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

Januar 2013

Bearbeitungszeitraum	12 Wochen
Matrikelnummer, Kurs	5837963 7012192, TAI10B2
Gutachter	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, Januar 2013

Simon Ebner, Volker Werling

Zusammenfassung

Inhaltsverzeichnis

1 Einleitung

Gestik im Sinne von kommunikativen Bewegungen stellt einen wesentlichen Teil der nonverbalen Kommunikation dar. Die Definition nach Kurtenbach und Hulteen (1990) besagt:

> A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key on a keyboard is neither observed nor significant. All that matters is which key was pressed.

Als solches bildet eine Geste abstrakte Strukturen und bildhafte Vorstellungen unmittelbar ab. Sie ist somit eine Körperbewegung, die Informationen enthält. Informationen, die herkömmliche Eingabegeräte, darunter Tastatur und Maus, nicht in dieser Form wiedergeben können. Ein Umstand, der sich besonders im Bereich der Mensch-Computer-Interaktion auswirkt.
Das Ziel der Mensch-Computer-Interaktion ist es, die Kommunikation intuitiv und unmittelbar zu gestalten. Genau an dieser Stelle setzen Gesten an, in dem sie eine bessere Schnittstelle zwischen Mensch und Maschine darstellen.
Die Spannweite von Realisierungen von Mensch-Maschine-Schnittstelle ist groß. Frühe Ansätze zeigen bereits das Potential, Gesten zur Steuerung und Nutzung von diversen Geräten zu verwenden. Das *Theremin* - ein 1919 erfundenes elektornisches Musikinstrument, das mit Händen berührungslos durch Beeinflussung eines elektromagnetischen Feldes gesteuert wird und dadurch Töne erzeugt - ist eine der ersten technischen Lösungen für eine Gestensteuerung. Eine weitere From, der *Datenhandschuh*, 1977 von Electronic

Visualization Laboratory Labs entwickelt [1], erregte großes Interesse und wurde unter anderem als *Power Glove* von Mattel vertrieben. Datenhandschuhe werden bis heute in den verschiedensten Bereichen eingesetzt.
Eine weitere Variante der Gestenerkennung ist die Bewegungsdetektion. Dabei wird über optische Sensoren der Körper oder einzelne Körperteile des Nutzers erkannt und somit die Steuerung durch Gesten ermöglicht. Eine frühe Lösung ist der sogenannte *Videoplace* [2], entwickelt von Myron Krueger Mitte der 70'er Jahre [3]. Ausgelegt als ein *Labor für künstliche Realität* war eine Person, durch ihn umgebende Projektoren und Videokameras, in der Lage seine Umgebung zu beeinflussen. Moderne Lösungen haben das Konzept der Detektion durch Kameratechnik weiterentwickelt und einer weiten Öffentlichkeit zugänglich gemacht. Darunter auch das *PLAYSTATIONEye* [4]. Der populärste Vertreter ist *Kinect for XBOX* mit über 18 Millionen verkauften Exemplaren [5]. Im folgenden wird mit jener Kinect gearbeitet.
In dieser Arbeit wird mit Hilfe der Kinect eine Gesten- und Spracherkennung entwickelt um eine Steuerung für einen mobilen Roboter zu implementieren. Aufgrund des Umfangs dieser Studienarbeit wird die Erarbeitung dieser Anwendung in zwei Teile aufgeteilt.
Der erste Teil der Arbeit beschäftigt sich vorranging mit der Realisierung der Gesten- und Spracherkennung. Die eigentliche Steuerung für einen mobilen Roboter wird vorerst durch eine Schnittstelle und einer Testumgebung ersetzt. Der Fokus liegt hierbei in erster Linie auf der verwendeten Technik, in diesem Fall der Kinect zur Bewegungsdetektion, den verwendeten Gesten und Sprachbefehlen, sowie der Umsetzung der Software und den darin verwendeten Technologien. Der zweite Teil wid an diese Arbeit anknüpfen und im weiteren die Umsetzung der Steuerung für und den Einsatz eines mobilen Roboters beschreiben.

[1] Sturman, D.J., Zeltzer, D. (January 1994). „A survey of glove-based input". IEEE Computer Graphics and Applications 14 (1): 30–39
[2] Beschreibung der Einrichtung „VIDEOPLACE". medienkunstnetz.de. Abgerufen Dezember 22, 2012
[3] Myron Krueger. Artificial Reality 2, Addison-Wesley Professional, 1991. ISBN 0-201-52260-8
[4] „PLAYSTATIONEye Brings Next-Generation Communication to PLAYSTATION3". us.playstation.com. Sony Computer Entertainment America. Abgerufen Dezember 21, 2012
[5] Takahashi, Dean (Januar 9, 2012). „Xbox 360 surpasses 66M sold and Kinect passes 18M units". venturebeat. Abgerufen Dezember 20, 2012

Die folgende Ausarbeitung spiegelt die Umsetzung des ersten Teils dieser Arbeit wieder.

2 Aufgabenstellung

3 Stand der Technik

3.1 Kinect

Kinect ist eine Gerät zur Dedektion von Bewegungen. Entwickelt wurde es von PrimeSense als Hardware zur Steuerung der Videokonsole XBOX 360 von Microsoft Corp. Nach der Ankündigung im März 2010 [1] war die Erweiterung mit dem Erscheinungsdatum 10. Novmeber 2010 in der Ausführung *Kinect for XBOX 360* in Europa erhältlich [2]. Nach einem sehr erfolgreichem Verkaufsstart [3] und anhaltender Nachfrage[4]. kündigte Microsoft am 9. Januar 2012 ein weiteres Modell der Kinect, die sogenannte *Kinect for Windows* für den 1. Februar 2012 an [5].
Darüber hinaus veröffentlichte Microsoft bereits am 16. Juni 2011 eine erste Version ihrer *Kinect for Windows SDK*[6]. Mit diesem Software Development Kit ist es möglich auf einer Windows 7 Plattform Anwendungen zu entwickeln, die eine Kinect als Eingabegerät verwenden.
Mit der Verfügbarkeit einer Kinect-Variante, die für den Einsatz am PC ausgelegt ist und der frei zugänglichen einer breiten Öffentlichkeit als Forschungsgegenschaft zugänglich,

[1] Pressemitteilung, der Veröffentlichung. Siehe Link. Microsoft.com. Abgerufen Dezember 20, 2012
[2] Erscheinungsdatum der *Kinect for XBOX 360*. Siehe Link. BBC UK. Abgerufen Dezember 20, 2012
[3] „Am schnellsten verkauftes Perepheriegerät für Spiele“. Guinnessworldrecords.com. Abgerufen Dezemeber 22, 2012
[4] Takahashi, Dean (Januar 9, 2012). „Xbox 360 surpasses 66M sold and Kinect passes 18M units“. venturebeat. Abgerufen Dezember 20, 2012
[5] „Ankündigung der *Kinect for Windows*“. blogs.msdn.com. Abgerufen Dezember 22, 2012
[6] „Microsoft Releases Kinect for Windows SDK Beta for Academics and Enthusiasts“. Microsoft.com. Abgerufen Dezember 21, 2012

was auch der ausschlaggebende Punkt für den Einsatz der Kinect in dieser Studienarbeit ist.

3.1.1 Hardware

Die beiden Kinect-Modelle unterscheiden sich in einigen Details[7]. Der bedeutendste Faktor ist das sogenannte *Near Mode* Feature:

> Near Mode enables the depth sensor to see objects as close as 40 centimeters and also communicates more information about depth values outside the range than was previously available. There is also improved synchronization between color and depth, mapping depth to color, and a full frame API. [7]

Neben dem aktualisierten Tiefensensor unterscheiden sich die beiden Varianten auch in einer höheren Auflösung der RGB-Kamera, die für eine Gestenerkennung relevant sein kann. Aus diesem Grund wird für diese Arbeit die *Kinect for Windows* genutzt.

Technische Daten

Eine Kinect entält innerhalb des Gehäuses, folgende Sensoren [8]:

- Eine RGB-Kamera mit einer Auflösung von 1280x960. Dies ermöglicht eine Farbbilderfassung
- Ein Infrarot (IR) Emitter und ein IR Tiefensensor. Der Emitter emittiert infrarote Lichtstrahlen und der Tiefensensor erfasst die an den Sensor reflektierten Strahlen. Die reflektierten Strahlen werden in Tiefeninformation umgewandelt, in dem der

[7] „Informationsseite über Unterschiede der Kinect Versionen“. Microsoft.com. Abgerufen Dezember 22, 2012

[8] „Kinect for Windows Sensor Components and Specifications“. msdn.microsoft.com. Abgerufen Dezember 21, 2012

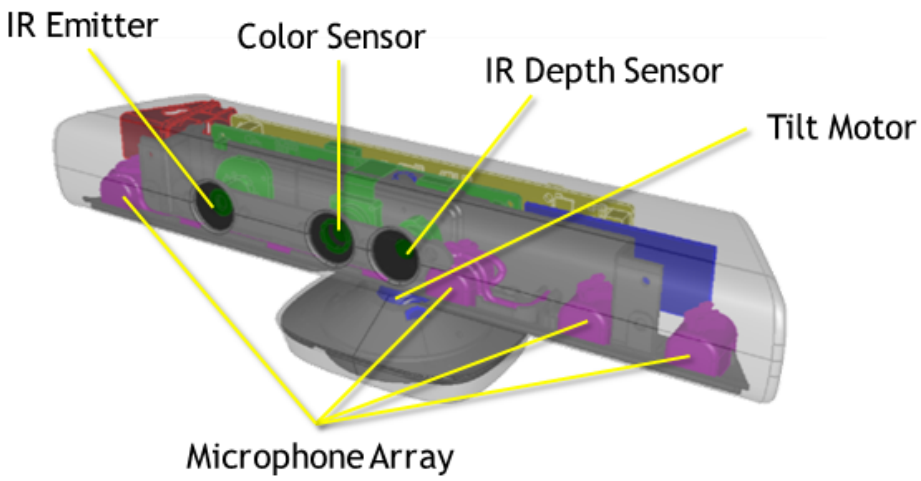

Abbildung 3.1: Schematische Ansicht der Sensoren einer *Kinect for Windows* (Quelle: Microsoft Corp.[1])

Abstand zwischen Objekt und Sensor bestimmt wird. Dies ermöglicht die Erfassung von Tiefenbildern

- Ein Mikrofonarray, das vier Mikrofone zur Soundaufnahme enthält. Die Anzahl der Mikrofone ermöglicht nicht nur die Aufzeichnung von Audiodaten, sondern auch die Lokalisierung der Soundquelle und die Richtung des Audiosignals
- Ein 3-Achsen-Beschleunigungssensor, konfiguriert für einen Bereich der zweifachen Erdbeschleunigung, um die gegenwärtige Ausrichtung der Kinect zu bestimmen
- Ein Kippmotor, zur automatisierten Justierung der Sensoren

Weitere Details der technischen Spezifikation einer *Kinect for Windows* sind in Tabelle 3.1 aufgelistet:

Tabelle 3.1: Kinect for Windows – technische Spezifikation [8]

Kinect	**Spezifikation**
Blickwinkel	43° vertikales, 57° horizontales Blickfeld
Vertikaler Neigebereich	±27°
Bildwiederholrate (Farb und Tiefensignal)	30 Bilder pro Sekunde (FPS)
Audioformat	16-kHz, 24-bit mono pulse code modulation (PCM)
Audioeingang	Ein Vier-Mikrofonarray mit 24-Bit Analog-Digital-Wandler (ADC)
Datensignal–Tiefensensor	640x480 16-bit, 30 Bilder pro Sekunde
Datensignal–RGB-Kamera	1280x960 16-Bit, 12 Bilder pro Sekunde 640x480 16-Bit, 30 Bilder pro Sekunde
Tiefensensorreichweite	0,4 – 4 m
Skelett Tracking System	Erkennung von bis zu sechs Benutzern, zwei davon trackbar/verfolgbar Verfolgung von 20 Gelenken pro aktivem Nutzer

[8] „Kinect for Windows Sensor Components and Specifications“. msdn.microsoft.com. Abgerufen Dezember 21, 2012

3.1.2 Software

Durch die Veröffentlichung eines Software Development Kits ist es möglich, die Kinect in eigene Pogramme einzubinden und neue Anwendungsfälle zu bearbeiten. Dazu hat Microsoft ebenfalls ein Handbuch veröffentlicht [Cor12], das Informationen und Ratschläge zum Entwurf von Anwendungen liefert.

Kinect for Windows SDK

Das *Kinect for Windows SDK* steht aktuell in der Version 1.6 [9] bereit. Dabei kann direkt in den Programmiersprachen C++, C# , und Visual Basic auf einer Windows 7 Plattform entwickelt werden.
Das SDK umfasst dabei folgende Funktionen [10]:

- Treiber und technische Dokumentation zur Implementierung von Anwendungen, die Kinect for Windows sensoren nutzen
- Referenz APIs und Dokumentation zur Programmierung von *managed* und *unmanaged* Code. Die APIs stellen mehrere Mediensignale bereit
- Beispiele und Best-Practice-Lösungen

Weitere Frameworks

Da Microsoft die Nutzungsmöglichkeiten seines SDK hinsichtlich verwendeter Pogrammiersprache und Plattform einschränkt, begannen Forscher eigene Frameworks und Treiber zur Nutzung der Kinect zu entwickeln[11]. PrimeSense selbst veröffentlichte Treiber

[9] Downloadseite des SDK. Microsoft.com. Abgerufen Dezember 21, 2012
[10] „Kinect for Windows Programming Guide“. msdn.microsoft.com. Abgerufen Dezember 23, 2012
[11] „Open Source Kinect contest has been won“. hackaday.com. November 11, 2010. Abgerufen Dezember 21, 2012

und Middleware für die Kinect [12], als das offizielle SDK noch nicht zur Verfügung stand. Das Ziel dieser Frameworks war es zu meist, Kinect-Steuerung unter Unix-Plattformen und Programmiersprachen, wie Java, nutzbar zu machen. Ein Teil dieser Ausarbeitung ist es, ein für die Aufgabenstellung und Zielsetzung der Studienarbeit passendes Framework zu bestimmen.
Verbreitete Lösungen, die im Kapitel 4 näher betrachtet werden, sind:

- OpenNI
- OpenKinect
- jnect

3.2 Java – Eclipse

Die Anwendung, die im Rahmen dieser Arbeit entsteht, wird auf Basis der Programmiersprache Java und der Entwicklungsumgebung Eclipse erstellt. Die Nutzung der Kinect und des Kinect SDK unterstützt nativ nicht die Entwicklung auf Basis von Java, wie in Abschnitt 3.1.2 festgestellt wurde. Da jedoch die Expertise der Autoren dieser Arbeit im Java-Umfeld liegt, wird die Anwendung dennoch in der Programmiersprache Java umgesetzt. TODO ... Eclipse

3.2.1 Grafische Oberfläche

Das Graphical User Interface (GUI) ist ein wesentlicher Bestandteil der Anwendung, da hierüber der gesamte Informationsaustausch mit dem Nutzer der Kinect stattfindet. Dabei wird auf verschiedene Techniken gesetzt, die im folgenden kurz beschrieben werden. Bei der grafischen Darstellung muss man zwischen Informationen der Anwendung und

[12] Mitchell, Richard (Dezember 10, 2010). „PrimeSense releases open source drivers, middleware for Kinect“. Joystiq. Abgerufen Dezember 22, 2012

Interaktion des Nutzers mit der Software unterscheiden. Dazu werden nämlich unterschiedliche Techniken verwendet.

Rich client platform

Die Rich client platform (RCP) ist Werkzeug zur Entwicklung von unabhängigen Software-Komponenten, die nicht nur auf GUI-Komponenten beschränkt ist [13]. Dabei liegt der Fokus auf Client-Applikation, worunter auch die Andwendung dieser Arbeit fällt. aelkrjqw';lkjads;

Die Lightweight Java Game Library

Die grafische Darstellung der Interaktion zwischen Akteur und Kinect ist ein wichtiger Bestandteil dieser Anwendung. Zur Anzeige aufwändiger grafischer Objekte wird auf die Lightweight Java Game Library (LWJGL)[14] gesetzt. Damit können einfach zweidimensionale Grafiken, zum Beispiel Kreise, oder komplexe dreidimensionale Ansichten in einer Java-Anwendung angezeigt werden. Diese Java-Bibliothek stellt eine API zum Zugriff und zur Verwendung der bekannten Open Graphics Library[15] bereit.

Eclipse Modeling Framework

Das Eclipse Modeling Framework (EMF)[16] ist ein Framework zur Modellierung von Java-Anwendungen. Es ermöglicht aus strukturierten Datenmodellen, Java-Code zu generieren. Für die Kinect-Anwendung sind lediglich die Bereiche des Frameworks relevant, die diese *Modelle* beschreiben.

[13] „Rich Client Platform“. wiki.eclipse.org. Abgerufen Januar 1, 2013
[14] „LWJGL Lightweight Java Game Library“. lwjgl.org. Abgerufen Januar 1, 2013
[15] „Startseite der Bibliothek“. opengl.org. Abgerufen Januar 1, 2013
[16] „Eclipse Modeling Framework Project (EMF)“. eclipse.org. Abgerufen Januar 1, 2013

Graphical Editing Framework

Zur grafischen Darstellung und grafischen Manipulation der zuvor vorgestellten EMF-Modelle existiert das sogenannte Graphical Editing Framework (GEF)[17]. Es stellt Methoden zur Verfügung, um Grafikeditoren und weitere Ansichten fr Eclipse Anwendungen zu erstellen. Für die Studienarbeit relevante Bereiche, sind die Funktionen, die eine vereinfachte Darstellung der eben genannten EMF-Modelle ermöglichen.

3.2.2 OSGI

3.3 Roboter - Ausblick

3.4 Mustererkennung

[17] „GEF (Graphical Editing Framework)“. eclipse.org. Abgerufen Januar 1, 2013

4 Kinect–Framework

4.1 Microsoft Kinect SDK

asdf

4.2 Java Frameworks

adsf

4.2.1 OpenNI

Beschreibung

adsf

Technische Daten

SWOT–Analyse

4.2.2 OpenKinect

Beschreibung

Technische Daten

SWOT–Analyse

4.2.3 jnect

Beschreibung

Technische Daten

SWOT–Analyse

4.2.4 Scoring–Modell

4.2.5 Analyseergebnis

adsasfd

5 Konzeption

5.1 Technische Vorgaben

5.1.1 Vorgaben durch Kinect

5.1.2 Vorgaben durch jnect–Framework

5.1.3 Service–Architektur

5.1.4 Abhängigkeit zu Robotertechnik

5.2 Fachliche Vorgaben

5.2.1 Vorgaben durch Mensch-Computer-Interaktion

5.2.2 Klassifikation der Anwendung

5.2.3 Analyse der Gesten- und Sprachsteuerung

6 Modelle zur Gesten- und Spracherkennung

6.1 Hidden-Markov-Modell

6.1.1 ToDo

6.2 Baum-Weich-Algorithmus

6.2.1 ToDo

6.3 ToDo

7 Gesten

7.1 Kreisbewegung

7.1.1 ToDo

7.2 Vorwärtsbewegung

7.2.1 ToDo

7.3 Haltesignal

7.3.1 ToDo

8 Sprachbefehle

8.1 ToDo

9 Implementierung

9.1 OSGI–Bundles

9.1.1 Gesten

9.1.2 Sprachbefehle

9.1.3 Roboterschnittstelle

9.1.4 Ausführbare Aktionen

9.2 Oberfläche

9.2.1 RCP–Anwendung

9.2.2 Grafische Darstellung

9.3 Ablaufbeschreibung

10 Ausblick - weitere Arbeiten

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Literaturverzeichnis

[Cor12] Microsoft Corp., editor. *HUMAN INTERFACE GUIDELINES*, volume v1.5.0. Microsoft Corp., 2012.