



Gesten- und Sprachsteuerung für einen mobilen Roboter mittels Kinect for Windows unter Java

Studienarbeit

für die Prüfung zum

Bachelor of Science

der Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

Simon Ebner, Volker Werling

Mai 2013

Bearbeitungszeitraum
Matrikelnummer, Kurs
Gutachter

12 Wochen
5837963 7012192, TAI10B2
Prof. Hans-Jörg Haubner

Erklärung

Wir erklären hiermit ehrenwörtlich:

1. dass wir unsere Studienarbeit mit dem Thema *Gesten- und Sprachsteuerung fuer einen mobilen Roboter mittels Kinect for Windows unter Java* ohne fremde Hilfe angefertigt haben;
2. dass wir die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet haben;
3. dass wir unsere Studienarbeit bei keiner anderen Prüfung vorgelegt haben;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Wir sind uns bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Karlsruhe, Mai 2013

Simon Ebner, Volker Werling

Zusammenfassung

Die Autoren Ebner und Werling haben diese Arbeit gemeinsam verfasst, wobei die Kapitel jeweils von einem der Autoren geschrieben wurden. Auf Herrn Ebner entfallen Kapitel ??, ??, ??, ?? und ?? und auf Herrn Werling die Kapitel ??, ??, ??, ?? und ??.

Inhaltsverzeichnis

1	Einleitung	6
2	Aufgabenstellung	7
2.1	Problemstellung und Ziel der Arbeit	7
2.2	Geplantes Vorgehen	8
2.3	Ausblick	9
3	Stand der Technik	10
4	Mensch-Computer-Interaktion	11
4.1	Man-Computer Symbiosis	11
4.2	Anwendung des Konzepts auf das Projekt	12
4.2.1	Die Anwendung RoCoVoMo	13
4.2.2	Eingabe- und Ausgabegeräte	13
4.2.3	Mensch - User	14
4.3	Herausforderungen der Mensch-Computer-Interaktion	15
4.4	Moderne Design Prinzipien	16
4.5	Design Methodik	17
5	Auswahl des Roboters	19
6	Lego-Framework	20
7	Konzept und Steuerung der Anwendung	21
7.1	Verwendete Gesten	21
7.1.1	Kreisbewegung	22
7.1.2	Vorwärtsbewegung	23

7.1.3	Haltesignal	23
7.1.4	Erweiterte Vorwärtsbewegung	23
7.1.5	Entriegeln — Blockieren	23
7.2	Trainingsmodul	23
7.3	Kinect—Modul	24
8	Implementierung	29
9	Ausblick - weitere Arbeiten	30
	Abbildungsverzeichnis	31
	Tabellenverzeichnis	32
	Quellcodeverzeichnis	33
	Literaturverzeichnis	34

1 Einleitung

2 Aufgabenstellung

2.1 Problemstellung und Ziel der Arbeit

Aus dem ersten Teil der Arbeit ging die Entwicklung einer Schnittstelle zwischen Mensch und Roboter hervor. Dabei wurden ein erstes Konzept für die verwendeten Gesten und Sprachbefehle, sowie eine erste Umsetzung Stufe der Softwarearchitektur der Anwendung vorgestellt. Ziel des zweiten Teils ist, mit Hilfe diesen ersten Ergebnissen eine Steuerung eines mobilen Roboters umzusetzen. Dabei werden weiterhin die Bibliotheken *jnect* zur Steuerung der Kinect mittels Java, und *jahmm* zur Nutzung der Hidden Markov Models (HMMs) verwendet. Bislang offen sind die Punkte des zu verwendenden mobilen Roboters und die konkrete Implementierung der Anwendung mit den dafür benötigten Funktionalitäten.

Daraus ergeben sich nun folgende Ziele:

- Analyse verfügbarer Ausführungen von mobilen Robotern
- Auswahl eines Robotermodells anhand gegebener Kriterien und Voraussetzungen
- Konzeption der Anwendung zur Steuerung des mobilen Roboters
- Implementierung der Anwendung
- Optimierung der Gestenerkennung (optional)
- Benutzerfreundlichkeit (Usability) der Anwendung

Dabei wird von der Problemstellung ausgegangen, dass im Rahmen dieser Studienarbeit die Wahl eines mobilen Roboters nur über vorhandene Modelle von mobilen Robotern möglich ist, die auch an der DHBW Karlsruhe verfügbar sind. Bei dieser Wahl sind auch die vorhandenen Restriktionen aus der bereits gewählten Programmiersprache, Java, zu beachten, da vermieden werden soll, mehrere Sprachen in der Entwicklung zu verwenden.

Das Kernziel, die korrekte Erkennung von Gesten und Sprache hängt von zahlreichen Faktoren ab, und benötigt ein hohes Maß an Testaufwand. Auf das Problem der Feineinstellung dieser Systeme durch empirische Ergebnisse wird eingegangen, ob das Ziel einer optimalen Lösung auch erreicht werden kann, ist ungewiss.

2.2 Geplantes Vorgehen

Aus dem ersten Teil der Studienarbeit ging unter anderem eine modular aufgebaute Rich client platform (RCP) Anwendung hervor. Da die einzelnen Module dieser Anwendung in Form von Eclipse Plugins separat verwendet werden können, kann die Arbeit an den unter 2.1 genannten Zielen, zu einem Großteil unabhängig voneinander in die Anwendung integriert werden.

Zu Beginn der Arbeit muss jedoch die Analyse und die Auswahl eines mobilen Roboters abgeschlossen werden, da eventuell auf Besonderheiten eines speziellen Robotermodells in der Anwendung gesonder eingegangen werden muss.

Nachdem also ein Modell ausgewählt wurde und die diversen Arbeiten an der Anwendung separat durchgeführt wurden, soll durch Tests und Optimierungen die Anwendung benutzerfreundlicher gestaltet werden, um eine höhere Qualität der Mensch-Computer-Interaktion zu erreichen.

2.3 Ausblick

Software die nicht weiterentwickelt wird, deren Fehler nicht behoben werden, die nicht auf Wünsche der User angepasst werden, und deren Funktionen nicht erweitert werden, werden auf kurz oder lang nicht mehr verwendet. Dies soll hier versucht werden, zu verhindern. Bereits während der Arbeiten an der Studienarbeit, steht der vorhandene Code der Anwendung unter einer kollaborativen Versionsverwaltung, wobei die Arbeit natürlich nur von den beiden Autoren durchgeführt wird. Um das Ziel dieser Studienarbeit, eine benutzerfreundliche Steuerung eines mobilen Roboters zu erstellen, über den Zeitraum der Studienarbeit weiter zu tragen, soll der Code der Anwendung im Anschluss der Studienarbeit frei verfügbar sein.

3 Stand der Technik

4 Mensch-Computer-Interaktion

Die Mensch-Computer-Interaktion ist die Lehre der Interaktion zwischen dem Menschen (dem Benutzer) und Computern. Dieser Begriff ist nicht zu verwechseln mit Mensch-Maschine-Schnittstelle oder Benutzerschnittstelle, die Teil des Fachbereichs Mensch-Computer-Interaktion sind.

Einer der Wegbegründer auf diesem Gebiet war J. C. R. Licklider. Unter der Bezeichnung *Man-Computer Symbioses* hatte er bereits 1960 Ziele und Probleme dieses Teilgebiets der Informatik postuliert [Lic68].

4.1 Man-Computer Symbiosis

Unter Man-Computer Symbiosis verstand Licklider die Entwicklung einer kooperativen Interaktion zwischen Mensch und Computer. Dabei soll eine enge Partnerschaft zwischen Mensch und den elektronischen Geräten geschaffen werden, so Licklider weiter. Vergleichbar heute mit den fortgeschrittenen Smartphonemodellen, die dem Nutzer nicht nur das Telefonieren ermöglichen, sondern auch Internetzugang bereitstellen, als Navigationssystem dienen, und als Spielekonsole fungieren, und aus diesen Gründen für viele Menschen unersetzlich sind. Weiterhin sind zwei Ziele bei einer Symbiose nach Licklider zu erreichen:

1. Computern formuliertes Denken zu ermöglichen

2. Kooperation in der Entscheidungsfindung und der Handhabung von komplexen Situationen von Mensch und Computer, ohne inflexible Abhängigkeiten auf vordefinierte Programme.

Zum damaligen Zeitpunkt bestanden einzelne Rechner aus meterlangen Schranken, die ganze Räume umfassten und über spezielle Terminals gesteuert wurden, die dazu noch einen Bruchteil der Rechenleistung heutiger Mobilgeräte besitzen. Durch den rasanten Wandel der IT-Welt sind daher nicht mehr alle Erkenntnisse für heutige Anwendungen interessant.

4.2 Anwendung des Konzepts auf das Projekt

Im Rahmen der hier dargestellten Anwendung ist es wichtig das Konzept der Man-Computer Symbiosis von dem abzutrennen, was North [?] als *mechanically extended man* bezeichnet. Dabei gibt der Benutzer alle für die Handlung entscheidenden Kriterien, wie Richtung, und Integration vor. Die mechanischen Teile stellen ausschließlich eine Erweiterung des Benutzers dar. Die Anwendung soll zwar aus einer Steuerung für einen mobilen Roboter bestehen, es sollen aber noch weitere Elemente enthalten sein, die über den Typ einer Erweiterung hinaus gehen und eine Form von Symbiose erreicht werden kann.

Ein Ziel dabei ist nach Licklider [Lic68], den Computer effektiv in Prozesse des *Denkens*, die in Echtzeit ablaufen müssen, zu integrieren. Aufgrund der Tatsache, dass Computer heute zu Tage aus mehreren Komponenten bestehen und auch zum Großteil durch ihre Peripherie bestimmt sind, die wiederum Rechenleistung des Computers zur vorgesehenen Nutzung benötigt, erweitern wir die Aussage von Licklider auf alle Komponenten, die an der Beziehung teilhaben. Diese werden im Folgenden näher betrachtet.

4.2.1 Die Anwendung RoCoVoMo

In der Anwendung *RoCoVoMo* geschieht dies durch die eingebaute Analyse der Gesten- und Sprachinformation durch HMMs. Die Komponenten Kinect und mobiler Roboter sind separate Teilnehmer der Mensch-Computer-Interaktion. Dabei unterstützt die Anwendung den Nutzer durch die stochastischen Auswertungen der Kinect-Daten und die intuitive Führung durch die Anwendung, in dem Gesten- und Spracherkennung dargestellt und entsprechend an dem mobilen Roboter weitergeleitet werden, und im Ernstfall (Blockade der Räder, fahren gegen ein Wandstück) gesonderte Routinen von der Anwendung ausgeführt werden, ohne dass der Nutzer hierbei eingreifen kann, oder sollte. Eine Beschreibung der Anwendung und weitere Details zu deren Implementierung können unter Kapitel 7 und Kapitel 8 nachgelesen werden.

4.2.2 Eingabe- und Ausgabegeräte

Kinect

Die Kinect Komponente besitzt herkömmlichen Eingabegeräten gegenüber einige klare Vorteile in Bezug auf die Mensch-Computer-Interaktion. So kann mittels Kinect, ein Teil der formulierten Denks, wie Licklider es nennt, auf den Computer übertragen werden, indem der Mensch Gesten und Sprachbefehle verwenden kann, die für den Menschen intuitiver zu handhaben sind, und darüber die Umsetzung der gewünschten Aktion dem Computer überlassen wird. Diese kann, der jeweiligen Aktion entsprechend, komplexe Funktionen umfassen, die der Mensch nicht notwendigerweise durchführen muss. Dabei wird zugleich das zweite Ziel in Lickliders Man-Computer Symbiosis erfüllt, in dem durch die Gestensteuerung unnötige Zwischenschritte, wie diverse Auswahldialoge oder gesonderte Eingabegeräte zu einem Großteil ersetzt werden, und sogleich ein hohes Maß an Kooperation zwischen Mensch und Computer erreicht werden kann.

Mobiler Roboter

Der mobile Roboter wird direkt über die Anwendung RoCoVoMo gesteuert, und durch die Kinect-Steuerung soll die Verbindung zwischen Mensch und mobilen Roboter unmittelbar gestalten. Für den Roboter bedeutet dies, ein hohes Maß an Flexibilität bezüglich Latenzzeit, Beschleunigung, und Bewegungsrichtung. Der Roboter muss unmittelbar reagieren, sobald eine Geste oder ein Sprachbefehl erkannt wird, so wie eine schnelle Bewegung gewünscht ist. Bezüglich Bewegung ist ein holonomer Roboter eine optimale Voraussetzung für eine intuitive Steuerung und einfache Gestenbefehle.

4.2.3 Mensch - User

Für eine effektive Mensch-Computer-Interaktion ist es für den Nutzer der Anwendung notwendig, einfach und schnell Gestenbefehle einzugeben, die genauso schnell in von der Anwendung in Aktionen umgesetzt werden sollen. Hierbei ist aber ebenso ein gewisses Training des Users nötig, da Gesten- und Sprachbefehle nicht die einfachste Form der Steuerung sind. Der Nutzer muss möglichst klar und ohne Dialekt Sprachbefehle geben, sowie er auch das Vokabular, dass die Anwendung umfasst, kennen muss.

Zu Beachten ist, dass Sprache eine redundante Form der Kommunikation ist, und Befehle für den untrainierten Nutzer eine andere Aktion der Anwendung suggerieren, als die in Wirklichkeit der Fall ist.

In Hinsicht auf Gesten, ist es wichtig, dass der User konzentriert und in möglichst konzentrierter Haltung Gesten ausführt. Soweit Gesten simpel und intuitiv gewählt wurden, ist es mit wenig Training möglich die Anwendung über Gesten zu steuern. Jedoch muss sich der Nutzer sich soweit auf die Anwendung konzentrieren, in dem er sich bewusst sein muss, dass jede Bewegung, die er während aktiver Gestensteuerung ausführt von der Anwendung wahrgenommen und analysiert und somit die Möglichkeit besteht, eine Aktion der Anwendung zu initiieren ohne dies beabsichtigt zu haben.

4.3 Herausforderungen der Mensch-Computer-Interaktion

Licklider beschreibt weiter Probleme und Herausforderungen der Man-Computer Symbiosis und der Mensch-Computer Kommunikation [LC62]. Auch wenn sich einige der Probleme heutzutage als neglierbar herausstellen, zeigt es doch einige Punkte auf, die bei der Konzeption einer Mensch-Computer-Interaktion zu beachten sind, denn selbst heute noch, werden Aufgaben, die schwer zu automatisieren sind, dem Menschen oft nur aus diesem Umstand übertragen, obgleich es sinnvoller wäre dies dem Computer zu überlassen und dem User nur die Aufgaben zuzuweisen, die für passend sind.

Da der Mensch in der Anwendung RoCoVoMo nicht einfach nur Bediener ist, sondern auch und vor allem zu erst den Umgang mit Gesten lernen muss stellen sich hierdurch auch eine Probleme, die behandelt werden müssen. Licklider betont hierbei insbesondere den kontinuierlichen Informationsaustausch zwischen Mensch und Computer. Das wichtigste Nebenziel ist, so Licklider weiter, die Zufriedenheit, den Spaß und die Bestätigung des Nutzers zu maximieren. In der modernen IT-Welt würde man dies mit dem Begriff *Gamification* verbinden, in dem der Nutzer positive Resonanz durch seine Aktionen erfährt.

In Hinsicht auf den Informationsaustausch ist aber gleichzeitig darauf zu achten, diesen für den Menschen verständlich und zugänglich zu gestalten. In der Anwendung RoCoVoMo beispielsweise, ist sollten die Körperelemente in einer Grafik angezeigt werden und nicht deren Koordinaten ausgegeben werden. Weiteres zu dem Konzept, das aus den hier geschilderten Problemen entsteht, ist in Kapitel 7 vermerkt.

Von den fünf Problemen, die Licklider schildert [LC62], sind nur noch zwei in Hinsicht auf moderne Anwendungen und die Gestensteuerung als Ausgangspunkt von Interesse:

1. Entwicklung eines Programms das es ermöglicht Echtzeitdaten zu verarbeiten und Informationsverarbeitungsprozesse zwischen Mensch und Computer unterstützt. Das System muss weiterhin *train-and-error* Operationen erlauben. Es muss weiter mit fehlerhaften Eingaben dynamisch umgehen können.

2. Lösung des Problems der Kooperation von Menschen untereinander bei der Entwicklung großer Anwendungen. Ohne effektives Teamwork ist Man-Computer Symbiosis nicht möglich, in dem Sinne, dass verschiedene Anwendungsbereiche auf verschieden konzipiert sein können und somit Unterschiede im Design und des Programms und Verwirrungen bei Nutzer auftauchen. Wenn beispielsweise der 'Schließen'-Button in einem Modul links, in einem anderen rechts liegt, weil verschiedene Entwickler diese implementiert haben, so ist keine intuitive Führung durch die Anwendung mehr möglich.

4.4 Moderne Design Prinzipien

Eine aktuelle Analyse der Mensch-Computer-Interaktion führt Vikas Chahar [Cha12] durch. Dabei wird Mensch-Computer-Interaktion als die Überschneidung von Informatik, Verhaltenswissenschaft, Design und weiterer Wissenschaftsfeldern beschrieben. Sie beinhaltet Software und Hardware und findet auf der Ebene der Benutzerschnittstelle statt. Weiterhin ist *MCI* auch von der Ergonomielehre abzugrenzen, die größeren Fokus auf eine physische Ebene richtet.

Betrachtet man nun eine solche Benutzerschnittstelle (*eng. user interface*), so führt Chahar folgende Prinzipien auf, die zu beachten sind:

- Ein früher Fokus auf Benutzer und Aufgaben: Es ist zu ermitteln, wie viele Nutzer, welche Aufgaben ausführen müssen und wie diese Aufgaben zu modellieren sind
- Empirische Messungen: Das Interface früh mit Nutzern testen, die die Anwendung auch tatsächlich benutzen werden. Dabei ist es hilfreich statistische Auswertungen durchzuführen
- Iteratives Design: Den vorherigen Punkten anschließend, sollten folgende Schritte iterativ durchgeführt werden:

1. Benutzerschnittstelle (re-)designen

2. Anwendung testen
3. Ergebnisse analysieren

4.5 Design Methodik

Diverse Methodiken verschiedener Techniken für Mensch-Computer-Interaktion existieren, meist daraus entstanden, wie Nutzer und technisches System interagieren. Aufgrund der vorliegenden Gesten- und Sprachsteuerung, stellt sich ein benutzerzentriertes Design (*eng. User-centered design (UCD)*) als sinnvoll heraus, da die Anwendung durch diese Befehle vom Nutzer aus bedient wird.

- User-centered design: UCD beruht auf der Idee, dass der Nutzer im Zentrum des Design einer Computer Anwendung steht. Dabei werden Wünsche, Notwendigkeiten und Grenzen des Benutzers eruiert und die Anwendung diesen Elementen entsprechend erstellt.
- Prinzipien für Benutzerschnittstellen (*eng. User Interface*) Design: Nach Charhar [Cha12] existieren sieben Prinzipien, die bei dem Design eines user interface zu jeder Zeit berücksichtigt werden sollten, nämlich Toleranz, Simplizität, Sichtbarkeit, Konsistenz, Struktur, Feedback, und Aufwand.

Oft werden diese Prinzipien und Begriffe verwendet, in dem man eine Anwendung einfach als "intuitiv" bezeichnet. Jedoch ist die Bezeichnung intuitiv kein technischer Indikator für eine gute Man-Computer Symbiosis, da dieser stets unterschiedlich ausgelegt und bestimmt wird. Jef Raskin [Ras94] beschreibt den Begriff intuitiv in Bezug auf Mensch-Computer-Interaktion als ein Synonym für etwas bekanntes. In einem Beispiel, an einer Anwendung, die mit einer Computer Maus gesteuert wird, zeigt er, dass dieses Peripheriegerät erst dann *intuitiv* in einem Programm nutzbar ist, sobald die Handhabung der Maus bekannt ist. In Bezug auf die Gesten- und Sprachsteuerung bedeutet dies, dass dem Nutzer erst das sprachliche Vokabular und die verwendbaren Gesten bekannt sein

müssen, bevor an eine dem Nutzer leicht fallende Verwendung der Anwendung zu denken ist.

Ebenso wichtig ist das Display Design. Zum Einen ist es wichtig, dem Nutzer alle nötigen Informationen und Rückmeldungen über und von der Anwendung anzuzeigen, zugleich muss vermieden werden, den Benutzer mit zu vielen Anzeigen und einem überladenen Bildschirmbereich zu verwirren.

Die Umsetzung und Auswirkung in der Anwendung RoCoVoMo all der zuvor aufgezeigten Prinzipien und Methoden werden im Kapitel 7 im Detail und an weiteren Beispielen beschrieben.

5 Auswahl des Roboters

6 Lego-Framework

7 Konzept und Steuerung der Anwendung

Ein Großteil der ersten Arbeit [EW13] betrafen die Konzeption und Entwicklung von Gesten- und Sprachbefehlen, die innerhalb einer Anwendung zur Steuerung eines mobilen Roboters benötigt werden und sinnvoll sind. Diese wurden von den Arbeiten an der Anwendung nochmals einer Revision unterzogen und nur in Bezug des zugrundeliegenden HMM in technischen Parametern für die Implementierung angepasst.

Das gesamte Vokabular an Gesten- Sprachbefehlen ist fest in der Anwendung RoCoVoMo verankert, und lässt nur durch Anpassungen im Code ändern. Dies ist beabsichtigt, da nur durch diese Begrenzung ein stabiler Gebrauch der Anwendung garantiert werden kann.

7.1 Verwendete Gesten

Gesten, die innerhalb der Anwendung genutzt werden sollen, müssen innerhalb eines HMM repräsentiert werden. Aus der Beschreibung des HMM [EW13] müssen daher einige Komponenten in die Anwendung RoCoVoMo integriert werden. Das bedeutet, dass für jede Geste Trainingsdaten vorhanden sein, beziehungsweise in das Programm eingegeben werden müssen, um diese später im laufenden Betrieb zu erkennen.

Die verwendbaren Gesten werden im folgenden noch einmal aufgelistet. Die entsprechenden Klassifikationen und wissenschaftlichen Beschreibungen wurden bereits in der

Ausarbeitung von Ebner und Werling [EW13] erörert und werden hier als bekannt vorausgesetzt.

7.1.1 Kreisbewegung

Diese Geste ermöglicht es den *Lego NXT* im Kreis fahren zu lassen. Sobald die Geste über das HMM und die vorhandenen Trainingsdaten abgeglichen wurde, wird die Anwendung die Aktion ausführen. Die Abbildung 7.1 zeigt die idealisierte Darstellung der Geste.

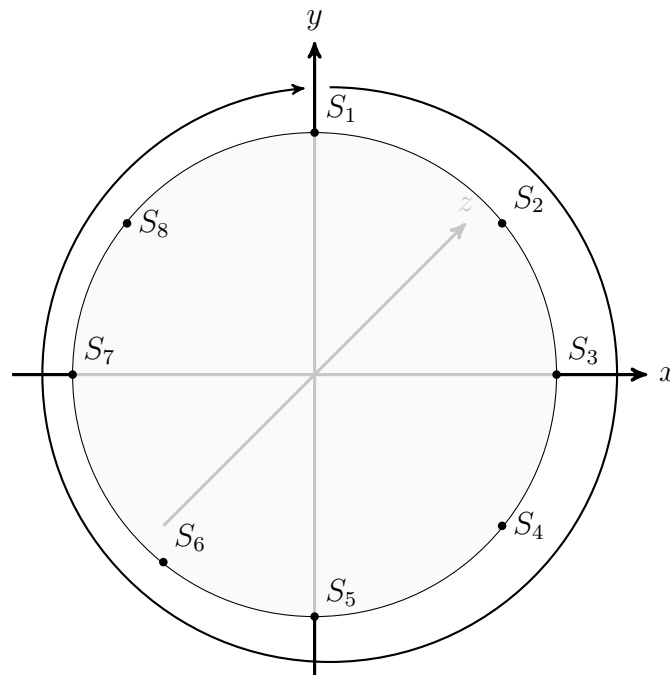


Abbildung 7.1: Abstrakte Darstellung einer Kreisbewegung im Koordinatensystem inklusiver ihrer 8 Zustandspunkte

Aufgrund der restriktionen aus dem mobilen Roboter, in dieser Arbeit der Lego NXT, ist aus dessen Motorspezifikationen und die hier gewählte Bauform, ist es nicht möglich einen beliebig großen oder kleinen Kreis zu fahren. Weiterhin unterstützt dies die in Abbildung 7.1 gezeigte Geste und das darunterliegende genutzte HMM nicht.

7.1.2 Vorwärtsbewegung

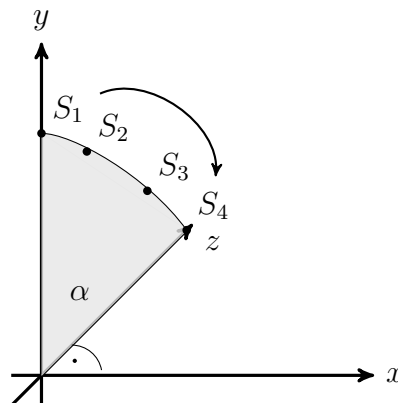


Abbildung 7.2: Abstrakte Darstellung einer Vorwärtsbewegung im Koordinatensystem inklusiver ihrer 4 Zustandspunkte

7.1.3 Haltesignal

7.1.4 Erweiterte Vorwärtsbewegung

Die oben beschriebene Vorwärtsbewegung beinhaltet keine Information über einen Winkel, in dem sich der Roboter fortbewegen soll. Um diese Information zu erhalten, muss die Geste und das darunter liegende HMM angepasst und erweitert werden.

7.1.5 Entriegeln — Blockieren

7.2 Trainingsmodul

Um die zuvor beschriebenen Gesten zu erkennen, müssen Trainingsdaten in das System eingelesen werden. Hierzu wurde ein Modul entwickelt und in die Anwendung integriert,

dass das schreiben und einfügen von Trainingsdaten in RoCoVoMo ermöglicht. Dabei wird der Kinect in einer Testumgebung gestartet und über das jnect Framework die Daten der gewünschten Geste, beziehungsweise des jeweiligen Body Elements eingelesen. Benötigte Informationen sind dabei nur die Koordinaten des jeweiligen Elements und die Anzahl der verwendeten HMM Zustandspunkte. Das jahmm Framework benötigt für die Errechnung eines HMM besonders gefilterte Daten, daher werden die Kinectinformationen in einem besonderen Format gespeichert, anschließend innerhalb der Anwendung ohne weitere Änderungen verwendet werden können. Die technischen Details dieses Moduls werden näher in Kapitel 8 beschrieben.

In Kapitel 4 wurde beschrieben, welche Merkmale entscheiden sind für eine gute Mensch-Computer-Interaktion und welche Gesichtspunkte berücksichtigt werden müssen. Die Kernfunktion, im Fall des Trainingsmoduls, ist das Einlesen von Kinectdaten über ein Interface. Dabei ist das Design und die einfache Bedienung der grafischen Oberfläche sehr wichtig. In dem Modul selbst können zur Bedienung jedoch noch keine Gesten verwendet werden, dies ist erst im Modul der Robotersteuerung möglich, da zuvor noch keine Trainingsdaten vorhanden sind. Da es sich weiterhin um ein User-centered design handelt ist es wichtig im Sinne der Man-Computer Symbiosis dem Benutzer eine Echtzeitverarbeitung und einfachen Informationsaustausch zu ermöglichen. Dies wird im Modul dadurch verwirklicht, in dem der Nutzer mit der Aufzeichnung seiner Bewegungen direkt und in Echtzeit eine Anzeige seiner Daten erhält und diese darüber hinaus manuell nachbearbeiten kann. Dabei wurde das Interface anhand der in Kapitel 4 vorgestellten Design Prinzipien von Chahar [Cha12] simpel und klar strukturiert und durch die manuelle Nachbearbeitung auf ein hohes Maß an Sichtbarkeit geachtet.

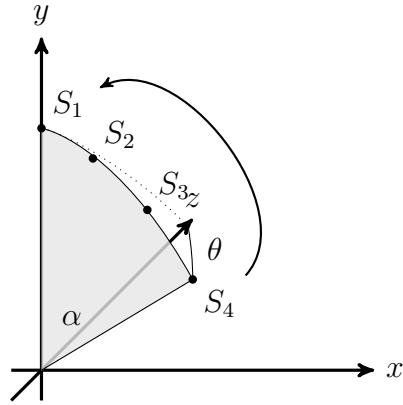
7.3 Kinect—Modul

Als ein Kernelement einer Man-Computer Symbiosis, gilt der Informationsaustausch. Jedoch muss dabei darauf geachtet werden, diesen in einer Form zu gestalten, dass der menschliche Teil der Beziehung dabei nicht von den Ausgaben überfordert wird, und

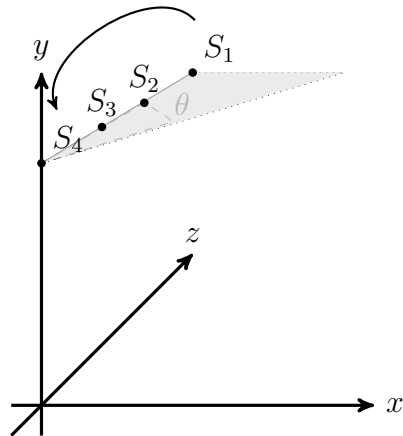
er diese in für ihn passender Form erhält. Im Falle der Gesten- und Sprachsteuerung mittels Kinect, werden zunächst lediglich drei dimensionale Koordinaten der einzelnen Körperelemente übermittelt. Diese sind für den Menschen nur äußerst umständlich zu lesen und stellen keine geeignete Form der Darstellung dar. Daher wurde ein Modul erstellt, dass mittels diesen Koordinaten ein grafisches Bild der Körpers des Menschen mit allen erfassten Elementen anzeigt, deren Koordinaten von der Kinect übermittelt werden. Hiermit wird ebenfalls nahe am User-centered design gearbeitet und die Prinzipien, wie Simplizität und Feedback verfolgt, in dem der Nutzer ein einfaches Bild seiner selbst sieht.

Ein weiteres Merkmal, dass mittels des zugrundeliegenden OSGi-Frameworks möglich ist - bereits in Teil 1 der Studienarbeit [EW13] vorgestellt - ist, dass dieses Modul in jedem Teil der Anwendung aufrufbar ist und der Benutzer zu jeder Zeit Feedback und grafische Informationen über seine Erfassung durch die Kinect erhalten kann. Dies kann auch als Hilfe bei der Modellierung weiterer Gesten, der Weiterentwicklung der Anwendung und dem Auffinden von Fehlern und Fehlverhalten bei der Erkennung von Gesten und der Umsetzung in die entsprechende Aktion dienen.

Details zur Umsetzung dieses Moduls werden in Kapitel 8 beschrieben.

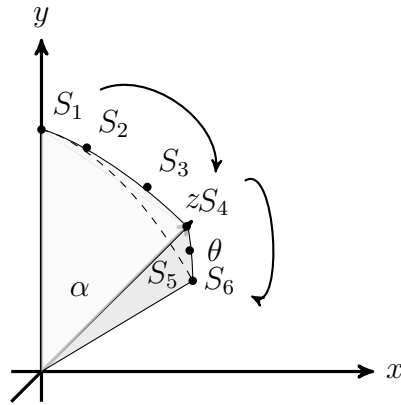


(a) Abstrakte Darstellung des Haltesignals aus einer möglichen Vorwärtswegung heraus, mit ihren 4 Zuständen

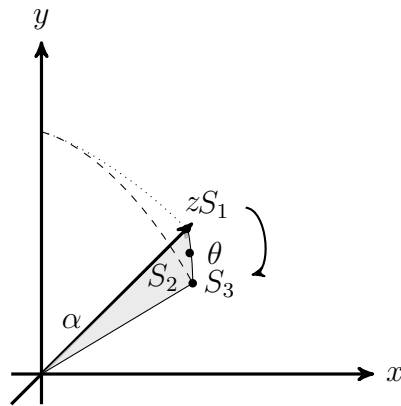


(b) Abstrakte Darstellung des Haltesignals aus einer möglichen Kreisbewegung heraus, mit ihren 4 Zuständen

Abbildung 7.3: Abstrakte Darstellung der Modelle für die Geste des Haltesignals mit jeweils 4 Zuständen



(a) Darstellung der gesamten erweiterten Vorwärtsbewegung mit 6 Zuständen



(b) Detaildarstellung der Seitwärtsbewegung der erweiterten Vorwärtsbewegung und ihrer 2 Zustände

Abbildung 7.4: Idealisierte Darstellung der 1. Variante einer erweiterten Vorwärtsbewegung mit 6 Zuständen

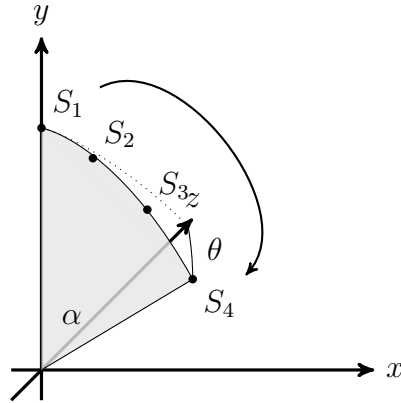


Abbildung 7.5: Idealisierte Darstellung der 2. Variante einer erweiterten Vorwärtsbewegung mit 4 Zuständen

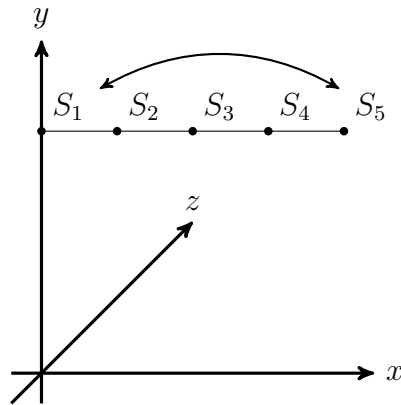


Abbildung 7.6: Idealisierte Darstellung der Geste zum Blockieren und Freigeben der Gesteneingabe mit 4 Zuständen

8 Implementierung

9 Ausblick - weitere Arbeiten

Abbildungsverzeichnis

7.1	Abstrakte Darstellung einer Kreisbewegung im Koordinatensystem inklusiver ihrer 8 Zustandspunkte	22
7.2	Abstrakte Darstellung einer Vorwärtsbewegung im Koordinatensystem inklusiver ihrer 4 Zustandspunkte	23
7.3	Abstrakte Darstellung der Modelle für die Geste des Haltesignals mit jeweils 4 Zuständen	26
7.4	Idealisierte Darstellung der 1. Variante einer erweiterten Vorwärtsbewegung mit 6 Zuständen	27
7.5	Idealisierte Darstellung der 2. Variante einer erweiterten Vorwärtsbewegung mit 4 Zuständen	28
7.6	Idealisierte Darstellung der Geste zum Blockieren und Freigeben der Gesteneingabe mit 4 Zuständen	28

Tabellenverzeichnis

Quellcodeverzeichnis

Literaturverzeichnis

- [Cha12] Vikas Chahar. An analytical study on hci. *International Journal of Computer Science and Management Studies*, 12(01):251–254, 2012.
- [EW13] Simon Ebner and Volker Werling. Gesten- und sprachsteuerung fuer einen mobilen roboter mittels kinect for windows unter java. Vol. 1:1–101, 2013.
- [LC62] J. C. R. Licklider and Welden E. Clark. On-line man-computer communication. In *Proceedings of the May 1-3, 1962, spring joint computer conference*, AIEE-IRE '62 (Spring), pages 113–128, New York, NY, USA, 1962. ACM.
- [Lic68] J. C. R. Licklider. Man-computer symbiosis. In W. D. Orr, editor, *Conversational Computers*, pages 3–5. Wiley, New York, 1968.
- [Ras94] Jef Raskin. Viewpoint: Intuitive equals familiar. *Commun. ACM*, 37(9):17–18, September 1994.