**Technische Universität Ilmenau**
Fakultät für Informatik und Automatisierung
Fachgebiet Neuroinformatik und Kognitive Robotik

# Nutzung von Tiefendaten zur Verbesserung der Wahrnehmung und Wiedererkennung von Personen für einen mobilen Trainingsroboter

Masterarbeit zur Erlangung des akademischen Grades Master of Science

**Alexander Vorndran**

Betreuer: M.Sc. Thanh Quang Trinh,
Dipl.-Inf. Markus Eisenbach,
FG Neuroinformatik und Kognitive Robotik

Verantwortlicher Hochschullehrer:
Prof. Dr. H.-M. Groß,
FG Neuroinformatik und Kognitive Robotik

Die Masterarbeit wurde am 07.06.2017 bei der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau eingereicht.

Erklärung: „Hiermit versichere ich, dass ich diese Masterarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet.“

Ilmenau, 07.06.2017 .....

Alexander Vorndran

# Inhaltsverzeichnis

**1 Einleitung** **1**
1.1 Motivation . . . . . 1
1.2 Anwendungsszenario . . . . . 2
1.3 Grundlegender Ansatz . . . . . 4
1.4 Überblick . . . . . 6

**2 Grundlagen** **9**
2.1 Roboterplattform „Ringo“ . . . . . 9
2.2 Kinect2-Tiefenkamera . . . . . 12

**3 Stabile Erfassung von Personen** **17**
3.1 Montage und Hardware . . . . . 18
3.2 Theoretische Grundlagen . . . . . 19
3.3 State-of-the-Art . . . . . 25
3.4 Verfahren . . . . . 31
3.5 Experimente . . . . . 45
3.6 Zusammenfassung und Ausblick . . . . . 52

**4 Gangbasierte Personenwiedererkennung** **55**
4.1 Theoretische Grundlagen . . . . . 56
4.2 State-of-the-Art . . . . . 59
4.3 Verfahren . . . . . 71
4.4 Proof-of-Concept . . . . . 77
4.5 Zusammenfassung und Ausblick . . . . . 83

**5 Zusammenfassung und Ausblick** **85**

**A Ergänzungen** **89**

**B Visualisierungen** **101**

**Literaturverzeichnis** **109**

# Kapitel 1

# Einleitung

## 1.1 Motivation

Durch die steigende durchschnittliche Lebenserwartung nimmt auch der Anteil von altersbedingten Erkrankungen in der Bevölkerung zu. Eines der dabei am häufigsten auftretenden Probleme ist Gelenkverschleiß in Hüfte und Knie. Bei fortgeschrittenem Verlauf lassen sich die Beschwerden derartiger Athrosen nur durch den Einsatz von künstlichen Gelenken - sogenannten Endoprothesen - lindern. Durch Fortschritte der medizinischen Forschung im Bereich Endoprothetik und gut ausgebildetes Fachpersonal, verlaufen derartige Eingriffe und die anschließende Regeneration in den meisten Fällen ohne schwerwiegende Komplikationen. Damit die Lebensqualität im Anschluss an den Eingriff auch langanhaltend verbessert wird, ist eine physiotherapeutische Begleitung des Patienten notwendig. Zusätzlich zur angeleiteten Physiotherapie hat spielt dabei das Eigentraining bereits wenige Tage nach der Operation eine wichtige Rolle. Der Patient führt dabei angepasst an seinen aktuellen Therapiezustand möglichst selbständig Übungen durch, um den Genesungsverlauf zu beschleunigen.

Neben Kräftigungs- und Beweglichkeitsübungen, fällt ein Großteil des Eigentrainings unter den Begriff der Gangschule. Im Zuge einer Gangschule erlernt der Patient eine gesunde Laufbewegung. Dabei sollte stets auf eine korrekte Ausführung der Übungen geachtet werden, um das erneute Auftreten gelenkschädigender Bewegungsmuster frühzeitig erkennen und gegebenenfalls korrigieren zu können. Diese Gangschule

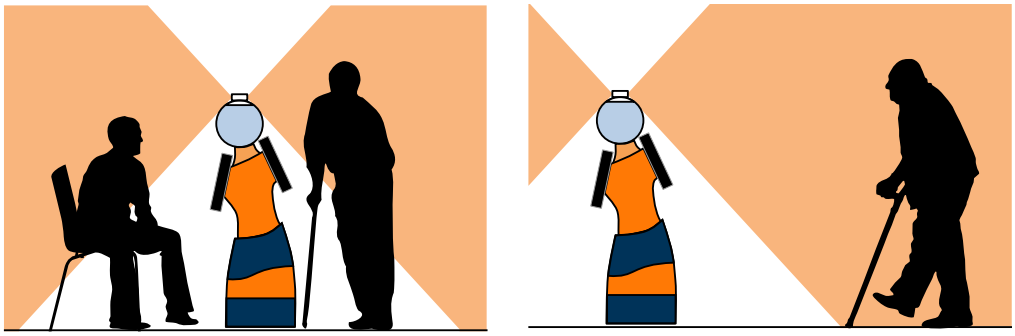
wird nach der Operation und in der Frühreha zumeist initial von einem Physiotherapeuten begleitet. Allerdings wird dadurch die Verfügbarkeit von Therapiekapazitäten beispielsweise im Bereich der unmittelbar post-operativen Mobilisierung am Patientenbett eingeschränkt. Daher steht ab einem gewissen Punkt der Maßnahmen meist wenig bis keine Rückmeldung für den Patient zur Verfügung. Technologisch assistiertes Eigentraining bietet daher ein hohes direktes und indirektes therapeutisches sowie ökonomisches Potential.

## 1.2 Anwendungsszenario

Durch den dynamischen Charakter der Gangschule eignen sich mobile Systeme, wie beispielsweise Assistenzroboter, besonders für den beschriebenen Anwendungsfall. Durch die hohe Flexibilität der mobilen Plattform entfällt die Notwendigkeit bestimmte Bereiche baulich um technische Vorrichtungen zu ergänzen, die bei seiner alltäglichen Nutzung gegebenenfalls als störend empfunden werden. Um eine derartige Anwendung realisieren zu können, wird ein Robotersystem benötigt, welches neben einem robusten, sozial-akzeptablem Mensch-Maschine-Interaktionsverhalten auch über geeignete Sensorik zur Erfassung der Patientenbewegung verfügt und mit Modelldaten über gesunde Gangmuster ausgestattet ist.
Im Rahmen des Projekts ROGER[1] ist es daher das Ziel der Partner aus dem Waldkrankenhaus „Rudolf Elle“ Eisenberg, der MetraLabs GmbH Ilmenau sowie dem Fachgebiet Neuroinformatik und Kognitive Robotik der Technischen Universität Ilmenau eine derartige Roboterplattform zu spezifizieren und zu realisieren. Ziel ist dabei ein vollständig autonomes System, das Patienten im Klinikalltag während des Gangtrainings begleiten und unmittelbar Rückmeldung über mögliche Abweichungen von der patientenspezifischen Soll-Bewegung, beziehungsweise Soll-Haltung, geben kann. Um dieses Ziel zu erreichen sind für die Dauer des Projektzeitraums zwei Ausbaustufen geplant.

[1] **ROGER**: **Ro**boterassistiertes **G**angtraining in der orthopädischen **R**ehabilitation
(Projektzeitraum: 01.11.2016 - 31.10.2019)
`http://www.tu-ilmenau.de/neurob/projects/roger/`
`http://www.roger-projekt.de/`

Die erste Stufe sieht vor, dass der Roboter den Patienten während seines Gangtrainings durchgängig begleitet und Sensorinformationen über seinen Verlauf aufzeichnet. Dieses Material soll anschließend aufbereitet und in einer Art und Weise dargestellt werden, die es einem eingewiesenen Physiotherapeuten erlaubt dem Patienten im Anschluss an das Training datengestütztes Feedback zu geben. Darauf aufbauend soll eine weitere Ausbaustufe umgesetzt werden, in welcher das Robotersystem vollständig autonom und möglichst unmittelbar, beispielsweise visuell und akustisch, Rückmeldung über die aktuelle Bewegung an den Patienten gibt. Dazu werden von Fachärzten und Therapeuten des Waldkrankenhaus „Rudolf Elle“ Eisenberg Normmodelle erarbeitet, die einen Vergleich mit den im Training erfassten Laufparametern ermöglichen.



(a) Anmeldung am Roboter (b) Training mit dem Roboter

**Abbildung 1.1:** Anmeldung und Training mit dem Roboter
*(a) Ein Patient meldet sich im Stehen oder Sitzen am Roboter an. (b) Anschließend fährt er in einigem Abstand vor ihm. Im Idealfall ist dieser stets groß genug, so dass die Person vollständig im Kamerabild erfasst werden kann.*

Der grundlegende Ablauf ist für beide Ausbaustufen gleich und soll in den Klinikalltag integriert werden. Der Patient meldet sich vor dem Beginn des Trainings im Wartebereich am Roboter an und wählt die gewünschte Gesamtdauer des Eigentrainings aus (ähnlich Abbildung 1.1(a)). Anschließend begeben sich Roboter und Patient gemeinsam in Richtung der Übungsstrecke. Dabei handelt es sich in der Regel um einen Abschnitt eines Stationsflures. Am Anfang wird eine kurze Laufstrecke von etwa 5 m markiert auf welcher der Patient möglichst ungestört laufen können sollte. Diese dient

zur Ermittlung des initialen, ausgeruhten Gangbildes, in der zweiten Ausbaustufe damit auch der Initialisierung des patientenspezifischen Modells. Für den weiteren Verlauf steht dann eine Strecke von etwa 50 m zur Verfügung auf der im Abstand von 10 m Stühle aufgestellt sind, die es dem Patienten erlauben sich kurz auszuruhen. Wie oft die Laufstrecke insgesamt abgegangen wird richtet sich nach der vom Patient gewählten Dauer und seinem Befinden, da jederzeit die Möglichkeit besteht das Training zu pausieren oder abzubrechen sollten sich Beschwerden einstellen. Der Roboter hat dabei während des gesamten aktiven Trainings die Aufgabe vor dem Patienten zu fahren und eine Frontalansicht des Trainingsverlaufs wie in Abbildung 1.1(b) mit der verbauten Sensorik zu dokumentieren.

Die erfassten Sensordaten dienen dazu dem Patienten im Anschluss (1. Stufe) oder unmittelbar während des Trainings (2. Stufe) Rückmeldung über seine Laufbewegung zu geben. Mögliche Auswertungsparameter sind dabei etwa die Schrittweite, Oberkörper- beziehungsweise Beckenneigung oder auch die gelaufene Strecke. Das medizinische Personal kann zusätzlich Absprache mit dem Patienten individuelle Trainingsschwerpunkte definieren.

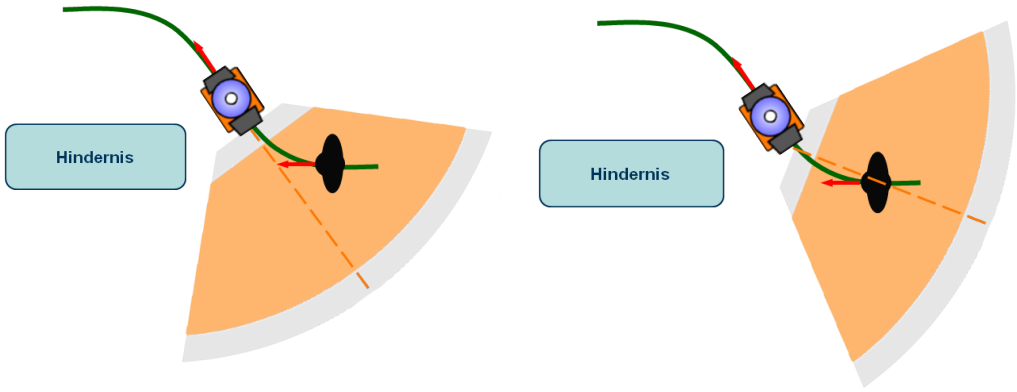
## 1.3 Grundlegender Ansatz

Der Patient soll im spezifizierten Szenario durchgehend durch den Assistenzroboter begleitet werden. In der aktiven Phase des Gangtrainings wird der Patient dabei frontal durch eine auf dem Roboter schwenkbar montierte Kinect-2-Tiefenkamera beobachtet. Um dieses Verhalten zu realisieren nutzt der Roboter eine Vielzahl von Basisfähigkeiten aus den Bereichen Pfadplanung, personenfokusierte Navigation, Situationserkennung [Trinh et al., 2015] sowie zur Erkennung, Verfolgung und Wiedererkennung von Personen [Wengefeld et al., 2016]. Aufbauend auf bereits erprobten Basisfähigkeiten aus dem ROREAS-Project [Scheidig et al., 2015, Gross et al., 2016] wurde zu Beginn des ROGER-Projekts eine erweiterte „Lotsen“-Funktion realisiert, welche die grundlegende Aufgabe übernimmt den Roboter in einem festgelegten Abstandsbereich vor dem Nutzer fahren zu lassen.

**Abbildung 1.2:** Beispiel für die Einsatzumgebung
*Das Bild zeigt die typische geplante Einsatzumgebung (Stationsflur) mit Stationswagen und zwei Patienten beim Eigentraining im Waldkrankenhaus „Rudolf Elle“ Eisenberg.*

Da es sich um eine nicht-holonome Roboterplattform handelt, führen die dabei gefahrenen Trajektorien dazu, dass der Nutzer beispielsweise bei Ausweichmanövern, die durch Personen oder statische Hindernisse (vergleiche dazu auch Abbildung 1.2) ausgelöst werden, aus dem Erfassungsbereich einer starr montierten rückwärtigen Kamera geraten kann (siehe dazu Abbildung 1.3(a)). Durch eine geeignete Ansteuerung der schwenkbaren Kamerahalterung, wie sie etwa in Abbildung 1.3(b) dargestellt ist, lassen sich derartige Situationen entschärfen und ein ruhigeres Kamerabild erzeugen.

Um diese Ansteuerung zu ermöglichen ist ein robustes Erkennen und Verfolgen des Nutzers notwendig. Die eingesetzte Roboterplattform verfügt dazu über verschiedene Sensoren, deren Informationen kombiniert werden um die geforderte robuste Interaktion zu gewährleisten. Zusätzlich zum Personentracking, das vor allem spatio-temporale Informationen nutzt um die Position einer Personen zu verfolgen, wird eine ansichtsbasierte Personenwiedererkennung eingesetzt um Trackingkonflikte, wie sie beispielsweise bei kreuzenden Personen auftreten, aufzulösen [Wengefeld et al., 2016]. Die dabei




(a) starrer Kameraaufbau (b) schwenkbarer Kameraaufbau

**Abbildung 1.3:** Sensor-Sichtfeld beim Ausweichen
*Ausweichen auf nicht-holonomen Trajektorien kann dazu führen, dass der Patient ungünstig im Sensorerfassungsbereich positioniert ist. Eine bessere Positionierung im Sensorbereich lässt sich bis zu einem gewissen Grad unabhängig von der Navigation durch eine schwenkbare Kamera realisieren.*

gewonnenen Informationen dienen als Grundlage für die Ansteuerung der schwenkbaren Tiefenkamera um den Nutzer möglichst optimal im Sensorerfassungsbereich platzieren zu können. Die durch die schwenkbare Tiefenkamera erfassten Sensordaten können zusätzlich genutzt werden um Personentracking und -wiedererkennung zu verbessern. Im betrachteten Anwendungsfall bieten sich vor allem markante Gangcharakteristiken als zusätzliches Merkmal an.

## 1.4 Überblick

Im folgenden Kapitel 2 wird zunächst die verwendete Roboterplattform und deren Sensorik detaillierter vorgestellt. Zusätzlich erfolgt ein Überblick über die technischen Daten der Kinect2-Tiefenkamera sowie die eingesetzte Skeletttracking-Toolbox von Microsoft. Anhand der dabei festgestellten Möglichkeiten und Einschränkungen wird in Kapitel 3 die Notwendigkeit einer schwenkbaren Tiefenkamera zur verbesserten Pa-

tientenbeobachtung motiviert. Dabei werden verschiedene Steuerkriterien entwickelt und anschließend experimentell bewertet. Bewertet wird hierbei primär die Position und Lage des Patienten im Sensorbereich und nicht die Qualität des Skeletttrackings, da diese sich außerhalb des Fokus der Masterarbeit bewegt. Schließlich wird in Kapitel 4 die gangbasierte Wiedererkennung von Personen als möglicher Anwendungsfall für die gewonnenen Tiefendaten untersucht und basierend auf einer prototypischen Umsetzung evaluiert. In beiden Teilen werden jeweils zu Beginn notwendige Grundlagen erläutert und ein Einblick in den entsprechenden State-of-the-Art gegeben. Danach werden die im Rahmen dieser Masterarbeit entwickelten Verfahren dargelegt und in jedem Abschnitt separat experimentell evaluiert. Die Masterarbeit schließt mit einer kompakten Zusammenfassung der Ergebnisse und stellt die Integration in die ROGER-Gesamtapplikation als themenübergreifenden Ausblick dar.

# Kapitel 2

# Grundlagen

Bevor in den beiden Schwerpunktabschnitten nochmals auf spezifische Grundlagen eingegangen wird, soll der folgende Abschnitt einen grundlegenden Überblick über die Roboterplattform und die verwendete Kinect2-Tiefenkamera sowie deren begleitendes Softwarepaket geben.

## 2.1 Roboterplattform „Ringo“

Der Roboter „Ringo“ ist ein mobiler Assistenzroboter welcher auf der SCITOS-G3-Plattform der MetraLabs GmbH Ilmenau aufbaut und am Fachgebiet Neuroinformatik und Kognitive Robotik der TU Ilmenau im Projekt ROGER eingesetzt wird.

### 2.1.1 Plattform und Sensorik

Durch seine kompakte Grundfläche von 45 cm × 55 cm bei einer Gesamthöhe von 1,5 m eignet er sich für öffentliche Einsatzumgebungen. Angetrieben wird die etwa 80 kg schwere Plattform durch einen differentiellen Radantrieb mit zwei separat ansteuerbaren Rädern und einem unmotorisierten Kastorrad zur Stabilisierung.
Die Verarbeitung der Sensorinformationen erfolgt über zwei fest mit Ringo verbundene Computer. Beide sind mit Mobilprozessoren der Intel-Core-i7-Familie ausgestattet. Einer der beiden Rechner hat zusätzlich die Aufgabe den Antrieb über dedizierte Motorcontroller anzusteuern.

Zur Darstellung von Daten, Benutzeroberflächen und zum Erfassen von Benutzereingaben verfügt Ringo über zwei Touchdisplays. Die Monitore sind in unterschiedlichen Höhen angebracht um eine Bedienung im Sitzen und aus dem Stand zu ermöglichen (siehe dazu auch Abbildung 1.1(a))

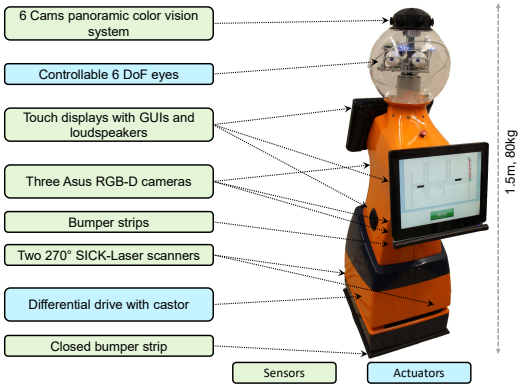



**Abbildung 2.1:** Roboterplattform „Ringo“ mit Verkleidung
*Überblick über die technischen Daten des Roboters „Ringo“ sowie über die verbaute Sensorik. Darstellung basierend auf* `http://www.tu-ilmenau.de/fileadmin/media/neurob/research/robots/ringo/RINGO-2016-03-features-diagram.pdf`

Das Sensorsystem des Roboters besteht aus einer omnidirektionalen Kameraanordnung bestehend aus vier Einzelkameras auf dem Kopf, drei RGB-D-Kameras von Asus sowie zwei Laserentfernungssensoren von SICK. Die Informationen der omnidirektionalen Farbkamera werden dabei primär für Personendetektion, -verfolgung und -wiedererkennung eingesetzt. Um diese Aufgaben robuster erfüllen zu können, werden zusätzlich die Daten der Laserentfernungsmesser genutzt [Volkhardt et al., 2013]. Ferner werden die gewonnenen Laserdaten zusammen mit den Tiefendaten der drei Asus-Kameras für Lokalisation, Navigation und Hindernisvermeidung genutzt. Um im Falle einer Kollision mit einem Hindernis eine Abschaltung der Motoren sicherzustellen, verfügt Ringo über eine umlaufende Leiste von sogenannten „Bumper“-Sensoren. Diese lösen bei Kontakt mit dem Hindernis eine entsprechende Sicherheitsabschaltung des Antriebs aus.

Zusätzlich zur vorgestellten Sensorik wird im ROGER-Projekt noch eine Kinect2-Tiefenkamera von Microsoft eingesetzt. Diese wird beweglich auf einer sogenannten Schwenk-Neige-Einheit montiert. In Neutralstellung ist die Kamera dabei entgegen der Fahrtrichtung ausgerichtet um Personen, die hinter dem Roboter laufen erfassen zu können. Eine detailliertere Vorstellung der Kinect2-Kamera erfolgt direkt anschließend im Abschnitt 2.2. Details zur Montage der Kamera sowie der Schwenk-Neige-Einheit werden als Teil von Kapitel 3 behandelt.

### 2.1.2 Software

Softwareseitig sind beide Rechner mit einem Standard-Linux-Betriebssystem ausgestattet. Als Abstraktionsschicht zwischen der Roboterhardware und komplexeren Robotikanwendungen wird die Middleware MIRA eingesetzt [Einhorn et al., 2012]. Dabei handelt es sich um ein hauptsächlich in C++ geschriebenes Framework, dass von der MetraLabs GmbH Ilmenau sowie dem Fachgebiet Neuroinformatik und Kognitive Robotik entwickelt und gepflegt wird.

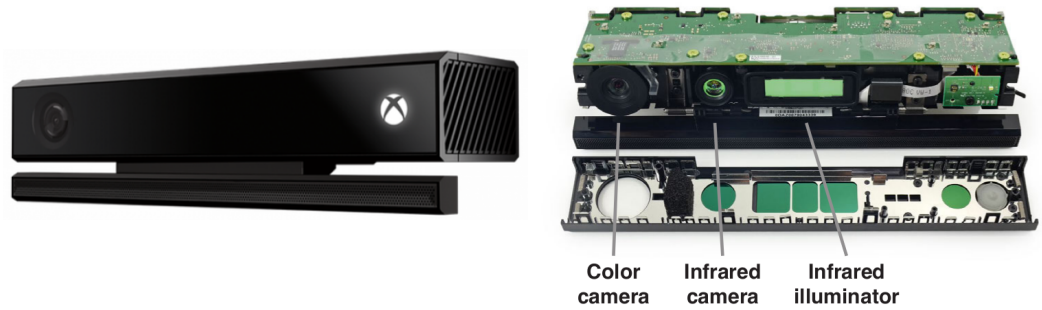
Der Fokus der Middleware liegt dabei darauf die Entwicklung und Nutzung von komplexen, modularen und verteilten Anwendungen zu ermöglichen. Dazu werden effiziente und leistungsfähige Mechanismen zur Intra- und Interprozesskommunikation sowie zur Konfiguration bereitgestellt. Dadurch sollen modularer Entwurf, Einsatz und Wartung von dynamisch konfigurierbaren meist roboterbasierten Anwendungen in komplexen alltagsnahen Szenarien ermöglicht werden. Ferner stellt MIRA eine integrierte Paketverwaltung zur Verfügung, welche den Zugriff und die Versionierung der verschiedenen modular entwickelten Teilkomponenten ermöglicht.

Zusätzlich wird eine Reihe von Werkzeugen zur Verfügung gestellt, welche beispielsweise die Darstellung oder Aufzeichnung von Sensordaten zur Laufzeit ermöglichen. Die aufgezeichneten Daten lassen sich anschließend wiedergeben um weitergehende Analysen zu ermöglichen.

## 2.2 Kinect2-Tiefenkamera

Bei der Kinect2 handelt es sich um eine Farb- und Tiefenkamera (RGB-D) von Microsoft. Sie wurde im November 2013 als Ergänzung zur Spielekonsole Xbox One vorgestellt. Ursprünglich als Erweiterung zur Gestensteuerung von Spielen und Home-Entertainment-Anwendungen konzipiert, fand die Kinect2 wie bereits ihr Vorgänger unter der Bezeichnung *Kinect for Windows* ab etwa Mitte 2014 Einzug in verschiedene Bereiche der Robotik- [Fankhauser et al., 2015] und teilweise auch der Medizinforschung [Pöhlmann et al., 2016]. Ausschlaggebend dafür ist vor allem der geringe Preis von unter 200 Euro pro Stück inklusive Zubehör, der deutlich unter dem Niveau vieler dedizierter Tiefenkameras von teilweise mehreren Tausend Euro liegt, bei einer gleichzeitig guten Abbildung von 3D-Szenen.

### 2.2.1 Hardware




(a) Kinect2-Sensor (geschlossen) (b) Kinect2-Sensor (offen)

**Abbildung 2.2:** Kinect2-Sensor
*Der Kinect2-Sensor besteht aus einer Full-HD-Farbkamera und einer aktiven Time-of-Flight-Tiefenkamera, bestehend aus Infrarot-Kamera und Infrarot-LEDs [Fankhauser et al., 2015]. Zusätzlich integriert Microsoft ein Multi-Array-Mikrofon, dass eine grobe Ortung von Geräuschquellen erlaubt.*

Anders als noch beim gleichnamigen Vorgänger, setzt Microsoft bei der Kinect2 für Tiefenaufnahmen die sogenannte Time-of-Flight-Methode ein. Zur Erfassung von Tiefendaten beleuchtet die Kinect2 dabei die Szene mit Lichtblitzen im infraroten Spek-

trum und ermittelt die Entfernung zu Objekten über die Laufzeit der Impulse. Es handelt sich also um eine aktives Sensorprinzip. Das erfasste Tiefenbild besteht aus $512 \times 424$ Pixeln (Öffnungswinkel horizontal/vertikal 70°/60°) und wird bis zu 30 mal pro Sekunde erfasst [Microsoft Corporation, 2014c]. Tiefendaten lassen sich etwa in einem Bereich von 0,5 m bis 4,5 m zuverlässig erfassen. Es sind auch Tiefenmessungen darüber hinaus möglich, allerdings mit stetig abnehmender Genauigkeit. Alle Daten wie beispielsweise Tiefen- und Farbbilder werden über einen USB3-Anschluss übertragen.

### 2.2.2 Kinect for Windows SDK 2.0

Zusätzlich zur bereits beschriebenen Hardware stellt Microsoft ein aktualisiertes Software-Paket (Software Developement Kit, kurz: *SDK*) für Entwickler im nichtkommerziellen Bereich zur Verfügung. Das ausschließlich unter Windows lauffähige SDK wird dabei häufig Synonym für eine Sammlung verschiedener Programme genutzt. Ein wichtiger Bestandteil des Softwarepakets stellt dabei das Kinect Studio dar. Es bietet verschiedene Funktionen um die über USB übertragenen Kinect-Daten anzuzeigen, aufzuzeichnen und wiederzugeben. Für die Datenaufnahme wird dabei ein proprietäres Dateiformat von Microsoft mit dem Namen *eXtended Event File* (kurz: *XEF*)[1] genutzt. Die entsprechenden Dateien lassen sich nur über das Kinect Studio oder die entsprechende C#-Programmierschnittstelle (*API*)[2] in .NET auslesen.

### 2.2.3 SDK-Skeletttracker

Ein weiterer Bestandteil des vorgestellten SDK ist eine von Microsoft entwickelte Lösung zur Schätzung eines Skelettes aus den Tiefendaten der Kinect2. Es handelt sich dabei um einen proprietären Algorithmus der aus den Tiefendaten Gelenkpunkte von bis zu sechs Personen schätzen und verfolgen kann. Jedes Skelett besteht dabei aus 25 Skelettpunkten, welche in Abbildung 2.3 dargestellt sind.

[1] https://msdn.microsoft.com/de-de/library/dn785306.aspx
[2] https://msdn.microsoft.com/en-us/library/microsoft.kinect.tools.aspx

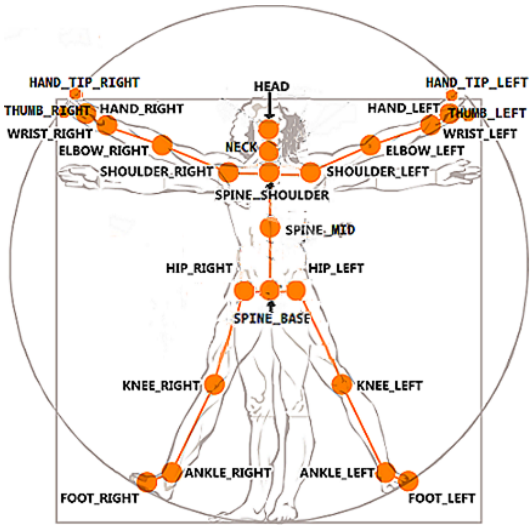



**Abbildung 2.3:** Kinect2-SDK-Skelettpunkte
*Die Darstellung zeigt alle 25 Skelettpunkte, die vom Skeletttracker des Kinect2-SDK aus den Tiefendaten ermittelt werden und ihre entsprechenden Namen in der SDK-API [*Microsoft Corporation*, 2014b].*

Aufgrund des ursprünglichen Entwicklungsziels die Kinect2 für die Steuerung von Bewegungsspielen beziehungsweise Heimkinoanwendungen, erfolgt die Schätzung ausgehend von Oberkörper und Gesicht. Dementsprechend filigran ist daher auch die Segmenteinteilung des Oberkörpers, der oberen Extremitäten insbesondere der Hand und des Gesichts. Für Aspekte der Ganganalyse sind vor allem die Gelenkpunkte im Bein- und Hüftbereich (Hüfte, Knie, Sprunggelenk, Vorfuß) sowie die Wirbelsäulenpunkte im Bereich des Kreuzbeins und der mittleren Brustwirbel von Interesse.

Aus den vom SDK festgelegten Grenzen für zuverlässige Tiefendaten ergibt sich ein Bereich von 0,5 m bis maximal 4,5 m Entfernung indem eine Skelettschätzung durchgeführt wird. Praktisch kommt es aber bereits ab einer Entfernung jenseits von rund 3,5 m zunehmend zu Ungenauigkeiten, vor allem im Bereich der unteren Extremitäten. Dieser Wert wird daher im Rahmen der Masterarbeit als obere Distanzgrenze des zuverlässigen Sensorbereichs angenommen.

In der Horizontalen ist der effektive Bereich in dem ein Skeletttracking möglich ist durch den Öffnungswinkel der Tiefenkamera von 70° beschränkt. Da im Rahmen der

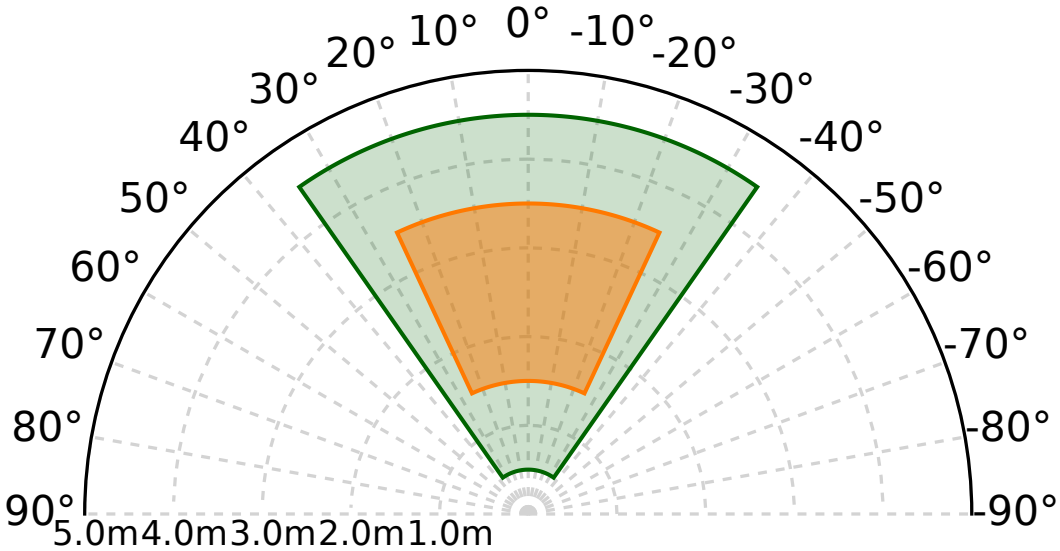



**Abbildung 2.4:** Sensorbereich des Kinect2-Tiefensensors und Zielbereich

*Der Sensorbereich laut Spezifikation (*$0{,}5 - 4{,}5\text{m},\ 70°$*) ist in grün dargestellt. Der orange markierte Sektor markiert den Bereich (*$1{,}5 - 3{,}5\text{m},\ 50°$*) indem sich eine Person aufhalten kann (Körpermitte) um bei einer optimalen Sensorposition vollständig und zuverlässig vom Skeletttracker erfasst zu werden.*

Arbeit die Position von Personen nur als Punkt (Körpermitte) betrachtet wird, muss ein zusätzlicher Saum beachtet werden. Für die Auswertung wird der Sensorbereich daher von 70° (±35°) auf 50° (±25°) eingeschränkt. Der nominelle Sensorbereich und der Zielbereich werden in Abbildung 2.4 graphisch verglichen.

# Kapitel 3

# Stabile Erfassung von Personen

Beim Einsatz in dynamischen öffentlichen Einsatzumgebung, werden hohe Anforderungen an die Navigationsleistung der Roboterplattform gestellt. Dies gilt vor allem in beengten Umgebungen wie beispielsweise Klinikfluren. Ausweichmanöver zum Zweck der Hindernisvermeidung oder im Rahmen des sozial akzeptablen Verhaltens gegenüber Personen sind unvermeidbar. Durch den nicht-holonomen Aufbau der eingesetzten Roboterplattform, ergeben sich dabei immer wieder Situationen in denen der Patient nicht mehr im Sensorbereich der im Gangtraining eingesetzten Kinect2-Tiefenkamera ist.

Durch den Einsatz einer Pan-Tilt-Unit (kurz: *PTU*, seltener zu deutsch Schwenk-Neige-Einheit) besteht die Möglichkeit die Kamera in zwei zusätzlichen rotatorischen Freiheitsgraden bis zu einem gewissen Grad unabhängig von der Roboterplattform zu positionieren. Diese Freiheitsgrade erlauben es in einigen Anwendungsfällen die Nachteile des nicht-holonomen Roboters auszugleichen um so den Nutzer besser im Blick zu behalten.

Nach einer kurzen Vorstellung der PTU-Hardware und einigen einleitenden Grundlagen, erfolgt zunächst ein Überblick zum State-of-the-Art aus dem Bereich *visual servoing* sowie angelehnter Verfahren. Anschließend wird die im Rahmen der Masterarbeit entwickelte Ansteuerung der PTU vorgestellt und experimentell evaluiert. Der Teil schließt mit einer kurzen themenspezifischen Zusammenfassung.

## 3.1 Montage und Hardware

Als Pan-Tilt-Unit kommt eine *PTU-D46-17.5* von Directed Perception zum Einsatz. Sie verfügt über einen Schwenkbereich von ±159° der ohne Last mit maximal 300°/s abgefahren werden kann. Die Neigung kann in einem Bereich von −47° bis 31° gewählt und ohne Last mit der gleichen Maximalgeschwindigkeit verfahren werden. Die Angaben gelten laut Hersteller bei maximal möglicher Versorgungsspannung von 36 VDC. Da der Roboter über die integrierte Spannungsversorgung nur maximal 24 VDC zur Verfügung stellt, werden bei montierter Kinect2 Geschwindigkeiten von etwa 150°/s erreicht.

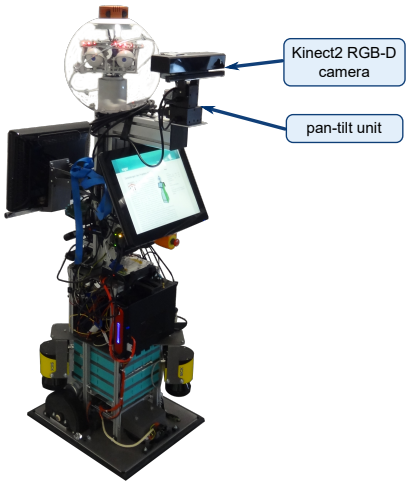



**Abbildung 3.1:** Ringo ohne Gehäuse mit PTU und Kinect2
*Die PTU wurde an einem zusätzlichen, etwa* 30 cm *langen Aluprofil an der Stützstruktur von Ringo in einer Höhe von* 1,20 m *befestigt. Durch die Montage auf der PTU befindet sich die Kinect2 auf etwa* 1,30 m.

Die PTU wird über einen externen Controller angesteuert, der über einen RS232-Anschluss mit einem der PCs auf Ringo verbunden wird. Über die serielle Verbindung können ASCII-Kommandos an die PTU gesendet werden, die eine Steuerung von Position und Geschwindigkeit im laufenden Betrieb ermöglichen. Zusätzlich besteht die Möglichkeit initial restriktivere Begrenzungen für Schwenk- und Neigewinkel zu wäh-

len. Ein vollständiger Überblick über alle Funktionen findet sich im online-verfügbaren Handbuch [Directed Perception, 2006].
Da die PTU bereits in einem früheren Projekt am Fachgebiet Neuroinformatik und Kognitive Robotik eingesetzt wurde, existiert ein MIRA-Paket, welches die wichtigsten ASCII-Kommandos gekapselt in C++-RPCs (englisch *remote procedure calls*) zur Verfügung stellt. An diesem Paket mussten im Rahmen der Masterarbeit nur kleine Veränderungen vorgenommen werden, um es an den aktuellen Stand der Middleware anzupassen.

## 3.2 Theoretische Grundlagen

Bevor die Ansteuerung abgeleitet wird, sollen im kommenden Abschnitt einige Begriffe und Konzepte vorgestellt werden, die bei deren Herleitung und Validierung Anwendung finden.

### Visual Servoing

*visual servoing* beschreibt die Aufgabe Roboter basierend auf den Informationen visueller Sensoren anzusteuern [Chaumette und Hutchinson, 2006]. Im Gegensatz zur bisherigen impliziten Verwendung des Begriffs „Roboter“ im Sinne einer mobilen Plattformen, umfasst diese Definition in der Literatur auch stationäre Systeme wie etwa Industrieroboter.
Ausgehend von dieser Definition werden in der Literatur zwei verschiedene Ausgangsszenarien beschrieben. Zum einen der sogenannte *eye-to-hand*-Fall, bei dem eine externe Kamera den Roboter und seine Umgebung beobachtet. Zum anderen die *eye-in-hand* genannte Situation bei dem der visuelle Sensor mit dem Roboter verbunden ist und die Szene relativ zu diesem wahrnimmt [Chaumette und Hutchinson, 2006].
Zusätzlich zur relativen Position der Kamera wird bei der Klassifikation von Visual-Servoing-Systemen häufig noch nach der Art der Merkmale entschieden, die zur Ableitung der Steuerung genutzt werden. Beim *image based visual servoing* (kurz: *IBVS*) werden die Merkmale direkt aus den Bildinformationen beispielsweise anhand mar-

kanter Punkte bestimmt. IBVS-Ansätze werden bereits seit Anfang der 1980er Jahre erforscht, beispielsweise in [Agin, 1980], [Weiss et al., 1987] oder [Feddema und Mitchell, 1989]. Eine Weiterentwicklung dieser rein bildbasierten Merkmale stellen Systeme aus dem Bereich des *position based visual servoing* (kurz: *PBVS*) dar. Dabei wird häufig Vorwissen über die Sensorik, Szene und darin enthaltene Objekte genutzt um eine 3D-Rekonstruktion/-Posenschätzung und -Steuerung durchzuführen. Arbeiten dazu finden sich etwa in [Weiss et al., 1987], [Wilson et al., 1996] oder [Thuilot et al., 2002].

### Koordinatensysteme und -transformationen

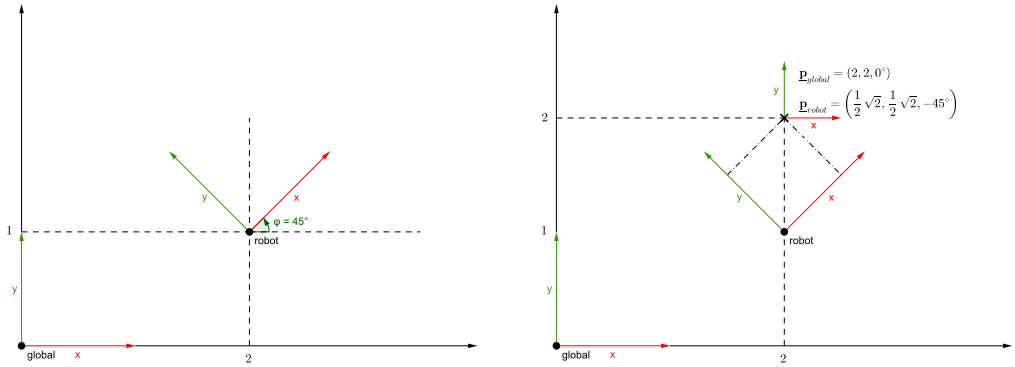
In der Robotik ist es üblich, mehr als ein Bezugssystem für die Angabe von Positionen und Posen zu nutzen. Dadurch erleichtern sich Beschreibungen und Berechnungen in vielen Situationen. Die einzelnen Koordinatensysteme können dabei durch Koordinatentransformationen ineinander überführt werden. Dabei bestehen die Transformationen stets aus Rotation und Translation. Während die Beschreibung der Translation in aller Regel eindeutig ist, besteht eine Reihe von Möglichkeiten die Rotation zu beschreiben. Beispielsweise finden je nach Anwendung Rotationsmatrizen, Euler-Winkel oder auch Quaternionen Verwendung. Jede Darstellungsform hat dabei spezielle Vor- und Nachteile, auf die im Folgenden aber nicht näher eingegangen werden soll. In jeder der genannten Darstellungsformen können Translation und Rotation sowohl fix als auch zeitvariant sein.

**Notation und Terminologie** Eine *Position* beschreibt die Lage eines Punktes ($n$ translatorische Freiheitsgrade in einem $n$-dimensionalen Raum) in einem Bezugs- beziehungsweise Koordinatensystem $V$. Um die Lage eines starren Körpers (englisch *rigid body*) vollständig zu beschreiben, wird zusätzlich die Orientierung ($n(n-1)/2$ rotatorische Freiheitsgrade in einem $n$-dimensionalen Raum) angegeben. Die kombinierte Beschreibung von Lage und Orientierung wird dann als *Pose* $\underline{\mathbf{p}}_V$ in $V$ bezeichnet.
Im Folgenden wird eine abstrakte Notation verwendet, welche von der letztlichen Realisierung der Transformation möglichst unabhängig sein soll. Sei $\underline{\mathbf{T}}_{W \leftarrow V}$ die Transformation, welche eine Pose aus dem $n$-dimensionalen Bezugssystem $V$ in das ebenfalls

$n$-dimensionale Bezugssystem $W$ überführt und $\underline{\mathbf{T}}_{W\leftarrow V}^{-1} = \underline{\mathbf{T}}_{V\leftarrow W}$ ihre Inverse. Ferner gelte $\underline{\mathbf{T}}_{W\leftarrow V} \circ \underline{\mathbf{T}}_{V\leftarrow U} = \underline{\mathbf{T}}_{W\leftarrow U}$ (Transitivität). Zur Vereinfachung der Notation wird ferner angenommen, dass die Transformationen nicht nur auf Posen sondern auch analog auf Positionen anwendbar sind.

**Beispiel** In einem einfachen Fall sei ein externes, zweidimensionales, kartesisches Koordinatensystem gegeben, in dem ein Beobachter die Pose des Roboters angibt. In Abbildung 3.2(a) befindet sich der Roboter zum aktuellen Zeitpunkt im externen Bezugssystem an den Koordinaten $\underline{\mathbf{x}}_{global} = (2,1)^{\mathrm{T}}$ und ist um 45° zu diesem gedreht (Pose: $\underline{\mathbf{p}}_{global}^{robot} = (2,1,45°)^{\mathrm{T}}$). Betrachtet wird nun ein Objekt mit der Pose $\underline{\mathbf{p}}_{global}^{sample} = (2,2,0°)$ (Abbildung 3.2(b)). Durch Anwendung der Transformation $\underline{\mathbf{T}}_{robot\leftarrow global}$ auf $\underline{\mathbf{p}}_{global}^{sample}$ ergibt sich deren Entsprechung im lokalen Roboterkoordinatensystem als:

$$\underline{\mathbf{p}}_{robot}^{sample} = \underline{\mathbf{T}}_{robot\leftarrow extern} \circ \underline{\mathbf{p}}_{global}^{sample} = \left(\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2}, -45°\right)$$




(a) Globales und lokales Koordinatensystem (b) Pose in globalen und lokalen Koordinaten

**Abbildung 3.2:** 2D-Beispiel für Koordinatensysteme und -transformation
*Dargestellt ist ein einfaches Beispiel für zwei Koordinatensysteme in einem zweidimensionalen Raum. In der rechten Graphik wird die Transformation einer Beispiel-Pose zwischen den zwei Koordinatensystemen geometrisch vorgeführt. Dazu wird die Position auf die aufspannenden Basisvektoren des jeweiligen Koordinatensystems projiziert. Für die Umrechnung der Orientierung wird der Winkel der x-Achse der Pose zur x-Achse der jeweiligen Basis ausgewertet.*

**Darstellung** Die Darstellung von Koordinatensystemen erfolgt ähnlich wie in Abbildung 3.2 meist über gefärbte Pfeile. Im Rahme dieser Arbeit wird dafür die übliche Farbkonvention für die einzelnen Achsen genutzt:

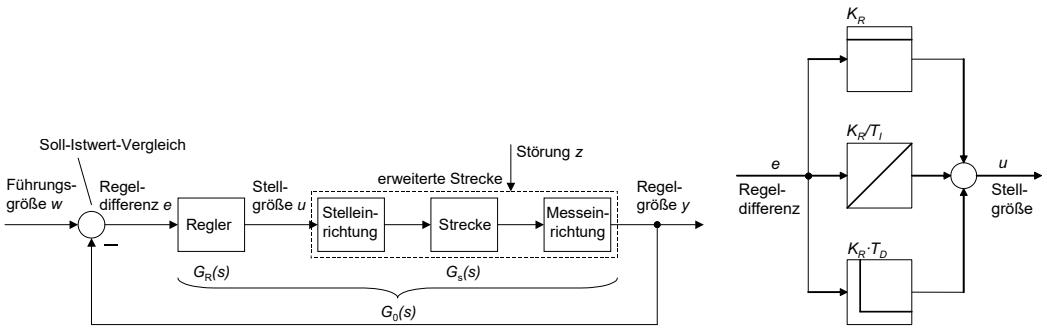
- x-Achse: rot
- y-Achse: grün
- z-Achse: blau

In zweidimensionalen Ansichten wird bei dreidimensionalen Koordinatensystem die dritte Achse, wenn sie in die Bildebene hinein zeigt mit $\times$ gekennzeichnet, andernfalls mit $\odot$. Oft wird die Markierung auch weggelassen, da es sich stets um rechtshändige Koordinatensysteme handelt und die Richtung der dritten Achse durch die beiden anderen eindeutig bestimmt wird.

Im Rahmen der folgenden Darstellungen werden Posen ebenfalls nach diesem Schema markiert. Wo möglich, werden die Achsen der Pose dabei kürzer dargestellt um die visuelle Unterscheidbarkeit zu verbessern.

### PID-Regler

Allgemein handelt es sich bei einem Regler um eine Systemkomponente welche die Ausgabe einer zu regelnden Strecke (Ist-Wert) durch geeignete Stellgröße auf einen vorgegebenen Soll-Wert bringen soll. Dazu wird der Systemausgang *zurückgeführt* und mit der Führungsgröße verglichen. Der Unterschied zwischen Soll- und Ist-Wert wird gemeinhin als Regeldifferenz oder -abweichung bezeichnet. Die sich daraus ergebende Feedback-Schleife ist in Abbildung 3.3 gut zu erkennen. In der englischsprachigen Literatur wird eine Regelung daher auch *closed-loop control* genannt.




(a) Blockschaltbild Standard-Regelkreis

(b) Grundstruktur PID-Regler

**Abbildung 3.3:** Grundlagen Regelkreis und Regler
*links: Blockschaltbild des zeitkontinuierlichen Standard-Regelkreises aus der Regelungs- und Systemtechnik. Durch die Rückführung des Systemausgangs ergibt sich die sogenannte Feedback-Schleife. rechts: paralleler PID-Regler (von oben: Proportional-, Integral- und Differentialkomponente) als mögliche Realisierung eines Reglers (beides nach [*Ament*, 2013]).*

Beim sogenannten PID-Regler handelt es sich nun um eine konkrete Möglichkeit die Regler-Komponente des Systems zu realisieren. Der Ansatz ist anhand des Blockschaltbilds Abbildung 3.3(b) leicht nachzuvollziehen. Das Fehlersignal $e$ wird parallel auf drei

„Teil-Regler“ gegeben, deren Ausgaben anschließend linear überlagert werden und die Stellgröße bilden.

- **P**roportional-Regler: Ausgabe ist direkt proportional zur Regeldifferenz $e$
- **I**ntegral-Regler: Ausgabe ist direkt proportional zum Integral über $e$ (verbesserte stationäre Genauigkeit, schlechtere Dynamik)
- **D**ifferential-Regler: Ausgabe ist proportional zur Änderung der Regelabweichung $e$ (schlechtere stationäre Genauigkeit, bessere Dynamik)

Durch die Wahl entsprechender Skalierungsfaktoren kann die Charakteristik des Reglers angepasst werden. Meist erfolgt eine iterative Bestimmung der Reglerparameter nach einer entsprechenden Modellierung der Strecke und bekannter Störeinflüsse in einer Simulationsumgebung wie MATLAB Simulink.

Die hier für die zeitkontinuierliche Modellierung getroffenen Aussagen treffen analog in den meisten Fällen auch unverändert für zeitdiskrete Realisierung zu. Allerdings ergeben sich in zusätzliche Beschränkungen durch die Abtastung des Systems. Diese betreffen vor allem die Möglichkeiten bei der Steuerung hin zu einer verbesserten Dynamik.

Für eine detaillierte Einführung in die Regelungs- und Systemtechnik, inklusive Modellierung, sei an dieser Stelle auf die Grundlagenliteratur etwa nach [Lunze, 2010] oder [Föllinger, 2016] verwiesen.

## 3.3 State-of-the-Art

Grundlegend basieren die meisten Visual-Servoing-Ansätze oder ähnliche Aktorsteuerverfahren auf der Minimierung eines Fehlers $\underline{\mathbf{e}}(t)$, der sich generell häufig wie folgt definieren lässt:

$$\underline{\mathbf{e}}(t) = \underline{\mathbf{s}}(\underline{\mathbf{m}}(t), \underline{\mathbf{a}}) - \underline{\mathbf{s}}^{\star}(t). \tag{3.1}$$

$\underline{\mathbf{s}}$ beschreibt in diesem Kontext eine Funktion, die Sensordaten $\underline{\mathbf{m}}(t)$ (Bilder, Beschleunigungswerte, ...) zusammen mit (optionalem) Vorwissen $\underline{\mathbf{a}}$ (beispielsweise Sensormodelle, Vorkenntnisse über die Geometrie des betrachteten Objekts, ...) auf einen $k$-dimensionalen Merkmalsvektor (Bildkoordinaten von markanten Punkten, Posen, Winkelgeschwindigkeiten, ...) abbildet. Der Fehler ergibt sich dann aus der Differenz des aktuellen Merkmalsvektors zum erwarteten Merkmalsvektor $\underline{\mathbf{s}}^{\star}(t)$ (mit $\underline{\mathbf{s}}^{\star}(t) = \underline{\mathbf{s}}^{\star} = const.$ für statische Ziele) [Chaumette und Hutchinson, 2006].
Wie bereits in den einleitenden Grundlagen erwähnt, unterscheiden sich die Visual-Servoing-Spielarten IBVS und PBVS deutlich bei der Gestaltung von $\underline{\mathbf{s}}$. Da die Ansteuerung im hier betrachteten Anwendungsfall in eine komplexe robotische Applikation eingebunden wird, die bereits über leistungsfähige Sensorik und darauf aufbauende Algorithmik verfügt, wird das Design der Merkmale für die einzelnen Verfahren nicht weiter vertieft. Der Fokus wird primär auf die in den einzelnen Publikationen realisierten Steuerverfahren sowie deren Randbedingungen und Grenzen gelegt. Außerdem werden im Folgenden nur Anwendungen betrachtet, die ihr Hauptaugenmerk nicht auf die Steuerung industrieller Roboterarme legen.

### ViSP-Toolbox

Eine sehr umfassende und gut dokumentierte Sammlung von verschiedenen Algorithmen aus dem Bereich des *visual servoing* sowie verschiedener anderer Computer-Vision-Aufgaben wird unter dem Namen *Visual Servoing Platform* [Marchand et al., 2005] (kurz *ViSP*) vom französischen Nationalen Forschungsinstitut für Informatik und Automatisierung (*INRIA*) zur Verfügung gestellt und gepflegt. Die Toolbox verfügt über eine Reihe von bereits vollständig implementierten Verfahren, welche die

Steuerung von Robotersystemen (mobil oder statisch) basierend IBVS- oder PBVS-Algorithmen ermöglichen. Die formale Notation sowie die Auswahl der implementierten Verfahren orientiert sich dabei an [Chaumette und Hutchinson, 2006], da der Autor dem Institut angehört. Neben der C++-Toolbox, existiert eine Anbindung an die freie Robotik-Middleware ROS [Quigley et al., 2009].

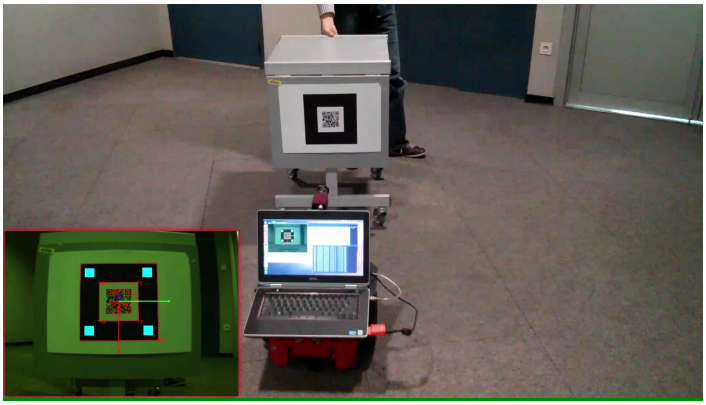


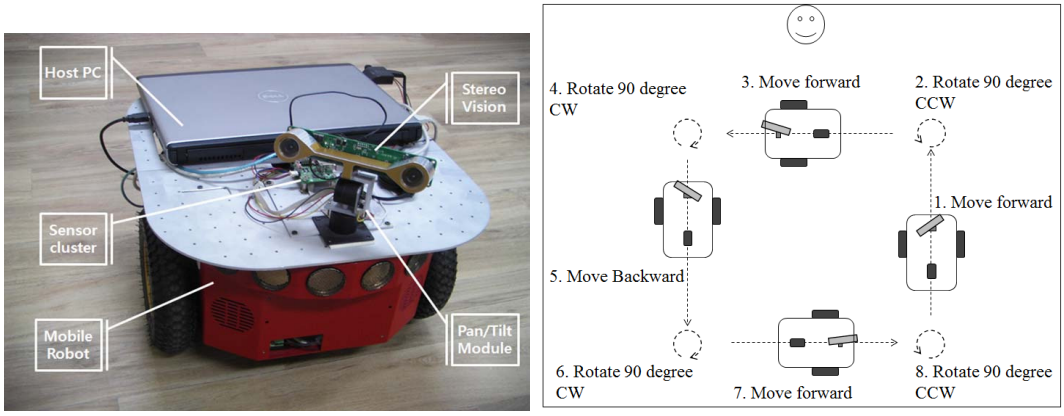
**Abbildung 3.4:** Visual-Servoing eines Pioneer 3D-DX mit ViSP
*Der differentielle Radantrieb des Pioneer 3D-DS wird über PBVS durch Verschieben des QR-Codes gesteuert. Das Ziel ist es, den Roboter frontal zum QR-Code auszurichten.*

Basierend auf dieser Schnittstelle existieren Beispiele wie sich ein Pioneer P3-DX mittels PBVS in einer Laborumgebung steuern lässt (siehe Abbildung 3.4). Beide Tatsachen sprechen für die Praxistauglichkeit der implementierten Verfahren und haben daher generell das Potential auch in komplexeren Szenarien wie etwa im ROGER-Projekt anwendbar zu sein. Um eine schnelle und effektive Entwicklungsarbeit mit der Toolbox zu gewährleisten, ist jedoch einige Einarbeitung oder die Realisierung einer entsprechenden Anbindung an MIRA notwendig.

## Mobile Plattformen

In [Bahn et al., 2011] wird eine einfache Winkelsteuerung für eine PTU realisiert, die auf einer nicht-holonomen Roboterplattform (Pioneer P3-DX) entsprechend

Abbildung 3.5(a) montiert ist. Die Steuerung ergibt sich dabei aus dem von den Autoren *motion-informed visual tracking* genannten Algorithmus.

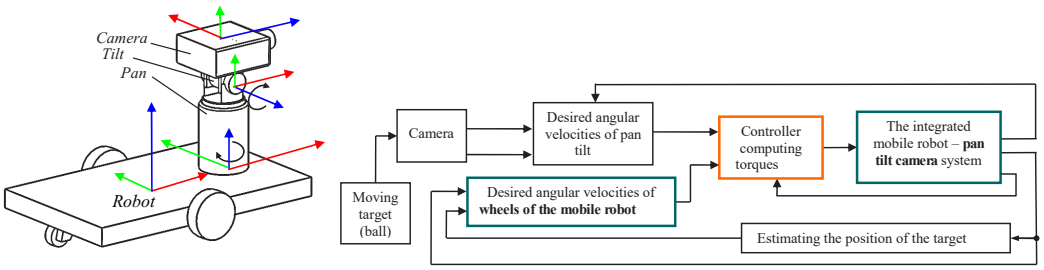



(a) Pioneer P3-DX mit PTU (b) Untersuchtes Szenario

**Abbildung 3.5:** Visual-Servoing-Setup bei [Bahn et al., 2011]
*(a) Nicht-holonomer Pioneer P3-DX mit differentiellen Radantrieb, Pan-Tilt-Unit und modifizierter BumbleBee2-Stereokamera. (b) Steueraufgabe für das visual servoing. Das Ziel wird in [Bahn et al., 2011] als Smiley veranschaulicht.*

Als primäre Quelle für die Visual-Servoing-Steuerung dient die auf der PTU schwenkbar montierte RGB-D-Kamera (BumbleBee2). Mit dieser Kamera wird eine initiale Suche nach dem statischen Ziel durchgeführt und dessen Position relativ zum Roboter basierend auf der Disparität berechnet. Während der Fahrt werden Radencoder- sowie Gyroskopmessdaten erfasst, um die Position des Roboters unter Einbeziehung der bekannten Kinematik zu schätzen und davon ausgehend die Schwenk- und Neigewinkel zu korrigieren. Zusätzlich wird laufend die Position des Ziels im Kamerabild ausgewertet. Ist diese trotz Odometrie-Korrekturen nicht mehr zentriert, erfolgt eine neue bildbasierte Berechnung der Steuerwinkel, bevor wieder in den Korrekturmodus gewechselt wird. Da sich die Experimente auf Laborbedingungen (rechteckige Trajektorie entsprechend Abbildung 3.5(b), maximale Geschwindigkeit $v_{max} = 0{,}2\,\mathrm{m/s}$, maximale Drehgeschwindigkeit $v_{rot,max} = 30\,°/\mathrm{s}$) und ein statisches Ziel beschränken, wird auf eine explizite Geschwindigkeitssteuerung verzichtet. Als Gütemaß ermitteln die Autoren die Winkelabweichung der direkten Sichtachse auf das Ziel von der Sichtachse der Ka-

mera. Ausgehend von zehn Testfahrten entlang der Trajektorie aus Abbildung 3.5(b) ergibt sich für das beschriebene System eine maximale Winkelabweichung von ±6,5°.



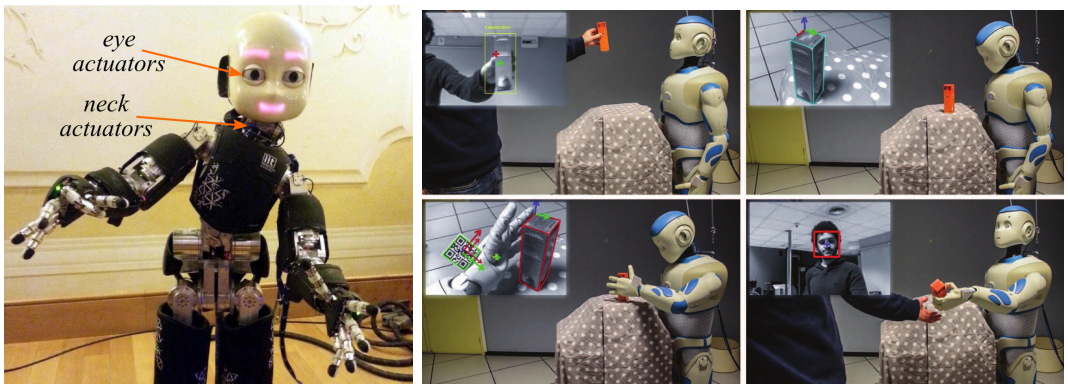

(a) Robotermodell (b) Ablauf der Steuerung

**Abbildung 3.6:** Visual-Servoing-Setup in [Tinh et al., 2014]
*(a) Für die Modellierung eingesetztes Robotermodell. (b) Übersichtsdiagramm der vorgeschlagenen Ansteuerung. Die grün hervorgehobenen Komponenten markieren Roboter und PTU, die beide durch den Drehmoment-Regler (orange) kontrolliert werden.*

Deutlich umfangreicher gestaltet sich der Fokus von [Tinh et al., 2014]. Unter Einbeziehung der Kinematik erfolgt hier der Entwurf eines Geschwindigkeitsreglers mit dem Ziel, eine sich bewegende Zielpose mittels einer schwenk- und neigbaren Kamera auf einem mobilen, nicht-holonomen Roboter (schematische Darstellung nach [Tinh et al., 2014] in Abbildung 3.6(a)) zu beobachten. Über eine umfangreiche Modellierung, basierend auf der Jacobi-Matrix des Systems, wird neben der Winkelgeschwindigkeit für die Pan- und Tilt-Aktorik zusätzlich noch die Steuerung des differentiellen Radantriebs ermittelt. Darauf aufbauend wird ein Drehmoment-Regler entworfen, der letztendlich die direkte Ansteuerung der Aktoren übernimmt. In den Entwurf des Reglers wird ein neuronales Netzwerk mit radialen Basisfunktionen integriert, welches die Unsicherheiten durch Modellfehler und Reibung mittels Online-Learning approximiert. Das damit aufgestellte Kontrollschema (Abbildung 3.6(b)) wird durch Simulation und den Nachweis von asymptotischer Stabilität validiert. Trotz soliden theoretischen Fundaments, erfolgt keine praxistaugliche Validierung etwa bezüglich des Verhaltens unter Rauscheinfluss. Daher kann anhand der Publikation keine Aussage über die in einem realen Szenario erreichbare Performanz getroffen werden.

Teilweise werden (visuelle) Kameratrackingsysteme beziehungsweise schwenkbar montierte Kameras auch ohne explizite Publikation in verschiedenen Robotikprojekten eingesetzt. Ein Beispiel dafür ist etwa das EU-Projekt STRANDS[1].

## Humanoide Roboter




(a) iCub (b) Romeo

**Abbildung 3.7:** Humanoide Roboter mit Visual Servoing
*(a) iCub-Roboter aus [Roncone et al., 2014]. Markiert sind die für das visual servoing genutzten Aktoren in Genick und den Augen. (b) Romeo-Roboter aus [Claudio et al., 2016]. Hier werden die Aktoren in Genick und Arm durch einen kombinierten IBVS/PBVS-Ansatz koordiniert.*

Jenseits der bisher vorgestellten Publikationen, existieren einige Arbeiten aus dem Bereich der humanoiden Robotik, die sich mit ähnlichen Fragestellungen beschäftigen. In der Arbeit von [Roncone et al., 2014] wird die Ansteuerung für den dem Mensch nachempfundenen Kopf des iCub [Metta et al., 2010] mit zwei Kameras abgeleitet. Als Ziel der Steuerung soll der Fixationspunkt der beiden Augenkameras unter Einfluss von Bewegung durch Ansteuerung der Aktoren in Hals und Augen stabil gehalten werden (Beispiel siehe Abbildung 3.7(a)). Die Autoren realisieren dazu einen Ansatz, der Informationen aus zwei verschiedenen Quellen nutzt. Zum einen das *Feedforward*-Komponente genannte Wissen über die Ansteuerung der einzelnen Aktoren unter Einbeziehung der bekannten Roboterkinematik. Zum anderen die Sensordaten

[1] `http://strands.acin.tuwien.ac.at/`

eines Beschleunigungssensors, genannt *Feedback-Komponente*, der externe Störeinflüsse erfassen kann. Da zunächst keine visuellen Informationen zur Steuerung genutzt werden, handelt sich nicht um *visual servoing* im engeren Sinne. Jedoch erfolgt die Bewertung der Ansteuerung in [Roncone et al., 2014] über die Auswertung des optischen Flusses.

Als weiteres Beispiel zeigen [Claudio et al., 2016] für den ebenfalls humanoiden Roboter Romeo, wie Verfahren aus IBVS und PBVS kombiniert werden können, um das Greifen von Objekten zu realisieren. Dazu wird ein vollständiger Ablauf von der Detektion des Objekts, über Posenschätzung bis hin zum Greifen und Übergeben beschrieben. Nach der Detektion des Objekts in der Szene wird ein IBVS-Algorithmus eingesetzt um das Objekt im Bild zu halten und die Zielpose für die Hand zu ermitteln (Abbildung 3.7(b), obere Reihe). Um Fehler und Unsicherheiten in der Kinematik zu kompensieren, nutzt Romeo ab einer gewissen Distanz zusätzlich PBVS um die Hand zum Greifen präziser positionieren zu können. Um die Posenschätzung der Hand zu erleichtern ist auf dieser ein QR-Code angebracht (erkennbar in Abbildung 3.7(b), untere Reihe, 1. Bild von links). Nach dem Greifen wird per Personendetektion ein Interaktionspartner gewählt und diesem, wiederum über eine Kombination der beiden Visual-Servoing-Varianten, das Objekt ausgehändigt (Abbildung 3.7(b), untere Reihe, 2. Bild von links). Bei der Implementierung greifen die Autoren auf die bereits vorgestellte ViSP-Toolbox zurück.

## Einordnung der Masterarbeit

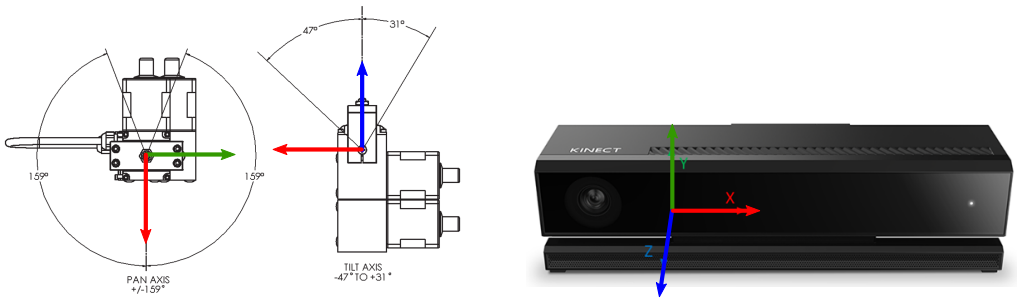
Das Verfahren, welches im folgenden Abschnitt erarbeitet wird gliedert sich in den State-of-the-Art als Variante des PBVS ein, da explizit eine Pose angesteuert wird. Generell besteht die Möglichkeit, sowohl ein eye-in-hand-Szenario, bei dem die auf der PTU montierte Kinect2 selbst die Zielpose vorgibt, als auch ein eye-to-hand-Szenario zu realisieren, bei dem die Zielpose durch eine externe Kamera ermittelt wird. Im Rahmen der Experimente wird die Nutzerhypothese des laser- und kamera-basierten Robotertrackings als Ziel gewählt. Es handelt sich also um eine klassische eye-to-hand-Situation.

## 3.4 Verfahren

Im Folgenden wird nun die Steuerstrategie der Pan-Tilt-Unit für den Einsatz im Rahmen des Gangtrainings beschrieben. Der hier beschriebene Ansatz realisiert dabei eine getrennte Steuerung von Schwenk- und Neigewinkel nach unterschiedlichen Kriterien, jeweils mit dem Ziel das Sichtfeld der montierten Kinect2-Kamera optimal auszunutzen.

### 3.4.1 Ausgangssituation

Nachdem der Hardware-Aufbau bereits zu Anfang dieses Kapitels (Seite 18) beschrieben wurde, folgen nun noch einige einleitende Details zur Integration in MIRA. Die in den theoretischen Grundlagen beschriebene Abbildung zwischen verschiedenen Koordinatensystem wird in MIRA durch eine *Transformation Framework* genannte Komponente realisiert. Intern nutzt das Transformation Framework einen *Transformation Tree*, der es erlaubt Beziehungen zwischen den verschiedenen Koordinatensystem des Roboters zu definieren. Die namensgebende Baumstruktur ergibt sich aus der Verknüpfung der *Knoten* genannten (lokalen) Koordinatensysteme mit einem übergeordneten Koordinatensystem. Jeder Knoten kann dabei wieder als Basis für beliebig viele weitere Koordinatensysteme genutzt werden. Wird nun eine Umrechnung zwischen zwei Bezugssystemen benötigt, übernimmt das Transformation Framework die notwendigen Operationen um eine Transformation zu berechnen, die diese Abbildung vollständig umsetzen kann. Da sich Beziehungen zwischen den Knoten auch mit der Zeit ändern können, werden auch die internen Kanten der Baumstruktur laufend aktualisiert. Zusätzlich wird auch die „Vergangenheit“ der Transformationen erfasst, so dass sich auch Abbildungen zu einem früheren Zeitpunkt ermitteln lassen. Vertiefende Implementierungsdetails finden sich in der Online-Referenzdokumentation des Transformation Framework [MIRA Reference Documentation, 2017].
Die Integration von PTU und Kinect2 erfolgt im Rahmen der Masterarbeit über eine Verkettung von drei Koordinatensystemen. Zunächst wird der sogenannte *PTUMount-Frame* relativ zum Roboterkoordinatensystem (*RobotFrame*) definiert. Die x-Achse des *RobotFrame* zeigt stets in Fahrtrichtung des Roboters, die y-Achse nach links




(a) lokales Koordinatensystem der PTU (b) lokales Koordinatensystem der Kinect2

**Abbildung 3.8:** lokale Koordinatensysteme von PTU und Kinect2
*(b) PTU-Koordindatensystem in Neutralstellung (nach [*FLIR Motion Control Systems, Inc*, 2013]). (b) Koordinatensystem der Kinect2-Tiefenkamera (nach [*Microsoft Corporation*, 2014a])*

(ausgehend von der Fahrtrichtung) und die z-Achse nach oben Richtung Kopf. Der *PTUMountFrame* beschreibt die Neutralstellung der PTU und ist relativ zum Basis-Koordinatensystem des Roboters im Wesentlichen um 180° um die Hochachse gedreht und etwa 1,2 m entlang der selben Achse nach oben verschoben. Durch den Anbau des Aluprofils ergibt sich außerdem eine Verschiebung von rund 30 cm in Richtung der negativen x-Achse des *RobotFrame*. Die aktuelle Ausrichtung der PTU wird durch die zuständige MIRA-Unit als *PTUFrame* relativ zum *PTUMountFrame* ausgeben. Der Unterschied zum *PTUMountFrame* liegt nur im Gier- und Nickwinkel, die den aktuellen Schwenk- und Neigewinkel der PTU abbilden. Zuletzt wird noch der *KinectFrame* mit dem *PTUFrame* verknüpft. Dies geschieht durch eine Verschiebung von 8 cm in z-, 2 cm in x- und 7 cm in y-Richtung. Zuletzt muss noch die Rotation durch Drehen um jeweils −90° um die y- und die x-Achse angepasst werden. Der Anordnung der einzelnen Koordinatensystem ist nochmals in Abbildung 3.9 veranschaulicht.

### 3.4.2 Vertikale Steuerung

Die Neigung soll genutzt werden um den Bereich zu maximieren, in dem die Person vollständig im vertikalen Sichtbereich der Kinect-Tiefenkamera erfasst werden kann. Dabei handelt es sich um eine grundlegende Voraussetzung um das Skelett zuverlässig

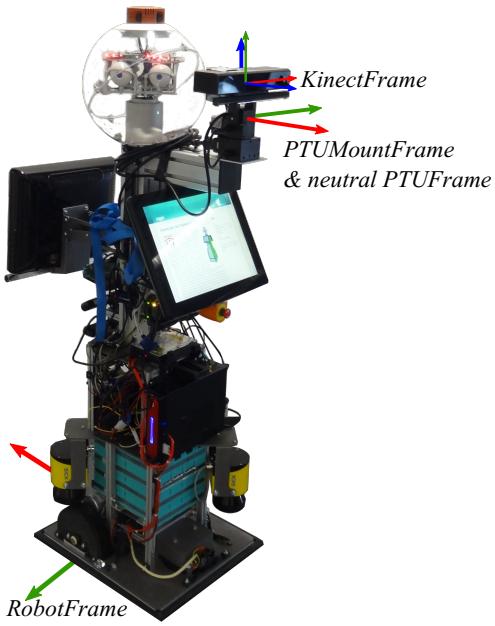



**Abbildung 3.9:** Veranschaulichung der Koordinatensysteme am Roboter
*Die nicht dargestellte z-Achse des RobotFrame, folgt der Säule an der unter anderem, Monitore, Kopf und PTU befestigt werden. Das Bild zeigt die PTU in Neutralstellung, daher stimmen PTUMountFrame und PTUFrame überein.*

schätzen zu können. Experimente im Rahmen einer parallel ablaufenden studentischen Arbeit [Jäschke, 2017] sowie eigene Vorexperimente legen nahe, dass das Verlassen des Bildbereichs vor allem am unteren Rand durch den im Kinect-SDK realisierten Skeletttracker nicht robust kompensiert werden können.

Um eine zuverlässige gangbasierte Wiedererkennung oder auch eine Ganganalyse mit dem mobilen Assistenzroboter zu ermöglichen, sollte der Bereich in dem der Patient vollständig vom Tiefensensor erfasst wird, maximiert werden. Dadurch führen Schwankungen im Abstand zwischen Patient und Roboter, die beispielsweise durch wechselnde Laufgeschwindigkeiten oder Navigationsmanöver entstehen, seltener zu Ausfällen des Skeletttrackings. Das Ziel dieses Abschnitts ist es daher ein Verfahren abzuleiten, dass einen optimalen Neigungswinkel ermittelt.

### Ansatz

Der optimale Neigungswinkel $\varphi$ ist dabei in erster Linie von drei Faktoren abhängig:

1. $h_m$ Montagehöhe der Kamera
2. $\gamma$ vertikaler Öffnungswinkel der Tiefenkamera
3. $h_p$ (maximale) Körpergröße der Patienten

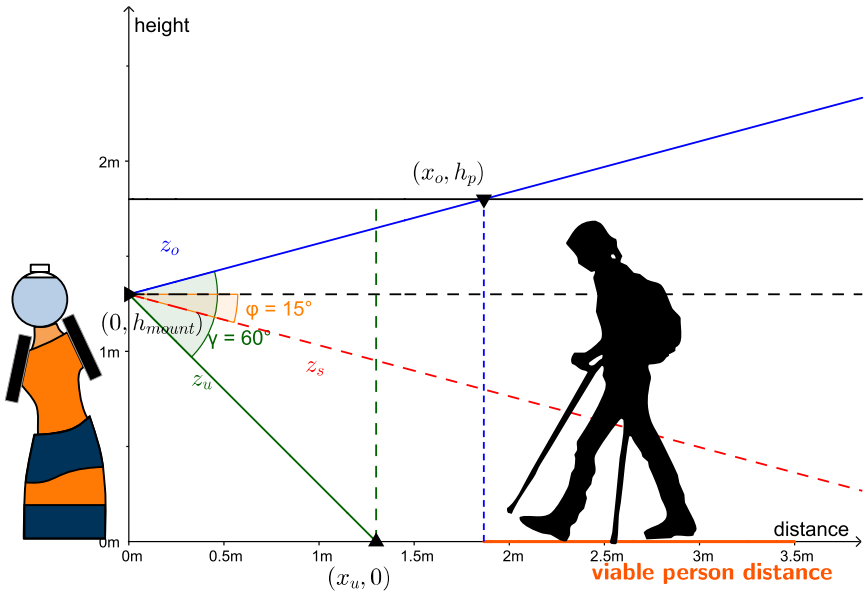



**Abbildung 3.10:** Geometrische Modellierung der PTU-Neigung
*Beispielhafte Modellierung eines Anwendungsszenarios mit einer Montagehöhe vom* 1,3 m*, einem vertikalen Öffnungswinkel* $\gamma = 60°$ *für Personen mit einer Körpergröße von bis zu* 1,8 m*. Zur besseren Darstellung wurde ein Neigungswinkel von* 15° *gewählt. Mit diesen Parametern lässt sich eine Person ab einer Entfernung von ungefähr* 1,9 m *vollständig im Tiefenbild erfassen.*

### Modellierung

Über geometrische Betrachtungen, wie sie beispielhaft in Abbildung 3.10 ausgeführt sind, lassen sich aus diesen Angaben Grenzen für die vollständige Sichtbarkeit ableiten.

Dazu wird die Sichtachse der Kamera, die um den Winkel $\varphi$ gegenüber der Geraden $z(x) = h_m$ ausgelenkt ist, als Gerade modelliert. Die Sichtachse bildet dabei den ersten Schenkel des Winkels, die Gerade $z(x)$ den zweiten.

$$z_s(x) = -\tan(\varphi)\, x + h_m \tag{3.2}$$

Analog wird die Modellierung für die obere und untere Begrenzung des Sensorbereichs $z_u(x)$ bzw. $z_o(x)$ durchgeführt. Daraus ergibt sich die folgende algebraische Beschreibung der beiden Geraden:

$$z_o(x) = -\tan\left(\varphi - \frac{\gamma}{2}\right) x + h_m \tag{3.3}$$

$$z_u(x) = -\tan\left(\varphi + \frac{\gamma}{2}\right) x + h_m \tag{3.4}$$

Betrachtet wird im Folgenden nun der Schnittpunkt des oberen Randes mit der festgelegten (maximalen) Körpergröße $z_c(x) = h_p$ und der des unteren Randes mit der Bodenebene $z_f(x) = 0$ für ein gegebenes $\varphi$.

$$x_o(\varphi) = \frac{h_m - h_p}{\tan\left(\varphi - \frac{\gamma}{2}\right)} \tag{3.5}$$

$$x_u(\varphi) = \frac{h_m}{\tan\left(\varphi + \frac{\gamma}{2}\right)} \tag{3.6}$$

Eine Person kann vollständig erfasst werden wenn sie weiter als

$$x_{min}(\varphi) = max(x_u(\varphi), x_o(\varphi)) \tag{3.7}$$

von der Kamera entfernt ist. Für $\varphi$ ergeben sich ferner bauartbedingte Grenzen $\varphi_{min}$ und $\varphi_{max}$. Zusätzlich liefert die Modellierung zwei Einschränkungen für die Schnittpunkte

$$x_o(\varphi) \geq 0 \tag{3.8}$$

$$x_u(\varphi) \geq 0 \tag{3.9}$$

### Optimierung

Angelehnt an die Prozessoptimierung lässt sich das Problem als Optimierungsproblem unter Steuer- und Zustandsbeschränkungen (vergleiche (3.8), (3.9)) beschreiben. Mit

dieser Herangehensweise lässt sich das folgende Optimierungskriterium formulieren

$$\min_{\varphi_{min} \leq \varphi \leq \varphi_{max}} \left\{ J = \max\left(x_u(\varphi), x_o(\varphi)\right) \right\} \tag{3.10}$$

Durch die Nichtlinearität und Periodizität der tan-Funktion ist eine geschlossene Lösung aufwendig. Jedoch lässt sich diese gut numerisch approximieren. Dazu empfiehlt es sich die Gleichung (3.10) als Minimierung einer Variablen zu formulieren:

$$\min_{\varphi_{min} \leq \varphi \leq \varphi_{max}} \{ J = |x_u(\varphi) - x_o(\varphi)| \} \tag{3.11}$$

Anschließend kann das Problem als numerisches Minimierungsproblem gelöst werden. Möglich ist dies beispielsweise über das BRENT-Verfahren [Brent, 2002], dass unter anderem über scipy[2] [Jones et al., 2001] in Python verfügbar ist. Außerdem existiert eine freie C++-Implementierung des Scientic Computing Departement der Florida State University[3].

### Beispielrechnung

Die hier durchgeführte Beispielrechnung basiert auf den Parameterwerten, die auch für Abbildung 3.10 ($h_m = 1{,}3\,\mathrm{m}, h_p = 1{,}8\,\mathrm{m}, \gamma = 60°$) zugrunde gelegt werden.
Für die Neutralstellung $\phi = 0°$ liefert die Gleichung (3.7) einen Wert von $x_{min}(0°) = 2{,}25$. Für das Beispiel aus Abbildung 3.10 ergibt sich $x_{min}(15°) = 1{,}87$.
Die Optimierung des Zielfunktionals (3.11) nach Brent liefert einen optimalen Neigungswinkel von $\varphi_{opt} \approx 11{,}3°$. Einsetzen in Gleichung (3.7) ergibt eine minimale Distanz von $x_{min}(11{,}3°) = 1{,}48$. Im Vergleich zu $\phi = 0°$ konnte der Bereich also um rund 77 cm vergrößert werden. Bezüglich $\varphi = 15°$ beträgt die Verbesserung noch 39 cm.

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize_scalar.html`, Stand: 26.04.2017

[3] Verfügbar unter `http://people.sc.fsu.edu/~jburkardt/cpp_src/brent/brent.html`, Stand: 26.04.2017

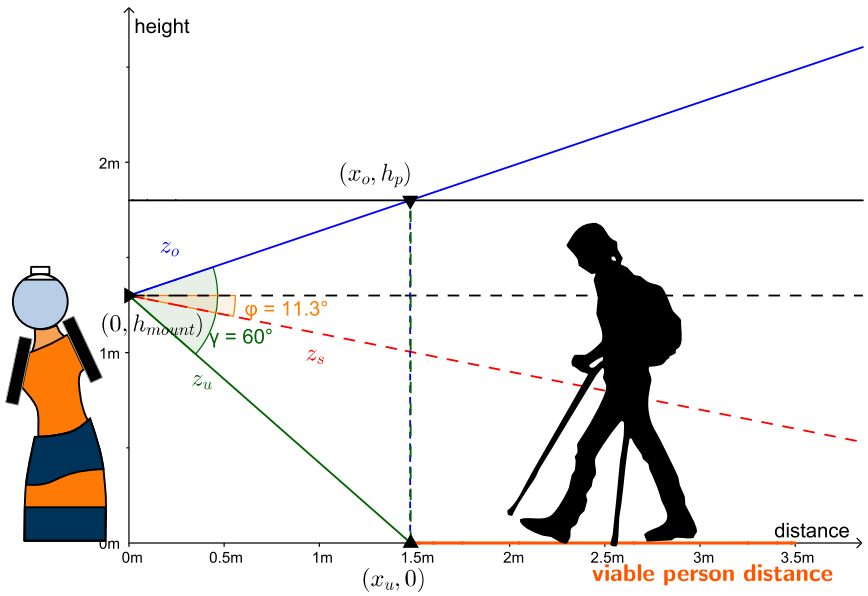


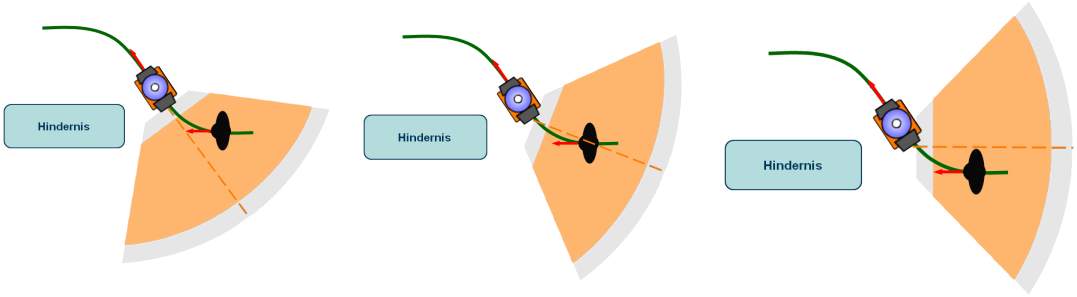

**Abbildung 3.11:** Optimaler Neigungswinkel

*Wie angenommen ist zu erkennen, dass im Optimalfall die beiden Schnittpunktkoordinaten $x_o$ und $x_u$ übereinstimmen. Außerdem ist der orange markierte Bereich in dem die Person vollständig im Bild erfasst wird, erkennbar größer als noch in Abbildung 3.10.*

### 3.4.3 Horizontale Steuerung

Nachdem zuvor der Entfernungsbereich optimiert wurde, in dem eine Person vollständig vom Sensor erfasst werden kann, liegt nun der Fokus auf Position und Orientierung im Bild. Ähnlich wie bereits zu Anfang der vorherigen Abschnitts beschrieben, nimmt die Qualität der Skelettschätzung auch am rechten beziehungsweise linken Rand des Sensorbereichs ab. Ein weiteres Kriterium, das Einfluss auf die Qualität der Skelettschätzung hat, ist die Orientierung der Person im Bezug zur Kinect2. Ideale Voraussetzungen bietet eine Frontalansicht, da hierbei keinerlei Eigenverdeckungen auftreten können. Daraus ergeben sich konkurrierende Zielstellungen für die Ansteuerung der

PTU. Wie in den beiden rechten Graphiken der Abbildung 3.12 zu erkennen ist, lassen sich eine zentrale Position im Sensorbereich und eine Frontalansicht nicht gleichzeitig realisieren, wenn der Nutzer nicht ohne horizontalen Versatz zur Fahrtrichtung des Roboter laufen kann. Da es sich bei dem im Projekt ROGER eingesetzten Roboter Ringo nicht um eine holonome Plattform handelt, lassen sich derartige Situationen vor allem bei Ausweichmanövern nicht vermeiden.




(a) starre Kamera (b) Nutzer zentriert (c) Nutzer frontal

**Abbildung 3.12:** Darstellung verschiedener Ansteuerungen
*Vergleich der Positionierung im Sensorbereich bei (a) starrer Kamera, (b) Zentrierung des Nutzers und (c) realisieren einer möglichst frontalen Ansicht .*

Im Folgenden werden daher nun zwei separate Ansätze beschrieben, mit denen jeweils eine Zielstellung erreicht werden kann. Anschließend erfolgt eine Erweiterung der Ansteuerung um eine Geschwindigkeitsregelung und eine prädiktive Komponente um den Anforderungen durch die Eigendynamik des Roboters besser begegnen zu können. Abschließend wird die Gesamtanwendung erläutert, in der die Subkomponenten zusammengeführt und an die Roboterapplikation angebunden werden.

### Zentrierte Ansicht

Eine statische Personenhypothese zentriert im Sensorsichtfeld zu positionieren lässt sich auf Basis der definierten Koordinatensysteme leicht realisieren. Dazu muss das Ziel $\underline{\mathbf{p}}^{target}_{PoseFrame}$ aus dem nicht näher bestimmten Koordinatensystem *PoseFrame* in den *PTUMountFrame* transformiert werden. Anschließend lässt sich der Pan-Winkel

$\vartheta$ durch Anwendung der Arkustangens-Funktion mit zwei Argumenten bestimmen (Definition siehe Anhang A.1, Seite 89). Da in der Praxis jedoch technische Rahmenbedingungen mit berücksichtigt werden müssen, lassen sich nicht alle Beobachtungswinkel realisieren. Zum einen sind die Grenzen der PTU mit $\pm 159°$ zu berücksichtigen, zum anderen muss die Verdeckung des Sichtbereichs durch den Kopf des Roboters bedacht werden. Für den im Rahmen der Masterarbeit realisierten Aufbau und das erwartete Einsatzszenario, ist eine Einschränkung auf den Bereich $\pm 90°$ sinnvoll. Als Grenzwerte für den Pan-Winkel ergibt sich daraus die symmetrische Definition von $\vartheta_{max} = -\vartheta_{min} = \frac{\pi}{2} = 90°$. Eine halbformale Beschreibung des Vorgehens ist nachfolgend in Algorithmus 3.13 dargestellt.

---

**Eingaben**

1 $\underline{\mathbf{p}}^{target}_{PoseFrame}$ *// Personenhypothese*

2 $\vartheta_{min}, \vartheta_{max}$ *// untere/obere Grenze der realisierbaren Winkel*

**Algorithmus**

**Koordinatentransformation:** ***// über Transformation Framework***

3 $\underline{\mathbf{T}}_{PTUMountFrame \leftarrow PoseFrame} \leftarrow$ `getTransform`$(PoseFrame, PTUMountFrame)$;

4 $\underline{\mathbf{p}}^{target}_{PTUMountFrame} \leftarrow \underline{\mathbf{T}}_{PTUMountFrame \leftarrow PoseFrame} \circ \underline{\mathbf{p}}^{target}_{PoseFrame}$;

**Winkelberechnung:**

5 $\vartheta \leftarrow \arctan2\left(\underline{\mathbf{p}}^{target}_{PTUMountFrame}.y(), \underline{\mathbf{p}}^{target}_{PTUMountFrame}.x()\right)$;

6 `wenn` $\vartheta < \vartheta_{min}$;

7 $\quad \vartheta \leftarrow \vartheta_{min}$;

8 `wenn` $\vartheta > \vartheta_{max}$;

9 $\quad \vartheta \leftarrow \vartheta_{max}$;

**Rückgabe**

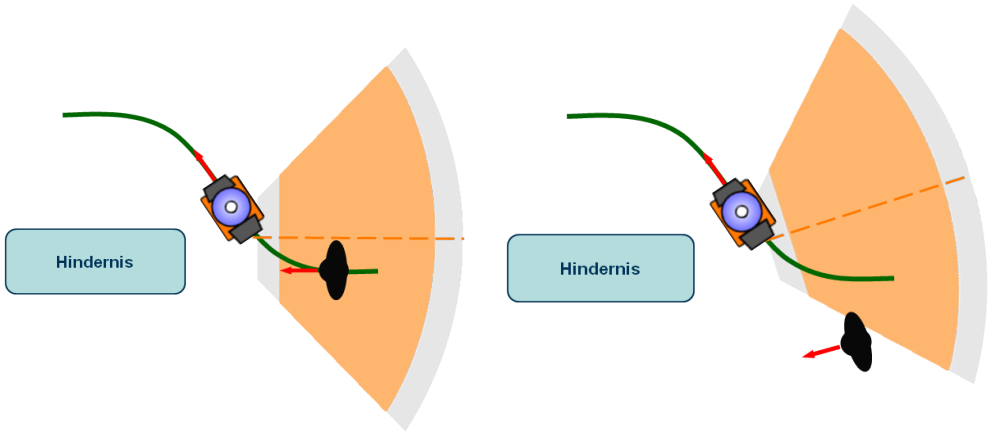
10 `Pan-Winkel` $\vartheta$

---

**Algorithmus 3.13:** Berechnung des Schwenk-Winkels - frontale Ansicht

### Frontale Ansicht

Unter der Voraussetzung, dass die Oberkörperorientierung der Person bekannt ist, lässt sich die Ansteuerung für eine möglichst frontale Ansicht ebenfalls leicht berechnen. Jedoch müssen dabei einige Besonderheiten beachtet werden, da diese nicht in allen Situationen realisierbar ist. Ein Beispiel dafür ist in Abbildung 3.14(b) dargestellt.




(a) gutartige Situation (b) problematische Situation

**Abbildung 3.14:** Grenzen für die frontale Ansicht
*Im Gegensatz zur Situation in der linken Graphik, kann im rechts dargestellten Szenario keine Frontalansicht realisiert werden.*

Daher muss im Algorithmus zur Berechnung der Ansteuerung eine Rückfallstrategie vorgesehen werden, die in derartigen Situationen Anwendung findet. In der formalen Beschreibung der Ansteuerung für eine möglichst frontale Ansicht in Algorithmus 3.15 ist für diesen Fall die zentrierte Ansicht aus 3.13 vorgesehen. Die Wahl dieser Variante geschieht dabei mit dem Hintergedanken, eine möglichst lückenlose Beobachtung des Nutzers zu realisieren. Dies ist vor allem dann von Bedeutung, wenn die Zielpose direkt aus den Daten der auf der PTU verbauten Kamera generiert werden soll. Allerdings sind auch andere Behandlungsmöglichkeiten, wie etwa das Anfahren der Neutralstellung, Beibehalten in der letzten Position oder Feedback an die Navigation möglich.

**Eingaben**

1 $\underline{\mathbf{p}}^{target}_{PoseFrame}$ // *Personenhypothese*

2 `Pan-Winkel` $\vartheta$ // *aus Algorithmus 3.13*

**Initialisierung**

3 $\Psi_{max} \leftarrow \frac{\pi}{2}$ ; // *Grenzwert, entspricht* $\pm 90°$

4 $\Delta\vartheta_{max} \leftarrow \frac{25°}{180°} \cdot \pi$ ; // *halber (nutzbarer) horiz. Sensoröffnungswinkel*

**Algorithmus**

**Koordinatentransformation:** ***// über Transformation Framework***

5 $\underline{\mathbf{T}}_{PTUMountFrame \leftarrow PoseFrame} \leftarrow$ `getTransform`$(PoseFrame, PTUMountFrame)$;

6 $\underline{\mathbf{p}}^{target}_{PTUMountFrame} \leftarrow \underline{\mathbf{T}}_{PTUMountFrame \leftarrow PoseFrame} \circ \underline{\mathbf{p}}^{target}_{PoseFrame}$;

**Winkelberechnung:**

7 $\Psi \leftarrow \underline{\mathbf{p}}^{target}_{PTUMountFrame}.yaw()$; // $\Psi \in (-\pi, \pi]$

8 $\vartheta_f \leftarrow \Psi - \operatorname{sgn}(\Psi) \cdot \pi$;

/* *Überprüfe Realisierbarkeit der Frontalansicht* */

9 `wenn` $|\vartheta_f| \leq \Psi_{max}$ `und` $|\vartheta_f - \vartheta| \leq \Delta\vartheta_{max}$;

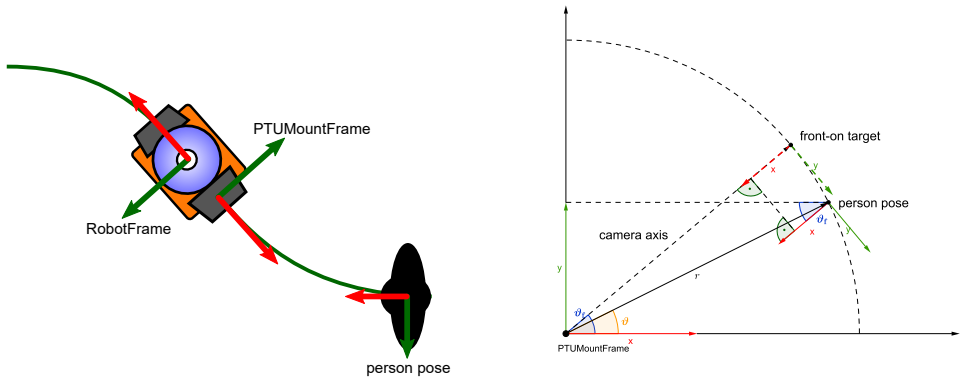
10 $\vartheta \leftarrow \vartheta_f$;

**Rückgabe**

11 `Pan-Winkel` $\vartheta$

**Algorithmus 3.15:** Berechnung des Schwenk-Winkels - frontale Ansicht

Basierend auf dem berechneten Schwenk-Winkel kann eine *virtuelle Zielpose* für die Kamera definiert werden. Die Ansteuerung dieser virtuellen Zielpose mit Algorithmus 3.13 für die zentrierte Erfassung von Personen, würde ebenfalls $\vartheta_f$ liefern. Eine Veranschaulichung der virtuellen Zielpose sowie ein geometrischer Nachweis für die Frontalansicht findet sich in Abbildung 3.16(b).




(a) Ausgangssituation (b) Berechnung im *PTUMountFrame*

**Abbildung 3.16:** Veranschaulichung zur Berechnung der Frontalansicht
*Geometrische Veranschaulichung des Ansatzes zum ermitteln des Steuerwinkels für Frontalansicht. Ausgehend von den Berechnungen ist die Ermittlung einer „virtuellen Pose“ möglich, die als Eingabe für Algorithmus 3.13 zum gleichen Schwenk-Winkel führt.*

### Geschwindigkeitsregelung

Anhand der bisher beschriebenen Algorithmen lässt sich der Winkel bestimmen der angesteuert werden muss, um die Kamera entsprechend der einzelnen Kriterien zu positionieren. Dabei wurde bisher aber die Dynamik des Systems vernachlässigt, die sich aus Änderungen der Roboter-Nutzer-Konstellation durch Bewegung ergibt. Um dem Rechnung zu tragen, wird die Ansteuerung um eine zweite, nachgelagerte Komponente zur Geschwindigkeitsregelung ergänzt.

Das im Rahmen dieser Masterarbeit realisierte Verfahren stützt sich dabei im Kern auf einen diskreten PID-Regler. Es handelt sich daher in erster Linie um einen reaktiven Ansatz, der eine definierte Regeldifferenz minimiert. Als Regelabweichung sei die Winkeldifferenz zwischen momentanem Schwenk- und dem Zielwinkel $\vartheta$ definiert.

Als Ausgabe erzeugt der Regler zu jedem Zeitpunkt eine Winkelgeschwindigkeit $v_\vartheta$, die an den internen Controller der PTU weitergeleitet und dort in Hardware-Kommandos umgesetzt wird. Um den Grenzen der PTU-Hardware Rechnung zu tragen, erfolgt

die Realisierung des PID-Reglers mit Beschränkung der Steuergröße ($|v_\vartheta| \leq v_{\vartheta,max}$). Diese Einschränkung kann, vor allem in der hier verwendeten diskreten Realisierung, zu Problemen durch eine nur langsam abnehmende Ausgabe des Integriergliedes führen [Åström und Hägglund, 1989]. Da die PTU jedoch näherungsweise selbst als Integrator modelliert werden kann, wird in der Realisierung des Reglers auf den I-Anteil verzichtet (Blockschaltbild in Abbildung 3.17). Es existiert jedoch auch eine Reihe von Algorithmen um diese Problematik von Seiten der Implementierung zu begegnen. Dazu sei an dieser Stelle auf weiterführende Literatur wie etwa [Åström und Hägglund, 1989] oder [Åström und Rundqwist, 2006] verwiesen.

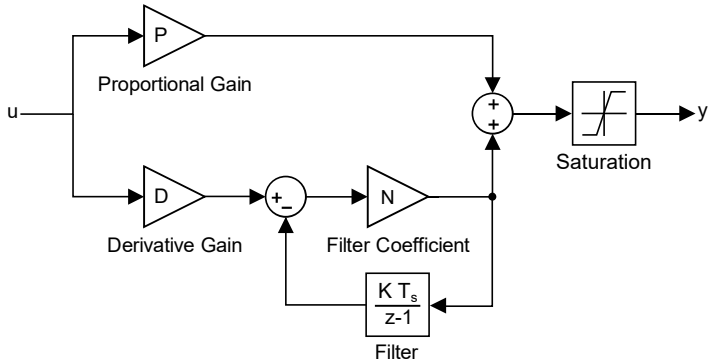



**Abbildung 3.17:** erweiterter PD-Regler - Blockschaltbild
*Als Eingabe u erhält der Regelkreis die absolute Differenz zwischen aktuellem Winkel und Zielwinkel. Als Ausgabe y wird die Geschwindigkeit für den nächsten Zeitschritt erzeugt. Das Blockschaltbild zeigt eine numerisch stabilere Variante des Differentialreglers.*

Die Bestimmung der Regler-Parameter $P$, $D$ und $N$ erfolgt basierend auf Daten aus einigen Vorversuchen. Dazu werden mögliche Fahrmanöver mit dem Roboter getestet und die relative Position des Nutzers erfasst. Anschließend werden die aufgezeichneten Daten in MATLAB/Simulink analysiert und die Parameter mit dem integrierten *PID Tuner* [MathWorks, 2017] iterativ angepasst. Der Entwurf und die Realisierung erfolgt dabei für eine Abtastzeit von 100 ms, da Raten jenseits von circa 15 Hz in entsprechenden Vorversuchen vermehrt zu Problemen mit der PTU-Controller-Hardware

geführt haben. Für eine *Response Time* von 250 ms und eine *Robustness* von 0,65 wurden die Parameter wie folgt bestimmt:

- $P = 6{,}459$
- $D = 2{,}652$
- $N = 8{,}528$

Nach einigen Tests auf dem Roboter wurden die Parameter wie folgt angepasst, um eine weniger abrupte Ansteuerung zu erreichen:

- $P = 6{,}459$
- $D = 1{,}5$
- $N = 12{,}5$

Neben einem ruhigeren Verlauf der Bewegung, führen diese Änderungen auch auf lange Sicht zu einer geringeren mechanischen Belastung der Motoren. Grund hierfür ist die langsamere Variation der Geschwindigkeit sodass in einem geringeren Umfang gegen das Trägheitsmoment der auf der PTU montierten Kinect2-Kamera gesteuert werden muss.

### Integration von Prädiktion

Die beschriebene Berechnung des Ansteuerwinkels und die Regelung der Geschwindigkeit erfolgt bisher rein reaktiv. Da der Roboter jedoch seinen Pfad für einen gewissen Zeithorizont unter Einbeziehung lokaler Faktoren wie Hindernissen relativ genau prädizieren kann, bietet es sich an diese Information zu nutzen und in die Berechnungen einfließen zu lassen. Dadurch ergeben sich vor allem Vorteile bei starken Lenkbewegungen. Um diesen Vorteil ausnutzen zu können, ist kein großer Planungshorizont notwendig. Das auf Ringo eingesetzte Verfahren zur lokalen Navigation verfügt beispielsweise über Informationen zum wahrscheinlichen Pfad in den nächsten 250 ms. Unter Einbeziehung von Kenntnissen über die Kinematik des Roboters, die über entsprechende Schnittstellen durch das MIRA-Framework bereit gestellt werden, ist so eine Vorausberechnung für die Roboter-Nutzer-Konstellation möglich. Bildet man diese auf eine Pose ab (siehe Algorithmus 3.18), lassen sich alle bisher beschriebenen Algorithmen analog anwenden. Die Nutzung der Information aus der Prädiktion erfolgt dabei implizit.

**Eingaben**

1 $\underline{\mathbf{p}}^{target}_{PoseFrame}$ // *Personenhypothese*

2 `trajectory` // *Robotertrajektorie innerhalb des Planungshorizonts*

**Initialisierung**

3 `robotModel` $\leftarrow$ getRobotModel() ; // *Informationen über die Kinematik des Roboters*

**Algorithmus**

**Prädiktion:**

/* *Berechne „Pseudotransformation“ durch Fahrt entlang der Trajektorie* */

4 $\underline{\mathbf{T}}^{prediction}_{RobotFrame \leftarrow RobotFrame} \leftarrow$ `robotModel.computeTransform(trajectory);`

**Anwendung:**

5 $\underline{\mathbf{p}}^{target}_{RobotFrame} \leftarrow \texttt{getTransform}(PoseFrame, RobotFrame) \circ \underline{\mathbf{p}}^{target}_{PoseFrame}$;

/* *Anwendung der „Pseudotransformation“* */

6 $\underline{\mathbf{p}}^{target}_{RobotFrame} \leftarrow \underline{\mathbf{T}}^{prediction}_{RobotFrame \leftarrow RobotFrame} \circ \underline{\mathbf{p}}^{target}_{RobotFrame}$;

7 $\underline{\mathbf{p}}^{target}_{PoseFrame} \leftarrow \texttt{getTransform}(RobotFrame, PoseFrame) \circ \underline{\mathbf{p}}^{target}_{RobotFrame}$;

**Rückgabe**

8 $\underline{\mathbf{p}}^{target}_{PoseFrame}$ // *prädizierte Lage der Personenhypothese*

**Algorithmus 3.18:** Einbeziehung der Robotertrajektorie in die Ansteuerung

Darüber hinausgehend besteht auch die Möglichkeit, Prädiktionsalgorithmen für die Trajektorie des Nutzers (beispielsweise lineare Prädiktion oder Kalman-Filter) einzusetzen. Dieser Ansatz wird im Rahmen der Masterarbeit allerdings nicht weiter betrachtet und bietet Anknüpfungspunkte für weiterführende Untersuchungen.

## 3.5 Experimente

Die Experimente zur Kamerasteuerung (*Keep-in-View*, kurz: *KIV*) werden im Rahmen der Masterarbeit im Zuse-Bau der Fakultät für Informatik an der TU Ilmenau durch-

geführt. Dazu wurden zwei Wegpunkte definiert, zwischen denen der Roboter autonom navigiert. Um eine anspruchsvollere Testumgebung zu erstellen, wird die Strecke um zusätzliche Hindernisse ergänzt, anhand derer typische Fahrsituationen im Klinikbetrieb nachvollzogen werden können.

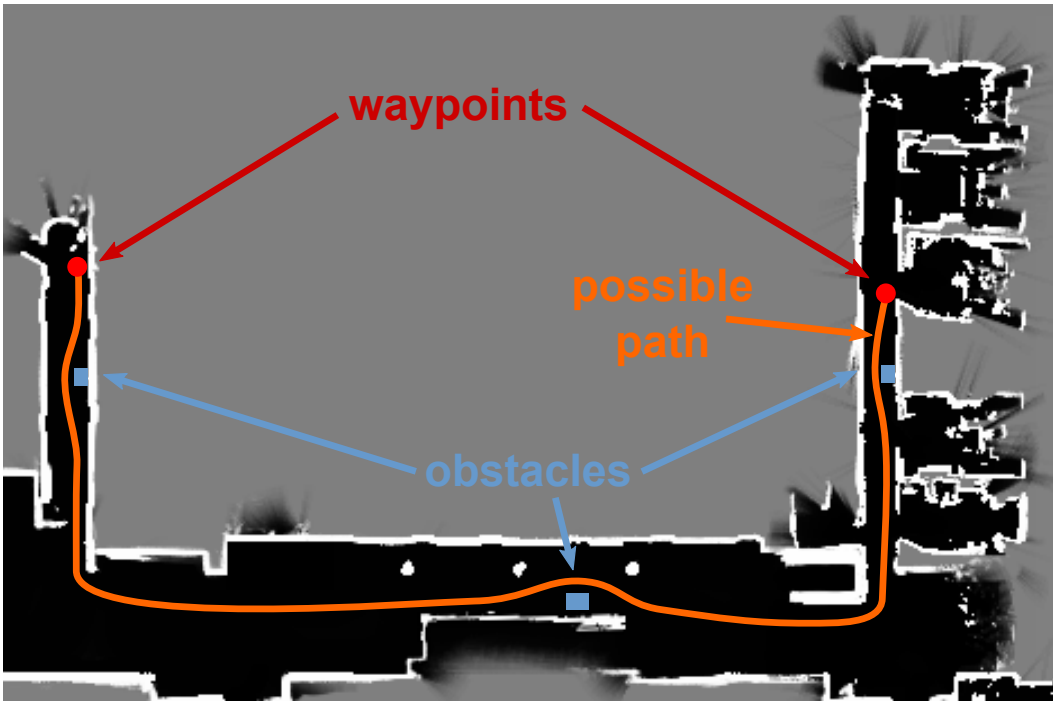



**Abbildung 3.19:** grober Aufbau der Teststrecke
*Kartenausschnitt Zuse-Bau (2. Obergeschoss) mit Markierungen für die Wegpunkte (rot), den zusätzlichen Hindernissen (hellblau) und einem möglichen Pfad (orange).*

Als Probanden wurden mehrere Mitarbeiter des Fachgebiets Neuroinformatik und Kognitive Robotik nacheinander gebeten dem Roboter zu Folgen. Als Zielpose wurde die Hypothese der Personentrackers mit Unterstützung der Wiedererkennung [Wengefeld et al., 2016] gewählt. Für die frontale Keep-in-View-Ansteuerung wird eine Schätzung der Oberkörperorientierung ausgehend von der Bewegungsrichtung der Person genutzt, da der Personentracker zur Laufzeit keine explizite Schätzung vornimmt. Testfahrten erfolgen jeweils mit deaktivierter Kamerasteuerung (*noKIV*) sowie mit zentrierter (*cKIV*) und frontaler Ansicht (*fKIV*). Für die Ansteuerung der PTU wird jeweils auch auf die Trajektorien-basierte Prädiktion der Roboterbewegung zurück-

gegriffen. Die Auswertung im Folgenden basiert auf 30 Fahrten (Hin- und Rückweg werden separat gezählt) zwischen den beiden Wegpunkten. Der Großteil der Fahrten (26) wurde dabei im Lotsen-Modus durchgeführt, in dem der Roboter den Abstand zum Nutzer selbstständig regelt. Zu Referenzzwecken wurden noch vier Zielanfahrten durchgeführt, während derer der Roboter mit langsamer Geschwindigkeit die Wegpunkte anfährt und der Nutzer selbstständig in möglichst gleichmäßigen Abstand folgt. Die Auswertung im Folgenden konzentriert sich dabei im Wesentlichen auf die Aspekte der Position und der Orientierung der Person im Sensorsichtfeld, da es sich hierbei um den Kern der Masterarbeit handelt. Eine vorläufige, kurze Auswertung der Beobachtungen zum Kinect-SDK-Skeletttracking auf der mobilen Plattform findet sich dennoch der Vollständigkeit halber im Anhang ab Seite 90.

### 3.5.1 Position im Sensorerfassungsbereich

Im Rahmen der Experimente wird laufend die Position der Zielpose im Bezug zur Kamera erfasst. Die Auswertung erfolgt hier in erster Linie unabhängig von der Distanz zur Kamera, da die Ansteuerung der PTU keinerlei Einfluss darauf nehmen kann. Dazu wird die kartesische Position in Kamerakoordinaten in Polarkoordinaten transformiert und anschließend der Winkel ausgewertet (Veranschaulichung siehe Abbildung 3.20).

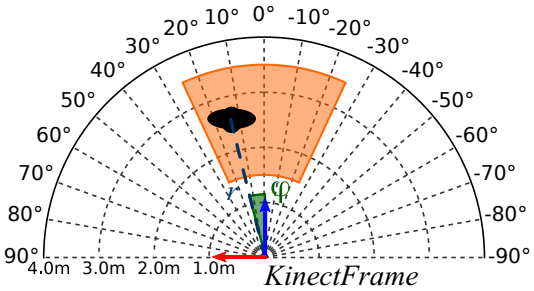
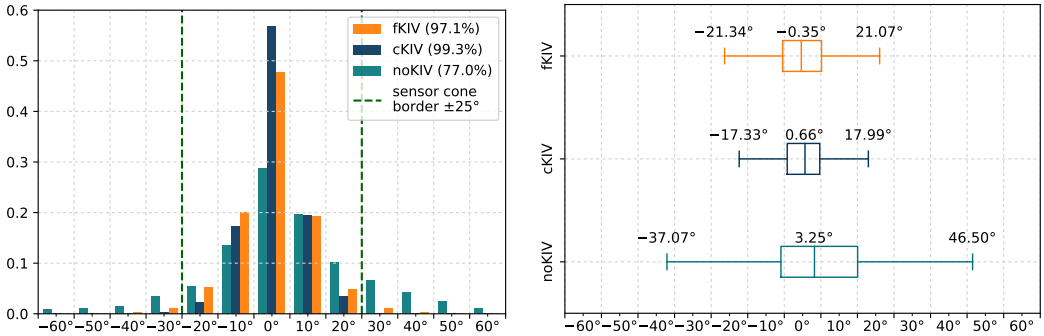



**Abbildung 3.20:** Auswertungsschema der Position im Sensorbereich
*Der Zielsensorbereich ist orange markiert und erstreckt sich in der Entfernung von* 1,5 m *bis* 3,5 m*, bei einer Sektorweite von* 50° (±25°).

Die Verteilung der dabei berechneten Winkel ist in Abbildung 3.21(a) graphisch als Histogramm dargestellt.




(a) Histogrammdarstellung (b) Box-Plot mit Quantilanalyse

**Abbildung 3.21:** graphische Auswertung der Position im Sensorerfassungsbereich *(a) relative Häufigkeit je Winkel und Verfahren. Die Zahl in Klammern gibt an, welcher Anteil der Datenpunkte zwischen den dunkelgrün markierten Grenzen des Sensorbereichs liegt. (b) Box-Plot der Verteilungen ohne Ausreißer (Erläuterung siehe Anhang A.3, Seite 92).*

Ein auf der Kurtosis basierender Test nach [Anscombe und Glynn, 1983] bestätigt für alle drei Stichproben die Annahme einer unimodalen Normalverteilung. Die vergleichende Bewertung der Verfahren anhand von Mittelwert und Standardabweichung nach Tabelle 3.1 beziehungsweise graphische über den Box-Plot in Abbildung 3.21(b) ist somit valide.

| Verfahren | Mittelwert $\mu$ | Standardabweichung $\sigma$ | Anteil im Sensorbereich |
|---|---|---|---|
| noKIV | 4,417° | 21,694° | 77,5% |
| cKIV | 0,629° | 7,802° | 99,3% |
| fKIV | −0,187° | 10,224° | 97,1% |

**Tabelle 3.1:** statistische Kennwerte der Position je Verfahren
*Die Tabelle enthält den empirischen Mittelwert, die Standardabweichung sowie den Anteil der Datenpunkte im ±25°-Bereich für alle drei getesteten Varianten.*

Wie erwartet weisen beide Keep-in-View-Verfahren eine deutlich geringere Streuung bezüglich der Position auf als der starre Aufbau. Der statistische Test auf gleiche Va-

rianz nach [Levene, 1960] zeigt signifikante Unterschiede der Streuungen für einen Signifikanzwert $p < 0{,}01$. Gleiches gilt für den Varianzunterschied zwischen beiden Keep-in-View-Verfahren (ebenfalls für $p < 0{,}01$). Dabei ist die Streuung für *cKIV* erwartungsgemäß geringer als bei der Ansteuerung auf Frontalansicht, da diese eine zentrale Position explizit zum Ziel hat. Dadurch wird auch der Sensorbereich bei starken Lenkmanövern seltener verlassen. Der Unterschied zwischen den beiden Keep-in-View-Varianten beträgt dabei rund 2-Prozentpunkte zugunsten der zentrierten Ansicht. Beide Verfahren können die Sensorabdeckung jedoch mit rund 20- beziehungsweise 22-Prozentpunkten gegenüber einem starren Kameraaufbau deutlich steigern. Das entspricht einer Steigerung von 25,97% beziehungsweise 28,57% im Vergleich zur starren Kamera.

Für die Mittelwerte lassen sich etwa über den t-Test in der Variante nach [Welch, 1947][4] ebenfalls signifikante Unterschiede feststellen. Allerdings lässt sich daraus keine inhaltliche Aussage über die Güte der jeweiligen Verfahren ableiten.

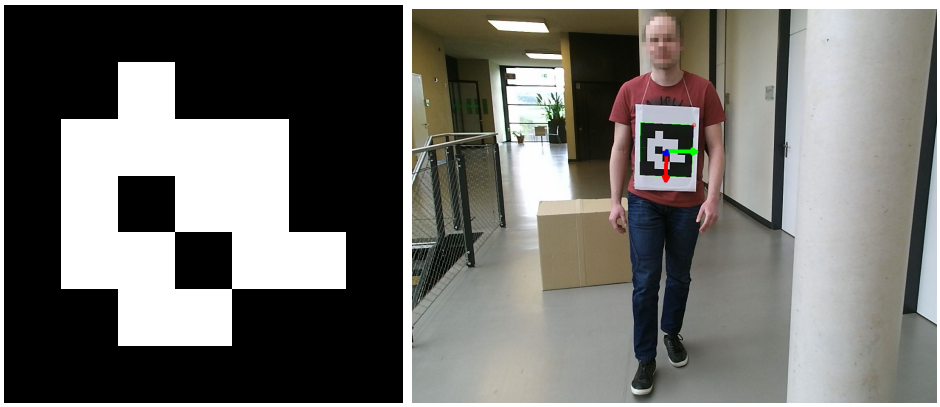
### 3.5.2 Oberkörperorientierung

Zusätzlich zur Position der Person im Sensorbereich wurde im Rahmen der Experimente die Oberkörperorientierung erfasst. Dazu wurden sogenannte ArUco-Marker [Garrido-Jurado et al., 2014] eingesetzt. Dabei handelt es sich um visuell detektierbare Marker, die üblicherweise im Augmented-Reality-Bereich eingesetzt werden (Beispiel siehe Abbildung 3.22).

Um die Marker zu erzeugen und detektieren zu können, wurde die Implementierung aus der OpenCV3-Bibliothek genutzt und an das MIRA-Framework angebunden. Die verwendete Implementierung erlaubt anschließend an die Detektion eine Schätzung der Pose im Kamerakoordinatensystem. Dadurch lässt sich die Oberkörperpose während der Aufnahmen ermitteln und anschließend auswerten.

Detektion und Posenschätzung erfolgen ausgehend vom Full-HD-Bild ($1920 \times 1080$ Pixel) der Kinect2-Farbkamera mit einer Frequenz von 8 Hz bis maximal 10 Hz.

[4] für Stichproben mit unterschiedlichem Umfang und Streuung

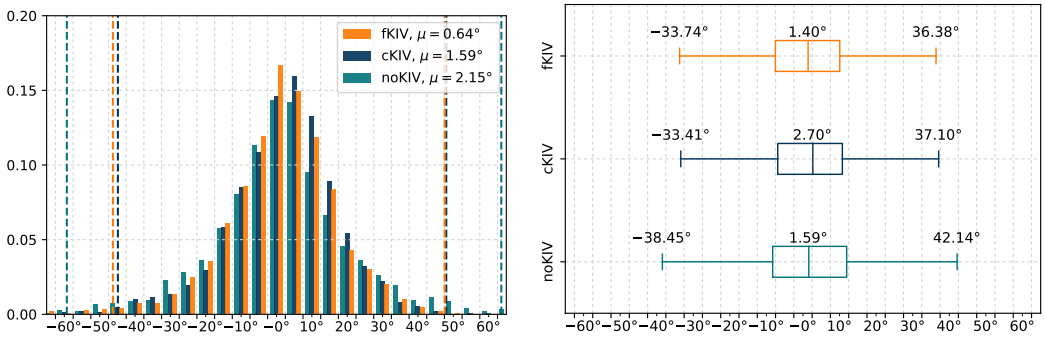


(a) Eingesetzter ArUco-Marker

(b) Beispiel einer Detektion

**Abbildung 3.22:** AR-Marker zur Schätzung der Oberkörperorientierung
*Um die Detektion auch aus größerer Entfernung und bei bewegungsbedingter Unschärfe sicherzustellen, wurde der Marker mit einer Kantenlänge von* 25 cm *ausgedruckt. Im rechten Bild ist eine Beispieldetektion und die aus der Posenschätzung ermittelten Koordinatenachsen der Pose dargestellt.*

Die relative Häufigkeit der Oberkörperorientierung im Bereich von $\pm 90°$ ist in Abbildung 3.23(a) als Histogramm dargestellt. Um die visuelle Bewertbarkeit zu verbessern, ist die x-Achse auf den Bereich $\pm 60°$ eingeschränkt. Da sich jedoch für alle drei Szenarien mehr als 99% ($\mu \pm 3\sigma$) der Daten innerhalb dieses Bereichs befinden, ist dadurch kein großer Informationsverlust zu erwarten. Zusätzlich gestützt wird diese Annahme auch durch die abnehmende Detektierbarkeit der Marker ab Verdrehungen von mehr als $\pm 80°$. Schließlich bleibt noch zu bemerken, dass Verdrehungen in dieser Größenordnung meist nicht mehr allein durch eine geeignete Ansteuerung der schwenkbaren Kamera kompensiert werden.
Der Kurtosis-Test bestätigt für alle drei Experimente eine unimodale Normalverteilung der Daten. Damit lässt sich eine Bewertung der Daten über einen Box-Plot wie in Abbildung 3.23(b) sowie die Auswertung von Mittelwert und Varianz basierend auf Tabelle 3.2 formal rechtfertigen.




(a) Histogrammdarstellung (b) Box-Plot mit Quantilanalyse

**Abbildung 3.23:** graphische Auswertung der Oberkörperorientierung
*(a) relative Häufigkeit der verschiedenen Oberkörperorientierungswinkel. Die Mittelwerte sind in der Legende des Histogramms angegeben, unterbrochene Linien markieren die $\mu \pm 3\sigma$-Grenzen der angenommenen Normalverteilung. (b) Box-Plot der Verteilungen ohne Ausreißer (Erläuterung siehe Anhang A.3, Seite 92).*

| Verfahren | Mittelwert $\mu$ | Standardabweichung $\sigma$ |
|---|---|---|
| noKIV | 2,149° | 20,478° |
| cKIV | 1,586° | 15,478° |
| fKIV | 0,645° | 15,635° |

**Tabelle 3.2:** statistische Kennwerte der Oberkörperorientierung je Verfahren
*Die Tabelle enthält den empirischen Mittelwert und die Standardabweichung für alle drei getesteten Varianten.*

Mit dem t-Test nach Welch lassen sich für den Vergleich zwischen starrer Kamera (*noKIV*) und Frontalansicht (*fKIV*) sowie für zentrierter Ansicht (*cKIV*) gegenüber Frontalansicht signifikante Unterschiede für den Mittelwert nachweisen (Signifikanzwert $p < 0{,}05$). Verschärft man den Signifikanzwert auf $p < 0{,}01$, kann kein signifikanter Unterschied zwischen den beiden PTU-Ansteuervarianten nachgewiesen werden. Ähnlich wie bereits bei der Auswertung der Position im vorherigen Abschnitt, besitzen Unterschiede in dieser Größenordnung jedoch kaum Praxisrelevanz.

Anhand des statistischen Tests nach [Levene, 1960] lässt sich ebenfalls mit $p < 0{,}05$ eine signifikant kleinere Streuung der Oberkörperorientierung für beide Ansteuerungsverfahren ($cKIV$/ $fKIV$) im Vergleich zum starren Kameraaufbau ($noKIV$) nachweisen. Der Nachweis gelingt jedoch nicht beim Vergleich der beiden Verfahren untereinander. Die für die Streuung getroffenen Aussagen halten auch einer Verschärfung des Signifikanzwertes auf $p < 0{,}01$ stand.

## 3.6 Zusammenfassung und Ausblick

Im vorangegangen Kapitel wurde der Einsatz der Schwenk-Neige-Einheit motiviert und ihre Spezifikation sowie die Montage auf der Roboterplattform Ringo dargestellt. Nach dem umfangreichen Grundlagenteil und einem kurzen Einblick in einen engen Teilbereich des State-of-the-Art, erfolgte die Vorstellung des im Rahmen der Masterarbeit entworfenen und realisierten Verfahrens. Dabei wurde die Steuerung für beide Achsen der PTU einzeln dargestellt und deren Parameter entsprechend dem Ziel der verbesserten Sensorausnutzung bestimmt. Die abschließende Evaluierung der Verfahren konnte, trotz leichter Schwächen bei der Frontalansicht und Problemen mit dem eingesetzten proprietären Skeletttracking (siehe Anhang *Auswertung des Skeletttrackings*, Seite 90), eine messbare Verbesserung der Aufenthaltswahrscheinlichkeit im Zielbereich von jeweils mehr als 25% gegenüber dem starren Kameraaufbau nachweisen.
Dennoch bleiben klare Anknüpfungspunkte offen. Beispielsweise liegt es nahe, die Oberkörperorientierungsschätzung zur Laufzeit zu verbessern, eine Prädiktion der Nutzertrajektorie zu integrieren oder eine feedbackgestützte Interaktion mit dem Navigationsverhalten des Roboter zu realisieren.
Auch der mögliche Einsatz des Kinect-SDK-Skeletttrackers auf der mobilen Roboterplattform muss kritisch betrachtet werden. Hier gestaltet sich die Problemanalyse sowie die Integration von Lösungsansätzen auf Ebene des SDK-Skeletttrackers aufgrund der nicht öffentlich verfügbaren Implementierung als schwierig. Denkbar ist allerdings beispielsweise eine verbesserte Trackingimplementierung, welche auf den Ausgaben des Kinect-SDK aufbaut. Ferner existiert eine Reihe von klassischer Ansätzen um Nutzerposen unmittelbar aus den Tiefendaten zu ermitteln [Ye et al., 2011, Handrich und

Al-Hamadi, 2013,Shotton et al., 2013]. Verbesserte Algorithmen zur automatischen Synthese von großen Trainingsdatensammlungen für das Training komplexerer Klassifikatoren ermöglichen außerdem den Einsatz von Deep Learning für diese Aufgabe. Als Beispiel für eine Arbeit, in der sowohl Synthese als auch Klassifikation realisiert werden, sei hier die Publikation von [Chen et al., 2016] genannt.

# Kapitel 4

# Gangbasierte Personenwiedererkennung

Um das Gangtraining mit dem Patienten zuverlässig durchführen zu können, muss sichergestellt werden, dass der Assistenzroboter nicht den Kontakt verliert. Dazu wird in der Regel eine Kombination aus Personentracking und -wiedererkennung genutzt [Wengefeld et al., 2016]. In der konkreten Anwendung handelt es sich dabei um eine ansichtsbasierte Wiedererkennung, die den Patient beim Training im Wesentlichen anhand seiner Kleidung identifiziert [Eisenbach et al., 2015b][1]. Durch die im Rahmen des ROGER-Projekts zusätzlich eingesetzte Kinect2-RGB-D-Kamera stehen für die Wiedererkennung von Personen nicht nur hochauflösende Farbbilder zur Verfügung, sondern auch Tiefendaten und gegebenenfalls die daraus abgeleitete Skelettschätzung für bis zu sechs Personen im Sensorbereich.

Da das Gangbild im Rahmen der mobilen Ganganalyse ohnehin kontinuierlich erfasst wird, bietet es sich an, die dabei gewonnenen Informationen in die Wiedererkennung einfließen zu lassen. Die Laufbewegung wird gemeinhin als charakteristisch für eine Person angesehen und zählt zu den sogenannten *weichen biometrischen Merkmalen* [Jain et al., 2004].

Im Rahmen des folgenden Kapitels soll daher nun ein Konzept erarbeitet werden, dass den möglichen Einsatz einer gangbasierten Wiedererkennung auf einer mobilen Platt-

[1] enthält eigene Beiträge

form konzeptionell realisiert. Dazu werden zunächst einige Grundlagen und Begriffe eingeführt, bevor ein umfangreicher Überblick über den State-of-the-Art präsentiert wird. Darauf aufbauend erfolgen einige prototypische Verfahrensideen, welche im Rahmen der gangbasierten Wiedererkennung genutzt werden können. Nach einem kurzen Proof-of-Concept der Verfahren anhand eines eingeschränkten Funktionstests, schließt das Kapitel mit einer kurzen Zusammenfassung sowie einem Ausblick auf mögliche Anknüpfungspunkte weiterführender Arbeiten.

## 4.1 Theoretische Grundlagen

Nachdem das vorherige Kapitel sich weitestgehend mit der Ansteuerung der schwenkbaren Kamera beschäftigt, erfolgt in diesem Abschnitt eine Einführung in die grundlegende Personenwiedererkennung. Anschließend werden einige Begriffe, die in diesem Kontext auftreten, genauer definiert.

### 4.1.1 Personenwiedererkennung

Personenwiedererkennung beschreibt im Allgemeinen die Wiedererkennung von Personen durch informationsverarbeitende Systeme. Im Folgenden liegt der Fokus dabei auf der *dynamischen Wiedererkennung* von Personen anhand einer Sequenz von Sensorinformationen. Dem gegenüber steht die sogenannte *one-shot-Wiedererkennung*, mit der eine Wiedererkennung basierend auf einer einzelnen Szenenaufnahme möglich ist.
Bevor der Wiedererkennungsvorgang beginnen kann, müssen zunächst Personen in der Szene detektiert werden. Da es sich bei der Personendetektion um ein komplexes und umfangreiches Themenfeld handelt, wird sie als abgeschlossene Blackbox-Komponente angesehen, auf die in dieser Masterarbeit nicht vertiefend eingegangen wird. Als Beispiele seien jedoch der bereits beschriebene SDK-Skeletttracker, die auf dem Roboter eingesetzte Detektion nach [Weinrich et al., 2012, Weinrich et al., 2014] oder ein aktueller, Deep-Learning-basierter Ansatz nach [Eisenbach et al., 2016] genannt.
Unter den detektierten Personen wird nun der Nutzer gewählt, der durch den Roboter begleitet werden soll. Im Rahmen der dynamischen Wiedererkennung erfolgt dabei ein

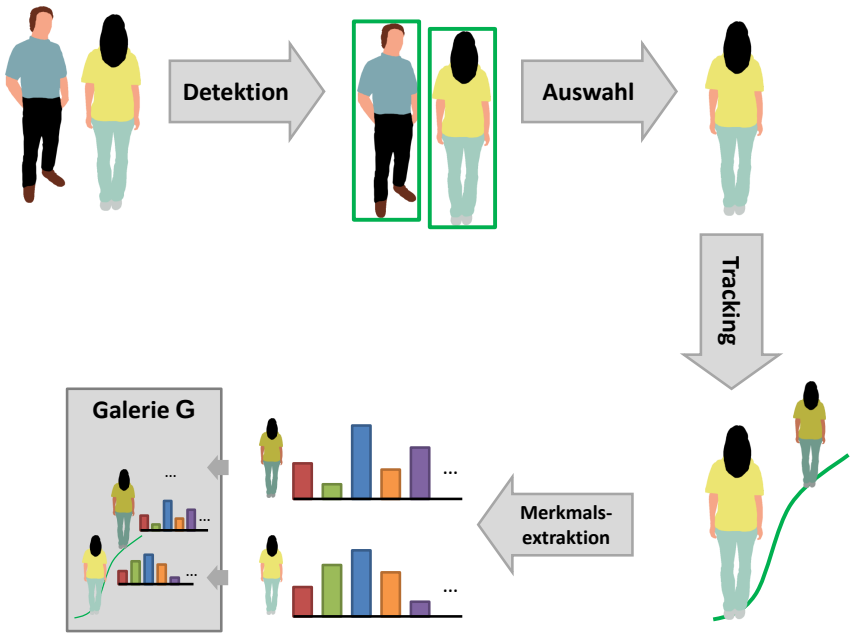



**Abbildung 4.1:** 1. Schritt: Aufbau der Galerie
*Nach der Personendetektion (symbolisiert durch grüne Umrandung) erfolgt die Auswahl des Nutzers, der später wiedererkannt werden sollen. Anschließend erfolgt ein initiales Tracking, bei dem eine Reihe von Daten erfasst und anschließend als Merkmalsvektoren codiert in der Galerie gespeichert werden.*

initiales Tracking, im Zuge dessen eine Bewegungssequenz der Person erfasst und in einer geeigneten Weise codiert und gespeichert wird. Diese Merkmalsrepräsentationen formen die Galerie $\mathcal{G}$ (Abbildung 4.1). Beim Gangtraining mit dem Roboter wird der Patient, mit dem das Training durchgeführt werden soll, während der Anmeldung am Roboter in die Galerie der kleidungsbasierten Wiedererkennung aufgenommen. Die Initialisierung der dynamischen Wiedererkennung erfolgt beispielsweise am Beginn der Laufstrecke.

Muss der Patient nun in einer anderen Szene wiedererkannt werden, müssen alle in Frage kommenden Personen zunächst detektiert und über einen kurzen Abschnitt beobachtet werden. In der Regel erfolgt dieser Prozess kontinuierlich während der Fahrt. Anschließend folgt der gleiche Merkmalsextraktionsprozess wie er bereits ansatzweise

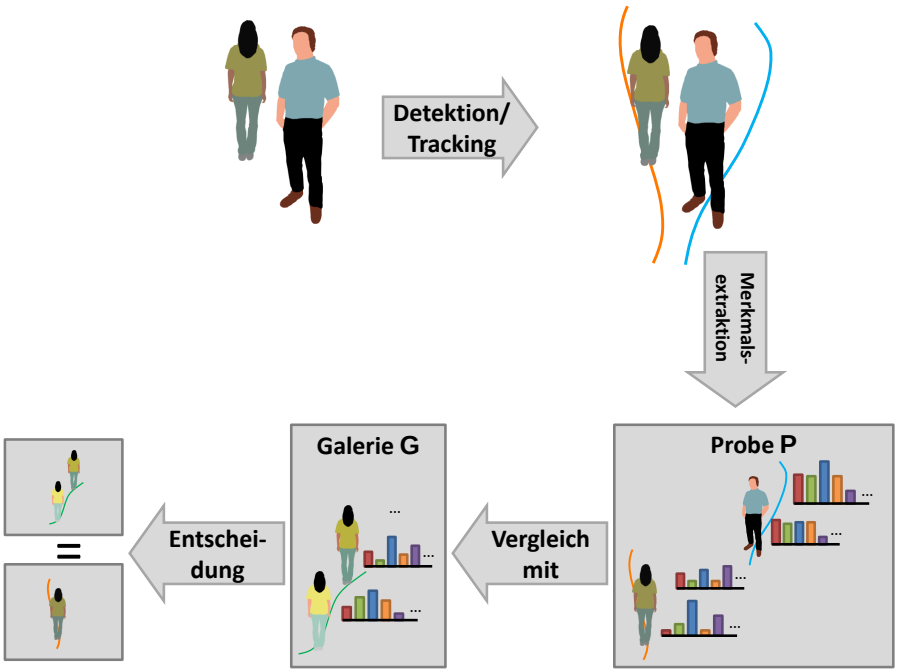



**Abbildung 4.2:** 2. Schritt: Wiedererkennung

*In darauffolgenden Szenen erfolgt wiederum Personendetektion/-tracking, gefolgt von der Merkmalsextraktion für alle Beobachtungen. Die Merkmalsvektoren werden dann mit der Galerie, die in Abbildung 4.1 aufgebaut wurde, verglichen. Anschließend wird die Person aus der Szene als wahrscheinlichster Nutzer ausgewählt, deren Merkmalsvektor die höchste Übereinstimmung mit der Galerie aufweist.*

für die Anmeldung beschrieben wurde, hier jedoch für alle Personen in der Szene. Diese Merkmalsvektoren formen die sogenannte Probe $\mathcal{P}$. Nun erfolgt der Vergleich mit gespeicherten Deskriptoren in der Galerie (Abbildung 4.2).

Dies geschieht in der Regel anhand einer sogenannten Metrik $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ (Definition siehe Anhang A.4.2, Seite 95), die die Distanz zwischen Merkmalsvektoren beschreibt. $\mathcal{G}$ und $\mathcal{P}$ bilden Teilmengen des sogenannten Merkmalsraums $\mathcal{X}$ (Definition siehe Anhang A.4.1, Seite 94). Diese Metrik kann über ein Standarddistanzmaß definiert oder durch Metric Learning (siehe beispielsweise [Vorndran, 2015],[Eisenbach et al., 2015b][2]) bestimmt worden sein. Nachdem die Metrikfunktion zwischen der Galerie $\mathcal{G}$

[2]enthält eigene Beiträge

und allen Elementen aus der Probe $\underline{\mathbf{x}}_p \in \mathcal{P}$ ausgewertet ist, wird die Person als Nutzer wiedererkannt, deren *Distanz-Wert (Score)* $d(\underline{\mathbf{x}}_p, \underline{\mathbf{x}}_g)$ über alle $\underline{\mathbf{x}}_p \in \mathcal{P}$ minimal ist. Da es in realen Einsatzszenarios auch vorkommen kann, dass keine der aktuell detektierten Personen in der Galerie enthalten ist, wird zusätzlich ein Schwellwert $d_{max}$ festgelegt (Abschnitt 4.1.2). Liegt der minimale Score über diesem Schwellwert, konnte in der aktuellen Szene keine erfolgreiche Personenwiedererkennung durchgeführt werden.

### 4.1.2 Wiedererkennugsszenarien und Leistungsbewertung

Sind, wie beispielsweise bei einem Benchmark-Datensatz, alle Personen die in einer Szene erkannt werden können in der Galerie enthalten, spricht man von einem *abgeschlossenen Szenario* (englisch *closed-set scenario*). In diesem Fall erfolgt die Entscheidung anhand der sortierten Reihenfolge der Score-Werte (Abschnitt A.4.2). Für abgeschlossene Szenarien lässt sich eine Cumulative Matching Characteristic (Anhang A.5, Seite 96) erstellen, anhand derer die Erkennungsleistung graphisch charakterisiert und verglichen werden kann. Die Bewertung im Rahmen der Experimente am Ende dieses Kapitels ist in diesen Szenario-Typ einzuordnen.
In einem unbeschränkten öffentlichen Einsatzfeld sind anders als im bisher beschriebenen Fall nicht alle Personen, denen der Roboter begegnen kann, von vornherein bekannt. Wie bereits beschrieben besteht die Galerie im betrachteten Einsatzszenario nur aus dem Patienten, der aktuell begleitet wird. In solchen *offenen Szenarien* (englisch *open-set scenario*) ist es notwendig einen Schwellwert zu definieren, bei dessen Überschreitung das System eine fehlgeschlagene Wiedererkennung signalisiert. Dieser Fall tritt beispielsweise auf, wenn der Roboter den Patienten, dessen Trainingsverlauf er dokumentieren soll, nicht mit der für die Wiedererkennung eingesetzten Sensorik erfassen kann.

## 4.2 State-of-the-Art

Nach [Jain et al., 2004] gliedert sich das Gangbild in die Familie der weichen biometrischen Merkmale ein. Dazu zählen außerdem beispielsweise Körpergröße, Hautfarbe

oder auch das Geschlecht. Im Vergleich zu bekannteren Repräsentanten aus der Familie der biometrischen Merkmale wie etwa Fingerabdruck oder Iris, weisen Merkmale aus dieser Kategorie eine geringere Diskrimanzkraft und Permanenz auf. Jedoch sind sie in der Regel einfacher, über größere Distanzen und ohne explizite Kooperation der zu erkennenden Person erfassbar.

Im Folgenden soll nun ein vertiefter Einblick in verschiedene Ansätze zur gangbasierten Wiedererkennung gegeben werden. Wie bereits in der Einleitung dieses Kapitels erwähnt, handelt es sich bei der gangbasierten Wiedererkennung um eine dynamisches Wiedererkennungsverfahren. Da es sich beim Gangbild um eine dynamisches Verhalten handelt, kann die Charakterisierung nur anhand einer Sequenz von Beobachtungen vorgenommen werden. Im Gegensatz zur häufig eingesetzten ansichts- beziehungsweise kleidungsbasierten Wiedererkennung (beispielsweise [Eisenbach et al., 2015b][3]) ist daher keine one-shot-Wiedererkennung möglich. Die Beschreibung des Gangbildes sowie die Datengrundlage für diese, wird in der Literatur auf vielfältige Weise gehandhabt. Es lassen sich jedoch zwei große Verfahrensfamilien finden, die jeweils charakteristische Vor- und Nachteile aufweisen. Hauptunterscheidungsmerkmal ist dabei auf welcher Basis die Beschreibung der Gangbewegung stattfindet. Davon ausgehend wird zwischen *modell-freien Ansätzen* und *modell-basierten Ansätzen* unterschieden.

Bei den modell-freien Ansätzen erfolgt die Beschreibung des Gangbildes direkt aus den Bild- beziehungsweise Tiefendaten. Meist wird zunächst die menschliche Silhouette extrahiert um den Hintergrund auszublenden. Anschließend wird die Dynamik der Bewegung in geeigneter Weise erfasst und gegebenenfalls in einen Deskriptor umgeformt. Dabei wird die menschliche Körperstruktur nicht explizit berücksichtigt. Es besteht jedoch die Möglichkeit den Ablauf des Gangzykluses beispielsweise über Hidden-Markov-Modelle zu modellieren.

Demgegenüber stehen die modell-basierten Ansätze, bei denen die Beschreibung des Gangbildes über eine explizite Modellierung der menschlichen Körperstruktur erfolgt. In der Regel ist dazu mindestens eine partielle Skelettschätzung notwendig. Die Komplexität des zugrundeliegenden Modells kann jedoch stark variieren. Anschließend wer-

[3] enthält eigene Beiträge

den die Parameter des eingepassten Modells beispielsweise Gelenkpositionen, Länge von Gliedmaßen oder daraus abgeleitete Merkmale für die Beschreibung genutzt.
Die wesentlichen Unterschiede sind in der nachfolgenden Aufzählung nochmals übersichtlich dargestellt. Zusätzlich sei bereits an dieser Stelle auf entsprechende Publikationen aus dem State-of-the-Art hingewiesen.

- modell-freie Ansätze
  - Vorteile
    - Beschreibung mehr oder weniger direkt aus Bild-/Tiefendaten
    - Silhouette vor allem bei statischem Hintergrund leicht zu extrahieren
    - breites Spektrum an möglichen Deskriptoren
  - Nachteile
    - häufig nur mit zusätzlichen Randbedingungen robust möglich
    - Einfluss durch Skalierung/Kamerapose muss ausgeglichen werden
    - in 2D vor allem für Seitenansicht geeignet
    - anfällig gegenüber Veränderungen in der Silhouette
  - Beispielpublikationen
    - 2D: Gait Energy Images [Han und Bhanu, 2006]
    - 3D: Gait Energy Volumes [Sivapalan et al., 2011]
      (Erweiterung von Gait Energy Images für 3D-Daten)
- modell-basierte Ansätze
  - Vorteile
    - skalierungs-/ansichtsinvariant
    - anatomische Randbedingungen explizit über Skelett
    - Merkmale lassen sich gut durch Menschen interpretieren
  - Nachteile
    - (partielle) Skelettschätzung notwendig
    - gegebenenfalls 3D-Rekonstruktion notwendig
  - Beispielpublikationen
    - Kinect (Skelett mit SDK 1.6) [Preis et al., 2012]
    - Kinect2 (Skelett mit SDK 2.0) [Ahmed et al., 2015]
    - 3D-Rekonstruktion aus mehreren Farbbildern [Kwolek et al., 2014]

Nachdem die wesentlichen Unterschiede aufgezeigt wurden, folgt nun ein detaillierter Einblick in ausgewählte Verfahren. Einen guten Überblick vor allem im Bereich der monokulare Ganganalyse bietet auch [Nixon und Carter, 2006].

### Modell-freie Ansätze

Zunächst liegt der Fokus dabei auf den modell-freien Ansätzen. Verfahren in dieser Teildisziplin wurden ab den 1990er Jahren publiziert. Getrieben durch die Fortschritte bei der Bewegungsanalyse aus Bildsequenzen veröffentlichten beispielsweise [Little und Boyd, 1998] ihren Ansatz, der im Kern auf einer Analyse des optischen Flusses im Umfeld der Person besteht. Dazu wird die Person ausgehend von einer fixen Kamera in einer Seitenansicht aufgenommen. Anschließend erfolgt ein Persontracking durch Hintergrundsubtraktion und eine Auswertung des optischen Flusses in der Umgebung der getrackten Person. Mit diesem Verfahren erzielen sie in ihrem Testszenario mit sechs Personen und jeweils sieben Bildsequenzen, basierend auf einem Nearest-Neighbor-Klassifikator, eine Klassifikationsgenauigkeit von etwa 90%.

Eine weitere Arbeit aus diesem Kontext stammt von [BenAbdelkader et al., 2001]. Ausgehend von der gleichen Situation, erfolgt eine Extraktion der Personensilhouette durch Hintergrundsubtraktion für jedes Element der Sequenz. Anschließend wird davon ausgehend eine paarweise Selbstähnlichkeitsmatrix für alle Silhouetten einer Sequenz ermittelt. Die Klassifikation wird dann in einem auf Trainingsdaten mit PCA ermittelten Unterraum, genannt *Eigengait*, mittels Nearest-Neighbor-Algorithmus durchgeführt. Auf den Daten von [Little und Boyd, 1998] erreichen die Autoren damit eine Klassifikationsgenauigkeit von 93%.

Zusätzlich sei hier noch das Verfahren nach [Han und Bhanu, 2006] genannt. Darin wird ein Bewegungsdeskriptor unter der Bezeichnung *Gait Energy Image* vorgestellt. Dabei handelt es sich um eine kompakte Beschreibung einer Sequenz normierter Silhouetten in einem einzigen Bild (Beispiel siehe Abbildung 4.3). Die Autoren begründen ihren Ansatz über Einsparungen beim Speicherplatz, schnelleres Matching und höhere Robustheit gegenüber Fehlern bei der Silhouettenberechnung in einzelnen Bildern.

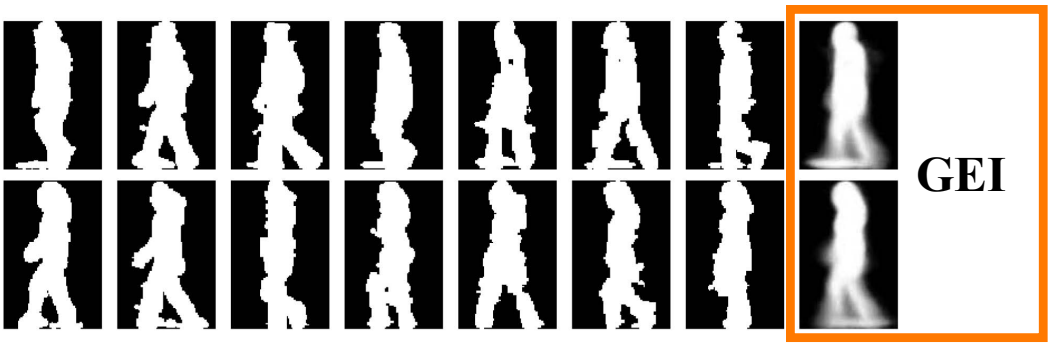
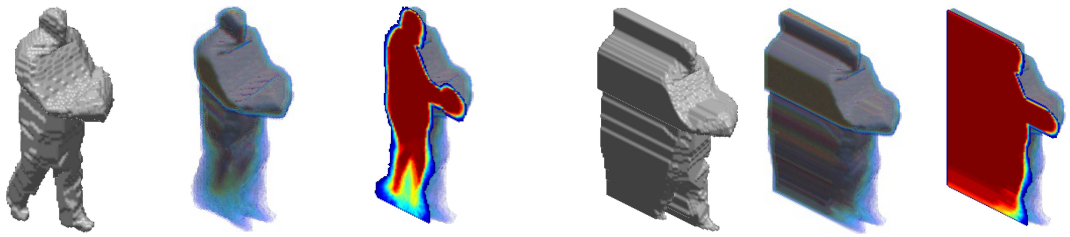



**Abbildung 4.3:** Beispiele für Gait Energy Images
*Zwei Beispiel für Gait Energy Images nach [*Han *und* Bhanu*, 2006]. Die vorgeschlagene Repräsentation (jeweils rechts außen) wurde zur Veranschaulichung durch eine Umrandung hervorgehoben.*

Zusätzlich wird ein komplexes Framework zur Wiedererkennung von Personen anhand der GEI-Deskriptoren unter Einbeziehung verschiedener möglicher Erweiterungen, wie etwa der Fusion mit synthetisch generierten Daten, beschrieben. Durch Experimente auf dem umfangreichen HumanID-Datensatz [Sarkar et al., 2005] (122 Person, 1870 Sequenzen, 12 Experimente mit steigendem Schwierigkeitsgrad), zeigen die Autoren die Gültigkeit ihrer Annahmen. Die erreichten Rang-1- und Rang-5-Raten der CMC erreichen oder übertreffen die zum Vergleich herangezogenen State-of-the-Art-Verfahren.
Alle bisher vorgestellten Verfahren haben jedoch die Einschränkung gemeinsam, dass sie auf zweidimensionalen Bilddaten arbeiten und daher in der Regel auf eine Seitenansicht der Person angewiesen sind um die entsprechenden Deskriptoren bestimmen zu können. Beispielsweise konnten [Yu et al., 2006] zeigen, dass die Leistungsfähigkeit des GEI-basierten Ansatzes mit zunehmendem Unterschied im Winkel bezüglich der Kamera zwischen Galerie und Probe drastisch abnimmt.
Mit dem Aufkommen von leistungsfähigeren Algorithmen zur Rekonstruktion von 3D-Aufnahmen basierend auf kalibrierten Mehrkamerasystemen und günstigen Tiefenkameras wie etwa der Kinect, konnte dieses Problem durch die Arbeit mit dreidimensionalen Daten teilweise gelöst werden. Als Vertreter sei hier die Publikation von [Sivapalan et al., 2011] genannt, welche die Idee der Gait Energy Images unter dem Namen *Gait Energy Volumes* von zweidimensionalen Silhouetten auf dreidimensionale

Voxelsilhouetten erweitert. Beispiele für GEV-Deskriptoren basierend auf einem kalibrierten Mehrkamerasystem und einem Tiefensensor (Kinect), der die Person frontal erfasst, sind in Abbildung 4.4 dargestellt.



(a) GEV-Deskriptor basierend auf einem Multi-Kamera-Setup

(b) GEV-Deskriptor basierend auf Kinect-Sensor

**Abbildung 4.4:** Beispiele für Gait Energy Volumes
*Jede Graphik zeigt (von links nach rechts) die Voxelrekonstruktion, das berechnete Gait Energy Volume und eine Schnittdarstellung des Deskriptors. Beim verwendeten Kinect-Sensor handelt es sich um die erste Generation.*

Experimente auf dem Motion-of-Body-Datensatz [Gross und Shi, 2001] (25 Personen, vier lange Sequenzen pro Person) zeigen eine erkennbare Verbesserung der Erkennungsleistung gegenüber GEI. Gleiches gilt für den von den Autoren zusammengestellten, institutsinternen Datensatz mit frontal aufgenommenen Tiefendaten.
Mit dem bereits angedeuteten Aufkommen der günstigen Tiefensensoren wie etwa der Kinect, verlagerte sich der Fokus von den hier beschriebenen modell-freien Verfahren deutlich stärker hin zu den modell-basierten Ansätzen, auf die nun im Folgenden eingegangen werden soll.

## Modell-basierte Ansätze

Auch modell-basierte Ansätze werden bereits seit längerer Zeit als Möglichkeit zur Bewegungsanalyse und Wiedererkennung von Personen untersucht. Ähnlich wie bei den bereits beschrieben modell-freien Ansätzen, wurde zunächst ausschließlich bildbasiert oder mit Motion Capturing gearbeitet, bevor die Entwicklung von robusten

3D-Rekonstruktions-Algorithmen und verbesserte, preisgünstige Sensor-Hardware 3D- und tiefenbasierte Ansätze ermöglichte.

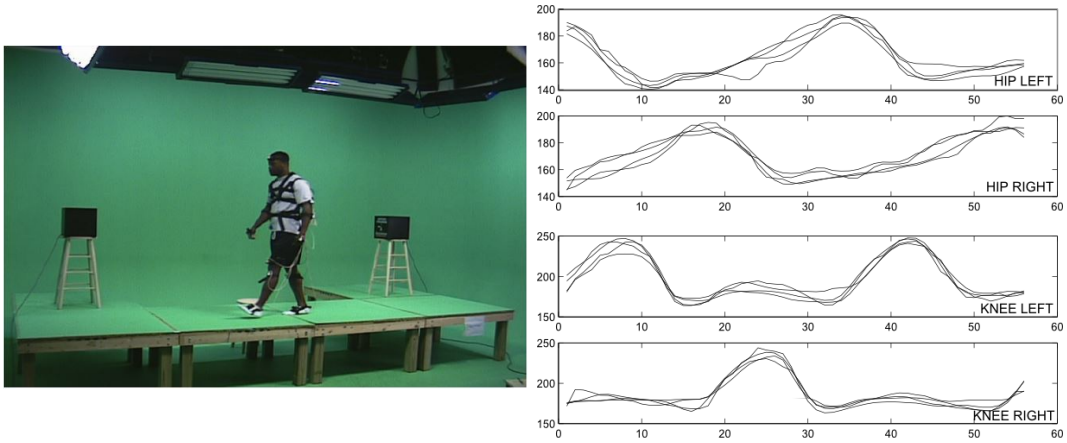
### Bildbasierte Modelle

Bereits in [Hogg, 1983] wurde ein Verfahren vorgestellt mit dem ausgehend von einer monokularen seitlichen Ansicht der Person eine Schätzung der Skelettstruktur ermöglicht wurde. Beispielsweise von [Chen und Lee, 1992] und [Rohr, 1993] folgten frühe Arbeiten, welche dabei gewonnenen Informationen auch für die Erkennung und Tracking von Personen nutzen. Zusätzlich zum Modell das sich aus der Gelenkstruktur des menschlichen Körpers ergibt, nutzen beide Ansätze ergänzende Modelle über den sequentiellen Ablauf der menschlichen Laufbewegung, die teilweise aus medizinischen Studien gewonnen wurden, um Verdeckungssituationen zu behandeln.

Mit der Publikation [Niyogi und Adelson, 1994] erschien kurz darauf die erste Arbeit zur Wiedererkennung anhand von Merkmalen die aus dem Gangbild abgeleitet werden. Ausgehend von einer Silhouette wird die Kontur der Person abgeleitet und anschließend ein einfaches Skelettmodell (Wirbelsäule, Ober-/Unterschenkel links/rechts) eingepasst. Die Behandlung von Eigenverdeckungen wird ebenfalls durch Wissen über den spatio-temporalen Verlauf der Bewegung realisiert. Anschließend erfolgt eine Extraktion der Gelenkwinkel in der Hüfte und im Knie für linkes und rechtes Bein. Nach einer zeitlichen Normalisierung und Verknüpfung der Signale, ist die Klassifikation mittels Nearest-Neighbor-Ansatz möglich. Experimente auf einem Datensatz mit 26 Sequenzen, die Personen in einer Seitenansicht zeigen, konnte mittels Leave-One-Out-Kreuzvalidierung eine Erkennungsrate von 81% erreicht werden.

Ein weiterer Ansatz wird durch [BenAbdelkader et al., 2002] beschrieben. Darin befassen sich die Autoren mit der Möglichkeit, Personen anhand sogenannter Zeit-Distanz-Parameter wiederzuerkennen. In der gangbasierten Wiedererkennung oder auch im medizinischen Bereich versteht man darunter aus dem Gangbild abgeleitete Maße, wie beispielsweise Schrittfrequenz (Kadenz) und Schrittlänge. Dazu wird die Person über einen Zeitraum von $10 - 15$s durch eine kalibrierte Kamera beobachtet. Anschließend erfolgt eine Schätzung der für einen Doppelschritt benötigten Zeit

und der gelaufenen Wegstrecke. Aus diesen beiden Parametern wird unter Annahme einer konstanten Geschwindigkeit die Schrittlänge und die Kadenz (Schritte pro Minute) berechnet. Die experimentelle Evaluation wird auf einem Datensatz mit 17 Personen (131 Sequenzen, ∅8 pro Person) mittels Leave-One-Out-Kreuzvalidierung durchgeführt. Die Sequenzen des Datensatzes wurden aus rund 30 m Entfernung bei geringer Auflösung aufgenommen. Eine Auswertung der Rangstatistik (CMC) liefert eine Wiedererkennungsrate von etwa 38% auf Rang 1 und etwa 95% auf Rang 5.




(a) Motion-Capturing-Aufbau (b) Beispieltrajektorien für Hüfte und Knie

**Abbildung 4.5:** Wiedererkennung anhand von Gelenkpunkttrajektorien nach [Tanawongsuwan und Bobick, 2001]
*(a) Datenaufnahme über magnetisches Motion-Capturing-System (2000er Jahre).*
*(b) Darstellung beispielhafter Gelenkpunkttrajektorien. Die dargestellten Trajektorien sind bereits zeitlich normiert und zeigen die Verläufe für verschiedene Personen.*

Ferner existieren Pionierarbeiten wie etwa [Tanawongsuwan und Bobick, 2001], welche die Nutzung der Gelenkpunkttrajektorien als Wiedererkennungsmerkmal basierend auf einem Motion-Capturing-Aufbau entsprechend Abbildung 4.5(a) untersuchen. Dazu werden die erfassten Schrittfolgen in Doppelschritte segmentiert und in der Varianz sowie zeitlich mittels Dynamic Time Warping normalisiert (Abbildung 4.5(b)). Für den evaluierten Datensatz mit 18 Personen und 106 Sequenzen, wird eine Wiedererkennungsrate von 42% für Rang 1 und etwa 78% auf Rang 5 angegeben. Den

Autoren gelingt es somit zu zeigen, dass sich aus Gelenkpunkttrajektorien Signaturen ermitteln lassen, die zur Wiedererkennung eingesetzt werden können.

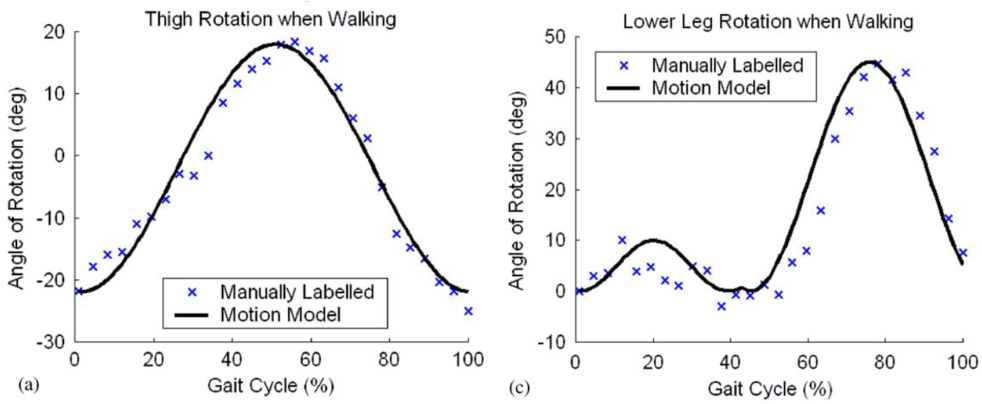



**Abbildung 4.6:** Gelenkpunktwinkeltrajektorien nach [Yam et al., 2004]
*Dargestellt ist der zeitliche Verlauf der Knie- und Hüftbeugung, der aus den ermittelten Gelenkpunkttrajektorien abgeleitet wird. Als Referenz wird jeweils der manuell erzeugte Ground-Truth-Verlauf genutzt.*

[Yam et al., 2004] gelingt es die Gültigkeit dieses Ansatzes auch für die bildbasierte Wiedererkennung zu zeigen. Zusätzlich werden einige Erweiterungen in der Merkmalsextraktion vorgeschlagen. Beispielsweise wird ausgehend von den Gelenkpunkttrajektorien der zeitliche Verlauf der Hüft- und Kniewinkel ermittelt (Beispiele siehe Abbildung 4.6). Die zeitlichen Verläufe werden anschließend normiert und per Fourier-Analyse in eine Betrags- und Phasendarstellung überführt. Zur Evaluation wird ein Datensatz mit 20 Personen herangezogen. Für jede Person wurde fünf seitliche Aufnahmen auf einem Laufband durchgeführt. Die Klassifikation dieser Sequenzen mittels k-Nearest-Neighbor-Methode wird basierend auf einer Leave-One-Out-Kreuzvalidierung eine mit einer korrekten Erkennungsrate von 85% angegeben. Eine Auswertung der Rangstatistik erfolgt nicht.

Abschließend sei noch der Ansatz nach [Bobick und Johnson, 2001] genannt, der sich konzeptionell von den bisher genannten Verfahren unterscheidet. [Bobick und Johnson, 2001] distanzieren sich dabei ein Stück weit von der dynamischen Natur der gangbasierten Wiedererkennung und werten lediglich den Prozess aus um einige stati-

sche Parameter abzuleiten. Bei den statischen Parametern handelt es sich dabei um die Körpergröße, die Länge des Oberkörpers sowie die Länge der Beine und der maximale Abstand zwischen rechten und linkem Fuß (Schrittweite). Die Schätzung dieser Werte geschieht aus der Laufbewegung heraus und wird in der sogenannten Double-Support-Phase durchgeführt. Dabei berühren beide Füße den Boden, der Abstand zwischen den Fußgelenken ist maximal. Um eine robustere Schätzung zu erhalten, erfolgt die Mittelwertbildung über mehrere Events. Die Ergebnisse der bildbasierten Schätzung werden experimentell über ein Motion-Capturing-System validiert. Eine Bewertung des Ansatzes hinsichtlich der Erkennungsleistung erfolgt basierend auf einer ROC-Kurve. Die Auswertung der Rangstatistik wird nicht durchgeführt.

### 3D- und tiefenbasierte Modelle

Ähnlich wie bei den zuvor präsentierten modell-freien Ansätzen, konnte die Entwicklung der modell-basierten Ansätze in großen Umfang vom Erscheinen günstiger Tiefenkameras profitieren. Dadurch wurde der Entwurf kompakterer, realwelttauglicher 3D-basierter Systeme möglich. Die Pionierarbeit zur gangbasierten Wiedererkennung ausgehend von Kinect-Tiefendaten/-Skelettdaten mittels eines modell-basierten Ansatzes erfolgte durch [Preis et al., 2012]. Das Verfahren stützt sich dabei auf den im SDK der Kinect (1. Version) realisierten Skelettracker. Basierend auf den 20 Skelettpunkten, die in dieser Version geschätzt werden, wird neben einer Reihe statischer Merkmale wie Körpergröße oder der Länge von Armen und Beinen, auch die Schrittlänge und Kadenz ermittelt. Die Auswahl der Merkmale orientiert sich dabei an den Arbeiten von [Bobick und Johnson, 2001] und [BenAbdelkader et al., 2002]. Nach einer experimentellen Evaluation mit acht Personen (67 Sequenzen), die seitlich von einer Kinect aufgenommen wurden, wird das Ergebnis mit einer korrekten Klassifikationsrate (Naive-Bayes-Klassifikation) von 91% angegeben. Es existiert eine Reihe weiterer Arbeiten zur Kinect1, die auf diesem Merkmalsset aufbauen und es gegebenenfalls um weitere Elemente wie Distanzen zum Oberkörperschwerpunkt [Sinha et al., 2013] oder Geschwindigkeitsmerkmale basierend auf einer Schätzung der Sagittalebene erweitern [Gianaria et al., 2014],. Außerdem erfolgten erste Vorschläge

zur automatischen Bestimmung relevanter Merkmale beispielsweise mittels künstlicher Neuronaler Netzwerke [Sinha et al., 2013] oder Mutual Information [Gianaria et al., 2014].

Der Gedanke zur Auswahl von relevanten Merkmalskombinationen automatische Verfahren einzusetzen wird auch von [Ahmed et al., 2015] in einer der ersten Arbeiten basierend auf der Kinect2 und dem aktualisierten Kinect-SDK 2.0 aufgegriffen. Dazu wird zunächst ein umfangreiches Merkmalsset basierend auf den Distanzen (*joint relative distance*, kurz: *JRD*) und Winkeln (*joint relative angle*, kurz: *JRA*) zwischen beinahe allen 25 Gelenkpunkten des Skeletts definiert. Ausgeschlossen werden dabei nur Punkte die eine hohe Rauschanfälligkeit haben, wie etwa die Verfeinerungen der Hand oder die Fußspitze. Anders als bisher üblich, werden damit auch Winkel zwischen Gelenkpunkten berechnet, die nicht miteinander verbunden sind. Gleiches gilt auch für die Abstände zwischen den Gelenkpunkten, so dass eine Mischung aus statischen (etwa Länge der Arme/Beine oder des Rumpfes) und dynamischen Distanzen (beispielsweise rechter Fuß ↔ linker Fuß, Kopf ↔ rechtes/linkes Knie) entsteht. Insgesamt ergibt sich so ein Pool von etwa 300 Merkmalen.

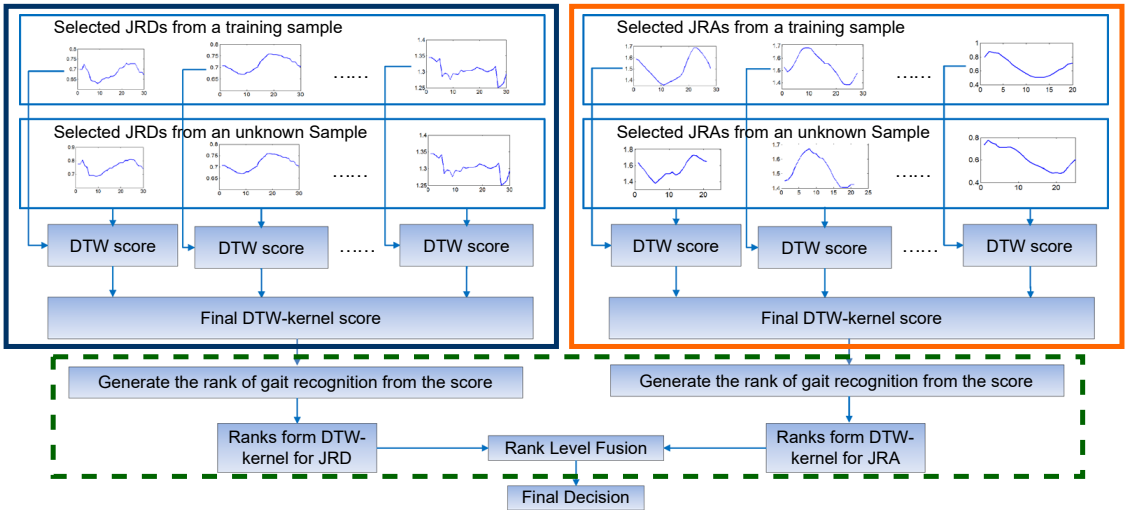



**Abbildung 4.7:** Ablauf der Wiedererkennung nach [Ahmed et al., 2015]
*Die Berechnung der Kernelcodierung, Matching und Ranking erfolgt für JRD- und JRA-Merkmale zunächst getrennt. Die Ergebnisse der Rankings werden anschließend fusioniert und bilden den finalen Score.*

Ausgehend von diesen Merkmalen wird ein Wiedererkennungsschema definiert, dessen Ablauf auch in Abbildung 4.7 dargestellt ist. Im Wesentlichen erfolgt dabei voneinander unabhängige Berechnung der Scores für JRA- und JRD-Merkmale. Anschließend werden basierend auf diesen Scores zwei separate Rankings erstellt. Die Ergebnisse der beiden Rankings werden anschließend durch Rank-Level-Fusion kombiniert und liefern die finale Reihenfolge auf der die CMC-Statistik ermittelt werden kann. In einer Variation dieses Ablauf, wird ein genetischer Algorithmus zur Auswahl bestimmter JRA-/JRD-Komponenten angewendet, dessen Erfolg an der Wiedererkennungsleistung des im Genom codierten Merkmal-Subsets gemessen wird.

Zur experimentellen Evaluation wird ein Datensatz mit 20 Personen (60 Sequenzen) genutzt. Eine getrennte Evaluation von JRA- und JRD-Merkmalen liefert eine Wiedererkennungsrate von etwa 78% bzw. 80% auf Rang 1 (CMC). Dabei wurden vom genetischen Algorithmus Subsets von 15 JRA- und 25 JRD-Merkmalen ausgewählt. Eine anschließende Rank-Level-Fusion der Ergebnisse aus beiden Subsets, liefert eine Wiedererkennungsrate von 92% auf Rang 1 (CMC). Als Referenz werten [Ahmed et al., 2015] den Ansatz von [Preis et al., 2012] auf ihrem Datensatz aus und ermitteln eine Wiedererkennungsrate von 84% auf Rang 1 der CMC-Statistik.

## Einordnung der Masterarbeit

Im Gegensatz zu den vorgestellten State-of-the-Art-Verfahren, soll im Rahmen der Masterarbeit der Ansatz für die gangbasierte Wiedererkennung auf einer mobilen Roboterplattform entwickelt werden. Durch die angedachte Eingliederung in die autonome Gangschule des ROGER-Projekts, bietet es sich an einen modell-basierten Ansatz zu realisieren. Die modell-basierte Verfahrensweise ausgehend vom Skelett bietet außerdem bessere Anhaltspunkte um Glättungs- und Interpolationsverfahren zu integrieren, um Probleme die beim Skeletttracking auf mobilen Plattformen auftreten teilweise kompensieren zu können. Beim Merkmalsdesign wird auf einen Ansatz ähnlich dem von [Preis et al., 2012] zurückgegriffen. Zusätzlich wird die Möglichkeit in Betracht gezogen, etwa den zeitlichen Verlauf von Knie und Hüftwinkel auszuwerten. Zur Be-

wertung der Wiedererkennungsleistung wird ein einfacher Matching-Ansatz gewählt, da sich Erweiterungen auch nach dem Proof-of-Concept noch integrieren lassen.

## 4.3 Verfahren

Nach dem umfangreichen Überblick über den State-of-the-Art zur gangbasierten Wiedererkennung, wird nun im Folgenden der im Rahmen der Masterarbeit entwickelte Ansatz für eine mobile Roboterplattform skizziert. Das Verfahren soll wie bereits bei der Einordnung in den State-of-the-Art beschrieben, modell-basiert realisiert werden. Als Datengrundlage dienen Tiefendaten beziehungsweise die darauf aufbauenden Skelettdaten der auf Ringo verbauten Kinect2. Wo möglich wird jedoch von der konkreten Realisierung der Skelettschätzung abstrahiert um mögliche Alternativen offen zu lassen. Um die Beschreibung verschiedener Skelettpunkte zu vereinfachen wird jedoch die Terminologie entsprechend dem Skelett des Kinect-SDK genutzt. Die numerische und namentliche Bezeichnung der einzelnen Punkte ist in Abbildung 4.8 veranschaulicht.
Bei der Beschreibung des Verfahrens wird auf folgende Aspekte eingegangen:

1. Datenerfassung
2. Vorverarbeitung
3. Segmentierung der Sequenz in Doppelschritte
4. Merkmalsextraktion
5. mögliche Integration in die Gesamtapplikation

Wo möglich wird dabei explizit auf Herausforderungen durch die mobile Plattform eingegangen.

### 4.3.1 Datenerfassung

Für die Datenerfassung wird vom Aufbau ausgegangen, wie er im Kapitel zur stabilen Erfassung von Personen im Abschnitt *Montage und Hardware* (Seite 18) beschrieben wird. Dabei ist die Kinect2 in etwa 1,3 m Höhe am Roboter angebracht und beobachtet den Patienten während der Fahrt, eventuell auch durch Zuhilfenahme der PTU, aus einer frontalen Perspektive. In der Regel erfolgt damit eine kontinuierliche Erfassung

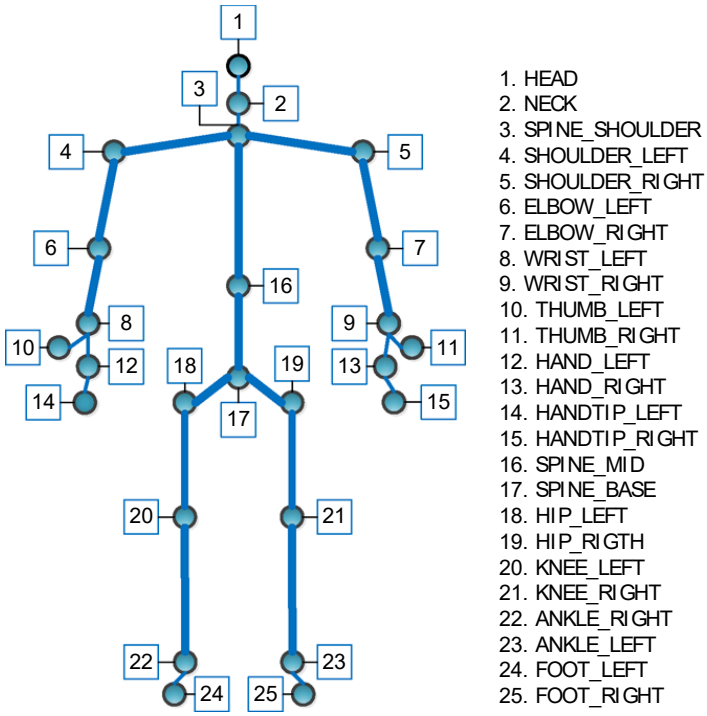


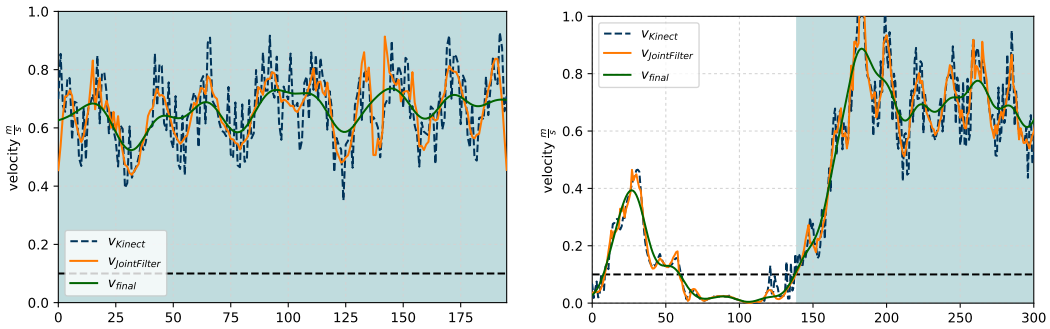

**Abbildung 4.8:** Bezeichnung der Kinect2-Skelettpunkte
*Numerische und namentliche Bezeichnung der Skelettpunkte aus [Ahmed et al., 2015]. Für den hier betrachteten Ansatz sind vor allem die die Gelenkpunkte 16 - 25 von Interesse.*

des Patienten. Die Datenerfassung wird solange fortgeführt wie ein stabiles Skelett geschätzt werden kann oder eine festgelegte Anzahl Samples (beispielsweise 450 $\approx$ 15 s) erfasst wurde. Für die Vereinfachung aller folgenden Beschreibung, sei ferner angenommen, dass die Skelettdaten unmittelbar nach der Aufnahme in ein festes, externes Bezugssystem überführt werden.

### 4.3.2 Vorverarbeitung

Durch die mobile Plattform wird die Unsicherheit der Kinect-Daten erhöht. Dies geschieht vor allem durch Vibrationen, da diese zu marginal sind um von der Odometrie vollständig erfasst und korrigiert werden zu können. Es empfiehlt sich daher die in der Datenerfassung gewonnenen Trajektorien und daraus abgeleitete Parameter zu filtern.

Das im Rahmen dieser Masterarbeit entworfene Verfahren wendet dazu einen Tiefpassfilter getrennt auf alle drei Raumkoordinaten an um grobe Abweichungen abzumildern. Andere Vorgehensweise wie etwa der Einsatz eines Kalman-Filters sind denkbar.
Als nächstes sollte geprüft werden, ob sich die Person oder die Personen im Kamerabild bewegen, da es sich dabei um eine Grundvoraussetzung für die gangbasierte Wiedererkennung handelt. Möglich ist eine derartige Analyse beispielsweise über die räumliche Position des `SPINE_BASE`-Markers (Becken). Dieser Marker bietet sich an, da er in der Regel relativ robust geschätzt werden kann. Auch andere Punkte aus dem Bereich der Wirbelsäule sind denkbar, da diese einer geringeren Dynamik unterliegen als Punkte an den Extremitäten. Um die Geschwindigkeit zu berechnen wird die Distanz des Beckenpunkts zwischen einem Sample und seinem 5. Vorgänger in der Sequenz ausgewertet und durch die verstrichene Zeit dividiert. Den Vergleich nicht unmittelbar mit dem nächsten Frame durchzuführen, reduziert die Fehleranfälligkeit gegenüber Positionsrauschen und Variationen in der zeitlichen Abfolge. Gegebenenfalls wird das Signal anschließend noch einmal mit einem Tiefpass gefiltert.




(a) gleichbleibende Geschwindigkeit (b) starke Änderung in der Geschwindigkeit

**Abbildung 4.9:** Geschwindigkeitsberechnung mit Rauschunterdrückung
*Dargestellt ist jeweils der Geschwindigkeitsverlauf, wenn die Distanz zum 5. Vorgänger-Sample berechnet wird. Blau markiert ist der Geschwindigkeitsverlauf der sich den Rohdaten der Kinect ergibt, orange nach Anwendung des Tiefpassfilters auf die Gelenkpunkte der Koordinaten und grün nach Anwendung eines weiteren Tiefpassfilters auf das Geschwindigkeitssignal* $v_{JointFilter}$. *Die Hintergrundmarkierung signalisiert den Bereich der als „in Bewegung“ klassifiziert wird.*

Abbildung 4.9 zeigt den Verlauf der Geschwindigkeitssignale nach verschiedenen Vorverarbeitungsstufen. Durch die Anwendung der Tiefpassfilterung lässt sich der Verlauf deutlich glätten. Periodische Schwankungen in der Geschwindigkeit lassen sich auf dem Bewegungszyklus des Menschen zurückführen. Als unterer Geschwindigkeitsschwellwert wird im Rahmen der prototypischen Untersuchung 0,1 $\frac{\mathrm{m}}{\mathrm{s}}$ gewählt. Für eine Anwendung im Rehabereich kann eine Korrektur des Schwellwertes je nach schwere der Einschränkung im Bewegungsapparat notwendig sein. Um den Einfluss durch unregelmäßige Bewegungen beim An- und Auslaufen zu minimieren, empfiehlt sich ferner die Analyse der Geschwindigkeitsänderung beispielsweise über die Streuung. Hier bietet sich unter anderem die Auswertung der laufenden Standardabweichung über etwa 1-2s an.

### 4.3.3 Segmentierung der Sequenz in Doppelschritte

Die Segmentierung der Sequenzen in Schritte oder Doppelschritte ist ein Standardverfahren in der gangbasierten Wiedererkennung und trägt der Periodizität der Laufbewegung Rechnung, wie sie in Abbildung 4.10 beispielhaft dargestellt ist. Sie dient als Grundlage für die Merkmalsextraktion und ermöglicht eine zeitliche Normalisierung der ermittelten Verläufe. Dadurch lassen sich Unterschiede in der Laufgeschwindigkeit bis zu einem gewissen Grad ausgleichen. Es existieren Untersuchungen wie etwa durch [Yam et al., 2004], die sich mit der Charakteristik der Gangbewegung bei unterschiedlichen Geschwindigkeiten (Gehen und Rennen) beschäftigen. Unter Berücksichtigung des betrachteten Einsatzszenarios, wird im folgenden von einer Gehgeschwindigkeit bis maximal 1,5 $\frac{\mathrm{m}}{\mathrm{s}}$ ausgegangen.

Die Veranschaulichung in Abbildung 4.10 zeigt eine volle Periode des vereinfachten menschlichen Ganzzyklus ausgehend von der sogenannten Double-Support-Phase, bei der beide Füße den Boden berühren. Anschließend erfolgt der Übergang in die Schwung- und schließlich die Standphase. Nach einer weiteren Schwungphase wird wieder der Double-Support erreicht, allerdings seitenverkehrt zur Ausgangssituation. Diese Bewegungssequenz wird als *Schritt* definiert. Wird zusätzlich noch der weitere Bewegungsverlauf (Schwung $\rightarrow$ Stand $\rightarrow$ Schwung) bis hin zur nächsten Double-

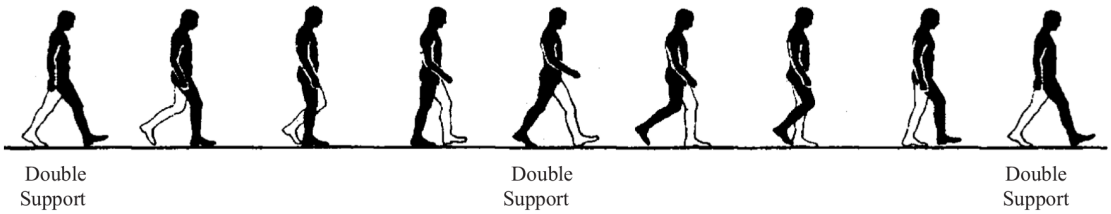
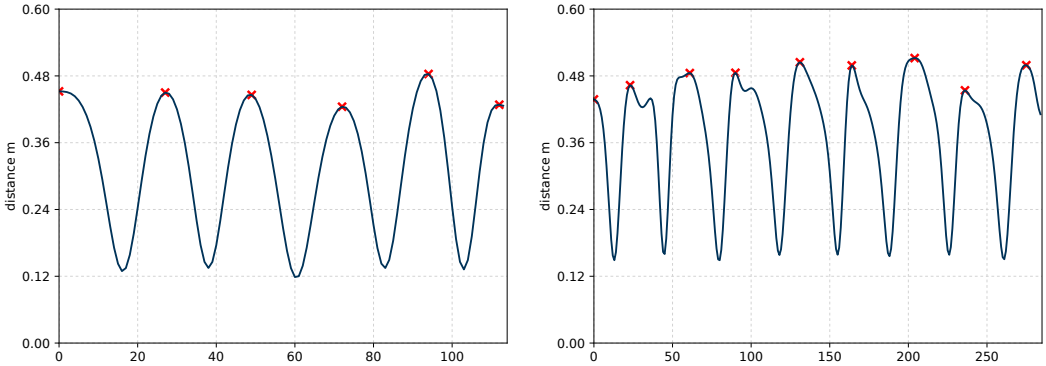



**Abbildung 4.10:** Die menschliche Laufbewegung
*Die Graphik nach [Yam et al., 2004] veranschaulicht den Zyklus einer typischen menschlichen Laufbewegung auf einer ebenen Fläche.*

Support-Phase betrachtet, spricht man von einem *Doppelschritt*. Nach einem Doppelschritt befindet sich die Person im Wesentlichen wieder in der Ausgangssituation der Bewegung.
Ein Kriterium, das die Erkennung der Double-Support-Phasen ermöglicht und auch in weiten Teilen der State-of-the-Art-Literatur eingesetzt wird (beispielsweise [Preis et al., 2012, Kwolek et al., 2014, Ahmed et al., 2015]), ist die Auswertung der Distanz zwischen den beiden Fußgelenkpunkten. Diese folgt dem periodischen Verlauf und zeigt lokale Maxima in den Double-Support-Phasen der Laufbewegung. Die Position dieser lokalen Maxima dient als Ausgangspunkt der Segmentierung. Zur Bestimmung dieser lokalen Maxima wird im Rahmen der Masterarbeit ein simples Verfahren angewendet, welches das Signal über einen adaptiven Schwellwert in disjunkte Teilsequenzen zerlegt, für die anschließend das globale Maximum ermittelt wird (siehe dazu auch Abbildung 4.11).

### 4.3.4 Merkmalsextraktion

Die Berechnung der Merkmale erfolgt basierend auf den tiefpassgefilterten Raumkoordinaten der verschiedenen Skelettpunkte. Zunächst wird der zeitliche Verlauf der beiden Hüftwinkel extrahiert. Dazu wird der Winkel ausgewertet, der zwischen der Wirbelsäule, repräsentiert durch `SPINE_BASE` und `SPINE_MID` und dem rechten (repräsentiert durch `HIP_RIGHT` und `KNEE_RIGHT`) beziehungsweise dem linken Oberschenkel (repräsentiert durch `HIP_LEFT` und `KNEE_LEFT`) aufgespannt wird. Die Betrach-




(a) wohldefinierte Maxima (b) unscharfe Maxima

**Abbildung 4.11:** Double-Support-Phasen-Erkennung
*Dargestellt ist jeweils Verlauf der Distanz zwischen den beiden Fußgelenken in blau und die ermittelten lokalen Maxima in rot. (a) zeigt einen Verlauf, in dem die Spitzen relativ scharf sind und die lokalen Maxima mit hoher Sicherheit die Double-Support-Phase treffen. In (b) sind die Spitzen weniger scharf ausgeprägt. Daher ist die Schätzung hier mit einer höheren Unsicherheit verbunden.*

tung dieser Winkel erlaubt die Klassifikation der zuvor anhand der lokalen Maxima berechneten Segmentierungspunkte ohne Kenntnis der Sagitallebene beziehungsweise der Laufrichtung. Dadurch lassen sich „gleichartige“ Double-Support-Phasen im Bewegungsablauf erkennen. Double-Support-Phasen werden im Rahmen dieser Arbeit als gleichartig angesehen, wenn beide Beine in der gleichen Position sind (beispielsweise rechtes Bein vorne, linkes Bein hinten). Beispielsweise sind die erste und die letzte Double-Support-Phase in Abbildung 4.10 gleichartig. Damit wird die Grundvoraussetzung geschaffen, den zeitlichen Verlauf der Merkmale getrennt für rechtes und linkes Bein zu betrachten. In der Literatur wird aber auch eine simple Einordnung jeder zweiten Double-Support-Phase in die gleiche Kategorie eingesetzt. In einem analogen Vorgehen, wird noch der zeitliche Verlauf für den rechten und den linken Kniewinkel extrahiert. Die beiden dynamischen Zeitverläufe werden anschließend basierend auf den kategorisierten Segmentierungspunkten in einzelne Abschnitte aufgeteilt und durch Resampling zeitlich normiert. Nach der zeitlichen Normierung erfolgt eine Mit-

telwertbildung über alle Abschnitte einer Sequenz. Zusätzlich wird die Streuung der Verläufe ermittelt.
Als weitere Merkmale lässt sich die Schrittweite direkt aus den Segmentierungszeitpunkten ableiten, indem die Distanz der Fußgelenkpunkte zwischen zwei aufeinanderfolgenden gleichklassifizierten Markern ermittelt wird. Analog zum Vorgehen bei der Extraktion der Gelenkwinkel, erfolgt eine Mittelwertbildung über alle Werte einer Sequenz. Gleiches gilt für die Berechnung der Kadenz, die sich aus der Anzahl der Schritte pro Zeiteinheit relativ direkt aus den Zeitpunkten berechnet werden kann. Tabelle 4.1 zeigt eine Kategorisierung der Merkmale. Die Einordnung der Kadenz und Schrittweite dynamische Größen basiert auf der Annahme, dass diese abhängig von der Laufgeschwindigkeit sind [BenAbdelkader et al., 2002, Preis et al., 2012].

| Merkmal | Typ | Normierung |
|---|---|---|
| Hüftwinkel | Zeitverlauf | Resampling |
| Kniewinkel | Zeitverlauf | Resampling |
| Schrittweite | dynamisch, skalar | -/- |
| Kadenz | dynamisch, skalar | -/- |
| Körpergröße | statisch, skalar | -/- |

**Tabelle 4.1:** Übersicht über die Wiedererkennungsmerkmale

Details zu den einzelnen Merkmalsextraktionsschritten folgen in den als Proof-of-Concept ausgelegten Experimenten im folgenden Abschnitt. Dort wird dann auch auf weitere Erfahrungen aus den Experimenten mit der mobilen Plattform eingegangen.

## 4.4 Proof-of-Concept

Da die folgenden Experimente als Proof-of-Concept ausgelegt sind, bleiben die Nachweise an viele Stellen unvollständig. Die Ergebnisse sollen daher viel mehr Anreize geben, an die sich weitere Untersuchungen anschließen können und müssen. Entsprechend dieser Ausrichtung und unter Berücksichtigung der bekannten Schwächen des eingesetzten Skelett-Trackings aus dem Kinect-SDK, erfolgen die Experimente unter

kontrollierten Bedingungen im Zuse-Bau der Fakultät für Informatik und Automatisierung an der TU Ilmenau mit sechs Mitarbeiten und Studenten aus dem Fachgebiet Neuroinformatik und Kognitive Robotik. Zwei Teilnehmer absolvieren die Trainingsstrecke noch ein weiteres Mal mit Unterarmgehstützen. Dadurch enthält der Datensatz nunmehr acht Personen, da sich das Gangbild mit und ohne Gehhilfen deutlich unterscheiden sollte. Bei allen in diesem Abschnitt getroffenen Aussagen ist daher der geringe Umfang des Datensatzes sowie die deutlich vom realen Einsatzszenario abweichende Personenzielgruppe zu beachten. Als Aufnahmeplattform wird Ringo mit deaktivierter horizontaler Kamerasteuerung und um 11° geneigter Kinect2 eingesetzt. Der Winkel wurde entsprechend dem Teil zur vertikalen Ansteuerung der PTU (*Vertikale Steuerung*, ab Seite 32) aus dem Kapitel zur stabilen Erfassung von Personen gewählt.

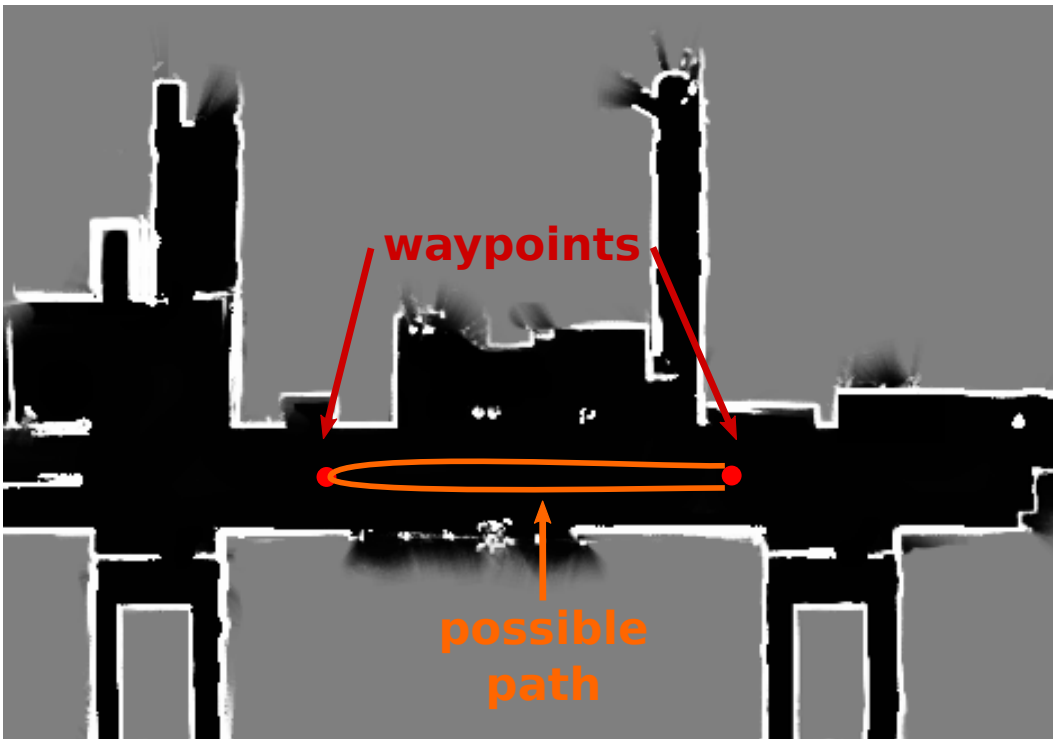



**Abbildung 4.12:** Aufnahmestrecke gangbasierte Wiedererkennung
*Dargestellt sind die beiden Wegpunkte zwischen denen die Zielanfahrten durchgeführt werden sowie eine mögliche Fahrstrecke.*

Wie in Abbildung 4.12 dargestellt, wurden zwei Wegpunkte definiert, zwischen denen der Roboter autonom Zielanfahrten mit geringer Geschwindigkeit (bis etwa 0,8 $\frac{\mathrm{m}}{\mathrm{s}}$) durchführt. Da der Roboter in diesem Navigationsmodus den Abstand zu Personen nicht selbstständig einhält, wurden die Probanden gebeten selbst auf dessen Einhaltung zu achten. Insgesamt wurden mit sechs Mitarbeiten 12 Fahrten (jeweils Hin- und Rückweg) durchgeführt, zwei davon mit Unterarmgehstützen. Der Datensatz umfasst daher insgesamt acht Personen. In jeder Aufnahme wurden entsprechend der Beschreibung im Abschnitt *Datenerfassung* Sequenzen mit stabiler Skelettschätzung mit einer maximalen Länge von etwa 400 Frames extrahiert und analysiert. Dadurch ergeben sich 52 Sequenzen für die jeweils die beschriebenen Merkmale extrahiert werden. Die zeitlichen Verläufe der Knie und Hüftwinkel werden jeweils für rechtes und linkes Bein durch Resampling auf eine einheitliche Länge von elf Datenpunkten normiert.

### 4.4.1 Datenerfassung

Bei der Datenerfassung im Rahmen der Experimente konnten Beobachtungen aus den Experimenten zur Kamerasteuerung bezüglich des Skeletttrackers teilweise bestätigt werden. Unter ruhigen Fahrbedingungen funktioniert die Skelettschätzung hinreichend, schnelle Lenkmanöver oder größere Unebenheiten der Fahrbahn führen auch weiterhin zum Abreißen des Trackings. Eine weitergehende Analyse soll jedoch nicht Bestandteil dieser Arbeit sein.

### 4.4.2 Segmentierung der Sequenz in Doppelschritte

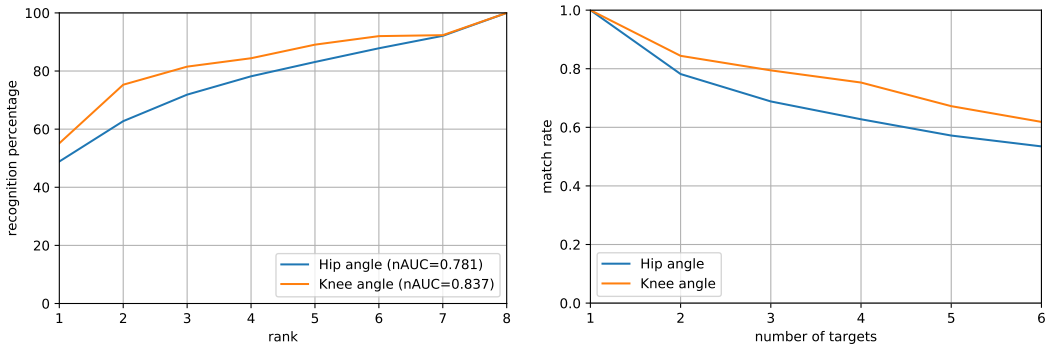
Der im vorherigen Abschnitt beschriebene Ansatz zur Segmentierung der Sequenzen in Doppelschritte, hat sich im Rahmen der Experimente als relativ robust erwiesen. Lediglich bei langen Sequenzen, während derer sich die Laufgeschwindigkeit der Person deutlich ändert, können Probleme auftreten. Da jedoch die maximale Länge der Sequenzen auf 450 Frames limitiert ist, entstehen dadurch kaum Probleme. Lediglich bei 2 der 52 Sequenzen konnten fehlende Schritte am Beginn beziehungsweise am Ende der Sequenz festgestellt werden.

### 4.4.3 Merkmalsextraktion

Bei der Merkmalsextraktion konnten im Rahmen der Experimente Erkenntnisse über die minimale und auch die maximale Dauer von Sequenzen gewonnen werden. Ausschlaggebend ist dabei im Wesentlichen die Anzahl der Schritte, die in diesen Sequenzen erfasst werden kann. Beispielsweise konnte anhand der Experimente ermittelt werden, dass bei den gelaufenen Geschwindigkeiten ($< 1\,\frac{\mathrm{m}}{\mathrm{s}}$) meist etwas mehr als 3 s ($\approx$ 90-100 Frames) notwendig sind, um mindestens zwei Schritte pro Seite zu erfassen. Einige Beispiele für die im Rahmen der Experimente ermittelten Zeitverläufe finden sich im Anhang B.1 ab Seite 101.

### 4.4.4 Gangbasierte Wiedererkennung

Aufbauend auf den bisher beschriebenen Vorleistungen, wird nun noch eine kurze Evaluation bezüglich der Leistungsfähigkeit der gangbasierten Wiedererkennung durchgeführt. Da der Datensatz mit acht Personen einen relativ geringen Umfang hat, werden für die Versuche keine Algorithmen gewählt, die zunächst einen Trainingsschritt benötigen. Alle acht Personen werden für den Testdatensatz genutzt. Grundlage der folgenden Untersuchungen ist daher ein Szenario, bei dem ein Deskriptor pro Person die Galerie bildet und unter den verbleibenden ein weiterer zufällig gewählt und in die Probe aufgenommen wird (*Single-vs-Single*). Um den Einfluss der zufällig getroffenen Auswahl zu minimieren, wird das Experiment $n = 100$ mal wiederholt und anschließend der Mittelwert über alle dabei ermittelten Leistungsmaße angegeben.
In einem ersten Versuch wird die Leistungsfähigkeit der Deskriptoren bewertet, die sich aus der zeitlichen Normierung der Hüft- und Kniewinkel ergeben. Die Merkmalsvektoren der rechten und linken Seite wurden dazu verkettet. Anschließend wird die Wiedererkennungsrate zwischen Hüfte und Knie einzeln bewertet. Eine Beschreibung der dazu angewandten Cumulative Matching Characterisik beziehungsweise der Synthetic Recogition Rate findet sich im Anhang (CMC - Seite 96, SRR - Seite 97).
Die Ergebnisse der Experimente die in Abbildung 4.13 graphisch dargestellt sind legen nahe, dass sich für das betrachtete Szenario der zeitliche Verlauf der Kniewinkel durchweg besser für die Wiedererkennung von Personen eignet. Im Vergleich zu den



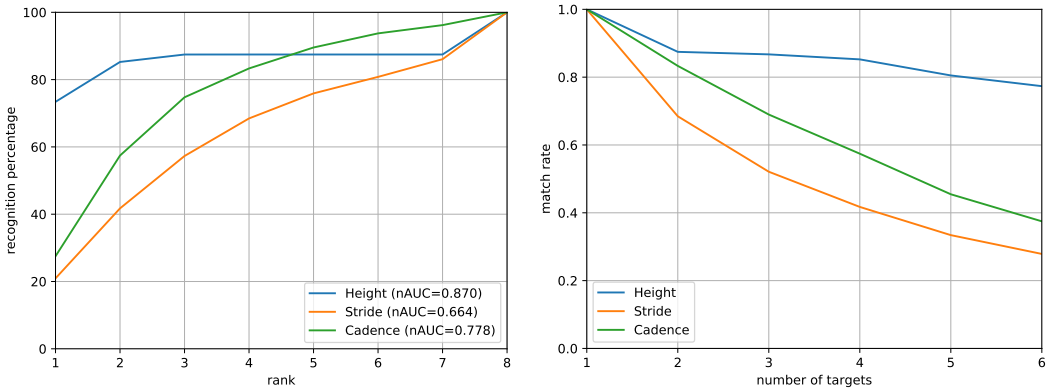

(a) Rangstatistik (CMC)

(b) Synthetic Recognition Rate

**Abbildung 4.13:** Wiedererkennungsleistung basierend auf den Zeitverläufen *(a) CMC-Rangstatistik (Single-vs-Single, n=100) für den vollen Datensatz mit 8 Personen. Die Zahl in Klammern gibt jeweils die normierte Fläche unter der Kurve* ($0 < nAUC \leq 1$) *an. (b) Synthetic Recognition Rate abgeleitet aus der CMC-Kurve.*

skalaren Kenngrößen (Körpergröße, Schrittweite, Kadenz), deren Ergebnisse für die gleichen experimentellen Ablauf in Abbildung 4.14 dargestellt sind, ordnet es sich bezüglich seiner Rang-1-Wiedererkennungsrate zwischen den beiden Schrittparametern (Schrittweite und Kadenz) und der Körpergröße ein. Wie erwartet erweist sich die Körpergröße als gutes Kriterium um Personen in einer kleinen Gruppe wiederzuerkennen, da es sich um ein leicht zu extrahierendes und sehr charakteristisches Merkmal handelt. Allerdings zeigt der Verlauf der CMC-Kurve in Abbildung 4.14(a) Schwächen jenseits von Rang 4. Hier ist der Kurvenverlauf beinahe waagrecht. Schrittweite und Kadenz zeigen sich in der Untersuchung als schwächste Merkmale, wohl auch weil keine explizite Normierung bezüglich der Laufgeschwindigkeit beim Merkmalsdesign berücksichtigt wurde.

Eine Möglichkeit die Wiedererkennungsleistung zu verbessern, bieten Verfahren aus der Information-Fusion (siehe Anhang, Seite 98). Exemplarisch wird im Folgenden die Rank-Level-Fusion des erfolgversprechendsten Merkmals aus den beiden Kategorien (zeitliche Verlauf der Kniegelenkwinkel und Körpergröße) untersucht. Das Ergebnis ist in Abbildung 4.15 und zeigt eine nochmals verbesserte Wiederkennungscharakteristik. Die ermittelte Rang-1-Wiedererkennungsrate von 73,2% ist weitestgehend identisch zur Wiedererkennung allein anhand der Höhe, jedoch kann der weitgehend waagerechte



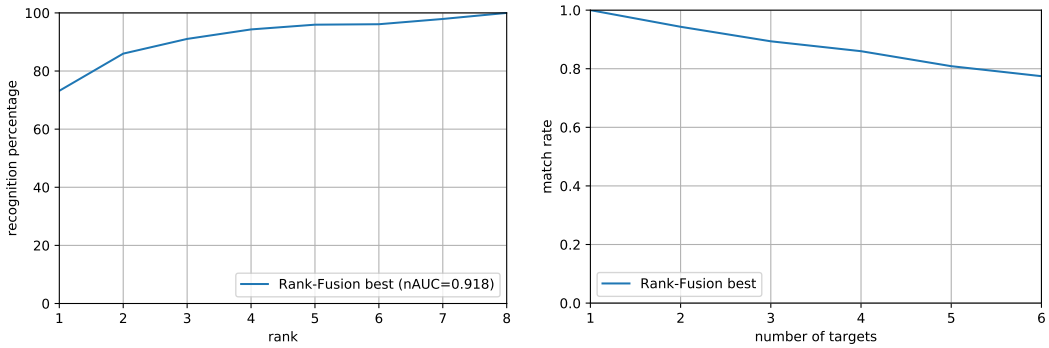

(a) Rangstatistik (CMC)

(b) Synthetic Recognition Rate

**Abbildung 4.14:** Wiedererkennungsleistung der skalaren Merkmale
*(a) CMC-Rangstatistik (Single-vs-Single, n=100) für den vollen Datensatz mit 8 Personen. Die Zahl in Klammern gibt jeweils die normierte Fläche unter der Kurve ($0 < nAUC \leq 1$) an. (b) Synthetic Recognition Rate abgeleitet aus der CMC-Kurve.*

Verlauf ab Rang 3 sichtbar angehoben werden. Diese Verbesserung zeigt sich auch in einer verbesserten erwarteten Match Rate in der SRR (Abbildung 4.15(b)) für zwei und drei Personen.

Zur besseren Vergleichbarkeit wird die Rangcharakteristik der untersuchten Merkmale und der Rank-Level-Fusion im Folgenden nochmals tabellarisch angegeben.




(a) Rangstatistik (CMC) (b) Synthetic Recognition Rate

**Abbildung 4.15:** Wiedererkennungsleistung Rank-Level-Fusion
*(a) CMC-Rangstatistik (Single-vs-Single, n=100) für den vollen Datensatz mit 8 Personen. Die Zahl in Klammern gibt jeweils die normierte Fläche unter der Kurve ($0 < nAUC \leq 1$) an. (b) Synthetic Recognition Rate abgeleitet aus der CMC-Kurve.*

| Merkmal | Rang 1 | Rang 5 |
|---|---|---|
| Hüftwinkel | 48,86% | 83,11% |
| Kniewinkel | 55,13% | 89,08% |
| Schrittweite | 20,94% | 75,90% |
| Kadenz | 27,51% | 89,54% |
| Körpergröße | 73,39% | 87,46% |
| Körpergröße+ Kniewinkel (Rank-Fusion) | 73,32% | 95,95% |

**Tabelle 4.2:** Übersicht über die Rangstatistik (CMC)

## 4.5 Zusammenfassung und Ausblick

Ausgehend vom State-of-the-Art konnte in diesem Kapitel die grundlegende Realisierung einer gangbasierten Wiedererkennung auf einer mobilen Roboterplattform erarbeitet und ihre Nutzbarkeit in einem eingeschränkten Versuch prinzipiell nachgewiesen werden. Anhand der betrachteten Einzelfeatures kann eine Person mit einer Wahr-

scheinlichkeit von bis zu 77% aus sechs Personen wiedererkannt werden. Exemplarisch wurde durch die Rank-Level-Fusion der zwei besten Merkmale aus den einzelnen Kategorien (zeitlicher Verlauf des Kniewinkels über den Gangzyklus und Körpergröße) ein möglicher Einsatz von Information-Fusion-Verfahren skizziert. Durch diesen einfachen, unparametrischen Ansatz lies sich die Wiederkennungswahrscheinlichkeit für zwei bis drei Personen um bis zu 7-Prozentpunkte verbessern.

Um jedoch die zu erwartende Leistungsfähigkeit im geplanten Einsatzszenario belastbar bewerten zu können, ist die Erstellung eines umfangreicheren Datensatzes unumgänglich. Neben einem größeren Umfang, werden sich vor allem Daten aus dem realen Einsatzfeld unter Einbeziehung der späteren Zielgruppe als hilfreich bei der Analyse erweisen. Anhand dieser Daten können dann auch weiterführende Wiedererkennungsverfahren etwa aus dem Bereich der Score-Level-Fusion oder des Metric Learning parametriert und evaluiert werden. Beispielsweise konnte in [Eisenbach et al., 2016][4] durch den Einsatz von Metric-Learning-Verfahren die Wahrscheinlichkeit einer erfolgreichen kleidungsbasierten Wiedererkennung aus bis zu sechs Personen in einem klinischen Einsatzfeld von 85% auf mehr als 95% zu steigern.

[4] enthält eigene Beiträge

# Kapitel 5

# Zusammenfassung und Ausblick

Zum Ende der Masterarbeit soll nun im Folgenden noch einmal eine kurze Zusammenfassung der vorgestellten Inhalte und Verfahren präsentiert werden. Als Ausblick wird eine mögliche Integration der beschriebenen Verfahren in die robotische Gesamtapplikation des ROGER-Projekts beschrieben.

## 5.1 Zusammenfassung

Ausgehend von der Beschreibung des angedachten Einsatzszenarios in Kapitel 1 und der Roboterplattform Ringo sowie der im ROGER-Projekt für die Ganganalyse eingesetzten Kinect2 in Kapitel 2, wurde zu Beginn von Kapitel 3 zunächst der Einsatz der Schwenk-Neige-Einheit (PTU) im Projektkontext motiviert. Nach einigen grundlegenden Aspekten sowie der Vorstellung der State-of-the-Art konnte die vertikale und horizontale Steuerung der PTU abgeleitet werden. Das Ziel der dabei entwickelten Verfahren war es, die Kinect2 geeignet zu positionieren, so dass eine optimale Ausnutzung des Sensorbereichs beispielsweise für die Ganganalyse realisiert wird. In der anschließenden Evaluation der Verfahren in einem der realen Einsatzumgebung nachempfundenen Szenario, konnten signifikante Verbesserungen gegenüber einem starren Kameraaufbau nachgewiesen werden. So konnte die Person beispielsweise in bis zu 99 von 100 betrachteten Fällen im anfangs definierten Zielsensorbereich erfasst werden.

Dies entspricht einer Verbesserung von rund 25% gegenüber einem starren Kameraaufbau.

Anschließend an diesen Aspekt der Arbeit, wurde die gangbasierte Wiedererkennung als weiterer möglicher Anwendungsfall für die durch die Kinect2 erfassten Tiefen- beziehungsweise Skelettdaten in Kapitel 4 motiviert. Nach einer umfangreichen Analyse des State-of-the-Art zur gangbasierten Wiedererkennung wurde ein prototypischer Ansatz entwickelt, anhand dessen eine Realisierung auf einer mobilen Roboterplattform möglich ist. Basierend auf einem eingeschränkten Szenario wurde die generelle Anwendbarkeit der gangbasierten Wiedererkennung untersucht und prinzipiell nachgewiesen. Durch die Anwendung von Rank-Level-Fusion konnte der Ansatz auf einfache Weise erweitert und die Wiedererkennungsleistung partiell weiter ausgebaut werden. Mit einer Wiedererkennungswahrscheinlichkeit von rund 77% bei einer Auswahl aus bis zu sechs Personen, ergibt sich ein solider Ausgangspunkt für weitere, anwendungsspezifische Untersuchungen.

## 5.2 Ausblick

Neben den bereits in den entsprechenden Kapiteln besprochenen Anknüpfungspunkten, wie beispielsweise einer verbesserten Lösung zum Skletttracking auf der mobilen Plattform oder dem Bedarf nach einer szenariospezifischen Evaluation der Wiedererkennungsleistung, besteht noch die Möglichkeit die Integration der Masterarbeit in die robotische Gesamtapplikation zu betrachten. Unter die Gesamtapplikation fallen alle Aspekte der robotischen Anwendung im Einsatzszenario. Die darin zusammengefassten Verfahren erstrecken sich über das gesamte Spektrum der sozialen Assistenzrobotik wie etwa Lokalisation, Navigation, Mapping, Mensch-Maschine-Interaktion, Personentracking/-wiedererkennung bis hin zur ROGER-spezifischen Ganganalyse.

Um die Komplexitutät zu reduzieren wird dabei die Wiedererkennung von Personen im Zusammenspiel mit dem Tracker [Wengefeld et al., 2016] als herausgelöste Komponente betrachtet und eine mögliche Integration einer gangbasierten Wiedererkennung schematisch erläutert.

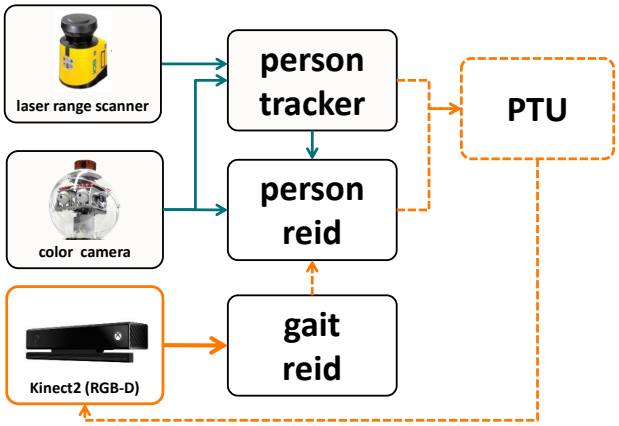



**Abbildung 5.1:** Integration der Masterarbeit in die Gesamtapplikation
*Dargestellt sind die bereits in die Applikation integrierten Laser-Scanner und Farbkameras des Roboters. Deren Daten werden zum Tracking und Wiedererkennen von Personen genutzt. Dabei kann die Wiedererkennung auch auf Daten aus dem Tracker zurückgreifen [*Wengefeld *et al., 2016]. Die gangbasierte Wiedererkennung basierend auf den Daten der Kinect2 wird als zusätzliche Komponente integriert, deren Informationen ebenfalls für die Wiedererkennung von Personen genutzt werden können. Die kombinierte Information aller Systeme kann dann genutzt werden um die PTU mit der Kinect2 zu positionieren.*

Eine Integrationsmöglichkeit ist in Abbildung 5.1 dargestellt. Dabei wird die gangbasierte Wiedererkennung als separate Komponente realisiert und zusätzlich zur bestehenden kleidungsbasierten Wiedererkennung und dem Persontracking in die Applikation eingebunden. Durch die parallele Nutzung beider Wiedererkennungsverfahren lassen sich deren Vorteile kombinieren und Nachteile ausgleichen. Ein entscheidender Nachteil der gangbasierten Wiedererkennung ist beispielsweise die Einschränkung auf bewegte Personen, die zur Erkennung außerdem einige Sekunden lang kontinuierlich beobachtet werden müssen. Diese Einschränkung gilt jedoch nicht für die ansichtsbasierte Wiedererkennung, anhand derer sich auch eine one-shot-Wiedererkennung realisieren lässt. In anderen Situationen in denen das Aussehen des Patienten beispielsweise durch wechselnde Beleuchtung stark variiert oder mehrere ähnlich gekleidete Perso-

nen sichtbar sind, kann die Analyse des Gangmusters als zusätzlicher Aspekt mit in die Erkennungsentscheidung einbezogen werden. Die kombinierte Aussagekraft von gangbasierter und kleidungsbasierter Wiedererkennung im Zusammenspiel mit dem multimodalen Personentracking bietet dann eine robuste Erkennungsleistung um beispielsweise die zuverlässige Ansteuerung der schwenkbar montierten Kinect2 für die Ganganalyse zu ermöglichen.

# Anhang A

# Ergänzungen

## A.1 Arkustangens mit zwei Argumenten

Die Definition des Arkustanges mit zwei Argumenten lautet wie folgt:

$$\text{arctan2} : \mathbb{R}^2 \to \left]-\pi, \pi\right]$$

mit

$$\text{arctan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{für } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{für } x < 0, y > 0 \\ \pi & \text{für } x < 0, y = 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{für } x < 0, y < 0 \\ +\frac{\pi}{2} & \text{für } x = 0, y > 0 \\ -\frac{\pi}{2} & \text{für } x = 0, y < 0 \\ 0 & \text{für } x = 0, y = 0 \end{cases}$$

Die genaue Definition kann je nach Implementierung variieren. Gleiches gilt für die Reihenfolge der Argumente. Die hier verwendete Definition ist etwa in Python und C++ umgesetzt.

## A.2 Auswertung des Skeletttrackings

Die folgende Auswertung des Skeletttrackings wird ergänzend zur experimentellen Bewertung der Kamerasteuerung im Kapitel zur stabilen Erfassung von Personen im Abschnitt *Experimente* (ab Seite 45) durchgeführt und ist nicht Kernbestandteil der Masterarbeit. Ferner sei an dieser Stelle darauf hingewiesen, dass dieser Aspekt nicht explizit in das Design der Experimente einbezogen wurde und die Aussagen im Folgenden stellenweise nicht auf einer breiten Datenbasis aufbauen.

Bei der Auswertung der Trackingqualität zeigen sich deutlich die Grenzen des für den stationären Betrieb entworfenen Kinect2-SDK-Skeletttrackers. Die folgenden Ausführungen konzentrieren sich im Wesentlichen auf die Zeitdauer über die ein kontinuierliches Skeletttracking des Nutzer erfolgt ist. Eine ausführliche Auswertung des Skeletttrackings bezüglich Genauigkeit ist kein primärer Bestandteil der Masterarbeit und daher auch nicht das Ziel dieses kurzen Abschnitts.

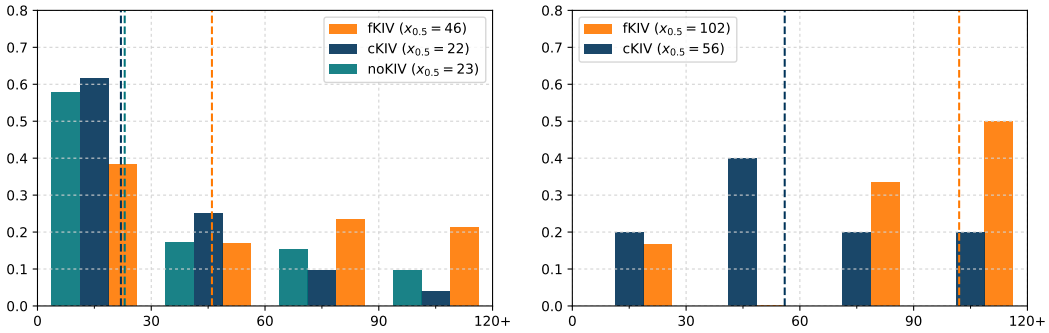
Die Analyse des Datenmaterials zeigt, dass unabhängig von der Ansteuerung häufig kein Skeletttracking während der Fahrt durchgeführt werden kann. Diese Tatsache zeigt sich deutlich, wenn der Anteil der Frames betrachtet wird, in denen das Skelett erfolgreich geschätzt wird. Aufgeschlüsselt nach den einzelnen Steuervarianten ergibt sich folgendes Ergebnis:

- ohne KIV: 31,09% (-/-%)
- cKIV: 31,11% (48,32%)
- fKIV: 42,59% (65,84%)

In Klammern angegeben ist der Anteil für die vier Zielanfahrten ohne Lotsen. Die hohe Diskrepanz der Anteile zwischen Losten und Zielanfahrten sich lässt unter Umständen damit erklären, dass die gefahrenen Trajektorien ohne aktive Lotsen-Funktion in der Regel glatter sind und die Fahrt mit einer gleichmäßigeren Geschwindigkeit erfolgt. Diese Tatsache führt dazu, dass eine erneute Schätzung der Bodenebene für den Skeletttracker durch den in der Kinect2 verbauten Beschleunigungssensor seltener ausgelöst wird. Während dieses Update-Prozesses erfolgt keine Schätzung von Nutzerposen durch den Algorithmus aus dem SDK. Ausgehend davon ist eine Bewertung der

einzelnen Verfahren nach diesem Kriterium nur unter Vorbehalt möglich. Die frontale Ansteuerung scheint jedoch in einigen Fällen eine bessere Ausgangslage zu bilden.

Neben der reinen Anzahl an Detektion bietet es sich außerdem an, die Länge der Abschnitte zu betrachten über die eine kontinuierliche Skelettschätzung möglich war. Im Rahmen dieser Auswertung wird ein Track als *kontinuierlich* angesehen, wenn die Lücke zwischen zwei aufeinanderfolgenden Skelettdetektionen weniger als 15 Frames (entspricht etwa 0,5 s bei 30 FPS) beträgt. Auf Basis dieser Festlegung ergeben sich 52 Sequenzen für den starren Kameraaufbau, 52 (10) für die zentrierte Ansteuerung und 47 (6) für die möglichst frontale Ansteuerung. In Klammern wird wiederum die Anzahl für die Zielanfahrten allein angegeben.




(a) alle Fahrten (b) nur Lotsen

**Abbildung A.1:** Vergleich der Tracklänge nach Verfahren

*Dargestellt wird die relative Häufigkeit für Abschnitte deren Länge zwischen den Werten der Achsenmarkierungen liegen. Die Einteilung erfolgt dabei sehr grob in Abschnitte von jeweils 30 Samples. Bei maximaler Framerate von* 30 *FPS, entspricht ein Abschnitt etwa einer Sekunde. Der vierte Abschnitt enthält alle Sequenzen, deren Länge größer als 90 Samples ist. Zusätzlich ist jeweils der Median markiert.*

Wie bereits bei der Bewertung der Einzeldetektionen zeigt sich bei der graphischen Auswertung in Abbildung A.1 eine deutliche Diskrepanz zwischen Lotsen und Zielanfahrt. Um die Vergleichbarkeit zu verbessern, werden die Kennwerte aus dem Histogramm im Folgenden noch einmal tabellarisch dargestellt.

| Verfahren | Median | $\Delta t \leq 1s$ | $1s < \Delta t \leq 2s$ | $2s < \Delta t \leq 3s$ | $\Delta t > 3s$ |
|---|---|---|---|---|---|
| noKIV | 23 | 57,7% | 17,3% | 15,4% | 9,62% |
| cKIV | 22 (56) | 61,5% (20,0%) | 25,0% (40,0%) | 9,6% (20,0%) | 3,8% (20,0%) |
| fKIV | 46 (102) | 38,3% (16,7%) | 17,0% (0,0%) | 23,4% (33,3%) | 21,3% (50,0%) |

**Tabelle A.1:** numerische Darstellung der Sequenzlängen je Verfahren
*In Klammern wird jeweils, sofern ermittelbar, der Anteil für die Zielanfahrten angegeben. Bei der Berechnung der Zeitdauer wird eine konstante Rate von* 30 *FPS angenommen.*

Allerdings bleibt auch weiterhin der Eindruck bestehen, dass die frontale Ansicht über die PTU eine bessere Ausgangslage bildet, da vor allem sehr kurze Tracks von bis zu einer Sekunde bis zu 50% seltener zu beobachten sind. Auch längere Tracks von mindestens 3 s Dauer werden in beiden Ausgangssituationen beinahe doppelt so häufig ermittelt. Der nichtdeterministische Einfluss durch das Roboterfahrverhalten auf den SDK-Skeletttracker kann allerdings auch hier nicht ausgeschlossen werden und zeigt bereits bei der anfangs genannten Anzahl der ermittelbaren Teilsequenzen sein vermeintliches Gewicht.

## A.3 Box-Plot

Der Box-Plot (Beispiel in Abbildung A.2) ist eine in der Statistik häufig eingesetzte Möglichkeit, statistische Kennwerte unimodal verteilter numerischer Daten visuell zu vergleichen.

Die Elemente des Plots basieren im Wesentlichen auf einer empirischen Quantilanalyse der Daten. Das empirische $p$-Quantil, mit $p \in (0, 1)$, einer Stichprobe beschreibt vereinfacht ausgedrückt den kleinstmöglichen Wert aus der Stichprobe, der größer als $100p\%$ aller Daten ist. Für häufig auftretende Werte haben sich außerdem Eigennamen eingebürgert. So bezeichnet man beispielsweise das 0,5-Quantil üblicherweise als Median und die Quantilwerte für $p = 0{,}25$ beziehungsweise $p = 0{,}75$ als unteres bezie-

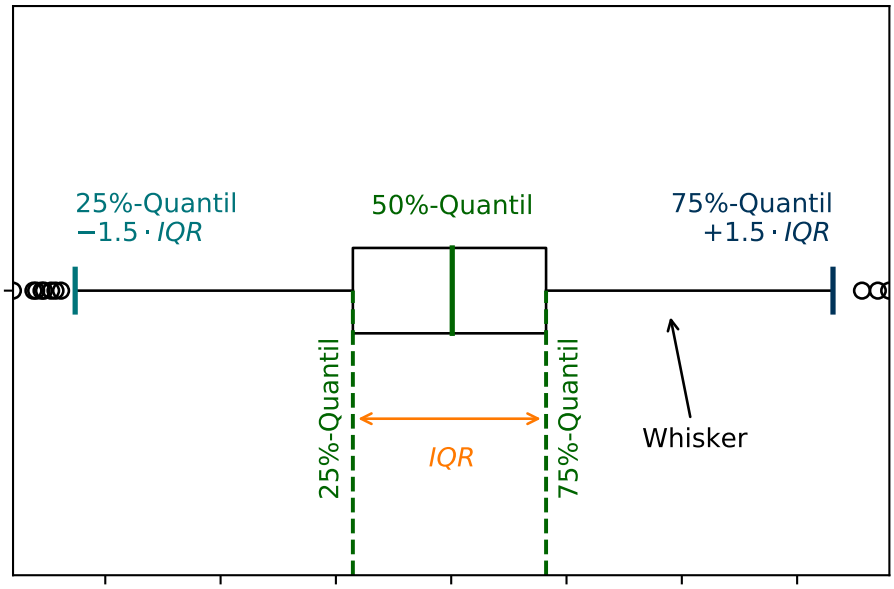



**Abbildung A.2:** Beispiel für einen Box-Plot

*Box-Plot über eine Stichprobe einer standardnormalverteilten Zufallsvariable mit 1000 Datenpunkten. Die Ausreißer werden durch Kreise angezeigt*

hungsweise oberes Quartil. Vor allem in der englischsprachigen Literatur finden sich auch die Kurzschreibweisen $Q1$ und $Q3$ für die beiden genannten Quartile.
Ausgehend davon ist der sogenannte *Interquartilabstand* (kurz: *IQR* von ***i**nter**q**uartil **r**ange*) als $IQR := Q3 - Q1$ definiert. Er gilt als Maß für die Streuung der mittleren 50% der Daten. Der IQR wird im Box-Plot als Rechteck markiert. Zusätzlich wird in diesem Rechteck meist noch der Median eingezeichnet. Ausgehend vom IQR wird ein Bereich definiert der sich zwischen $Q1 - 1{,}5 \cdot IQR$ und $Q3 + 1{,}5 \cdot IQR$ erstreckt[1]. Ausdehnung und Grenzen dieses Bereichs werden durch die sogenannten *Whisker* markiert. Datenpunkte außerhalb dieses Bereiches werden in der Regel als *Ausreißer* bezeichnet und zusätzlich durch frei wählbare Symbole repräsentiert oder weggelassen. Liegt keiner der Datenpunkte außerhalb Bereichsgrenzen, wird die Whisker-Endmarkierung entsprechend dem Minimum beziehungsweise Maximum der Stichprobe gewählt.

[1] auch andere Grenzen möglich

## A.4 Definitionen - Personenwiedererkennung

Im Folgenden wird noch die Definition der Begriffe *Merkmalsraum* und *Metrik* unter dem Gesichtspunkt der Personenwiederkennung angelehnt an [Vorndran, 2015] angegeben. Hierbei handelt es sich um wichtige Grundbegriffe, die jedoch im Kontext der Masterarbeit keine hervorgehobene Bedeutung haben.

### A.4.1 Merkmalsraum

Ein Merkmalsraum $\mathcal{X}$ beschreibt einen Vektorraum, in dem alle möglichen Vektorrepräsentationen einer bestimmten Merkmalscodierung liegen. Es handelt sich dabei in der Regel um einen Unterraum des $\mathbb{R}^n$, wobei $n$ der Merkmalsdimension entspricht [Weisstein, 2015].

Für normierte Histogrammmerkmale beschreibt der Merkmalsraum einen Ausschnitt aus der Oberfläche einer Hyperkugel ($l_2$-Normierung) beziehungsweise eines Hyperquaders ($l_1$-Normierung). Das Zentrum der gedachten Hyperkörper liegt dabei in $\underline{\mathbf{0}} \in \mathbb{R}^n$. Für normierte zweidimensionale Vektoren $\underline{\mathbf{x}} \in \mathbb{R}^2$ lassen sich die entsprechenden Merkmalsräume leicht geometrisch interpretieren (Abbildung A.3).

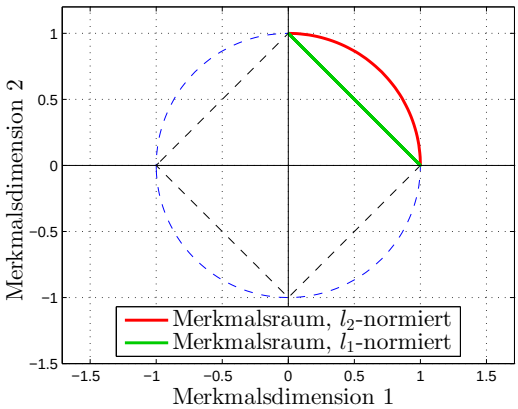



**Abbildung A.3:** Beispiel für einen Merkmalsraum im $\mathbb{R}^2$

*Der Merkmalsraum, der durch $l_2$-normierte Histogramme im $\mathbb{R}^2$ aufgespannt wird beschreibt einen Viertelkreis mit Radius 1. Bei einer $l_1$-Normierung bildet er eine Kante eines um 45° gekippten Quadrats mit Kantenlänge $\sqrt{2}$.*

Merkmalevektoren aus dem Kontext der kleidungsbasierten Personenwiedererkennung decken ein breites Spektrum an Dimensionen ab. Sie finden sich in den meisten Fällen im Bereich $20 \leq n \leq 10000$. Für hochdimensionale Merkmalsvektoren werden oft Techniken zur Dimensionsreduktion eingesetzt, um die Komplexität des aufgespannten Raums beherrschbar zu machen und den Berechnungsaufwand oder den Speicherbedarf zu senken.
Wird eine Sequenz von Beobachtung als Basis für die Merkmalsberechnung gewählt, variiert die Dimensionalität in einem noch weiteren Bereich. Eine triviale Codierung einer Sequenz von $k$ Beobachtungen, die jeweils mit einem $n$-dimensionalen Deskriptor beschrieben werden, führt beispielsweise zu einem Vektor $\underline{\mathbf{x}} \in \mathbb{R}^{k \cdot n}$ (alternativ $\underline{\mathbf{x}} \in \mathbb{R}^{k \times n}$).

### A.4.2 Metrik

Eine Funktion $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ in einem Merkmalsraum $\mathcal{X}$, wird in der Personenwiedererkennung *Metrik* genannt, wenn sie die folgenden Kriterien nach [Deza und Deza, 2006] für alle Merkmalsvektoren $\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j, \underline{\mathbf{x}}_k \in \mathcal{X}$ erfüllt:

$$d(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j) \geq 0 \qquad \text{(Nicht-Negativität)} \tag{A.1a}$$

$$d(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j) = 0 \Leftrightarrow \underline{\mathbf{x}}_i = \underline{\mathbf{x}}_j \qquad \text{(Axiom der Selbstähnlichkeit)} \tag{A.1b}$$

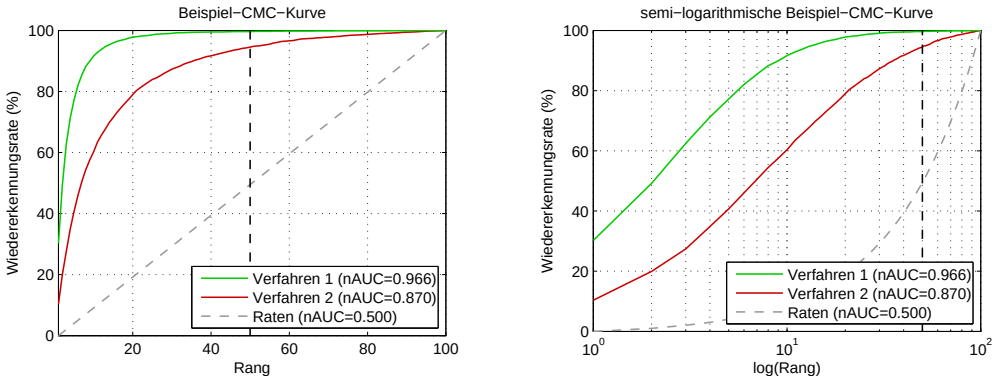
$$d(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j) = d(\underline{\mathbf{x}}_j, \underline{\mathbf{x}}_i) \qquad \text{(Symmetrie)} \tag{A.1c}$$

$$d(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j) \leq d(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_k) + d(\underline{\mathbf{x}}_k, \underline{\mathbf{x}}_j) \qquad \text{(Dreiecksungleichung)} \tag{A.1d}$$

Weniger formal bedeuten diese Kriterien, dass der Unterschied beziehungsweise die Distanz zwischen zwei Merkmalsvektoren anhand einer Metrik durch einen Funktionswert (Score) $\geq 0$ beschrieben wird. In der Regel erfüllen gängige Vektornormen, wie etwa die Euklidische Distanz, die Manhattan-Distanz oder Histogrammvergleichsmaße, beispielsweise die Bhattacharyya-Distanz, diese Bedingungen. Eine umfangreiche und systematische Übersicht, die sich mit verschiedenen Histogrammvergleichsmaße beschäftigt, findet sich in der Arbeit von [Cha, 2008]. Alle genannten Beispiele finden daher auch Anwendung beim Vergleich zwischen Elementen der Galerie $\mathcal{G}$ und der Probe $\mathcal{P}$ (Abschnitt 4.1.1).

## A.5 Cumulative Matching Characteristic

Für die Leistungsbewertung von Personenwiedererkennungssystemen wird für abgeschlossene Szenarien (Abschnitt 4.1.2, Seite 59) die sogenannte Cumulative Matching Characteristic eingesetzt. Die Zuordnung der wiederzuerkennenden Personen (Probe) zu einer Person der Galerie erfolgt in diesem Szenario basierend auf einer sortierten Reihenfolge der durch die angewendete Metrik zu den Galeriebildern ermittelten Distanzwerte. Anschließend wird basierend auf diesem Ranking eine Statistik erstellt. Betrachtet man den Fall bei dem jede Person nur ein Galeriebild hat, wird für jedes Bild der Probe anhand von Labelinformationen ermittelt, an welcher Stelle im Ranking das zugehörige Galeriebild steht. Dieser Prozess wird für alle Bilder der Probe wiederholt und die Häufigkeiten der Positionen erfasst. Nach einer entsprechenden Normierung erhält man so eine Wahrscheinlichkeitsdichteverteilung die angibt, wie hoch die Wahrscheinlichkeit ist an einer bestimmten Stelle der Rangfolge das korrekte Ergebnis der Wiedererkennung zu finden. Da diese absolute Position selten von Interesse ist ermittelt man im nächsten Schritt die kumulierte Wahrscheinlichkeit für einen Rang indem alle Wahrscheinlichkeiten bis zu diesem Rang aufsummiert werden. Der akkumulierte Wert beschreibt nun die Wahrscheinlichkeit das korrekte Ergebnis zu finden wenn man alle Kandidaten bis zum entsprechenden Rang berücksichtigt. Dieser Wert wird auch Wiedererkennungsrate genannt. Die graphische Darstellung wird meist kurz als CMC-Kurve bezeichnet und trägt die Wiedererkennungsrate über dem Rang auf. Wird der Fokus vor allem auf die vorderen Ränge ($\leq 10$) gelegt, wird der Rang mit einer logarithmischen Achsenteilung dargestellt. Häufig wird neben der Wiedererkennungsrate an bestimmten Positionen auch die normalisierte Fläche unter der CMC-Kurve (nAUC) als Performancemaß angeben. Der Wert der nAUC liegt zwischen 0 und 1, wobei 0.5 der Wiedererkennung durch Raten entspricht. Wünschenswert sind daher Verfahren die Werte nahe 1 liefern.




(a) linearer Rang (b) logarithmischer Rang

**Abbildung A.4:** Beispiel für eine CMC- und eine logCMC-Kurve
*CMC-Kurven für zwei fiktive Verfahren bei einer Galeriegröße von 100 Personen. Verfahren 1 ist dabei auf allen Rängen besser als Verfahren 2 und hat demzufolge auch eine größere Fläche unter der Kurve. Betrachtet man die einzelnen Ränge würde Verfahren 1 in rund* 30% *der Fälle an erster Stelle das korrekte Ergebnis liefern, Verfahren 2 in rund* 10% *der Fälle. Betrachtet man die ersten zehn Ränge ergibt sich eine Wiedererkennungsrate von etwa* 90% *für das 1., respektive* 60% *für das 2. Verfahren.*

## A.6 Synthetic Recognition Rate

Bei der *Synthetic Recognition Rate* (kurz: *SRR*) handelt es sich um ein Gütemaß, dass aus der Cumulative Matching Characteristic abgeleitet werden kann. Algebraisch lässt sich der Verlauf der SRR wie folgt beschreiben:

$$SRR(M) = CMC(N/M)$$

$M \in \mathbb{N}$ sei dabei die Stelle der SRR-Kurve, die berechnet werden soll, $CMC(k)$ die CMC an für den Rang $k$ und $N \in \mathbb{N}$ der Umfang des Benchmarkdatensatzes für welchen die CMC ermittelt wurde. Für nicht ganzzahlige Verhältnisse $N/M$, wird üblicherweise eine geeignete Interpolation zwischen den nächsten Rängen der CMC-Kurve angewandt. Interpretieren lässt sich der Wert der SRR-Kurve an einer Stelle

$k \in \mathbb{N}$ dabei als Wahrscheinlichkeit, mit der $k$ der $N$ Individuen aus dem betrachteten Benchmarkdatensatz gegenübergestellt und wiedererkannt werden sollen.

## A.7 Übersicht Information Fusion

Unter Begriff *Information Fusion* fällt eine große Familie von Verfahren, welche sich mit der geeigneten Kombination von Informationen zur Verbesserung der Klassifikations- und Wiedererkennungsleistung beschäftigt. Die Reichweite der verfolgten Ansätze ist sehr umfassend, im folgenden soll daher nur ein Überblick über die Hauptaspekte der Verfahren gegeben werden, damit eine Einordnung der im Rahmen der Masterarbeit eingesetzten Verfahren möglich ist. Eine umfangreiche Übersicht findet sich etwa bei [Beyerer et al., 2006].

Der Kern aller Verfahren ist, dass Informationen verschiedener Quellen geeignet kombiniert werden, um die Klassifikations- oder Wiedererkennungsleistung zu steigern. Ohne Beschränkung der Allgemeinheit, sei im folgenden nur noch von der Wiedererkennung die Rede. Die Kombination der Informationen kann dabei auf verschiedenen Abstraktionsleveln stattfinden:

- Signal Level: Kombination von Rohdaten, beispielsweise Bilder zweier Kameras
- Feature Level: Kombination der Merkmalsvektoren die aus den Rohdaten extrahiert werden
- Score Level: (gewichtetes) Aufsummieren der Scores nach dem Matching verschiedener Merkmale
- Rank Level: Fusion der sortierten Listen (Rankings) ausgehend vom Matching
- Decision Level: Kombination der finalen Entscheidung mehrerer Klassifikatoren

Signal-Level-Fusion beschreibt die unterste Abstraktionsebene, in der Signale unmittelbar nach der Erfassung durch entsprechende Sensoren miteinander kombiniert werden. In der Regel bietet sich dieser Ansatz ausschließlich für gleichartige Sensoren (gleicher Sensortyp, gleiche Modalität, Beobachtung der gleichen Szene) an. Signal-Level-Fusion soll für den betrachteten Anwendungsfall nicht explizit untersucht, da nur Aufnahmen einer Tiefenkamera genutzt werden.

Auf dem nächsthöheren Abstraktionsniveau befindet sich die Feature-Level-Fusion. In der Regel werden dabei ausgehend von den Rohsignalen (auch verschiedenartiger Sensoren), Merkmalsvektoren ermittelt, gegebenenfalls normiert und meist durch Konkatenation in einen neuen Deskriptor umgeformt. Häufig haben die dabei entstandenen Merkmalsvektoren eine hohe Dimensionalität, so dass oft Verfahren zur Dimensionsreduktion eingesetzt werden, um die Komplexität zu senken. Ferner ist dabei zu beachten, dass die Merkmalsvektoren zur gleichen Domäne gehören, beispielsweise dass alle Teilmerkmale aus normierten Histogrammen bestehen. Dadurch lassen sich ungleichmäßige Einflüsse der verschiedenen Komponenten beim Einsatz einfacher Metrikfunktionen (euklidische Distanz, Manhattan-Distanz) vermeiden. Im betrachteten Anwendungsfall lassen sich etwa die zeitlichen Verläufe der rechten und linken Seite sowie gegebenenfalls auch die zeitlichen Verläufe der Hüft- und Kniewinkel auf diese Weise fusionieren.

Im Gegensatz zur eben betrachteten Feature-Level-Fusion, werden bei der Score-Level-Fusion nicht die Merkmalsvektoren selbst, sondern die Scores aus dem Vergleich mit dem gleichen Merkmalsvektor einer anderen Person kombiniert. Bevor die Einzelscores, meist gewichtet, aufsummiert werden, erfolgt in der Regel eine Score-Normierung auf einen einheitlichen Wertebereich. Eine Übersicht über verschiedene lineare und nicht-lineare Normierungsansätze sowie ein Verfahren zur Ermittlung der Gewichte findet sich in [Eisenbach et al., 2015a][2]. Einfache aber weitverbreitete Ansätze zur Normierung sind dabei etwa die Min-Max-Normierung oder eine Normierung auf Mittelwertfreiheit und gleiche Varianz. Zur Bestimmung der Normierungsparameter sind in der Regel Trainingsdaten erforderlich, daher wird im Rahmen der Experimente zunächst nicht auf diesen Ansatz zurückgegriffen.

Auch die Fusion der anschließend auf den Scores berechnet Rangfolge ist möglich. Dieser Ansatz wird Rank-Level-Fusion genannt. Ein Algorithmus der hierbei angewendet wird ist der sogenannte *Borda-Count*. Dabei wird jedem Rang eine Punktzahl zugeordnet. Anschließend wird für jedes Objekt im Ranking die Gesamtpunktzahl als Summe über alle Rankings berechnet. Basierend auf dieser Gesamtpunktzahl wird dann ein

[2] enthält eigene Beiträge

neues Ranking erstellt. Gegebenenfalls ist dazu noch eine Regel notwendig, welche bei gleicher Punktzahl die finale Reihenfolge festlegt. Da die Rankings zunächst pro Matchingfunktion getrennt ausgeführt werden, ist keine Normierung der Scores notwendig. Das Verfahren kann also ad-hoc ohne Trainingsdaten zur Laufzeit angewendet werden. Auf dem höchsten Abstraktionsniveau befindet sich die sogenannte Decision-Level-Fusion. Hierbei werden die Ergebnisse verschiedener Klassifikatoren kombiniert. Häufig wird die finale Klassifikation basierend auf einer (gewichteten) Mehrheitsentscheidung realisiert. Da es sich bei betrachteten Wiedererkennungssituation jedoch um ein Mehrklassenproblem handelt, kann dieser Ansatz hier nicht angewendet werden.

# Anhang B

# Visualisierungen

## B.1 Dynamische Merkmale - gangbasierte Wiedererkennung

Im Folgenden werden beispielhaft einige Verläufe der im Rahmen der Experimente zur gangbasierten Wiedererkennung (Seite 77) extrahierten Hüft- und Kniewinkeltrajektorien dargestellt. Die Trajektorien des linken Knies beziehungsweise der linken Hüfte werden stets in grün, dargestellt. Die der rechten Seite in rot. In der Darstellung der Sequenzen sind die Segmentierungspunkte entsprechend ihrer Klassifizierung Cyan (Schritt links) und Magenta (Schritt rechts) eingefärbt. Im Hintergrund der zeitlich normierten Verläufe auf der rechten Seite der Plots, wird die Normwertkurve (Mittelwert: unterbrochene schwarze Linie, Standardabweichung: grau markierter Bereich) aus dem Ganglabor des Waldkrankenhaus „Rudolf Elle“ Eisenberg zu Referenzzwecken ebenfalls angezeigt. Beide Verläufe einer Abbildung sind dabei der gleichen Person zugeordnet. Bei Abbildung B.2 handelt es sich um zeitliche Verläufe, die für eine der beiden Personen mit Unterarmgehstützen ermittelt wurden.

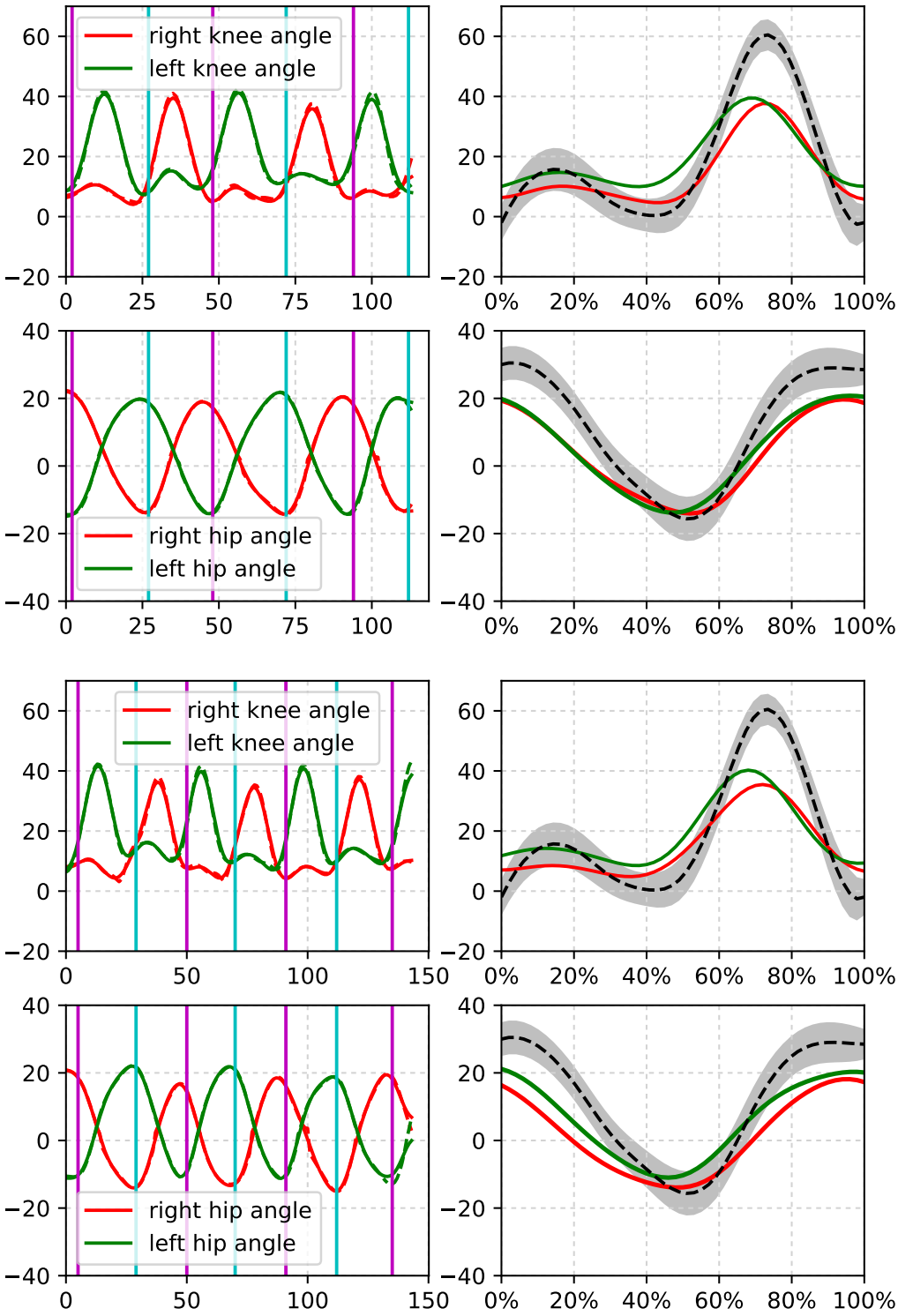



**Abbildung B.1:** Zeitverläufe für Knie- und Hüftwinkel - Beispiel 1

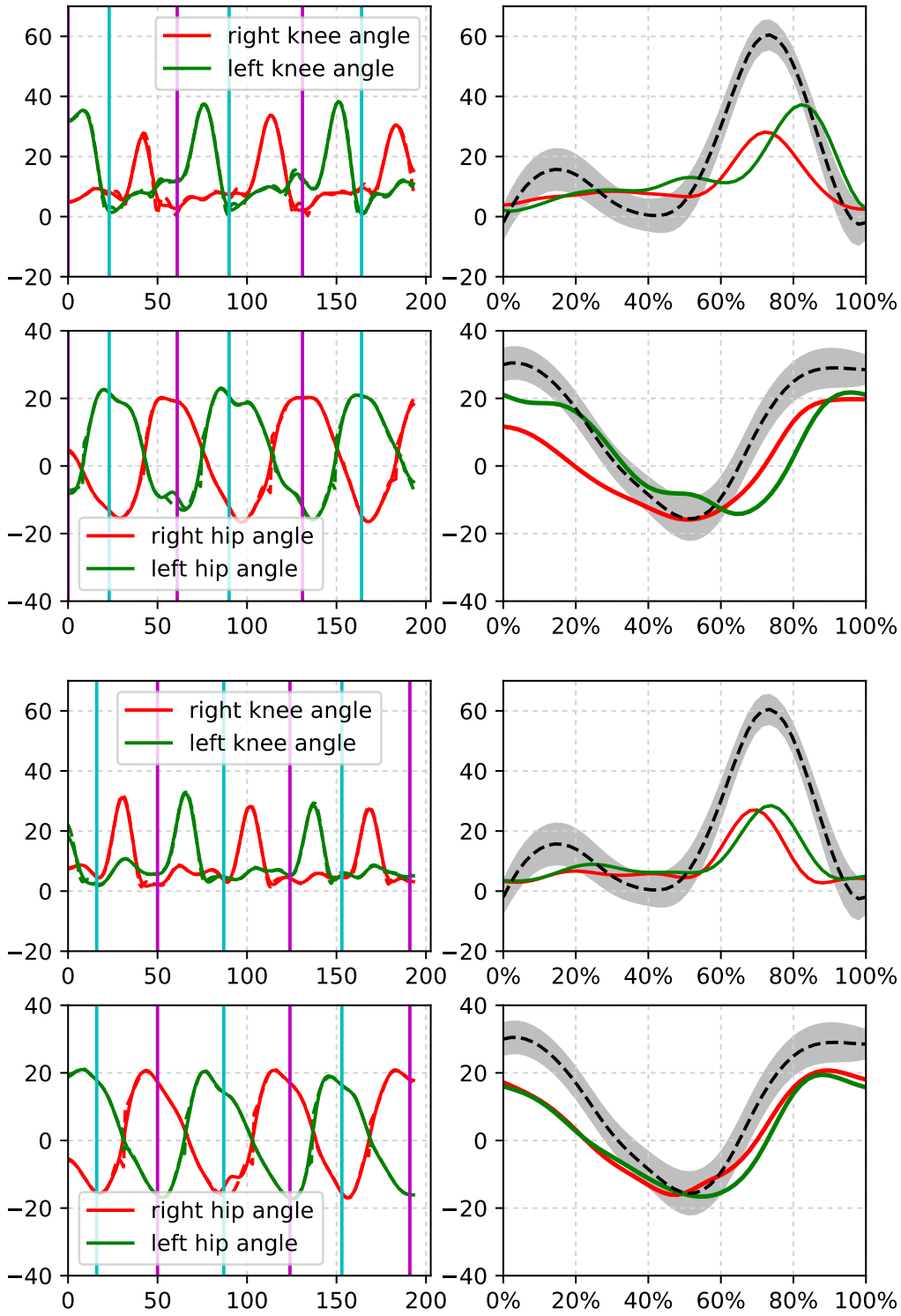



**Abbildung B.2:** Zeitverläufe für Knie- und Hüftwinkel - Beispiel 2

# Abbildungsverzeichnis

1.1 Anmeldung und Training mit dem Roboter . . . . . 3
1.2 Beispiel für die Einsatzumgebung . . . . . 5
1.3 Sensor-Sichtfeld beim Ausweichen . . . . . 6

2.1 Roboterplattform „Ringo“ mit Verkleidung . . . . . 10
2.2 Kinect2-Sensor . . . . . 12
2.3 Kinect2-SDK-Skelettpunkte . . . . . 14
2.4 Sensorbereich des Kinect2-Tiefensensors und Zielbereich . . . . . 15

3.1 Ringo ohne Gehäuse mit PTU und Kinect2 . . . . . 18
3.2 2D-Beispiel für Koordinatensysteme und -transformation . . . . . 22
3.3 Grundlagen Regelkreis und Regler . . . . . 23
3.4 Visual-Servoing eines Pioneer 3D-DX mit ViSP . . . . . 26
3.5 Visual-Servoing-Setup bei [Bahn et al., 2011] . . . . . 27
3.6 Visual-Servoing-Setup in [Tinh et al., 2014] . . . . . 28
3.7 Humanoide Roboter mit Visual Servoing . . . . . 29
3.8 lokale Koordinatensysteme von PTU und Kinect2 . . . . . 32
3.9 Veranschaulichung der Koordinatensysteme am Roboter . . . . . 33
3.10 Geometrische Modellierung der PTU-Neigung . . . . . 34
3.11 Optimaler Neigungswinkel . . . . . 37
3.12 Darstellung verschiedener Ansteuerungen . . . . . 38
3.13 Berechnung des Schwenk-Winkels - frontale Ansicht . . . . . 39
3.14 Grenzen für die frontale Ansicht . . . . . 40
3.15 Berechnung des Schwenk-Winkels - frontale Ansicht . . . . . 41
3.16 Veranschaulichung zur Berechnung der Frontalansicht . . . . . 42

3.17 erweiterter PD-Regler - Blockschaltbild . . . . . 43
3.18 Einbeziehung der Robotertrajektorie in die Ansteuerung . . . . . 45
3.19 grober Aufbau der Teststrecke . . . . . 46
3.20 Auswertungsschema der Position im Sensorbereich . . . . . 47
3.21 graphische Auswertung der Position im Sensorerfassungsbereich . . . . 48
3.22 AR-Marker zur Schätzung der Oberkörperorientierung . . . . . 50
3.23 graphische Auswertung der Oberkörperorientierung . . . . . 51

4.1 1. Schritt: Aufbau der Galerie . . . . . 57
4.2 2. Schritt: Wiedererkennung . . . . . 58
4.3 Beispiele für Gait Energy Images . . . . . 63
4.4 Beispiele für Gait Energy Volumes . . . . . 64
4.5 Wiedererkennung anhand von Gelenkpunkttrajektorien nach [Tanawongsuwan und Bobick, 2001] . . . . . 66
4.6 Gelenkpunktwinkeltrajektorien nach [Yam et al., 2004] . . . . . 67
4.7 Ablauf der Wiedererkennung nach [Ahmed et al., 2015] . . . . . 69
4.8 Bezeichnung der Kinect2-Skelettpunkte . . . . . 72
4.9 Geschwindigkeitsberechnung mit Rauschunterdrückung . . . . . 73
4.10 Die menschliche Laufbewegung . . . . . 75
4.11 Double-Support-Phasen-Erkennung . . . . . 76
4.12 Aufnahmestrecke gangbasierte Wiedererkennung . . . . . 78
4.13 Wiedererkennungsleistung basierend auf den Zeitverläufen . . . . . 81
4.14 Wiedererkennungsleistung der skalaren Merkmale . . . . . 82
4.15 Wiedererkennungsleistung Rank-Level-Fusion . . . . . 83

5.1 Integration der Masterarbeit in die Gesamtapplikation . . . . . 87

A.1 Vergleich der Tracklänge nach Verfahren . . . . . 91
A.2 Beispiel für einen Box-Plot . . . . . 93
A.3 Beispiel für einen Merkmalsraum im $\mathbb{R}^2$ . . . . . 94
A.4 Beispiel für eine CMC- und eine logCMC-Kurve . . . . . 97

B.1 Zeitverläufe für Knie- und Hüftwinkel - Beispiel 1 . . . . . 102

B.2 Zeitverläufe für Knie- und Hüftwinkel - Beispiel 2 . . . . . 103

# Literaturverzeichnis

[Agin, 1980] Agin, Gerald J (1980). *Computer Vision Systems for Industrial Inspection and Assembly.*. IEEE Computer, 13(5):11–20.

[Ahmed et al., 2015] Ahmed, Faisal, P. P. Paul und M. L. Gavrilova (2015). *DTW-based kernel and rank-level fusion for 3D gait recognition using Kinect*. The Visual Computer, 31(6-8):915–924.

[Ament, 2013] Ament, Christoph (2013). *Skript zur Vorlesung Regelungs- und Systemtechnik 1*. Vorlesungsskript, TU Ilmenau, Fachgebiet Systemanalyse. Ilmenau.

[Anscombe und Glynn, 1983] Anscombe, F. J. und W. J. Glynn (1983). *Distribution of the Kurtosis Statistic b2 for Normal Samples*. Biometrika, 70(1):227.

[Åström und Hägglund, 1989] Åström, Karl Johan und T. Hägglund (1989). *Integrator Windup and How to Avoid It*. In: *American Control Conference*, S. 1693–1698.

[Åström und Rundqwist, 2006] Åström, Karl Johan und L. Rundqwist (2006). *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society.

[Bahn et al., 2011] Bahn, Wook, J. Park, C.-h. Lee, T.-i. Kim, T. Lee, M. M. Shaikh, K.-s. Kim und D.-i. Cho (2011). *A Motion-Information-Based Vision-Tracking System with a Stereo Camera on Mobile Robots*. In: *International Conference on Robotics, Automation and Mechatronics (RAM)*, S. 252–257. IEEE.

[BenAbdelkader et al., 2002] BenAbdelkader, Chiraz, R. Cutler und L. Davis (2002). *Stride and Cadence as a Biometric in Automatic Person Identification and Verification.* In: *IEEE International Conference on Automatic Face Gesture Recognition (FGR)*, S. 372–377, Washington, DC. IEEE.

[BenAbdelkader et al., 2001] BenAbdelkader, Chiraz, R. Cutler, H. Nanda und L. Davis (2001). *EigenGait: Motion-Based Recognition of People Using Image Self-Similarity.* In: *International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, S. 284–294. Springer, Berlin, Heidelberg.

[Beyerer et al., 2006] Beyerer, Jürgen, F. Puente Leon und K.-D. Sommer, Hrsg. (2006). *Informationsfusion in der Mess- und Sensortechnik*, Eisenach. Universitätsverlag Karlsruhe, Karlsruhe.

[Bobick und Johnson, 2001] Bobick, Aaron und A. Johnson (2001). *Gait Recognition Using Static, Activity-Specific Parameters.* In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Bd. 1, S. 423–430. IEEE Comput. Soc.

[Brent, 2002] Brent, Richard P. (2002). *Algorithms for Minimization Without Derivatives.* Dover Publications, Mineola, New York, 2. Aufl.

[Cha, 2008] Cha, Sung-Hyuk (2008). *Taxonomy of Nominal Type Histogram Distance Measures.* In: *Proceedings of the American Conference on Applied Mathematics (MATH)*, S. 325–330. World Scientific and Engineering Academy and Society (WSEAS).

[Chaumette und Hutchinson, 2006] Chaumette, François und S. Hutchinson (2006). *Visual Servo Control Part I: Basic Approaches.* IEEE Robotics and Automation Magazine (RAM), 13(4):82–90.

[Chen et al., 2016] Chen, Wenzheng, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or und B. Chen (2016). *Synthesizing Training Images*

*for Boosting Human 3D Pose Estimation*. In: *International Conference on 3D Vision (3DV)*, S. 479–488. IEEE.

[Chen und Lee, 1992] Chen, Zen und H.-J. Lee (1992). *Knowledge-Guided Visual Perception of 3-D Human Gait from a Single Image Sequence*. Transactions on Systems, Man, and Cybernetics (SMC), 22(2):336–342.

[Claudio et al., 2016] Claudio, Giovanni, F. Spindler und F. Chaumette (2016). *Vision-Based Manipulation with the Humanoid Robot Romeo*. In: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, S. 286–293. IEEE.

[Deza und Deza, 2006] Deza, Michel-Marie und E. Deza (2006). *Dictionary of distances*, Kap. General Definitions. Elsevier.

[Directed Perception, 2006] Directed Perception (2006). *Computer Controlled PAN-TILT UNIT Model PTU-D46*. Online-Handbuch (FLIR Legacy Products). `http://www.flir.com/uploadedFiles/Directed_Perception/Documents/Products/PTU-D46/PTU-manual-d46.pdf`, Stand: 22.05.2017.

[Einhorn et al., 2012] Einhorn, Erik, T. Langner, R. Stricker, Ch. Martin und H.-M. Gross (2012). *MIRA - Middleware for Robotic Applications*. In: *International Conference on Intelligent Robots and Systems (IROS)*, S. 2591–2598.

[Eisenbach et al., 2015a] Eisenbach, Markus, A. Kolarow, A. Vorndran, J. Niebling und H.-M. Gross (2015a). *Evaluation of Multi Feature Fusion at Score-Level for Appearance-based Person Re-Identification*. In: *Int. Joint Conf. on Neural Networks (IJCNN)*, S. 469–476. IEEE.

[Eisenbach et al., 2016] Eisenbach, Markus, D. Seichter, T. Wengefeld und H.-M. Gross (2016). *Cooperative Multi-Scale Convolutional Neural Networks for Person Detection*. In: *International Joint Conference on Neural Networks (IJCNN)*, S. 267–276, Vancouver, BC, Canada. IEEE.

[Eisenbach et al., 2015b] Eisenbach, Markus, A. Vorndran, S. Sorge und H.-M. Gross (2015b). *User Recognition for Guiding and Following People with a*

*Mobile Robot in a Clinical Environment*. In: *International Conference on Intelligent Robots and Systems (IROS)*, S. 3600–3607.

[Fankhauser et al., 2015] Fankhauser, Péter, M. Bloesch, D. Rodriguez, , R. Kaestner, M. Hutter und R. Siegwart (2015). *Kinect v2 for Mobile Robot Navigation: Evaluation and Modeling*. In: *IEEE International Conference on Advanced Robotics (ICAR)*, S. 388–394.

[Feddema und Mitchell, 1989] Feddema, John T. und O. R. Mitchell (1989). *Vision-Guided Servoing with Feature-based Trajectory Generation (for robots)*. Transactions on Robotics and Automation (TRA), 5(5):691–700.

[FLIR Motion Control Systems, Inc, 2013] FLIR Motion Control Systems, Inc (2013). *Model PTU-D46 Configuration Guide*. Online-Handbuch (FLIR Legacy Products). `http://www.flir.com/uploadedFiles/CVS_Americas/MCS/Products/PTU-D46_47/PTU-D46-Config-Guide.pdf`, Stand: 22.05.2017.

[Föllinger, 2016] Föllinger, Otto (2016). *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*. VDE Verlag, 12. Aufl.

[Garrido-Jurado et al., 2014] Garrido-Jurado, Sergio, R. Muñoz-Salinas, F. Madrid-Cuevas und M. Marín-Jiménez (2014). *Automatic generation and detection of highly reliable fiducial markers under occlusion*. Pattern Recognition, 47(6):2280–2292.

[Gianaria et al., 2014] Gianaria, Elena, M. Grangetto, M. Lucenteforte und N. Balossino (2014). *Human Classification Using Gait Features*. In: *International Workshop on Biometrics*, S. 16–27. Springer International Publishing.

[Gross et al., 2016] Gross, Horst-Michael, A. Scheidig, M. Eisenbach, Th. Q. Trinh und T. Wengefeld (2016). *Assistenzrobotik für die Gesundheitsassistenz - ein Beitrag zur Evaluierung der Praxistauglichkeit am Beispiel eines mobilen Reha-Roboters*. In: *German AAL Conference (AAL)*, S. 58–67. VDE.

[Gross und Shi, 2001] Gross, Ralph und J. Shi (2001). *The CMU Motion of Body (MoBo) Database.* Technischer Bericht, Robotics Institute, Pittsburgh, PA.

[Han und Bhanu, 2006] Han, Ju und B. Bhanu (2006). *Individual Recognition Using Gait Energy Image.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 28(2):316–322.

[Handrich und Al-Hamadi, 2013] Handrich, Sebastian und A. Al-Hamadi (2013). *A Robust Method for Human Pose Estimation Based on Geodesic Distance Features.* In: *International Conference on Systems, Man, and Cybernetics (SMC)*, S. 906–911, Manchester, UK. IEEE.

[Hogg, 1983] Hogg, David (1983). *Model-Based Vision: a Program to See a Walking Person.* Image and Vision Computing, 1(1):5–20.

[Jain et al., 2004] Jain, Anil K., S. C. Dass und K. Nandakumar (2004). *Soft Biometric Traits for Personal Recognition Systems.* In: Zhang, David und A. K. Jain, Hrsg.: *International Conference on Biometric Authentication (ICBA)*, S. 731–738, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Jäschke, 2017] Jäschke, Bianca (2017). *Machbarkeitsuntersuchung zur Ganganalyse im klinischen Umfeld mittels eines mobilen Trainingsroboters und 3D-Tiefenkameras.* Masterarbeit - Fachgebiet Neuroinformatik und Kognitive Robotik, TU Ilmenau. Ilmenau.

[Jones et al., 2001] Jones, Eric, T. Oliphant, P. Peterson et al. (2001). *SciPy: Open source scientific tools for Python.* `http://www.scipy.org/`, Stand: 25.04.2017.

[Kwolek et al., 2014] Kwolek, Bogdan, T. Krzeszowski, A. Michalczuk und H. Josinski (2014). *3D Gait Recognition Using Spatio-Temporal Motion Descriptors.* In: *Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, S. 595–604. Springer, Cham.

[Levene, 1960] Levene, Howard (1960). *Robust tests for equality of variances.* Contributions to Probability and Statistics, 1:278–292.

[Little und Boyd, 1998] Little, James J und J. E. Boyd (1998). *Recognizing People by Their Gait: The Shape of Motion Recognizing People by Their Gait: The Shape of Motion.* Videre: Journal of Computer Vision Research (JCVR), 1.

[Lunze, 2010] Lunze, Jan (2010). *Regelungstechnik 1.* Springer-Lehrbuch. Springer Berlin Heidelberg, Berlin, Heidelberg, 8. Aufl.

[Marchand et al., 2005] Marchand, E., F. Spindler und F. Chaumette (2005). *ViSP for Visual Servoing: A Generic Software Platform with a Wide Class of Robot Control Skills.* IEEE Robotics and Automation Magazine (RAM), 12(4):40–52.

[MathWorks, 2017] MathWorks (2017). *PID Tuner.* Online-Dokumentation. `https://de.mathworks.com/help/control/ref/pidtuner-app.html`, Stand: 20.05.2017.

[Metta et al., 2010] Metta, Giorgio, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. Von Hofsten, K. Rosander, M. Lopes und J. Santos-Victor (2010). *The iCub Humanoid Robot: An Ppen-Systems Platform for Research in Cognitive Development.* Neural Networks, 23(8):1125–1134.

[Microsoft Corporation, 2014a] Microsoft Corporation (2014a). *Coordinate mapping.* Windows Dev Center. `https://msdn.microsoft.com/de-de/library/dn785530.aspx`, Stand: 23.05.2017.

[Microsoft Corporation, 2014b] Microsoft Corporation (2014b). *JointType Enumeration.* Windows Dev Center. `https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx`, Stand: 24.04.2017.

[Microsoft Corporation, 2014c] Microsoft Corporation (2014c). *Kinect-Hardware.* Windows Dev Center. `https://developer.microsoft.com/de-de/windows/kinect/hardware`, Stand: 20.04.2017.

[MIRA Reference Documentation, 2017] MIRA Reference Documentation (2017). *Transformation Framework*. Online-Referenzdokumentation. `http://www.mira-project.org/MIRA-doc/TransformationFrameworkPage.html`, Stand: 23.05.2017.

[Nixon und Carter, 2006] Nixon, Mark S. und J. N. Carter (2006). *Automatic Recognition by Gait*. Proceedings of the IEEE, 94(11):2013–2024.

[Niyogi und Adelson, 1994] Niyogi, Sourabh A. und E. H. Adelson (1994). *Analyzing and Recognizing Walking Figures in XYT*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 469–474. IEEE Comput. Soc. Press.

[Pöhlmann et al., 2016] Pöhlmann, Stefanie T. L., E. F. Harkness, C. J. Taylor und S. M. Astley (2016). *Evaluation of Kinect 3D Sensor for Healthcare Imaging*. Journal of Medical and Biological Engineering (JMBE), 36(6):857–870.

[Preis et al., 2012] Preis, Johannes, M. Kessel, M. Werner und C. Linnhoff-Popien (2012). *Gait Recognition with Kinect*. Workshop on Kinect in Pervasive Computing.

[Quigley et al., 2009] Quigley, Morgan, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler und A. Y. Ng (2009). *ROS: An Open-Source Robot Operating System*. In: *ICRA Workshop on Open Source Software*, Bd. 3, S. 5.

[Rohr, 1993] Rohr, Karl (1993). *Incremental Recognition of Pedestrians from Image Sequences*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 8–13. IEEE Comput. Soc. Press.

[Roncone et al., 2014] Roncone, Alessandro, U. Pattacini, G. Metta und L. Natale (2014). *Gaze Stabilization for Humanoid Robots: A Comprehensive Framework*. In: *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, S. 259–264. IEEE.

[Sarkar et al., 2005] Sarkar, Sudeep, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother und K. W. Bowyer (2005). *The Human ID Gait Challenge Problem: Data Sets, Performance, and Analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 27(2):162–177.

[Scheidig et al., 2015] Scheidig, Andrea, E. Einhorn, Ch. Weinrich, M. Eisenbach, St. Mueller, Th. Schmiedel, T. Wengefeld, Th. Q. Trinh, H.-M. Gross, A. Bley, R. Scheidig, G. Pfeiffer, S. Meyer und S. Oelkers (2015). *Robotischer Reha-Assistent zum Lauftraining von Patienten nach Schlaganfall: erste Ergebnisse zum Laufcoach*. In: *German AAL Conference (AAL)*, S. 436–445. VDE.

[Shotton et al., 2013] Shotton, Jamie, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman und A. Blake (2013). *Efficient Human Pose Estimation from Single Depth Images*. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 35(12):2821–2840.

[Sinha et al., 2013] Sinha, Aniruddha, K. Chakravarty und B. Bhowmick (2013). *Person Identification using Skeleton Information from Kinect*. International Conference on Advances in Computer-Human Interactions (ACHI), S. 101–108.

[Sivapalan et al., 2011] Sivapalan, Sabesan, D. Chen, S. Denman, S. Sridharan und C. Fookes (2011). *Gait Energy Volumes and Frontal Gait Recognition Using Depth Images*. In: *International Joint Conference on Biometrics (IJCB)*, S. 1–6. IEEE.

[Tanawongsuwan und Bobick, 2001] Tanawongsuwan, Rawesak und A. Bobick (2001). *Gait Recognition from Time-normalized Joint-angle Trajectories in the Walking Plane*. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Bd. 2, S. II–726–II–731. IEEE Comput. Soc.

[Thuilot et al., 2002] Thuilot, Benoit, P. Martinet, L. Cordesses und J. Gallice (2002). *Position Based Visual Servoing: Keeping the Object in the*

*Field of Vision.* In: *International Conference on Robotics and Automation (ICRA)*, Bd. 2, S. 1624–1629. IEEE.

[Tinh et al., 2014] Tinh, N. V., P. T. Cat, P. M. Tuan und T. T. Q. Bui (2014). *Visual Control of Integrated Mobile Robot-Pan Tilt-Camera System for Tracking a Moving Target.* In: *International Conference on Robotics and Biomimetics (RO-BIO)*, S. 1566–1571. IEEE.

[Trinh et al., 2015] Trinh, Than Quang, Ch. Schroeter und H.-M. Gross (2015). *"Go ahead, please": Recognition and Resolution of Conflict Situations in Narrow Passages for Polite Mobile Robot Navigation.* In: *International Conference on Social Robotics (ICSR)*, S. 643–653. Springer.

[Volkhardt et al., 2013] Volkhardt, Michael, Ch. Weinrich und H.-M. Gross (2013). *People Tracking on a Mobile Companion Robot.* In: *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, S. 4354–4359.

[Vorndran, 2015] Vorndran, Alexander (2015). *Evaluation von Distance-Metric-Learning für die Kleidungs-basierte Wiedererkennung von Personen im klinischen Einsatzfeld.* Bachelorarbeit - Fachgebiet Neuroinfomatik und Kognitive Robotik, TU Ilmenau. Ilmenau.

[Weinrich et al., 2012] Weinrich, Christoph, Ch. Vollmer und H.-M. Gross (2012). *Estimation of Human Upper Body Orientation for Mobile Robotics Using an SVM Decision Tree on Monocular Images.* In: *International Conference on Intelligent Robots and Systems (IROS)*, S. 2147–2152.

[Weinrich et al., 2014] Weinrich, Christoph, T. Wengefeld, Ch. Schroeter und H.-M. Gross (2014). *People detection and distinction of their walking aids in 2D laser range data based on generic distance-invariant features.* In: *IEEE International Symposium on Robot and Human Interactive (RO-MAN)*, S. 767–773.

[Weiss et al., 1987] Weiss, Lee, A. C. Sanderson und C. P. Neuman (1987). *Dynamic Sensor-Based Control of Robots with Visual Feedback.* Journal on Robotics and Automation (JRA), 3(5):404–417.

[Weisstein, 2015] Weisstein, Eric W. (2015). *Vector Space. From MathWorld–A Wolfram Web Resource.* `http://mathworld.wolfram.com/VectorSpace.html`.

[Welch, 1947] Welch, Bernard L. (1947). *The Generalization of 'Student's' Problem when Several Different Population Variances are Involved.* Biometrika, 34(1/2):28–35.

[Wengefeld et al., 2016] Wengefeld, Tim, M. Eisenbach, Th. Q. Trinh und H.-M. Gross (2016). *May I be your Personal Coach? Bringing Together Person Tracking and Visual Re-identification on a Mobile Robot.* In: *International Symposium on Robotics (ISR)*, S. 141–148. VDE.

[Wilson et al., 1996] Wilson, William J., C. C. Williams Hulls und G. S. Bell (1996). *Relative End-Effector Control using Cartesian Position Based Visual Servoing.* Transactions on Robotics and Automation (TRA), 12(5):684–696.

[Yam et al., 2004] Yam, Chew Yean, M. S. Nixon und J. N. Carter (2004). *Automated person recognition by walking and running via model-based approaches.* Pattern Recognition, 37(5):1057–1072.

[Ye et al., 2011] Ye, Mao, Xianwang Wang, R. Yang, Liu Ren und M. Pollefeys (2011). *Accurate 3D Pose Estimation From a Single Depth Image.* In: *International Conference on Computer Vision (ICCV)*, S. 731–738. IEEE.

[Yu et al., 2006] Yu, Shiqi, D. Tan und T. Tan (2006). *A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition.* In: *International Conference on Pattern Recognition (ICPR)*, Bd. 4, S. 441–444.