\mathcal{N}_{orth} = \overrightarrow{\mathcal{P}_2\mathcal{P}_1} \bullet \mathbf{HA}_K \tag{5.8}$$

$$\overline{\mathcal{P} - \mathcal{B}_K} \parallel HA_K \;\Leftrightarrow\; \overrightarrow{\mathcal{B}_K\mathcal{P}} \bullet \mathbf{HA}_K = \left|\overrightarrow{\mathcal{B}_K\mathcal{P}}\right| \cdot |\mathbf{HA}_K| \tag{5.9}$$

$$\mathcal{N}_{par} = \frac{\overrightarrow{\mathcal{B}_K\mathcal{P}}}{\left|\overrightarrow{\mathcal{B}_K\mathcal{P}}\right|} - \frac{\mathbf{HA}_K}{|\mathbf{HA}_K|} \tag{5.10}$$

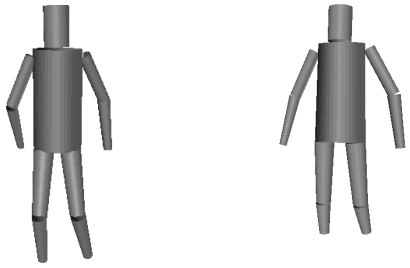
Stufe 2: Verifikation und Adaption gefundener Mensch-Hypothesen

Die in Stufe 1 bestimmten Kontur-Hypothesen werden in der zweiten Stufe bewertet und gegebenfalls noch adaptiert mit Hilfe der 3D-Sensordaten und weiterer Annahmen über die Welt und die Systemkonfiguration. Schließlich wird die Größe der akzeptieren Personen geschätzt auf Grundlage eines Referenzmodells. Die durchlaufene Prozesskette umfasst die folgenden 4 Schritte:

(1) Bestimmung eines Referenzpunkts Für die weiteren Berechnungen (z.B. die Bestimmung der Körpergröße) wird ein Referenzpunkt auf dem Körper im 3D-Koordinatensystem benötigt. Als Referenzpunkt $\mathcal{P}_{ref}$ wird der Mittelpunkt der Verbindung zwischen den in Gl. 5.6 beschriebenen Punkten $\mathcal{P}_1$ und $\mathcal{P}_2$ genutzt, siehe Gleichung 5.11. Abb. 5.3(d) zeigt den Referenzpunkt eingezeichnet in ein Bild der Sensordaten.

$$\mathcal{P}_{ref} = \frac{\mathcal{P}_1 + \mathcal{P}_2}{2} \qquad [5.11]$$

(2) Schätzung von Modellgröße und Skalierung In diesem Schritt wird ein VooDoo-Körpermodell in einer fixen Konfiguration als Referenzmodell genutzt. Verschiedene Modelle und Konfigurationen können an dieser Stelle genutzt werden. In Abb. 5.5 werden zwei typische Referenzmodelle gezeigt, die sich in verschiedenen Szenarien bewährt haben.



a) b)

Abb. 5.5.: Beispiele für Referenz-Körpermodelle wie sie für die Initialisierung des *VooDoo*-Trackingsystems eingesetzt werden: a) Körpermodell mit Standardverhältnissen der Körperteile und Beinen in voller Länge. b) Alternatives Körpermodell mit verkürzten Unterschenkeln und einem breiteren Torso als üblich.

Die Wahl des Referenzmodells schränkt die Konfiguration der zu erkennenden Personen ein, das heißt nur Personen mit einer Pose ähnlich der des Referenzmodells können initialisiert werden. Das gewählte Referenzmodell wird genutzt um ein 3D-Modell des als Hypothese erkannten

Menschen zu generieren und in das Trackingsystem einzufügen, nachdem es durch Skalierung auf die passende Größe transformiert wurde.

$$\mathcal{S} = \frac{\mathrm{d}(E_B, \mathcal{P}_{Referenzmodell})}{\mathrm{d}(E_B, \mathcal{P}_{Hypothese})} \qquad [5.12]$$

Die Skalierung $\mathcal{S}$ wird gemäß Gleichung 5.12 berechnet mit $\mathcal{P}_{Referenzmodell}$ die Position des zu $\mathcal{P}_{Hypothese}$ analogen Referenzpunktes im Referenzmodell und E_B die Bodenebene. Nur hier wird Wissen über die Lage des verwendeten Koordinatensystems zum Weltkoordinatensystem benötigt, das entweder manuell oder durch zusätzliche 3D-Bildverarbeitung aus den Sensordaten extrahiert werden muss. Die Funktion $\mathrm{d}(E,P)$ bestimmt den Abstand zwischen Ebene E und Punkt P.

Die ermittelte Größe der Person $\mathrm{d}(E_B, \mathcal{P}_{Hypothese})$ kann zur Bewertung der Hypothese eingesetzt werden, indem es mit einem Intervall gültiger Körpergrößen verglichen wird.

(3) Kollisionsprüfungen In der Praxis ist es möglich, dass mehrere Modelle in einer sehr kleinen Region erkannt werden, die alle von der gleichen Person "verursacht"werden. Gründe dafür können Rauschen in den Daten oder ein ungünstiger Blickwinkel des Sensors sein. Da in solchen Fällen natürlich nur ein Modell wirklich initialisiert werden soll, wird für jedes initialisierte Modell eine Boundingbox bestimmt. Bevor eine der Hypothesen zu einem neuen Modell instanziiert wird, wird erst noch geprüft, dass der Referenzpunkt $\mathcal{P}_{Hypothese}$ des Modells nicht innerhalb einer der vorhandenen Boundingboxen liegt.

Das Detailverhalten des Algorithmus' kann gelenkt werden über die Wahl der Dimensionen der Boundingboxen. Bei der Verwendung minimaler Boundingboxen können auch nahe beieinander stehende Personen noch initialisiert werden, aber im Gegenzug ist ein hochauflösender Sensor nötig, um die nahe beieinander stehenden Körper noch trennen zu können (und damit ein Verschmelzen der Modelle zu verhindern). Bei der Verwendung von größeren Boundingboxen, die durch die Vergrößerung einer minimalen Boundingbox um einen konstanten Faktor bestimmt wird, werden weniger Modelle initialisiert, dafür kann das System im Gegenzug auch mit einem Sensor niedriger Auflösung noch gute Initialisierungen liefern.

(4) Feinjustierung des Modelle In Anschluss an die Instanziierung wird eine abschließende Bewertung und Adaption der neuen Modelle vorgenommen. Durch Messungenauigkeiten in den vorherigen Schritten ist eine Adaption des Modells an die realen Daten noch notwendig. Mehr oder strengere Einschränkungen in den vorherigen Schritten können diesen Fehler zwar verkleinern, aber nicht vollständig elimineren. Die dadurch gewonnene Genauigkeit steht dabei aber nicht in einem günstigen Verhältnis gegenüber der erhöhten Komplexität und dem zusätz-

lichen Berechnungsaufwand. Die Bewertung des Modells erfolgt über einen Konfidenzwert, dessen Berechnung im Folgenden beschrieben wird. Der Konfidenzwert kann interpretiert werden als Qualität des Modells in Bezug auf die Sensordaten.

Zur Berechnung der Konfidenz des vollen Modells werden zuerst die Konfidenzwerte $\mathcal{K}_T$ der einzelnen Körperteile T bestimmt mittels Gleichung 5.13, in der S_T für die gewichtete Summe der vom Trackingalgorithmus dem Körperteil zugeordneten Punkte steht, A_T für die Näherung der Oberfläche des Körperteils, und N für die minimale Anzahl von Punkten, die pro Körperteil zugeordnet sein müssen, damit eine minimale Trackingqualität gesichert werden kann.

$$\mathcal{K}_T = \max\left(1, \frac{S_T}{A_T \cdot N}\right) \qquad [5.13]$$

Die gewichtete Summe S_T der Sensordaten für ein Körperteil T wird aus allen dem Körperteil zugeordneten Punkten (Sensordaten oder künstliche Messpunkte wie sie von Gelenkmodellen eingefügt werden, siehe Kapitel 3.1) berechnet. Die minimale Zahl der Punkte N, die einem Körperteil zugeordnet sein müssen hängen von der Größe und der Relevanz des Körperteils für das Tracking eines ganzen Menschen ab. Die genäherte Oberfläche A_T des Körperteils wird mittels Gleichung 5.14 berechnet.

$$A_T = \frac{p_{B,T} \cdot \eta}{l_T} \left(\mathrm{md}_{boden}(T) + \mathrm{md}_{deckel}(T)\right) \qquad [5.14]$$

In der Gleichung steht η für einen konstanten Skalierungsfaktor. Die Länge l_T des Körperteils, sowie $\mathrm{md}_{boden}(T)$ bzw. $\mathrm{md}_{deckel}(T)$ für den maximalen Durchmesser von unterer (Boden) bzw. oberer (Deckel) Ellipse des Körperteil T modellierenden Zylinders können direkt aus dem Modell gelesen werden. Die prozentuale Nutzung $p_{B,T}$ von Punkten innerhalb der Boundingbox B_T um das Körperteil T misst das Verhältnis *aller* Punkte in Boundingbox B_T zur vom ICP genutzen Teilmenge dieser Punkte. Für diese Berechnung werden Boundingboxen für alle Körperteile benötigt. Schließlich wird ein Konfidenzwert größer als 1 auf 1 normalisiert.

Die Trackingqualität $\mathcal{Q}_M$ für ein vollständiges Körpermodell M wird anschließend aus den Konfidenzen für jedes Körperteil bestimmt in jedem Verifikationsschritt. Dazu wird eine gewichtete Summe der Konfidenzen berechnet, siehe Gleichung 5.15 für die Berechung der Qualität. Die Gewichte der Konfidenzen $\mathcal{K}_T$ können entweder entsprechend der Relevanz des Körperteils T gesetzt werden, oder identisch für alle Körperteile gewählt werden, Tabelle 5.1 listet die verwendeten Gewichtswerte auf.

$$\mathcal{Q}_M = \sum_T w_T \mathcal{K}_T \qquad [5.15]$$

Tab. 5.1.: Gewichtung der einzelnen Körperteile für die Bewertung der Qualität eine neu initialisierten Modells.

Körperteil T	Gewicht w_T
Torso	0.1
Kopf	0.1
Oberarme	0.1
Unterarme	0.1
Oberschenkel	0.1
Unterschenkel	0.1

Das Zeitintervall zwischen den Verifikationsschritten kann angepasst werden, um ein gewünschtes Systemverhalten zu erreichen. Ein großes Intervall hat die Wirkung, dass ein Mensch lange in der Sensorsicht verharren muss, bis die Initialisierung mit einem passenden Modell abgeschlossen wird. Im anderen Fall besteht die Gefahr, dass zu wenig Informationen über den beobachteten Menschen vorliegen, was zu einem suboptimalen Modell als Ergebnis führen kann.

Im Verlauf der Feinjustierung eines Modells wird die Qualität $\mathfrak{Q}_M$ des Modells bewertet und mit einem Schwellwert θ_{Mo} verglichen. Ein Überschreiten des Schwellwertes bedeutet, dass das Modell kleiner als die modellierte Person ist. Infolgedessen wird das Modell um einen konstanten Faktor größer skaliert. Diese Skalierung wird gegebenenfalls in jedem Schritt ausgeführt (das Modell „wächst“) bis die Qualität den Schwellwert erreicht, und die Initialisierung abgeschlossen wird. Der Schwellwert dient also als optimales Verhältnis zwischen Sensordaten und Modell in Bezug auf die oben definierten Konfidenzen.

Die Qualität kann in jedem Schritt nur entweder gleich bleiben oder abnehmen, da die Oberfläche A_T der einzelnen Körperteile zunimmt beim Wachsen des Modells (siehe Gleichung 5.14), während die gewichtete Summe der Punkte S_T gleich bleibt (vergleiche Gleichung 5.13). Wenn die Qualität kleiner als der Schwellwert ist, ist entweder das Modell größer als die Person, oder das Modell passt nicht zu den Sensordaten, d.h. die beobachteten Daten wurden nicht von einem Menschen generiert oder der Mensch steht ein gänzlich anderer Pose als das geladene Modell, sodass das Tracking nicht korrekt loslaufen kann. In beiden Fällen wird das Modell verworfen, um den Aufwand für die Entscheidung zwischen diesen beiden Fällen einzusparen.

5.2.3. Bewertung der Modell-Initialisierung

Der beschriebene Algorithmus ist in das *VooDoo*-Trackingsystem integriert und erfolgreich im Einsatz. Die Initialisierung sich in qualitativen Tests als robust gegenüber Schwankungen in

Körpergröße, Bekleidung und Lichtverhältnissen erwiesen. Weitere Details zur Evaluation des Initialisierungsmoduls werden in Kapitel 7.1.1 diskutiert.

5.3. Gelenkwinkel-Begrenzungen

Trotz der existierenden Gelenkwinkelbeschränkungen in *VooDoo* stellen Fehlstellungen der Gelenke, insbesondere im Bereich der Arme, das Hauptproblem des Trackingsystems dar. Zur Abhilfe wurde eine integrierte Repräsentation für Gelenkwinkel-Begrenzungen entwickelt, die innerhalb des ICP-Algorithmus' die Einhaltung gültiger Gelenkstellungen erzwingt [Lösch et al., 2011], unter Verwendung künstlicher Messpunkte (analog zur Modellierung der Gelenkwinkelbeschränkungen), die durch aus der menschlichen Anatomie bekannten Werten parametrisiert werden.

Die Darstellung des entwickelten Ansatzes beginnt mit der Darstellung der benötigten erweiterten Gelenkmodelle. Anschließend wird die Einbindung dieser Modelle in das Tracking erläutert, gefolgt von der Beschreibung der Integration mittels künstlicher Messpunkte.

5.3.1. Formale Definition

Gelenkwinkelbegrenzungen werden modelliert als parametrisierbare Bedingungen die im Tracking verwendet werden können, um das Auftreten von Fehlstellungen der Gelenke zu erkennen. Wie auch die übrigen Teile des Trackings ist dieser Ansatz sehr allgemein und kann für beliebige Modelle genutzt werden, nicht nur für das Tracking von Menschen. Erst die Auswahl geeigneter Parametersätze instanziiert den Ansatz für das Tracking von Personen, Details zu den beschränkten Gelenken sind in Abschnitt 5.3.4 diskutiert. Die Beispiele in den folgenden Abschnitten wurden ebenfalls aus dem Personentracking entnommen.

Im Folgenden bezeichnet w_a eine aktuelle Winkelmessung, die in unterschiedlichen Konventionen genutzt werden kann (z.B. RPY, Euler-Konvention), abhängig vom Gelenkwinkel und der zu realisierenden Begrenzung der Bewegung (diese Freiheit der Darstellung ermöglicht es, unterschiedliche Begrenzungen auf natürliche Art auszudrücken). Die Nullstellung für diese Messungen wird durch eine Standard-Modellkonfiguration vorgegeben, die in Abb. 5.6 gezeigt wird. Mit w_g wird allgemein ein *Grenzwinkel* bezeichnet, der nicht überschritten werden darf. Im Weiteren werden drei Arten von Gelenkwinkelgrenzen unterschieden, die auf eine Gelenk-Drehachse angewandt werden können. Alle Arten von Bedingungen können unabhängig voneinander im gleichen Gelenk in verschiedenen Freiheitsgraden kombiniert werden, wenn es erforderlich ist. Das bedeutet, dass jede Gelenkwinkel-Begrenzung üblicherweise nur *einen* Freiheitsgrads betrifft, die anderen aber unabhängig davon begrenzt werden können. Beispiels-

weise kann der menschliche Ellbogen (als Scharniergelenk) nicht zur Seite gebeugt werden (d.h. hier tritt eine sehr strenge Begrenzung in Kraft), und auch in der eigentlichen Bewegungsachse ist nur die Beugung in Richtung des Bizeps möglich, aber nicht in die Gegenrichtung (also eine asymmetrische Begrenzung der Bewegung).

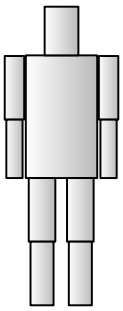


Abb. 5.6.: Vereinfachtes Zylindermodell des menschlichen Körpers, wie es im VooDoo-Trackingsystem verwendet wird. Die Konfiguration des Modells ist die Standard-Modellkonfiguration (Nullstellung) des Modells.

Gelenkwinkelgrenze Typ 1

Die einfachste Form einer Gelenkwinkelbegrenzung, definiert in Gleichung 5.16, ist eine symmetrische Begrenzung des Gelenkwinkels. Typ 1 ist beispielsweise genutzt für die Seitbeugung des Ellbogens. Da in dieser Richtung keine Beugung möglich ist, aber die Einräumung eines kleinen Bewegungsspielraums sich günstig auf das Tracking auswirkt, wird eine kleine Konstante k für die Begrenzung dieser Rotation verwendet mit $w_g = k$.

$$|w_a| < w_g \qquad [5.16]$$

Gelenkwinkelgrenze Typ 2

Begrenzungen vom Typ 2 sind die nächstkomplexe Stufe, definiert in Gleichung 5.17, mit den konstanten $w_{g_{min}}$, $w_{g_{max}}$ als Bezeichnungen für den unteren und oberen Grenzwinkel. Damit sind unterschiedliche obere und untere Grenzen für einen Freiheitsgrad möglich. Typ 2 wird beispielsweise zur Beschreibung der Bewegungsgrenzen des menschlichen Kopfes genutzt.

$$w_{g_{min}} < w_a < w_{g_{max}} \qquad [5.17]$$

Gelenkwinkelgrenze Typ 3

Gelenkwinkelgrenzen vom Typ 3 modellieren die komplexeste Art von Begrenzungen, für Fälle in denen der Operationsbereich eines Gelenks von der aktuellen Konfiguration eines anderen Gelenks abhängt, üblicherweise der Stellung des „Elterngliedes“ (d.h. des nächsten Gelenkes in Richtung des Torsos). Beispielsweise kann der Ellbogen vor dem Körper weiter gebeugt werden als im Rücken, aufgrund der unterschiedlichen Schulterstellungen. Solche komplexen Bedingungen erfordern einen zusätzlichen Term in der Grenzbedingung, wobei das Hinzufügen eines einzelnen, konstanten Wertes nicht genügt. Der Wert des Terms muss sich kontinuierlich und monoton mit der Bewegung des Elterngliedes ändern, bis zur vollen Flexibilität des Gelenkes. Daher wird als zusätzlicher Faktor eine kontinuierliche, streng monoton wachsende Funktion benötigt. Gleichung 5.18 zeigt die allgemeine Form der Gelenkwinkelgrenzen-Bedingung, wobei e_a für die Stellung (aktueller Winkel) des Elterngliedes steht, $w_{d_{min}}$ und $w_{d_{max}}$ für den maximalen zusätzlichen Operationsraum für die untere bzw. obere Grenze des Bewegungsraums, und η für eine Funktion, die einen Abbildungsfaktor for den zuätzlichen Operationswinkel berechnet anhand der Stellung des Elterngliedes.

$$w_{g_{min}} + \eta(e_a) \cdot w_{d_{min}} \; < \; w_a \; < \; w_{g_{max}} + \eta(e_a) \cdot w_{d_{max}} \qquad [5.18]$$

Die Funktion η, definiert in Gleichung 5.19, berechnet für eine gegebene Stellung eine Körpergliedes (als Winkel) den Anteil der zusätzlichen Bewegungsfreiheit, der einem abhängigen Glied ermöglicht wird. Als Eingabe dient die aktuelle Stellung des Elterngliedes e_a, als zusätzliche Parameter müssen noch die Winkel $e_{g_{min}}$ bzw. $e_{g_{max}}$ angegeben werden, die die Stellung des Elternteils angeben, aber der die minimale bzw. maximale zusätzliche Bewegungsfreiheit für das abhängige Glied gewährt wird.

$$\eta(e_a) = \begin{cases} 0.0 & , \quad e_a \leq e_{g_{min}} \\ \sin(e_a) & , \quad e_{g_{min}} < e_a < e_{g_{max}} \\ 1.0 & , \quad e_a \geq e_{g_{max}} \end{cases} \qquad [5.19]$$

Zusammenfassend besteht die Definition einer Gelenkwinkel-Begrenzung aus einem Tupel $(T, A, \mathbf{w}_g)$, mit folgender Bedeutung der Elemente:

T Typ der Gelenkwinkelbegrenzung (1-3, entsprechend der obigen Definitionen).

A Betroffene Rotationsachse des Körperteils, deren Bewegung begrenzt wird.

$\mathbf{w}_g$ Ein Vektor mit einem oder mehreren Grenzwinkeln, wobei die benötigte Anzahl und die exakte Interpretation der Winkel vom Typ der Begrenzung abhängt, vgl. Gleichungen 5.16–5.19.

Körpermodelle können optional um solche Begrenzungen erweitert werden, bei fehlender Definition werden implizite Definitionen für das Trackingmodell verwendet, die die volle Bewegungsfreiheit der einzelnen Gelenke zulassen.

5.3.2. Integration in ICP-Tracking

Um die beschriebenen zusätzlichen Informationen in das ICP-Tracking zu integrieren, was ein wesentlicher Punkt dieser Modellierung ist, muss die Verarbeitungskette des Tracking-Frameworks erweitert werden, sodass in jedem Tracking-Schritt die Gelenke überprüft werden hinsichtlich eventueller Verletzungen der definierten Grenzen, und entsprechende Korrekturen müssen durchgeführt werden.

Wie in Abb. 5.7 dargestellt, wird die Überprüfung und Korrektur von Gelenkwinkelbegrenzungen nach der Berechnung der Nächster Punkt-Relationen durchgeführt. Die Überprüfung besteht dabei aus 4 Schritten, die nacheinander für jedes mit Begrenzungen versehene Gelenk durchgeführt werden:

1. Berechnung der aktuellen Position (in 6D, d.h. Position und Lage) des betroffenen Körperteils.

2. Vergleich der aktuellen Konfiguration mit den definierten Gelenkwinkel-Grenzen.

3. Im Fall einer Fehlstellung werden künstliche Messpunkte zur Korrektur der Abweichung generiert. Die Positionierung dieser Punkte wird in Abschnitt 5.3.3 erläutert.

4. Wenn im vorherigen Schritt neue Messpunkte generiert wurden, müssen sie mit passendem Gewicht in den globalen Punkte-Cache des Trackings eingefügt werden. Die Wahl passender Gewichte wird ebenfalls in Abschnitt 5.3.3 erläutert.

Abb. 5.8 zeigt die Schritte als Ablaufdiagramm. Im Gesamtablauf erfolgt nach der Auswertung der Gelenkwinkel*begrenzungen* die Auswertung der Gelenkwinkel*beschränkungen*, die insbesondere auch den Zusammenhalt der einzelnen Körperteile im Rahmen des Trackings sicherstellen. Anschließend wird das Tracking wie üblich mit der kleinste Quadrate-Optimierung fortgesetzt.

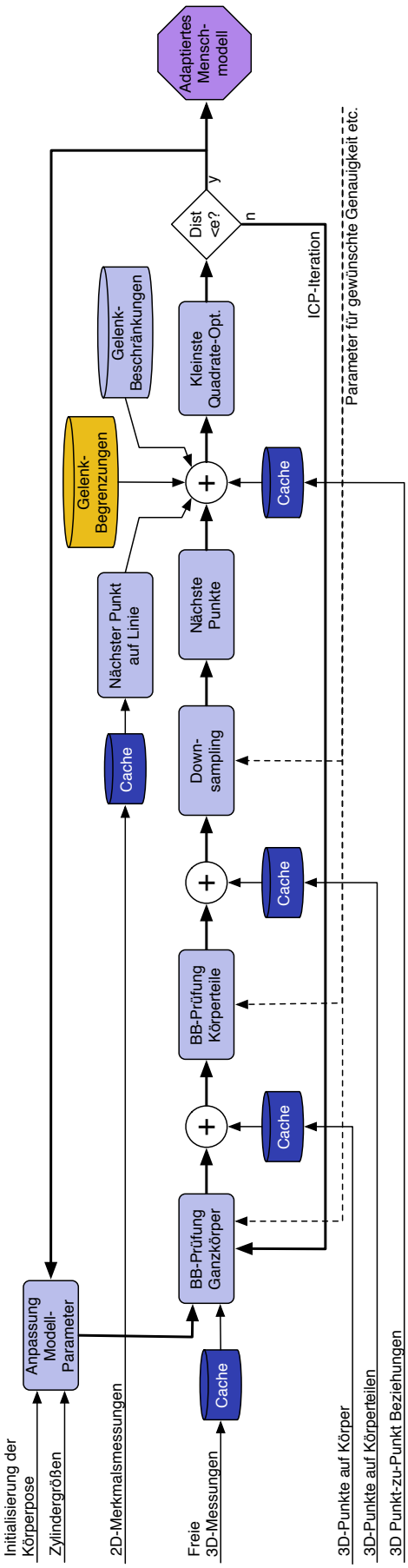


Abb. 5.7.: Prozesskette des VooDoo-Trackings wie ursprünglich dargestellt in Abb. 3.1, erweitert um die Benutzung von Modellen für die Begrenzung von Gelenkwinkeln (orange markiert). Aus den Gelenkwinkelgrenzen werden künstliche Messungen generiert, die zusammen mit den *Nächsten Punkten* als Eingabe für die Kleinste Quadrate-Optimierung dienen.

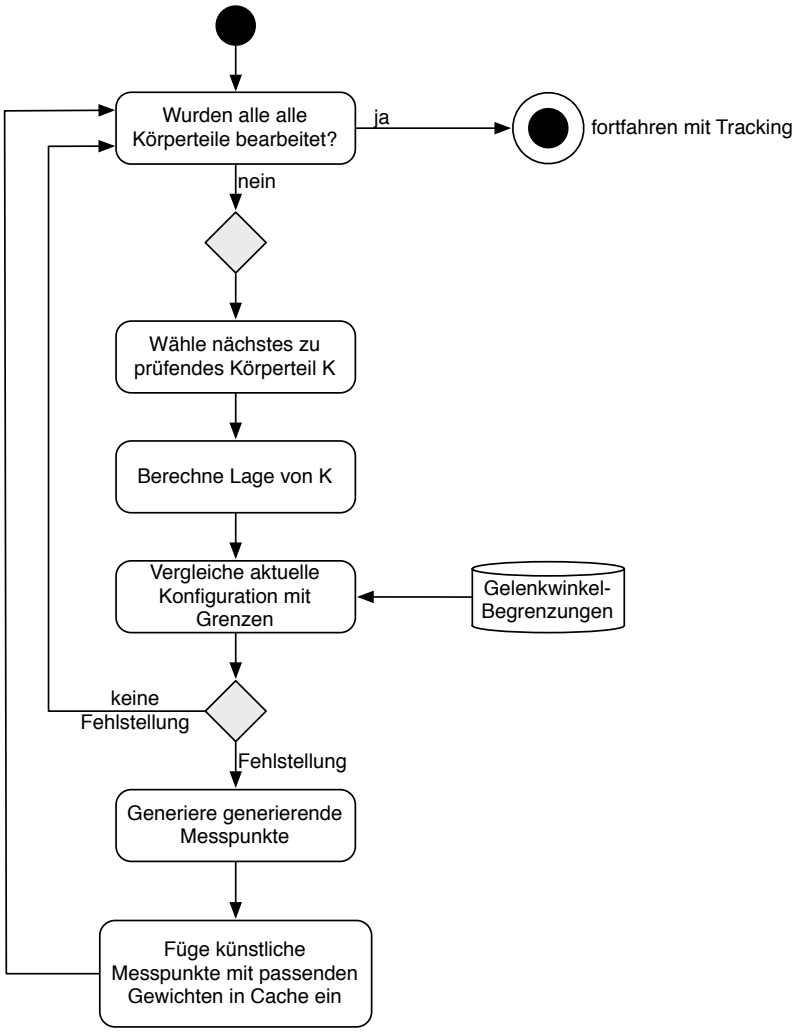


Abb. 5.8.: Ablauf der Überprüfung und Korrektur von Gelenkwinkel-Begrenzungen im *VooDoo*-Tracking.

5.3.3. Korrektur von Fehlstellungen

Bei der Benutzung der Gelenkwinkelgrenzen zur Korrektur sind drei Faktoren in ihrem Zusammenspiel entscheidend für den Erfolg: (1) Die Positionierung der künstlichen Messpunkte, die die Korrektur bewirken sollen. (2) Die Wahl der korrespondierenden Punkte auf dem Modell zu den künstlichen Messpunkten. (3) Die Gewichtung der künstlichen Messpunkte. Die ersten beiden Punkte sind abhängig von der Rotationsachse, die begrenzt werden soll, während die Gewichtung unabhängig davon ist.

Für die Positionierung und Korrespondenzwahl muss unterschieden werden zwischen Rotationen um eine der Achsen der Basisellipse eines Körperteil-Zylinders (wobei die Achsen den Durchmessern der Ellipse entsprechen), im folgenden als *laterale Rotation* bezeichnet, sowie Rotation um die Längsachse eines Körperteils, im Folgenden als *longitudinale Rotation* bezeichnet.

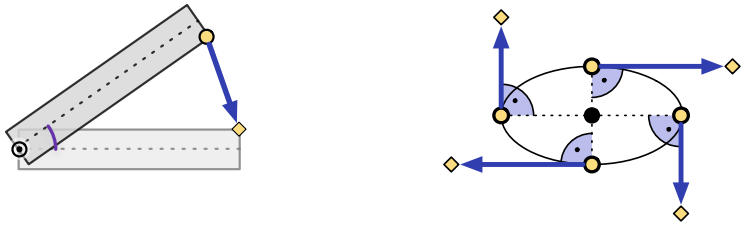
In den nächsten Abschnitten werden zunächst die Punkte (1) und (2) abhängig von der Rotationsachse behandelt, anschließend wird die Gewichtung diskutiert. Schließlich wird im letzten Abschnitt noch eine Erweiterung der Gelenkwinkel-Begrenzungen vorgestellt, die durch eine Relaxation der Grenzwinkel eine starke Verringerung der Fehlstellungen ermöglicht.

Laterale Rotation

Für die Korrektur einer Fehlstellung bei lateralen Rotationen genügt das Hinzufügen einer einzelnen künstlichen Messung. Für die Positionierung wird der zu korrigierende Zylinder als Basis gewählt. Die Korrekturmessung wird am Ende dieser Basis plaziert, in demjenigen Endpunkt eines Ellipsen-Durchmessers, der die kürzeste Entfernung zu dem fehlgestellten Glied aufweist. Der entsprechende Korrespondenzpunkt auf dem Modell wird in analoger Weise als Endpunkt desjenigen Ellipsendurchmessers am Zylinderende gewählt, der die geringste Entfernung zum hypothetischen Basiszylinder hat. Abb. 5.9(a) zeigt eine vereinfachte zweidimensionale Darstellung dieses Ansatzes.

Longitudinale Rotation

Bei einer Grenzwinkelverletzung einer Rotation um eine Längsachse genügt im Allgemeinen ein einzelner künstlicher Messpunkt nicht mehr zur Korrektur der Fehlstellung, Abb. 5.10 zeigt ein Beispiel für die zur endgültigen Lösung führenden Überlegungen. Die Beispiele in der Abbildung sind vereinfacht, indem einerseits der 2D-Fall einer Ellipse dargestellt wird statt eines verallgemeinerten Zylinders in 3D, und andererseits keine echten Messungen gezeigt werden. Wie die Abbildungen 5.10(a) und 5.10(c) zeigen, kann es selbst in diesem vereinfachten Fall



(a) (b)

Abb. 5.9.: Schematische Darstellung künstlicher Messpunkte und korrespondierender Modellpunkte für die Korrektur von Verletzungen von Gelenkwinkel-Begrenzungen. Gelbe Rauten bezeichnen künstliche Messungen, gelbe Punkte Korresponzpunkte des Modells. Grüne Pfeile zeigen die resultierende Kraft (Auslenkung) auf den Zylinder. (a) Vereinfachte Darstellung in 2D für ein Scharniergelenk (laterale Rotation). (b) Draufsicht auf Rotation entlang der Längsachse (longitudinale Rotation).

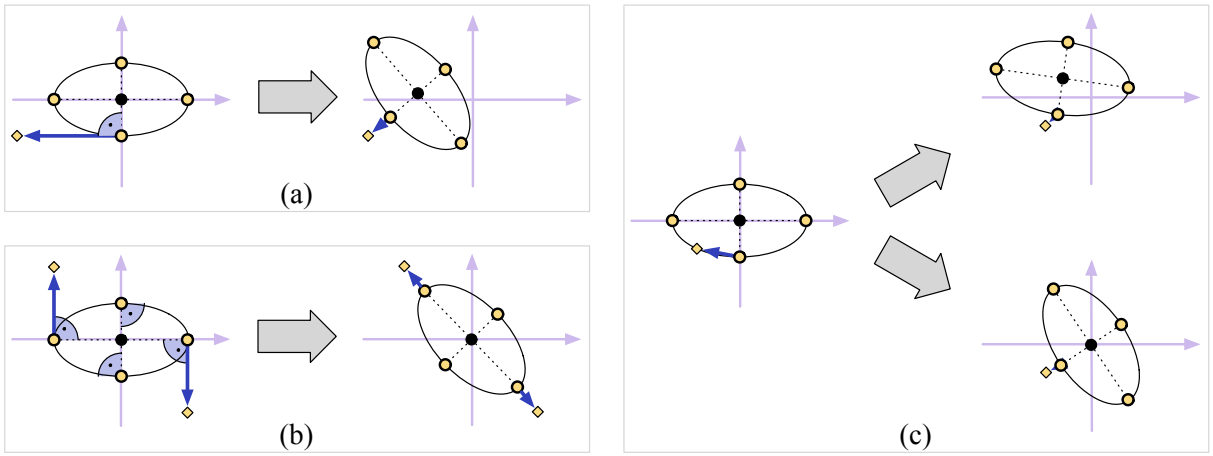


Abb. 5.10.: Vereinfachte Darstellung (in 2D und ohne zusätzliche Messungen) der Ergebnisse bei der Korrektur einer longitudinalen Fehlstellung mit Hilfe von weniger als vier künstlichen Messungen: (a) 1 Korrespondenz platziert auf einer Tangenten, (b) 2 Korrespondenzen platziert auf 2 gegenüberliegenden Tangenten, (c) 1 Korrespondenz platziert auf Zylinder/Ellipse, mit zwei möglichen Korrekturergebnissen (abhängig von Sensormessungen sind beide Ergebnisse möglich).

zu einer Auslenkung des Modells kommen, obwohl eigentlich nur eine Rotation zur Korrektur der Stellung bewirkt werden sollte. Die beiden Teilabbildungen zeigen dabei unterschiedliche Möglichkeiten zur Platzierung der künstlichen Messungen, einmal auf einer Tangenten entlang der Ellipse in Abb. 5.10(a), einmal *auf* der Ellipse mit dem Winkel der Fehlstellung in Abb. 5.10(c). Um dieses Problem zu beheben, genügen eigentlich 2 Korrespondenzen, die so platziert werden, dass sie sich gegenseitig die jeweiligen unerwünschten Effekte neutralisieren, siehe Abb. 5.10(b). Allerdings ist in der Praxis auch die Korrektur mit 2 Korrespondenzen nicht immer robust, da die Sensormessungen (aufgrund der Perspektive) nicht symmetrisch auf allen Seiten des Zylinders verteilt sind. Um eine robuste Korrektur zu erreichen, werden daher 4 Punktepaare genutzt, wie in Abb. 5.9(b) dargestellt. Die Korrespondenzpunkte auf dem Modell werden an den Endpunkten der Ellipsen-Durchmesser der Basis-Ellipse des Zylinder platziert, die Messpunkte in der gleichen Ebene (in der auch die Ellipse liegt) jeweils auf einer Tangente an der Ellipse, in Richtung der benötigten Korrektur-Rotation.

Gewichtung der Messpunkte

Jedem der künstlichen Messpunkte wird ein Gewicht g zugewiesen, wie in Abschnitt 3.1 erklärt. Für Korrekturpunkte, die die Einhaltung von Gelenkwinkelgrenzen erzwingen, wird g dynamisch berechnet, als Wert proportional zur Stärke der Abweichung zur nächsten gültigen Stellung. Damit wird die auf fehlstehende Körperteile wirkende, korrigierende Kraft stärker mit größeren Fehlstellungen. Gleichung 5.20 beschreibt die Berechnung des Basisgewichts g_{Basis}, wobei w_a für die aktuelle Stellung (Winkel) des Körperteils, und w_g für den verletzten Grenzwinkel steht. Ein graphische Darstellung der verschiedenen Parameter wird in Abb. 5.11 gezeigt, die aus dem Gewicht resultiertende Kraft auf das fehlgestellte Körperteil ist als orangener Pfeil dargestellt, der vom Ende des fehlgestellten Körperteils zur nächsten gültigen Stellung weist.

$$g_{Basis} = |w_a - w_g| \qquad [5.20]$$

Aus praktischen Gründen wird der Wertebereich für g ist begrenzt, das minimale Gewicht ist 1.0, das maximale erlaubte Gewicht ist definiert als Anteil ρ_G der Summe aller Gewichte der realen Messungen zugehörig zu Körperteil K, die vollständige Berechnung ist in Gleichung 5.21 gegeben. Dabei steht $S_{G,K}$ für die Gesamtsumme der Gewichte aller Messungen von Körperteil K.

$$g = \begin{cases} 1.0 & \text{für} \quad g_{Basis} \leq 1.0 \\ g_{Basis} & \text{für} \quad 1.0 < g_{Basis} < S_{G,K} \\ \rho_G \cdot S_{G,K} & \text{für} \quad S_{G,K} \leq g_{Basis} \end{cases} \qquad [5.21]$$

Ein Wert von $\rho_G = 0,75$ zeigt in der Praxis gute Resultate, da hier den realen Messungen noch genug Gewicht eingeräumt wird. Wenn mehr als eine künstliche Messung für einen Grenzwinkel eingefügt wird, wird das mittels Gleichung 5.21 berechnete Gewicht homogen zwischen den künstlichen Messungen aufgeteilt.

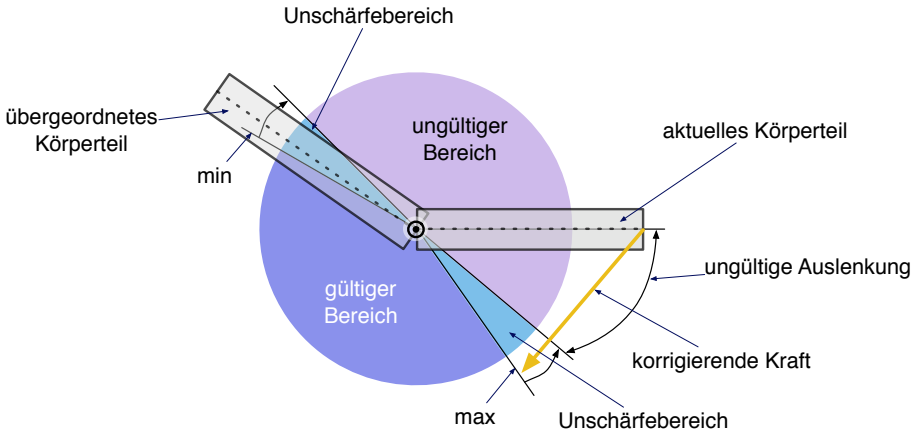


Abb. 5.11.: Parameter bei der Modellierung von Gelenkwinkelgrenzen: Erlaubter (gültiger) Bewegungsbereich (blau), Unschärfebereich (cyan), Bereich ungültiger Stellungen (violett), resultierende Kraft auf fehlgestelltem Körperteil (orangener Pfeil).

Unschärfe-Parameter

Zwei Einflüsse führen bei der Verwendung der Gelenkwinkel-Grenzen wie oben beschrieben zu einer hohen Anzahl von Fehlstellungen. Die beiden Einflüsse sind Rauschen in den Sensordaten und die Tatsache, dass es mehrere ICP-Schritte dauern kann, bis eine Fehlstellung vollständig korrigiert ist. Um den Effekt dieser Faktoren zu verringern, wurde ein zusätzlicher Parameter β_G eingeführt, der eine gewisse Unschärfe (engl. *blur*) für die Gelenkwinkel-Grenzen erlaubt. Er ermöglicht eine globale Anpassung der Gelenkwinkel-Grenzen, sodass kleine Abweichungen in den eigentlich ungültigen Winkelbereich toleriert werden. Abb. 5.11 zeigt den Unschärfebereich als blaue Erweiterung des gültigen Bewegungsbereichs. Eine systematische Evaluation des Effekts verschiedener Werte für β_G (Details in Abschnitt 7.1.2) zeigt, dass schon ein kleiner Wert von 5° genügt um eine signifikante Reduktion der Fehlstellungen während des Trackings zu erreichen.

5.3.4. Anatomische Bewegungsgrenzen beim Menschen

In der anatomischen Fachliteratur wird zwischen einer großen Menge unterschiedlicher Gelenke unterschieden, sowohl Drehgelenke (beispielsweise Kugelgelenke *articulation spheroida* und Scharniergelenke *articulatio ginglymus*) als auch Schubgelenke (beispielsweise Schlittenglenke *articulation delabens*) existieren. Jeder dieser Typen hat eigene Bewegungsbeschränkungen, aber für die verwendete Modellierung genügt die Betrachtung der Gelenke in Hals, Schultern, Ellbogen, Hüfte und Knie. Aus anatomischer Sicht sind die Bewegungen der zugehörigen Körperteile meist nicht das Ergebnis eines einzelnen Gelenkes, sondern der vereinigten Bewegung mehrerer Gelenke, siehe auch [Kapandji et al., 2007; Kapandji and Kandel, 1988; Kapandji and Honoré, 2008] für Hintergrundinformationen. Abb. 5.12 zeigt als Beispiel die resultierenden Bewegungsgrenzen aus der Kombination der verschiedenen Gelenke im menschlichen Hals.

Menschliche Bewegungen werden durch verschiedene Mechanismen beschränkt, typischerweise entweder durch Knochen (in Form von Anschlägen) oder durch Sehnen bzw. Muskeln. Das Problem der Beschreibung solcher Grenzen wird verkompliziert durch die Tatsache, dass einige Gelenkwinkel-Grenzen von der Stellung des jeweiligen Eltern-Körperteils abhängen, beispielsweise hängt die mögliche Rotation des Unterarms ab von der Position des Oberarms, wie in Abb. 5.13 dargestellt.

Durch Kombination räumlich eng beieinander liegender Gelenke in jeweils ein „aggregiertes“ Gelenk erhält man die in VooDoo-Modellen genutzten Gelenktypen:

- *Kugelgelenk*: Hüftgelenke (zwischen Oberkörper und Oberschenkeln) und Schultergelenke sowie das Halsgelenk können als Kugelgelenke modelliert werden.
- *Scharniergelenk*: Knie und Ellbogen werden als Scharniergelenke modelliert.

Unter Beachtung dieser Vereinfachung können die benötigten Gelenke mit den drei Gelenktypen aus Abschnitt 5.3.1 modelliert werden. Die dafür benötigten Parameter sind in Tab. 5.2 aufgeführt.

Tab. 5.2.: Tabelle der für die Parametrisierung der Gelenkwinkelbegrenzungen genutzten Parameter, nach Körperteilen geordnet.

Körperteil	**Bewegung**	**Untere Grenze**	**Obere Grenze**
Kopf	Laterale Flexion	$[\ -35°\ ,\ +35°\ ]$	
	Extension / Flexion	$[\ -40°\ ,\ +65°\ ]$	
	Rotation	$[\ -50°\ ,\ +50°\ ]$	
Schulter	Flexion / Extension aus Grundposition	$[\ -40°\ ,\ +170°\ ]$	
	Flexion / Extension in 90°-Position (gestreckter Arm)	$[\ -50°\ ,\ +160°\ ]$	
	Abduction / Adduction	$[\ -40°\ ,\ +180°\ ]$	
	Innere / äußere Rotation	$[\ -70°\ ,\ +60°\ ]$	
	Innere / äußere Rotation in 90°-Position	$[\ -90°\ ,\ +70°\ ]$	
Ellbogen	Flexion / Extension	$[\ -10°\ ,\ +150°\ ]$	
	Pronation / Supination	$[\ -90°\ ,\ +90°\ ]$	
Hüfte	Abduction / Adduction (gestreckte Hüfte)	$[\ -30°\ ,\ +50°\ ]$	
	Abduction / Adduction (gebeugte Hüfte)	$[\ -20°\ ,\ +80°\ ]$	
	Äußere / innere Rotation (gestreckte Hüfte)	$[\ -30°\ ,\ +40°\ ]$	
	Äußere / innere Rotation (gebeugte Hüfte)	$[\ -50°\ ,\ +40°\ ]$	
	Extension / Flexion	$[\ -20°\ ,\ +140°\ ]$	
Knie	Flexion / Extension	$[\ -150°\ ,\ +10°\ ]$	
	Innere / äußere Rotation	$[\ -40°\ ,\ +10°\ ]$	

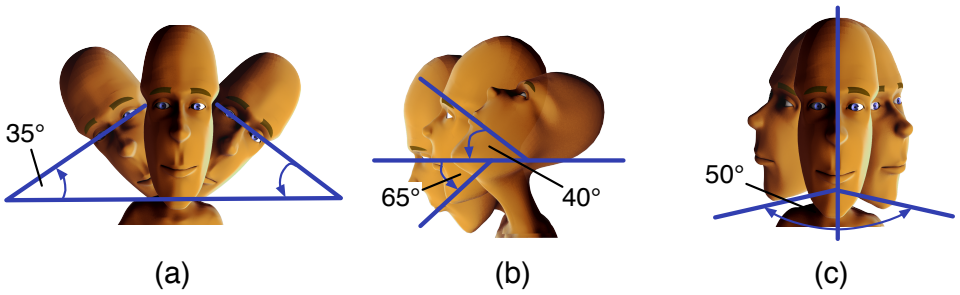


Abb. 5.12.: Bewegungsraum des menschlichen Halses: (a) Lateral flexion (b) Extension/Flexion (c) Rotation. Abbildung nach [Schünke et al., 2007].

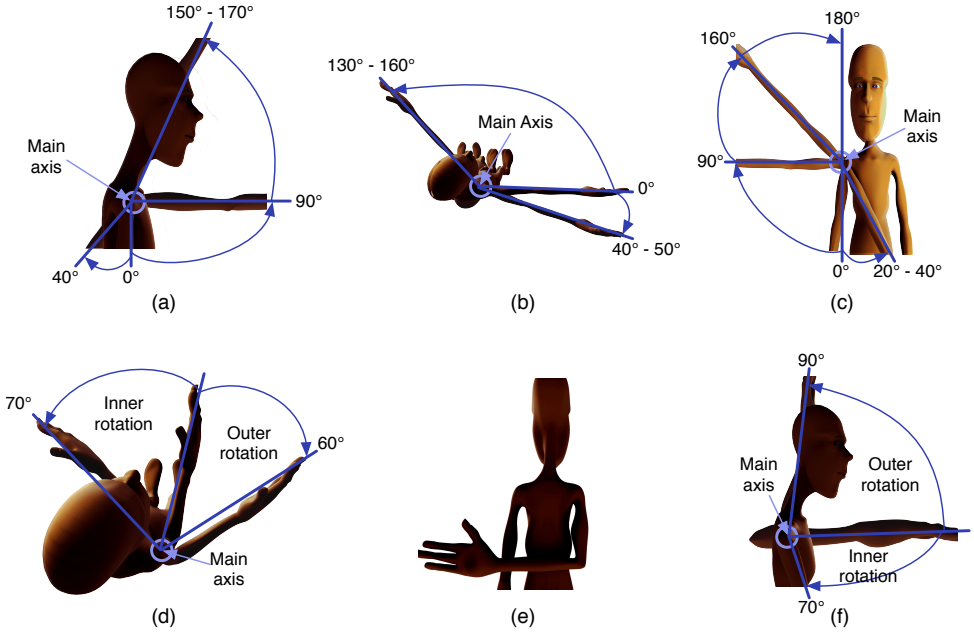
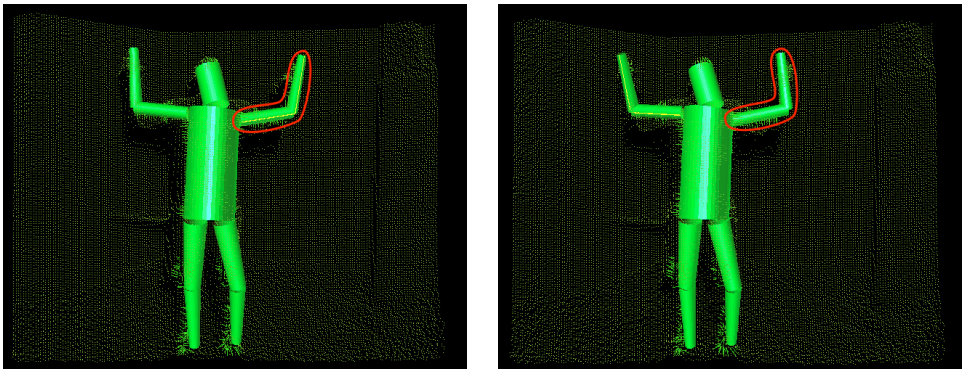


Abb. 5.13.: Bewegungsraum der menschlichen Schulter: (a) Flexion/Extension (b) Flexion/Extension in 90°-Position (c) Abduction/Adduction (d) Innere/Äußere Rotation (e) Innere Rotation 95° (f) Innere/Äußere Rotation in 90°-Position. Abbildung nach [Schünke et al., 2007].

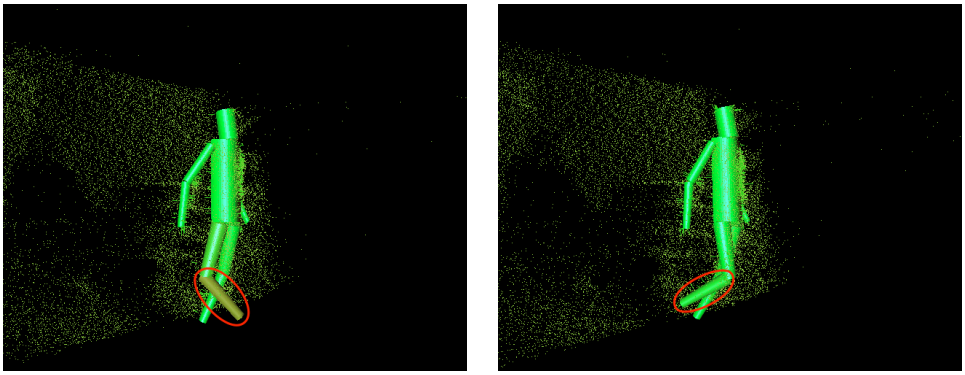
5.3.5. Bewertung der Gelenkwinkelgrenzen-Modellierung

Die Nutzung der Gelenkwinkelgrenzen im *VooDoo*-Tracking liefert substantiell bessere Ergebnisse, mit einer Verbesserung der summierten Fehlstellungen um einen Faktor von rund $4,5$. In Abb. 5.14 werden zwei typische Beispiele für die erreichte Verbesserung durch den Einsatz der Gelenkwinkelgrenzen gezeigt. Die Bilderpaare zeigen jeweils das Trackingergebnis auf den gleichen Sensordaten, links ohne und rechts mit Einsatz der Gelenkwinkelgrenzen. Die Bilder in Abb. 5.14(a) zeigen die Korrektur einer fehlerhaften Rotation des Oberarms, Abb. 5.14(b) die

Korrektur einer Beugung des linken Unterschenkels in die falsche Richtung. Weitere Details zur Evaluation der Gelenkwinkelgrenzen werden in Kapitel 7.1.2 diskutiert.



(a)



(b)

Abb. 5.14.: Zwei Beispiele für die unterschiedlichen Ergebnisse des Trackings ohne und mit Verwendung von Gelenkwinkelgrenzen. Das linke Bild in jedem der Bilderpaare zeigt das Tracking ohne Verwendung der Gelenkwinkelgrenzen, das rechte Bild mit den Gelenkwinkelgrenzen. (a) Links: Der linke Oberarm ist um 180° verdreht, was in einer Beugung des Unterarms „nach hinten“ resultiert. Rechts: Korrekte Stellung. (b) Links: Der rechte Unterschenkel ist fehlerhafterweise „nach vorne“ gebeugt. Rechts: Korrekte Stellung bei Verwendung der Gelenkwinkelgrenzen.

6. Interpretation & Klassifikation von menschlichen Bewegungen auf Basis von Trackingsequenzen und Modellwissen

Aufbauend auf der Personenbeobachtung wird die eigentliche Interpretation der beobachteten Bewegungen zur Erkennung von Aktivititäen vorgenommen, die den Schwerpunkt der vorliegenden Arbeit darstellt. In diesem Kapitel werden die Arbeiten zur Lösung dieser Aufgabe dargestellt und in das Gesamtkonzept eingeordnet. Das Kapitel gliedert sich wie folgt. In Abschnitt 6.1 wird zunächst das in Kapitel 4 vorgestellte Lösungskonzept in eine konkrete Architektur instantiiert. Die Darstellung der jeweiligen Details folgen in den nächsten Abschnitten, gegliedert nach den Hauptteilen des Konzeptes: In Abschnitt 6.2 wird die Repräsentation und Extraktion von Merkmalen dargestellt, im folgenden Abschnitt die Auswahl einer Teilmenge von relevanten Merkmalen für spezifische Aktivitäten. Schließlich beschreibt Abschnitt 6.4 wie mittels der gewählten Merkmale Aktivitäten erkannt werden können.

6.1. Überblick und Architektur

Die Erkennung von Aktivitäten baut den in der *Datenakquisition* gewonnenen Daten auf, in die auch die in Kapitel 5 beschriebenen Verbesserungen mit einfließen. In diesem Schritt werden Personen und Teile der Umwelt beobachtet und diese Perzeptionen in einen Datenstrom fusioniert, der als Eingabe für die folgende Abstraktion in der *Merkmalsextraktion* dient.

Abbildung 6.1 zeigt noch einmal die konzeptuelle, erweiterte Prozesskette, die in Kapitel 4 eingeführt wurde. Im folgenden werden die Details der drei auf die *Datenakquisition* folgenden Komponenten dargestellt. In der *Merkmalsextraktion* werden die Sensordaten abstrahiert zu einer Kette von Symbolen, die unabhänging von den verwendeten Sensoren sind. Die wichtige Frage dabei ist es, welche Merkmale aus den Sensordaten berechnet werden sollen. Die mit der Lösung dieser Frage Beziehung stehenden Forschungen werden in Abschnitt 6.2 in drei Teilen präsentiert. In Abschnitt 6.2.1 wird die entwickelte, sensor-unabhängige Repräsentation beschrieben, in Abschnitt 6.2.2 die Extraktion der Merkmale aus Sensordaten, und in Abschnitt 6.2.3 ein darauf aufbauender Ansatz zur automatischen Exploration eines Merkmalsraums für unbekannte Domänen. Aus den resultierenden hochdimensionalen Symbolketten

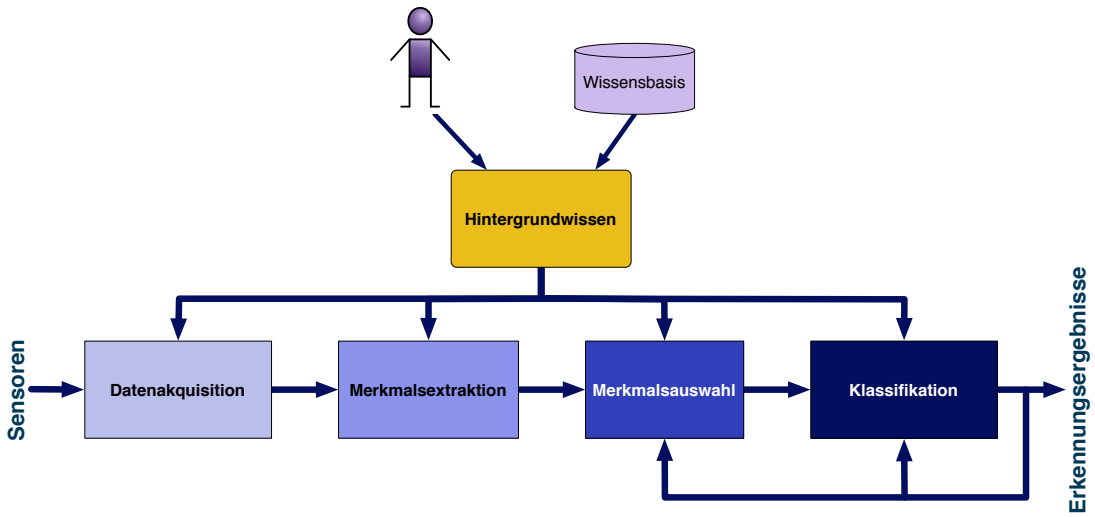


Abb. 6.1.: Erweiterte abstrakte Prozesskette für die Erkennung von Aktivitäten. Hintergrundwissen, das entweder aus Interaktion direkt mit dem Benutzer oder aus einer Wissensbasis stammt, wird als zusätzliche Informationsquellen genutzt, darüberhinaus gibt es eine Rückführung der Klassifikationsergebnisse in die Merkmalsauswahl und die Klassifikation.

wird in der *Merkmalsauswahl* eine Teilmenge relevanter Merkmale ausgewählt für die Erkennung spezifischer Aktivitäten. Hierzu wurde ein Ansatz entwickelt, der aktive (auf mathematischen Relevanzmaßen), interaktive (durch vom Benutzer in direkter Interaktion bereitgestelltes Hintergrundwissen) und passive (aus in einer Wissensbasis abgelegtem Hintergrundwissen) Herangehensweisen verbindet. Details dieses Ansatzes werden in Abschnitt 6.3 beschrieben. Nach der Auswahl der relevanten Merkmale werden in der *Klassifikation* Erkenner für spezifische Aktivitäten trainiert bzw. für die Erkennung genutzt. Das allgemeine Vorgehen für Training und Erkennung wird in Abschnitt 6.4.1 beschrieben, mit Details zur mehrschichtigen Erkennung in Abschnitt 6.4.2, Erklärungen zu den verwendeten Erkennern in den Abschnitten 6.4.3 und 6.4.4, sowie der Ergebnis-Nachbehandlung in Abschnitt 6.4.5. Der Einbindung von Hintergrundwissen in den Trainings- und Klassifikationsprozess ist Abschnitt 6.4.6 gewidmet.

Zur Realisierung der Erkennung wurde diese abstrakte Prozesskette in drei konkrete, aufeinander aufbauende Teilprozessketten weiter aufgeteilt, die jeweils zur Lösung einzelner Problemaspekte zum Einsatz kommen. Die in Abb. 6.2 gezeigte Prozesskette zeigt die Komponenten für die Erschließung neuer Anwendungsdomänen. Ausgehend von Trainingsdaten werden hier mittels verschiedener Merkmalsextraktor-Komponenten sehr viele Merkmale berechnet zur Exploration des Merkmalsraums der Domäne. Mit Hilfe der Trainingsdaten wird dann eine Suche nach für diese Domäne interessanten Merkmalen durchgeführt, deren Resultat persistent gespeichert und in der Folge für alle in dieser Domäne zu lernenden und zu erkennenden Aktivitäten genutzt wird.

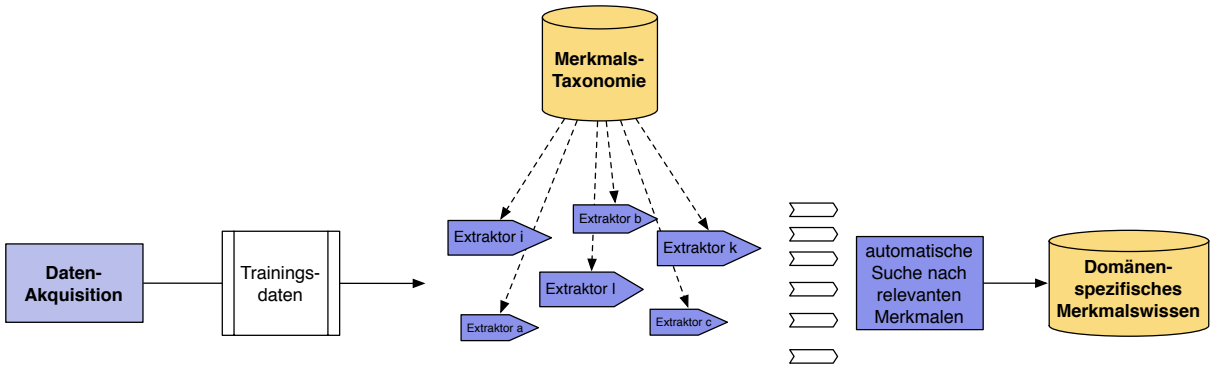


Abb. 6.2.: Darstellung der Prozesskette zur Exploration relevanter Merkmale einer gegeben Domäne, in der Trainingsdaten vorliegen. Nach Abschluss der Suche werden die Merkmale in einem Speicher für Domänen-spezifisches Merkmalswissen abgelegt (Wissensbasen sind orange markiert, die wichtigen Teilschritte, die im Haupttext näher erläutert werden, in grün).

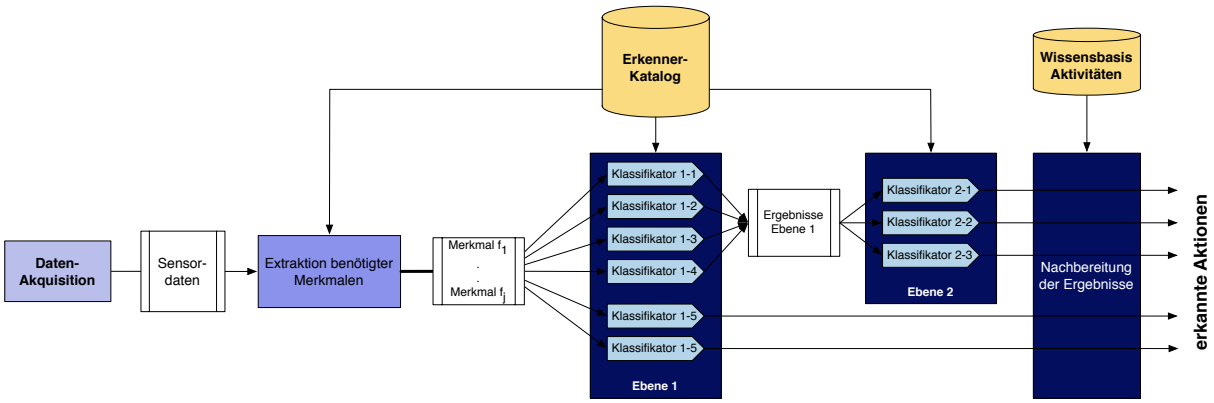


Abb. 6.3.: Darstellung der Prozesskette zur Erkennung und Interpretation von menschlichen Bewegungen. Aus Sensordaten werden die für genutzte Erkenner benötigten Merkmale extrahiert und als Eingabe an die Klassifikatoren übergeben. In einem Nachbehandlungsschritt werden deren Resultate aufbereitet (Wissensbasen sind orange markiert, die wichtigen Teilschritte, die im Haupttext näher erläutert werden, in grün).

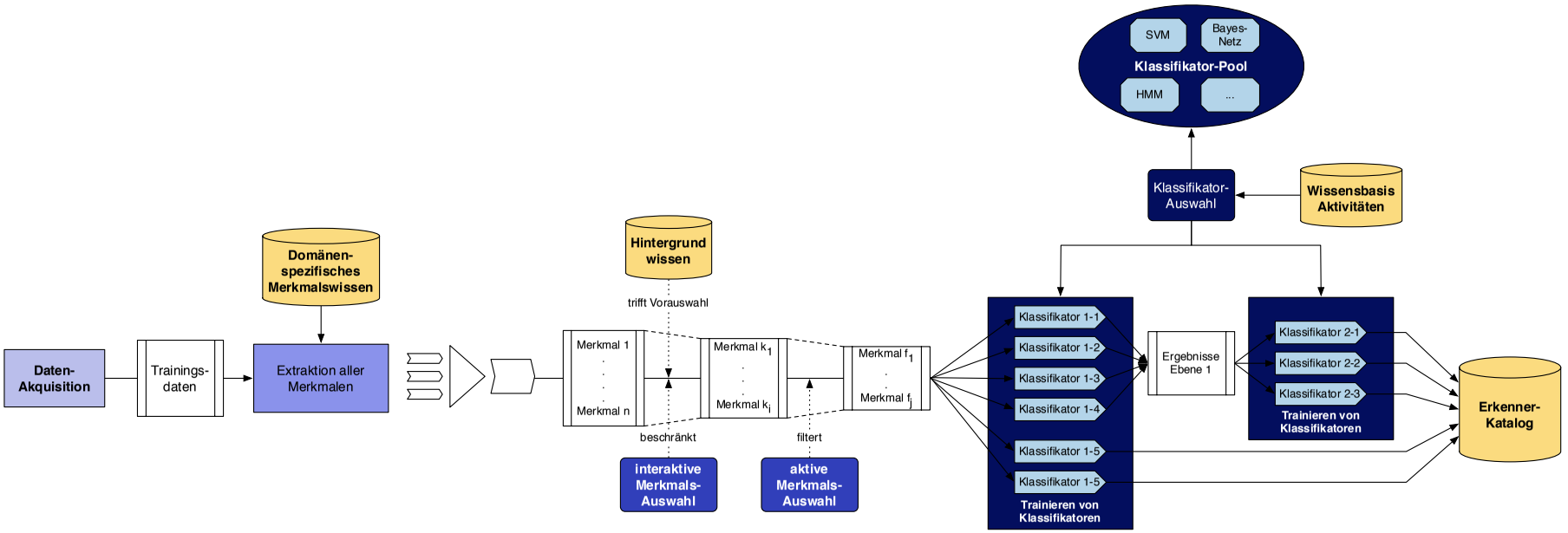


Abb. 6.4.: Darstellung der Prozesskette zum Einlernen einer neuen Aktivität. Unter Nutzung der Domänen-spezifischen Merkmale werden zunächst die Merkmale der Trainingsdaten extrahiert, anschließend werden die für die Aktivität relevanten Merkmale ausgewählt. Mit diesen Merkmalen werden schließlich Klassifikatoren zur Erkennung der Aktivität trainiert (entweder direkt oder mit einer Indirektion über die Erkennung von Grundbewegungen in Ebene 1), und trainierte Klassifikatoren in einem persisten Erkenner-Katalog abgelegt (Wissensbasen sind orange markiert, die wichtigen Teilschritte, die im Haupttext näher erläutert werden, in grün).

Zum Erlernen neuer Aktivitäten wird die in Abb. 6.4 gezeigt Prozesskette eingesetzt. Aus aufgezeichneten Trainingsdaten werden die in der vorigen Prozesskette bestimmten, Domänenspezifischen Merkmale extrahiert. In der folgenden Merkmalsauswahl wird der ursprünglich hochdimensionale Datenstrom auf die relevanten Merkmale beschränkt. Mit diesen Merkmale werden schließlich entschieden, abhängig von der zu lernenden Aktivität und ausgewählt unter Einsatz von zusätzlichem Hintergrundwissen über menschliche Aktivitäten, ob eine direkte Erkennung der Aktivität möglich ist, oder ob eine Indirektion durch die Verwendung zweier Erkennungsebenen notwendig ist. In letzterem Fall werden auf der ersten Ebene Grundbewegungen erkannt, die auf der zweiten Ebene zur Erkennung der eigentlichen Aktivität genutzt werden. Trainierte Erkenner werden in einem Erkenner-Katalog für die spätere Verwendung gespeichert.

Für den Einsatz der Erkennung in Anwendungen wird die in Abb. 6.3 dargestellte Prozesskette genutzt. Abhängig von der Verwendung werden die benötigten Erkenner aus dem Erkenner-Katalog ausgewählt. Aus Sensordaten werden die für diese Erkenner benötigten Merkmale extrahiert, und als Eingabe an die trainierten Klassifikatoren übergeben. Die Resultate werden vor der Rückgabe noch einem Nachbehandlungsschritt unterzogen, der durch eventuell vorhandenes Wissen über das Verhältnis der erkannten Aktivitäten zueinander noch eine Korrektur der Erkennungsresultate vornimmt.

6.2. Merkmale

In diesem Abschnitt werden die für die Erkennung verwendeten Merkmale und die damit in Zusammenhang stehenden Forschungsarbeiten beschrieben. Abschnitt 6.2.1 beschreibt die eingesetzte Repräsentation für Merkmale, Abschnitt 6.2.2 die Gewinnung der Merkmale aus Sensordaten. Abschließend diskutiert Abschnitt 6.2.3 die automatische Exploration einer initialen Merkmalsmenge für bisher unbekannte Domänen.

6.2.1. Repräsentation von Merkmalen

In verschiedenen Anwendungen werden sehr unterschiedliche Merkmale eingesetzt, die Bandbreite reicht von nur rohen oder nur wenig gefilterten Messwerten bis hin zu komplex berechneten Merkmalen wie etwa SIFT-Merkmalen ([Lowe, 1999]) in der Bildverarbeitung. Für das in dieser Arbeit entwickelte System werden allerdings spezielle Anforderungen an die Repräsentation gestellt, die im nächsten Abschnitt diskutiert werden. Die zur Lösung entwickelte Repräsentation basiert aus sogenannten *Merkmalsextraktions-Modulen (MEMs)* und wird anschließend beschrieben.

Anforderungen an Merkmalsrepräsentation

Die vorliegende Arbeit zielt auf die Anwendung in Domänen, in denen im vorhinein nicht alle interessanten Klassen (Aktivitäten) und für die Unterscheidung nützliche Merkmale bekannt sind. Daher wird hier eine generische Repräsentation für Merkmale benötigt, die die Berechnung und eindeutige Identifikation beliebig komplexer Merkmale erlaubt, und damit auch die in den folgenden Abschnitten präsentierte automatische Merkmalsexploration erlaubt. Es gibt drei Anforderungen, die eine solche Repräsentation erfüllen soll:

(I) **Erweiterbarkeit für neue Daten**: Benötigt, um auf die Verwendung zusätzlicher Sensoren, detaillierterer Modelle, neuer Messungen o.ä. zu reagieren.

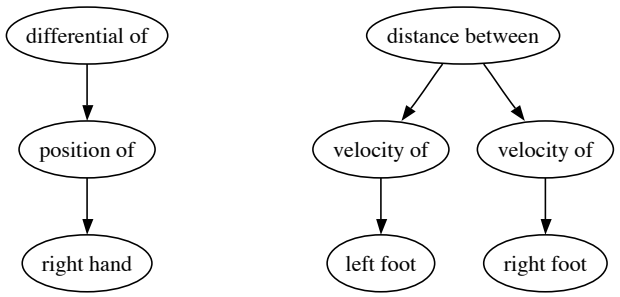
(II) **Erweiterbarkeit für neuartige Extraktionsmethoden**: Nötig, um auch neuartige Analyseverfahren einfach in das Gesamtsystem zu integrieren.

(III) **Symbolische Interpretierbarkeit**: Nötig, um die Nutzung symbolischer Inferenzmethoden und die Interpretation und Kontrolle der Merkmale durch den menschlichen Benutzer zu erlauben.

Diese Anforderungen können von sonst üblichen, handkodierten Merkmalen nicht erfüllt werden. Stattdessen wurde eine Repräsentation entwickelt, die die Methode zur Berechnung eines Merkmals direkt in dessen Repräsentation kodiert.

Merkmale als Baum-Struktur

Zur Verwirklichung der Anforderungen (I) – (III) wurde eine Repräsentation entwickelt, die Merkmale nicht nur durch einen Namen beschreibt (und eine davon getrennte Berechnung in Programmcode), sondern die Repräsentation selbst beschreibt die Art und Weise, wie das Merkmal berechnet werden kann. Die Kernidee dabei ist die Verwendung von Operator-ähnlichen *Merkmalsextraktor-Modulen (MEMs)* (engl. *Feature extractor modules (FEMs)*), die jeweils eine fest definierte, meist mathematische, Operation auf ihren Eingangsdaten durchführen und eindeutig durch ihren Namen identifiziert werden. Mit MEMs werden beliebige Merkmale als systematische Strukturen konstruiert, die als Bäume interpretiert werden können, wenn man den äußersten MEM einer geschachtelten Struktur als Wurzel interpretiert. Zwei Beispiele werden sind in Abb. 6.5 gezeigt. Der linke Baum stellt die graphische Repräsentation des Merkmals `[differential of [position of [right hand]]]`, die Geschwindigkeit der rechten Hand, dar. Der rechte Baum repräsentiert `[distance between [velocity of [left foot]] [velocity of [right foot]]]`, den Unterschied in der Geschwindigkeit des linken und des rechten Fußes.



(a) (b)

Abb. 6.5.: Einfache Beispiele für die Interpretation der neuen Merkmalsrepräsentation als Baum: (a) Geschwindigkeit der rechten Hand. (b) Geschwindigkeitsunterschied zwischen den Bewegungen von linkem Fuß und rechtem Fuß.

Die Repräsentation zielt hauptsächlich auf die Behandlung numerischer Merkmale (für die die in Abschnitt 6.2.2 präsentierten MEMs anwendbar sind), aber auch andere Merkmalsarten können durch die Definition entsprechend kompatibler MEMs behandelt werden, beispielsweise nominale Merkmale wie Objektzustände.

Bewertung der Merkmals-Repräsentation

Die beschriebene Repräsentation für Merkmale erfüllt Anforderung (III) durch die symbolische Schreibweise und die Darstellung als Baumstrukturen. Durch den systematischen Aufbau aus Symbolen kann die Darstellung von Menschen kontrolliert und verstanden werden, solange verständliche Bezeichnungen für die einzelnen MEMs gewählt werden.

Die Anforderungen (I) und (II) werden durch die Modularisierung der Berechnungen mittels MEMs ebenfalls erfüllt. Das Integrieren neuer Daten wird durch die Definition eines MEM ermöglicht, der als innerstes Datenelement (entsprechend als Blatt des zugehörigen Baumes) dient. Ebenso können neue Analysemethoden als zusätzlicher MEM integriert werden. Durch die gemeinsame Struktur aller MEMs sind diese neuen Module sofort kompatibel mit den existierenden Modulen und verwendbar zur Definition neuer Merkmale.

Die Repräsentation ist theoretisch nicht beschränkt bezüglich der modellierbaren Merkmale. In der Praxis hat es sich allerdings als nützlich erwiesen (siehe dazu auch Abschnitt 6.2.2) eine Typisierung vorzunehmen, um Einschränkungen hinsichtlich der Kompatibilität zwischen verschiedenen Daten bezüglich einer Operation einzuführen. Durch die Definition entsprechender Einschränkungen werden „Merkmale“ wie etwa die Differenz zwischen einer Geschwindigkeit und einem Winkel verhindert, die logisch betrachtet keine Bedeutung haben können.

Durch die offene Struktur ergibt sich darüberhinaus eine weitere interessante Eigenschaft. Auch neue, unbekannte Merkmale können automatisch generiert werden, indem eine Kombination von MEMs gebildet wird. Das ist die Grundlage für den in Abschnitt 6.2.3 entwickelten *automatischen Merkmals-Explorationsprozess*, der die schnelle Erschließung neuer Domänen und Sensorumgebungen erlaubt.

6.2.2. Extraktion von Merkmalen

Die Merkmalsrepräsentation beruht auf der Verwendung von *Merkmalsextraktoren (MEMs)*, die als atomare Berechungseinheiten betrachtet werden können. Der folgende Abschnitt definiert das MEM-Konzept, und erklärt die Unterscheidung in verschiedene MEM-Typen. Anschließend wird beschrieben, wie sie zur Berechnung von Merkmalen kombiniert werden können, und schließlich werden die implementierten und genutzten MEMs vorgestellt und diskutiert.

Definition *Merkmalsextraktor*

Ein *Merkmalsextraktor-Modul (MEM)* wird nach außen definiert durch einen eindeutigen Namen und eine fixe Anzahl von Eingabeslots. Ein MEM liefert ein eindeutiges Resultat und ist semantisch durch die Operation definiert, die er durchführt. Die Eingabeslots können typisiert werden, mit grundlegenden Standard-Datentypen (nominale Daten, Zahlen, etc.); eine weitere Typisierung mit konkreteren Typen wie Geschwindigkeiten, Winkel etc. ist auf abstrakterer Ebene möglich, wie im Folgenden noch beschrieben wird. Aus dem Zusammenspiel der Kombination von Eingabetypen und Operation hat auch das Resultat einen Typ.

Konzeptuell werden aus praktischen Gründen zwei Typen von MEMs unterschieden. Für die Gewinnung initialer Merkmale werden formale MEMs definiert (im Folgenden als *initiale Merkmalsextraktoren (*i*MEMs)* bezeichnet), die keine Operation auf Eingabedaten durchführen, sondern nur Rohdaten entgegennehmen und die Daten selbst oder (bei mehrdimensionalen Daten) oder Teile davon als Merkmale in einem zu MEMs kompatiblen Format für weitere Berechnungen zur Verfügung stellen, um eine einheitliche Behandlung zu erlauben. Beispiele für von iMEMs gewinnbare Merkmale sind Positionen, Winkel oder auch direkt Geschwindigkeiten. Definierendes Merkmal der iMEMs ist es, dass als Eingabe nur sensor-spezifische Datentypen verwendet werden können. Aufbauend auf den von *i*MEMs zur Verfügung gestellten Daten können *komplexe MEMs (*k*MEMs)* eingesetzt werden, die aus den noch rohen Daten der *i*MEMs einfache und komplexe Merkmale extrahieren können, beispielsweise Beschleunigungen oder Periodizitäten. Kennzeichnend für *k*MEMs ist die mehrfache Anwendbarkeit auch innerhalb eines Pfades von Blatt zu Wurzel im Merkmalsbaum.

Wie schon oben erwähnt, zielt das MEM-Konzept zwar stark auf die Berechnung und Analyse numerischer Merkmale, für die Operationen als die Berechnung mathematischer Funktionen definiert werden können, aber prinzipiell steht auch der Verwendung von nominalen Merkmalen nichts im Wege, solange entsprechende MEMs definiert werden.

Verwendung und Bezeichnungszuweisung

Zur Repräsentation eines Merkmals werden ein oder mehrere MEMs kombiniert in der in Abschnitt 6.2.1 beschriebenen Weise. Unter Einbeziehung des detaillierteren Konzeptes der zwei MEM-Typen *i*MEM und *k*MEM kann die Struktur eines ein Merkmal beschreibenden Baums genauer spezifiert werden. Die Blätter eines Baumes sind (ein oder mehrere) *i*MEM, aus denen durch die Anwendung eines oder mehrerer *k*MEMs das endgültige Merkmal berechnet wird.

Durch die Freiheiten bei der Definition von MEMs können verschiedene Beschreibungen für das gleiche Merkmal existieren, nicht nur durch eine unterschiedliche Reihenfolge der Anwendung von kommutativen MEMs. MEMs können zur Durchführung von Berechnungen unterschiedlicher Abstraktion definiert werden, die je nach Problemdomäne unterschiedlich gut geeignet sind. Als Beispiel für diesen Sachverhalt zeigt Abb. 6.6 zwei Möglichkeiten zur Berechnung von Polarkoordinaten, unter Verwendung verschiedener MEMs (dabei stehen die mit x bzw. y bezeichneten Knoten als Platzhalter für beliebige Positionen).

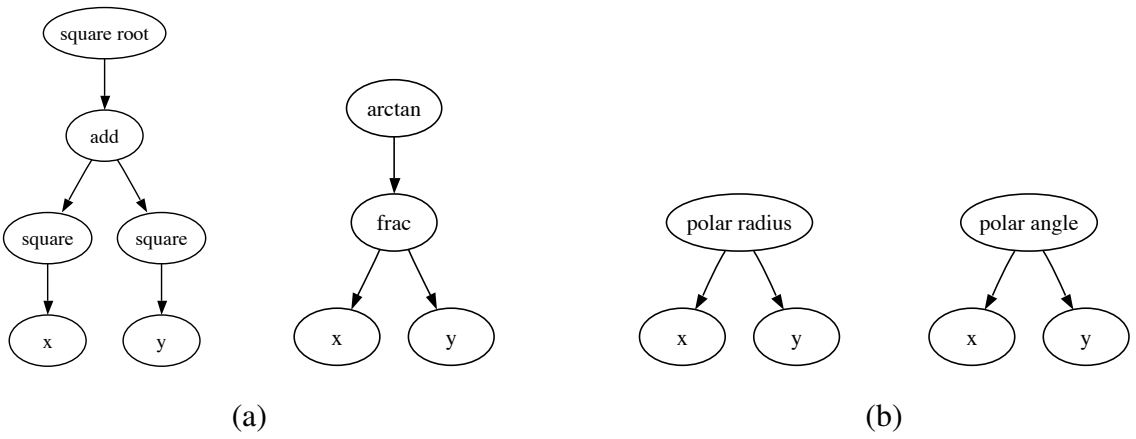


Abb. 6.6.: Beispiele für unterschiedliche Repräsentation von Polarkoordinaten als MEM-basierte Bäume: (a) Verwendung einfacher MEMs. (b) Verwendung komplexer MEMs.

Aus den Bezeichnungen der einzelnen MEMs kann eine einfache Bezeichnung für das resultierende Merkmal zusammengesetzt werden, die direkt auf der Berechnung des Merkmals basiert. In diesem Sinne ist die Bezeichnung aussagekräftig, und im Falle von Merkmalen, die nicht aus zu vielen MEMs zusammengesetzt sind, auch gut erfassbar für menschliche Benutzer.

Mit steigender Anzahl von MEMs in der Berechnung eines Merkmals wird die zugewiesene Bezeichung allerdings komplexer und damit schlechter verständlich. Darüberhinaus ist an dieser Stelle anzumerken, dass auf diese Art keine semantisch korrekte Bezeichnung gefunden werden kann (z.B. „Geschwindigkeit“ für ein Merkmal „Ableitung der Position“).

Existierende MEMs

Im Rahmen dieser Arbeit wurden MEMs für die Verwendung mit Ganzkörperbeobachtungen aus einem Trackingsystem und Feinbeobachtungen mittels Datenhandschuhen entwickelt. Alle MEMs halten nicht nur das aktuelle Ergebnis, sondern eine Zeitreihe der letzten Ergebnisse, um die Berechnung von weiter von ihnen abgeleitete Merkmalen zu ermöglichen. Die Länge der Zeitreihe beträgt etwa 3 Sekunden (was etwa 60 Frames bei einer Sensorgeschwindigkeit von 20 Hz entspricht), den Ergebnissen aus [Otero et al., 2006] über die übliche Dauer von einfachen Handlungen folgend.

iMEMs Als initiale MEMs wurden Module entwickelt, die im Bereich der Ganzkörperbeobachtung die Resultate von entsprechenden Trackingsystemen (siehe Abschnitt **??** für Informationen zu den verwendeten Systemen) für die weitere Verarbeitung bereitstellen können. Mit Eingabe der Modelle der Ganzkörperbeobachtung extrahieren die jeweiligen *i*MEMs absolute und relative Positionen und absolute und relative Winkel der einzelnen Körperteile, aus denen die Modelle zusammengesetzt sind. Aus der Eingabe der Modelle der Hand- und Fingerbeobachtungen werden Position und Lage der Hände sowie relative Winkel und Positionen der Fingergelenke extrahiert. Eine vollständige Tabelle der aus den implementierten iMEMs verfügbaren Merkmale sind in Anhang A.1 aufgelistet.

kMEMs Die implementierten *k*MEMs implementieren Sensor-unabhängige Operationen auf den Daten. Sowohl für die Untersuchung und Beschreibung von Merkmalen in der Ganzkörperbeobachtung als auch bei der Beobachtung von Objekthandhabungen kommen die gleichen kMEMs zum Einsatz.

Der Nutzen im Einsatz von *k*MEMs stammt aus der Anreicherung der Information in den resultierenden Merkmalen, indem die Informationen aus mehreren Merkmalswerten zusammengefasst werden. Die einzelnen Werte können dabei die Ausprägungen unterschiedlicher Merkmale zum gleichen Zeitpunkt (z.B. eine Differenz) oder auch die Ausprägungen eines Merkmals zu verschiedenen Zeitpunkten sein (z.B. ein Mittelwert). Anhang A.2 zeigt eine vollständige Liste der implementierten *k*MEMs.

6.2.3. Automatische Exploration von Merkmalen

Die bisher präsentierten Details zur Repräsentation von Merkmalen als Baumstruktur mit Hilfe von Merkmalsextraktormodulen bieten große Freiheiten zur Modellierung unterschiedlicher Merkmale, der Abstraktion von Sensordaten zu sensor-unabhängigen Merkmalen und zur einfachen Anbindung neuer Extraktoren. Allerdings erfordert auch diese Repräsentation immer noch eine manuelle Auswahl und Modellierung der zu nutzenden Merkmale. Es ist beispielsweise nicht möglich, das System ohne aufwändige, manuelle Modellierung in neuen Domänen einzusetzen.

Um dieses Problem zu lösen wäre daher ein automatischer *Merkmals-Explorationsprozess* wünschenswert. Solch ein Prozess erlaubt es, ausgehend von den verfügbaren MEMs für bisher unbekannte Domänen automatisch Merkmale zu generieren, und (durch Verwendung von Beispieldaten) auf ihre Relevanz für die Domäne zu prüfen. Die folgenden Abschnitte präsentieren den entwickelten Ansatz, der diese Eigenschaften aufweist.

Ablauf der Merkmals-Exploration

Der Ablauf der Merkmalsexploration ist in Abb. 6.7 skizziert, die die beiden aufeinanderfolgenden Hauptschritte der Exploration zeigt. Zuerst werden Merkmale generiert. Ausgehend von einer initialen Merkmalsliste (die die Merkmale enthält, die von *i*MEMs extrahiert werden können), werden ein oder mehrere Merkmale gewählt, um durch Anwendung eines MEMs ein neues Merkmal zu generieren, das das der Merkmalsliste hinzugefügt wird. Für die Auswahl und Generierung der Merkmale sind verschiedene Strategien anwendbar, die im folgenden Abschnitt im Detail diskutiert werden. Die Generierung wird fortgesetzt bis eine Abbruchbedingung erreicht ist. Die resultierende Merkmalsliste dient als Eingabe für den Evaluationsschritt. Eine Menge möglichst repräsentativer Trainingsdaten wird genutzt, um den Nutzen jedes Merkmals zur Unterscheidung zwischen verschiedenen Klassen zu bestimmen (siehe unten). Nach der Evaluation aller Merkmale werden irrelevante Merkmale (d.h. Merkmale, die in der Evaluation nur eine niedrige Bewertung erhalten haben) entfernt, sodass die resultierende Liste nur noch relevante Merkmale enthält.

Strategien zur Merkmals-Generierung

Zur Generierung neuer Merkmale sind verschiedene Strategien möglich. Da *k*MEMs mehrfach angewandt werden können, existiert in den meisten Fällen kein natürlicher Endpunkt für den Generierungsprozess. Stattdessen muss eine Abbruchbedingung gewählt werden, die einen Kompromiss zwischen benötigten Ressourcen (Zeit und Speicherplatz) auf der einen und einem

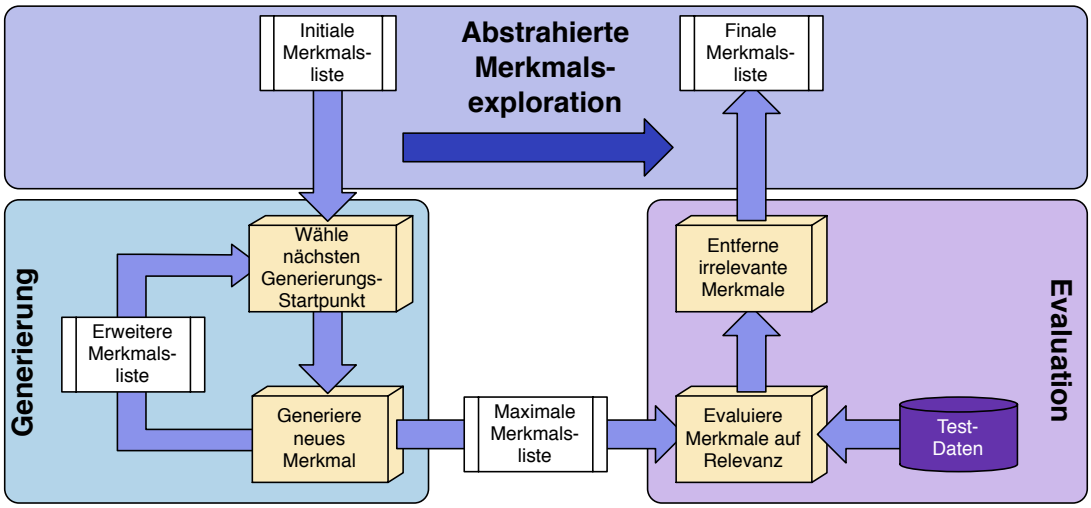


Abb. 6.7.: Diagramm zur Veranschaulichung der einzelnen Schritte der Merkmalsexploration. Beginnend mit einer initialen Merkmalsliste werden erst neue Merkmale generiert (im gelben Block links unten), die anschließend evaluiert und gegebenfalls als irrelevant befundene Merkmale wieder entfernt werden (grüner Block rechts unten).

größeren explorierten Merkmalsraum auf der anderen Seite darstellt. Da bei der Verwendung expliziter Abbruchbedingungen nicht mehr der vollständige mögliche Merkmalsraum generiert wird, muss zwischen verschiedenen Möglichkeiten, wie die Merkmale generiert werden, gewählt werden, da die Auswahl des nächsten Ausgangsmerkmals und die Auswahl des anzuwendenden MEMs in Verbindung mit der Abbruchbedingung starken Einfluss auf die resultierende Merkmalsmenge haben. Drei Strategien werden in dem System eingesetzt:

A Eine Strategie ist es, verschiedene (alle vorhanden) MEMs auf das gleiche Ausgangsmerkmal anzuwenden, und erst dann zu einem anderen Ausgangsmerkmal überzugehen. Ein Beispiel für diese Strategie ist in Abb. 6.8 dargestellt. Sie zeigt die Anwendung verschiedener Operationen auf die Position der rechten Hand, mit dem Resultat der Geschwindkeit, der mittleren Position etc.. Diese Strategie weist eine gewisse Analogie zu einer Breitensuche auf.

B Eine weitere Strategie folgt analog einer Tiefensuche dem Ansatz, wiederholt neue MEMs auf das Resultat des vorherigen Generierungsschrittes anzuwenden. Ein Beispiel für diese Strategie ist in Abb. 6.9 dargestellt. Sie zeigt die Erweiterung der Position der rechten Hand zu ihrer Geschwindigkeit und zu ihrer Beschleunigung.

C Eine dritte Strategie ist es, die Parameter des äußersten MEMs des Ausgangsmerkmals zu variieren. Dieser Ansatz ist interessant für mehrstellige Operationen, die nur schwierig

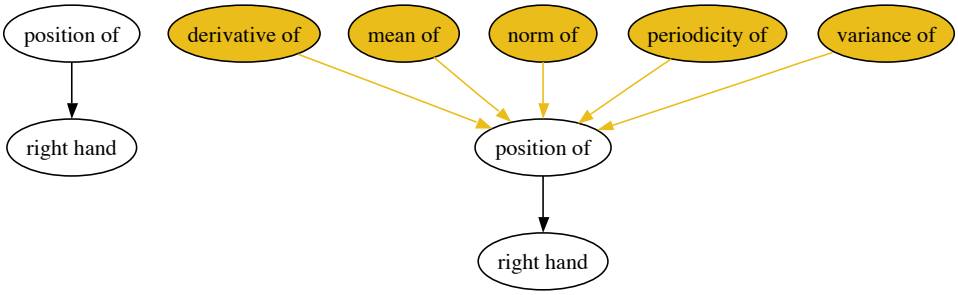


Abb. 6.8.: Beispiel für die Generierung neuer Merkmale durch die Erweiterung eines Merkmalsbaums mit verschiedenen neuen Wurzelknoten. Links: Ausgangsmerkmal. Rechts: Erweiterungen mittels verschiedener neuer Wurzeln (in rot).

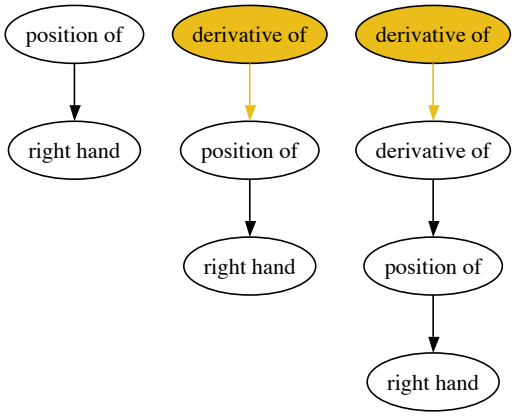


Abb. 6.9.: Beispiel für die Generierung neuer Merkmale durch die mehrfache Erweiterung einer Merkmalsbaums mit neuen Wurzelknoten. Von links nach rechts: Hinzufügen neuer Wurzel zum jeweils vorherigen Baum als Ausgangsmerkmal.

mit den vorherigen Strategien integriert werden können. Ein Beispiel für diese Strategie ist in Abb. 6.10 dargestellt. Sie zeigt die Variation des zweiten Parameters bei einem MEM, der als Operation die Berechnung der Distanz zwischen den Eingabeparametern durchführt. Die Variation im Beispiel ersetzt die Position der linken Hand mit der des Kopfes, der linken Schulter und so weiter.

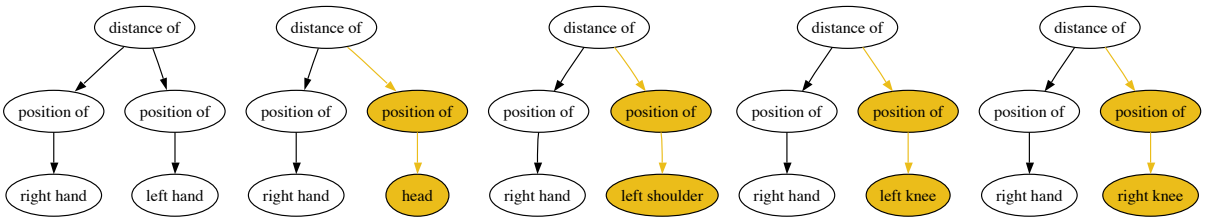


Abb. 6.10.: Beispiel für die Generierung neuer Merkmale durch die Variation der Parameter des Wurzelknotens. Von links nach rechts: Variation des zweiten Parameters eines Distanz-Operators.

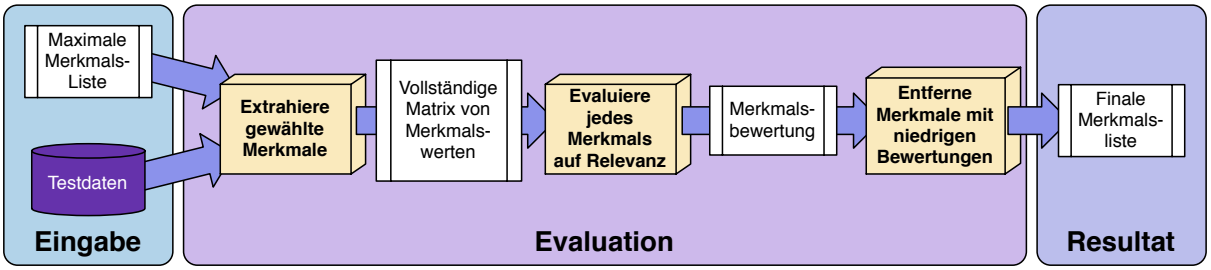


Abb. 6.11.: Diagramm der Prozessschritte in der Evaluation der automatisch generierten Merkmale (Detailansicht des grünen Blocks rechts unten in Abb. 6.7). Mit einer Liste aller Merkmale und einem Satz Testdaten werden alle Merkmale extrahiert und die jeweilige Relevanz bestimmt. Die resultierenden Bewertungen werden genutzt, um anhand eines Schwellwertes irrelevante Merkmale aus der Liste aller Merkmale zu entfernen.

Durch Kombinieren dieser Strategien können schon beliebig komplexe Merkmalsbeschreibungen generiert werden. In der Praxis treten aber noch zwei Probleme auf. Erstens hat der Generierungsprozess kein definiertes Ende, daher wird für reale Anwendungen eine explizite Abbruchbedingung benötigt. Zweitens werden unter den bisherigen Annahmen beliebige, auch offensichtlich bedeutungslose Merkmale generiert, beispielsweise könnte die Differenz aus einem Winkel und einer Position als Merkmal generiert werden. Als Lösung werden heuristische Strategien ergänzend zu den obigen Generierungsstrategien eingesetzt. Als Strategien für die Entscheidung über den Abbruch können die folgenden Ansätze zum Einsatz kommen:

MEM-Anzahl Die Anzahl der Extraktoren, aus denen ein einzelnes Merkmal aufgebaut wird, wird durch einen Schwellwert beschränkt.

Schachtelungstiefe Die maximale Tiefe der die Merkmale beschreibenden Bäume wird durch einen Schwellwert beschränkt.

Für die Steuerung der Generierung können darüberhinaus die folgenden Heuristiken genutzt werden:

Inkompatible Typen Merkmale des gleichen Wertebereichs (Ganzzahlen, reelle Zahle, ...) können trotzdem aus semantischer Sicht inkompatibel sein, beispielsweise Positionen (kartesische Koordinaten) und Winkel. Die Anwendung dieser Strategie stellt sicher, dass bei mehrstelligen MEMs nur semantisch kompatible Merkmale als Eingabe genutzt werden, um rechnerisch mögliche, aber aus semantischer Sicht unsinnige Merkmale (z.B. die Differenz aus einer Geschwindigkeit und einem Winkel) zu verhindern.

Äquivalente Bäume Diese Strategie vermeidet die Generierung mehrerer Instanzen des gleichen Merkmals aufgrund verschiedener äquivalenter Beschreibungen. In der Praxis tritt dieses Problem vor allem bei symmetrischen Operationen (z.B. Kovarianz) auf und bei Operator-Paaren, deren Anwendungsreihenfolge vertauscht werden kann(z.B. Ableitung einer Differenz / Differenz einer Ableitung).

Generierungsreihenfolge Über die Reihenfolge der Anwendung der drei grundlegenden Generierungsstrategien muss irgendwie entschieden werden. Hier sind unterschiedliche Lösungen möglich, deren Resultat unterschiedliche Reihenfolgen sind, in denen Merkmale generiert werden. Da andererseits eine vollständige, erschöpfende Exploration des Merkmalsraums im Allgemeinen nicht möglich ist, ist die Wahl der Generierungsreihenfolge hier entscheiden, wenn auch abhängig vom gewählten Abbruchkriterium.

Bewertung generierter Merkmale

Die Bewertung der generierten Merkmale im Evaluationsschritt weist eine nahe Verwandtschaft zu dem Problem der automatisierten Merkmalsauswahl auf. Einen guten Überblick über dieses Gebiet liefern [Guyon and Elisseeff, 2003; Liu and Motoda, 2008]. Allerdings gibt es auch deutliche Unterschiede. Während es sich bei der Merkmalsauswahl meist um ein überwachtes Problem handelt (von unüberwachten Merkmalskonstruktionsverfahren wie beispielsweise PCA abgesehen), bei denen die Klassen, zwischen denen die Merkmale zu unterscheiden helfen sollen, bekannt sind, sind im vorliegenden Fall nur einige repräsentative Klassen bekannt. Im Folgenden wird zuerst der allgemeine Evaluationsansatz beschrieben, anschließend wird auf Details zur Bewertung der Relevanz einzelner Merkmale eingegangen.

Evaluationsansatz Der Ansatz für die Evaluation ist in Abb. 6.11 dargestellt, im Detail ist der Ablauf folgendermaßen definiert:

Eingabe	$\mathcal{M}_{all}$	Liste der Merkmale, die evaluiert werden sollen
	T	Menge gelabelter Trainingsdaten (möglichst repräsentativ für die betrachtete Problemdomäne)
	θ_{Me}	Schwellwert für die Bewertung ein Merkmals als relevant

Algorithmus

1: Für jede Sequenz in der Testdatenmenge T werden alle Merkmale, die in der Merkmalsliste $\mathcal{M}_{all}$ beschrieben sind, extrahiert.

2: Die vollständige Menge extrahierter Merkmale wird genutzt, um die Relevanz jedes einzelnen Merkmals für die Trennung zwischen verschiedenen Klassen der Testdaten T zu bestimmen. Das Zwischenergebnis dieses Schritts ist eine Liste von Bewertungen für alle Merkmale in $\mathcal{M}_{all}$.

3: Mit Hilfe der in Schritt **2** berechneten Liste von Bewertungen und des Schwellwertes θ_{Me} werden irrelevante Merkmale aus der Liste $\mathcal{M}_{all}$ entfernt. Die Wahl von θ_{Me} bildet einen Kompromiss zwischen Größe der genutzten Merkmalsmenge und Qualität der Merkmalsmenge.

Ausgabe Finalisierte, verkleinerte Teilmenge $\mathcal{M}_{potRel}$ der Eingabeliste $\mathcal{M}_{all}$ die nur noch die bezüglich der Testdaten T relevanten Merkmale enthält.

Relevanzmaß Das in Schritt **2** verwendete Relevanzmaß ist das zentrale Element der Merkmalsevaluation. Es weist jedem Merkmal eine *Bewertung* aus dem Intervall $[0,0;1,0]$ zu bezüglich einer Menge von Daten.

Für die Berechnung der Relevanz existieren verschiedene Ansätze, die in Abschnitt 3.2 allgemein und in Abschnitt 6.3 spezifisch für die Verwendung in der Aktivitätserkennung dargestellt werden. Sowohl bei den dort eingeführten Filter-Ansätzen als auch bei Wrapper-Ansätzen werden Merkmale üblicherweise als Teilmenge einer vorher gewählten Grundmenge von Merkmalen gewählt für ein spezifisches Klassifikationsproblem. Im vorliegenden Fall handelt es sich aber um eine etwas anders gelagerte Problemstellung. Ausgehend von einer sehr großen Menge möglicher Merkmale sollen nicht die sehr wenigen relevanten Merkmale für eine spezische Klasse gefunden werden. Stattdessen wird eine (Teil-)Menge dieser Merkmale gesucht, aus der dann in der späteren Anwendung relevante Teilmenge von Merkmalen für das jeweils vorliegende Problem in der gewählten Domäne gewählt werden kann. Einerseits ist es dadurch

nicht möglich, dass für jede zukünftige Klasse Trainingsdaten vorliegen (da diese Klassen zum Zeitpunkt der Merkmalsdefinition noch unbekannt sind), andererseits ist es für ein Merkmal nicht nötig, eine perfekte Klassifikation aller Trainingsdaten zu ermöglichen. Obwohl Filter und Wrapper wohlverstanden und gut analysiert sind, sind sie doch nicht unbedingt dafür geeignet, die im vorliegenden Fall nötige vorhersagende Merkmalsauswahl für unbekannte Klassen durchzuführen. Für dieses speziellere Problem wird die *potentielle Relevanz* von Merkmalen in der folgenden Darstellung definiert als:

Definition 6.1 (Potentielle Relevanz)**.** Ein Merkmal ist *potentiell relevant* bezüglich einer Menge gelabelter Daten, wenn mindestens eine Datensequenz existiert, für die der Wert des Merkmals nur wenig variiert, aber die Werte für mindestens zwei Datensequenzen variieren.

Mit dieser Definition kann man ein einfaches Relevanzmaß definieren. Für die Evaluation eines Merkmals werden dessen erstes und zweites stochastisches Moment in jeder Datensequenz berechnet. Das Merkmal kann zur Trennung zwischen zwei Datensequenzen genutzt werden, wenn die Varianzen $\mathrm{Var}(S_1)$, $\mathrm{Var}(S_2)$ in jeder der beiden Sequenzen möglichst niedrig, und die Differenz der Mittelwerte $\mu(S_1), \mu(S_2)$ in den beiden Sequenzen möglichst groß ist. Die Bewertung $\mathcal{R}_{ME}$ eines Merkmals bezüglich zweier Sequenzen S_1, S_2 erfolgt gemäß Gl. 6.1. Die Relevanz eines Merkmals wächst mit jedem Datensequenzen-Paar, zwischen denen das Merkmal trennen kann.

$$\mathcal{R}_{ME} = \frac{[\mu(S_1) - \mu(S_2)]^2}{\mathrm{Var}(S_1) + \mathrm{Var}(S_2)} \qquad [6.1]$$

Auswahl von Merkmalen Die im vorhergehenden Abschnitt eingeführte potentielle Relevanz eines Merkmals bezüglich der Trennung einzelner Aktivitäten wird genutzt, um die vollständige Liste generierter Merkmale von wenig nützlichen Merkmalen zu befreien. Dazu wird zunächst die potentielle Relevanz jedes Merkmals für jedes Paar zweier Aktivitäten berechnet. Anschließend wird ein Schwellwert θ_{Me} verwendet, um die Nützlichkeit des Merkmals binär zu interpretieren. Abschließend werden die Merkmale verworfen, die für weniger als eine gewählte Anzahl n_{mR} als nützlich eingestuft wurden.

Wahl der Parameter θ_{Me} und n_{mR} Die noch offenen Parameter des Verfahrens sind der Schwellwert θ_{Me} und die Mindestanzahl getrennter Aktivitäten n_{mR}, die gemeinsam darüber entscheiden, welche Merkmale gehalten und welche verworfen werden. Wie in der Beschreibung des Ansatzes angedeutet, haben niedrige Werte für beide Parameter zur Folge, dass weniger Merkmale verworfen und damit mehr Merkmale in der Liste behalten werden. Die Begründung für dieses Vorgehen wäre, dass es für die Gesamtperformanz des Systems günstiger ist,

einige irrelevante Merkmale zu behalten, als relevante Merkmale zu verlieren. Im Gegensatz dazu haben hohe Werte zur Folge, dass kleinere Merkmalsmenge präferiert werden. Der Nachteil dabei ist, dass Merkmale verloren werden können, die zwar nicht relevant für die Testdaten sind, aber in in der späteren realen Anwendung relevant sind.

Da beim Training von Aktivitäten noch eine spezifische Merkmalsauswahl stattfindet (Details hierzu werden in Abschnitt 6.3 erläutert) sind die Nachteile durch das Behalten einiger irrelevanter Merkmale (etwas höhere Laufzeit der späteren Merkmalsextraktion und -auswahl) geringer als die Nachteile durch das Verwerfen relevanter Merkmale (was im Allgemeinen in einer schlechteren Erkennungsleistung resultiert). Trotzdem muss ein vernünftiger Kompromiss für eine insgesamt günstige Systemperformanz gefunden werden.

Wie die detaillierte Evaluation des Einflusses der beiden Parameter in Anhang **??** zeigt, ist der Einfluss von θ_{Me} stärker, aber ab etwa $\theta_{Me} = 5$ flacht die Kurve der Merkmalsanzahl ab, und der Einflus von n_{mR} wird sichtbar. Die verwendeten Werte sind $\theta_{Me} = 8,6$ und $n_{mR} = 10$, so dass das Ergebnis des Evaluationsschritts eine zwar ausgedünnte, aber noch relativ große Merkmalsmenge ist.

Bewertung des Ansatzes

Die vorgestellte Merkmalsrepräsentation bietet die Möglichkeit, automatisch große Mengen von Merkmalen zu generieren, was in dem Verfahren zur automatischen Exploration auch genutzt wird. Die einfache Erweiterbarkeit durch unabhängige MEMs ist insbesondere dann wünschenswert, wenn neue Messungen (beispielsweise von neuen Sensoren) oder neue Analysemethoden integriert werden sollen.

Der Ansatz hat aber auch Nachteile. Da die MEMs beliebig kombiniert werden können, hängt der Abbruch der Generierung von neuen Merkmalen stark von den eingesetzten Strategien ab, und die Terminierung ist nicht einfach ersichtlich. Daher wird ein pragmatischer Ansatz für dieses Problem genutzt, nämlich die Komplexität (Schachtelungstiefe) der resultierenden Merkmalsbeschreibungen zu beschränken. Da die Anzahl der MEMs endlich ist, ist damit auch die Anzahl der Merkmalsbäume einer bestimmten Tiefe endlich und damit ein Abbruch garantiert. Dafür tritt durch diese Abbruchbedingung aber ein weiterer Parameter hinzu, und durch den künstlichen Abbruch ist es möglich, dass relevante Merkmale übersehen werden, wenn ihre Beschreibung komplexer ist als von dem gewählten Schwellwert zugelassen.

Eine quantitative Bewertung der Qualität der finalen Merkmalsmenge ist nicht direkt messbar, da es schwierig zu zeigen ist, dass es keine besseren (nicht generierten) Merkmale gibt. Daher wird als Annäherung die Erkennungsqualität bei der Verwendung der generierten Merkmale gemessen. In wohlverstandenen Domänen können die resultierenden Ergebnisse zusätzlich mit

denen von manuell modellierten Merkmalen verglichen werden. Wenn die finale Merkmalsmenge klein genug ist, kann die Qualität des Evaluationsschritts bewertet werden, indem die verbleibenden Merkmale mit der Teilmenge von Merkmalen verglichen werden, die ein klassischer Merkmalsauswahlalgorithmus selektiert. Der Vergleich wird auch in diesem Fall durch Trainieren und Evaluieren von Klassifikatoren durchgeführt.

6.2.4. Zusammenfassung

Der vorgestellte Ansatz zur Repräsentation von Merkmalen und zur Definition von relevanten Merkmalsmengen ist generisch und einfach erweiterbar. Durch die erweiterbare Merkmalsrepräsentation ist die automatische Generierung von Merkmalen und damit die Exploration großer Merkmalsräume möglich.

Der Vorteil der Erweiterbarkeit ist auf die Zukunft ausgerichtet. Mit der stetig voranschreitenden Entwicklung neuer Sensortechnologien und Fortschritten in der Merkmalsanalyse ist es ein notwendiges Merkmal der Repräsentation, solche fortschrittlichen Erweiterungen mit einem Minimum an Aufwand zeitnah in das System integrieren zu können, und ebenfalls zeitnah und automatisch erweiterte Basis-Merkmalsmengen zu bestimmen.

6.3. Merkmalsauswahl

In diesem Abschnitt wird die Merkmalsauswahl-Komponente beschrieben. Beim Trainieren neuer Aktivitäten hat diese Komponente die Aufgabe, aus der Menge extrahierter Merkmale diejenige kleinere Teilmenge zu wählen, die für eine zu lernende Aktivität besonders relevant sind. Durch die Auswahl wird die Robustheit der Erkennung insbesondere bei verrauschten und wenigen Trainingsdaten erhöht. Abschnitt 6.3.1 gibt einen Überblick über die Struktur der Komponente, die beiden folgenden Abschnitte erklären die Details der aktiven, passiven und interaktiven Teilkomponenten.

6.3.1. Konzept

Um auch bei kleinen Trainingsdatenmengen eine robuste Merkmalsauswahl zu ermöglichen, wurde ein Konzept entwickelt, das eine aktive Merkmalsauswahl auf Basis eines *Fast Correlation-based Filters* (Abschnitt 6.3.2) mit einer passiven Vorauswahl durch Hintergrundwissen und einer interaktiven Einbeziehung des Menschen (Abschnitt 6.3.3) kombiniert. Eine initiale Merkmalsliste wird – falls Hintergrundwissen über die zu lernende Aktivität vorhanden ist – anhand des Wissens modifiziert. Der Benutzer lenkt anschließend interaktiv die grobe Ausrichtung der Merkmale, während die aktive Merkmalsauswahl die aus den Trainingsdaten hervorgehende

Feinauswahl relevanter Merkmale trifft. Dieser Ablauf ist in Abb. 6.12 in Diagrammform dargestellt.

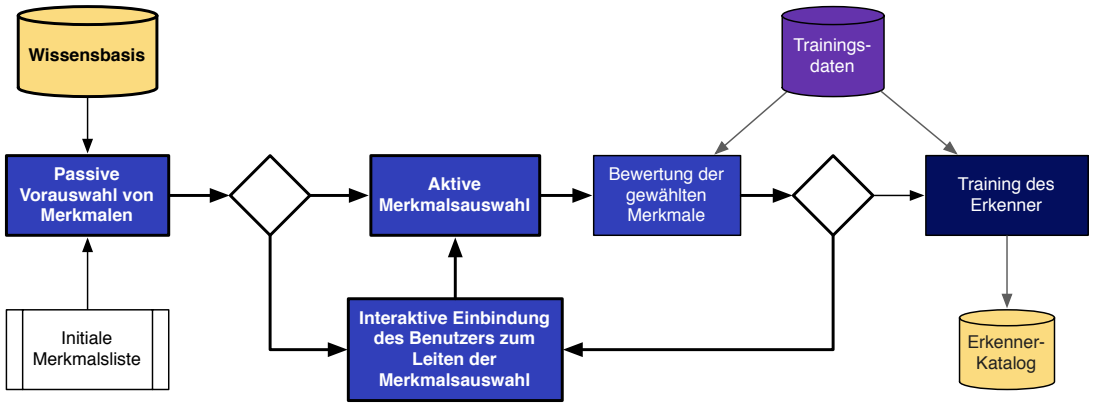


Abb. 6.12.: Darstellung des Ablaufs bei der Merkmalsauswahl, in der eine passive Auswahl aufgrund von Hintergrundwissen, eine aktive Auswahl anhand der Merkmalsrelevanz in Bezug auf Trainingsdaten und interaktiv vom Benutzer akquiriertes Hintergrundwissen genutzt werden.

6.3.2. Aktive Auswahl relevanter Merkmale

Für die automatische Auswahl relevanter Merkmalsteilmengen stehen die in Kapitel 3.2 beschriebenen Ansätze zur Auswahl. Da das entwickelte System unterschiedliche Klassifikatoren unterstützt, ist einer der groß Vorteile von Wrapper-Verfahren nicht gegeben, nämlich die Nutzung des gleichen Klassifikators für die Merkmalsauswahl und die spätere Erkennung. Da die Merkmalsauswahl auch in der Interaktion mit dem Benutzer genutzt wird, ist darüberhinaus eine möglichst geringe Laufzeit und damit Verzögerungen in der Interaktion gewünscht. Daher wird ein Filter-Ansatz verfolgt, der günstige, nicht Klassifikator-spezifische Merkmale auswählt. Um einen Algorithmus auszuwählen, wurden drei vielversprechende Ansätze verglichen bezüglich der Resultate der mit ihnen ausgewählten Merkmale:

- *Correlation-based Feature Subset Selection*-Algorithmus (CbFSS) nach [Hall, 2000]
- *Fast Correlation-based Filter*-Algorithmus (im Folgenden abgekürzt als FCbF) nach [Yu and Liu, 2003] und [Yu and Liu, 2004]
- *Relief-F*-Algorithmus nach [Kira and Rendell, 1992] und [Kononenko, 1994]

Sowohl FCbF als auch CbFSS sind deutlich schneller als Relief(-F) bei gleichzeitig besseren Ergebnissen. Im direkten Vergleich zwischen CbFSS und FCbF hat ersterer eine minimal

bessere TP-Rate, während letzterer eine deutlich höhere Präzision erreicht (für Details siehe Anhang C). Daher wurde der FCbF-Algorithmus zur Integration gewählt.

Im Folgenden wird zunächst der FCBF-Algorithmus detailliert beschrieben, dann der dabei verwendete Merkmals-Diskretisierungsalgorithmus. Schließlich wird der gewählte Ansatz bewertet.

Fast Correlation-based Filter-Algorithmus

Der FCbF wählt aus einer gegebenen Merkmalsmenge **M** bezüglich einer Datenmenge **D** und einer Klasse K eine Teilmenge relevantester Merkmale $\mathcal{M}_{rel}$. Als relevant gelten hierbei Merkmale, die *prädominant* (engl. *predominant*) gemäß der unten folgenden Definitionen sind.

Symmetrische Unsicherheit Als Evaluationskriterium wird beim FCbF die *Symmetrische Unsicherheit* genutzt, die die Korrelation zwischen zwei Merkmalen misst. Die symmetrische Unsicherheit SU ist in Gleichung 6.2 definiert, der dafür benötigte Informationsgewinn IG in Gleichung 6.3, die Entropie H in den Gleichungen 6.4 und 6.5. Dabei steht $P(x)$ für die a priori-Wahrscheinlichkeit von x, $P(x|y)$ für die a posteriori-Wahrscheinlichkeit von x gegeben y.

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right] \quad [6.2]$$

$$IG(X|Y) = H(X) - H(X|Y) \quad [6.3]$$

$$H(X) = -\sum_{x_i \in X} P(x_i) \cdot \log_2 P(x_i) \quad [6.4]$$

$$H(X|Y) = -\sum_{y_k \in Y} P(y_k) \sum_{x_i \in X} P(x_i|y_k) \cdot \log_2 P(x_i|y_k) \quad [6.5]$$

Der Informationsgewinn ist umso niedriger, je unabhängiger die beiden betrachteten Variablen sind bzw. je weniger Information über die eine Variable aus Wissen über die andere Variable gewonnen werden kann. In diesem Sinne ist ein Merkmal X ähnlicher zu einem Merkmal Y als zu einem Merkmal Z wenn gilt: $IG(Y|X) > IG(Z|X)$. Der SU-Wert liegt in dem Intervall $[0,1]$, wobei $SU(X,Y) = 0$ bedeutet, dass X und Y unabhängig voneinander sind. Wenn $SU(X,Y) = 1$ gilt, dann kann der Wert des einen Merkmals durch Kenntnis des anderen Merkmals vollständig vorhergesagt werden. Um die obigen Formeln auch zusammen mit kontinuierlichen Merkmalen verwenden zu können, müssen die Merkmale diskretisiert werden. Alternativ

können die entsprechenden kontinuierlichen Formeln zur Berechnung von Entropie, Informationsgewinn bzw. symmetrische Unsicherheit verwendet werden, die aber eigene Probleme bei der Berechnung einführen verglichen mit den diskreten Versionen.

Prädominanz Als relevante Merkmale für die Auswahl werden solche Merkmale angesehen, die die Eigenschaft haben, *prädominant* (engl. predominant) zu sein. Diese Eigenschaft wird in den Definitionen 6.2 und 6.3 definiert. Die Prädominanz eines Merkmals M_i gegenüber einem Merkmal M_k kann umgangssprachlich (und leicht vereinfacht) ausgedrückt werden als die Eigenschaft von M_i, stärker mit M_k korreliert zu sein als M_k mit der Klasse korreliert ist.

Definition 6.2 (Prädominante Korrelation)**.** Die Korrelation zwischen einem Merkmal M_i ($M_i \in \mathbf{M}$) und der Klasse K ist *prädominant*, wenn $SU(M_i,K) \geq \delta_{SU}$ (mit δ_{SU} als Mindestwert für die Korrelation des Merkmals mit der Klasse) und wenn für $\mathbf{M}'_i = \{M_k \in \mathbf{M} : M_k \neq M_i\}$ gilt:

$$\forall M_k \in \mathbf{M}'_i : \; SU(M_i,K) > SU(M_k,M_i)$$

Falls doch ein solches M_k zu M_i existiert (für das gilt $SU(M_k,M_i) \geq SU(M_i,K)$), wird es als *redundant-gleichgestelltes Merkmal* (engl. *redundant peer*) zu M_i bezeichnet.

Definition 6.3 (Prädominantes Merkmal)**.** Ein Merkmal M ist *prädominant* zu einer Klasse K, wenn entweder seine Korrelation zu K prädominant ist oder nach der Entfernung seiner redundant-gleichgestellten Merkmale prädominant werden kann.

Relevanz in FCbF Unter Nutzung der Definition der Prädominanz kann ein spezieller Relevanzbegriff für Merkmale bei der Verwendung des folgenden FCbF-Merkmalsauswahlalgorithmus definiert werden in Definition 6.4.

Definition 6.4 (Relevante Merkmale)**.** Ein Merkmal ist *relevant*, wenn es prädominant für die Vorhersage des Klassenkonzepts ist.

Algorithmus Der FCbF-Algorithmus in in Alg. 6.1 definiert. Er arbeitet in zwei Abschnitten. Zuerst wird die symmetrische Unsicherheit zwischen allen Merkmalen und der Klasse berechnet, und eine geordnete Liste $\mathbf{M}_{list}$ der Merkmale gebildet, deren SU größer als ein definierter Schwellwert δ_{SU} ist. Im zweiten Abschnitt werden aus $\mathbf{M}_{list}$ alle redundanten Merkmale entfernt, bis eine Liste übrigbleibt, die ausschließlich prädominante Merkmale enthält. Das Vorgehen zum Entfernen redundanter Merkmale ähnelt dabei dem Sieb des Eratosthenes. Beginnend mit dem ersten Element M_p von $\mathbf{M}_{list}$ wird für alle folgenden Elemente M_q geprüft, ob

$SU(M_p, M_q) \geq SU(M_q, K)$ ist. Falls ja, sagt M_p das Merkmal M_q besser vorher als M_q die Klasse K vorhersagen kann, und M_q kann aus der Liste entfernt werden. Anschließend wird mit dem nächsten Element in $\mathbf{M}_{list}$ fortgefahren. Die am Ende in $\mathbf{M}_{list}$ verbleibenden Merkmale werden als $\mathcal{M}_{rel}$ zurückgegeben.

Algorithmus 6.1 Fast Correlation-based Filter-Algorithmus zur automatischen Auswahl eines relevanten Merkmalsteilmenge nach [Yu and Liu, 2003].

Eingabe: Trainingsdaten $\mathbf{D}$, Schwellwert θ_{SU}

Ausgabe: Teilmenge $\mathcal{M}_{rel}$ relevanter Merkmale

1: **for** $i = 1 \ldots N$ **do**
2: Berechne $SU(M_i, C)$
3: **if** $SU(M_i, C) \geq \theta_{SU}$ **then**
4: Füge M_i zu $\mathbf{M}_{list}$ hinzu
5: **end if**
6: **end for**
7: Sortiere $\mathbf{M}_{list}$ absteigend nach $SU(M_i, C)$
8: $M_p \leftarrow$ erstesElement($\mathbf{M}_{list}$)
9: **repeat**
10: $M_q \leftarrow$ nächstesElement($\mathbf{M}_{list}$, M_p)
11: **repeat**
12: $M'_q \leftarrow$ nächstesElement($\mathbf{M}_{list}$, M_q)
13: **if** $SU(M_p, M_q) \geq SU(M_q, C)$ **then**
14: Entferne M_q aus $\mathbf{M}_{list}$
15: **end if**
16: $M_q \leftarrow M'_q$
17: **until** $M_q = NULL$
18: $M_p \leftarrow$ nächstesElement($\mathbf{M}_{list}$, M_p)
19: **until** $M_p = NULL$
20: $\mathcal{M}_{rel} \leftarrow \mathbf{M}_{list}$

Diskretisierung der Merkmale

Um Algorithmus 6.1 mit der in Gleichung 6.2 gegebenen Formel zur Berechnung der symmetrischen Unsicherheit einzusetzen, müssen die fraglichen Merkmale diskret sein. Da viele Merkmale in der Praxis als kontinuierliche Daten vorliegen, muss eine Diskretisierung der Werte vorgenommen werden. Bei der Diskretisierung wird der Wertebereich einer kontinuierlichen

Variablen in eine endliche Anzahl von Intervallen partitioniert. Das Resultat ist ein *Diskretisierungsschema* $\mathcal{D}$, das die Form $\mathcal{D} = \{[d_0, d_1], (d_1, d_2], \ldots, (d_{n-1}, d_n]\}$ hat.

Es gibt verschiedene Verfahren zur Diskretisierung, von einfachen äquidistanten Intervallen bis zu komplexen Verfahren, die die Länge der Intervalle und die Intervallgrenzen abhängig von zusätzlichen Informationen wählen [Kotsiantis and Kanellopoulos, 2006].

Für die vorliegende Aufgabe wurde der *Class-Attribute Interdependence Maximization (CAIM)*-Algorithmus aus [Kurgan and Cios, 2003] [Kurgan and Cios, 2004] adaptiert. Dieser Algorithmus wählt die Anzahl der diskreten Intervalle automatisch, und bestimmt die Intervallgrenzen anhand der gegenseitigen Abhängigkeit zwischen den Attributwerten und der zugeordneten Klasse. Um diese Abhängigkeit zu bestimmen, müssen Trainingsdaten vorliegen, bei denen die Klasse der Daten bekannt ist.

CAIM-Kriterium Zur Bewertung möglicher Intervallgrenzen wird das sogenannte *CAIM-Kriterium* eingesetzt, das die Abhängigkeit zwischen einer Klassenvariablen K und einem Diskretisierungsschema D für ein Merkmal M bewertet.

Die Berechnung des Kriteriums baut auf einer *Quanta-Matrix* genannten zweidimensionalen Häufigkeitsmatrix auf, siehe Abb. 6.13. Zum Aufbau der Quanta-Matrix werden die Diskretisierung von M und die Klassenvariable K als unabhängige Zufallsvariablen behandelt. Die h_{ki} in der Tabelle sind die Anzahl der Werte in den Daten, die zu Klasse k gehören und innerhalb von Intervall i liegen.

	Diskretisierungsschema D					Klasse
Klasse	$[d_0, d_1]$	$\ldots$	$(d_{i-1}, d_i]$	$\ldots$	$(d_{n-1}, d_n]$	insgesamt
K_1	h_{11}	$\ldots$	h_{1i}	$\ldots$	h_{1n}	M_{1+}
$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$
K_k	h_{i1}	$\ldots$	h_{ki}	$\ldots$	h_{kn}	M_{k+}
$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$
K_S	h_{S1}	$\ldots$	h_{Si}	$\ldots$	h_{Sn}	M_{S+}
Intervall insgesamt	M_{+1}	$\ldots$	M_{+i}	$\ldots$	M_{+n}	M

Abb. 6.13.: Darstellung einer Quanta-Matrix zur Berechnung des CAIM-Kriteriums. Klassenvariable K und diskretisierte Merkmalsvariable M werden als unabhängige Zufallsvariablen behandelt. Die Einträge h_{ki} sind die Häufigkeiten für Klasse k und Intervall i (in Diskretisierungsschema D) bezüglich einer gegeben Datenmenge $\mathbf{D}$.

Formel 6.6 gibt die genaue Berechnungsvorschrift für das CAIM-Kriterium an. Dabei steht K für eine Klassenvariable, $D = \{[d_0, d_1], (d_1, d_2], \ldots, (d_{n-1}, d_n]\}$ für das zu bewertende Diskreti-

sierungsschema mit n Intervallen und M für das betrachtete (und zu diskretisierende) Merkmal. Mit max_r wird die maximale Häufigkeit h_{ki} in Spalte i der Quanta-Matrix bezeichnet (die Häufigkeit der am stärksten in dem fraglichen Intervall i vertretenen Klasse), M_{+i} bezeichnet die Gesamtanzahl der Werte des Merkmals M in den Trainingsdaten **D**, die in das fragliche Intervall i fallen.

$$CAIM(K,D|M) = \frac{\sum_{i=1}^{n} \frac{max_i^2}{M_{+i}}}{n} \quad [6.6]$$

Das CAIM-Kriterium kann genutzt werden, um verschiedene Diskretisierungsschemata zu vergleichen und damit unterschiedliche Wahlen für Intervallgrenzen zu bewerten. Indem von einem Grundschema ausgegangen wird, das durch Teilung eines vorhandenen Intervalls erweitert wird, kann eine Menge verfeinerter Diskretisierungsschemata generiert werden, die mittels des CAIM-Kriteriums verglichen werden können.

Adaptierter CAIM-Algorithmus Der CAIM-Algorithmus (sowohl das Original als auch die vorliegende Adaption), der die Berechnung eines günstigen Diskretisierungsschemas unter Verwendung des CAIM-Kriteriums durchführt, läuft in zwei Schritten ab. Im ersten Schritt werden Kandidaten für Intervallgrenzen generiert und ein initiales Diskretisierungsschema festgelegt, das aus einem einzigen Intervall besteht. Im zweiten Schritt werden dann sukzessive neue Grenzen zu dem Diskretisierungsschema hinzugefügt, die lokal maximale CAIM-Werte liefern und das Schema verfeinern. Die Verfeinerung des Diskretisierungsschemas wird mindestens solange durchgeführt, bis so viele Intervalle wie Klassen vorliegen. Anschließend wird weiter verfeinert, solange durch das Verfeinern eines Intervalls noch eine Verbesserung in Bezug auf das CAIM-Kriterium festgestellt wird.

Der verwendete Algorithmus 6.2 unterscheidet sich von dem in [Kurgan and Cios, 2003] beschriebenen Algorithmus, um der Struktur der vorliegenden Daten Rechnung zu tragen, die als kontinuierliche Zeitreihen aus der Beobachtung von menschlichen Bewegungen vorliegen, und damit stetig und dicht sind (und nur durch die Abtastung diskretisiert). Daher wird in Zeile 4 bei der Auswahl möglicher Intervallgrenzen nur die in den Daten **D** aufgetretenen Merkmalswerte genutzt, nicht wie im Original auch Mittelwerte zwischen aufeinanderfolgenden Werten. Darüberhinaus werden zwei Parameter δ_{min} und ε_{max} genutzt, um den Einfluss zunahe beieinander liegender Merkmalswerte auf die Rechenzeit zu verringern, indem die Daten ausgedünnt werden. Merkmalswerte werden nur als Intervallgrenzen in Betracht gezogen, wenn der nächstkleinere Merkmalswert in den Daten weiter als δ_{min} entfernt ist. Um bei sehr eng beieinanderliegenden Werten trotzdem noch Intervallgrenzen zu produzieren, werden auf diese Art aber höchstens ε_{max} Werte übersprungen, bevor wieder ein Merkmalswert als Kandidat für eine In-

tervallgrenze der Diskretisierung mit aufgenommen wird. In der Evaluation haben sich für diese beiden Parameter die Werte $\delta_{min} = 0,001$ und $\varepsilon_{max} = 200$ als günstig erwiesen.

Bewertung

Die Auswahl relevanter Merkmale mit dem gewählten FCbF-Algorithmus hat Eigenschaften und Resultate, die vergleichbar mit anderen aktuellen Algorithmen sind, sowohl bezüglich der Laufzeit angeht, als auch bezüglich der Güte der gewählten Merkmale (siehe Anhang C.1 für Details).

In der Praxis sind die automatisch gewählten Merkmale in den meisten Fällen nicht optimal, was aber (da alle evaluierten Algorithmen die gleichen Probleme haben) nicht eine Schwäche des FCbF-Verfahrens ist, sondern in den zur Verfügung stehenden Daten begründet ist. Die Datenaufzeichnung, Segmentierung und Vorklassifizierung (engl. *labelling*) sind ein aufwändiger und fehlerträchtiger Prozess, sodass in den meisten Fällen zwar genügend Daten für den Teilschritt der Diskretisierung vorhanden sind, aber einige in der Realität irrelevante Merkmale aufgrund der nicht repräsentativen Daten rechnerisch sehr stark zur Trennung zwischen den verschiedenen Klassen beitragen können. Daher ist es in den meisten Fällen nötig, dass noch eine menschliche Kontrolle der ausgewählten Merkmale durchgeführt wird.

6.3.3. Merkmalsauswahl durch interaktives Einbringen von Hintergrundwissen

Um die Vorteile der aktiven und der manuellen Auswahl relevanter Merkmale zu verbinden, wurde ein Ansatz entwickelt, der den Benutzer in den Auswahlprozess einbindet, ohne ihm die vollständige Entscheidung über die Relevanz einzelner Merkmale aufzubürden [Lösch et al., 2008]. Der Ansatz basiert auf der Idee des Roboters als Begleiter und Interaktionspartner des Menschen. Der Benutzer interagiert mit dem System in der Art eines Lehrers, der einen Schüler leitet und den Trainingsprozess verfeinert, aber nicht vollständig übernimmt.

Um die Erweiterbarkeit des Systems auch bei sich ändernden Domänen und Merkmalen zu gewährleisten (die durch die einfache Erweiterbarkeit der MEMs nötig ist), wird zur Verbindung zwischen Merkmalen und Benutzerschnittstellen eine Taxonomie verwendet, die einen abstrakten Blick auf die Merkmale erlaubt. Aufbau und Definition der verwendeten Taxonomien wird in Abschnitt 6.3.3 beschrieben, gefolgt von der Einbindung in die interaktive Merkmalsauswahl. Die definierten Taxonomien bieten einige weitere Verwendungsmöglichkeiten, die in Abschnitt 6.3.3 skizziert werden. Anschließend erfolgt noch eine Bewertung der Vor- und Nachteile dieses Ansatzes.

Algorithmus 6.2 Adaptierter CAIM-Algorithmus zur Merkmalsdiskretisierung.

Eingabe: Merkmale $M_1, \ldots, M_n$

Ausgabe: Diskretisierungsschema $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$

for all M_i **do**
  {***Schritt 1***: Initialisierung}
  Finde Minimum- und Maximumwerte w_{min}, w_{max} von Merkmal M_i
  $G_{Kand} \leftarrow$ alle unterschiedlichen Werte von M_i außer w_{min}, w_{max}
  Sortiere G_{Kand} aufsteigend
  Initiales Diskretisierungsschema $D \leftarrow \{[w_{min}, w_{max}]\}$
  $GlobalCAIM \leftarrow 0$
  {***Schritt 2***: Erweiterung des Diskretisierungsschemas}
  $k \leftarrow 1$
  loop
    for all $g \in G_{Kand}$ **do**
      $D_g \leftarrow D$ erweitert mit g als zusätzliche Intervallgrenze
      $CAIM_g \leftarrow CAIM(K, D_g|M_i)$
    end for
    $CAIM \leftarrow \max_{g \in G_{Kand}} CAIM_g$
    $G \leftarrow \arg\max_{g \in G_{Kand}} CAIM_g$
    if $CAIM > GlobalCAIM$ oder $k < S$ **then**
      Erweitere D mit akzeptierter Intervallgrenze G
      $GlobalCAIM \leftarrow CAIM$
      Entferne G aus G_{Kand}
    else
      Beende Schleife
    end if
    $k \leftarrow k + 1$
  end loop
  $\mathcal{D}_i \leftarrow D$
end for

Merkmalstaxonomien

Um auch bei sich ändernden Merkmalsmengen eine konsistente Abbildung zwischen den Merkmalen und der Benutzerinteraktion zu ermöglichen, müssen die Merkmale systematisch geordnet werden. Hierzu wurde eine Repräsentation für Merkmalstaxonomien entwickelt, die die Darstellung verschiedener Zusammenhänge zwischen Merkmalen ermöglicht.

Repräsentation Taxonomien werden als gerichtete azyklische Graphen modelliert, wobei Knoten Mengen von Merkmalen repräsentieren, und Kanten Relationen zwischen diesen Merkmalsteilmengen darstellen. Die Relationen können allgemein als IST-TEIL-VON-Relation interpretiert werden, spiegeln dabei aber nicht notwendigerweise eine entsprechende physische Abhängigkeit wider. Weitere semantische Information bezüglich der Zusammenhänge zwischen den Merkmalen ist in der Graphstruktur repräsentiert, wenn nicht die vollständige Potenzmenge von der vollständigen Merkmals menge repräsentiert wird (was bedeuten würde, dass jede beliebige Kombination von Merkmalen einen Sinnzusammenhang darstellt).

Grundeinheit der Repräsentation sind einelementige Mengen, die je ein einzelnes Merkmal enthalten. Da alle Merkmale eindeutig identifizierbar sind, können verschiedene Elemente dieser Repräsentation (die jeweils einer Menge von Merkmalen entsprechen) sehr einfach korrekt vereinigt werden. Änderungen an tief in der Hierarchie angeordneten Knoten durch Hinzufügen oder Entfernen von Merkmalen können mittels Mengenoperationen (Vereinigung, Schnitt) durch den Graph propagiert werden.

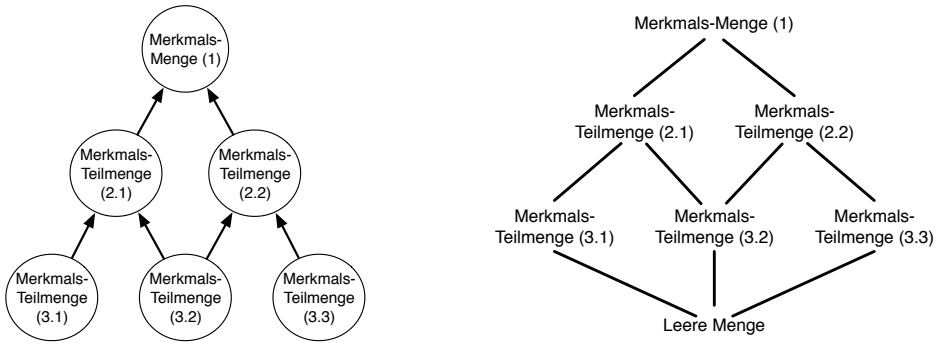


Abb. 6.14.: Abstrakte Beispiele für Merkmalstaxonomien. (a) Beispiel einer einfachen von Hierarchie. (b) Darstellung der Menge aus (a) mit ergänzter leerer Menge, in Form eines Hasse-Diagramms.

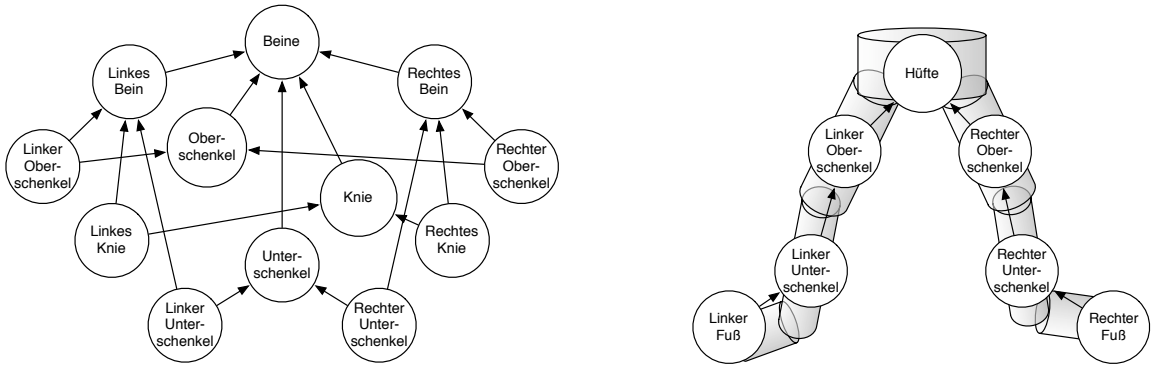
Abb. 6.14(a) zeigt eine beispielhafte, von konkreten Merkmalen abstrahierte Merkmalstaxonomie. Die Struktur solcher Taxonomien entspricht der eines Hasse-Diagramms (zum Vergleich ein entsprechendes Hasse-Diagramm mit ergänzter leerer Menge in Abb. 6.14(b)) bzw.

einer endlichen Halbordnung. Die von den Knoten repräsentierten Merkmalsmengen, zusammen mit der durch die Kanten definierten Ordnung und den Mengenoperationen Vereinigung $\cup$ und Schnittmengenbildung $\cap$ bilden die algebraische Struktur eines *Verbands*. Der Verband kann zu einem vollständigen Verband ergänzt werden mit der vollständigen Merkmalsmenge als Maximum, der leeren Menge als Minimum (mit Verbindungen zu allen einelementigen Mengen) und durch das Hinzufügen aller gemeinsamen Teilmengen (Schnittmengen) von je zwei Elementen. Ein vollständiger Verband erlaubt das Finden von Infimum und Supremum für jede beliebige Teilmenge von Verbandselementen. Im Rahmen der hier verwendeten Merkmalstaxonomien bedeutet die Suche nach einem Supremum zweier Teilmengen T_1 und T_2 die Suche nach dem kleinsten Element der Taxonomie, das alle in den Teilmengen T_1 und T_2 enthaltenen Merkmale enthält.

Konkrete Beispiele Drei konkrete Merkmalstaxonomien der beschriebenen Art wurden konstruiert und untersucht, mit dem Ziel, sie für die Abbildung zwischen Benutzereingaben und gemeinten Merkmalsmengen zu ermöglichen. Die folgenden Taxonomien unterscheiden sich in der Motivation für die gewählte Ordnung und daraus folgend auch in der resultierenden Struktur.

Körperhierarchie Menschen verwenden viele unterschiedliche Begriffe für die verschiedenen Körperteile, und gehen dabei von dem impliziten Wissen aus, dass beispielsweise der *Unteram* Teil des *Arms* ist. Die Körperteile bilden eine Hierarchie, die in einer Taxonomie abgebildet werden kann, im Folgenden als *Körperhierarchie-Merkmalstaxonomie (KHMT)* bezeichnet. Die Zuordnung der Merkmale erfolgt dabei anhand des für die Berechnung des Merkmals zugrundeliegenden Körperteils. Der in Abb. 6.15(a) gezeigte Ausschnitt der so konstruierten Taxonomie (zur Verbesserung der Übersichtlichkeit zeigen die einzelnen Knoten eine Klartextbezeichnung) bildet einen Teil des Unterkörpers ab.

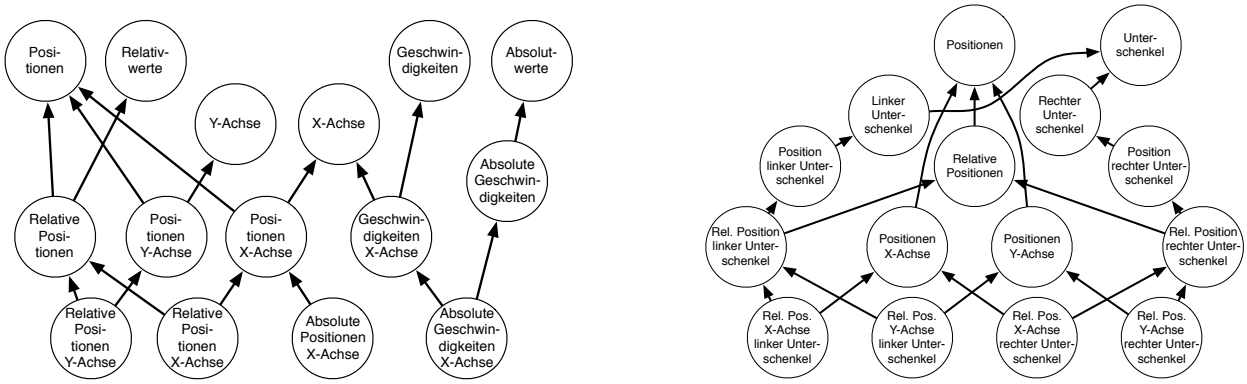
Kinematische Kette In ähnlicher Art kann eine Taxonomie auf der vom menschlichen Körper gebildeten kinematischen Kette gebildet werden, im Folgenden als *Kinematische Ketten-Merkmalstaxonomie (KKMT)* bezeichnet. Dabei werden die Merkmale geordnet anhand der Eigenschaft des zugrundeliegenden Körperteils, sich automatisch mitzubewegen bei Bewegungen eines anderen Körperteils. Beispielsweise ist es bei einer Bewegung des Oberarms sehr wahrscheinlich, dass sich auch die entsprechende Hand bewegt. Der in Abb. 6.15(b) gezeigte Ausschnitt der Taxonomie bildet den Unterkörper nach.



(a) (b)

Abb. 6.15.: Ausschnitte aus konkreten Merkmalstaxonomien: (a) Ausschnitt aus der Körperhierarchie-Merkmalstaxonomie, (b) Ausschnitt aus der Kinematische Ketten-Merkmalstaxonomie.

Merkmalstyp Ein anderer Ausgangspunkt wird bei dieser Taxonomie verfolgt. Hier werden die Typen der Merkmale analysiert. Es gibt Merkmale unterschiedlicher Art (Positionen, Winkel, Geschwindigkeiten, etc.), die relativ zueinander oder absolut (also relativ zum globalen Koordinatensystem) betrachtet werden können. Eine nach diesen und ähnlichen Gesichtspunkten aufgebaute Taxonomie ist für verschiedene Anwendungsfälle interessant, beispielsweise sind für die Unterscheidung zwischen einem einfachen Handausstrecken und einer Boxbewegung die Geschwindigkeiten dabei eine wichtigere Information als die Winkel des Arms oder die Handpositionen. Die Überführung in eine Taxonomie führt zur *Merkmalstyp-Merkmalstaxonomie (MTMT*, von der ein Ausschnitt in Abb. 6.16(a) gezeigt ist.



(a) (b)

Abb. 6.16.: Weitere Ausschnitte aus Merkmalstaxonomien: (a) Ausschnitt aus der Merkmalstyp-Merkmalstaxonomie, (b) Ausschnitt aus der Merkmalstaxonomie, entstanden durch Fusion von KHMT und MTMT.

Fusion von Merkmalstaxonomien Die in den vorherigen Abschnitten vorgestellten Merkmalstaxonomien basieren auf unterschiedlichen Sichten auf die Merkmale. Die Taxonomien sind unterschiedlich gut geeignet für verschiedene Anwendungen. Für die meisten Zwecke wird KHMT ähnlich gut funktionieren wie KKMT, aber KHMT ist sehr verschieden von MTMT. Während KHMT beispielsweise einen direkten Zugriff auf Merkmale eines bestimmten Körperteils erlaubt (was in MTMT so nicht möglich ist), erlaubt MTMT einfachen Zugriff spezielle Klassen von Merkmale, wie beispielsweise relative Positionen (was wiederum in KHMT nicht möglich ist). Daher liegt die Idee nahe, verschiedene Taxonomien so zusammenzufügen, dass zwar die jeweiligen Vorteile der Elterntaxonomien beibehalten werden, aber die Nachteile abgeschwächt oder vollständig ausgeschaltet werden. Wenn möglich sollte sogar eine feinere Taxonomie herauskommen, die (den beiden gegebenen Beispielen folgend) auch Elemente wie die relative Positionen der Hände als Merkmalsmengen beinhaltet.

Eine einfache Vereinigung der beiden Mengen erfüllt diese Anforderungen nicht, da auf diese Art keine Verbindungen zwischen den Elementen der verschiedenen Taxonomien konstruiert werden. Im allgemeinen sind nur die einelementigen Mengen (und das alle Merkmale enthaltende Element) identisch zwischen zwei beliebigen Taxonomien. Stattdessen ist es nötig, den Abschluss unter paarweisem Schnitt aller Elemente der vereinigten Menge zu bilden, um die noch fehlenden Elemente zu konstruieren, die die Taxonomie vervollständigen und dabei auch die Verbandseigenschaft der fusionierten Taxonomie wieder herstellen.

Beispielhaft zeigt Abb. 6.16(b) einen Ausschnitt aus der durchgeführten Fusion von KHMT und MTMT (die gewählten Bezeichnungen der Knoten dienen der übersichtlicheren Darstellung, sie können nicht automatisch gewählt werden). Schon dieser kleine Ausschnitt zeigt, dass die Größe der fusionierten Taxonomie rapide ansteigt im Vergleich zur Größe der fusionierten Taxonomien. Neben der erwähnten Benennung der Elemente kann auch die Abbildung zwischen Elementen der Taxonomie und Benutzereingaben bei der Fusion von Taxonomien im Allgemeinen nicht automatisch erweitert werden, wie das bei der Taxonomie selbst der Fall ist.

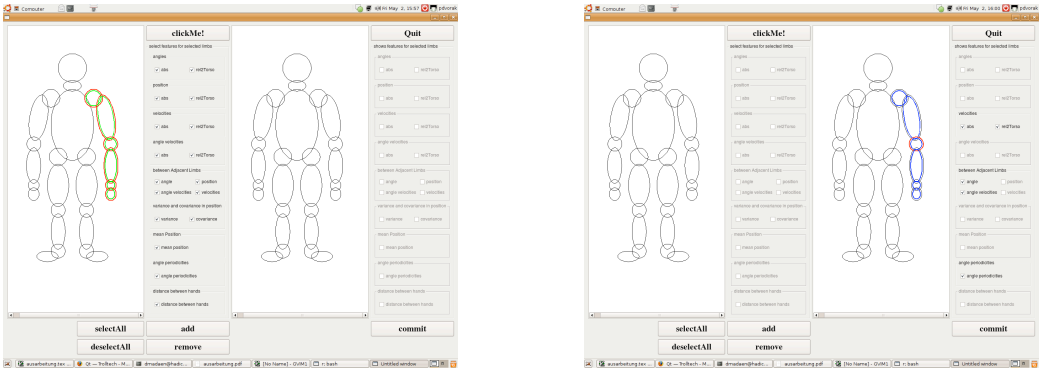
Taxonomien zur interaktiven Merkmalsauswahl

Die im vorherigen Abschnitt beschriebenen Taxonomien werden genutzt, um eine Abbildung zwischen den intern genutzten Modellen des Aktivitätserkennungssystems und den von Benutzern interaktiv gegebenen Hinweisen zur Merkmalsauswahl zu ermöglichen.

In Analogie zum menschlichen Lernen mit einem Lehrer, der Hinweise und Hilfen gibt, soll das System von Hilfestellungen des Benutzers geleitet werden können, die auch beim menschlichen Lernprozess auf natürliche Art gegeben werden. Wenn ein Benutzer eine neue Aktivität eintrainieren möchte, kann er direkt Hinweise geben, beispielsweise welche Körperteile das

System beachten und welche außer Acht gelassen werden sollen. Diese Information wiederum wird mit Hilfe der verwendeten Taxonomie in entsprechende Repräsentationen abgebildet werden.

Diese Informationen ersetzen nicht die aktive Merkmalsauswahl des Systems, sondern ergänzen diesen durch Manipulation der Merkmalsgrundmenge. Die Benutzerhinweise können aus Sicht des Systems über verschiedene Kanäle, z.B. per Spracheingabe oder GUI erfolgen, ein Beispiel für letztere Verwendung wird in Abb. 6.17 gezeigt.



(a) (b)

Abb. 6.17.: Interaktive GUI zur Einbindung des Benutzers in die Merkmalsauswahl (links der Arbeitsbereich zum Anlegen von Auswahlen, rechts die gesammelte bisherige Auswahl): (a) Auswahl des linken Arms mit einigen Merkmalstypen, (b) Rechter Arm in der endgültigen Auswahl, die angezeigten Merkmalstypen gehören zum (rot) markierten Ellbogen.

Der wichtigste Aspekt, der für diese Verwendung zu leisten ist, ist die Abbildung von Benutzereingaben auf die definierten Taxonomieelemente. Abhängig von der verwendeten Benutzerschnittstelle kann es beispielsweise notwendig sein, auch Eingaben wie „schau hier her“ in die Taxonomie abzubilden. Zu diesem Zweck können die Elemente der Taxonomie mit semantischen Angaben annotiert werden, die diese Abbildung unterstützen. Da diese Abbildung zeitlich unabhängig von der Verwendung für das Trainieren von Aktivitäten durchgeführt werden kann, kann sie offline von einem Spezialisten vorgenommen werden, während weniger erfahrene Benutzer sie nur einzusetzen brauchen.

Weiterführender Taxonomie-Einsatz

Der Merkmalsauswahlprozess kann durch Ausnutzung der Verbandeigenschaften der verwendeten Taxonomien noch weiter unterstützt werden. Dabei werden schon trainierte Klassifikatoren genutzt, um vor Beginn der Merkmalsauswahl zusätzliche Informationen zu den Trainingsdaten zu gewinnen. Die Trainingsdaten werden mit darin erkannten Aktivitäten bzw. den

von diesen Aktivitäten genutzten Merkmalen annotiert. Unter der Annahme, dass die zur Erkennung dieser Aktivitäten genutzten Merkmale bzw. die den Merkmalen zugrundeliegenden Bewegungen nicht gleichzeitig eine andere Aktivität anzeigen können, können diese Merkmale von der weiteren Suche nach relevanten Merkmalen ausgeschlossen werden. Durch Nutzung der Verbandstruktur kann die Menge der auszuschließenden Merkmale allerdings noch erweitert werden, indem die von der erkannten Aktivität „belegten“ Körperteile (bzw. die zu ihnen gehörigen Merkmale) identifizert und ausgeschlossen werden. Diese Identifikation entspricht in einem Verband gerade dem Finden des Supremums der einzelnen Merkmale.

Bewertung des Ansatzes

Die Evaluationen zeigt, dass die Einbindung des Benutzers zu besseren Resultaten bei der Merkmalsauswahl führt, siehe Abschnitt 7.3.2. Im Durchschnitt genügen dabei zwei Verfeinerungsschritte durch den Benutzer, um dieses Ziel zu erreichen. Insbesondere die Generalisierung der mit den gewählten Merkmalen trainierten Erkenner wird bei der Verwendung von wenigen Trainingsdaten oder Trainingsdaten von nur einer Person verbessert.

6.3.4. Passive Merkmalsauswahl durch Hintergrundwissen

Die Merkmalsauswahl wird auch von einer passiven Komponente unterstützt, die durch Nutzung von in einer Wissensbasis abgelegtem Hintergrundwissen über Aktivitäten und die sie definierenden Merkmale eine Vorauswahl bezüglich interessanter Merkmale treffen kann.

Die Idee ähnelt dabei der im vorigen Abschnitt beschriebenen Einbindung der interaktiv erlangten Hinweise von Benutzern. Entweder durch eine ähnliche Benutzerinteraktion wie oben beschrieben oder durch andere Quellen (denkbar ist hier beispielsweise eine Anreicherung der Wissensbasis durch Data Mining im World Wide Web) kann die Wissensbasis befüllt werden. Beim Auswählen von Merkmalen für eine neue Aktivität wird zunächst geprüft, ob in der Wissensbasis für diese Aktivität Informationen vorliegen. Die Aktivität wird dabei durch ihren Namen identifiziert, synonyme Bezeichnungen und ähnliche Probleme müssen dabei gesondert behandelt werden, siehe auch Abschnitt 6.4.6 für Lösungsideen. Im Erfolgsfall wird die Informationen wie in Abschnitt 6.3.3 genutzt, um die Merkmalsliste zu reduzieren, die anschließend für die weitere Reduzierung an die aktive/interaktive Merkmalsauswahl weitergegeben wird.

Die Wissensbasis kann neben einer separaten Erweiterung um neues Wissen auch durch die im normalen Ablauf vorgenommenen Benutzereingaben für die interaktive Auswahl ergänzt werden.

6.4. Klassifikation

Dieser Abschnitt stellt die Arbeitsweise der Klassifikationskomponente im Detail dar. Nach der Beschreibung des inneren Architektur in Abschnitt 6.4.1 werden die einzelnen Teile in den darauf folgenden Abschnitten 6.4.2 – 6.4.5 beschrieben. Schließlich fasst Abschnitt 6.4.6 die Art des verwendeten Hintergrundwissens und die Details der Verwendung zusammen.

6.4.1. Aufbau der Klassifikation

Aktivitäten unterscheiden sich in verschiedenen Bewertungsdimensionen voneinander. Es gibt komplexere und weniger komplexe Aktivitäten, Aktivititäten die von stärkeren Bewegungen bestimmt sind und solche, die eher statisch ausgeführt werden. Viele dieser Unterschiede korrelieren mit der Effektivität, die unterschiedlicher Klassifikatoren beim Einsatz zur Erkennung dieser Aktivitäten zeigen.

Um die bestmöglichen Ergebnisse zu erzielen, wurde deshalb eine Architektur für die Klassifikation konzeptioniert, die geeignete Klassifikator- und Struktur-Kombinationen wählen kann. Die beiden Abbildungen 6.18 und 6.19 zeigen die beim Lernen bzw. beim Erkennen zum Einsatz kommenden Aspekte des Systems. Die eigentliche Klassifikation ist realisiert als zweischichtige Architektur (in beiden Abbildungen mit (1) markiert), die unterschiedliche Erkenner (insbesondere auch die mit (2) markierten zusammengesetzten Klassifikatoren und die mit (3) markierten auf Bewegungsprimitiven basierenden HMMs) integrieren kann, und zusätzlich durch Hintergrundwissen (mit (5) markiert) und eine nachgeschaltete Ergebnisaufbereitung (4) ergänzt wird.

Beim Lernen neuer Aktivitäten wird eine Entscheidung für den/die zu verwendenden Klassifikatoren getroffen, und die entsprechenden Modelle werden unter Nutzung der Trainingsdaten trainiert. Bei der Erkennung werden die vorher gelernten Modelle aus dem Erkenner-Katalog geladen, und mit den Merkmalen abgefragt, die aus den aktuellen Daten extrahiert wurden. Die Ergebnisse werden einer Nachbehandlung zur Korrektur von Fehlern und Inkonsistenzen unterzogen.

Die zweischichtige Struktur und Details zur Klassifikatorauswahl werden in Abschnitt 6.4.2 diskutiert, die beiden mit (2) und (3) markierten Erkenner-Konzepte der zusammengesetzten Klassifikatoren und der Bewegungsprimitive-basierten HMMs werden in den folgenden Abschnitten 6.4.3 und 6.4.4 präsentiert. Die Ergebnis-Nachbehandlung wird in Abschnitt 6.4.5 diskutiert, und schließlich fasst Abschnitt 6.4.6 die einzelnen Details zur Verwendung von Hintergrundwissen in der Klassifikation einheitlich zusammen.

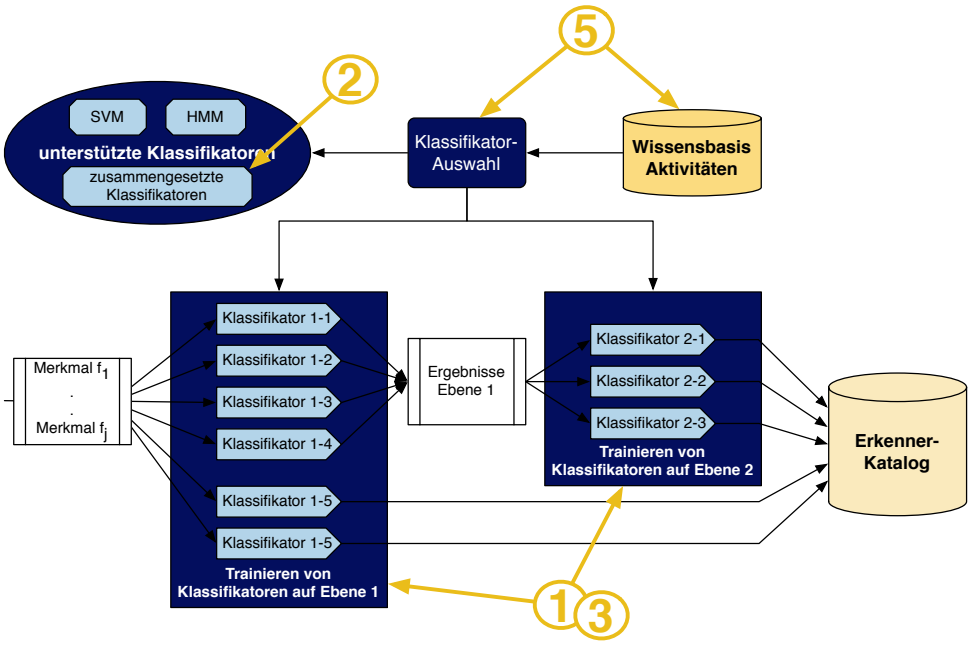


Abb. 6.18.: Zusammenspiel der einzelnen Komponenten der Klassifikation beim Lernen neuer Aktivitäten.

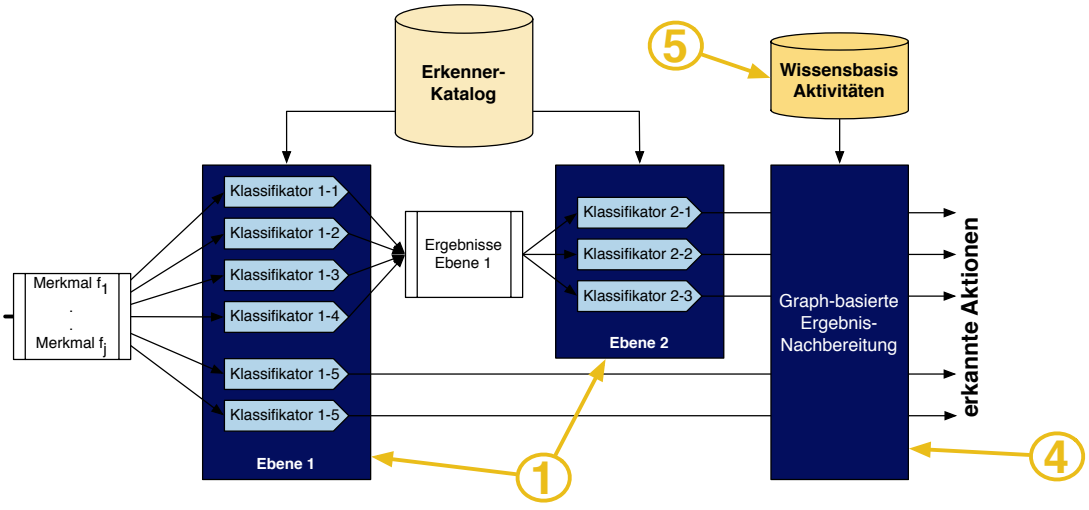


Abb. 6.19.: Zusammenspiel der einzelnen Komponenten der Klassifikation bei der Erkennung von Aktivitäten.

6.4.2. Zweischichtige Erkenner-Architektur

Die Schichtenarchitektur erlaubt die flexible Verwendung Erkennern auf einer oder zwei Schichten, d.h. eine direkte Erkennung bei Verwendung einer Schicht für eine Aktivität, oder die Indirektion der Erkennung über die Erkennung von Teilhandlungen auf der ersten Schicht und die Erkennung der eigentlichen Aktivität auf der zweiten Schicht. Das Ziel dabei ist es, eine passende Behandlung sowohl von strukturell komplexen Aktivitäten, die aus einer längeren Folge von einzelnen, unterschiedlichen Bewegungen bestehen, als auch von sehr einfach strukturierten Aktivitäten zu erreichen.

Für einfache Aktivitäten kann direkt ein Erkenner für diese Aktivität trainiert werden. Auch hier sind je nach Aktvitität nicht alle Klassifikatoren gleich gut geeignet. Beispielsweise sind für statische Aktivitäten wie stehen oder (ohne Bewegung) auf etwas zu zeigen SVMs sehr gut geeignete Klassifikatoren, während für einfache periodische Aktivitäten wie Winken HMMs erheblich bessere Ergebnisse zeigen. Falls entsprechendes Hintergrundwissen vorhanden ist (s. Abschnitt 6.4.6), wird die Auswahl des Klassifikators anhand dieses Wissens durchgeführt. Falls dieses Wissen fehlt, kann von einem menschlichen Experten ein Klassifikator ausgewählt werden. Falls kein Hintergrundwissen vorhanden ist oder genutzt werden soll, werden alle unterstützten Typen von Erkennern trainiert, und abhängig von der Qualität der anschließenden Evaluationsergebnisse der einzelnen Erkenner gewichtet fusioniert (s. Abschnitt 6.4.3 für die Details der Fusion).

Für komplexere Aktivitäten (im obigen Sinn) wird die Erkennung in zwei Schritten durchgeführt, indem zuerst auf Ebene 1 wichtige Teilbewegungen erkannt werden, die den Erkennern auf Ebene 2 als Eingabe dienen. Speziell können auf Ebene 1 sogenannte *Bewegungsprimitive* als Teilbewegungen erkannt werden, die auf Ebene 2 als *Human Activity Language (HAL)*-ähnliche Eingabesymbole genutzt werden, um als Eingabe für HMMs zu dienen. Dieser Teil der Erkennung wird in Abschnitt 6.4.4 erläutert. Die Entscheidung über den Einsatz des zweischichtigen Ansatzes wird wie beschrieben anhand der Komplexität der Aktivität getroffen. Die nötige Information über die Komplexität wird anhand von Hintergrundwissen getroffnen, das wiederum entweder in der Wissenbasis persistent vorliegt, oder vom Benutzer bereitgestellt wird.

6.4.3. Zusammengesetzte Klassifikatoren

In einigen typischen Szenarien führt der Einsatz eines einzelnen, trainierten Erkenners nicht zum gewünschten Ziel, obwohl er eigentlich gut geeignet für die betrachtete Aktivität ist. Für solche Fälle wird das Konzept eines *zusammengesetzten Klassifikators (ZK)* eingesetzt, das im

Folgenden beschrieben wird. Ein ZK kombiniert zwei oder mehr Klassifikatoren gleichen oder unterschiedlichen Typs (wobei auch die rekursive Verwendung von ZKs möglich ist) zu einem neuen Klassifikator. Es gibt zwei Haupteinsatzzwecke für die zusammengesetzten Klassifikatoren:

(1) Erkenner unterschiedlichen Typs für die gleiche Aktivität.

(2) Erkenner für verschiedene Aktivitäten, die zusammen eine allgemeinere Aktivität abbilden.

Bei (1) handelt es sich um den in 6.4.2 erwähnten Fall, dass keine Entscheidungsgrundlage zur Wahl eines Klassifikatortyps zur Verfügung steht. In diesem Fall wird ein Erkenner von jedem unterstützten Typ für die Aktivität gelernt, und zu einem ZK zusammengesetzt.

Bei (2) können mit diesem Konzept Erkenner für verschiedene Aktivitäten gemeinsam genutzt und geeignet fusioniert werden, um entweder eine allgemeinere Aktivität zu erkennen (beispielsweise ein allgemeines WINKEN zusammengesetzt aus WINKEN MIT LINKER HAND und WINKEN MIT RECHTER HAND), oder um für Aktivitäten, für die es deutlich unterschiedliche Ausführungen gibt, jeweils dediziert einen robusten Erkenner für jede der Ausführungen zu trainieren, und hinterher zu kombinieren. Ein Beispiel ist Winken, das auf einige deutlich unterschiedliche Arten ausgeführt werden kann.

Der entscheidende Aspekt bei der Kombination der einzelnen Erkenner liegt dabei in der Fusion der Einzelergebnisse der untergeordneten Erkenner, in [Al-Ani and Deriche, 2002] wird eine Reihe von unterschiedlichen Möglichkeiten aufgelistet. Der hier eingesetzte Mechanismus zur Fusion kann unterschiedliche Operatoren einsetzen, abhängig vom gewünschten Effekt. Die folgenden drei Operatoren stehen für den Einsatz zur Verfügung (wobei unabhängig vom genutzten Operator immer *alle* Einzelklassifikationen k_i eines ZK mit dem gewählten Operator zusammengefasst werden):

- Konjunktion (logisches UND): Das fusionierte Ergebnis von n Einzelergebnissen R_i ist das Minimum aller dieser Werte:

$$\mathcal{K} = \min_i k_i$$

- Disjunktion (logisches ODER): Das fusionierte Ergebnis von n Einzelergebnisse R_i ist das Maximum aller dieser Werte:

$$\mathcal{K} = \max_i k_i$$

- Gewichtetes Mittel: Das fusionierte Ergebnis $\mathcal{K}$ ergibt sich als gewichtetes arithmetisches Mitetl Einzelergebnisse k_i:

$$\mathcal{K} = \frac{\sum_i w_i k_i}{\sum_i w_i} = \sum_i W_i k_i \qquad \text{mit } W_i = \frac{w_i}{\sum_i w_i} \quad \left(\Rightarrow \sum_i W_i = 1 \right)$$

Die Auswahl des geeigneten Operators ergibt sich direkt aus dem intendierten Einsatzzweck. Für die Kombination unterschiedlicher Erkenner für die gleiche Aktivität wird das gewichtete Mittel eingesetzt, wobei die Gewichte w_i abhängig von den Evaluationsergebnissen des Einlernens gesetzt werden, und in der späteren Verwendung noch adaptiert werden können.

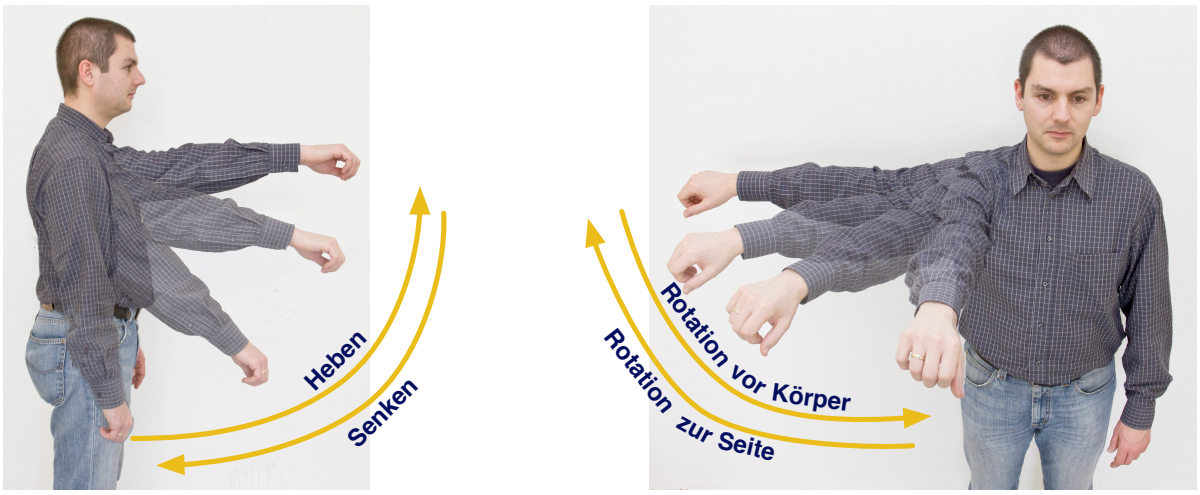
Die Disjunktion wird genutzt für das oben gegebene Beispiel deutlich unterschiedlicher Ausführungen der gleichen Aktivitäten. Die Konjunktion kann verwendet werden um Erkenner verschiedener Aktivitäten zu einer neuen, komplexen Aktivität zusammenzusetzen. Beispielsweise kann auf diese Weise ein WINKEN und ein GEHEN zusammengesetzt werden zu einem GEHEND WINKEN (im Gegensatz zu einem STEHEND WINKEN).

6.4.4. HMM auf Bewegungsprimitiven

Für die Klassifikation komplexer Aktivitäten gibt es neben der bisher präsentierten Möglichkeit, direkt Erkenner für einzelne Aktivitäten zu trainieren, auch den Ansatz, analog zum Vorgehen in der Spracherkennung ein *Alphabet* von Grundbewegungen zu identifizieren und zu erkennen, mit dessen Hilfe anschließend komplexere Aktivitäten zusammengesetzt werden können. Ein wichtiger Schritt in diese Richtung ist die Entwicklung der *Human Activity Language (HAL)* durch Aloimonos et al. in [Guerra-Filho and Aloimonos, 2006a,b].

Auf Ebene 1 der Erkennung wird eine Erkennung von Bewegungsprimitiven auf HAL-ähnlichen Symbole durchgeführt. Für jeden Freiheitsgrad eines Gelenks wird prinzipiell zwischen Rotationen im und gegen den Uhrzeigersinn unterschieden (anschaulicher ausgedrückt z.B. beim Ellbogen zwischen Beuge- und Streckbewegungen, siehe [Butsch, 2009]), Abb. 6.20 zeigt beispielhaft zwei solche Bewegungspaare für den rechten Arm. Durch die allgemeine Definition der Bewegungsprimitive besteht keine Notwendigkeit, sie für jede neue Aktivität neu zu erlernen. Stattdessen können einmalig besonders robuste und gut generalisierende Erkenner für sie eintrainiert werden, die anschließend für alle Erkennungen von Bewegungsprimitiven genutzt werden.

Die aus den erkannten Bewegungsprimitiven resultierenden Daten dienen als Eingabe für die Erkennung auf Ebene 2. Hier werden HMMs eingesetzt, um für die Erkennung der behandelten komplexen Aktivitäten auch mehrteilige Bewegungssequenzen und teilweise Wiederholungen von Teilbewegungen robuste Resultate zu erreichen. Da die Daten mehrdimensional sind, muss



(a) (b)

Abb. 6.20.: Beispiele für Bewegungsprimitive des rechten Arms für die Erkennung auf Ebene 2: (a) Heben und Senken des Arms vor dem Körper, (b) Drehen des Oberarm von der Seite vor den Körper und umgekehrt.

ein entsprechend erweitertes HMM eingesetzt werden. Daher werden die in Kapitel 3.3.2 eingeführten *Coupled Hidden Markov Models (CHMM)* genutzt.

Für komplexe Aktivitäten zeigt dieser Ansatz deutlich bessere Erkennungsergebnisse als eine direkte Erkennung, beispielsweise mit SVMs, während umegekehrt für Posen-abhängige, durch nur wenig Bewegungen bestimmte Aktivitäten von SVMs mit besseren Erkennungsergebnissen behandelt werden.

6.4.5. Ergebnis-Nachbehandlung

Die von den Erkennern gelieferten Ergebnisse werden noch einem Nachbehandlungsschritt unterzogen, der Inkonsistenzen korrigiert. Die behandelten Fehler resultieren aus der Systemeigenschaft, die unabhängige Kombination von Erkennern für einzelne Aktivitäten zu erlauben. Das ist zwar, wie in Kap. 4 definiert wurde, eine wünschenswerte Eigenschaft des Systems. Aber zur Realisierung genügt es nicht, die Erkenner gemeinsam laufen zu lassen. Ein Beispiel für die dabei auftretenden Probleme ist die kombinierte Verwendung von Erkennern für WINKEN und AM KOPF KRATZEN. Wenn man nur den rechten Arm betrachtet, sollte sich die Wahrscheinlichkeit der Erkennung dieser beiden Aktivitäten genau zu $1,0$ (bei Annahme einer geschlossenenen Welt (engl. *closed world assumption*, d.h. nur diese beiden Akt. sind überhaupt möglich) oder maximal auf $1,0$ (bei Annahme einer offenen Welt (engl. *open world assumption*, d.h. es werden auch andere, nicht betrachtete Aktivitäten für möglich gehalten) addieren. In der Praxis kann es aber sein, dass sich die erkannten Einzelwahrscheinlichkeiten zu einer

erheblich größeren Summe addieren. In solchen Fällen inkonsistenter Ergebnisse muss daher eine entsprechende Korrektur vorgenommen werden. Um einen einheitlichen, flexibel erweiterbaren Rahmen für die Nachbehandlung der Erkennungsergebnisse bereitzustellen, wurde eine wissensbasierte Ergebniskorrektur entwickelt.

Das Wissen über die Abhängigkeiten zwischen Aktivitäten wird als Graph $G_{Nb} = (V_{Nb}, E_{Nb})$ repräsentiert. Aktivitäten werden als Knoten in V_{Nb} repräsentiert, während die Kanten in E_{Nb} des Graphen das sich-gegenseitig-ausschließen von Aktivitäten repräsentieren. Da diese Eigenschaft nicht transitiv wirkt (Beispiel: Sowohl Winken mit rechts als auch Zeigen mit links können nicht gleichzeitig mit beidhändigem Tragen eines Gegenstandes auftreten, sind aber voneinander unabhängig), genügt es, im Graphen für jede aktuell genutzte (erkannte) Aktivität zu prüfen, ob es eine Verbindung zu einer der anderen aktuellen Aktivitäten gibt. Alle auf diese Art zu A benachbarten Aktivitäten werden in einer Menge $N_{GA}(A)$ zusammengefasst, die damit alle bekannten Aktivitäten enthält, die nicht gleichzeitig mit A auftreten können (*Gegenseitiger Ausschluss*). Die entsprechend bereinigte Wahrscheinlichkeit $P_{korr}(A)$ für Aktivität A ergibt sich dann mittels Gl. 6.7.

$$P_{korr}(A) = \frac{P(A)}{\max\left\{1,0\ ;\ P(A) + \sum\limits_{N \in N_{GA}(A)} P(N)\right\}} \qquad [6.7]$$

Als Beispiel für das Vorgehen ist in Abb. 6.21 ein Auszug aus einem verwendeten Graphen dargestellt. Der Ausschnitt zeigt 3 Aktivitäten (von den Knoten dargestellt), die in unterschiedlichen Verhältnis zueinander stehen.

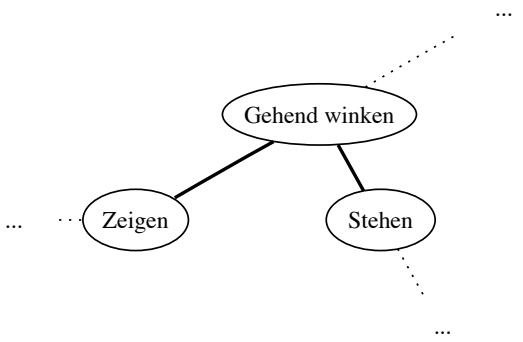


Abb. 6.21.: Auszug aus Nachbehandlungs-Graph, der Informationen über sich gegenseitig ausschließende Aktivitäten (repräsentiert durch starke Kanten) enthält.

GEHEND WINKEN schließt sich gegenseitig sowohl mit ZEIGEN als auch mit STEHEN aus (kenntlich durch die Kante zwischen den beiden Paarungen), aber die beiden letzteren können unabhängig voneinander gleichzeitig auftreten (erkennbar an der fehlenden direkten Ver-

bindung). Dementsprechend ergeben sich die korrigierten Erkennerergebnisse für diese drei Aktivitäten (unter der Annahme, dass keine weiteren Kanten existieren, die beachtet werden müssten):

$$P_{korr}(\text{GEHEND WINKEN}) = \frac{P(\text{GEHEND WINKEN})}{\max\{1,0\ ;\ P(\text{GEHEND WINKEN})+P(\text{ZEIGEN})+P(\text{STEHEN})\}}$$

$$P_{korr}(\text{ZEIGEN}) = \frac{P(\text{ZEIGEN})}{\max\{1,0\ ;\ P(\text{ZEIGEN})+P(\text{GEHEND WINKEN})\}}$$

$$P_{korr}(\text{STEHEN}) = \frac{P(\text{STEHEN})}{\max\{1,0\ ;\ P(\text{STEHEN})+P(\text{GEHEND WINKEN})\}}$$

Der Vorteil dieses Verfahrens liegt in der Lokalität der Korrektur. Da die Menge der zu betrachtenden Aktivitäten/Knoten sich auf die Menge der direkten Nachbarn der in Frage stehenden Aktivität beschränkt, ist keine weitergehende Suche im Hintergrundwissens nötig. Andererseits muss die Summation und die Bestimmung des Maximums in jedem Zeitschritt aus den aktuellen Ergebnissen vorgenommen werden, eine Vorberechnung von Teilergebnissen ist daher nicht möglich.

6.4.6. Einbindung von Hintergrundwissen

In der bisherigen Darstellung der Arbeit war bis hierher an verschiedenen Stellen die Einbindung von Hintergrundwissen in die Aktivitätserkennung thematisiert, allerdings jeweils mit dem Blick auf die jeweils benötigte Information und Verwendungsweise. Dieser Abschnitt fasst das Hintergrundwissen in einer einheitlichen Darstellung zusammen. An zwei Stellen im Klassifikationsprozess (markiert mit (5) in Abb. 6.18 und 6.19) wird Hintergrundwissen eingesetzt:

1. Beim Lernen neuer Aktivitäten wird Wissen benötigt, um geeignete Klassifikatoren und eine geeignete Erkennungsstruktur (direkte Erkennung oder Erkennung in zwei Schritten) auszuwählen.

2. In der Nachbehandlung wird Hintergrundwissen verwendet, um bei der Kombiniation der Erkennung von verschiedenen Aktivitäten die Verträglichkeit zu prüfen und gegebenenfalls Erkennungsergebnisse anzupassen.

Für die Auswahl des geeignetsten Erkenners und seiner Struktur wird Wissen über die *Struktur der Aktivität* benötigt: Aktivitäten können in ihrer Struktur einfach oder komplex sein, eher

statisch (mit wenig Bewegung verbunden) oder dynamisch (mit starker Bewegung verbunden). Das bedeutet, dass an dieser Stelle Daten über einzelne Aktivitäten benötigt werden.

Für die korrigierenden Nachbehandlung wird Wissen über *Verbindungen zwischen Aktivitäten* genutzt, speziell (wie in Abschnitt 6.4.5 beschrieben) die Eigenschaft von Aktivitäten, sich gegenseitig auszuschließen. Das bedeutet, dass hier Daten über Relationen zwischen Aktivitäten benötigt werden.

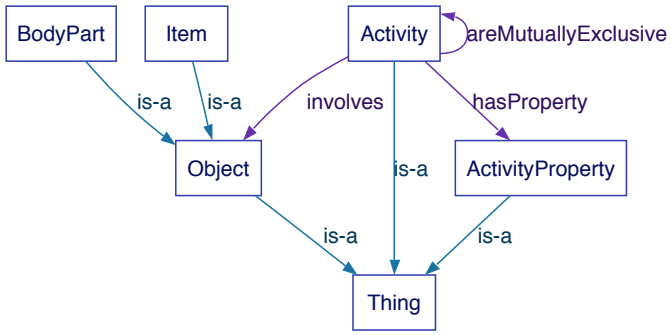


Abb. 6.22.: Graphische Darstellung der Struktur des Hintergrundwissens über Aktivitäten (Inhalt der TBox der dahinterliegenden Ontologie).

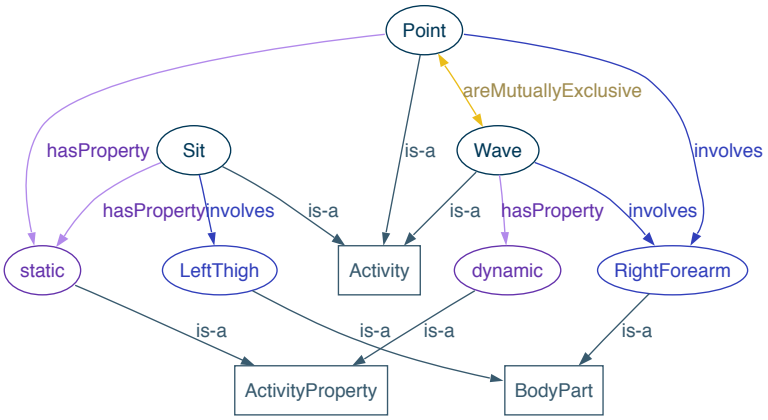


Abb. 6.23.: Auszug aus der ABox (Individuen) zur oben dargestellten TBox der Aktivitäts-Ontologie.

Alle diese benötigten Daten können gemeinsam in einer strukturierten Wissenbasis hinterlegt werden. Das Wissen ist in einer *Ontologie* organisiert, der eine Beschreibungslogik [Baader, 2003] als Formalismus zugrundeliegt. Abb. 6.22 zeigt den Inhalt der zugehörigen TBox in graphischer Darstellung, die die Struktur der Klassen und Relationen zwischen ihnen vorgibt. Die drei Hauptklassen repräsentieren Aktivitäten (als Activity), Objekte allgemein (als Object), die als Unterklassen wiederum Körperteile (als BodyPart) und Gegenstände (als Item) haben, und Eigenschaften von Aktivitäten (als ActivityProperty). Die Klassen werden ergänzt von drei Relationstypen: areMutuallyExclusive verbindet zwei Aktivitäten, und stellt damit das Wissen dar,

das für die Ergebnis-Nachbehandlung benötigt wird. Die Relation hasProperty verbindet eine Aktivität mit Aktivitätseigenschaften, involves mit einem Objekt, so dass diese beiden Relation Informationen für die Auswahl von geeigneten Klassifikatoren darstellen.

Ein Ausschnitt des darauf aufgebauten Wissens (d.h. ein Auszug der ABox) ist in Abb. 6.23 dargestellt. Das Beispiel zeigt drei Aktivitäten (Zeigen als Point, Winken als Wave, Sitzen als Sit), zwei Aktivitätseigenschaften (statisch als static und dynamisch als dynamic) sowie zwei Körperteile (rechter Unterarm als RightForearm und linker Oberschenkel als RightThigh) als examplarische Individuen mit den zwischen diesen Individuen gültigen Relationen: Winken und Zeigen (gemeint ist hier jeweils die entsprechende Aktivität mit dem rechten Arm, der Übersicht halber wird dieses Detail nicht dargestellt) schließen sich gegenseitig aus, beide involvieren den rechten Unterarm, Sitzen dagegen involviert den linken Oberschenkel. Zeigen und Sitzen haben die Eigenschaft, eher statisch zu sein, während Winken eine dynamische Aktivität darstellt.

Die Verwendung des Wissens aus der Ontologie für die konkreten Anwendungen geschieht durch Abfrage der benötigten Datenauszüge (quasi eine Projektion des gesamten Wissen auf den benötigten Teil), wobei der Zugriff über die Bezeichnung der benötigten Aktivitäten erfolgt. Bezugnehmend auf das in Abb. 6.23 dargestellte Beispiel, könnte die Ergebnis-Nachbehandlung beispielsweise auf der Individuum Point zugreifen, und unter allen über die Relation areMutually-Exclusive verbundenen Aktivitäten prüfen, ob diese auch aktuell verwendet werden und eine entsprechende Bearbeitung notwendig ist. Analog kann beim Trainieren einer Aktivität, über die Wissen in der Wissensbasis vorliegt, über den Namen der Aktivität das Wissen bezüglich der Eigenschaften der Aktivität (über die Relation hasProperty) und über die in der Aktivität involvierten Objekte (über die Relation involves) gefunden werden für die Auswahl eines geeigneten Erkenners.

An dieser Stelle ist auch eine günstige Möglichkeit zur Behandlung von synonymen Bezeichnungen für Aktivititäten gegeben, indem in der Ontologie eine Aktivität mit der synonymen Bezeichnung und als zusätzliches Fakt die Gleichheit dieser Aktivität mit der schon existierenden, synonymen Aktivität eingefügt wird. Die Verknüpfungen der so neu eingefügten Instanz einer Aktivität können dann durch automatische Inferenz bestimmt werden, ohne fehleranfällige manuelle Eingriffe.

7. Experimente & Evaluation

In diesem Kapitel wird das vorgestellte Konzept zur Aktivitätserkennung und seine Leistungsfähigkeit anhand verschiedener Experimente validiert und evaluiert. Die Resultate werden in vier Abschnitten präsentiert. Abschnitt 7.1 stellt die Evaluation der Erweiterungen der Personenbeobachtung dar, die folgenden Abschnitte präsentieren die Evaluation des Gesamtsystems anhand der drei Teilprozessketten zur Erschließung neuer Anwendungsdomänen (in Abschnitt 7.2), zum Lernen neuer Erkenner (in Abschnitt 7.3) und zur eigentlichen Erkennung von Aktivitäten (in Abschnitt 7.4). Abschnitt 7.5 schließlich fasst die in diesem Kapitel präsentierten Ergebnisse zusammen. Darüberhinaus sind einige zusätzliche Resultate der Evaluation einzelner Komponenten in Anhang C zusammengestellt.

7.1. Evaluation von Verbesserungen der Bewegungsbeobachtung

7.1.1. Evaluation der Modellinitialisierung

Die neu entwickelte Initialisierung für ICP-basierte Trackingsysteme (beschrieben in Abschnitt 5.2) wird in Form des Trackingsystems *VooDoo* in autonomen Serviceroboter-Experimenten mit dem Roboter ALBERT II verwendet. In diesem Rahmen wird der Algorithmus erfolgreich und regelmäßig von etwa 10 verschiedenen Personen unterschiedlicher Körpergröße eingesetzt (in einem Größenintervall von ungefähr 160cm bis ungefähr 185cm).

Tab. 7.1 zeigt die Ergebnisses einer systematischen Evaluation des Ansatzes, bei der 4 Personen unterschiedlicher Statur (Größe und Figur) mehrfach in den Bereich der Sensorbeobachtung gebracht und die Ergebnisse der Initialisierung aufgezeichnet wurden. Die Tabelle zeigt der Vergleich der wahren Größe der Personen und die jeweiligen Ergebnisse der automatischen Initialisierung.

Im Durchschnitt wurde dabei eine Abweichung von $-26,21\,\text{cm}$ erreicht, mit einem 1-Quartil von $-42,75\,\text{cm}$ und einem 3-Quartil von $-9,05\,\text{cm}$, d.h. in die Initialisierung generiert im Normalfall ein Modell, das kleiner ist als die beobachtete Person. Da der Tracking-Algorithmus in der Praxis besser funktioniert wenn das benutzte Modell etwas kleiner als die getrackte Person ist, stellt dies eine wünschenswerte Eigenschaft dar. Abb. 7.1 stellt den Vergleich der Größenschätzung mit der realen Größe der betrachteten Individuen graphisch wider.

Tab. 7.1.: Ergebnisse einer Anzahl von Testläufen der Modell-Initialisierung für das *VooDoo*-Trackingsystem. Aufgezeichnet wurden die Resultate der Initialisierung von 4 Testpersonen in jeweils 8 Testläufen (unter identischen Bedingungen). Zum Vergleich sind die realen Größen der Testpersonen mit angegeben.

		Größenschätzung [cm]								
Person	**Größe** [cm]	1	2	3	4	5	6	7	8	Ø
1	173.0	140.4	177.3	172.9	179.8	183.5	173.3	141.1	151.8	165.01
2	175.0	132.7	137.5	146.8	159.9	130.3	128.2	135.0	151.4	140.23
3	182.0	173.4	181.5	150.5	179.5	137.9	151.8	172.8	150.0	162.18
4	191.0	166.3	145.1	143.5	168.9	143.2	146.1	119.2	157.8	148.76

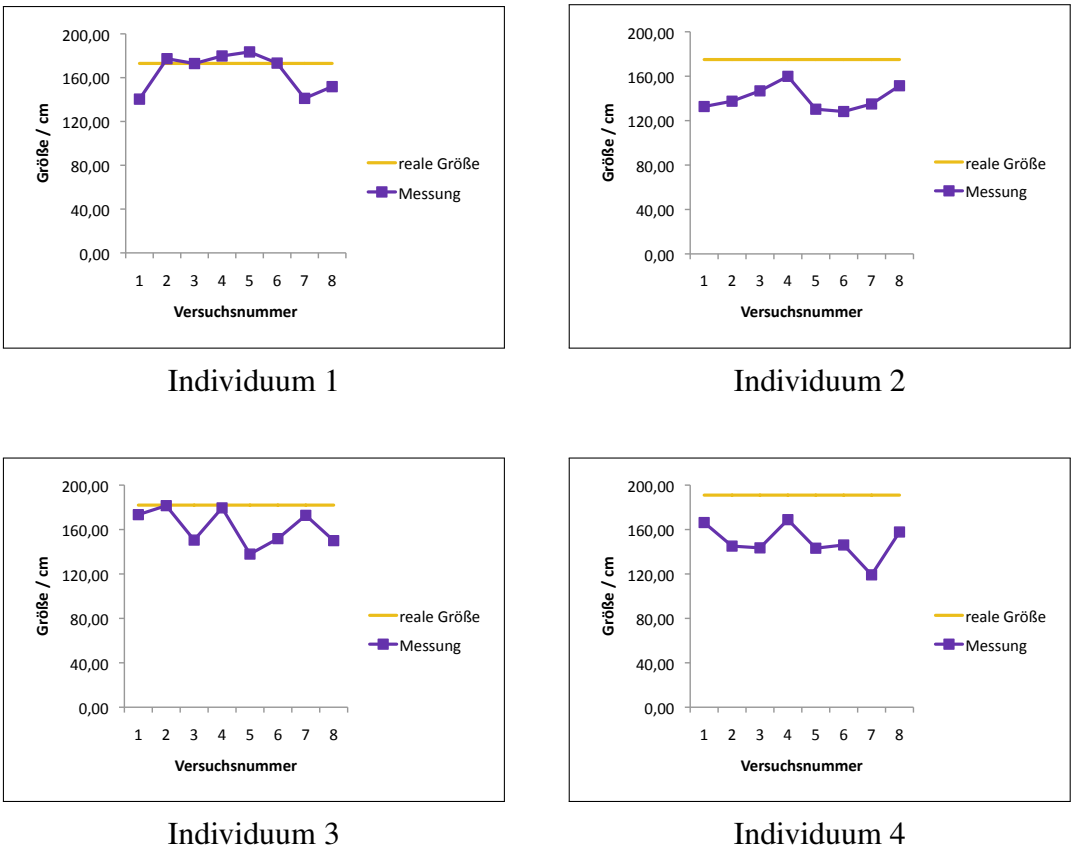
Da diese Daten ohne spezielle Vorbereitungen bezüglich Beleuchtungsbedingungen oder anderer Parameter gesammelt wurden, spiegeln diese Daten das Verhalten des Systems unter realen Bedingungen dar.

Aus qualitativer Sicht arbeitet das Verfahren für beliebige Individuen ohne weitere Personalisierung der Parameter, solange die Systemparameter (Position des Sensors, die für die Schätzung der Größe des Menschen benötigt wird) korrekt bekannt sind. Das System initialisiert eine Person im Sichtbereich im Schnitt nach etwa $15 - 20$s. Es ist dabei wichtig anzumerken, dass sich die Person bewegen kann ohne die Initialisierung zu verhindern, solange die Bewegungen nicht zu schnell sind und der Mensch dabei im Sichtbereich des Sensors bleibt. Insbesondere ist damit eine Initialisierung möglich, während die beobachtete Person auf die Kamera zugeht.

7.1.2. Evaluation der Gelenkwinkelgrenzen in *VooDoo*

Die Evaluation der in Abschnitt 5.3 präsentierten Gelenkwinkelgrenzen-Modellierung wurde auf einem Standard-PC unter Verwendung eines SwissRanger 3000-Tiefensensors durchgeführt. Sowohl ohne als auch mit Verwendung der Gelenkwinkelgrenzen läuft das System mit etwa 20Hz. Die annähernd gleiche Laufzeit resultiert aus der unaufwändigen Berechnung einer nur geringen Zahl zusätzlicher Messpunkte, die generiert werden müssen. Da das ICP-Tracking linear mit der Anzahl der Messpunkte skaliert [Knoop et al., 2006a], fällt die Verwendung der zusätzlichen Messpunkte nicht ins Gewicht.

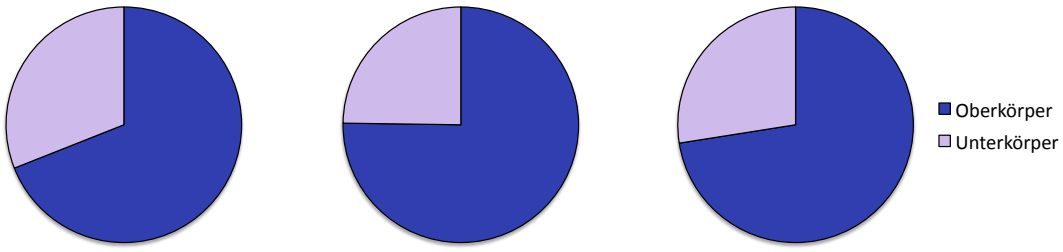
Das Verfahren wurde anhand 12 aufgezeichneter Bewegungssequenzen unterschiedlicher Länge evaluiert, um eine identische Datenbasis für den Vergleich des Trackings mit und ohne Verwendung der Gelenkwinkelgrenzen zur Verfügung zu haben. Die Sequenzen 4, 5, 6 und 7 beinhalten hauptsächlich Bewegungen des Oberkörpers, die Sequenzen 8, 9 und 12 haupt-



Individuum 1 Individuum 2

Individuum 3 Individuum 4

Abb. 7.1.: Graphischer Vergleich der aus der Initialisierung gewonnenen Größenschätzung mit der realen Größe der getesteten vier Individuen.



(a) (b) (c)

Abb. 7.2.: Verteilung der Gelenkwinkelgrenzen-Verletzungen zwischen Oberkörper und Unterleib: (a) Testsequenz 2 (b) Testsequenz 6 (c) Testsequenz 8.

sächlich Bewegungen der Beine. Die übrigen Sequenzen 1, 2, 3, 10 und 11 zeigen Bewegungen des ganzen Körpers. Der leichte Fokus auf Bewegungen des Oberkörpers resultiert aus der Erkenntnis, dass die meisten Verletzungen der Bewegungsgrenzen am Oberkörper auftreten. Dies wird anhand der Diagramme in Abb. 7.2 deutlich, der Oberkörper ist in allen Sequenzen für mindestens doppelt so viele Fehlstellungen verantwortlich wie der Unterleib.

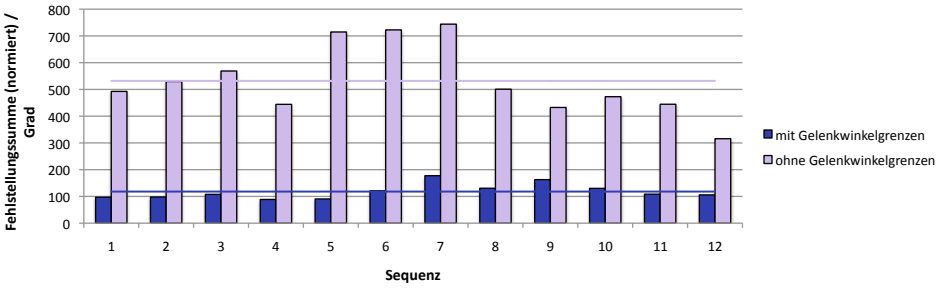


Abb. 7.3.: Normalisierte Summe der Fehlstellungen für jede Testsequenz ohne und mit Verwendung der Gelenkwinkelgrenzen-Korrektur.

Zwei Gründe sind hierfür verantwortlich. Zum einen sind die Arme, insbesondere die Unterarme, meist dünner als die Beine, sodass weniger Messpunkte auf den Armen als auf den Beinen von der Sensorik generiert werden können. Daher ist das Tracking der Arme generell unzuverlässiger als das der Beine. Zum anderen sind die Bewegungen der Arme in der Praxis erheblich flexibler als die der Beine, da die Arme eine offene kinematische Kette darstellen, während die Beine in den meisten Bewegungen durch die Position der Füße auf dem Boden erheblich in ihrer Beweglichkeit eingeschränkt sind.

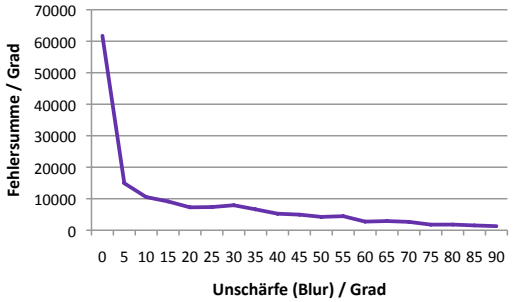


Abb. 7.4.: Einfluss des Unschärfeparameters β_G auf die Summe der Abweichungen bei Grenzwinkelverletzungen.

Mit Hilfe der in Abschnitt 5.3.4 präsentierten Gelenkwinkelgrenzen wurden die Winkelfehlstellungen in jedem Trackingschritt bestimmt. Abb. 7.3 zeigt die summierten Fehler mit und ohne Verwendung der Gelenkwinkelgrenzen, normalisiert mit der Länge der Sequenzen. Die

horizontalen Linien markieren den jeweiligen mittleren Fehler. Wie erwartet, fällt der summierte Fehler deutlich, von einem mittleren Fehler von 532° ohne Verwendung der Gelenkwinkelgrenzen auf einen mittleren Fehler von 118° bei Verwendung der Gelenkwinkelgrenzen, was einer Verbesserung um einen Faktor von annähernd $4,5$ entspricht.

Die Ergebnisse wurden erzieht mit einem Unschärfefaktor $\beta_G = 5°$. Dieser Wert wurde durch Evaluation des Effekts verschiedener Werte für diesen Faktor auf alle Testsequenzen gewählt. Das Resultat dieser Auswertung der Ergebnisse bei verschiedenen Parameterwerten ist in Abb. 7.4 dargestellt. Wie deutlich zu erkennen ist, zeigen schon kleine Werte für β_G große Verbesserungen. Die Ergebnisse für $\beta_G \geq 55°$ sind nicht mehr wirklich verlässlich, da bei dieser großen Freiheit das Tracking ab und zu das Modell verlieren kann. In solchen Fällen stellt die Summe der Fehlstellungen keine vollständige Abbildung der Fehler des Trackings mehr dar.

Insgesamt zeigen diese Ergebnisse, dass das entwickelte Verfahren die auftretenden Fehlstellungen drastisch reduziert. Diese verbesserte Korrektheit der Ergebnisse erlaubt damit insbesondere eine robuste Verwendung des Trackings für ein darauf aufbauendes, modell-basiertes Aktivitätserkennungssystem, das darauf angewiesen ist, dass ähnliche Posen eines Menschen auch immer ähnliche Ergebnisse des Trackings zur Folge haben (und beispielsweise keine häufig auftretenden, systematischen Verwechslungen von Posen einzelner Körperteile auftreten).

7.2. Erschließung neuer Anwendungsdomänen

Dieser Abschnitt beschreibt die experimentelle Evaluation der Erschließung neuer Anwendungsdomänen durch die automatische Exploration von geeigneten Merkmalen. Zunächst werden in Abschnitt 7.2.1 die Randbedingungen der Evaluation dargestellt, gefolgt von der Beschreibung des Ablaufs der Evaluation und Details zu den gemessenen Größen in Abschnitt 7.2.2. Schließlich werden die Ergebnisse des Experiments und ihre Bedeutung in Abschnitt 7.2.3 diskutiert.

7.2.1. Evaluationsdomänen

Die Anwendungsdomäne, die in der Evaluation erschlossen werden soll (und in den folgenden Abschnitten auch zur Evaluation der weiteren Prozessschritte eingesetzt wird) ist die Beobachtung von Ganzkörperbewegungen, wie sie für das Lernen von Missionsmodellen beispielsweise bei Schmidt-Rohr [Schmidt-Rohr et al., 2010] eingesetzt wird.

Interessante Aktivitäten, die erkannt werden sollen, sind in dieser Anwendung einerseits Gesten aus dem Bereich der Interaktion (beispielsweise Winken), andererseits Aktivitäten, die ein Roboter nachahmen soll (wie beispielsweise nehmen oder ablegen von Gegenständen), und

schließlich direkt für die Kommandierung benötigte Aktionen wie beispielsweise Zeigegesten. Zusätzlich werden einige Aktivitäten verwendet, die in betrachteten Küchen- und Cateria-Szenarien häufig auftreten und daher erkannt werden sollten, wie beispielsweise Trinken und Sitzen. Schließlich wurden noch einige Aktivitäten betrachtet, die in den genannten Szenarien keine Bedeutung haben, aber durch die Art ihrer Ausführung zusätzliche Bewegungsaspekte abdecken, die von den anderen gewählten Aktivitäten nicht abgedeckt werden. Darunter fallen insbesondere die Aktivitäten FLIEGEN (ein gleichzeitiges, synchrones Heben und Senken beider Arme), KICKEN (eine Kickbewegung wie beim Treten eines Fussballes) und die beiden TANZBEWEGUNG1 und TANZBEWEGUNG2, die zwei unterschiedliche Arten einer synchronisierten Bewegung von beiden Armen und Oberkörper darstellen.

7.2.2. Durchführung der Evaluation

Eine quantitative Bewertung der Merkmale, die in der Domänen-Erschließung generiert werden, ist nicht direkt möglich. Daher wird hier auf eine indirekte Messung ihrer Qualität zurückgegriffen, indem mit Hilfe dieser Merkmale die Erkennung für eine Reihe von Aktivitäten trainiert und ihre Erkennungsqualität gemessen wird. Dieser Schritt wird sowohl mit Aktivitäten, von denen Daten für die Generierung der Merkmale genutzt wurden, als auch für neue Aktivitäten durchgeführt. Der vollständige Ablauf der Evaluation ist in Abb. 7.5 dargestellt, insbesondere auch die Aufteilung der aufgezeichneten Daten zur Verwendung in den verschiedenen Schritten. Nur von einem Teil der aufgezeichneten Aktivitäten werden Daten für die Suche nach relevanten Merkmalen genutzt, jeweils etwa die Hälfte der zu diesen Aktivitäten gehörigen Sequenzen werden dafür verwendet. Diese Aufteilung der Daten ist in der Abbildung konzeptionell als Tortendiagramm dargestellt.

Das Training und die Evaluation der Erkennungsergebnisse wird sowohl mit den aus der Exploration resultierenden Merkmalen, als auch mit einer Menge von 320 manuell definierten Merkmalen durchgeführt, deren Ergebnisse als Vergleichspunkt dienen. Weitere Details zu dieser Vergleichsmerkmalsmenge sind in Anhang B.1 zu finden.

Eine vollständige Liste der genutzten Aktivitäten zeigt Tabelle 7.2. Aus dieser Aktivitätsmenge wurden KICKEN (mit links), IN DIE HOCKE GEHEN, AUS HOCKE AUFSTEHEN, HOCKEN, WINKEN (mit beiden Armen), bestimmte ZEIGEN-Varianten (eine mit rechts, eine mit links), GEGENSTAND WEGSTELLEN, TANZBEWEGUNG1, TANZBEWEGUNG2 und FLIEGEN *nicht* als Daten für die Exploration eingesetzt, sondern nur zur Evaluation genutzt (in Training und Erkennung). Jeweils die Hälfte der Sequenzen wird für das Training verwendet, die Trainingsdaten und die andere Hälfte der Sequenzen für die Evaluation der Erkennung. Die für die Explora-

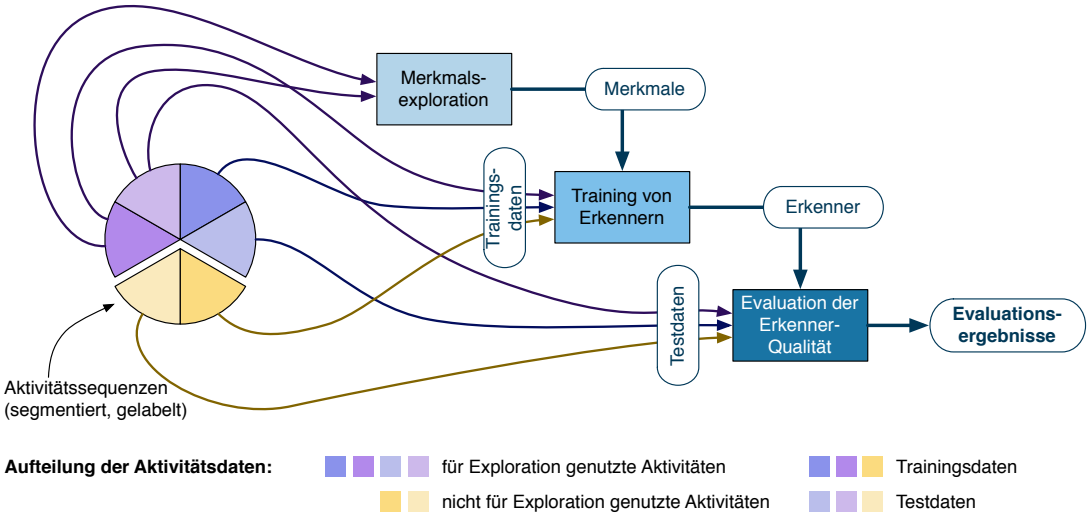


Abb. 7.5.: Überblick der Hauptschritte zur Evaluation der Merkmalsexploration. Die Aufteilung der dabei verwendeten Daten ist durch das Tortendiagramm links und die Verwendung in den einzelnen Schritten konzeptionell repräsentiert.

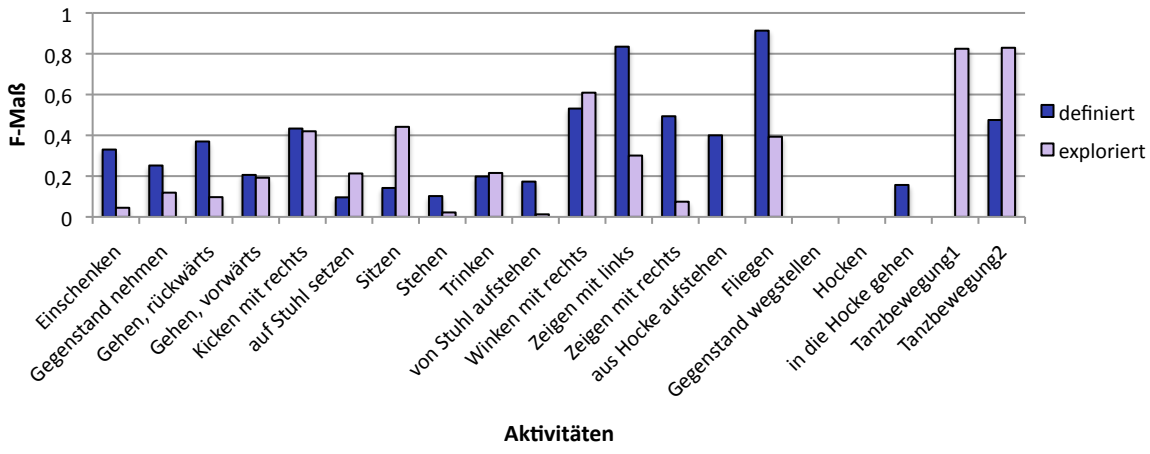


Abb. 7.6.: Vergleich der Erkenner für verschiedene Aktivitäten, trainiert auf unterschiedlichen Merkmalen (einmal manuell definiert, einmal exploriert), anhand des F-Maßes.

Tab. 7.2.: Liste der für die Evaluation der Merkmalsexploration genutzten Aktivitäten, mit Details zur Anzahl der aufgezeichneten Einzelsequenzen und der Gesamtzahl der Frames pro Aktivität (alphabetisch geordnet).

Bezeichnung	**Varianten**	**#Sequenzen**	**#Frames**
AUF STUHL SETZEN		3	186
AUS HOCKE AUFSTEHEN		2	50
EINSCHENKEN		4	240
FLIEGEN		1	263
GEGENSTAND NEHMEN	mit rechts	3	133
GEGENSTAND WEGSTELLEN		1	47
GEHEN	vorwärts, rückwärts	6	480
HOCKEN		2	47
IN DIE HOCKE GEHEN		2	99
KICKEN	mit rechts, mit links	13	398
SITZEN		3	514
STEHEN		7	244
TANZBEWEGUNG1		1	317
TANZBEWEGUNG2		4	111
TRINKEN		4	264
VON STUHL AUFSTEHEN		3	96
WINKEN	linker Arm, rechter Arm, beide Arme	9	614
ZEIGEN	mit links, mit rechts, verschiedene Zielpunkte	9	366

tion eingesetzten Sequenzen und die für Training/Evaluation verwendeten Sequenzen sind zur Hälfte identisch, zur Hälfte komplementär gewählt.

7.2.3. Evaluationsergebnisse

Generiert wurden 32.820 Merkmale, aus denen abhängig von den verwendeten Parametern für die Exploration unterschiedlich viele potentiell relevante Merkmale bestimmt wurden. Für eine detaillierte Untersuchung des Zusammenhangs zwischen Parameterwahl und Größe der resultierenden Merkmalsmenge sei auf Anhang C.2 verwiesen. Für die folgenden Ergebnisse wurde eine Menge von 10.050 Merkmalen benutzt, die einen guten Kompromiss darstellen (bestimmt mit den Parametern $\theta_{Me} = 8,6$ und $n_{mR} = 10$). Diese wurden auf die oben beschriebene Art genutzt für das Lernen und spätere Erkennen verschiedener Aktivitäten, und die Ergebnisse dieser Erkennung wurden den Ergebnissen bei der Verwendung mit einer manuell vordefinierten Menge von 320 Merkmalen gegenübergestellt.

Die naheliegende Wahl einer Konfusionsmatrix zur Darstellung der Erkennungsergebnisse verbietet sich, weil in einigen Sequenzen mehr als eine Aktivität gleichzeitig auftreten (beispielsweise Winken während des Gehens). Dadurch ist keine einfache 1-zu-1-Zuordnung von Sequenzen zu Aktivitäten möglich. Stattdessen zeigt Tabelle 7.3 einen Vergleich zwischen der Erkennungsqualität bei Verwendung der alten (manuell definierten) und der neuen (automatisch bestimmten) Merkmalsmenge anhand der zwei Maßzahlen *Genauigkeit* (engl. *precision*) und *Trefferquote* (engl. *recall*), definiert in den Gleichungen 7.1 und 7.2. Dabei steht r_p für korrekte positive Erkennungen, f_p für inkorrekte positive Erkennungen und f_n für inkorrekte negative Erkennungen.

$$\text{Genauigkeit} = P(\text{korrekt positiv erkannt}|\text{positiv erkannt}) = \frac{r_p}{r_p + f_p} \quad [7.1]$$

$$\text{Trefferquote} = P(\text{positiv erkannt}|\text{tatsächlich positiv}) = \frac{r_p}{r_p + f_n} \quad [7.2]$$

$$F = 2 \cdot \frac{\text{Genauigkeit} \cdot \text{Trefferquote}}{\text{Genauigkeit} + \text{Trefferquote}} \quad [7.3]$$

Um eine übersichtlichere Diagrammdarstellung des Vergleichs zu erlauben, werden Genauigkeit und Trefferquote nach Gleichung 7.3 zum *F-Maß F* (engl. F_1*-score*) kombiniert. Der Vergleich der Erkennungsergebnisse bei der Verwendung des manuell definierten gegenüber der Verwendung der automatisch generierten Merkmale ist in Abb. 7.6 dargestellt.

Tab. 7.3.: Vergleich von Trefferquote und Genauigkeit der Erkennung von Aktivitäten mit Klassifikatoren trainiert auf einer manuell definierten Merkmalsmenge und der mittels automatischer Exploration bestimmten Merkmalsmenge. Die Aktivitäten im oberen Teil der Tabelle wurden auch zur Exploration von Merkmalen genutzt, die Aktivitäten im unteren Teil nicht. Hinweis: '-' in der Spalte für die Genauigkeit steht für nicht ermittelbare Werte in Fällen, in denen $r_p = f_p = 0$ gilt.

Aktivität	**Alte Merkmalsmenge**		**Neue Merkmalsmenge**	
	Trefferquote	*Genauigkeit*	*Trefferquote*	*Genauigkeit*
AUF STUHL SETZEN	0,3172	0,0565	0,4839	0,1366
EINSCHENKEN	0,3338	0,3267	0,0311	0,0801
GEGENSTAND NEHMEN MIT RECHTS	0,5888	0,1603	0,2617	0,0769
GEHEN, RÜCKWÄRTS	0,5377	0,2818	0,6462	0,0524
GEHEN, VORWÄRTS	0,1183	0,7904	0,119	0,4972
KICKEN MIT RECHTS	0,5563	0,3548	0,6298	0,3149
SITZEN	0,284	0,0946	0,4669	0,4188
STEHEN	0,3852	0,0589	0,0123	0,1
TRINKEN	0,2311	0,1738	0,3068	0,166
VON STUHL AUFSTEHEN	0,1186	0,3202	0,0064	0,5833
WINKEN MIT RECHTS	0,4049	0,7719	0,681	0,5509
ZEIGEN MIT LINKS	0,7162	1,0	0,25	0,3776
ZEIGEN MIT RECHTS	0,5275	0,4637	0,0505	0,1429
AUS HOCKE AUFSTEHEN	0,28	0,7	0,0	-
FLIEGEN	0,8403	1,0	0,3954	0,391
GEGENSTAND WEGSTELLEN	0,0	0,0	0,0	-
HOCKEN	0,0	0,0	0,0	0,0
IN DIE HOCKE GEHEN	0,1313	0,194	0,0	0,0
KICKEN MIT LINKS	0,2071	0,0455	0,4852	0,1966
TANZBEWEGUNG1	0,0	0,0	0,8738	0,7803
TANZBEWEGUNG2	0,6983	0,36	0,7328	0,9551

Allgemein muss zunächst darauf hingewiesen werden, dass die Erkennungsqualität in diesen Evaluationsläufen relativ schlecht ist, weil für diesen Vergleich keine der für gute Ergebnisse nötigen Verbesserungen bei der Auswahl von Aktivitäts-spezifischen Merkmalen oder bei der Durchführung der Erkennung verwendet wurde, um möglichst unverfälscht den Einfluss der Basis-Merkmalsmenge zu untersuchen. Dabei ist zu erkennen, dass tendenziell die Erkennung bei den Aktivitäten, die nicht in der Exporation genutzt wurden, etwas schlechter ist.

Auch wenn die Ergebnisse mit den aus der Exploration gewonnenen Merkmalen Stärken und Schwächen zeigen und bei der Erkennung einzelner Aktivitäten schlechter sind als mit den manuell definierten Merkmalen, so muss der Vorteil des automatischen Ansatzes betont werden. Die manuelle Definition der Merkmale erfordert Zeit, Verständnis für die betrachtete Domäne, und Erfahrung mit dem System. Dagegen ist für die automatische Bestimmung von potentiell relevanten Merkmalen nur die Bereitstellung von Trainingsdaten nötig, wie sie in der eigentlichen Anwendung für der Lernen spezifischer Aktivitäten sowieso benötigt werden.

7.3. Einlernen neuer Aktivitäten

Das im Folgenden vorgestellte Experiment evaluiert den Lernteil des Prozesses, in dem neue Aktivitäten in das System eingelernt werden. Das Experiment findet in der in Abschnitt 7.2.1 vorgestellten Evaluationsdomäne statt. Der genaue Ablauf der Evaluation ist in Abschnitt 7.3.1 beschrieben, die Ergebnisse werden im folgenden Abschnitt 7.3.2 präsentiert und interpretiert.

7.3.1. Durchführung der Evaluation

Für das Experiment wurden die in Tab. 7.4 aufgelisteten Datensätze verwendet. Als Gütemaß für das Ergebnis des Trainings dient wiederum die Qualität der Erkennung, die mit den neu gelernten Erkennern erreicht werden kann, gemessen anhand Genauigkeit und Trefferquote sowie dem daraus kombinierten F-Maß (definiert in den Gleichungen 7.1-7.3).

Um mit diesem Gütemaß den Trainingsprozess zu untersuchen, wurden für die Aktivitäten Klassifikatoren trainiert auf verschiedenen Merkmalsmengen, zu deren Auswahl die entwickelten Verbesserungen des Trainingsprozesses genutzt wurden. Für jede Aktivität wurden verschiedene relevante Merkmalsteilmengen durch sukzessive Verwendung von weiteren Prozesserweiterungen ausgewählt:

- *alle Merkmale:* Die aktive Auswahl von relevanten Merkmalen erfolgt auf der vollen Grundmerkmalsmenge. Alle von dem Algorithmus als relevant betrachteten Merkmale werden für das Training verwendet.

Tab. 7.4.: Liste der für die Evaluation des Trainings genutzten Aktivitäten, mit Details zur Anzahl der genutzten Merkmale in den drei verglichenen Varianten, sowie den als Hintergrundwissen verwendeten involvierten Körperteile bei der Aktivität (alphabetisch geordnet).

Bezeichnung	**Körperteile**	**#Merkmale** ***alle***	**#Merkmale** ***vorausgewählt***	**#Merkmale** ***interaktiv***
AUF STUHL SETZEN	Unterleib und Torso	44	20	13
EINSCHENKEN	Oberkörper mit Kopf	39	32	20
FLIEGEN	Arme	57	33	23
GEGENSTAND NEHMEN	Arme	35	16	4
GEHEN, RÜCKWÄRTS	Unterleib	39	17	13
GEHEN, VORWÄRTS	Unterleib	55	27	20
IN DIE HOCKE GEHEN	Unterleib und Torso	43	17	12
KICKEN MIT RECHTS	Unterleib	47	26	19
SITZEN	Unterleib und Torso	52	26	19
STEHEN	Unterleib	33	9	8
TANZBEWEGUNG2	Oberkörper	32	26	19
TRINKEN	Oberkörper mit Kopf	48	34	24
VON STUHL AUFSTEHEN	Unterleib und Torso	27	20	14
WINKEN	Oberkörper mit Kopf	31	25	22
ZEIGEN MIT LINKS	Arme und Kopf	33	24	15
ZEIGEN MIT RECHTS	Arme und Kopf	62	37	21

- *vorausgewählte Merkmale:* Die aktive Auswahl erfolgt einer einer Merkmalsmenge, die durch den Einsatz von Hintergrundwissen reduziert wurde. Für dieses Experiment wurde dazu Wissen über die an einer Aktivität beteiligten Körperregionen genutzt, wie es in Tab. 7.4 annotiert ist.

- *interaktiv verfeinerte Merkmale:* In Interaktion mit dem Benutzer werden die durch Hintergrundwissen beschränkten Merkmalsmengen noch weiter verfeinert.

Mit den so gewählten Merkmalen wurden einfache Erkenner (nicht zusammengesetzt, Verwendung nur einer Ebene) trainiert. Jeder dieser Erkenner wurde zur Erkennung der Trainingsdaten und zusätzlicher, im Training nicht verwendeter Datensätze genutzt, die Resultate ausgewertet anhand der oben beschriebenen Maßzahlen.

7.3.2. Evaluationsergebnisse

Die Zahl der für das Training (und später die Erkennung) verwendeten Merkmale nimmt deutlich ab, wenn Hintergrundwissen oder zusätzlich sogar noch die Benutzerinteraktion im Auswahlprozess eingesetzt wird, wie das Diagramm in Abb. 7.7 deutlich zeigt. Im Mittel werden nur 0,4-mal so viele Merkmale bei der voll Verwendung des vollen Prozesses gewählt wie bei Verwendung nur der aktiven Merkmalsauswahl.

Die Ergebnisse der mit den verschiedenen Merkmalsmengen trainierten Erkenner sind in Tab. 7.5 aufgelistet. Die Auflistung ist alphabetisch geordnet und zeigt die Genauigkeit und die Trefferquote, die jeder Klassifikator auf den Evaluationsdaten erreichen konnte. Eine etwas übersichtlichere Darstellung der Ergebnisse zeigt Abb. 7.8, in der das kombinierte F-Maß der Erkenner als Balkendiagramm gezeigt ist.

Wie man in dem Diagramm sieht, sind die Ergebnisse mit den Merkmalen, die unter Nutzung von Hintergrundwissen und Interaktion gewählt wurden, zwar teilweise etwas schlechter als bei der Nutzung von mehr Merkmalen, aber in vielen Fällen auch besser, teilweise sogar sehr deutlich. Ein klares Beispiel für diesen Fall ist das ZEIGEN MIT RECHTS.

Die Ergebnisse bei der Erkennung der Aktivitäten IN DIE HOCKE GEHEN und VON STUHL AUFSTEHEN sind sehr schlecht, wweil gerade diese Aktivitäten, für die darüberhinaus nur relativ wenig Trainingsdaten vorlagen, Verwechslungen miteinander und mit anderen Aktivitäten leicht möglich sind.

Ein aus den Messungen der reinen Erkennungsqualität nicht ersichtlicher Vorteil ist die Aufwandsreduktion, die mit einer kleineren Zahl von Merkmalen einhergeht. Durch diese Dimensionsreduktion ist ein schnelleres Training möglich, und auch die Laufzeit während der Erkennung ist geringer, obwohl sich die Laufzeit der Erkennung in bisherigen Testläufen auf verschie-

Tab. 7.5.: Vergleich von Trefferquote und Genauigkeit der Erkennung von Aktivitäten mit Klassifikatoren trainiert auf unterschiedlichen Merkmalsmengen resultierend aus der Merkmalsauswahl: Aktive Auswahl aus der vollständigen Merkmalsmenge, Auswahl auf einer durch Hintergrundwissen reduzierten Menge, und schließlich aus der noch zusätzlich durch Benutzerinteraktion verfeinerten Merkmalsmenge.

Aktivität	**Aktive Auswahl**		**Aktive & Passive Auswahl**		**Aktive, Passive & Interaktive Auswahl**	
	Trefferquote	*Genauigkeit*	*Trefferquote*	*Genauigkeit*	*Trefferquote*	*Genauigkeit*
AUF STUHL SETZEN	0,4839	0,1366	0,2581	0,1179	0,2204	0,0998
EINSCHENKEN	0,0311	0,0801	0,3203	0,0932	0,0796	0,1878
GEGENSTAND NEHMEN MIT RECHTS	0,2617	0,0769	0,0561	0,1538	0,028	0,1071
GEHEN, RÜCKWÄRTS	0,6462	0,0524	0,0943	0,0088	0,3301	0,0414
GEHEN, VORWÄRTS	0,119	0,4971	0,0518	0,381	0,0651	0,8162
FLIEGEN	0,3954	0,391	0,0	0,0	0,3194	0,2569
IN DIE HOCKE GEHEN	0,0	0,0	0,0	0,0	0,0909	0,0224
KICKEN MIT RECHTS	0,6298	0,3149	0,1799	0,1825	0,2768	0,2589
SITZEN	0,4669	0,4188	0,4358	0,1272	0,4261	0,1391
STEHEN	0,0123	0,1	0,1025	0,0231	0,4426	0,0774
TANZBEWEGUNG2	0,7328	0,955	0,5776	0,0812	0,6379	0,0955
TRINKEN	0,3068	0,166	0,0833	1,0	0,2386	0,7778
VON STUHL AUFSTEHEN	0,0064	0,5833	0,0	0,0	0,0146	0,3556
WINKEN MIT RECHTS	0,681	0,5509	0,6656	0,5564	0,9294	0,6222
ZEIGEN MIT LINKS	0,25	0,3776	0,8851	0,0774	0,9189	0,0885
ZEIGEN MIT RECHTS	0,0504	0,1429	0,2431	0,7067	0,3257	0,7396

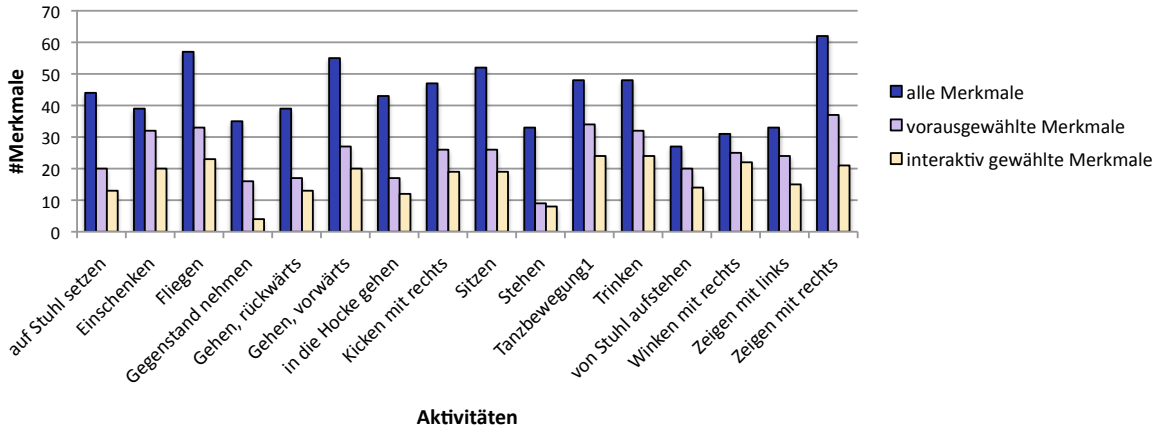


Abb. 7.7.: Vergleich der Größe der genutzten Merkmale für die einzelnen Evaluations-Aktivitäten beim Einsatz nur der aktiven Merkmalsauswahl (als *alle* bezeichnet in der Legende), der zusätzlichen Nutzung von Hintergrundwissen zur Einschränkung der Grundmerkmalsmenge (passive Auswahl, als *vorgewählt* bezeichnet in der Legende), und bei der Nutzung des vollständigen Prozesses inklusive der interaktiven Einbeziehung des Benutzers (als *interaktiv* bezeichnet in der Legende).

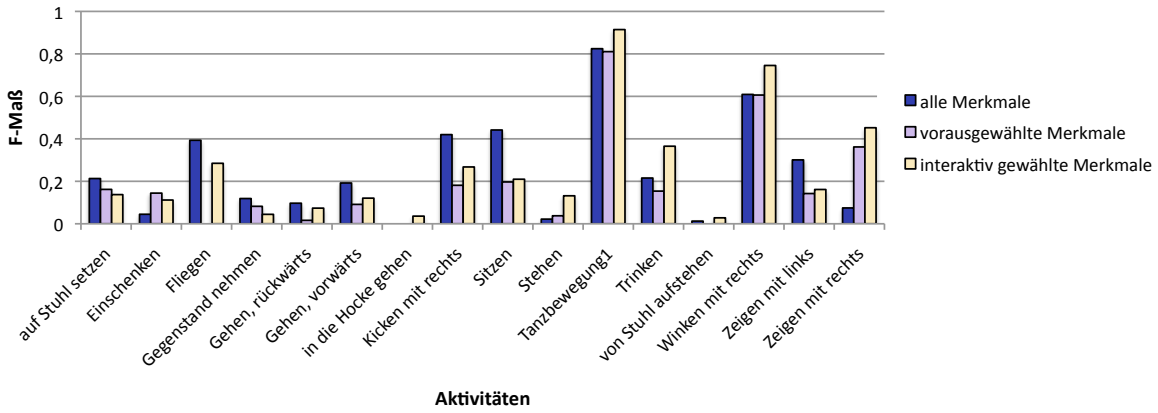


Abb. 7.8.: Vergleich der Erkenner für verschiedene Aktivitäten, trainiert auf unterschiedlichen Merkmalsmengen resultierend aus der Merkmalsauswahl (gewählt aus der vollständigen Merkmalsmenge, aus einer mittels Hintergrundwissen reduzierten Menge, und schließlich noch zusätzlich in Interaktion mit dem Benutzer verfeinert), anhand des F-Maßes.

denen Systemen in der Praxis nicht als Flaschenhals erwiesen hat. Schließlich gibt es noch einen nicht nicht direkt quantifizierbaren Vorteil durch die Verwendung von weniger Merkmalen: Für die Übertragbarkeit zwischen verschiedenen robotischen Systemen ist eine Bereitstellung gewisser Grundmerkmale notwendig, aus denen die in der Erkennung verwendeten Merkmale abstrahiert werden können. Es kann vorkommen, dass auf einem System nicht alle Messungen, die auf einem anderen System zur Verfügung stehen, ebenfalls gemacht werden können. In diesem Fall kann ein diese Messungen verwendender Erkenner natürlich nicht genutzt werden. Durch die Verringerung der Anzahl verwendeter Merkmale vergrößert sich im Gegenzug aber wieder die Wahrscheinlichkeit, dass zumindest diese (wenigen) Merkmale auch auf einer anderen Plattform gemessen werden können.

7.4. Erkennung von Aktivitäten

Das in diesem Abschnitt geschilderte Experiment dient der Evaluation der Verwendung des Aktivitätserkennungsystems mit trainierten Erkennern für verschiedene Aktivitäten. Das Experiment nutzt die im Rahmen des vorigen Experiments trainierten Erkenner, der genaue Ablauf der Evaluation ist in Abschnitt 7.4.1 beschrieben, die Resultate werden im darauf folgenden Abschnitt 7.4.2 präsentiert und interpretiert.

7.4.1. Durchführung der Evaluation

In diesem Experiment wurden die im vorigen Experiment trainierten Erkenner genutzt. Sie wurden auf zwei getrennte Datensätze D_1 und D_2 angewandt. D_1 und D_2 sind zusammengestellt aus Sequenzen, die mit einer einer Kombination aus Kinect-Sensor und OpenNiTE-Tracking aufgezeichnet wurden. D_1 enthält die im Training der Erkenner genutzten Daten, der andere Datensatz D_2 enthält in diesem Sinn „echte“ Evaluationsdaten, die nicht im Rahmen des Trainings verwendet wurden. Tabelle 7.6 listet die Größe der jeweiligen Daten auf.

Es wurden jeweils zwei Erkennungsläufe durchgeführt, einmal ohne und einmal mit Verwendung der Ergebnis-Nachbereitung. Um die auf dem Einsatz von Hintergrundwissen basierende Ergebnis-Nachbereitung verwenden zu können, muss die Erkennung der Aktivitäten gleichzeitig erfolgen (wie in einem echten Einsatzszenario).

Zur Bewertung der Erkennung wird wie in den anderen Experimenten die Qualität der Erkennung anhand der Maßzahlen Genauigkeit, Trefferquote und als Kombination davon F-Maß betrachtet.

Tab. 7.6.: Liste der für die Evaluation des Erkennungsprozesses genutzten Aktivitäten mit Details zur Größe der einzelnen Datensätze.

Bezeichnung	$\lvert D_1 \rvert$	$\lvert D_2 \rvert$
AUF STUHL SETZEN	139	47
EINSCHENKEN	131	109
FLIEGEN	84	179
GEHEN, RÜCKWÄRTS	138	74
GEHEN, VORWÄRTS	177	91
IN DIE HOCKE GEHEN	62	37
KICKEN MIT RECHTS	126	163
SITZEN	307	207
STEHEN	57	187
TANZBEWEGUNG1	102	215
TANZBEWEGUNG2	51	65
TRINKEN	131	133
VON STUHL AUFSTEHEN	54	42
WINKEN	107	153
ZEIGEN MIT LINKS	90	58
ZEIGEN MIT RECHTS	112	106

7.4.2. Evaluationsergebnisse

Die Tabellen7.7 und 7.8 zeigen die erreichten Ergebnisse der unterschiedlichen Testläufe. Eine einfacher erfassbare Darstellung ist das Diagramm in Abb. 7.9, das die resultierenden F-Maße der Erkennung der genutzten Aktivitäten darstellt.

Tab. 7.7.: Vergleich der erreichten Trefferquote und Genauigkeit der trainierten Erkenner auf den Daten von Evaluationsdatensatz D_1, der auch für der Trainieren der Erkenner genutzt wurde.

Aktivität	**ohne Ergebnisnachbereitung**		**mit Ergebnisnachbereitung**	
	Trefferquote	*Genauigkeit*	*Trefferquote*	*Genauigkeit*
AUF STUHL SETZEN	0,3309	0,6765	0,3525	0,6125
FLIEGEN	1,0	0,9231	1,0	0,9438
GEHEN, RÜCKWÄRTS	0,7899	0,586	0,7464	0,7687
GEHEN, VORWÄRTS	0,0056	0,2	0,0	0,0
IN DIE HOCKE GEHEN	0,3387	0,0968	0,1942	0,6136
KICKEN MIT RECHTS	0,2857	1,0	0,2937	1,0
SITZEN	0,0261	0,6667	0,0977	0,9091
STEHEN	0,7193	0,2303	0,6667	0,0,3363
TANZBEWEGUNG1	0,9608	1,0	0,9706	1,0
TANZBEWEGUNG2	1,0	0,8947	0,7843	0,9756
TRINKEN	0,4886	0,4886	0,0916	0,1967
VON STUHL AUFSTEHEN	0,3519	0,1439	0,5556	0,3061
WINKEN MIT RECHTS	0,9813	1,0	0,9813	1,0
ZEIGEN MIT LINKS	0,1111	0,3706	0,8111	0,5141
ZEIGEN MIT RECHTS	0,75	0,8	0,75	0,8155

An den Daten und im Diagramm ist deutlich zu erkennen, dass die Erkennung auf den unbekannten Evaluationsdaten schlechter abschneidet als auf den Trainingsdaten, wie es zu erwarten war. Es sind einige Ausreißer zu erkennen, bei denen die Erkennung insgesamt nur schlechte Ergebnisse liefert. Für die vier Aktivitäten AUF STUHL SETZEN, IN DIE HOCKE GEHEN, SITZEN und VON STUHL AUFSTEHEN sind Verwechslungen zwischen diesen Aktivitäten, die in einigen Aspekten (Merkmalen) starke Ähnlichkeiten aufweisen, eine wahrscheinliche Erklärung.

Die Ergebnis-Nachbereitung, die Hintergrundwissen über die Zusammenhänge zwischen verschiedenen Aktivitäten verwendet, liefert auf den Daten von Datensatz D_1 keine große Verbesserung, hält aber in den meisten Fällen die Qualität. Bei Daten von Datensatz D_2 dagegen sind teilweise deutliche Verbesserungen der Erkennung zu sehen, beispielsweise bei den Aktivitäten KICKEN MIT RECHTS und ZEIGEN MIT RECHTS.

Tab. 7.8.: Vergleich der erreichten Trefferquote und Genauigkeit der trainierten Erkenner auf den Daten von Evaluationsdatensatz D_2, der nicht im Training der Klassifikatoren verwendet wurde.

Aktivität	**ohne Ergebnisnachbereitung**		**mit Ergebnisnachbereitung**	
	Trefferquote	*Genauigkeit*	*Trefferquote*	*Genauigkeit*
AUF STUHL SETZEN	0,0	0,0	0,0	0,0
FLIEGEN	1,0	0,4398	0,5698	0,3411
GEHEN, RÜCKWÄRTS	0,0	0,0	0,0	0,0
GEHEN, VORWÄRTS	0,0	0,0	0,0	-
IN DIE HOCKE GEHEN	0,0	0,0	0,0	0,0
KICKEN MIT RECHTS	0,1104	1,0	0,1718	1,0
SITZEN	0,0	-	0,0	-
STEHEN	0,2727	0,1144	0,385	0,1674
TANZBEWEGUNG1	0,9349	1,0	0,8419	1,0
TANZBEWEGUNG2	0,5077	0,3173	0,3231	1,0
TRINKEN	0,4361	0,5577	0,0752	0,2703
VON STUHL AUFSTEHEN	0,0	0,0	0,0	0,0
WINKEN MIT RECHTS	1,0	0,9087	0,6986	0,9444
ZEIGEN MIT LINKS	0,7414	0,2067	0,7586	0,2558
ZEIGEN MIT RECHTS	0,5377	0,6129	0,6981	0,9024

Bei manchen Aktivitäten, beispielsweise TRINKEN, ist ein deutlicher Abfall der Erkennungsqualität zu erkennen, sowohl auf den Trainingsdaten in D_1 als auch auf den Evaluationsdaten in D_2. Auch hier ist Erklärung eine Unsicherheit des Systems, bei der neben dem für einen Datensatz korrekte Erkenner auch weitere Aktivitäten als sehr wahrscheinlich angesehen werden. Durch die Verwendung der Ergebnis-Nachbereitung erfolgt in dieser Situation eine Angleichung der erkannten Likelihoods. Das ist für die meisten Anwendungen auch das geeignete Verhalten, da in diesem Fall das Problem der Unsicherheit, welche der verschiedenen Aktivitäten die wahrscheinlichere ist, deutlich erkennbar sein muss. Damit wird einer der Vorteile von der Verwendung von Multiklassifikatoren, der durch das Training einzelner, aber kombinierbarer Klassifikatoren aufgegeben wurde, zurückgewonnen.

Für einen besseren Vergleich des Erkennungsverhaltens mit und ohne Verwendung der Ergebnis-Nachbereitung zeigen die beiden Abb. 7.10 und 7.11 die Plots der Erkennungswahrscheinlichkeit einer Teilmenge der Aktivitäten, einmal ohne Verwendung der Ergebnis-Nachbereitung (Abb. 7.10), einmal mit ihrer Verwendung (Abb. 7.11). Die Werte stammen aus einem Sequenz mit folgendem Ablauf: Zunächst ein Winken mit der rechten Hand, anschließend ein Zeigen nach rechts auf einen Gegenstand auf einem Tisch. Nach einer kurzen Pause nach links zeigen, und anschließend rückwärts aus dem Sichtbereich treten. Dabei wurde in der Erkennung die vollständige Menge trainierter Aktivitäten verwendet, nur das Diagramm blendet der Übersichtlichkeit halber einige Aktivitäten aus. An verschiedenen Stellen, beispielsweise um Frame 170 und Frame 250 herum, ist erkennbar wie die Likelihood einer Aktivität (hier ZEIGENRECHTS bzw. ZEIGENLINKS) abnimmt, wenn die einer anderen, nicht gleichzeitig möglichen Aktivität zunimmt.

7.5. Zusammenfassung und Bewertung der Ergebnisse

In diesem Kapitel wurden Experimente mit dem Gesamtsystem dargestellt, die die Leistungsfähigkeit und die Grenzen des entwickelten Ansatzes zeigen. Insbesondere zeigen die aufeinander aufbauenden Experimente zur Merkmalsfindung, dem Trainieren neuer Aktivitäten und der Verwendung des Systems zur Erkennung einer Menge von Aktivitäten das Zusammenspiel der drei Teilprozesse des Ansatzes.

Insgesamt liefern die Verbesserungen des Trackings die erwarteten Effekte zur Initialisierung und Stabilisierung der Personenbeobachtung. Die Erschließung neuer Anwendungsdomänen durch die automatische Suche nach potentiell relevanten Merkmalen liefert ohne Benutzereingriffe außer der Bereitstellung einer Referenzmenge von Trainingsdaten eine Merkmalsmenge, die sich auch mit handmodellierten Merkmalen messen kann, dabei aber kein Expertenwis-

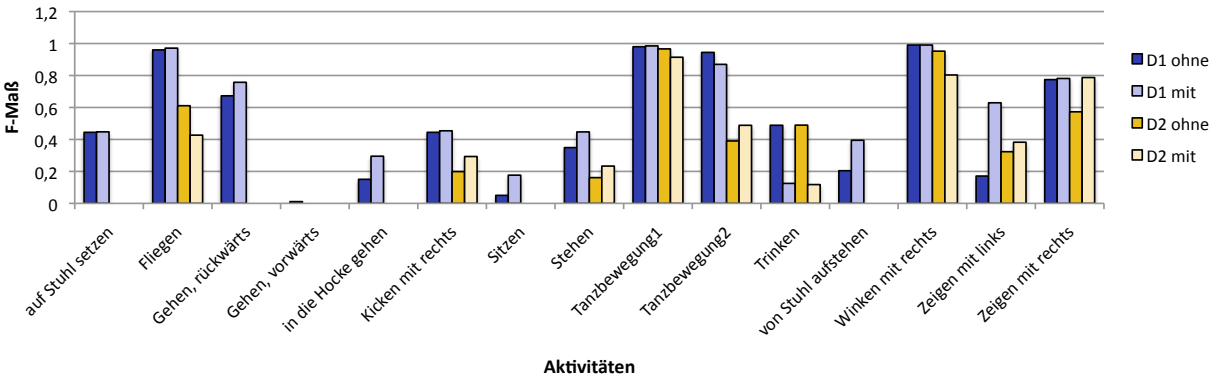


Abb. 7.9.: Vergleich der Erkennungsqualität auf den Datensätzen D_1 und D_2, jeweils mit und ohne Verwendung der Ergebnis-Nachbereitung (in der Legende bezeichnet mit *D1 ohne*, *D1 mit*, *D2 ohne* und *D2 mit*). Zur Bewertung der Qualität wird hier das aus Trefferquote und Genauigkeit kombinierte F-Maß verwendet.

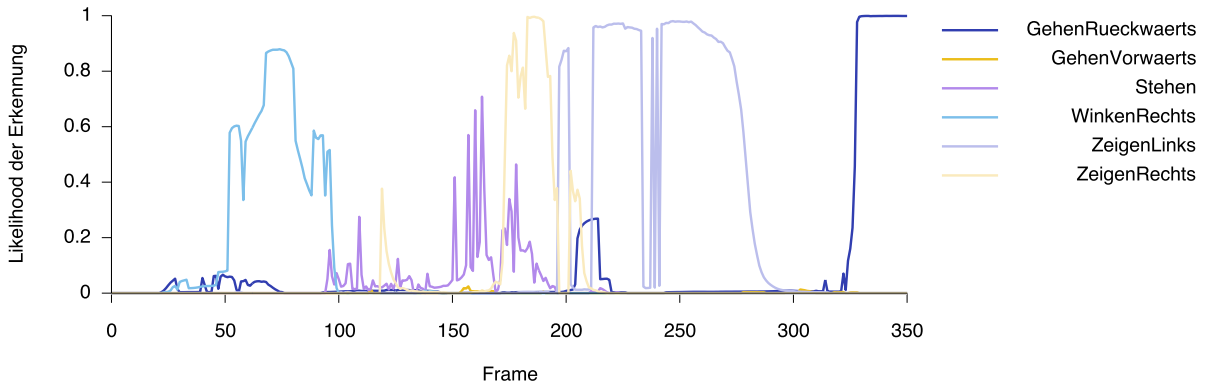


Abb. 7.10.: Darstellung der Erkennung einer Auswahl von Aktivitäten *ohne* Verwendung der Ergebnis-Nachbereitung.

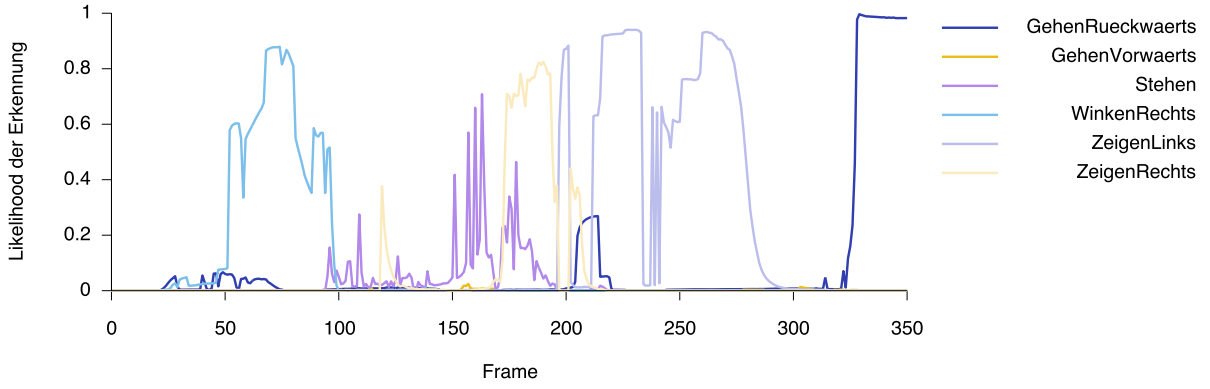


Abb. 7.11.: Darstellung der Erkennung einer Auswahl von Aktivitäten *mit* Verwendung der Ergebnis-Nachbereitung.

sen zur manuellen Modellierung der Merkmalsmenge benötigt. Beim Einlernen neuer Erkenner wurde die Verbesserung untersucht, die die gelernten Erkenner in der Verwendung zeigen, wenn in das Training Hintergrundwissen und Benutzerinteraktion integriert werden. Die Evaluation zeigt hier eine deutliche Abnahme der benötigten Merkmale, bei tendenziell besserer Erkennungsqualität. Das Experiment zur Verwendung des Systems zeigt schließlich, dass die Ergebnis-Nachbereitung einige Nachteile des Ansatzes, die Erkenner getrennt zu trainieren abmildern kann. Gleichzeitig ist hier zu sehen, dass die getrennt gelernten Erkenner einfach zu kombinieren und gemeinsam zu verwenden sind. Das bezeugt eine höhere Flexibilität für verschiedene Anwendungen, als wenn die Aktivitäten nur in einer festen, gemeinsamen Kombination erkannt werden könnten.

Die Experimente wurden alle mit jeweils nur einer gleichzeitig beobachteten Person durchgeführt. Die entwickelten Ansätze zur Verbesserung der Beobachtung sind auch für mehrere gleichzeitig beobachtete Personen verwendbar, Einschränkungen hierbei gibt es nur durch die verwendeten Sensoren beziehungsweise den verfügbaren Blickwinkel. In der Praxis ist dadurch der Einsatz für mehr als zwei Personen selten möglich. Die Klassifikation von Aktivitäten unterliegt einerseits den gleichen Einschränkungen wie die Beobachtung (da die Resultate der Beobachtung als Eingabe dienen). Darüberhinaus ist hier prinzipiell eine Verwendung für beliebig viele Personen gleichzeitig möglich. Dabei wird aber kein Vorteil daraus gezogen, dass die Aktivitäten mehrerer gleichzeitig beobachteter Personen miteinander zusammenhängen können. Um auch diesen Aspekt in Betracht zu ziehen, müssten entsprechende Merkmale definiert werden, die aus der Beobachtung mehrerer Personen extrahiert werden können. Diese mögliche Erweiterung wurde im Rahmen der vorliegenden Arbeit aber nicht untersucht.

8. Schlussbetrachtungen

In den vorangehenden Kapiteln wurde ein Ansatz zur Verfolgung und symbolischen Erkennung von menschlichen Bewegungen vorgestellt. In diesem Kapitel sollen zunächst die zentralen Beiträge zusammengefasst und anschließend bezüglich ihrer Leistungsfähigkeit und Grenzen diskutiert werden. Abschließend werden mögliche weiterführende Arbeiten abgeleitet.

8.1. Beitrag und Ergebnisse

Der Bereich der Service-Robotik orientiert sich zunehmend an Anwendungen in realen menschzentrierten Umgebungen, beispielsweise in der Küche oder Kellnerdienste in öffentlichen Situationen. In solchen Umgebungen stellt die Beobachtung und symbolische Erkennung von Bewegungen und Aktivitäten von Menschen eine Schlüsselfähigkeit dar. Die möglichen Anwendungen sind vielfältig und erstrecken sich von der Verwendung im Programmieren durch Vormachen über die Unterstützung der Interaktion von Mensch und Maschine bis hin zur wichtigen Informationsquelle für proaktive autonome Robotersteuerungen.

Ausgehend von dieser Bandbreite an Anwendungen wurden verschiedene Lösungsansätze entwickelt. Häufig betrachtete Aspekte sind dabei die erreichbare Erkennungsrate und Laufzeitanforderungen für bestimmte Anwendungen. In vielen Systemen ist dabei die Menge der erkannten Aktivitäten eingeschränkt und nicht erweiterbar, und häufig ist die Erkennung auch stark auf die verwendeten Datenquellen abgestimmt. Wie ein Aktivitätserkennungssystem aufgebaut sein muss, um ein einfaches Skalieren des Systems sowohl zwischen verschiedenen Sensorumgebungen als auch auf verschiedene Arten von Aktivitäten zu erlauben, bleibt dabei offen. Auch die Übertragbarkeit erlernter Erkenner für spezifische Aktivitäten, um die Erkennung einer Aktivität nicht auf jedem Roboter neu eintrainieren zu müssen, ist ein selten beachtetes Problem.

Diese Punkte standen im Fokus der vorliegenden Arbeit. Sie liefert dabei insbesondere die folgenden Beiträge zu diesen Forschungsfragen:

- Verbesserungen der Menschbeobachtung mit ICP-basierten Trackingverfahren bezüglich zweier Aspekte. Zum Einen wurde ein Verfahren für die Initialisierung des Trackings basierend auf Tiefenkameradaten entwickelt. Zum Anderen wurde ein Ansatz zur Ver-

wendung von Hintergrundwissen über den menschlichen Körper zur Verbesserung der Trackingresultate untersucht.

- Die Erschließung neuer Anwendungsdomänen erfordert die Definition geeigneter, auf die verfügbaren Sensoren und die geplante Anwendung passender Merkmale. Hierzu wurde eine automatische Exploration von Merkmalen vorgeschlagen, die ausgehend von einer Menge von Trainingsdaten selbständig potentiell relevante Merkmale in einer Domäne konstruiert und bezüglich ihrer potentiellen Relevanz analysiert.

- Für das Trainieren einer robusten Erkennung für eine Aktivität ist eine Auswahl der wirklich relevanten, diese Aktivität gut beschreibenden Merkmale notwendig. Hierzu wurde ein Ansatz vorgeschlagen, der eine aktive Auswahl von Merkmalen durch das System mit Hintergrundwissen und einer interaktiven Einbindung des Benutzers verbindet. Dadurch können die vor allem bei kleinen Trainingsdatenmengen auftretenden Probleme einer rein algorithmischen Auswahl teilweise ausgeglichen werden.

- Der Aufbau der eigentlichen Erkennung ist flexibel, und erlaubt die Verwendung unterschiedlicher Klassifikatoren in einer oder zwei Schichten, um für Aktivitäten unterschiedlicher Charakteristika und Komplexität die jeweils geeignetste Form zu bieten. Durch Einsatz einer Wissensbasis mit Informationen über Aktivitäten wird eine weitere Verbesserung der Erkennungsergebnisse erzielt.

Das Gesamtsystem wurde anhand einer Reihe von Aktivitäten evaluiert, die in typischen experimentellen Szenarien in der Mensch-Roboter-Interaktion und beim Programmieren durch Vormachen auftreten können. Dabei wurden die Leistungsfähigkeit und -grenzen des Systems aufgezeigt, und der Einfluss der einzelnen Teilkomponenten auf die Gesamtperformanz bewertet.

8.2. Diskussion

Die im Rahmen dieser Arbeit entwickelten Repräsentationen und Verfahren stellen in ihrer Gesamtheit ein integriertes Konzept dar, das die Erkennung von Aktivitäten auf verschiedenen Ebenen anhand unterschiedlicher Merkmale erlaubt. Eine Leitidee ist dabei die Verwendung von Hintergrundwissen als Hebel zum Ausgleich von kleinen Mengen von Trainingsdaten und anderen typischen Problemen. Aus der Darstellung der einzelnen Teillösungen, ihrer Integration in eine gemeinsame Architektur und der darauf durchgeführte Evaluation lassen sich die folgenden aufschlussreichen Punkte ableiten:

- Die Qualität des Trackings ist durch die entwickelten Verbesserungen auf einem Stand, der die Verwendung in experimentellen Robotik-Szenarien erlaubt. Für eine Verwendung in echten Alltagssituationen in deutlich engeren Haushaltsumgebungen ist die Robustheit allerdings noch nicht ausreichend. In jüngster Vergangenheit sind hier aber auch im kommerziellen Consumer-Bereich deutliche Fortschritte zu beobachten, die die Verfügbarkeit einer solchen Technologie in naher Zukunft wahrscheinlich erscheinen lassen.

- Die automatische Konstruktion und Suche von potentiell relevanten Merkmalen zeigt ihren Nutzen insbesondere bei der Erschließung bisher nicht behandelter Domänen. Für wohlverstandene Domänen ist auch die manuelle Definition von Merkmalen möglich, und kann insgesamt zu einer kleineren Menge von Merkmalen führen. Für unbekannte Domänen ist das aber nicht einfach möglich. Hier spielt der Ansatz klar seine Vorteile aus. Trotzdem bleibt festzuhalten, dass nicht vollständig beliebige Merkmale generiert werden können, sondern nur solche, die durch die vorgegebenen Operatoren berechnet werden können (was ein impliziter Bias des Systems ist). Daher kann auch nicht garantiert werden, dass keine relevanten Merkmale übersehen wurden.

- Der verbleibende Nachteil auch bei Verwendung der automatischen Merkmalsexploration ist die Notwendigkeit vorklassifizierter Trainingsdaten. Im Allgemeinen ist die Vorklassifikation von Daten ein aufwändiger und „teurer“ Prozess, der aber – im Gegensatz zur manuellen Definition von Merkmalen – nur die natürlich-menschliche Vertrautheit mit beispielhaften Aktivitäten der Domäne erfordert, kein Expertenwissen für die möglichen und günstigen Merkmale.

- Die Merkmalsauswahl ist durch die Verwendung von zusätzlichen Hintergrundwissen zur Vorauswahl und durch die interaktive Einbindung des Benutzers in die Feinauswahl deutlich verbessert worden, wie auch die Evaluation zeigt. Dafür stellt sich ein Grundierungs-Problem, denn es muss eine Zuordnung zwischen Wissensbasis/Benutzereingaben auf der einen und verwendeten Merkmalen auf der anderen Seite geschaffen werden. Die entwickelte Repräsentation von Merkmalstaxonomien stellt einen ersten Schritt in diese Richtung dar, aber eine vollständige autonome Grundierung (die allerdings auch nicht Ziel dieser Arbeit sein sollte) ist damit nicht möglich, hier muss auf aktive Modellierung zurückgegriffen werden.

- Die flexible Struktur der Erkennung ermöglicht die Anpassung auf Aktivitäten unterschiedlicher Komplexität, ein Ansatz, der sich in der praktischen Erwendung als äußerst robust erwiesen hat.Allerdings muss die Entscheidung, was die geeignetste Struktur des

Erkenners für eine bestimmte Aktivität ist, noch vom trainierenden Benutzer getroffen werden.

- Die konzeptuelle Entscheidung, die Erkennung auf abstrakte Merkmale, die von verschiedenen Systemen stammen können, zu stützen, zeigt die erwarteten Vorteile. Zusammen mit der Unabhängigkeit der Erkenner für unterschiedliche Aktivitäten, die andererseits über die wissensbasierte Ergebnis-Nachbereitung verknüpft werden, ist in der Tat eine Übertragbarkeit gelernter Erkenner zwischen verschiedenen Systemen möglich. Sie ist aber davon abhängig, dass die von den Erkennern benötigten Merkmale auch verfügbar sind. Neben der Problematik der Beobachtbarkeit können hier auch system-spezifische Transformationen der Messungen notwendig sein, die sowohl den Aufwand für die Integration der Aktivitätserkennung, als auch den Aufwand für die Verwendung (da eventuelle Transformationen der Messungen in jedem Zeitschritt durchgeführt werden müssen) stark erhöhen können.

- Die Kombinierbarkeit der Klassifikatoren einzelner Aktivitäten ist robust und verbessert die Wiederverwendbarkeit in verschiedenen Szenarien. Dabei wird davon ausgegangen, dass der Roboter in der Lage ist, anhand der aktuellen Situation eine Auswahl möglicher auftretender (und damit interessanter) Aktivitäten zu treffen. Die Entwicklung solcher Entscheidungsmechanismen wurde im Rahmen dieser Arbeit aber nicht behandelt.

Trotz der insgesamt vielversprechenden Ergebnisse des entwickelten Ansatzes wirft die Verwendung von Hintergrundwissen natürlich die Frage auf, woher es stammt beziehungsweise ob entsprechendes Hintergrundwissen autonom durch das System beschafft werden kann. Im Rahmen dieser Arbeit wurde dieser Aspekt nicht untersucht, sondern das benötigte Hintergrundwissen durch einen menschlichen Experten beigetragen. Einige Teile des genutzten Hintergrundwissens wiederum sind schon in der vorliegenden Form ausreichend, und müssen daher auch nicht weiter ausgebaut werden, beispielsweise das zur Verbesserung des Tracking genutzte Basiswissen über den menschlichen Körper. Andere Wissensbasen stellen in der vorliegenden Form einen Beweis für die Verwendbarkeit des Gesamtansatzes dar, müssten aber für eine weitergehende Verwendung in anderen Szenarien entsprechend der Anforderungen weiter gefüllt werden. Hier steckt ein Teil des Wissens, das nicht einfach autonom beschafft werden kann, in der Modellierung der Wissensstruktur. Die eigentlichen Inhalte, beispielsweise über die in einer gewissen Aktivität involvierten Körperteile, könnten beispielsweise auch durch Text Mining in einer allgemein zugänglichen Enzyklopädie wie etwa Wikipedia[1] gefunden werden.

[1] Homepage: `http://www.wikipedia.de`

8.3. Ausblick

Aus den in der Diskussion aufgeworfenen Punkten sind verschiedene Erweiterungen und Verbesserungen zur Fortführung der bisherigen Arbeit ableitbar, von denen die wichtigsten im Folgenden kurz ausgeführt werden.

Für die praktische Verwendung ist eine weitergehende Integration der gesamten Prozesskette notwendig, sodass sie vollständig auf einem als Zielsystem dienenden Roboter ausgeführt werden kann. Dazu bedarf es insbesondere einer Weiterentwicklung der Bedienung des Systems und der Anbindung einer geeigneten Spracherkennung, die leistungsfähig genug ist, die notwendigen Äußerungen des Benutzers umzusetzen. Gleichzeitig sind dazu an verschiedenen Stellen der Architektur Änderungen notwendig, die die Einbindung des Benutzers durch autonome Komponenten ersetzt, die beispielsweise durch weitere Lernverfahren oder datengetrieben die notwendigen Entscheidungen treffen können. Beispielhaft sei hier auf die Entscheidung einer konkreten Struktur des Erkenners abhängig von der Aktivität verwiesen.

Ein kritischer Punkt im ganzen Ablauf ist dabei die Notwendigkeit von vorklassifizierten Trainingsdaten. Während die eigentliche Aufzeichnung einerseits kaum zu vermeiden ist, andererseits aber schon mit den verfügbaren Systemen sehr einfach ist, liegt das eigentliche Problem in der Segmentierung und Vorklassifikation der Daten. Durch eine Ausführung auf einem robotischen System wird sich dieses Problem noch verschärfen. Zur Lösung dieses sich abzeichnenden Problems sind daher Forschungen im Bereich der automatischen Segmentierung und unüberwachter Lernverfahren zur Ballung der Daten notwendig, um die Vorklassifikation unter minimaler Benutzerinteraktion ausführen zu können.

Schließlich sollte die Akquisition des benötigten Hintergrundwissens automatisiert werden. Ein denkbarer Weg wäre es hier, Textmining in frei verfügbaren Datenbanken in Internet zu betreiben, um die benötigten Informationen über Aktivitäten zu sammeln, deren Bezeichnung wiederum vom menschlichen Trainer stammt.

Die Realisierung und Integration dieser Punkte würde es einem Robotersystem ermöglichen, autonom oder mit nur sehr wenig Benutzereingriffen die robuste Erkennung neuer menschlicher Aktivitäten zu erlernen. In dieser Arbeit wurde ein Grundstein für ein solches System realisiert, das einen natürlichen Umgang auch mit nicht-sprachlichen Aktivitäten von Menschen erlaubt.

A. Implementierte Merkmalsextraktoren

A.1. Initiale Merkmalsextraktoren (iMEMs) für Ganzkörperbeobachtung

Typ	Details
Position (P)	Absolute Position des Torsos
P	Absolute Position des Kopfes
P	Position des Kopfes relativ zu Torso
P	Absolute Position des rechten Oberarms
P	Position des rechten Oberarms relativ zu Torso
P	Absolute Position des linken Oberarms
P	Position des linken Oberarms relativ zu Torso
P	Absolute Position des rechten Unterarms
P	Position des rechten Unterarms relativ zu Torso
P	Position des rechten Unterarms relativ zu rechtem Oberarm
P	Absolute Position des linken Unterarms
P	Position des linken Unterarms relativ zu Torso
P	Position des linken Unterarms relativ zu linkem Oberarm
P	Absolute Position des rechten Oberschenkels
P	Position des rechten Oberschenkels zu Torso
P	Absolute Position des linken Oberschenkels
P	Position des linken Oberschenkels zu Torso
P	Absolute Position des rechten Unterschenkels
P	Position des rechten Unterschenkels zu Torso
P	Position des rechten Unterschenkels zu rechtem Oberschenkel
P	Absolute Position des linken Unterschenkels
P	Position des linken Unterschenkels zu Torso
P	Orientierung des linken Unterschenkels zu linkem Oberschenkel
Winkel (W)	Absolute Orientierung des Torso
W	Absolute Orientierung des Kopfes
W	Orientierung des Kopfes relativ zu Torso
W	Absolute Orientierung des rechten Oberarms
W	Orientierung des rechten Oberarms relativ zu Torso

Typ	Details
W	Absolute Orientierung des linken Oberarms
W	Orientierung des linken Oberarms relativ zu Torso
W	Absolute Orientierung des rechten Unterarms
W	Orientierung des rechten Unterarms relativ zu Torso
W	Orientierung des rechten Unterarms relativ zu rechtem Oberarm
W	Absolute Orientierung des linken Unterarms
W	Orientierung des linken Unterarms relativ zu Torso
W	Orientierung des linken Unterarms relativ zu linkem Oberarm
W	Absolute Orientierung des rechten Oberschenkels
W	Orientierung des rechten Oberschenkels zu Torso
W	Absolute Orientierung des linken Oberschenkels
W	Orientierung des linken Oberschenkels zu Torso
W	Absolute Orientierung des rechten Unterschenkels
W	Orientierung des rechten Unterschenkels zu Torso
W	Orientierung des rechten Unterschenkels zu rechtem Oberschenkel
W	Absolute Orientierung des linken Unterschenkels
W	Orientierung des linken Unterschenkels zu Torso
W	Orientierung des linken Unterschenkels zu linkem Oberschenkel

A.2. Komplexe Merkmalsextraktoren (kMEMs)

Bezeichnung	Details
Ableitung	Berechnung der Ableitung einer Zeitreihe von Merkmalen, angeñähert durch den Differenzenquotient.
Abstand	Berechnung des euklidischen Abstands zwischen zwei Vektoren (kompatiblen Typs), d.h. euklidische Norm des Differenzvektors.
Betrag	Berechnung der euklidischen Norm eines Merkmalsvektors.
Differenz	Berechnung der Differenz zweier Merkmalswerte (die vom gleichen Typ sein müssen).
Differenzwinkel	Berechnung des Winkels zwischen zwei Merkmalsvektoren (mit kompatiblem Typ), beispielsweise zweier Geschwindigkeitsvektoren.
Kovarianz	Berechnung der Kovarianz zweier Zeitreihen von Merkmalen (beliebigen Typs).
Mittelwert	Berechnung des arithmetischen Mittels einer Merkmals-Zeitreihe (beliebigen Typs).
Periodizität	Berechnung der Periodizität (Hauptfrequenz) einer Zeitreihe von Merkmalen.
Varianz	Berechnung der Varianz einer Merkmals-Zeitreihe (beliebigen Typs).

B. Zusätzliche Daten zur Evaluation

Dieser Anhang stellt zusätzliche Details zur Evaluation in Kapitel 7 zusammen, die über den dort benötigten Detailgrad hinausgehen.

B.1. Vergleichs-Merkmalsmenge

Die Evaluation der Erschließung neuer Anwendungsdomänen in Kapitel 7.2 verwendet als Vergleichsdaten die Resultate, die mit einer manuell definierten Merkmalsmenge erzielt werden können. Die dabei verwendeten Merkmale sind die in [Lösch et al., 2007] definierten 320 Merkmale, die im Detail aus den in Tabelle B.1 aufgelisteten Merkmalen bestehen.

Tab. B.1.: Liste der aus [Lösch et al., 2007] übernommenen manuell definierten Merkmale, mit denen die aus der automatischen Merkmalsexploration generierten Merkmale in Abschnitt 7.2 verglichen wurden.

Nummer	Merkmal
1 – 3	Absolute Orientierung des Torsos
4 – 6	Absolute Orientierung des Kopfes
7 – 9	Winkel zwischen Kopf und Torso
10 – 12	Absolute Orientierung des linken Oberarms
13 – 15	Winkel zwischen linkem Oberarm und Torso
16 – 18	Absolute Orientierung des rechten Oberarms
19 – 21	Winkel zwischen rechtem Oberarm und Torso
22 – 24	Absolute Orientierung des linken Oberschenkels
25 – 27	Winkel zwischen linkem Oberschenkel und Torso
28 – 30	Absolute Orientierung von rechtem Oberschenkel
31 – 33	Winkel zwischen rechtem Oberschenkel und Torso
34 – 36	Absolute Orientierung von linkem Unterarm
37 – 39	Winkel zwischen linkem Unterarm und Oberarm
40 – 42	Absolute Orientierung von rechtem Unterarm
43 – 45	Winkel zwischen rechtem Unterarm und Oberarm
46 – 48	Absolute Orientierung von linkem Unterschenkel

Nummer	Merkmal
49 – 51	Winkel zwischen linkem Unterschenkel und Oberschenkel
52 – 54	Absolute Orientierung von rechtem Unterschenkel
55 – 57	Winkel zwischen rechtem Unterschenkel und Oberschenkel
58 – 60	Absolute Position des Kopfes
61 – 63	Relative Position des Kopfes zum Torso
64 – 66	Absolute Position der linken Hand
67 – 69	Relative Position der linken Hand zum Torso
70 – 72	Relative Position der linken Hand zum linken Oberarm
73 – 75	Absolute Position der rechten Hand
76 – 78	Relative Position der rechten Hand zum Torso
79 – 81	Relative Position der rechten Hand zum rechten Oberarm
82 – 84	Absolute Position des linken Fusses
85 – 87	Relative Position von linkem Fuß zu Torso
88 – 90	Relative Position von linkem Fuß zu linkem Oberschenkel
91 – 93	Absolute Position des rechten Fusses
94 – 96	Relative Position von rechtem Fuß zu Torso
97 – 99	Relative Position von rechtem Fuß zu rechtem Oberschenkel
100 – 102	Absolute Winkelgeschwindigkeit des Torsos
103 – 105	Absolute Winkelgeschwindigkeit des Kopfes
106 – 108	Relative Winkelgeschwindigkeit von Kopf zu Torso
109 – 111	Absolute Winkelgeschwindigkeit des linken Oberarms
112 – 114	Relative Winkelgeschwindigkeit des linken Oberarms zum Torso
115 – 117	Absolute Winkelgeschwindigkeit des rechten Oberarms
118 – 120	Relative Winkelgeschwindigkeit des rechten Oberarms zum Torso
121 – 123	Absolute Winkelgeschwindigkeit des linken Oberschenkels
124 – 126	Relative Winkelgeschwindigkeit des linken Oberschenkels zum Torso
127 – 129	Absolute Winkelgeschwindigkeit des rechten Oberschenkels
130 – 132	Relative Winkelgeschwindigkeit des rechten Oberschenkels zum Torso
133 – 135	Absolute Winkelgeschwindigkeit des linken Unterarms
136 – 138	Relative Winkelgeschwindigkeit des linken Unterarms zum Torso
139 – 141	Relative Winkelgeschwindigkeit von linkem Unteram zu Oberarm
142 – 144	Absolute Winkelgeschwindigkeit des rechten Unterarms
145 – 147	Relative Winkelgeschwindigkeit des rechten Unterarms zum Torso
148 – 150	Relative Winkelgeschwindigkeit von rechtem Unteram zu Oberarm
151 – 153	Absolute Winkelgeschwindigkeit des linken Unterschenkels
154 – 156	Relative Winkelgeschwindigkeit von linkem Unterschenkel zu Torso

Nummer	Merkmal
157 – 159	Relative Winkelgeschwindigkeit von linkem Unterschenkel zu Oberschenkel
160 – 162	Absolute Winkelgeschwindigkeit des rechten Unterschenkels
163 – 165	Relative Winkelgeschwindigkeit von rechtem Unterschenkel zu Torso
166 – 168	Relative Winkelgeschwindigkeit von rechtem Unterschenkel zu Oberschenkel
169	Ungerichtete absolute Winkelgeschwindigkeit des Torsos
170	Ungerichtete absolute Winkelgeschwindigkeit des Kopfes
171	Ungerichtete Winkelgeschwindigkeit des Kopfes relativ zu Torso
172	Ungerichtete absolute Winkelgeschwindigkeit des linken Oberarms
173	Ungerichtete Winkelgeschwindigkeit des linken Oberarms relativ zu Torso
174	Ungerichtete absolute Winkelgeschwindigkeit des rechten Oberarms
175	Ungerichtete Winkelgeschwindigkeit des rechten Oberarms relativ zu Torso
176	Ungerichtete absolute Winkelgeschwindigkeit des linken Oberschenkels
177	Ungerichtete Winkelgeschwindigkeit des linken Oberschenkels relativ zu Torso
178	Ungerichtete absolute Winkelgeschwindigkeit des rechten Oberschenkels
179	Ungerichtete Winkelgeschwindigkeit des rechten Oberschenkels relativ zu Torso
180	Ungerichtete absolute Winkelgeschwindigkeit des linken Unterarms
181	Ungerichtete Winkelgeschwindigkeit des linken Unterarms relativ zu Torso
182	Ungerichtete Winkelgeschwindigkeit des linken Unterarms relativ zum Oberarm
183	Ungerichtete absolute Winkelgeschwindigkeit des rechten Unterarms
184	Ungerichtete Winkelgeschwindigkeit des rechten Oberarms relativ zu Torso
185	Ungerichtete Winkelgeschwindigkeit des rechten Unterarms relativ zum Oberarm
186	Ungerichtete absolute Winkelgeschwindigkeit des linken Unterschenkels
187	Ungerichtete Winkelgeschwindigkeit des linken Unterschenkels relativ zu Torso
188	Ungerichtete Winkelgeschwindigkeit des linken Unterschenkels relativ zum Oberschenkel
189	Ungerichtete absolute Winkelgeschwindigkeit des rechten Unterschenkels
190	Ungerichtete Winkelgeschwindigkeit des rechten Unterschenkels relativ zu Torso
191	Ungerichtete Winkelgeschwindigkeit des rechten Unterschenkels relativ zum Oberschenkel
192 – 194	Absolute Geschwindigkeit des Torsos
195 – 197	Absolute Geschwindigkeit des linken Ellbogens
198 – 200	Relative Geschwindigkeit des linken Ellbogens zu Torso
201 – 203	Absolute Geschwindigkeit des rechten Ellbogens
204 – 206	Relative Geschwindigkeit des rechten Ellbogens zu Torso
207 – 209	Absolute Geschwindigkeit des linken Knies
210 – 212	Relative Geschwindigkeit des linken Knies zu Torso

Nummer	Merkmal
213 – 215	Absolute Geschwindigkeit des rechten Knies
216 – 218	Relative Geschwindigkeit des rechten Knies zu Torso
219 – 221	Absolute Geschwindigkeit der linken Hand
222 – 224	Relative Geschwindigkeit der linken Hand zu Torso
225 – 227	Relative Geschwindigkeit der linken Hand zu Oberarm
228 – 230	Absolute Geschwindigkeit der rechten Hand
231 – 233	Relative Geschwindigkeit der rechten Hand zu Torso
234 – 236	Relative Geschwindigkeit der rechten Hand zu Oberarm
237 – 239	Absolute Geschwindigkeit des linken Fusses
240 – 242	Relative Geschwindigkeit des linken Fusses zu Torso
243 – 245	Relative Geschwindigkeit des linken Fusses zu Oberschenkel
246 – 248	Absolute Geschwindigkeit des rechten Fusses
249 – 251	Relative Geschwindigkeit des rechten Fusses zu Torso
252 – 254	Relative Geschwindigkeit des rechten Fusses zu Oberschenkel
255	Ungerichtete absolute Geschwindigkeit des Torsos
256	Ungerichtete absolute Geschwindigkeit des linken Ellbogens
257	Ungerichtete Geschwindigkeit des linken Ellbogens relativ zu Torso
258	Ungerichtete absolute Geschwindigkeit des rechten Ellbogens
259	Ungerichtete Geschwindigkeit des rechten Ellbogens relativ zu Torso
260	Ungerichtete absolute Geschwindigkeit des linken Knies
261	Undirected velocity of the left knee relative to torso
262	Ungerichtete absolute Geschwindigkeit des rechten Knies
263	Undirected velocity of the right knee relative to torso
264	Ungerichtete absolute Geschwindigkeit der linken Hand
265	Ungerichtete Geschwindigkeit der linken Hand relativ zu Torso
266	Ungerichtete Geschwindigkeit der linken Hand relativ zu Oberarm
267	Ungerichtete absolute Geschwindigkeit der rechten Hand
268	Ungerichtete Geschwindigkeit der rechten Hand relativ zu Torso
269	Ungerichtete Geschwindigkeit der rechten Hand relativ zu Oberarm
270	Ungerichtete absolute Geschwindigkeit des linken Fusses
271	Ungerichtete Geschwindigkeit des linken Fusses relativ zu Torso
272	Ungerichtete Geschwindigkeit des linken Fusses relativ zu Oberschenkel
273	Ungerichtete absolute Geschwindigkeit des rechten Fusses
274	Ungerichtete Geschwindigkeit des rechtes Fusses relativ zu Torso
275	Ungerichtete Geschwindigkeit des rechtes Fusses relativ zu Oberschenkel
276 – 278	Varianz der relativen Position von linker Hand zu Torso

Nummer	Merkmal
279 – 281	Varianz der relativen Position von rechter Hand zu Torso
282 – 284	Varianz der relativen Position von linkem Fuß zu Torso
285 – 287	Varianz der relativen Position von rechtem Fuß zu Torso
288 – 290	Kovarianz der Position von linker und rechter Hand
291 – 293	Kovarianz der Position von linkem und rechtem Fuß
294 – 296	Kovarianz der Position von linker und rechter Hüfte
297 – 299	Kovarianz der Position von linkem und rechtem Knie
300 – 302	Mittlere Position der linken Hand relativ zu Torso
303 – 305	Mittlere Position der rechten Hand relativ zu Torso
306 – 308	Mittlere Position des linken Fusses relativ zu Torso
309 – 311	Mittlere Position des rechten Fusses relativ zu Torso
312	Periodizität des Winkels zwischen Torso und linkem Oberarm
313	Periodizität des Winkels zwischen Torso und rechtem Oberarm
314	Periodizität des Winkels zwischen Torso und linkem Oberschenkel
315	Periodizität des Winkels zwischen Torso und rechtem Oberschenkel
316	Periodizität des Winkels zwischen linkem Unterarm und Oberarm
317	Periodizität des Winkels zwischen rechtem Unterarm und Oberarm
318	Periodizität des Winkels zwischen linkem Unterschenkel und Oberschenkel
319	Periodizität des Winkels zwischen rechtem Unterschenkel und Oberschenkel
320	Abstand zwischen beiden Händen

C. Evaluation einzelner Komponenten

C.1. Vergleich von Merkmalsauswahl-Algorithmen

Dieser Abschnitt beschreibt die Ergebnisse eines Vergleichs von 3 Merkmalsauswahlalgorithmen:

- *Correlation-based Feature Subset Selection*-Algorithmus (CbFSS) nach [Hall, 2000]
- *Fast Correlation-based Filter*-Algorithmus (im Folgenden abgekürzt als FCbF) nach [Yu and Liu, 2003] und [Yu and Liu, 2004]
- *Relief-F*-Algorithmus nach [Kira and Rendell, 1992] und [Kononenko, 1994]

Der hier wiedergegebene Vergleich wurde im Rahmen einer Studienarbeit vorgenommen [Hesse, 2010].

C.1.1. Testbedingungen

Die Tests wurden auf einem Standard-PC mit folgenden Komponenten durchgeführt: AMD Athlon 64 X2 Dual Core Prozessor 6000, 4 Gb RAM.

Zum Vergleich der Qualität der Ergebnisse der Merkmalsauswahl wurden drei unterschiedliche Klassifikatoren eingesetzt: SVMs (im Folgenden: SMO), Entscheidungsbäume (im Folgenden: C4.5), Neuronale Netze (im Folgenden: MLP).

Als Daten wurden Datensätze von zwei Personen mit je 11 verschiedenen Aktivitäten verwendet, die Datensätze bestehen jeweils aus etwa 20.000 Aufnahmen. Zur Evaluation wurde jeder Klassifikator mit den gewählten Merkmalen und den Daten je einer Person trainiert, und auf den Daten der anderen Person evaluiert. Die Ergebnisse beider Durchgänge wurden gemittelt.

C.1.2. Ergebnisse

Wiedergegeben werden hier Trefferquote (engl. recall) und Genauigkeit (engl. precision) der Erkennungsergebnisse mit den trainierten Klassifikatoren.

	FCbF		CFS		Relief-F	
	Trefferquote	Genauigkeit	Trefferquote	Genauigkeit	Trefferquote	Genauigkeit
SMO	0,37	0,57	0,40	0,47	0.36	0,52
C4.5	0,44	0,47	0,49	0,50	0,36	0,50
MLP	0,42	0,58	0,40	0,49	0,46	0,49
Ø	0,41	0,54	0,43	0,49	0,39	0,50

Bezüglich der Laufzeiten wurden ebenfalls große Unterschiede festgestellt. Während FCbF Resultate in 1-5 Minuten und CbFSS in 1-2 Minuten lieferten, lief Relief-F etwa 2-3 Stunden pro Datensatz.

C.2. Zusätzliche Evaluationsergebnisse der Merkmalsexploration

C.2.1. Zusammenhang zwischen Parametern und resultierender Merkmalsmenge

Zur Untersuchung des Zusammenhangs zwischen den wählbaren Parametern θ_{Me} und n_{mR} auf der einen und der Größe der resultierenden Merkmalsmenge auf der anderen Seite wurde die Generierung und Auswahl von potentiell relevanten Merkmalen mit den in Abschnitt 7.2.2 beschriebenen Daten mit verschiedenen Parameterkombinationen ausgeführt.

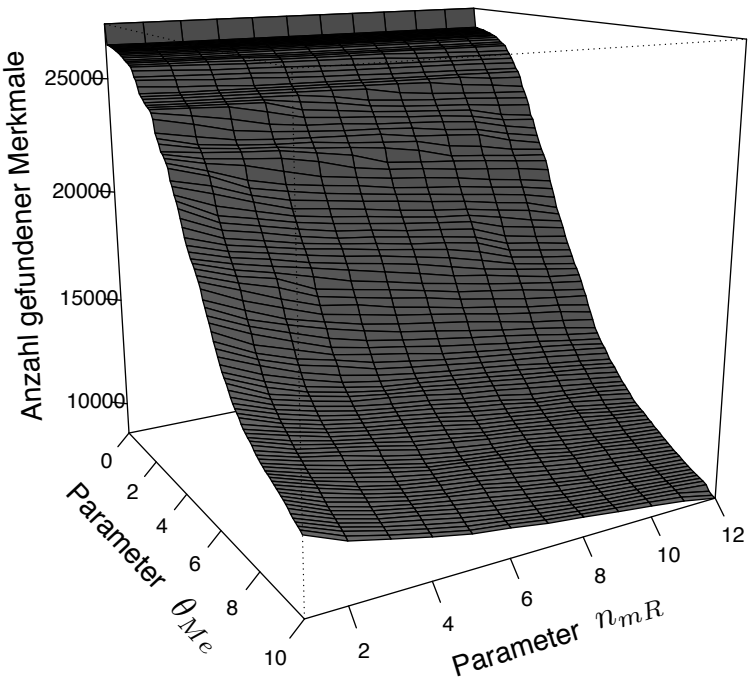


Abb. C.1.: Einfluss der Parameter θ_{Me} und n_{mR} auf die Anzahl der Merkmale, die nach dem Evaluationsschritt der Merkmalsexploration nicht verworfen wurden.

Abb. C.1 zeigt eine graphische Darstellung der erzielten Resultate. Deutlich erkennbar ist, dass der Einfluss des Schwellwerts θ_{Me} auf die Anzahl der Merkmale stärker ist als der Einfluss des Parameters n_{mR}. Allerdings flacht die Abnahme der Merkmale ab etwa $\theta_{Me} = 5$ ab, gleichzeitig wird der Einflus von n_{mR} deutlicher erkennbar.

D. Abbildungsverzeichnis

E. Tabellenverzeichnis

F. Literaturverzeichnis

Ankur Agarwal and Bill Triggs. Recovering 3D Human Pose from Monocular Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:44–58, 2006.

J.K. Aggarwal and Q. Cai. Human motion analysis: a review. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 90–102, 16 June 1997.

Ahmed Al-Ani and Mohamed Deriche. A New Technique for Combining Multiple Classifiers using The Dempster-Shafer Theory of Evidence. *Journal of Artificial Intelligence Research*, 17:333–361, Novembert 2002.

A. Ali and J.K. Aggarwal. Segmentation and recognition of continuous human activity. In *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on*, pages 28–35, 8 July 2001.

Colin M. Angle (CEO bei iRobot®). iRobot CEO Discusses Q4 2010 Results - Earnings Call Transcript. Zitiert unter `http://seekingalpha.com/article/252090-irobot-ceo-discusses-q4-2010-results-earnings-call-transcript?source=yahoo`, zuletzt geprüft am 15.11.2011.

Isaac Asimov. *Alle Roboter-Geschichten*. Number ISBN: 3404240820. Bastei-Lübbe, 7. auflage edition, September 2004.

Pedram Azad. *Visual Perception for Manipulation and Imitation in Humanoid Robots*. Number ISBN: 3642042287. Springer, Berlin, 1 edition, November 2009, URL: `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000011294`.

Pedram Azad, Aleš Ude, Rüdiger Dillmann, and Gordon Cheng. A full body human motion capture system using particle filtering and on-the-fly edge detection. In *International Conference on Humanoid Robots (Humanoids)*, Santa Monica, USA, 2004.

Pedram Azad, Tamim Asfour, and Rüdiger Dillmann. Toward an unified representation for imitation of human motion on humanoids. In *IEEE International Conference on Robotics and Automation (ICRA 2007), Proceedings*, Rome, Italy, 2007.

Franz Baader, editor. *The Description Logic Handbook*. Number ISBN: 0-521-78176-0. Cambridge University Press, Cambridge, UK, 2003.

Sumit Basu. A linked-hmm model for robust voicing and speech detection. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages 816–819. IEEE, 6-10 April 2003.

Regine Becher, Ingo Boesnach, Peter Steinhaus, and Rüdiger Dillmann. From subject to object and back - combining human motions and object properties to understand user actions. In *2nd International Workshop on Human-Centered Robotic Systems (HCRS 2006), Proceedings*, 2006.

N. Benoudjit, D. François, M. Meurens, and M. Verleysen. Spectrophotometric variable selection by mutual information. *Chemometrics and Intelligent Laboratory Systems*, 74(2): 243–251, December 28 2004. <ce:title>Chimiometrie 2003</ce:title>.

P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Computer Vision, 2005 (ICCV 2005) Tenth IEEE International Conference on*, volume 2, pages 1395–1402, 17–21 Oct. 2005.

Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

Matthew Brand. Coupled hidden markov models for modeling interacting processes. Learning and Common Sene Technical Report 405, MIT, 1996.

Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition 1997, Proceedings of the IEEE Computer Society Conference on*, pages 994–999, 17–19 June 1997.

O. Brdiczka, P. Reignier, and J.L. Crowley. Automatic development of an abstract context model for an intelligent environment. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 35–39, 8–12 March 2005.

O. Brdiczka, P. Reignier, J.L. Crowley, D. Vaufreydaz, and J. Maisonnasse. Deterministic and probabilistic implementation of context. In *Pervasive Computing and Communications Work-*

shops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on, page 5 pp., Lab. GRAVIR, INRIA Rhone-Alpes, Montbonnot, France, 13–17 March 2006a.

O. Brdiczka, P.C. Yuen, S. Zaidenberg, P. Reignier, and J.L. Crowley. Automatic acquisition of context models and its application to video surveillance. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 1175–1178, INRIA Rhône-Alpes, Franc, 20-24 Aug. 2006b.

Alexander Butsch. Ansatz zur Aktivitätserkennung auf HAL-Sequenzen mittels mehrschichtiger HMMs. Master's thesis, Institut für Anthropomatik, Fakultät für Informatik, Karlsruher Institut für Technologie, März 2009.

Sylvain Calinon and Aude Billard. Stochastic gesture production and recognition model for a humanoid robot. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2769–2774, 28 Sept.-2 Oct. 2004.

Sylvain Calinon and Aude Billard. Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm. In *Proceedings of the International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005.

C. Cedras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing (IVC)*, 13(2):129–155, March 1999.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan. Context is key. *Communications of the ACM*, 48(3):49–53, March 2005.

J. L. Crowley. Context aware observation of human activities. In *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 909–912, 26-29 Aug. 2002.

J. L. Crowley. Situated observation of human activity. In *Computer Vision for Interactive and Intelligent Environment, 2005*, pages 97–108, 17-18 Nov. 2005.

Shalom Darmanjian, Sung-Phil Kim, Michael C. Nechyba, Jose Principe, Johan Wessberg, and Miguel A.L. Nicolelis. Independently Coupled HMM Switching Classifier for a bimodel Brain-Machine Interface. In *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pages 379–384. IEEE, 2006.

James W. Davis and Aaron F. Bobick. The representation and recognition of human movement using temporal templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 928–934. Media Lab., MIT, Cambridge, MA, USA ;, 17–19 June 1997.

D. Demirdjian. Enforcing constraints for human body tracking. In *2003 Conference on Computer Vision and Pattern Recognition Workshop*, volume 9, pages 102–109, Madison, Wisconsin, USA, 2003.

D. Demirdjian, K. Tollmar, K. Koile, N. Checka, and T. Darrell. Activity maps for location-aware computing. In *Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 70–75, 2002.

Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, February 2005.

Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *Conference on Computer Vision and Pattern Recognition (CVPR '00)*, volume 2, page 2126, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5:4–7, 2001.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Number ISBN: 978-0-471-05669-0. Wiley-Interscience, 2nd edition edition, 2001.

T.V. Duong, H.H. Bui, D.Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 838–845, 20-25 June 2005.

Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *Ninth IEEE International Conference on Computer Vision (ICCV '03)*, volume 2, page 726, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

M. Ehrenmann, R. Zollner, O. Rogalla, S. Vacek, and R. Dillmann. Observation in programming by demonstration: Training and execution environment. In *Proceedings of Third IEEE International Conference on Humanoid Robots, October 2003, Karlsruhe*, Karlsruhe and Munich, Germany, 2003.

Tobias Feldmann, Ioannis Mihailidis, Sebastian Schulz, Dietrich Paulus, and Annika Wörner. Online Full Body Human Motion Tracking Based on Dense Volumetric 3D Reconstructions from Multi Camera Setups. In *KI 2010, 33rd Annual German Conference on Artificial Intelligence*, 2010.

Mathias Fontmarty, Frederic Lerasle, and Patrick Danes. Data Fusion within a modified Annealed Particle Filter dedicated to Human Motion Capture. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3391–3396, 2007.

Mathias Fontmarty, Frédéric Lerasle, and Patrick Danés. Towards real-time markerless human motion capture from ambiance cameras using an hybrid particle filter. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, number ISBN: 978-1-4244-1765-0, pages 709–712, San Diego, CA, 12-15 Oct. 2008.

D. François, F. Rossi, V. Wertz, and M. Verleysen. Resampling methods for parameter-free and robust feature selection with mutual information. *Neurocomputing*, 70(7-9):1276–1288, March 2007. Advances in Computational Intelligence and Learning - 14th European Symposium on Artificial Neural Networks 2006.

Lena Gorelick, Moshe Blank, E. Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2247–2253, December 2007.

Richard D. Green and Ling Guan. Continuous human activity recognition. In *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, volume 1, pages 706–711, 6-9 Dec. 2004a.

Richard D. Green and Ling Guan. Quantifying and Recognizing Human Movement Patterns From Monocular Video Images - Part I: A New Framework for Modeling Human Motion. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(2):179–190, feb. 2004b.

Gutemberg Guerra-Filho and Yiannis Aloimonos. Human activity language: Grounding concepts with a linguistic framework. In *Semantic Multimedia*, volume 4306/2006 of *Lecture Notes in Computer Science*, pages 86–100. Springer Berlin / Heidelberg, 2006a.

Gutemberg Guerra-Filho and Yiannis Aloimonos. A sensory-motor language for human activity understanding. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, number ISBN: 1-4244-0200-X, pages 69–75, 4-6 Dec. 2006b.

Isabelle Guyon and André Elisseeff. An Introduction to Variable and Feature Selection. *The Journal of Machine Learning Research - SPECIAL ISSUE*, 3(ISSN:1532-4435):1157–1182, March 2003.

guzugi (Wikipedia). Roomba robotic vacuum cleaner, URL: `http://en.wikipedia.org/wiki/File:Roomba3g.jpg`.

Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 359–366. Morgan Kaufmann, San Francisco, CA, 2000.

Nikolas Hesse. Entwicklung eines Verfahrens zur Selektion relevanter numerischer Merkmale für Klassifikationsprobleme. Studienarbeit, Institut für Anthropomatik, Fakultät für Informatik, Karlsruher Institut für Technologie (KIT), April 2010.

Nils Hofemann, Jannik Fritsch, and Gerhard Sagerer. Recognition of deictic gestures with context. In Carl Edward Rasmussen, Heinrich H. Bulthoff, Bernhard Scholkopf, and Martin A. Giese, editors, *Pattern Recognition: 26th DAGM Symposium*, volume 3175/2004 of *Lecture Notes in Computer Science*, pages 334–341, Tubingen, Germany, August 30 - September 1 2004. Springer Berlin / Heidelberg.

S. Hongeng, F. Brémond, and R. Nevatia. Representation and optimal recognition of human activities. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 818–825, 13-15 June 2000.

Berthold K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America. A, Optics and image science*, 4(4):629–642, 1987.

Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3):334–352, 2004.

Zsolt L. Husz, Andrew M. Wallace, and Patrick R. Green. Human activity recognition with action primitives. In *Advanced Video and Signal Based Surveillance, 2007 (AVSS 2007), IEEE Conference on*, pages 330–335, London, Sept. 2007.

Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 159–163, New York, NY, USA, 2005. ACM Press.

Antoni Jaume i Capo, Javier Varona, Manuel Gonzalez-Hidalgo, Ramon Mas, and Franciso J. Perales. Automatic human body modeling for vision-based motion capture. In *in Short Communication Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'06)*, number ISBN 80-86943-05-4, Plzen (Czech Republic), February 2006.

T. Inamura, Y. Nakamura, H. Ezaki, and I. Toshima. Imitation and primitive symbol acquisition of humanoids by the integrated mimesis loop. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 4208–4213, 21–26 May 2001.

I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra. Feature subset selection by bayesian network-based optimization. *Artificial Intelligence*, 123(1-2):157–184, 2000.

Rainer Jäkel, Sven R. Schmidt-Rohr, Zhixing Xue, Martin Lösch, and Rüdiger Dillmann. Learning of probabilistic grasping strategies using programming by demonstration. In *IEEE International Conference on Robotics and Automation (ICRA 2010)*, May 2010.

Xiaofei Ji and Honghai Liu. Advances in view-invariant human motion analysis: A review. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(1):13–24, January 2010.

George John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, volume 129, pages 121–129. Morgan Kaufmann, 1994.

Ioannis A. Kakadiaris and Dimitri Metaxas. Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, 30:191–218, 1998. 10.1023/A:1008071332753.

A. I. Kapandji, Raoul Tubiana, and Louis Honore. *The Physiology of the Joints*, volume 1: The Upper Limb. Churchill Livingstone, 6 edition, 2007.

I. A. Kapandji and Louis Honoré. *The Physiology of the Joints*, volume 3: The Spinal Column, Pelvic Girdle and Head. Churchill Livingstone, 6 edition, 2008.

I. A. Kapandji and Matthew J. Kandel. *The Physiology of the Joints*, volume 2: Lower Limb. Churchill Livingstone, 1988.

V. Kellokumpu, M. Pietikainen, and J. Heikkila. Human activity recognition using sequences of postures. In *oc. IAPR Conference on Machine Vision Applications (MVA 2005)*, pages 570–573, Tsukuba Science City, Japan, 2005.

Kenji Kira and Larry A. Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *Proceedings of the 10th National Conference on Artificial Intelligence*, AAAI'92, pages 129–134. AAAI Press, 1992.

S. Knoop, S. Vacek, and R. Dillmann. Modeling joint constraints for an articulated 3d human body model with artificial correspondences in icp. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 74–79, Dec. 5, 2005.

S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Walt Disney Resort, Orlando, Florida, May 15 2006a.

Steffen Knoop. *Interaktive Erstellung und Ausführung von Handlungswissen für einen Serviceroboter*. Number ISBN: 978-3-86644-189-7. Universitätsverlag Karlsruhe, 2007, URL: `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000007205`.

Steffen Knoop, Stefan Vacek, Klaus Steinbach, and Rüdiger Dillmann. Sensor fusion for model based 3d tracking. In *Proceedings of MFI 2006*, Heidelberg, Germany, 2006b.

Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann. Fusion of 2d and 3d sensor data for articulated body tracking. *Robotics and Autonomous Systems*, (57):321–329, 2009.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273 – 324, 1997.

Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In Francesco Bergadano and Luc De Raedt, editors, *Machine Learning: ECML-94*, volume 784 of *Lecture Notes in Computer Science*, pages 171–182. Springer Berlin / Heidelberg, 1994, URL: `http://dx.doi.org/10.1007/3-540-57868-4_57`.

Sotiris Kotsiantis and Dimitris Kanellopoulos. Discretization Techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58, 2006.

Peter Kovesi. Symmetry and asymmetry from local phase. In *Tenth Australian Joint Conference on Artificial Intelligence*, 1997.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138–1–066138–16, June 23 2004.

Dana Kulić and Yoshihiko Nakamura. Incremental learning of human behaviors using hierarchical hidden markov models. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4649—4655, 2010.

Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. Representability of human motions by factorial hidden markov models. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2388–2393, San Diego, USA, Oct. 29 – Nov. 2 2007.

Dana Kulić, Hirotaka Imagawa, and Yoshihiko Nakamura. Online acquisition and visualization of motion primitives for humanoid robots. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 1210–1215, Sept. 27-Oct. 2 2009a.

Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. Online segmentation and clustering from continuous observation of whole body motions. In *Robotics, IEEE Transactions on*, volume 25, pages 1158–1166, Oct. 2009b.

Lukasz Kurgan and Krzysztof Cios. Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm. In *Proceedings of International Conference on Machine Learning and Applications (ICMLA 2003)*, pages 30–36, 2003.

Lukasz A. Kurgan and Krzysztof J. Cios. CAIM Discretization Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, pages 145–153, 2004.

Hyeon-Kyu Lee and Jin H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 1999.

Huan Liu and Hiroshi [Hrsg.] Motoda, editors. *Computational Methods of Feature Selection*. Number ISBN 1-58488-878-4 in Chapman & Hall/CRC data mining and knowledge discovery series. Chapman & Hall/CRC, Boca Raton, Fla. [u.a.], 2008.

Martin Lösch, Sven Schmidt-Rohr, Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann. Feature set selection and optimal classifier for human activity recognition. In *Robot and Human Interactive Communication 2007 (ROMAN 2007), 16th IEEE International Symposium on*, Jeju Island, Korea, Aug 26-29 2007.

Martin Lösch, Sven R. Schmidt-Rohr, and Rüdiger Dillmann. Making feature selection for human motion recognition more interactive through the use of taxonomies. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication*, Munich, Germany, August 1–3 2008. Technische Universität München.

Martin Lösch, Stefan Gärtner, Steffen Knoop, Sven R. Schmidt-Rohr, and Rüdiger Dillmann. A human body model initialization approach made real-time capable through heuristic constraints. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6, Munich, Germany, 22-26 June 2009.

Martin Lösch, Dirk Mayer, Sven R. Schmidt-Rohr, Rainer Jäkel, and Rüdiger Dillmann. Modelling of joint angle limits for icp-based body tracking. In *The 15th Internation Conference on Advanced Robotics (ICAR 2011)*, Tallinn, Estonia, June 20–23 2011.

David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

A. Madabhushi and J.K. Aggarwal. A bayesian approach to human activity recognition. In *Visual Surveillance, 1999. Second IEEE Workshop on, (VS'99)*, pages 25–32, 26 June 1999.

A. Madabhushi and J.K. Aggarwal. Using head movement to recognize activity. In *Pattern Recognition (ICPR), 2000. Proceedings. 15th International Conference on*, volume 4, pages 698–701, 2000.

Jani Mäntyjärvi, Johan Himberg, and Tapio Seppänen. Recognizing human motion with multiple acceleration sensors. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 2, pages 747–752. IEEE, 2001.

Paulo Menezes, Frédéric Lerasle, Jorge Dias, and Raja Chatila. A Single Camera Motion Capture System dedicated to Gestures Imitation. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, number ISBN: 0-7803-9320-1, pages 430–435, Tsukuba, 5 Dec. 2005.

Paulo Menezes, Frédéric Lerasle, and Jorge Dias. Data Fusion for 3D Gestures Tracking using a Camera mounted on a Robot. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, number ISBN: 0-7695-2521-0, pages 464–467, Hong Kong, 2006.

Anurag Mittal, Liang Zhao, and Larry S. Davis. Human Body Pose Estimation Using Silhouette Shape Analysis. In *Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on*, pages 263–270. IEEE, 21–22 July 2003.

Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding (CVIU)*, 81(3):231–268, 2001.

Darnell J. Moore and Irfan A. Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *Eighteenth national conference on Artificial intelligence*, pages 770–776, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

Darnell J. Moore, Irfan A. Essa, and Monson H. Hayes. Object spaces: Context management for human activity recognition. GVU Technical Report GIT-GVU-98-26, Georgia Institute of Technology, 1998.

Darnell J. Moore, Irfan A. Essa, and Monson H. Hayes. Exploiting human actions and object context for recognition tasks. In *Seventh International Conference on Computer Vision (ICCV'99)*, volume 1, pages 80–86, 1999.

Quirin Mühlbauer, Kolja Kühnlenz, and Martin Buss. A model-based algorithm to estimate body poses using stereo vision. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication*, pages 285–290, Munich, Germany, August 1–3 2008. Technische Universität München.

C.L. Nehaniv. Classifying types of gesture and inferring intent. In The Society for the Study of Artificial Intelligence and Simulation of Behaviour, editors, *Proc. AISB'05 Symposium on Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, pages 74–81, 2005.

C.L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami. A Methodological Approach Relating the Classification of Gesture to Identification of Human Intent in the Context of Human-Robot Interaction. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 371–377, 13–15 Aug. 2005.

Kai Nickel. *Visuelle Benutzermodellierung mit Tracking und Zeigegestenerkennung für einen humanoiden Roboter*. PhD thesis, Universität Karlsruhe (TH), 2008, URL: `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000010452`.

Kai Nickel, Edgar Seemann, and Rainer Stiefelhagen. 3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 565–570. Interactive Syst. Labs, Universitt Karlsruhe, Germany;, 17–19 May 2004.

Mihoko Niitsuma, Kouhei Kawaji, Kazuki Yokoi, and Hideki Hashimoto. Extraction of human-object relations in intelligent space. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication*. Technische Universität München, August 1–3 2008.

F. Niu and M. Abdel-Mottaleb. View-invariant human activity recognition based on shape and motion features. In *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on*, number ISBN 0-7695-2217-3, pages 546–556, 13-15 Dec. 2004.

Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang. Human activity detection and recognition for video surveillance. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, volume 1, pages 719–722, 27-30 June 2004.

Oliver Nuria, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8. Adaptive Syst. & Interaction, Microsoft Res., Redmond, WA, USA, 14–16 Oct 2002.

N. Otero, S. Knoop, Chrystopher L. Nehaniv, Dag Syrdal, Kerstin Dautenhahn, and R. Dillmann. Distribution and recognition of gestures in human-robot interaction. In *Robot and Human Interactive Communication, 15th IEEE International Symposium on*, Hatfield, UK, 2006. IEEE Institute of Electrical and Electronics Engineers.

Vasu Parameswaran and Rama Chellappa. View Independent Human Body Pose Estimation from a Single Perspective Image. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:16–22, 2004.

Michael Pardowitz. *Inkrementelles und interaktives Lernen von Handlungswissen für Haushaltsroboter*. PhD thesis, Universität Karlsruhe (TH), 2007, URL: `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000007098`.

Sangho Park and J.K. Aggarwal. Segmentation and Tracking of Interacting Human Body Parts under Occlusion and Shadowing. In *Workshop onMotion and Video Computing (MOTION 2002), Proceedings of*, pages 105–111. IEEE, 2002.

V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):677–695, July 1997.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, August 2005.

Selwyn Piramuthu and Riyaz T. Sikora. Iterative feature construction for improving inductive learning algorithms. *Expert Systems with Applications: An International Journal*, 36(2): 3401–3406, March 2009.

Otniel Portillo-Rodriguez, Oscar O. Sandoval-Gonzalez, Carlo A. Avizzano, Emanuele Ruffaldi, Davide Vercelli, and Massimo Bergamasco. Development of a 3d real time gesture recognition methodology for virtual environment control. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication*, pages 279–284, Munich, Germany, August 1–3 2008. Technische Universität München.

Lawrence R. Rabiner. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

F. Rossi, A. Lendasse, D. François, V. Wertz, and M. Verleysen. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics and Intelligent Laboratory Systems*, 80(2):215–226, February 2006.

Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, Alina Soos, and Michael Beetz. Towards 3d object maps for autonomous household robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3191–3198, Oct. 29 – Nov. 2 2007.

Radu Bogdan Rusu, Jan Bandouch, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Action recognition in intelligent environments using point cloud features extracted from silhouette sequences. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication*, pages 267–272, Munich, Germany, August 1–3 2008. Technische Universität München.

Lawrence K. Saul and Michael I. Jordan. Boltzmann Chains and Hidden Markov Models. In *Advances in Neural Information Processing Systems 7*, pages 435–442. Morgan Kaufmann Publishers Inc., 1995.

Sven R. Schmidt-Rohr, Steffen Knoop, Martin Lösch, and Rüdiger Dillmann. Reasoning for a multi-modal service robot considering uncertainty in human-robot interaction. In *Proceedings of the 3rd International Conference on Human-Robot Interaction (HRI 2008)*, 2008.

Sven R. Schmidt-Rohr, Martin Lösch, and Rüdiger Dillmann. Learning flexible, multi-modal human-robot interaction by observing human-human-interaction. In *In Proceedings of the 19th IEEE International Symposium in Robot and Human Interactive Communication*, 2010.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Number ISBN: 0262194759. MIT Press, 2001.

Michael Schünke, Erik Schulte, and Udo Schumacher. *PROMETHEUS Lernatlas der Anatomie: Allgemeine Anatomie und Bewegungssystem*. Georg Thieme Verlag, 2007.

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(10):379–423, 623–656, Juli & Oktober 1948.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Number ISBN: 0521813972. Cambridge University Press, 2004.

Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, USA, June 20–25 2011.

Rainer Stiefelhagen, C. Fügen, R. Gieselmann, H. Holzapfel, Kai Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2422–2427. Interactive Syst. Labs, Karlsruhe Univ., Germany;, 28 Sept. – 2 Oct. 2004.

Michel Verleysen, Fabrice Rossi, and Damien François. Advances in Feature Selection with Mutual Information. In Michael Biehl, Barbara Hammer, Michel Verleysen, and Thomas Villmann, editors, *Similarity-Based Clustering*, volume 5400 of *Lecture Notes in Computer Science*, pages 52–69. Springer Berlin / Heidelberg, 2009, URL: `http://dx.doi.org/10.1007/978-3-642-01805-3_4`.

Paul Viola and Michael Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2001.

Timothy Vollmer. PR2 robot from Willow Garage, URL: http://www.flickr.com/photos/sixteenmilesofstring/5749414416/in/photostream/.

Marek Vondrak, Leonid Sigal, and Odest Chadwicke Jenkins. Physical simulation for probabilistic motion tracking. In *Computer Vision and Pattern Recognition (CVPR 2008)*, June 2008.

L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, March 2003.

Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for svms. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, 2001.

Christian Wojek, Kai Nickel, and Rainer Stiefelhagen. Activity recognition and room-level tracking in an office environment. In *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pages 25–30, September 2006.

Ying Wu, Thomas S. Huang, and N. Mathews. Vision-based gesture recognition: A review. *Lecture Notes in Computer Science*, 1739:103–115, 1999.

Alper Yilmaz and Mubarak Shah. Actions sketch: a novel action representation. *Computer Vision and Pattern Recognition, 2005 (CVPR 2005). IEEE Computer Society Conference on*, 1:984–989, 20–25 June 2005.

Lei Yu and Huan Liu. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In Tom Fawcett and Nina Mishra, editors, *20th International Conference on Machine Learning (ICML 2003), Proceedings of the*, volume 20, page 856, Washington, DC, 2003. AAAI Press.

Lei Yu and Huan Liu. Efficient Feature Selection via Analysis of Relevance and Redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, December 2004.