



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Marcus Rödiger

Interaktive Steuerung von Computersystemen mittels
Erkennung von Körpergesten

Marcus Rödiger

Interaktive Steuerung von Computersystemen mittels Erkennung
von Körpergesten

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai v. Luck
Zweitgutachter: Prof. Dr. Andreas Meisel

Abgegeben am 03. Mai 2010

Marcus Rödiger

Thema der Masterarbeit

Interaktive Steuerung von Computersystemen mittels Erkennung von Körpergesten

Stichworte

Motion Capturing; Gestenerkennung; Bildmustererkennung; Seamless Interaction; A.R.T. Tracker; HCI

Kurzzusammenfassung

Die vorliegende Arbeit beschreibt die Entwicklung einer Plattform für die Forschung im Bereich der interaktiven Steuerung von Computersystemen mit Hilfe von Handgesten. Zu diesem Zweck wird eine Applikation entwickelt und vorgestellt, die über Gesten gesteuert wird. Im Speziellen wird auf mögliche Verfahren und Techniken zur Erkennung von Körpergesten eingegangen. Das Softwaresystem läuft im Kontext eines Katastrophen-Rettungsleitstandes, als zentrales Datenanzeige- und Steuergerät der Leitstandsmitarbeiter.

Title of the paper

Interactive control of computer systems with pattern recognition of body gestures

Keywords

motion capturing; gestures recognition; pattern recognition; seamless interaction; A.R.T. Tracker; HCI

Abstract

This master thesis describes the evaluation of a research platform for interactiv control of computer systems with hand gesture. For this an application is designed and developped which is actuated by gestures. Especially we describe methods and technics of pattern recognition of body gestures. The softwaresystem is created in the context with an emergency control room as central data display and control unit for the emergency control room staff.

Inhaltsverzeichnis

1. Einleitung	7
1.1. Motivation	7
1.2. Zielsetzung	8
1.3. Gliederung	8
2. Grundlagen	10
2.1. Ubiquitous Computing	10
2.2. Seamless Interaction	10
2.3. Tracking von Körperbewegungen	12
2.3.1. Trackingsysteme	12
2.3.2. Freiheitsgrade	13
2.3.3. ART Tracker	14
2.4. Gesten	17
2.4.1. Begriffserklärung	17
2.4.2. Gesten als Kommunikationsmittel	18
2.4.3. Statische und dynamische Gesten	19
2.4.4. Start/Stop Erkennung von Gesten	20
2.4.5. Kettenbildung von Gesten	20
2.4.6. Attribute einer Geste	21
3. Analyse	22
3.1. Anwendungsszenario	22
3.2. Handgestensteuerung versus Maus/Tastatur	24
3.3. Funktionale Anforderungen	24
3.3.1. Scratch der Anwendung	24
3.3.2. Aufbau und Ablauf	25
3.3.3. Anforderungen an die Datenerfassung	27
3.3.4. Anwendungsfälle	28
3.4. Interaktionstechniken	28
3.4.1. Drag-and-Drop	29
3.4.2. Funktionsauswahl	30
3.4.3. Drehregler	31
3.4.4. Objektselektierung	32

3.4.5. Auswahl eines Bildbereiches	32
3.5. Erforderliche Gesten	33
3.6. Nicht funktionale Anforderungen	33
3.6.1. Einbeziehung der dritten Dimension	33
3.6.2. Interaktivität und Feedback	33
3.6.3. Sonstige nicht funktionale Anforderungen	34
3.7. Fazit	34
4. Design	35
4.1. Grundarchitektur	35
4.2. Architektur zur Erkennung von Gesten	36
4.2.1. Start/Stop-Erkennung	38
4.2.2. Datenerhebung	39
4.2.3. Merkmalsextraktion	43
4.2.4. Klassifikation	44
4.2.5. Anwendungsschnittstelle	50
4.3. Abschluss	50
5. Zusammenfassung	52
5.1. Fazit	52
5.2. Ausblick	53
Literaturverzeichnis	54
A. Gestenbeschreibung	57
A.1. Zeigergeste	57
A.2. Daumenschaltergeste	60
A.3. Finger zusammenführen	60
A.4. Richtungsbewegung	61
A.5. Fingerkreisbewegung	62

Abbildungsverzeichnis

2.1. Abbildung von reale auf virtuelle Objekte	11
2.2. Elektromechanisches Trackingsystem	12
2.3. Koordinatensystem 6DOF	14
2.4. Kameraanordnung im Labor	15
2.5. Body	16
2.6. OK Geste	19
3.1. Minority Report UI	23
3.2. Skizze Powerwall Katalogansicht	26
3.3. Skizze Powerwall Playerdarstellung	26
3.4. Koordinatensystem Datenerfassung	27
3.5. Drag'n'Drop Geste	29
3.6. Mögliche Icondarstellungen	30
3.7. Menü Funktionsauswahl	31
3.8. Greifhand	31
3.9. Auswahl eines Bildbereiches	32
4.1. Model-View-Controller-Pattern	35
4.2. Aufbau Controller	37
4.3. Architektur Gestenerkennung	38
4.4. 2D-Datenraum	41
4.5. Sensorhandschuh	42
4.6. Koordinatensystem des Bodys	43
4.7. Wahrscheinlichkeitsverteilung Gaußfunktion	45
4.8. Abbildung der Zeigergeste	46
4.9. Fuzzy-Funktion	47
4.10. Gauß-Funktion	47
4.11. Wahrscheinlichkeitsverteilung Zeigergeste	49
A.1. Abbildung der Zeigergeste	58
A.2. Treppenbildung	61

1. Einleitung

Ubiquitous Computing ist seit Jahren ein Schlagwort für die Zukunft von Computern. Die Anzahl von Computern in der Umgebung des Nutzers steigt stetig. Immer mehr und immer komplexere Aufgaben werden von Computern in unserem unmittelbaren Umfeld übernommen und sollen so unser Leben erleichtern. Die immer größer werdende Anzahl und Komplexität der Geräte, macht es jedoch erforderlich, dass die Interaktion mit diesen Geräten für den Menschen einfacher gemacht werden sollte. Forschungsgebiete der Seamless Interaction gewinnen dabei immer mehr an Wichtigkeit. Das Ziel ist dabei die Interaktion zwischen Mensch und Computer natürlich und intuitiv für den Benutzer erscheinen zu lassen. Der Benutzer stellt sich dabei nicht auf das Eingabeparadigma der Maschine ein, sondern das Computersystem passt sich an die vertraute und gewohnte Welt des Nutzers an. Dieser Bruch im bis jetzt vorherrschenden Nutzungskonzept mit Computern, ist Thema vieler Forschungsgebiete. In verschiedensten Bereichen wird nach neuen Möglichkeiten geforscht, auf natürliche Weise mit dem Computersystem zu interagieren. Es ist daher sinnvoll neue Interaktionskonzepte zu betrachten und Ihre Möglichkeiten und Wege der Umsetzbarkeit aufzuzeigen.

1.1. Motivation

Frederic Kaplan fragt in seiner Keynote bei der ICMI-MLMI¹ 2009, ob gestenbasierte Interfaces nicht die Zukunft der Mensch-Maschine-Interaktion darstellen. ([Kaplan \(2009\)](#)) Viele Punkte sprechen für eine Nutzung von Gesten als Interaktionsmedium. Gesten nutzt der Mensch bereits seit dem Anfang seiner Existenz als Mittel, um mit anderen zu kommunizieren. Dabei können Gesten die verbale Sprache unterstützen (z.B. Handzeichen zur Unterstreichung bestimmter Worte) oder sogar, wie es bei der Gebärdensprache der Fall ist, ganz ersetzen. Warum sollte diese Art der Interaktion nicht auch zwischen Mensch und Maschine genutzt werden? Viele Anwendungsfälle für die Steuerung mit Gesten sind denkbar, in welchem andere Eingabekanäle nur schlecht nutzbar sind. Vor allem die Nutzung von natürlichen Gesten, alleine oder in Kombination mit anderen Kommunikationskanälen, bieten das Potenzial, den Umgang mit Computersystemen durch den Nutzer auf eine ganz neue Ebene zu heben. Der

¹ICMI-MLMI: International Conference on Multimodal Interfaces and Machine Learning for Multimodal Interaction

Nutzer interagiert mit dem Computersystem, so wie er mit einem Menschen interagieren würde.

Viele Fragestellungen in diesem Feld der Benutzereingabe sind bis heute noch offen. Dazu zählen besonders die Nutzbarkeit in einem realistischen Umfeld, die Wahrnehmung und die Akzeptanz solcher Technik durch die unterschiedlichen Benutzerkreise.

Im Bereich der zweidimensionalen Gestensteuerung wird intensiv geforscht und es sind bereits praxistaugliche Geräte im Einsatz.² Die Forschung im Bereich der Interaktion mittels dreidimensionaler Gesten ist nicht so intensiv. Obwohl dreidimensionale Gesten viel eher der natürlichen Interaktion entsprechen als zweidimensionale, sind viele Fragestellungen in diesem Bereich noch ungeklärt. Ein Grund dafür ist, dass es an den nötigen Forschungssystemen fehlt, mit dessen Hilfe es möglich ist, Fragen aus dem Bereich der dreidimensionalen Gestensteuerung gezielt zu behandeln. Diese Arbeit soll die Grundlagen für die Entwicklung eines solchen Forschungssystems schaffen, um mit dessen Hilfe weitere Fragestellungen aus dem Bereich interaktiver, dreidimensionaler Gestensteuerung beantworten zu können.

1.2. Zielsetzung

Ziel dieser Arbeit ist es eine Plattform zu entwerfen, mit dessen Hilfe Fragestellungen aus dem Bereich „Gesten getriebener Mensch-Maschine-Interaktion“ bearbeitet werden können. Zu diesem Zweck wird eine Applikation entwickelt und vorgestellt, in welcher eine interaktive Steuerung mittels Körpergesten erfolgt. Die Anwendung setzt auf ein mögliches Anwenderszenario für Gestensteuerung auf, mit dessen Hilfe grundlegende Fragestellungen erörtert werden können. Der Fokus dieser Arbeit liegt auf der Entwicklung eines Systems, welches mindestens den Kommunikationskanal der Gestensteuerung nutzt und der Konzeption der für eine Gesteerkennung nötigen Verfahren.

1.3. Gliederung

Die Arbeit teilt sich in vier große Abschnitte:

Im Kapitel 2 werden die für diese Arbeit nötigen Grundlagen beschrieben. Es werden sowohl Techniken zum Tracking von Körperbewegungen vorgestellt, als auch der Begriff und die Eigenschaften von Gesten erläutert.

Im Kapitel 3 wird das zugrunde liegende Anwendungsszenario vorgestellt und die Anforderungen an die Applikation beschrieben.

²z.B. Multitouchgesten auf dem iPhone <http://www.apple.com>

Im Kapitel 4 werden konkrete Lösungen entwickelt, um die vorher gestellten Anforderungen umsetzen zu können. Insbesondere wird auf die Architektur und Technik zur Erkennung von Handgesten eingegangen.

Am Schluss werden in Kapitel 5 die Ergebnisse der Arbeit zusammengefasst und ein Ausblick gegeben. Der Anhang A enthält eine ausführliche Übersicht, der definierten Gesten und Ihrer jeweiligen Erkennungsstrategien.

2. Grundlagen

Ziel dieser Arbeit ist die Entwicklung eines Systems zur Steuerung von Computersystemen mittels Körpergesten. Für diesen Zweck sind einige Grundlagen nötig. Als erstes werden die Bereiche „Ubiquitous Computing“ und „Seamless Interaction“ näher erläutert. Danach werden Trackingsysteme zur Bewegungserfassung vorgestellt und klassifiziert. Anschließend wird der Begriff Geste definiert und mögliche Klassifikationen von Gesten vorgestellt.

2.1. Ubiquitous Computing

Unter dem, vor allem durch Mark Weisers Aufsatz „The Computer for the 21st Century“, geprägten Begriff Ubiquitous Computing oder auch allgegenwärtiges Rechnen, versteht man die Allgegenwertigkeit von Computern in unserer Umgebung ([Weiser \(1999\)](#)). Die damalige Vision sagte voraus, dass Computer in unserer Umgebung einen immer höheren Anteil einnehmen und uns immer mehr Aufgaben abnehmen werden. Dabei schwindet die Benutzerwahrnehmung der Computer in diesen Geräten. Computer werden in unserer Umgebung benutzt, ohne sie als solche unmittelbar wahrzunehmen.

Beispiele hierfür gibt es heutzutage bereits viele: Der Computer in der EC-Karte, im Kinderspielzeug oder in der Kaffeemaschine sind nur einige Dinge, welche wir täglich um uns haben und benutzen, ohne direkt auf ein Computersystem zu schließen.

Ein wichtiges Merkmal dieser Entwicklung ist die immer stärker steigende Vernetzung dieser Geräte untereinander, wodurch zum Beispiel gemeinsame Ressourcen und Informationen effizient genutzt werden können.

2.2. Seamless Interaction

Für Benutzer ist die Bedienung eines Computers besonders einfach, wenn er die Art, wie er mit dem System interagiert, bereits aus anderen Bereichen seines Lebens kennt. Daher sollte das Ziel sein, unter Zuhilfenahme von Fähigkeiten und Techniken welche dem Nutzer vertraut sind,

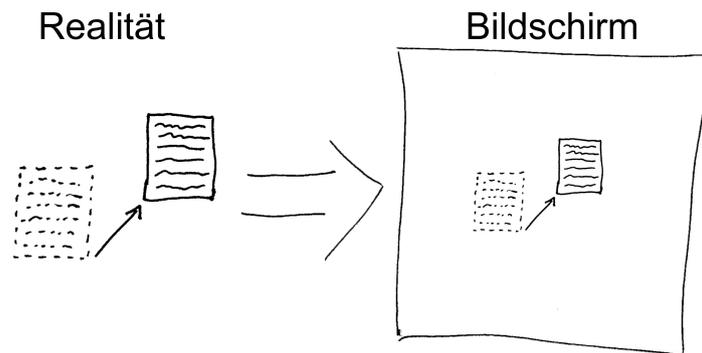


Abbildung 2.1.: Abbildung von reale auf virtuelle Objekte

mit dem Computer zu Interagieren. Dieses Konzept beschreibt „Seamless Interaction“. [Ishii u. a. \(1994\)](#) definiert zwei Punkte, die für „Seamless Interaction“ beachtet werden müssen.

Als erstes geht es um die Übertragbarkeit von realen in virtuelle Arbeitsabläufe. Besonders eindringlich sind Nutzungskonzepte für den Benutzer, wenn er sie bereits aus der realen Welt kennt. Ein Beispiel ist die Nutzung einer Papiermetapher für die Darstellung und Manipulation von Bildern auf dem Bildschirm. Das Objekt auf dem Bildschirm sieht aus und verhält sich dabei größtenteils wie ein Blatt Papier. Der Nutzer bildet sein bereits für reales Papier existierendes mentales Modell auf das virtuelle Papier ab ([Abbildung 2.1](#)). Dadurch kann der Benutzer, ihm bereits bekannte Arbeitsabläufe, wie das Verschieben oder Drehen, auf das virtuelle Objekt übertragen.

Als zweites geht es um den Wechsel zwischen unterschiedlichen Arbeitsmodi und -räumen. Dieses muss vor allem bei Gruppenarbeiten beachtet werden, bei denen ein ständiger Wechsel zwischen persönlichem und öffentlichem Arbeitsraum stattfindet.

Ziel bei Seamless Interaction ist es nicht unbedingt reale Gegebenheiten auf virtuelle Objekte „naturgetreu“ zu übertragen, sondern vielmehr das beim Benutzer vorhandene mentale Modell mit den Möglichkeiten des Computers zu kombinieren. Dadurch ist es möglich die Vorteile einer natürlichen Interaktion mit der Geschwindigkeit von Computersystemen zu kombinieren ([Roßberger \(2008\)](#)).

Vor allem dreidimensionale Gestensteuerung bietet hier den Vorteil, dass der Anwender mit der Nutzung diverser Gesten bereits aus der realen Welt vertraut ist und diese Art der Interaktion leicht übertragen kann. Dadurch ist es möglich, Computersysteme zu entwickeln, welche für den Benutzer auf einfache und natürliche Weise bedienbar sind.



Abbildung 2.2.: Elektromechanisches Trackingsystem³

2.3. Tracking von Körperbewegungen

Für die Realisierung der späteren Applikation ist es notwendig, die Körperbewegungen des Anwenders zu erfassen und daraus Gesten abzuleiten. Der erste Schritt, die Erfassung der Bewegung von Körperteilen, übernehmen Trackingsysteme. Diese Systeme messen Kennwerte der Bewegung einzelner Körperteile in einem zeitlichen Zusammenhang und leiten diese Informationen an das Gestenerkennungssystem weiter. Im folgenden soll allgemein auf Trackingsysteme und speziell auf das an der Hochschule vorhandene Trackingsystem eingegangen werden.

2.3.1. Trackingsysteme

Für die Erfassung von Körperbewegungen stehen eine Vielzahl unterschiedlicher Trackingsysteme mit unterschiedlichen Technologien zur Verfügung. Diese Systeme können in drei Kategorien unterteilt werden:

1. Inside-In Systeme

³Bild aus http://www.medien.ifi.lmu.de/fileadmin/mimuc/hs_2004/vortrag_pattis.pdf

Bei diesen Systemen befinden sich die Sensoren auf dem Körper des Benutzers und werden direkt in Signale umgewandelt. Ein Beispiel ist die Nutzung eines elektromechanischen Skelettes. Dabei erfolgt die Messung über Potentiometer, welche über bewegten des mechanischen Skeletts am Körper verstellbar werden. Abbildung 2.2 zeigt ein solches System.

2. Inside-Out Systeme

Bei diesem System sind die Sensoren direkt am Körper des Benutzers angebracht. Die Aufnahme, der von den Sensoren gesendeten Signalen, erfolgt jedoch von entfernt positionierten Empfängern. Ein Beispielsystem ist die Arbeit, mit am Körper befestigten Funk-sendern und außen stehenden Sensoren, welche die ankommende Feldstärke messen.

3. Outside-In Systeme

Bei Outside-In Systemen messen die Sensoren von außen die Körperstellungen des Benutzers in einem Bereich in welchem der Nutzer steht. Die Messsignale werden dabei von außen auf den Körper geworfen und die zurück kommenden Reflexionen gemessen. Beispiel hierfür ist das in dieser Arbeit verwendete optische Erfassungssystem. Bei diesem System wird von außen Infrarotlicht ausgestrahlt, welches dann von am Körper befestigten reflektierenden Markern zurück geworfen und mit außen stehenden Kameras erfasst wird.

Es gibt für jede dieser Kategorien eine Vielzahl von technischen Lösungen. Eine detaillierte Übersicht finden Sie in den Arbeiten von [Heitsch \(2008\)](#), [Boetzer \(2008b\)](#), [Allen u. a. \(2001\)](#) und [Noack \(2007\)](#).

Alle dort vorgestellten Technologien haben gemeinsam, dass sie physikalische Werte statisch oder dynamisch erfassen und an das verarbeitende System weitergeben. Die jeweils eingesetzten Trackingtechnologien unterscheiden sich, für ein angeschlossenes System, meistens nur in der Art der erfassten Daten, in der Präzision der Erfassung und dem verwendeten Datenmodell. Einen umfassenderen Überblick zur Art der gelieferten Daten und den dazugehörigen Genauigkeitsanforderungen in unserem Szenario wird im dazu gehörigen Analyse- und Designkapitel erörtert ([3.3.3](#) sowie [4.2.2](#)).

2.3.2. Freiheitsgrade

Für die spätere Umsetzung ist es wichtig den Begriff und die Bedeutung der unterschiedlichen Freiheitsgrade („Degree of Freedom“; Abgekürzt DOF) zu verstehen.

Unter dem Begriff „Degree of Freedom“ (DOF) versteht man die verschiedenen möglichen Freiheitsgrade von Bewegungen. Dabei umfassen die ersten drei DOFs die Translationen in

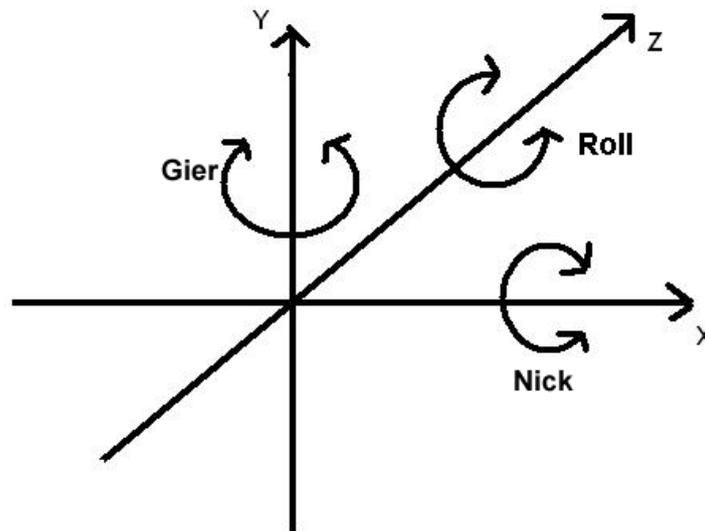


Abbildung 2.3.: Koordinatensystem 6DOF

die verschiedenen Richtungsachsen (x, y, z).⁴ Die nächsten drei beschreiben die Drehbewegung um die Richtungsachsen, welche als Gier-, Roll- und Nickwinkel bezeichnet werden. Die Kombination aus Translation und Drehbewegung erlaubt dabei sechs Freiheitsgrade.⁵ Unterschiedliche Trackingsysteme erlauben hierbei das Erfassen einer unterschiedlichen Anzahl von Freiheitsgraden. Das hier eingesetzte ART-Tracker System beherrscht die Erfassung sowohl von 3DOF-Objekten als auch von 6DOF-Objekten. Abbildung 2.3 zeigt die jeweiligen Freiheitsgrade eines Objektes für 6DOF.

2.3.3. ART Tracker

Für die Realisierung unseres Szenarios verwenden wir das im Labor der Hochschule für angewandte Wissenschaften stehende Trackingsystem ART-Tracker der Firma A.R.T GmbH.⁶ Es ist ein optisches Outside-In System und besteht aus einem Tracking Server, auf welchem die Firmensoftware DTrack ausgeführt wird und aus sechs speziellen FPGA-Infrarotkameras, welche mit dem Server verbunden sind. Alle diese Komponenten können als Komplettsystem von der Firma A.R.T.GmbH bezogen werden.

Das System besteht dabei mindestens aus zwei Kameras. Alle eingesetzten Kameras werden in den Ecken des Messraumes fest installiert und auf die Mitte des Raumes und den Benutzer ausgerichtet. In unserer Laborinstallation sind vier Kameras im Frontbereich und zwei im

⁴Bezeichnet als 3DOF.

⁵Bezeichnet als 6DOF.

⁶<http://www.ar-tracking.de>

rückwärtigen Raum angebracht. Abbildung 2.4 zeigt die Anordnung der Kameras im Labor der HAW. Sie senden in regelmäßigen Abständen einen Infrarotblitz aus und erfassen das zurückgeworfene IR-Licht als Bilder. Diese werden in den Kameras aufbereitet und an den zentralen Server weitergeleitet. Der Server errechnet aus den Informationen der Kamerabilder die Position der einzelnen Marker und sendet diese als TCP/IP- oder UDP-Paket in ein angeschlossenes Netzwerk. Von dort können die Positionsdaten von Systemen eingelesen und ausgewertet werden. Bei Aufbau des Systems ist eine einmalige Kalibrierung der installierten Kameras erforderlich.

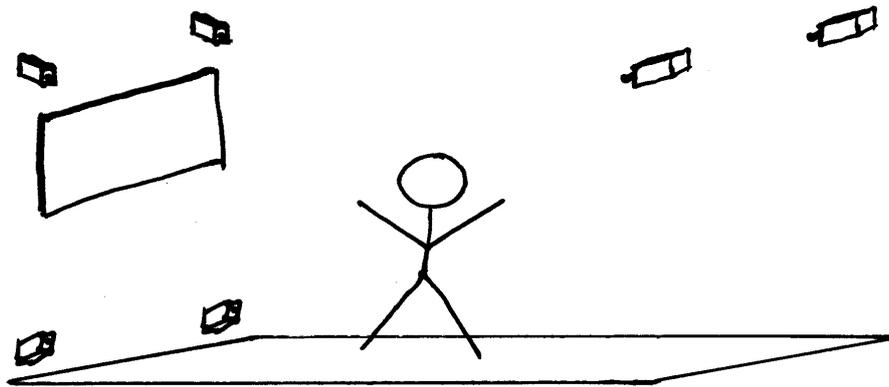


Abbildung 2.4.: Kameraanordnung im Labor

Die Positionserfassung basiert beim ART-Tracker auf aktiven oder passiven Markern. Diese sind kleine Infrarotlicht ausstrahlende oder rückstrahlende Elemente (z.B. Kugeln mit Reflektionsfolie). Die Marker werden an dem zu vermessenden Objekt befestigt und von den Kameras des ART-Tracker System erfasst. Bei der Erfassung müssen immer zwei Kameras den jeweiligen Marker sehen, um die Position des Markers eindeutig bestimmen zu können. Zu sehr verdeckte Marker werden nicht erkannt und führen zu Lücken in der Aufzeichnung. Darüber hinaus muss der Erfassungsbereich frei von nicht gewünschten Infrarotlichtquellen und stark reflektierenden Objekten sein. Ansonsten würde das System diese Objekte fälschlicherweise als reflektierende Marker interpretieren.

Es werden, wie schon beschrieben, zwei Arten von Rückgabeobjekten mit verschiedenen Freiheitsgraden vom ART System unterschieden. Diese an das angeschlossene System zurück gelieferten Objekte sind:

- **Marker**

Marker sind einzelne aktive oder passive Messobjekte, welche an dem zu vermessenden Objekt befestigt werden. Die Marker werden von mehreren fest im Raum angeordneten

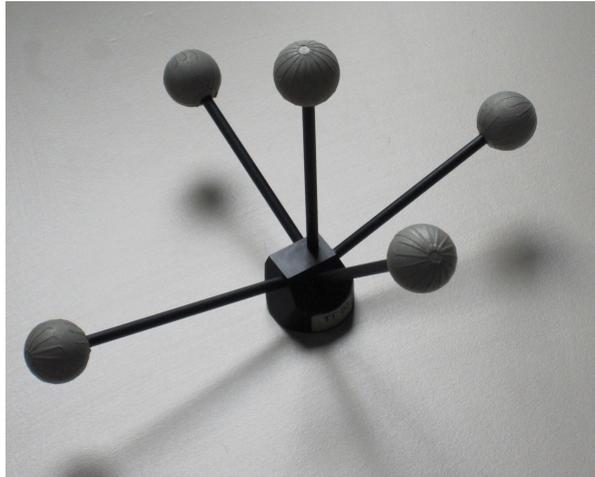


Abbildung 2.5.: Body

IR-Kameras erfasst. Aus den einzelnen Kamerabildern wird dann die Position des Messobjektes im dreidimensionalen Raum errechnet. Der Freiheitsgrad für Marker ist 3DOF, wodurch es möglich ist die Position im Raum, nicht jedoch die Drehorientierung des Objektes zu bestimmen.

- **Bodys**

Diese Markerkonstrukte bestehen aus mehreren einzelnen Markern. (Für 6DOF mindestens vier.) Diese Marker sind, mittels nicht reflektierender Stangen, in einem festen Abstand zueinander angeordnet. Die einzelnen Abstände der Marker zueinander werden vorher im System hinterlegt. Aufgrund dieser bekannten Abstände der Marker, ist das System nicht nur in der Lage die Position des Bodys im dreidimensionalen Raum, sondern auch seine eigene Drehorientierungen um einen vorher definierten Punkt des Bodys zu erfassen. Der Freiheitsgrad von Bodys ist maximal 6DOF. Der Body besitzt also neben der Position im globalen Koordinatensystem, noch ein eigenes Koordinatensystem, welches die eigene Drehorientierung angibt. [Abbildung 2.5](#) zeigt einen solchen Body.

Die einzelnen Bodys des ART-Tracker Systems unterscheiden sich untereinander in der Anzahl der Marker und den jeweiligen Abständen der Marker zueinander. Dadurch können Sie vom Trackingsystem eindeutig voneinander unterschieden werden. Die Firma A.R.T GmbH liefert einen großen Satz an vorgefertigten Bodys. Dabei geht die Bandbreite von Brillenaufsätzen, bis hin zu Komplettsystemen zur Erfassung der Grundbewegungen sämtlicher Körperteile.

Neben der eigentlichen Positionserkennung von Markern und Bodys, erkennt das Trackingsystem selbstständig die Identität von Objekten aus alter und neuer Messung und vergibt jedem

erkannten Objekt eine eindeutige ID. Die dazu genutzte Technik arbeitet mit stochastischen Verfahren um die Identität eines Objektes festzustellen.

Der ART-Tracker ist für die Erkennung der Identität von Bods recht robust, bei einzelnen Markern führt das Gerät öfters zur Erkennung des identischen Markers als neues Objekt und damit zur Vergabe einer neuen ID. Situationen hierfür sind schnelle Positionsveränderungen, mehrere dicht aneinander liegende Marker oder das nicht durchgehende Erkennen der einzelnen Marker (z.B. durch Verdeckung).

Das Trackingsystem ist trotzdem sehr solide und wird bereits in der Praxis, wie zum Beispiel in der Autoindustrie oder der Medizin, erfolgreich eingesetzt. Es benötigt eine feste Kamerainstallation und eine einmalige Kalibrierung des Kamerasystems. Die Qualität der Erkennung hängt sehr von der Kalibrierung, von IR-Störquellen im Aufnahmebereich und der Anzahl und Positionierung der IR-Kameras im Raum ab.

2.4. Gesten

2.4.1. Begriffserklärung

In dieser Arbeit geht es um die Steuerung von Computersystemen mittels Gesten. Dafür muss als erstes erläutert werden, was Gesten eigentlich sind, welche Arten von Gesten es gibt und welche für unsere Zwecke von Interesse sind.

Die Bezeichnung Geste kann je nach Betrachtungsweise sehr unterschiedlich aufgefasst werden. „Geste“ bedeutet eine Handlung mit symbolischem Charakter. Im allgemeinen Sprachgebrauch versteht man unter einer Geste jedoch die Bewegung von Körperteilen (besonders der Arme und Hände), durch die man etwas Bestimmtes zum Ausdruck bringt.⁷ Eine tiefer gehende Klassifikation und Definition des allgemeinen Begriffs Geste findet man unter [Pavlocic u. a. \(1997\)](#).

In dieser Arbeit wird sich auf Gesten, welche mittels Körperbewegungen ausgelöst werden, konzentriert. Dabei werden unbewusste Handlungen von bewussten Handlungen zur Kommunikation unterschieden. Bewusste Gesten dienen aktiv der Kommunikation (z.B. Stop-Haltung der Hand), während unbewusste Gesten meistens für andere Kommunikationskanäle unterstützend genutzt werden (z.B. auf etwas Zeigen, von dem gerade gesprochen wird.). Diese Arbeit konzentriert sich auf bewusst durchgeführte Gesten zur direkten Kommunikation.

Wie bei [Rahimi und Vogt \(2008\)](#) und [Boetzer \(2008a\)](#) werden auch in dieser Arbeit die Gesten zur Interaktion mit Computersystemen als manipulative Gesten verstanden. Der Nutzer möchte mit seiner Handlung den Zustand des Computersystems ändern bzw. manipulieren.

⁷<http://de.thefreedictionary.com/Geste>

Die Arbeit von [Boetzer \(2008a\)](#) teilt die manipulativen Gesten in drei Arten ein, welche sich in dem für sie jeweils benötigtem Benutzertraining unterscheiden. Diese Einteilung soll für diese Arbeit übernommen werden.

- **Bewegungsverfolgung**

Bewegungsverfolgung oder besser genannt physikalische Interpretation, sind Gesten, welche unmittelbar eine Entsprechung in der realen Welt haben. Beispiel hierfür ist das Verschieben von virtuellen Objekten auf einem Touchscreen. Der Benutzer assoziiert das virtuelle Abbild und die durchführende Handlung mit realen Gegenständen, physikalischen Regeln und Funktionen, wodurch die Interaktion mit dem Objekt stark erleichtert wird und der Trainingsaufwand des Benutzers gering ist.

- **Kontinuierliche Gesten**

Kontinuierliche Gesten sind Handlungen, welche in der realen Welt keine wirkliche Entsprechung haben. Sie basieren jedoch auf Metaphern und bekannten mentalen Modellen. In unserer Anwendung beispielsweise wird mittels einer Zeigergeste auf einen Punkt auf dem Bildschirm gezeigt. Der Nutzer kann diese Handlung mit einem aus seinem Finger kommenden Laserpointer assoziieren, welcher sich als einfacher Punkt auf dem Bildschirm äußert. Der Trainingsaufwand für solche Gesten ist höher. Die Geste ist jedoch in den meisten Fällen für den Benutzer leicht verständlich und einfach durchzuführen.

- **Symbolisch-manipulative Gesten**

Symbolisch-manipulative Gesten sind Gesten, für die es keine wirkliche reale Entsprechung gibt. Das mentale Modell ist bei diesen Gesten sehr kompliziert und führt zu einem hohen Trainings- und Verständnisaufwand. Ein Beispiel aus der späteren Anwendung ist die Funktionsselektion. Dabei wird mittels der Bewegung der Hand in eine bestimmte Richtung eine bestimmte Funktion angewählt. Das die Handbewegung in eine Richtung zu einer Funktionsauswahl führt, muss dem Benutzer zuerst vermittelt werden. Die symbolisch-manipulativen Gesten sind schwerer in ihrer Anwendung und erhöhen den Trainingsaufwand, erweitern jedoch das mögliche Gestenrepertoire um einen großen Satz möglicher Gesten.

2.4.2. Gesten als Kommunikationsmittel

Körpergesten werden in der ganzen Welt als bewusstes oder unterbewusstes Kommunikationsmittel verwendet. Sei es als Unterstützung verbaler Kommunikation oder als Ausdruck von Gefühlen und Stimmungen. Dabei ist die Interpretation von Gesten immer abhängig von äußeren Einflüssen wie Kulturkreis, Bildung sowie die derzeitige Situation. Das im westlichen Raum gängige Handzeichen für OK bzw. super (Daumen und Zeigefinger der Hand bilden einen



Abbildung 2.6.: OK Geste

Kreis, Abbildung 2.6) ist in Japan das Zeichen für Geld und in bestimmten Ländern kann es als „Du bist Wertlos“ interpretiert werden.⁸ Daher muss bei der Kommunikation mittels Gesten immer der richtige Kontext zwischen Sender und Adressat geklärt sein. Dieses gilt besonders im Bereich der Mensch-Computer-Interaktion, da der Computer nur schwer den Kontext der Kommunikation erkennen kann. Training und gemeinsame Absprachen bilden eine Möglichkeit Fehlinterpretationen zu vermeiden.

2.4.3. Statische und dynamische Gesten

Für diese Arbeit müssen zwei Arten von Gesten unterschieden werden: Statische und dynamische Gesten.

Statische Gesten sind hierbei zeitlose Handlungen. Die Einnahme einer bestimmten, vorher definierten, Haltung der Hand ist in unserer späteren Beispielanwendung bereits eine statische Geste, welche zu einer bestimmten Interaktion mit dem System führt.

Im Gegensatz dazu gibt es dynamische Gesten. Diese Gesten werden über einen gewissen Zeitraum hinweg durchgeführt. In unserer späteren Anwendung ist dieses die Bewegung der Hand in eine bestimmte Richtung. Zur Erkennung dieser Geste ist es erforderlich, die Handbewegung über eine gewisse Zeit hinweg durchzuführen.

Das Erkennen von statischen Gesten ist im Gegensatz zu dynamischen Gesten einfacher, da bei dynamischen Gesten die Einbeziehung der Zeit die Komplexität stark erhöht und eine saubere Start-Stop Erkennung nötig ist.

⁸http://www.rp-online.de/reise/ratgeber/Wenn-Gesten-obszoen-werden_bid_16035.html

2.4.4. Start/Stop Erkennung von Gesten

Um eine einzelne Geste eindeutig erkennen und auswerten zu können, ist es notwendig den Zeitpunkt zu bestimmen, wann die einzelne Geste beginnt und wann sie aufhört. Wichtig ist die Beantwortung dieser Frage, um im weiteren Verlauf die einzelnen Gesten zeitlich voneinander klar trennen und von einander unterscheiden zu können.

Eine saubere Erkennung des Start- und Endzeitpunktes einer Geste erleichtert zusätzlich gewollte kommunikative Gesten von einfachen Handbewegungen zu unterscheiden. Die Lösung dieser Aufgabe ist nicht immer trivial. Im dreidimensionalen Raum tritt bei der Interaktion mit dem System immer wieder das Problem auf, dass das System nicht unterscheiden kann, ob der Benutzer eine gezielte Geste ausführt und eine Reaktion erwartet oder ob die Bewegung, welche er gerade macht, aus anderen Gründen erfolgt.

Andere Interaktionssysteme können den Start- und Endzeitpunkt leichter erkennen. Bei der Interaktion mittels Touchscreen beispielsweise wird im Interaktionsmodell festgelegt, dass eine Geste mit Berührung der Touchoberfläche beginnt und mit dem Loslassen von der Oberfläche endet. Bei unserer Interaktion im dreidimensionalen Raum ist so eine Vereinbarung nicht so leicht möglich, da unsere Sensoren ständig die Bewegungen des Benutzers erfassen.

Ein weiterer Vorteil den Start- und Endzeitpunkt einer Geste zu erkennen, ist die Unterscheidung von mehreren aufeinander folgenden Gesten. Aus den Informationen, wann die eine Geste aufhört und wann die nächste Geste anfängt, können der Bildung von Gestenketten dienen. Diese Gestenketten können wieder eigene komplexere Gesten darstellen. Im nächsten Abschnitt wird auf die Kettenbildung von Gesten näher eingegangen.

Das automatische Erkennen von Start und Ende einer kommunikativen Geste ist, wie bereits erwähnt, überaus komplex und soll in dieser Arbeit nicht eingehend untersucht werden. Der Block der Start/Stop Erkennung wird daher im späteren Verlauf als vorhanden und funktions-tüchtig angesehen.

2.4.5. Kettenbildung von Gesten

Mehrere Grundgesten können, durch Aneinanderreihung, zu einer neuen Geste kombiniert werden. Die einzelnen Grundgesten sind dann wie Buchstaben eines Alphabets und werden so zu komplexeren Worten kombiniert. Ein Beispiel ist die Aneinanderreihung folgender Gesten:

1. „Zusammenführen der Finger“
2. „Handbewegung in eine Richtung“
3. „Auseinanderführen der Finger“

Diese Kombination kann von der Anwendung als Drag and Drop Geste erkannt werden und führt beispielsweise zum Verschieben eines Icons auf einem Desktop. In diesem Fall entsteht durch die durchgeführte Kettenbildung eine neue komplexere Geste, welche als eigenständige Geste vom System erkannt werden kann.

Neben der Zusammenfassung von zeitlich versetzten Gesten, können auch Gesten zusammen gefasst werden, welche zur selben Zeit aktiv sind. Das Bewegen der Hand in eine Richtung mit dabei unterschiedlichen Handhaltungen, kann beispielsweise vom System als jeweils neue eigenständige Geste interpretiert werden.

Bei der Kombination von Gesten muss im Modell darauf geachtet werden, dass es Gesten gibt, welche nur nacheinander ausgeführt werden können, weil sie sich gegenseitig ausschließen. Eine Hand kann zur selben Zeit nur nach links oder rechts bewegt werden. Andere Gesten sind jeweils in Ihrer Durchführung von einander abhängig. Die eine Geste kann nur ausgeführt werden, wenn parallel eine andere Geste ausgeführt wird.

Diese Erkenntnis ist für die spätere Modellbildung von Relevanz. Es muss zum einen sicher gestellt werden, dass sich gegenseitig ausschließende Gesten nicht zur selben Zeit eingesetzt werden.

Zusätzlich ist es für die Anwendung möglich, mittels des entworfenen Modells, bei vorher definierten Gestenketten, eine gewisse Annahme zu treffen, welche Geste als nächstes auftreten sollte. Dieses ist für die Anwendung von Nutzen, da eine gewisse Erwartungshaltung mit einbezogen wird, um die Qualität der Erkennung zu erhöhen.

2.4.6. Attribute einer Geste

Alle Gesten einer Gestenklasse haben, neben den gemeinsamen Eigenschaften, auch ausgesuchte Attribute, welche sich in Ihrem Wert von Geste zu Geste unterscheiden. Zum Beispiel besitzen die Gesten der Gestenklasse „Handbewegung in eine Richtung“ als Gemeinsamkeit die Bewegungsrichtung. Die einzelnen Gesten unterscheiden sich jedoch in ihrer Startposition voneinander. Die Startposition ist daher eine spezifische Eigenschaft der jeweiligen konkreten Geste. Je nach Modell, können für die verschiedenen Gestenklassen Attribute definiert und beim Erkennen der Geste mit übergeben und ausgewertet werden. Durch die Attribute ist es einer Anwendung möglich, neben der Erkenntnis, dass eine Geste durchgeführt wird, zusätzliche Informationen zu erhalten und darauf zu reagieren.

Die Interaktion mittels Gesten in einem Softwaresystem kann sehr komplex sein, daher ist im folgenden eine gründliche Analyse der Anforderungen an unser späteres System nötig.

3. Analyse

In dieser Arbeit soll ein System entwickelt werden, mit dessen Hilfe Fragestellungen aus dem Bereich der dreidimensionalen Gestensteuerung bearbeitet werden können. Zu diesem Zweck wird eine beispielhafte Anwendung in einem fiktiven Anwendungsszenario entworfen und deren Anforderungen spezifiziert. Das Anwendungsszenario dient dabei als allgemeiner Rahmen für unterschiedlichste Interaktionstechniken.

3.1. Anwendungsszenario

Einige Forschungsarbeiten an der HAW Hamburg befassen sich mit der Entwicklung von Geräten, Technologien und Anwendungen für Rettungseinsätze. Als ein Thema wird auf die Unterstützung der Einsatzkräfte bei großen Katastrophen eingegangen. Dabei geht es zum einen um die Gewinnung von Informationen über den Einsatzort und die Einsatzkräfte mittels intelligenter, verteilter Technik ([Tetzlaff \(2008\)](#); [Gerling \(2008\)](#)). Zum anderen geht es um die Entwicklung eines Rettungsleitstandes zur Lageanalyse und Koordination der Einsatzkräfte am Katastrophenort. Der Leitstand dient dabei als Bündelung aller für den Einsatz nötiger Informationen. Alle von der Technik und den Einsatzkräften vor Ort erfassten Informationen, (Wärmebilder, Positionsdaten u.v.m.) werden hier zusammen getragen. Die Aufgabe der Mitarbeiter im Leitstand ist hierbei die Sichtung und Bewertung der eingehenden Daten und die Koordinierung aller beteiligten Einsatzkräfte. Bei Katastrophen ist vor allem eine schnelle und gute Koordination aller Einsatzkräfte von Nöten. Der Leitstand der Zukunft, welcher an der HAW entwickelt wird, soll die Bewältigung dieser Extremsituation unterstützen.

Derzeit besteht der Leitstand aus einer hochauflösenden interaktiven Powerwall⁹ zur Anzeige und Selektion wichtiger Informationen, aus einem multitouch fähigen Tisch mit integrierter Bildschirmoberfläche zur kollaborativen Zusammenarbeit mehrerer Experten und aus einzelnen Handhelds für den persönlichen Informationsabruf der Leitstandmitarbeiter. Eine genauere Beschreibung des Leitstandes finden Sie unter [Boetzer \(2008b\)](#). Die detailliertere Beschreibung eines, im Leitstand verwendeten physikbasierten multitouchfähigen Tisches wird in der Arbeit von Philip Roßberger beschrieben ([Roßberger \(2008\)](#)). Alle diese genannten Geräte stehen im Labor der Hochschule für angewandte Wissenschaften zur Verfügung.

⁹Powerwall: Sehr große hochauflösende Bildschirmwand.



Abbildung 3.1.: Minority Report UI¹⁰

Diese Arbeit konzentriert sich auf die interaktive Powerwall. Die Powerwall soll dem Einsatzleiter oder einem anderen Leitstandmitarbeiter die Möglichkeit geben schnellen und einfachen Zugriff auf alle nötigen Informationen zu erhalten. Der Mitarbeiter sichtet und analysiert auf der Powerwall alle eingehenden Informationen und leitet diese an die jeweiligen Experten weiter. Die gelieferten Informationen können dabei Live Videos von Feuerwehrteams im Einsatz, aber auch Archivaufnahmen vom Ort vor dem Unglück oder andere Informationen, welche für den Einsatz erforderlich sind, sein. Der Leitstandmitarbeiter soll dabei in die Lage versetzt werden schnell und einfach auf die für ihn relevanten Informationen interaktiv zugreifen zu können. Elementar für die Qualität des Systems ist eine schnelle und effektive Steuerung der Anwendung und die leichte übersichtliche Auswahl von Informationen. Diese Anforderungen soll ein hier entwickeltes System mit Hilfe von Gestensteuerung leisten.

Die Umsetzung dieses Teils des Szenarios orientiert sich an einer Szene aus dem Film *Minority Report*, in welcher Tom Cruise vor einem Bildschirm durch eine fantastische Zukunftsvision mittels Handbewegungen navigiert (Abbildung 3.1). Dabei nutzt der Darsteller Gesten, um auf einem großen Display die einzelnen Aufzeichnungen abzuspielen und zu manipulieren.

Um die Umsetzbarkeit des gezeigten Szenarios zu zeigen, wird eine beispielhafte Anwendung entworfen. Vereinfacht ausgedrückt handelt es sich bei der Anwendung um ein Medien Anzeige- und Abspielsystem, welches über einfache Gesten gesteuert wird und seine Ergebnisse auf der Powerwall ausgibt. Der Fokus hierbei liegt auf der berührungslosen Interaktion frei im Raum mit Hilfe von definierten Körpergesten.

Der Nutzer könnte dabei die Anwendung mit dem gesamten Repertoire an möglichen Körpergesten, wie beispielsweise Beinbewegungen oder Kopfhaltungen, steuern. Zur Vereinfachung konzentrieren wir uns bei unserem Leitstand jedoch nur auf ausgesuchte Gesten, welche mit

¹⁰Bild aus dem Film "Minority Report" von 20th Century Fox Film Corporation

der Hand ausgeführt werden. Mittels der Handgesten soll gezeigt werden, dass eine Steuerung mittels Gesten grundsätzlich möglich ist.

Dieses Szenario ist auch unter der Einbeziehung mehrerer Nutzer, also als Collaborative Workspaces denkbar, in welchem die Nutzer die Objekte auf der Powerwall gemeinsam manipulieren und nutzen können. Für unseren Teilbereich der Interaktion soll in dieser Arbeit darauf nicht näher eingegangen werden. Wir konzentrieren uns auf die Interaktion mit einer Person. Die gemeinsame Arbeit mit mehreren Personen an einem interaktiven Tisch oder anderem Bürogeräten wird in anderen Arbeiten genauer beschrieben ([Roßberger \(2008\)](#), [Neale u. a. \(2004\)](#)).

3.2. Handgestensteuerung versus Maus/Tastatur

Grundsätzlich könnte die Interaktion in unserem Szenario auch mit der klassischen Maus und Tastatur Kombination durchgeführt werden. Die Benutzung von Handgesten bietet gegenüber den gängigen Eingabemitteln einige andere Ansätze und Möglichkeiten. In unserem Szenario eines Katastrophenfalls ist eine schnelle und einfache Interaktion erforderlich. Der Leitstandmitarbeiter hat die Möglichkeit durch Bewegungen der Hand mit dem Computersystem direkt zu interagieren und Informationen abzurufen. Dabei können die Gesten aus natürlichen, dem Benutzer bereits aus der realen Welt vertrauten und neuen speziell eingeführten Gesten bestehen. Im Gegensatz zur Tastatur und Maus, sorgen natürliche Gesten für eine schnelle Akzeptanz des Nutzers und lassen eine einfache Bedienung komplexer Systeme zu ([Micire u. a. \(2009\)](#); [Bolt und Herranz \(1992\)](#)). Der Nutzer ist dabei auch nicht unmittelbar an einen Ort für die Eingabe, wie einen Eingabetisch, gebunden. Er kann sich zu einem gewissen Grad frei im Raum bewegen, was in der Hektik eines Katastrophenleitstandes von Vorteil ist.

3.3. Funktionale Anforderungen

3.3.1. Scratch der Anwendung

Unsere eigentliche Anwendung soll zwei Arbeitsmodi beinhalten. Zum einen die Navigation in einem Multimediakatalog, in welchem die einzelnen Informationen abgelegt sind. Zum anderen ein Modus zur Darstellung bzw. Wiedergabe der jeweils ausgewählten Informationen. Im Navigationsmodus bekommt der Anwender die Informationen als Vorschaulemente in einer übersichtlichen Form (z.B. als Katalog) angezeigt. Mittels Gestensteuerung durchsucht er die Daten und kann die Elemente organisieren und anwählen. Der Wiedergabemodus dient zum

einen dem klassischen Wiedergeben der Medien mit den gängigen Start-, Stop- und Vorspulfunktionen. Darüber hinaus soll es auch möglich sein, das laufende Medium (zum Beispiel Videos) mittels Gesten auf verschiedene Arten zu manipulieren. Beispiele solcher Manipulationen sollen hier das Rein- und Rauszoomen in einzelne Bildbereiche und das Ändern der Art der Bilddarstellung sein.

Folgende Funktionalitäten sollen die beiden Modi leisten:

Modus 1 - Katalog

- Übersichtliche Anzeige der aktuellen Mediaobjekte
- Navigieren durch eine Struktur von Mediaobjekten
- Anordnen und Sortieren der Mediaobjekte (inklusive Verschieben/Gruppieren)
- Auswählen und Starten einzelner oder mehrerer Mediaobjekte

Modus 2 - Videoplayer

- Abspielen von Videos
- Standard Player Funktionen (Vor- und Zurückspulen, Pause, Stoppen, vorheriges/nächstes Video)
- Zoom in und aus einzelnen Bildbereichen
- Einstellen der Wiedergabeparameter (Helligkeit, Geschwindigkeit)

Die beiden Abbildungen [3.2](#) und [3.3](#) zeigen jeweils ein mögliches Aussehen der beiden Modi auf dem Bildschirm. Auf der ersten Abbildung sind die einzelnen Mediaobjekte und die für die Gruppierung genutzten virtuellen Container abgebildet. Die zweite Abbildung ist ein Beispiel für die Anzeige im Abspielmodus mit Timecode und Darstellung der Lautstärke.

3.3.2. Aufbau und Ablauf

Der Systemaufbau in unserem Szenario ist recht einfach. Der Benutzer steht in der Mitte des Raumes. Von dort aus interagiert er mit dem System durch das Durchführen von Handgesten. Für die Gesten steht ihm der gesamte freie Raum in seiner direkten Umgebung zur Verfügung.

Die Ausgabe erfolgt visuell über einen großen, hochauflösenden Bildschirm (Powerwall). Die Powerwall ist ca. drei Meter vor dem stehenden Benutzer in Augenhöhe platziert.

Die Abbildung [2.4](#) aus dem Abschnitt [2.3.3](#) zeigt die grundsätzliche Anordnung der von uns verwendeten Geräte im Labor der Hochschule für angewandte Wissenschaften.

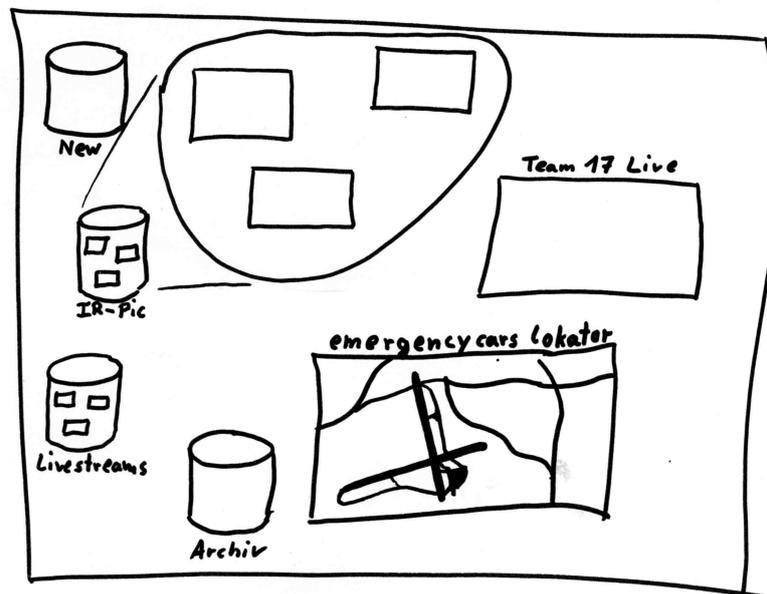


Abbildung 3.2.: Skizze Powerwall Katalogansicht

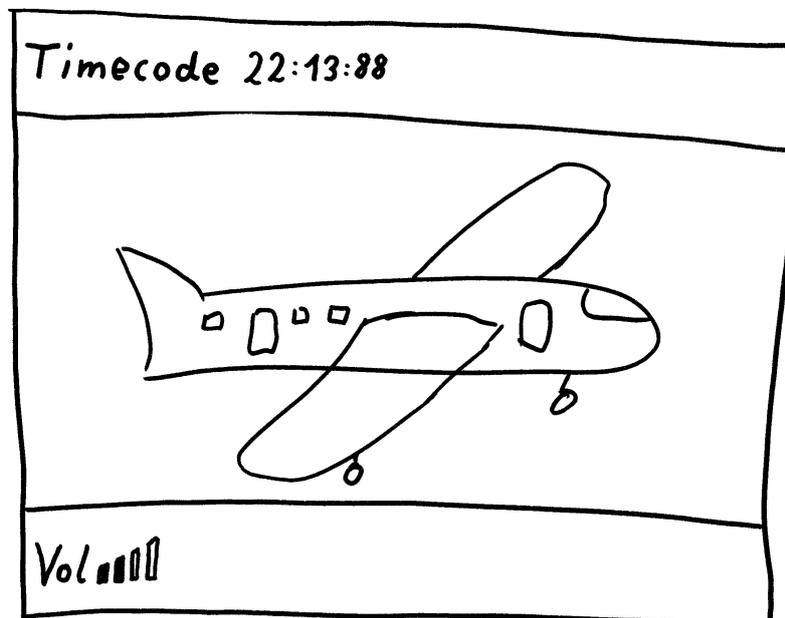


Abbildung 3.3.: Skizze Powerwall Playerdarstellung

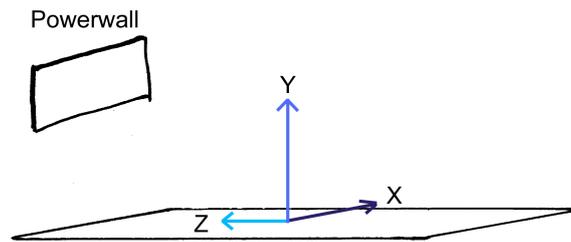


Abbildung 3.4.: Koordinatensystem Datenerfassung

Für den weiteren Verlauf ist es nötig zu definieren mit welchem Koordinatensystem und welcher Ausrichtung gearbeitet wird:

Das für die globale Positionsbestimmung genutzte Koordinatensystem hat seinen Ursprung auf dem Boden mittig im Raum. Wenn der Benutzer auf diesem Punkt steht und nach vorne Richtung Powerwall blickt, schaut er parallel zur Z-Achse. Schaut er nach Rechts, schaut er parallel zur X-Achse. Schaut er nach Oben, schaut er auf der Y-Achse entlang (Abbildung 3.4).

3.3.3. Anforderungen an die Datenerfassung

Es gibt einige Anforderungen unseres Szenarios an die Datenerfassung. Unser Szenario erfordert das Erkennen von natürlichen Bewegungen der Hand. Zu diesem Zweck soll das Erfassungssystem über eine recht hohe Präzision und Genauigkeit, insbesondere bei der Erfassung der einzelnen Punkte, verfügen. Neben dem Erkennen der Handposition soll auch die Position der Fingerspitzen erfasst werden können.

Die Handbewegungen können in unserem Szenario langsam oder schnell ausgeführt werden. Um daraus resultierende Messfehler zu vermeiden, sollte das Erfassungssystem kleine Messintervalle verwenden und die Bewegung kontinuierlich erfassen.

Der Benutzer will mit dem Computersystem direkt interagieren. Ein unmittelbares Feedback auf die Bewegungen des Benutzer ist nötig. Daher muss die Erfassung die Daten schnell und in Echtzeit erfolgen.

Wir betrachten in den meisten Fällen Körperbewegungen, welche über einen gewissen Zeitraum ablaufen. Das System muss daher die einzelnen Messpunkte nicht nur kontinuierlich erfassen, sondern auch über die Zeit den selben vorherigen Messpunkten zuordnen und gegenüber den anderen eindeutig unterscheiden können.

3.3.4. Anwendungsfälle

Ziel der Interaktion ist es Bild- und Videodaten zu betrachten und zu manipulieren. Der Leitstandmitarbeiter hat die Aufgabe, ankommende Daten zu sortieren, zu sichten und gegebenenfalls an andere Systeme weiter zu leiten. Aus dieser Aufgabenstellung ergeben sich mehrere Anwendungsfälle. Wie bei Philipp Roßberger ([Roßberger \(2008\)](#)) bezeichnet der Begriff „Datenobjekt“ auch hier den Oberbegriff für Bild- und Videodaten.

Datenobjekt positionieren Der Anwender kann die auf dem Bildschirm liegenden Objekte frei positionieren.

Datenobjekt löschen Ein einzelnes Datenobjekt kann bei der Voranalyse als unwesentlich klassifiziert und gelöscht bzw. archiviert werden.

Datenobjekt fokussieren/zoomen Ein einzelnes Datenobjekt kann in den Betrachtungsmodus versetzt werden. Dadurch wird dieses Objekt fokussiert und exklusiv auf dem Bildschirm dargestellt.

Datenobjekt betrachten/abspielen Der Anwender kann das Datenobjekt anschauen. Dabei ist es möglich einzelne Bereiche des Objektes heran zu zoomen und bei Videos die Geschwindigkeit der Wiedergabe zu ändern.

Datenobjekt weiterleiten Das Datenobjekt kann an andere Mitarbeiter und Geräte zur weiteren Bearbeitung weitergeleitet werden.

Der Anwendungsfall „Datenobjekt weiterleiten“ liegt nicht im Focus dieser Arbeit und ist nur der Vollständigkeit halber hier aufgelistet. Er sollte jedoch bei collaborativem Einsatz des Systems berücksichtigt werden, da in diesem Fall der einfache Austausch von Daten und Informationen ein elementarer Bestandteil des Gesamtsystems wird.

Die gesamte Interaktion des Benutzers mit den Datenobjekten, wie das Selektieren und Verschieben eines Objektes, soll hierbei mittels Körpergesten erfolgen. Im nächsten Abschnitt wird auf die einzelnen Interaktionstechniken zur Durchführung dieser Anwendungsfälle eingegangen.

3.4. Interaktionstechniken

Die Anwendungsfälle erfordern eine Reihe von Interaktionstechniken. Eine Zusammensetzung aus diesen bilden die Anwendungsfälle ab. Im Folgenden werden die einzelnen Elemente vorgestellt und die nötigen Bewegungsgesten genannt.

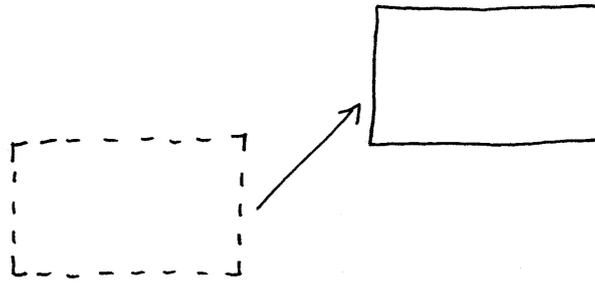


Abbildung 3.5.: Drag'n'Drop Geste

3.4.1. Drag-and-Drop

Für den Anwendungsfall der Positionierung der Datenobjekte wird eine Interaktionstechnik für das Verschieben von Datenobjekten auf dem Bildschirm benötigt. Dieser Anwendungsfall wird innerhalb des Modus eins benötigt, um Datenobjekte zu sortieren und Cluster zu bilden. Abbildung 3.5 zeigt das einfache Verschieben eines Datenobjektes auf dem Bildschirm. Der Ablauf ist vergleichbar mit der klassischen Maussteuerung, bei welcher der Mauszeiger auf das Objekt bewegt wird, das Objekt mittels Maustaste festgehalten und durch Verschieben der Maus an seinen Bestimmungsort bewegt wird.

Die Interaktionstechnik besteht aus zwei Gesten. Die erste Geste ist eine Zeigergeste, um den Cursor auf das Objekt zu bewegen und später zu verschieben. Die andere Geste ist eine Schaltergeste, welche mittels Daumenbewegung das eigentliche Datenobjekt „greift“ und festhält.

Für die Zeigergeste formt der Benutzer die Finger seiner Hand zu einer „Pistole“ (Kleiner-, Ring- und Mittelfinger angelegt und Zeigefinger und Daumen abgespreizt). Mit dem Zeigefinger zeigt der Benutzer in die Richtung des Bildschirms. Das System errechnet einen „Auftreffpunkt“ auf der Bildschirmoberfläche und zeigt dort einen Cursor an. Durch bewegen des Zeigefingers bzw. der gesamten Hand wird der Cursor auf dem Bildschirm bewegt.

Die zweite verwendete Geste ist die Daumenschaltergeste. Sie ist eine statische Ja/Nein Geste. Die Gestenbedingung ist erfüllt, wenn der Daumen so abgespreizt ist, dass er vom Zentrum der Hand aus nach oben ragt.

Durch die Gesten ergibt sich folgender Ablauf für eine Drag-and-Drop Aktion:

Mittels der Zeigergeste wird der Cursor auf das zu verschiebende Objekt bewegt. Mit der Aktivierung der Daumenschaltergeste wird das Objekt gegriffen. Der Cursor und das darunter gegriffene Objekt werden mit der Zeigergeste an den Zielort bewegt. Zum Schluss wird mit dem Deaktivieren der Daumenschaltergeste das Objekt losgelassen und die Interaktion beendet.

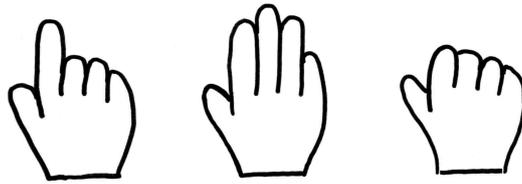


Abbildung 3.6.: Mögliche Icondarstellungen

Als Feedback sollte das Bild des Cursors den jeweiligen Stand der Drag-And-Drop Aktion widerspiegeln. In der Abbildung 3.6 sind mögliche Icons für den Cursor abgebildet. Das linke abgebildete Icon wird angezeigt, wenn die Daumenschaltergeste aktiviert wurde, jedoch kein Objekt unter dem Cursor liegt. Das mittlere Icon wird im losgelassenen Zustand angezeigt. Das rechte Icon wird angezeigt, wenn ein unter dem Cursor liegendes Objekt mittels der Daumenschaltergeste „gegriffen“ wurde.

3.4.2. Funktionsauswahl

In einigen Anwendungskontexten wird eine Interaktionstechnik zur Auswahl von Funktionen benötigt. Unter anderem bei den Anwendungsfällen „Datenobjekte löschen“ und „Datenobjekt betrachten/abspielen“.

Je nach Anwendungssituation bietet das Datenobjekt die Möglichkeit durch Auswahl aus einer Liste von möglichen Funktionen das Objekt zu manipulieren. Die dazu gehörige Interaktion besteht aus zwei Gesten. Die erste Geste dient zum Aufruf und Anzeige des Funktionsmenüs. Mit der zweiten Geste wird eine der möglichen Funktionen ausgewählt.

Der erste Teil wird durch eine „Finger zusammenführen“-Geste realisiert. Wenn die Fingerspitzen zusammen geführt werden, erscheint auf dem Bildschirm ein Funktionsmenü mit Icons oder Texten als Repräsentanten für die jeweilige Funktion. Die Abbildung 3.7 zeigt beispielhaft eine Skizze für das Auswahlmenü bei der Videodarstellung.

Mit der zweiten Geste wird die jeweilige Funktion selektiert. Die Geste ist aktiv, wenn die Fingerspitzen eine bestimmte Zeit in eine gemeinsame Richtung bewegt werden. Die möglichen Bewegungsrichtungen sind äquivalent zu der Anordnung der Icons auf dem Bildschirm um den Mittelpunkt. Sie bilden so acht mögliche Richtungen. Diese Richtungen liegen auf der X - Y - Ebene, parallel zur Powerwall. Die Bewegung in eine der Richtungen aktiviert so die dort jeweils angesiedelte Funktion.

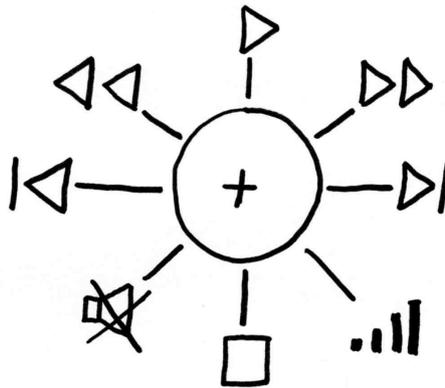


Abbildung 3.7.: Menü Funktionsauswahl

3.4.3. Drehregler

Für das Einstellen der Lautstärke oder der Wiedergabegeschwindigkeit wird ein Interaktionsobjekt zum Einstellen von Werten benötigt. (z.B. im Anwendungsfall: „Datenobjekt betrachten/abspielen“)

Die Anwendung wechselt zu diesem Zweck in einen Reglermodus, in dem ein Bild eines klassischen HIFI Reglers erscheint. Der Benutzer nimmt mit der Hand eine Position ein, als würde er einen echten Regler greifen. Er führt die Finger der Hand dabei zusammen. (Abbildung 3.8) Durch drehen der Finger auf einer Kreislinie nach links bzw. rechts wird dabei der Wert erniedrigt bzw. erhöht. Mit dem Loslassen des Reglers, also dem Öffnen der Hand, wird der Regelmodus wieder verlassen. Als visuelles Feedback zeigt die Anwendung den einzustellenden Wert und das sich mitdrehende Abbild des Reglers.



Abbildung 3.8.: Greifhand

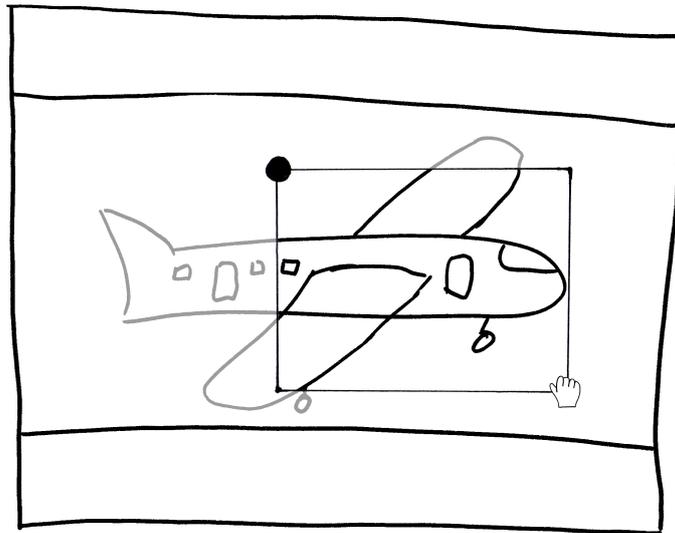


Abbildung 3.9.: Auswahl eines Bildbereiches

3.4.4. Objektselektierung

Für das gleichzeitige Löschen oder Verschieben mehrerer Datenobjekte ist es erforderlich, die einzelnen Objekte gemeinsam zu selektieren. Nach dem Wechsel in den Selektiermodus, wird mittels der oben bereits beschriebenen Zeigergeste ein Cursor auf ein Datenobjekt bewegt und dieses mit Aktivieren der Daumenschalter-Geste selektiert. Die Selektion wird optisch über eine Hervorhebung des Datenobjektes angezeigt. Durch mehrmaliges Wiederholen der Aktion können so mehrere Datenobjekte zusammen selektiert werden.

3.4.5. Auswahl eines Bildbereiches

Diese Interaktionstechnik wird zum Auswählen eines Teilbereiches eines Videos oder Bildes benötigt. Der Benutzer soll so in die Lage versetzt werden, Details auf Datenobjekten zu vergrößern und genauer betrachten zu können. Nach dem Wechsel in diesen Modus werden, mittels der Zeiger- und Daumenschaltergeste innerhalb des angezeigten Datenobjektes, nacheinander zwei Punkte ausgewählt. Zusammen bilden diese die gegenüberliegenden Eckpunkte eines Rechtecks. Nach der Auswahl des ersten Punktes wird als Feedback ein Rechteck angezeigt, welches vom ersten Punkt und der aktuellen Cursorposition aufgespannt wird. Nach der Auswahl des zweiten Punktes wird der von den zwei Punkten umspannende Bildbereich heran gezoomt. [Abbildung 3.9](#) zeigt die Selektion eines Bildbereiches innerhalb des Datenobjekts.

3.5. Erforderliche Gesten

Wie aus dem Abschnitt 3.4 Interaktionstechniken ersichtlich, sind für die einzelnen Interaktionstechniken des Szenarios die Umsetzung folgender Gesten erforderlich:

- Zeiger
- Daumenschalter
- Finger zusammenführen
- Richtungsbewegung
- Fingerkreisbewegung

Eine Beschreibung der Gesten, die aus diesen Geste sich ergebenden Features und die mögliche Klassifikationsstrategie wird beispielhaft im Designkapitel 4.2.4 und vollständig im Anhang A näher beschrieben.

3.6. Nicht funktionale Anforderungen

3.6.1. Einbeziehung der dritten Dimension

Mit der hier entwickelten Plattform soll es möglich sein, Gestensteuerung mittels dreidimensionaler Gesten zu betreiben. Die Applikation muss in der Lage sein dreidimensionale Körperbewegungen zu erfassen und die daraus resultierenden Gesten zu erkennen. Der zur Verfügung stehende Eingabebereich ist dabei, die Gesamte vom Nutzer mit seinen Armen direkt erreichbare Umgebung, ohne dabei seine Standposition zu verlassen.

3.6.2. Interaktivität und Feedback

Durch das Fehlen, des bei anderen Systemen vorhandenen haptischen Feedbacks (Beispielsweise die Auflagefläche bei Touchscreens), ist es erforderlich, auf andere Arten eine Reaktion auf die Eingaben des Nutzers zu liefern. Als Lösung wird hier ein Feedback mittels visueller und akustischer Elemente bevorzugt. Alternativ oder ergänzend wäre auch ein Feedback mittels spezieller im Handel erhältlicher Handschuhe möglich, welche an den Fingernkuppen vibrierende Drähte angebracht haben (ARTGmbH). Diese werden hier, unter anderem aus Kostengründen, nicht verwendet.

3.6.3. Sonstige nicht funktionale Anforderungen

Aufgrund der hohen Interaktivität und der zeitkritischen Arbeit im Leitstand muss die Verarbeitung der Eingabe und die Reaktion auf diese, auch bei der Darstellung von vielen Elementen und bei komplexen Gesten, schnell und mit einer Mindestreaktionszeit geschehen.

Das System und das Interaktionsmodell muss einen gewissen Grad an Fehlertoleranz haben, um falsche Eingaben oder Fehlinterpretationen zu kompensieren. Darüber hinaus gibt es noch die klassischen, nicht funktionalen Anforderungen an ein Softwaresystem.¹¹

3.7. Fazit

Im zurück liegenden Kapitel wurden das Anwendungsszenario und die Anforderungen an unser Softwaresystem vorgestellt. Das Szenario soll als Beispiel für eine komplexe Interaktion zwischen Mensch und Maschine mittels Gesten dienen. Das entworfene System soll die Möglichkeit bieten Fragestellungen aus dem Bereich Gestensteuerung zu beantworten. Die hier entworfenen Gesten und Interaktionstechniken decken einen Großteil von gängigen Usecases alter Systeme ab. Dazu zählen das Selektieren von Objekten, das Einstellen von Werten oder die Auswahl von Funktionen.

Die Gesten bieten einen Grundstock an Möglichkeiten zur Interaktion. Mit Ihnen ist es möglich zu zeigen, dass eine Steuerung mittels Gesten prinzipiell machbar ist. Darüber hinaus könnte das System leicht durch das Hinzunehmen anderer Gesten und Interaktionstechniken erweitert, ergänzt und optimiert werden.

¹¹[http://de.wikipedia.org/wiki/Anforderung_\(Informatik\)](http://de.wikipedia.org/wiki/Anforderung_(Informatik))

4. Design

Im folgenden Kapitel wird auf Ideen und mögliche Techniken zur Umsetzung eines im Analysekapitel vorgestellten Systems eingegangen. Dabei geht es insbesondere um den Bereich der Erkennung von Handgesten im dreidimensionalen Raum und der sich daraus ergebenden möglichen Interaktionsformen.

4.1. Grundarchitektur

Für die Softwaregrundarchitektur der Anwendung verwenden wir das Model-View-Controller-Pattern.

Das Model-View-Controller-Pattern ist ein Architekturmuster zur Unterteilung des Softwaresystems in einen Daten-, einen Steuer- und einen Anzeigeteil, welche lose gekoppelt werden. Abbildung 4.1 zeigt den Aufbau des MVC-Patterns. Vorteil dieser Vorgehensweise ist die leichte Austauschbarkeit der einzelnen Komponenten. Für das selbe Softwaresystem können Anwender spezifische Komponenten entwickelt werden. Die Entwicklung jeweils einer View für den Entwickler und für den Endbenutzer ist leicht umsetzbar. Nähere Informationen zur Nutzung des MVC-Pattern sind unter [Wikipedia](#) oder in der gängigen Fachliteratur verfügbar ([Miller u. a. \(2009\)](#)).

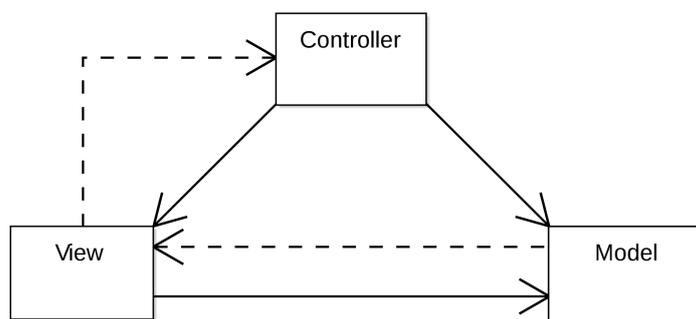


Abbildung 4.1.: Model-View-Controller-Pattern

Im Zuge des Ziels die Anwendung mit Gesten zu steuern, liegt der Focus dieser Arbeit nicht auf dem Model, mit den enthaltenen Datenstrukturen für die Wiedergabe der Datenobjekte, sondern auf View und Controller. Wobei der Controller die Gestenerkennung und Programmlogik übernimmt und die View die Daten des Models und das Feedback der Gestensteuerung für den Benutzer anzeigt. View und Controller bilden ganzheitlich betrachtet eine Einheit bei der interaktiven Steuerung der Anwendung durch Gesten, wobei im Controller die hier wichtigere Arbeit, die Gestenerkennung, stattfindet. Im Controllerteil werden die unterschiedlichen Gesten erkannt und lösen je nach Zustand des Modells eine definierte Aktion aus.

Der Controller besteht aus zwei unterschiedlichen Teilen:

Der erste Teil ist die Anwendungslogik, welche das Model und die View manipuliert. Dieser Teil führt alle im Abschnitt Anwendungsfälle beschriebenen Aktionen wie das Selektieren, Starten und Stoppen von Datenobjekten durch. Die dahinter liegende Logik soll sich an der Funktion eines normalen Mediaplayers orientieren und wird hier nicht weiter betrachtet.

Darüber hinaus nimmt die Anwendungslogik, vom zweiten Teil des Controllers, die Eingabe entgegen und manipuliert die View so, dass das Feedback der Eingaben des Benutzers dargestellt wird.

Der zweite Teil der Controllerkomponente enthält die Eingabekomponenten durch den Benutzer. Hier werden die verschiedenen Eingaben aufgenommen, interpretiert und einheitlich an die Anwendungslogik weitergeleitet. Die Benutzereingabe soll in unserem Szenario mit Hilfe von Gesten erfolgen. Zu diesem Zweck muss dieses Element eine Komponente mit Funktionen zum Erkennen von Körpergesten bereitstellen.

Die dafür nötige Gestenerkennung kann als Spezialisierung der allgemeinen Bildmustererkennung angesehen werden und führt so, wie im nächsten Abschnitt beschrieben, zu einer Architektur, welche der von klassischen Bildmustererkennungssystemen sehr ähnlich ist.

Beide teile des Controllers sind über eine einheitliche Schnittstelle lose gekoppelt, um die Nutzung anderer Gestenerkennungssysteme oder anderer Eingabekanäle wie Sprache, alternativ oder in Kombination, zu ermöglichen (Abbildung 4.2).

4.2. Architektur zur Erkennung von Gesten

Das hier zu entwickelnde System soll auf Eingaben mittels Körpergesten reagieren. Zu diesem Zweck soll im folgenden Abschnitt eine mögliche Komponente und damit verbundene Architektur zur Erkennung von Handgesten vorgestellt werden.

Die Erkennung einer oder mehrerer Gesten durch das Computersystem kann, wie bereits erwähnt, auf das Problemfeld der allgemeinen Mustererkennung zurückgeführt werden (Duda u. a. (2001)). Wie in der klassischen Bildmustererkennung erfordert auch dieses System, zur

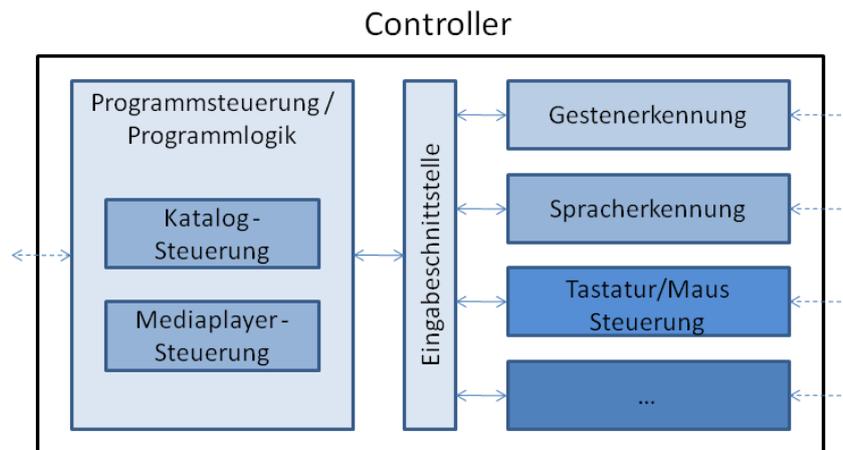


Abbildung 4.2.: Aufbau Controller

Gestenerkennung, eine Reihe von Verarbeitungsschritten. Diese Schritte beinhalten das Erfassen der Körperbewegungen des Benutzers mittels Sensoren, das Verarbeiten dieser Sensorinformationen und das Zuordnen des Ergebnisses zu einer vorher definierten Gestenklasse.

Johann Heitsch nennt in seiner Arbeit einen guten und einfachen Ansatz für eine mögliche Systemarchitektur zur Erkennung von Gesten ([Heitsch \(2008\)](#)). Seine beschriebene Architektur besteht aus vier Bereichen:

1. Datenquellen
2. Vorverarbeitung
3. Feature Extraktion
4. Klassifikation

Dieser Ansatz wird hier teilweise übernommen. Schritt 1 und 2 werden in unserem System zu einem Schritt zusammengefasst. Bei dem hier verwendeten ART-Tracker System findet intern die aufgeführte Vorverarbeitung bereits beim Erfassen der Daten statt.

Als neuer vierter Bereich wird die Schnittstelle zur Anwendungslogik hinzugefügt. Diese Schnittstellenkomponente hat lesenden Zugriff auf die Daten aller wichtigen unteren Bereiche. Dieser Zugriff stellt der Anwendung ein Maximum an Informationen zur Verfügung und macht dadurch die situationsbedingte Interpretation der Geste in der Anwendung möglich.

An die Seite der Merkmalsextraktion und der Datenerhebung wird noch eine Komponente zur Erkennung des Anfangs und Endes einer Geste oder einer Gestenkombination hinzugefügt.

Die nun erstellte mögliche Architektur besteht aus folgenden fünf Blöcken:

1. Erfassen der Körperbewegung mittels Sensoren

2. Extrahieren von Merkmalen/Features aus den gewonnenen Sensordaten
3. Klassifikation der Merkmale/Features und Bestimmung der Geste
4. Weiterleitung der Daten an das angeschlossene System
5. Start/Stop Erkennung

Abbildung 4.3 zeigt die daraus resultierende Architektur unseres Systems zur Gestenerkennung.

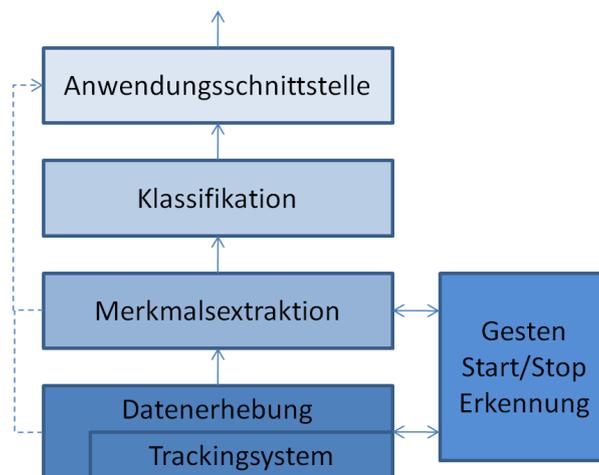


Abbildung 4.3.: Architektur Gestenerkennung

Die einzelnen Komponenten sind hierbei in Ihrer Qualität und Leistungsfähigkeit stark von den unter Ihnen liegenden Komponenten abhängig. Insbesondere Klassifikation und Merkmalsextraktion haben eine starke Wechselwirkung. Beide sind eng verzahnt in Ihrem Verhältnis von möglicher Leistungsfähigkeit und Komplexität. Die Qualität der Klassifikation ist stark von den vom Merkmalsextraktor gelieferten Merkmalen abhängig. Umso mehr „passende“ Merkmale geliefert werden, umso leichter gelingt eine Klassifikation. Jedoch muss der Aufwand diese Merkmale zu extrahieren im richtigen Verhältnis zur daraus entstehenden Qualität stehen. Darüber hinaus ist es nicht sinnvoll Merkmale zu übergeben, welche für die jeweilige Klassifikation nicht relevant sind und völlig umsonst extrahiert wurden. Ähnlich verhält es sich mit der Merkmalsextraktion und der Datenerhebung. Je besser die Datenerhebung durchgeführt wird, umso leichter ist eine Extraktion von sinnvollen Merkmalen möglich.

4.2.1. Start/Stop-Erkennung

Das Problem der Start/Stop-Erkennung wurde bereits im Grundlagenkapitel näher erläutert.

Das automatische Erkennen von Start und Ende einer Geste ist überaus komplex und soll in dieser Arbeit nicht eingehend untersucht werden. Der Block der Start/Stop Erkennung wird hier einfach als vorhanden und funktionstüchtig angesehen. Falls nötig bedienen wir uns einfacher technischer Hilfsmittel und Vereinbarungen, um das Problem zu kompensieren bzw. zu umgehen. Die Benutzung eines einfachen Schalters oder Tasters, welcher angibt, ob eine Gesteneingabe läuft ist eine Möglichkeit. Eine andere Variante ist bestimmte eindeutige Start und Stop Gesten zu definieren, welche die eigentliche Geste umschließen und klar von ihr unterschieden werden können. Ein Beispiel wäre eine Ruhephase, welche vor und nach jeder Geste eingehalten wird um die Gesten voneinander zu trennen. Dieses sind einige Möglichkeiten, auf die konkrete Umsetzung soll in dieser Arbeit jedoch nicht eingegangen werden.

Im Folgenden werden nun die vier in der Architektur beschriebenen Hauptbereiche der Gestenerkennung näher erläutert:

1. Datenerhebung
2. Merkmalsextraktion
3. Klassifikation
4. Anwendungsschnittstelle

4.2.2. Datenerhebung

Im ersten Schritt der Gestenerkennung werden die Bewegungen des Benutzers vom System erfasst. Das Trackingsystem erfasst die Daten und sendet sie an die Datenerhebungskomponente. Die Datenerhebungskomponente bereitet diese Daten einheitlich auf und gibt sie an die oberen Schichten weiter. Wie bereits im Grundlagenkapitel 2.3 erwähnt, gibt es eine Vielzahl von Verfahren und Systemen um die Position und Bewegung des Körpers zu erfassen. Daher soll eine lose Kopplung von Trackingsystem und Datenerhebung es ermöglichen, die Eingabedaten von unterschiedlichen Trackingsystemen zu verarbeiten.

Die Bewegung von Körperteilen des Menschen, wie zum Beispiel seiner Hand, mit all Ihren Muskeln und Möglichkeiten, ist im Detail für die meisten Anwendungen zu komplex. Daher findet in allen Fällen schon bei der Datenerfassung eine Reduktion der Informationen statt. Es wird für die Applikation ein einfacheres Modell des Körperteils entworfen. Je nachdem welche Daten zur Erfassung ausgewählt werden und wie diese erfasst werden, führt die Messung zu einem mehr oder weniger hohem Messumfang und daraus resultierender Messgenauigkeit der eigentlichen Bewegung. Als Beispiel für Systeme mit geringem Messumfang sei hier der Eingabestick der Entertainment Konsole Nintendo Wii¹² genannt. Der Eingabestick wird in der Hand gehalten und misst über Beschleunigungssensoren die ungefähre Beschleunigung der

¹²[urlhttp://de.wikipedia.org/wiki/Wii](http://de.wikipedia.org/wiki/Wii)

Hand des Nutzers. Diese Beschleunigungswerte sind für das Modell der meisten Wii-Spiele ausreichend. Arbeiten zum Thema „Gestenerkennung und Interaktion mit dem Wii-Kontroller“ werden unter [Scherfgen und Herpers \(2009\)](#) sowie [Schlömer u. a. \(2008\)](#) vorgestellt.

Viele der professionell eingesetzten Techniken liefern die genauen Positionen bestimmter Marker bzw. Bodys am Körper zu einem bestimmten Zeitpunkt. Eine dieser Technologien soll auch für diese Arbeit verwendet werden. Das an der Hochschule für Angewandte Wissenschaften vorhandene System, erfasst einzelne markante Messpunkte der Hand. Durch die zeitliche Verknüpfung der einzelnen Messwerte entsteht so eine stetige Bewegung der Punkte im Raum.

Die Qualität der eigentlichen Erfassung ist stark von der Präzision der Sensoren, der Aufnahmegeräte, der Anzahl der Messpunkte und dem verwendeten Messintervall abhängig. Umso mehr Informationen über die Körperbewegung erfasst werden, umso höher ist die Wahrscheinlichkeit, dass die Geste richtig erkannt und zugeordnet werden kann. Dadurch erhöht sich auch die Komplexität des dahinter liegenden Modells stark. Daher sollte bei der Auswahl der Art der Daten immer zwischen Nutzen und Aufwand abgewogen werden.

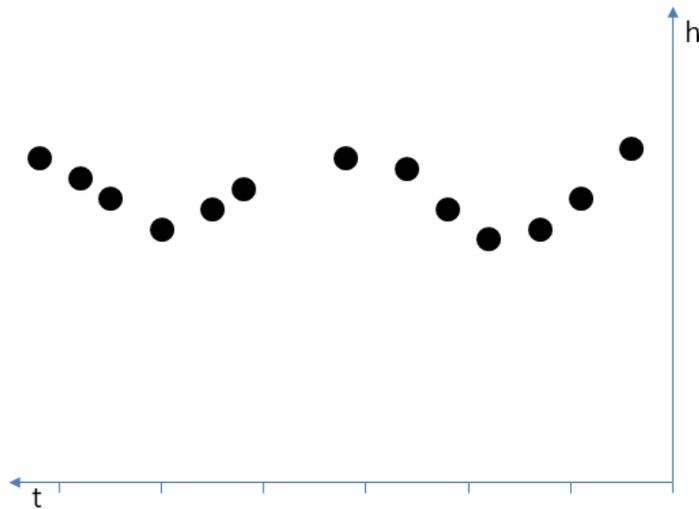


Abbildung 4.4.: 2D-Datenraum

Art der Daten und Abbildung im Datenraum

Für diese Arbeit verwenden wir zur Datenerfassung, den im Labor der Hochschule für angewandte Wissenschaften vorhandenen ART-Tracker. Ein optisches Trackingsystem, welches bereits im Grundlagenkapitel [2.3.3](#) vorgestellt wurde.

Dieses Erfassungssystem erfüllt die im Analysekapitel [3.3.3](#) beschriebenen Anforderungen an die Datenerfassung. Neben einer hohen Präzision und kleinem Messintervall, bietet uns dieses System die Möglichkeit zwei Arten von Elementen zu erfassen. Zum einen erfassen wir die Positionsdaten von einzelnen unabhängigen Messpunkten (hier genannt Marker) und zum anderen erfassen wir jeweils die Positionsdaten und darüber hinaus die Orientierung von vorher definierten Messpunktgebilden (Body, siehe [2.3.3](#)). Die Messdaten der einzelnen Elemente werden, zusammen mit Ihrer ID und dem Zeitpunkt der Messung, in einem mehrdimensionalen Datenraum abgelegt.

Zur Veranschaulichung eines solchen Datenraums ist in Grafik [4.4](#) ein beispielhafter zweidimensionaler Datenraum mit einem Messpunkt abgebildet. Die horizontale Achse stellt dabei die vergangene Zeit t dar und die vertikale Achse die gemessene Höhe eines Messpunktes h , welcher unregelmäßig in seiner horizontalen Position verändert wurde. Pro Datenart ist im Datenraum eine weitere Dimension nötig.

Die vergangene Zeit t könnte dabei theoretisch unendlich sein. Um die Komplexität zu verringern und die Werte besser verarbeiten zu können, wird das Maximum durch zwei Faktoren eingeschränkt. Zum einen tritt die jeweilige Geste nur in einem endlichen Zeitabschnitt auf, welcher durch die Start-Stop-Erkennungskomponente ermittelt wird. Alle außerhalb liegenden

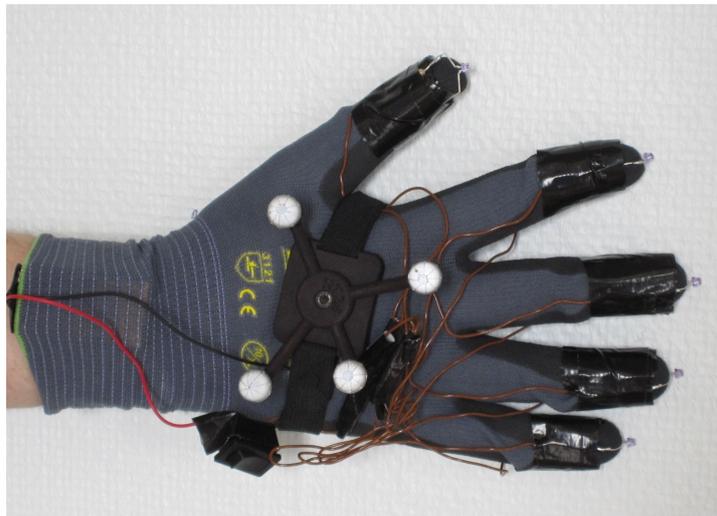


Abbildung 4.5.: Sensorhandschuh

Daten sind nicht von Interesse. Zusätzlich wird, zur Vereinfachung der Gestenerkennung, in der entwickelten Anwendung, die Größe des Intervalls welches betrachtet wird auf ein Maximum begrenzt (z.B. maximal 10 Sec.). Ältere Daten werden für die Gestenerkennung nicht mehr heran gezogen und verworfen. Dadurch sind die Gesten jedoch auf eine maximale Dauer beschränkt. Eine Erkennung von Gesten welche länger als t dauern ist nicht mehr möglich. Bei der Interaktion mit dem System werden vor allem kurze prägnante Gesten eingesetzt, dadurch tritt ein Problem mit der Überschreitung des Zeitmaximums nur in seltenen Fällen auf.

Für das Tracking werden, bei dem von uns eingesetzten ART-Tracker System, Marker an allen zu erfassenden Positionen der Hand benötigt. Zu diesem Zweck wurde ein einfacher Handschuh entwickelt, welcher sechs einzelne Punkte der Hand erfasst. Zum einen ist an jeder Fingerspitze jeweils eine aktive Infrarot-LED befestigt (im weiteren Verlauf als Marker oder Fingermarker bezeichnet), welche über eine im Handschuh integrierte 9V Blockbatterie gespeist wird. Als zweites ist ein Konstrukt aus passiven Markern (im weiteren Verlauf als Body bzw. Handbody bezeichnet) am Handrücken befestigt. Die aktiven LED's dienen zur Erfassung der Position der einzelnen Fingerspitzen. Der Handbody dient zur Erfassung des Drehsinns und der Position der kompletten Hand. Abbildung 4.5 zeigt den entwickelten Handschuh mit Marker IR-LED's und Handbody.

Der Handbody könnte dabei auch mittels genaueren aktiven IR-LED's realisiert werden, jedoch wurde, in der prototypischen Umsetzung, zur Arbeitserleichterung auf ein Standardbody der Firma A.R.T GmbH zurück gegriffen.

Das ART-Tracker System erfasst die Marker der Finger und den Body der Hand. Das System liefert für die Marker an den Fingerspitzen die Messpunkt ID, den Messzeitpunkt und die Po-

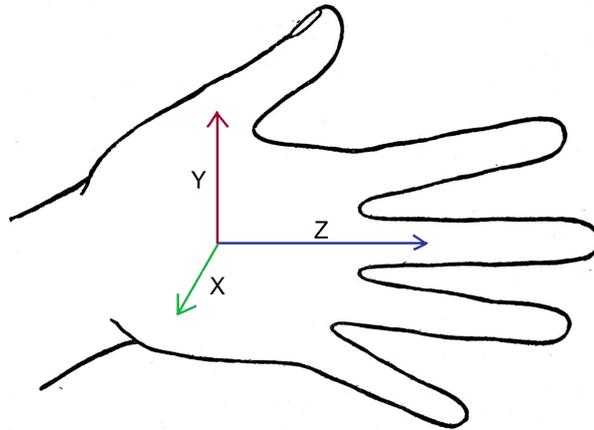


Abbildung 4.6.: Koordinatensystem des Bodys

sition im dreidimensionalen Raum (ID, i , x , y , z). Beim Body liefert das System die ID, den Messzeitpunkt, die Position im Raum und die Drehorientierung des Bodys (ID, i , x , y , z , u , v , w). Das für den Drehsinn des Handbodys definierte Koordinatensystem, relativ zur Hand, zeigt [Abbildung 4.6](#).

Diese erfassten Daten bilden die Datenbasis für unser Modell der Hand. Die Informationen von Fingermarkern und Handbody werden nach jeder Messung vom Trackingsystem an die Datenerfassungskomponente geliefert. Die Datenerfassungskomponente trägt die neuen Informationen in den Datenraum ein und reicht diesen an die Merkmalsextraktion zur Auswertung weiter.

4.2.3. Merkmalsextraktion

Die Merkmalsextraktion ist eines der wichtigsten Bestandteile des Systems. Hierbei werden die aus der Datenerfassung erhobenen Daten für eine Klassifikation aufbereitet, zusammengefasst und transformiert. Die wichtigen Informationen, welche für die spätere Klassifikation der Gesten von Bedeutung sind, werden hier als Feature extrahiert.

Es werden bei der Merkmalsextraktion für unsere Gesten passende Features gesucht. Diese leiten sich aus dem von uns gebildeten Modell der Hand ab (Modellbildung für Bildmustererkennung siehe [Duda u. a. \(2001\)](#)). Gesucht werden vor allem Features, welche bei unterschiedlich durchgeführten Gesten der selben Gestenklasse den gleichen Wert haben. Ein Beispiel ist die Erkennung der Gestenklasse „Handbewegung in eine bestimmte Richtung“. Die genaue Position des Messpunktes relativ zum Koordinatenursprung ist bei dieser Geste nicht von Interesse. Es wird lediglich die Bewegungsrichtung der Hand benötigt. Aus den jeweiligen von der Datenerfassung gelieferten absoluten Positionsdaten wird deshalb die relative Positionsveränderung

als Feature extrahiert. Dieses Feature ist invariant für alle Gesten dieser Gestenklasse. Egal wo die Startposition der konkreten Geste ist, die Bewegungsrichtung ist bei allen Gesten der Gestenklasse die selbe. Für uns wichtige Invarianten sind hierbei die Translationsinvarianz, die Rotationsinvarianz, die Skalierungsinvarianz und in einigen Fällen die Zeitinvarianz. Ziel der Merkmalsextraktion ist es, Merkmale der konkreten Geste in allgemeine Features der Gestenklasse zu transformieren.

Darüber hinaus sollten die Features unseres Modells so gewählt werden, dass dessen Werte für die Gesten der gemeinsamen Gestenklasse gleich sind und sich von anderen Gestenklassen klar unterscheiden.

Mit Hilfe der extrahierten Features entsteht so ein Merkmalsvektor. Er beinhaltet alle bei einer Messung aufgenommenen Features. Dieser Vektor wird dem jeweiligen Klassifikator zur Klassifikation übergeben. Alle möglichen Merkmalsvektoren des Systems zusammen werden als Merkmalsraum bezeichnet und bilden so alle, für die Klassifikation möglichen, Eingabewerte ab.

Eine gute Merkmalsextraktion und damit verbunden, eine starke Reduktion und Aufbereitung der Informationen sorgt dafür, dass der jeweilige Klassifikator recht simpel aufgebaut werden kann. Während eine oberflächliche Merkmalsextraktion es erforderlich macht, dass der anschließende Klassifikator komplex aufgebaut wird, um das Modell korrekt abzubilden ([Duda u. a. \(2001\)](#); [Bishop \(2006\)](#)).

Für die Auswahl der richtigen Features müssen erst einmal die Gestenklassen beschrieben werden, welche unser System erkennen soll. Daraus leiten wir die jeweiligen Features mit ihren Invarianten ab. Die Auswahl der richtigen Features unseres Modells ist nicht trivial, Kriterium hierbei ist jedoch, dass sie leicht aus den Daten zu extrahieren, robust gegen Messfehler und gut zum Unterscheiden von unterschiedlichen Gestenklassen sind. Zur Veranschaulichung werden im Abschnitt [4.2.4](#) die Features für eine beispielhafte Geste ausgewählt und die Klassifikationsstrategie beschrieben. Im Anhang [A](#) wird eine komplette Übersicht über alle hier verwendeten Gesten, deren Features und der für diese Geste zugrunde liegenden Klassifikationsstrategie dargestellt.

4.2.4. Klassifikation

Die Klassifikation ist bei guter Informationslage relativ einfach. Wenn für jeden möglichen Merkmalsvektor im Merkmalsraum seine Klassenzugehörigkeit bekannt ist, ist es leicht auf die Klassenzugehörigkeit zu schließen. In der Praxis ist die damit verbundene Abbildungsfunktion jedoch gar nicht bzw. nur unvollständig bekannt oder zu komplex um eine exakte Zuordnung durchführen zu können. Daher bedient man sich unterschiedlichster Klassifikationsverfahren,

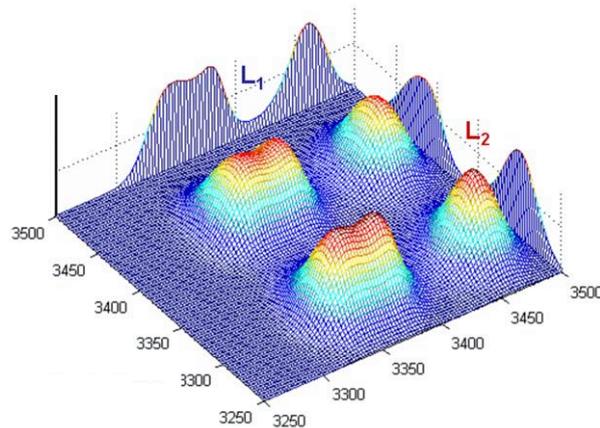


Abbildung 4.7.: Wahrscheinlichkeitsverteilung Gaußfunktion

um bei zu geringer Informationslage oder zu komplexen Abbildungsfunktionen trotzdem zu einem brauchbaren Resultat zu gelangen. Eine Liste gängiger Klassifikationsverfahren und Ihrer Eigenschaften finden Sie unter [Meisel \(2010\)](#) sowie unter [Bishop \(2006\)](#).

Wichtig ist hier, dass die einzelnen Klassifikationsverfahren einen Wahrscheinlichkeitsraum für jede Gestenklasse aufbauen und aufgrund diesem den eingehenden Merkmalsvektor der Wahrscheinlichkeit der jeweiligen Gestenklasse zuordnen.

Ein gängiges und auch hier genutztes Verfahren ist, dass vorher einzelne Merkmalsvektoren ermittelt werden, welche der jeweiligen Klasse zugehören. Diese bilden die Referenzvektoren der Gestenklasse. Mittels einer Abstandsfunktion wird der Abstand des zu prüfenden Merkmalsvektor zu diesen Referenzvektoren bestimmt. Die Wahrscheinlichkeit der Klassenzugehörigkeit sinkt nun je größer der Abstand zu den Referenzvektoren wird.

Abbildung 4.7 zeigt einen solchen Wahrscheinlichkeitsraum mit 6 Repräsentanten der Gestenklasse und der Nutzung der Gaußfunktion um ein sanftes Abfallen der Wahrscheinlichkeit mit steigendem Abstand zu erreichen.

Für unsere Anwendung bedienen wir uns eines solchen Klassifikationsverfahrens, welches mittels definierten Stützvektoren und daran knüpfenden Wahrscheinlichkeitsfunktionen eine Aussage über die Zugehörigkeit eines Merkmalsvektors zu einer Gestenklasse trifft. Dieses soll in folgendem Abschnitt anhand eines Beispiels veranschaulicht werden. Es wird stellvertretend eine in unserer Anwendung verwendete Geste genommen und an dieser das Verfahren der Klassifikation gezeigt. Die in unser Anwendung darüber hinaus verwendeten Gesten und ihre Klassifikationsstrategie wird im Anhang A erläutert.

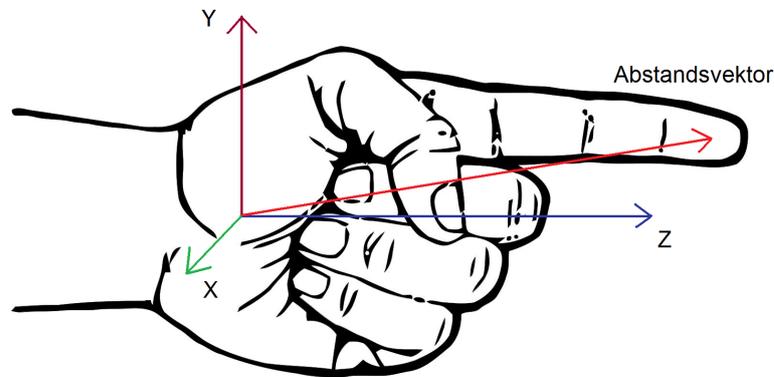


Abbildung 4.8.: Abbildung der Zeigergeste

Zeigergeste

Die Funktion der Zeigergeste innerhalb der Anwendung wird im Abschnitt [3.4.1](#) Drag-and-Drop Interaktionstechniken genauer beschrieben. In diesem Abschnitt werden die genutzten Features und das verwendete Klassifikationsverfahren genauer beschrieben.

Die Geste ist aktiv, wenn maximal ein Finger der Hand nach vorne ausgestreckt wird (Abbildung [4.8](#)).

Als Feature werden für diese Geste die Länge l_i des Abstandsvektors zwischen Handbody und jedes erkannten Fingermarkers ermittelt. Zusätzlich wird der dazugehörige Winkel α_i zwischen der Z-Achse des Handbodys und des jeweiligen Abstandsvektors bestimmt. Diese Features dienen als Ausgangsbasis zur Bestimmung des Wahrscheinlichkeitswertes ob maximal ein Finger nach vorne gerichtet ist und so die Zeigergestehaltung eingenommen wurde. [Abbildung 4.8](#) zeigt die Zeigergestehaltung mit dem Handbodykoordinatensystem und einem Abstandsvektor.

Die prototypische Umsetzung hat gezeigt, dass der Handbody auf der Handoberseite nicht mittig, sondern etwas versetzt liegt und dieses die Berechnung des Abstandsvektors etwas verfälscht. Daher wird vor der Berechnung des Abstandsvektors die Position des Handbodys um einen konstanten, vorher mittels Kalibrierung bestimmten Vektor auf der X- und Y-Achse verschoben.

Die zusammen gehörenden Features werden nun zu einem Merkmalsvektor $M_i(l_i; \alpha_i)$ zusammen gefasst und mit den Referenzvektoren verglichen.

Die zu der Gestenklasse gehörenden Referenzvektoren werden bei diesem Klassifikator vorher manuell festgelegt:

Alle Vektoren, bei welcher die Länge des Abstandsvektors l_i über der Länge l_{min} liegt und der Winkel α genau 0 Grad beträgt, stellen unsere Referenzvektoren dar. Bei diesen Vektoren wird die Wahrscheinlichkeit, dass es sich um die gesuchte Geste handelt, auf hundert Prozent gesetzt.

Die Anzahl der hier vorher definierten Referenzvektoren ist für die Erkennung relativ gering. Sie führen zu einer geringen Erkennungsquote und zu einem starken Gefälle bei der Wahrscheinlichkeitsbestimmung. (Es gibt nur 0% oder 100% Zugehörigkeit zur Gestenklasse.) Der Benutzer müsste genau eine der Längen und genau den Winkel treffen, damit die Geste akzeptiert wird. Das ist in der Praxis nur schwer möglich. Darüber hinaus unterliegt unser Modell, unsere Berechnungsmethode, sowie die eigentliche Messung, einer gewissen Ungenauigkeit, was zusätzlich zu einer geringeren Erkennungsquote führt. Um diese Punkte mit einzubeziehen und eine gewisse Toleranz bei der Eingabe durch den Benutzer zu erlauben, wird angenommen, dass die Wahrscheinlichkeit im Umkreis der Referenzvektoren nicht abrupt auf Null fällt, sondern kontinuierlich mit steigendem Abstand zu den Referenzvektoren abnimmt.

Einige Klassifikatoren gestalten diese Abnahme linear (Fuzzy-Klassifikator Abbildung 4.9). Andere verteilen die Wahrscheinlichkeit mittels einer Gauß-Glockenkurve (Abbildung 4.10), um eine sanftere Abstufung zu erreichen.

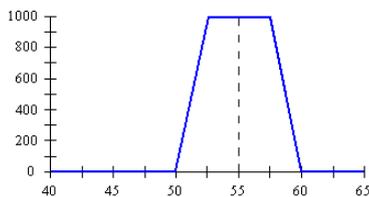


Abbildung 4.9.: Fuzzy-Funktion

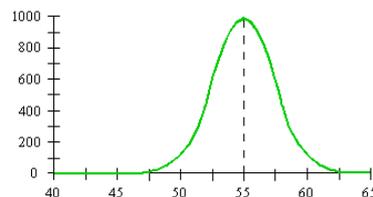


Abbildung 4.10.: Gauß-Funktion

Jede dieser Klassifikationssysteme hat Ihre spezifischen Vor- und Nachteile. Jedoch haben sie alle gemeinsam, dass der harte Wechsel zwischen 0% und 100% mittels einer Funktion abgeschwächt wird.

In unserem Klassifikator nutzen wir eine Funktion auf Basis von quadratischen Gleichungen, um eine nicht lineare Abnahme zu erreichen. Diese ist für unsere Zwecke ausreichend und leicht umzusetzen. Wir erhöhen also mit unserer Funktion die Erkennungswahrscheinlichkeit in der Umgebung unserer Referenzvektoren. Je nachdem wie tolerant die Anwendung ist, akzeptiert sie mit dieser Technik auch in der Nähe liegende Merkmalsvektoren und die dazugehörigen Gesten als Gesten dieser Gestenklasse.

Folgende Funktion dient zur Berechnung der Wahrscheinlichkeit der Zeigergeste:

$$f_i(l_i, \alpha_i) := \left(\left(\begin{array}{ll} 1 & l_{\min} < l_i \\ \frac{l_i^2}{l_{\min}^2} & \text{otherwise} \end{array} \right) \left(\left(\begin{array}{ll} 0 & \alpha_{\max} < |\alpha_i| \\ -\frac{\alpha_i^2}{\alpha_{\max}^2} + 1 & \text{otherwise} \end{array} \right) \right)$$

l_{\min} und α_{\max} sind hierbei Stellparameter, mit denen die Toleranz der Erkennung eingestellt werden kann. Ein verringern von l_{\min} führt zur Akzeptanz von kürzeren bzw. nicht komplett ausgestreckten Fingern als Zeiger. Ein erhöhen von α_{\max} führt dazu, dass Finger akzeptiert werden, welche nicht exakt parallel zur Hand-Z-Achse ausgerichtet wurden.

Die Werte für l_{\min} und α_{\max} werden in unserem System mittels Schätzen und Training auf den Benutzer und die gewünschte Toleranz angepasst. Ein möglicher resultierender Funktionsgraph mit $l_{\min} = 100$ und $\alpha_{\max} = 15$ Grad wird in Abbildung 4.11 dargestellt.

Die Berechnung der einzelnen Wahrscheinlichkeit findet für alle fünf Finger separat statt.

Da in unserem Modell der Zeigergeste nur ein Finger gestreckt sein soll, wird als zusätzliche Bedingung überprüft, ob nur ein Wahrheitswert über einem vorher definiertem Schwellwert f_{\min} liegt. Sollte keiner oder mehr als ein Wahrheitswert f_i über diesem Wert liegen, wird die resultierende Wahrscheinlichkeit auf 0 gesetzt. Wenn nur ein Wahrheitswert darüber liegt, wird dieser einzelne Wahrscheinlichkeitswert an die Anwendungsschnittstelle als Ergebnis der jeweiligen Klassifikation weitergeleitet.

Das Trackingsystem ist nicht in der Lage die einzelnen hier benutzen Fingermarker am Handschuh den jeweiligen Fingern zuzuordnen. Durch unseren Algorithmus wird die Geste immer erkannt wenn maximal ein Finger der Hand gestreckt ist, unabhängig davon welcher Finger. Die Geste ist also auch beispielsweise mit dem Mittelfinger möglich. Dieses wird für diese Anwendung akzeptiert, denn solch eine Durchführung der Geste ist zwar ungewöhnlich, führt jedoch zum eigentlichen Ziel der Interaktion, nämlich dem Zeigen auf einen Punkt mit einem Finger.

Bei ersten Tests, mit einer prototypischen Implementierung, wurde die Möglichkeit einen anderen Finger zum Zeigen verwenden zu können, vom Benutzer, nicht einmal bemerkt.

Die gezeigte Darstellung des Klassifikationsalgorithmus für die Zeigergeste soll das Grundprinzip unseres Klassifikationsverfahrens veranschaulichen. Weitere Informationen über die verwendeten Features und Klassifikationsstrategien der anderen Gesten sind im Anhang A aufgeführt.

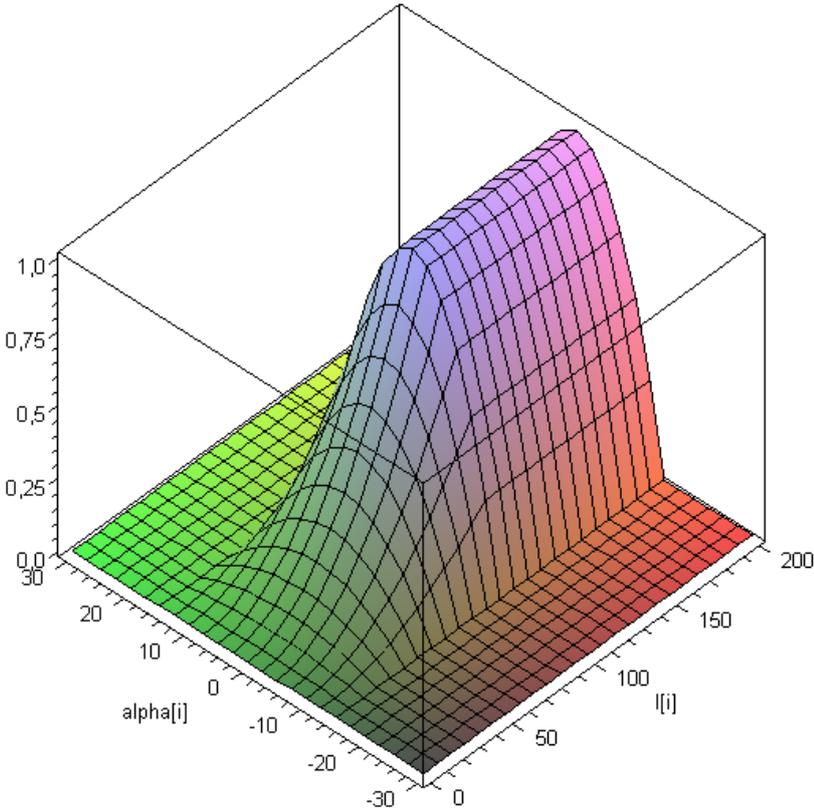


Abbildung 4.11.: Wahrscheinlichkeitsverteilung Zeigergeste

4.2.5. Anwendungsschnittstelle

Die Anwendungsschnittstelle dient der einheitlichen Kommunikation zwischen Gestenerkennungssystem und der Anwendung. Diese Schnittstelle macht die Komponente der Gestenerkennung austauschbar. Dadurch ist die Nutzung eines alternativen Gestenerkennungssystem in der Anwendung möglich oder die Nutzung des hier entwickelten Erkennungssystem in anderen Anwendungen.

Die Realisierung einer eindeutigen und verlässlichen Gestenerkennung ist meistens zu langsam oder zu komplex, daher beschränken sich die gängigen Klassifikationssysteme auf die Gewinnung der Aussage über die Wahrscheinlichkeit, ob eine bestimmte Geste auftritt. Sie wandeln diese über einen einfachen Schwellwert in eine Ja-/Nein-Aussage um (z.B. Wahrscheinlichkeit $>80\%$ Ja, $\leq 80\%$ Nein) und liefern diese Aussage an die Anwendung. In dieser Arbeit soll eine andere Vorgehensweise verwendet werden.

Die hier realisierte Schnittstelle liefert keine reine Ja/Nein Aussage, sondern die in der Klassifikation errechnete Wahrscheinlichkeit an die Anwendung. Dadurch ist es der Anwendung möglich, unter Einbeziehung seines eigenen internen Zustandes, die Klassifikation zu bewerten. Ein Beispiel ist hier die Durchführung einer Drag-and-Drop Aktion. Die Anwendung erwartet als erstes eine Zeigergeste. Wenn die Gestenerkennung eine Zeigergeste mit einer niedrigen Wahrscheinlichkeit (z.B. 60%) erkennen, so akzeptiert sie diese vielleicht trotzdem, auch wenn andere Gesten eine höhere Wahrscheinlichkeit besitzen. Die Anwendung hebt die Wahrscheinlichkeit für von ihr erwartete Gesten an und senkt die Wahrscheinlichkeit aller anderen Gesten ab, um so eine bessere Aussage treffen zu können. Dieses Verfahren soll die Qualität der Erkennung in der Praxis erhöhen und eine bessere Erkennungsquote ermöglichen.

Wenn die Anwendung die Geste akzeptiert, meldet sie dieses an die Anwendungsschnittstelle zurück. Diese liefert der Anwendung zur konkreten Geste gehörende spezifische Attribute. Auf Attribute wurde bereits im Grundlagenkapitel [2.4.6](#) eingegangen. Das Attribut der Zeigergeste ist in unserer Anwendung zum Beispiel die X-Y-Koordinate auf einem virtuellen Abbild der Powerwall, auf welche der Benutzer mit der Zeigergeste zeigt.

4.3. Abschluss

Im Designkapitel wurde die allgemeine Architektur der Software und im speziellen die Architektur und Umsetzungsmöglichkeit der Gestenerkennung vorgestellt. Es wurde Anhand der „Zeigergeste“ eine mögliche Variante der Merkmalsextraktion und dazugehöriger Klassifikation aufgezeigt.

Die konsequente Nutzung des MVC-Patterns macht es dabei möglich, flexibel einzelne Komponenten auszutauschen.

Innerhalb des Controllers wurde die Programmlogik für die Funktionen des Katalogs und des Mediaplayer dabei klar von der Komponente der eigentlichen Benutzereingabe getrennt. Diese Trennung ermöglicht eine flexible Nutzung von unterschiedlichen Systemen zur Eingabe. Neben der hier verwendeten Eingabe mittels Gesten, könnte auch die Eingabe durch Sprache oder der klassischen Tastatur genutzt werden. Das System erlaubt es so, die Gestenerkennung neben anderen Eingabeformen zu betreiben. Die Gestenerkennungskomponente arbeitet mit einer aufeinander aufbauenden Mehrschicht-Architektur. Die einzelnen Schichten sind größtenteils voneinander unabhängig konzipiert und erlauben es so, einzelnen Elemente einfach durch äquivalente zu ersetzen.

Im Gegensatz zu klassischen Klassifikationssystemen liefert die Gestenerkennungskomponente keine reine Ja/Nein Aussage an die angeschlossene Anwendung, sondern einen Wahrscheinlichkeitswert in Prozent. Dadurch ist es der Anwendung möglich, abhängig von ihrem internen Zustand flexibel die jeweilige Klassenzugehörigkeit zu bewerten. Dieses führt zu einer Erhöhung der Qualität der jeweiligen Erkennung.

5. Zusammenfassung

Diese Arbeit hatte zum Ziel, eine Plattform zu entwerfen, mit dessen Hilfe weitere Fragestellung aus dem Bereich „Gesten getriebener Mensch-Maschine-Interaktion“ bearbeitet und beantwortet werden können. Zu diesem Zweck wurde ein Softwaresystem entwickelt, welches auf Basis eines fiktiven Anwendungsszenario eine Steuerung mit der Hilfe von Handgesten ermöglicht. Es wurden beispielhafte Interaktionstechniken und Gesten für die spezifische Applikation entworfen. Eine mögliche Architektur wurde entwickelt und vorgestellt. Speziell ist dabei auf die Technik zur Erkennung und Klassifizierung von Gesten eingegangen worden. Dieses wurde anhand einer Beispielgeste näher erläutert.

5.1. Fazit

Die schnell voran schreitende Entwicklung im Bereich der Gestenerkennung, die ständig steigende Akzeptanz dieser neuen Eingabetechnik und die Vorteile der natürlichen Interaktion, lassen vermuten, dass sich die Steuerung von Computersystemen mittels Gesten in immer mehr Anwendungsbereichen etablieren wird. Diese Arbeit soll hierzu einen Beitrag leisten.

Auch wenn die Frage, aus dem Motivationsabschnitt, ob gestenbasierte Steuerung das Computerinterface der Zukunft darstellt, an dieser Stelle nicht beantwortet werden konnte, so bietet das entwickelte und vorgestellte System die Möglichkeit, diese und andere Fragen gezielt zu bearbeiten. Das entwickelte System dient dabei als Einstiegsplattform, mit der Fragestellungen wie beispielsweise die Akzeptanz von natürlichen und unnatürlichen Gesten oder die Präzision von Gesteneingaben im Praxisumfeld gezielt behandelt werden können.

Die entwickelte Applikation stellt zu diesem Zweck dem Benutzer einen Grundstock an Gesten und Interaktionstechniken zur Verfügung, mit dessen Hilfe ein Großteil bekannter und teilweise auch neue Usecases abgedeckt werden können. Die Anwendung bietet dem Nutzer die Möglichkeit, Informationen auf einem großen hochauflösenden Bildschirm zu sichten und zu bearbeiten. Die Steuerung des Systems kann dabei ausschließlich über Handgesten des Anwenders erfolgen.

5.2. Ausblick

Einige Fragestellungen, die sich aus diesem entwickelten Softwaresystem ergeben, sollten in weitergehenden Arbeiten geklärt werden. Dazu zählt unter anderem die Frage wie schnell und effizient die hier entwickelte Applikation bei der praktischen Anwendung ist und ob der Nutzer diese Art der Eingabe für sich akzeptiert.

Auch andere interessante Fragestellungen bedürfen noch einer genaueren Betrachtung. Diese Arbeit untersuchte nur die Gesteneingabe mit einer Hand. Noch mehr Möglichkeiten der Eingabe würden entstehen, wenn der Benutzer beide Hände, den Kopf oder andere Körperteile zur Eingabe nutzen könnte.

Zur Vereinfachung wurde bei der Eingabe als alleiniges Interaktionsmittel die Gesteneingabe genutzt. Jedoch könnte die Applikation um die Möglichkeit der multimodalen Interaktion erweitert werden. Beispielsweise würde eine Kombination aus Spracheingabe und Gestik die Eingabemöglichkeiten des Benutzers enorm erweitern. Dadurch wäre es dem Leitstandmitarbeiter in unserem Szenario beispielsweise möglich, auf ein Objekt zu zeigen und seinen Manipulationswunsch verbal zu äußern.

Diese Arbeit konzentrierte sich auf die Gesteneingabe von einem Nutzer. In vielen Szenarien, wie bei unserem Leitstand, sind Konstellationen denkbar, in welcher mehrere Anwender gemeinsam und kooperativ, mittels Gestensteuerung, dasselbe System bedienen. Die dabei auftretenden Konflikte für die Applikation und die dazu gehörige Gestenerkennung bedürfen zusätzlich einer genaueren Betrachtung.

Das hier gewählte Szenario und die dafür entwickelte Anwendung ist nur ein kleiner Teil von möglichen Anwendungsgebieten für GEstensteuerung. Sei es als universelle Eingabeunterstützung für Hörgeschädigte oder als einfache Steuerung von Elektrogeräten (z.B. Fernseher). Viele Bereiche sind denkbar, in denen dreidimensionale Gestensteuerung eine gute Ergänzung oder gar Alternative darstellen kann.



Literaturverzeichnis

- [ARTGmbH] : *Website A.R.T. GmbH.* – URL <http://www.ar-tracking.de/>
- [Allen u. a. 2001] ALLEN, B. D. ; BISHOP, Gary ; WELCH, Greg: *Tracking: Beyond 15 Minutes of Thought* / University of North Carolina at Chapel Hill Department of Computer Science. URL www.cs.unc.edu/~tracker/.../SIGGRAPH2001_CoursePack_11.pdf, 2001. – Forschungsbericht
- [Bishop 2006] BISHOP, Christopher M. ; BISHOP, Christopher M. (Hrsg.): *Pattern Recognition and Machine Learning.* Springer Science+Business Media, LLC, 2006
- [Boetzer 2008a] BOETZER, Joachim: *Bewegungs- und gestenbasierte Applikationssteuerung auf Basis eines Motion Trackers*, Hochschule für angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008
- [Boetzer 2008b] BOETZER, Joachim: *Gestenbasierte Interaktion mit Hilfe von Multitouch und Motiontracking*, Hochschule für angewandte Wissenschaften Hamburg, Dissertation, 2008
- [Bolt und Herranz 1992] BOLT, Richard A. ; HERRANZ, Edward: *Two-handed gesture in multi-modal natural dialog.* In: *UIST '92: Proceedings of the 5th annual ACM symposium on User interface software and technology.* New York, NY, USA : ACM, 1992, S. 7–14. – ISBN 0-89791-549-6
- [Duda u. a. 2001] DUDA, Richard O. ; HART, Peter E. ; STORK, David G. ; DUDA, Richard O. (Hrsg.): *Pattern Classification.* Wiley-Interscience, 2001
- [Gerling 2008] GERLING, Mirco: *Indoor-Navigationsunterstützung durch Wearable Computing für Such- und Rettungsmissionen unter eingeschränkten Sichtverhältnissen*, Hochschule für angewandte Wissenschaften, Masterarbeit, 2008
- [Heitsch 2008] HEITSCH, Johann: *Ein Framework zur Erkennung von dreidimensionalen Gesten*, Hochschule für angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008
- [Ishii u. a. 1994] ISHII, Hiroshi ; KOBAYASHI, Minoru ; ARITA, Kazuho: *Iterative design of seamless collaboration media.* In: *Commun. ACM* 37 (1994), Nr. 8, S. 83–97. – ISSN 0001-0782

- [Kaplan 2009] KAPLAN, Frederic: Are gesture-based interfaces the future of human computer interaction? In: *ICMI-MLMI '09: Proceedings of the 2009 international conference on Multimodal interfaces*. New York, NY, USA : ACM, 2009, S. 239–240. – ISBN 978-1-60558-772-1
- [Meisel 2010] MEISEL, Andreas: *Vorlesungsskript: Robot Vision*. 2010. – URL http://www.informatik.haw-hamburg.de/wp_robot_vision.html
- [Micire u. a. 2009] MICIRE, Mark ; DESAI, Munjal ; COURTEMANCHE, Amanda ; TSUI, Katherine M. ; YANCO, Holly A.: Analysis of natural gestures for controlling robot teams on multi-touch tabletop surfaces. In: *ITS '09: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. New York, NY, USA : ACM, 2009, S. 41–48. – ISBN 978-1-60558-733-2
- [Miller u. a. 2009] MILLER, Frederic P. ; VANDOME, Agnes F. ; MCBREWSTER, John: *Architektur- und Entwurfsmuster: Architekturmuster, Model View Controller, Presentation-Abstraction-Control, Pipes and Filters, Schichtenarchitektur, ... Mikrokern, Reflexion*. Alphascript Publishing, 2009. – 132 S
- [Neale u. a. 2004] NEALE, Dennis C. ; CARROLL, John M. ; ROSSON, Mary B.: Evaluating computer-supported cooperative work: models and frameworks. In: *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2004, S. 112–121. – ISBN 1-58113-810-5
- [Noack 2007] NOACK, Kai K.: *Leitfaden zum Motion Capturing - Theorieteil*, FHTW Berlin, Bachelorarbeit, 2007. – URL http://firsttai.de/ext/mocap/Ein-Leitfaden-zum-Motion-Capturing-MoCap-Guide_KaiKajusNoack.pdf
- [Pavlocic u. a. 1997] PAVLOCIC, Vladimier I. ; SHARMA, Rajeev ; HUANG, Thomas S.: Interpretation of Hand Gestures for Human-Computer Interaction: A Review. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 1997
- [Rahimi und Vogt 2008] RAHIMI, Mohammadali ; VOGT, Matthias: *Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie*, Hochschule für angewandte Wissenschaften Hamburg, Dissertation, 2008
- [Roßberger 2008] ROSSBERGER, Philipp: *Physikbasierte Interaktion in kollaborativen computergestützten Umgebungen*, Hochschule für angewandte Wissenschaften, Masterarbeit, 2008
- [Scherfgen und Herpers 2009] SCHERFGEN, David ; HERPERS, Rainer: 3D tracking using multiple Nintendo Wii Remotes: a simple consumer hardware tracking approach. In: *Future Play '09: Proceedings of the 2009 Conference on Future Play on @ GDC Canada*. New York, NY, USA : ACM, 2009, S. 31–32. – ISBN 978-1-60558-685-4

- [Schlömer u. a. 2008] SCHLÖMER, Thomas ; POPPINGA, Benjamin ; HENZE, Niels ; BOLL, Susanne: Gesture recognition with a Wii controller. In: *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*. New York, NY, USA : ACM, 2008, S. 11–14. – ISBN 978-1-60558-004-3
- [Tetzlaff 2008] TETZLAFF, Olaf: *Bodymonitoring: Entwicklung eines Prototypen für intelligente Kleidung*, Hochschule für angewandte Wissenschaften Hamburg, Masterarbeit, 2008
- [Weiser 1999] WEISER, Mark: The computer for the 21st century. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 3 (1999), Nr. 3, S. 3–11. – ISSN 1559-1662
- [Wikipedia] WIKIPEDIA: *Beschreibung Modell-View-Controller*. – URL http://de.wikipedia.org/wiki/Model_View_Controller

A. Gestenbeschreibung

In diesem Abschnitt werden die von uns in der Analyse identifizierten Gesten näher beschrieben. Für jede Geste werden die Features definiert und die Klassifikationsstrategie aufgezeigt. Folgende fünf Gesten wurden im Analysekapitel identifiziert. Diese müssen für unsere Anwendung erkannt und ausgewertet werden.

- Zeiger
- Daumenschalter
- Finger zusammenführen
- Richtungsbewegung
- Fingerkreisbewegung

A.1. Zeigergeste

Die Zeigergeste wurde bereits im Abschnitt 4.2.4 Klassifikation beschrieben. Es soll hier nur der Vollständigkeit halber noch einmal die verwendeten Feature, die Wahrscheinlichkeitsfunktion und ergänzend die für diese Geste erstellten Attribute beschrieben werden.

Als Feature werden für diese Geste die Länge l_i des Abstandsvektors zwischen Handbody und jedes erkannten Fingermarkers ermittelt. Zusätzlich wird der dazugehörige Winkel α_i zwischen der Z-Achse des Handbodys und des jeweiligen Abstandsvektors bestimmt. Diese Features dienen als Ausgangsbasis zur Bestimmung des Wahrscheinlichkeitswertes ob maximal ein Finger nach vorne gerichtet ist und so die Zeigergestehaltung eingenommen wurde. Abbildung A.1 zeigt die Zeigergestehaltung mit dem Handbodykoordinatensystem und einem Abstandsvektor.

Die prototypische Umsetzung hat gezeigt, dass der Handbody auf der Handoberseite nicht mittig, sondern etwas versetzt liegt und dieses die Berechnung des Abstandsvektors etwas verfälscht. Daher wird vor der Berechnung des Abstandsvektors die Position des Handbodys um einen konstanten, vorher mittels Kalibrierung bestimmten Vektor auf der X- und Y-Achse verschoben.

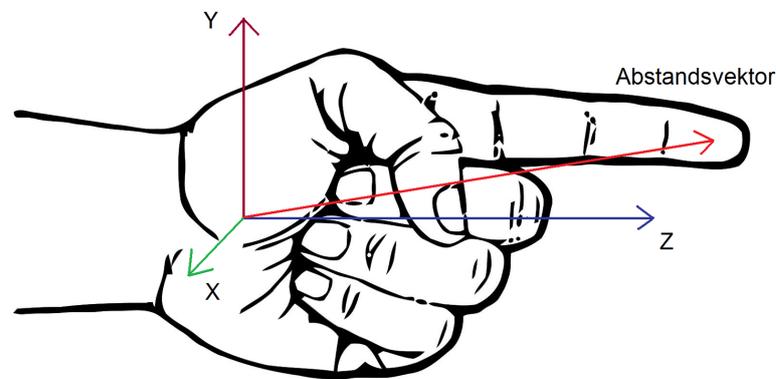


Abbildung A.1.: Abbildung der Zeigergeste

Die zusammengehörenden Features werden nun zu einem Merkmalsvektor $M_i(l_i; \alpha_i)$ zusammengefasst und mit den Referenzvektoren verglichen.

Die zu der Gestenklasse gehörenden Referenzvektoren werden bei diesem Klassifikator vorher manuell festgelegt:

Alle Vektoren, bei welcher die Länge des Abstandsvektors l_i über der Länge l_{min} liegt und der Winkel α genau 0 Grad beträgt, stellen unsere Referenzvektoren dar. Bei diesen Vektoren wird die Wahrscheinlichkeit, dass es sich um die gesuchte Geste handelt, auf hundert Prozent gesetzt.

Die Anzahl der hier vorher definierten Referenzvektoren ist für die Erkennung relativ gering. Sie führen zu einer geringen Erkennungsquote und zu einem starken Gefälle bei der Wahrscheinlichkeitsbestimmung. (Es gibt nur 0% oder 100% Zugehörigkeit zur Gestenklasse.) Der Benutzer müsste genau eine der Längen und genau den Winkel treffen, damit die Geste akzeptiert wird. Das ist in der Praxis nur schwer möglich. Darüber hinaus unterliegt unser Modell, unsere Berechnungsmethode, sowie die eigentliche Messung, einer gewissen Ungenauigkeit, was zusätzlich zu einer geringeren Erkennungsquote führt. Um diese Punkte mit einzubeziehen und eine gewisse Toleranz bei der Eingabe durch den Benutzer zu erlauben, wird angenommen, dass die Wahrscheinlichkeit im Umkreis der Referenzvektoren nicht abrupt auf Null fällt, sondern kontinuierlich mit steigendem Abstand zu den Referenzvektoren abnimmt.

In unserem Klassifikator nutzen wir eine Funktion auf Basis von quadratischen Gleichungen, um eine nicht lineare Abnahme zu erreichen. Diese ist für unsere Zwecke ausreichend und leicht umzusetzen. Wir erhöhen also mit unserer Funktion die Erkennungswahrscheinlichkeit

in der Umgebung unserer Referenzvektoren. Je nachdem wie tolerant die Anwendung ist, akzeptiert sie mit dieser Technik auch in der Nähe liegende Merkmalsvektoren und die dazugehörigen Gesten als Gesten dieser Gestenklasse.

Folgende Funktion dient zur Berechnung der Wahrscheinlichkeit der Zeigergeste:

$$f_i(l_i, \alpha_i) := \left(\left[\begin{array}{ll} 1 & l_{\min} < l_i \\ \frac{l_i^2}{l_{\min}^2} & \text{otherwise} \end{array} \right] \left(\left[\begin{array}{ll} 0 & \alpha_{\max} < |\alpha_i| \\ -\frac{\alpha_i^2}{\alpha_{\max}^2} + 1 & \text{otherwise} \end{array} \right] \right)$$

l_{\min} und α_{\max} sind hierbei Stellparameter, mit denen die Toleranz der Erkennung eingestellt werden kann. Ein verringern von l_{\min} führt zur Akzeptanz von kürzeren bzw. nicht komplett ausgestreckten Fingern als Zeiger. Ein erhöhen von α_{\max} führt dazu, dass Finger akzeptiert werden, welche nicht exakt parallel zur Hand-Z-Achse ausgerichtet wurden.

Die Werte für l_{\min} und α_{\max} werden in unserem System mittels Schätzen und Training auf den Benutzer und die gewünschte Toleranz angepasst. Ein möglicher resultierender Funktionsgraph mit $l_{\min} = 100$ und $\alpha_{\max} = 15$ Grad wird in Abbildung 4.11 dargestellt. Die Berechnung der einzelnen Wahrscheinlichkeit findet für alle fünf Finger separat statt.

Da in unserem Modell der Zeigergeste nur ein Finger gestreckt sein soll, wird als zusätzliche Bedingung überprüft, ob nur ein Wahrheitswert über einem vorher definiertem Schwellwert f_{\min} liegt. Sollte keiner oder mehr als ein Wahrheitswert f_i über diesem Wert liegen, wird die resultierende Wahrscheinlichkeit auf 0 gesetzt. Wenn nur ein Wahrheitswert darüber liegt, wird dieser einzelne Wahrscheinlichkeitswert an die Anwendungsschnittstelle als Ergebnis der jeweiligen Klassifikation weitergeleitet.

Das Trackingsystem ist nicht in der Lage die einzelnen hier benutzen Fingermarker am Handschuh den jeweiligen Fingern zuzuordnen. Durch unseren Algorithmus wird die Geste immer erkannt wenn maximal ein Finger der Hand gestreckt ist, unabhängig davon welcher Finger. Die Geste ist also auch beispielsweise mit dem Mittelfinger möglich. Dieses wird für diese Anwendung akzeptiert, denn solch eine Durchführung der Geste ist zwar ungewöhnlich, führt jedoch zum eigentlichen Ziel der Interaktion, nämlich dem Zeigen auf einen Punkt mit einem Finger.

Bei ersten Tests, mit einer prototypischen Implementierung, wurde die Möglichkeit einen anderen Finger zum Zeigen verwenden zu können, vom Benutzer, nicht einmal bemerkt.

Bei Akzeptanz der Geste werden die dazugehörigen Attribute von der Anwendung angefordert. Mittels der absoluten Position des Handbodys und dem Abstandsvektor wird eine Gerade in

den Raum gelegt und der Kollisionspunkt mit einer virtuellen Ebene, welche die Powerwall repräsentiert berechnet. Aus diesem Kollisionspunkt ergibt sich die relative X- und Y-Koordinate auf dieser Ebene. Diese errechnete Koordinate, sofern vorhanden, wird der Anwendungsschnittstelle als Parameter übergeben und gilt als Punkt auf welche die Zeigergeste zeigt.

Das Berechnungsmodell und die dazu gehörige Koordinatenberechnung auf der Powerwall verfügt über eine gewisse Ungenauigkeit. Zum Beispiel variiert die Anatomie der Hand von Benutzer zu Benutzer etwas. Diese Ungenauigkeit wird in der Anwendung mittels eines Regelkreislaufs kompensiert. Der Benutzer bekommt dazu die errechnete Position auf dem Bildschirm angezeigt und korrigiert dieses durch stetiges Bewegen von Finger und Hand. Die Ungenauigkeit wird also, wie bei unterschiedlichen Bewegungsgeschwindigkeiten einer Maus auf unterschiedlichen Untergründen, durch unterbewusstes stetiges Anpassen durch den Benutzer korrigiert.

A.2. Daumenschaltergeste

Die Daumenschaltergeste wird in unserer Anwendung zusammen mit der Zeigergeste verwendet. Daher muss Sie sich von dieser klar unterscheiden lassen können. Für diese Geste wird sich die Eigenschaft zu nutze gemacht, dass nur der Daumen von der Handfläche nach oben weg abgespreizt werden kann. Das eigentliche Erkennungsverfahren ist wie bei der Zeigergeste prinzipiell das Selbe. Auch hier wird eine Wahrscheinlichkeitsfunktion mit einem Abstandsvektor und einem Winkel verwendet. Als Eingangsfeature wird bei dieser Geste jedoch der Winkel zwischen Abstandsvektor und **Y-Achse** des Handbody's übergeben und nicht der Winkel zwischen Abstandsvektor und der Z-Achse.

Die zur Berechnung der Wahrscheinlichkeit genutzte Formel ist mit der Formel der Zeigergeste identisch. Die Parameter l_{min} und α_{max} werden hierbei an die Werte für den kürzeren Daumen angepasst.

Das für die Geste spezifische Attribut ist nur die Aussage ob die Geste erkannt wurde oder nicht.

A.3. Finger zusammenführen

Diese Geste kann zum Greifen von Objekten, dem Wechsel in einen anderen Eingabemodus oder der Trennung unterschiedlicher Gesten dienen. Die Geste soll erkannt werden wenn die Fingerspitzen der Hand nah bei einander bzw. weit voneinander entfernt sind.

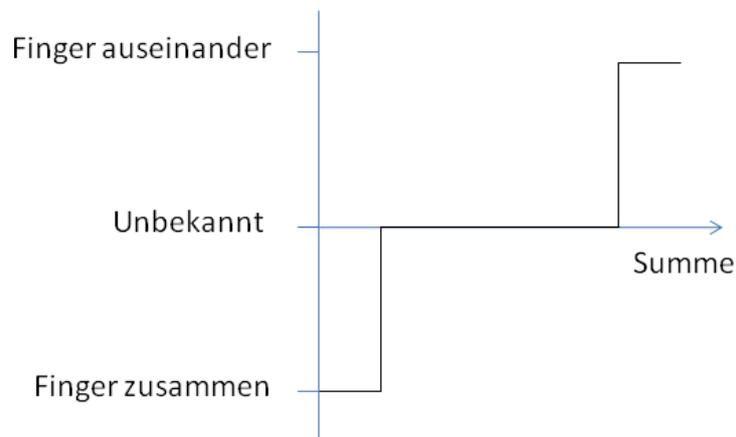


Abbildung A.2.: Treppenbildung

Als erstes werden die Abstände aller erkannten Fingermarker zueinander errechnet und die Summe gebildet. Es wird definiert, dass die Finger nah bei einander liegen wenn die Summe ein bestimmtes Minimum s_{min} unterschreitet. Die Finger liegen weit auseinander, wenn die Summe ein Maximum s_{max} überschreitet. Dazwischen liegende Werte führen zur Aussage, dass die Finger weder nah noch weit von einander entfernt sind. Die Werte für s_{min} und s_{max} müssen mittels Training und Kalibrierung für jeden Benutzer einzeln bestimmt werden.

Die daraus resultierende Wahrscheinlichkeitsfunktion führt zu einer Art Treppenverlauf. (Abbildung A.2) In den anderen Fällen haben wir diesen Verlauf mittels einer nicht linearen Funktion geglättet. In diesem Fall verzichten wir darauf, da in der Applikation eine „sanfte“ Erkennung der Geste nicht unbedingt nötig ist.

A.4. Richtungsbewegung

Diese Geste soll erkannt werden, wenn die Hand des Benutzers für eine bestimmte Zeit überwiegend in eine Richtung bewegt wird. Zum Vergleich dient dazu ein vorher definierter Referenzvektor. Eingesetzt wird diese Geste bei der Funktionsauswahl. Dort bewegt der Nutzer seine Hand in die Richtung der Funktion, welche er aufrufen möchte.

Um die Bewegungsrichtung zu Erkennen ermittelt die Anwendung immer aus zwei zeitlich folgenden Positionsdaten des Handbodys den Richtungsvektor der Hand. Der Winkel zwischen dem jeweiligen Richtungsvektor und dem Referenzvektor stellt die dazugehörige Abweichung dar. Dieser Winkel kann zwischen 0 und 180 Grad liegen. Mittels der folgenden Wahrscheinlichkeitsfunktion wird daraus die Wahrscheinlichkeit ermittelt mit der sich der Handbody in die gewünschte Richtung bewegt hat. Die jeweilige Wahrscheinlichkeit wird anschließend mit der

Länge des jeweiligen Richtungsvektors ins Verhältnis zur Gesamtlänge aller Richtungsvektoren gesetzt. Dadurch hat ein kurzer Richtungsvektor nur einen geringen, aber ein langer Vektor einen großen Einfluss auf das Endergebnis.

Die Wahrscheinlichkeitsformel für das oben erläuterte Verfahren lautet:

$$p := \frac{\sum_{i=1}^n f\left(\left(\frac{\alpha_i}{90}\right)^2 \cdot l_i\right)}{\sum_{i=1}^n l_i}$$

n ist die Anzahl zur Verfügung stehender Richtungsvektoren.

α_i ist der Winkel zwischen Referenzvektor und Richtungsvektor i (0-180 Grad)

l_i ist die Länge des Richtungsvektors i

Die Quadrierung stellt sicher, dass eine Bewegung entgegengesetzt des Referenzvektors die Wahrscheinlichkeit stärker senkt, als eine Bewegung in Richtung des Referenzvektors die Wahrscheinlichkeit erhöht. Dadurch ist eine Abweichung von der Sollrichtung im geringen Maße erlaubt, hat jedoch starken negativen Einfluss auf die Erkennungswahrscheinlichkeit.

Für die Berechnung werden alle Positionsdaten und dazugehörige Richtungsvektoren einbezogen, welche nicht länger als t Sekunden in der Vergangenheit liegen. Dadurch wird die Dauer der Geste und der Berechnungsaufwand begrenzt.

Der errechnete Wahrscheinlichkeitswert wird abschließend an die Anwendungsschnittstelle weitergeleitet.

A.5. Fingerkreisbewegung

Die Fingerkreisbewegungsgeste wird eingesetzt, um Werte in der Anwendung interaktiv einzustellen. Ein Beispiel ist die Lautstärke eines Videos.

Für diese Geste positioniert der Nutzer seine Fingerspitzen kreisförmig als ob er einen runden Drehregler halten würde. (Abbildung 3.8) Anschließend bewegt er die Fingerspitzen, auf einer Kreisbahn um einen Mittelpunkt, nach Links oder Rechts. Es wird dadurch vom Benutzer dem System kein exakter Wert mitgeteilt, sondern ein bereits vorhandener Wert erhöht oder erniedrigt.

Die Geste wird erkannt, wenn sich die einzelnen Fingerspitzen, auf einer gemeinsamen Kreisbahn, in die selbe Richtung bewegen. Zur Erkennung der Geste müssen zwei Fragen beantwortet werden. Erstens, bewegen sich die einzelnen Fingerspitzen jeweils auf einer Kreisbahn mit einem für eine Hand realistischen Durchmesser? Und zweitens, bewegen sich die einzelnen Fingerspitzen um einen gemeinsamen Mittelpunkt? Die Beantwortung dieser Fragen, in Ihrer allgemeinen Form, ist sehr komplex. Ein möglicher Ansatz ist die Nutzung einer Hough-Transformation für Kreise¹³ wie es in der Arbeit von Boetzer (2008b) beschrieben wird.

Bei der prototypischen Umsetzung mit dem Hough-Verfahren wurde festgestellt, dass der sich ergebene Rechenaufwand bei hoher Abstrakte recht hoch ist. Zur Vereinfachung wurde daher, der Radius des zu erkennenden Kreises fest definiert und die Kreisbewegung der Finger auf Bewegungen auf der globalen X-Y-Ebene beschränkt. Dieses führte zu einer starken Verbesserung der Erkennungsgeschwindigkeit.

Die ganze Erkennung muss mit einer gewissen Toleranz versehen werden, da der Nutzer nicht in der Lage ist seine Fingerspitzen in einer exakten Kreisbahn zu bewegen. Durch diese Toleranz können kleine Bewegungen auf der Kreisbahn, wie hier bei der „Kreisbewegung“-Geste, und kurze, gerade Bewegungen in eine Richtung, wie bei der „Richtungsbewegung“-Geste, nur schwer von einander unterschieden werden. Daher muss die Anwendung entweder einen Mindestradius definieren, welcher zur Erkennung der Geste führt oder das parallele Erkennen der „Kreisbewegung“-Geste und „Richtungsbewegung“-Geste ausschließen.

Wenn die „Kreisbewegung“-Geste erkannt wurde, wird als Attribut der Drehwinkel, relativ zur Anfangsposition der Geste, an die Anwendung übergeben und daraus der neu einzustellende Wert errechnet.

¹³<http://de.wikipedia.org/wiki/Hough-Transformation>

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 03. Mai 2010

Ort, Datum

Unterschrift