



# **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**  
**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, Januar 2013

Simon Ebner, Volker Werling

## Zusammenfassung

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
<b>2</b>	<b>Aufgabenstellung</b>	<b>9</b>
<b>3</b>	<b>Stand der Technik</b>	<b>10</b>
3.1	Hardware . . . . .	10
3.1.1	Kinect . . . . .	10
3.1.2	Roboter - Ausblick . . . . .	12
3.2	Software . . . . .	12
3.2.1	Microsoft Kinect SDK . . . . .	12
3.2.2	Java . . . . .	12
3.2.3	GUI . . . . .	12
3.3	Mustererkennung . . . . .	12
<b>4</b>	<b>Kinect-Framework</b>	<b>13</b>
4.1	Microsoft Kinect SDK . . . . .	13
4.2	Java Frameworks . . . . .	13
4.2.1	OpenNI . . . . .	13
4.2.2	OpenKinect . . . . .	14
4.2.3	jnect . . . . .	14
4.2.4	Scoring-Modell . . . . .	14
4.2.5	Analyseergebnis . . . . .	14
<b>5</b>	<b>Konzeption</b>	<b>15</b>
5.1	Technische Vorgaben . . . . .	15
5.1.1	Vorgaben durch Kinect . . . . .	15
5.1.2	Vorgaben durch jnect-Framework . . . . .	15

5.1.3	Service-Architektur . . . . .	15
5.1.4	Abhängigkeit zu Robotertechnik . . . . .	15
5.2	Fachliche Vorgaben . . . . .	15
5.2.1	Vorgaben durch Mensch-Computer-Interaktion . . . . .	15
5.2.2	Klassifikation der Anwendung . . . . .	15
5.2.3	Analyse der Gesten- und Sprachsteuerung . . . . .	15
<b>6</b>	<b>Modelle zur Gesten- und Spracherkennung</b>	<b>16</b>
6.1	Hidden-Markov-Modell . . . . .	16
6.1.1	ToDo . . . . .	16
6.2	Baum-Weich-Algorithmus . . . . .	16
6.2.1	ToDo . . . . .	16
6.3	ToDo . . . . .	16
<b>7</b>	<b>Gesten</b>	<b>17</b>
7.1	Kreisbewegung . . . . .	17
7.1.1	ToDo . . . . .	17
7.2	Vorwärtsbewegung . . . . .	17
7.2.1	ToDo . . . . .	17
7.3	Haltesignal . . . . .	17
7.3.1	ToDo . . . . .	17
<b>8</b>	<b>Sprachbefehle</b>	<b>18</b>
8.1	ToDo . . . . .	18
<b>9</b>	<b>Implementierung</b>	<b>19</b>
9.1	OSGI-Bundles . . . . .	19
9.1.1	Gesten . . . . .	19
9.1.2	Sprachbefehle . . . . .	19
9.1.3	Roboterschnittstelle . . . . .	19
9.1.4	Ausführbare Aktionen . . . . .	19
9.2	Oberfläche . . . . .	19
9.2.1	RCP-Anwendung . . . . .	19
9.2.2	Grafische Darstellung . . . . .	19
9.3	Ablaufbeschreibung . . . . .	19

10 Ausblick - weitere Arbeiten	20
Abbildungsverzeichnis	21
Tabellenverzeichnis	22
Quellcodeverzeichnis	23
Literaturverzeichnis	24
Glossar	25

# 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 Form, der *Datenhandschuh*, 1977 von Electronic

Visualization Laboratory Labs entwickelt <sup>1</sup>, erregte großes Interesse und wurde unter anderem als *Power Glove* von Mattel vertrieben. Datenhandschuhe werden bis heute in den verschiedensten Bereichen eingesetzt. Eine moderne 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. Der populärste Vertreter ist *Kinect* mit über 18 Millionen verkauften Exemplaren <sup>2</sup>. 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 vorrangig 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.

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

---

<sup>1</sup>Sturman, D.J., Zeltzer, D. (January 1994). „A survey of glove-based input“. IEEE Computer Graphics and Applications 14 (1): 30–39

<sup>2</sup>Takahashi, Dean (Januar 9, 2012). „Xbox 360 surpasses 66M sold and Kinect passes 18M units“. venturebeat. Abgerufen Dezember 20, 2012



## 2 Aufgabenstellung

## 3 Stand der Technik

### 3.1 Hardware

#### 3.1.1 Kinect

Kinect ist ein Gerät zur Detektion von Bewegungen. Entwickelt wurde es von PrimeSense als Hardware zur Steuerung der Videokonsole XBOX 360 von Microsoft Corp. . Nach der Ankündigung im März 2010 <sup>1</sup> war die Erweiterung mit dem Erscheinungsdatum 10. November 2010 in der Ausführung *Kinect for XBOX 360* in Europa erhältlich <sup>2</sup>. Nach einem sehr erfolgreichem Verkaufsstart <sup>3</sup> und anhaltender Nachfrage<sup>4</sup>. kündigte Microsoft am 9. Januar 2012 ein weiteres Modell der Kinect, die sogenannte *Kinect for Windows* für den 1. Februar 2012 an <sup>5</sup>.

Die beiden Modelle unterscheiden sich in einigen Details<sup>6</sup>. Der bedeutendste Faktor ist das sogenannte *Near Mode* Feature:

---

<sup>1</sup>Pressemitteilung, der Veröffentlichung. Siehe Link. Microsoft.com. Abgerufen Dezember 20, 2012

<sup>2</sup>Erscheinungsdatum der *Kinect for XBOX 360*. Siehe Link. BBC UK. Abgerufen Dezember 20, 2012

<sup>3</sup>„Am schnellsten verkauftes Peripheriegerät für Spiele“. Guinnessworldrecords.com. Abgerufen Dezember 22, 2012

<sup>4</sup>Takahashi, Dean (Januar 9, 2012). „Xbox 360 surpasses 66M sold and Kinect passes 18M units“. venturebeat. Abgerufen Dezember 20, 2012

<sup>5</sup>„Ankündigung der *Kinect for Windows*“. blogs.msdn.com. Abgerufen Dezember 22, 2012

<sup>6</sup>„Informationsseite über Unterschiede der Kinect Versionen“. Microsoft.com. Abgerufen Dezember 22, 2012

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. <sup>6</sup>

Neben dem aktualisierten Tiefensensor unterscheiden sich die beiden Varianten in keinen technischen Eigenschaften, die für eine Gestenerkennung relevant sind.

## Technische Daten

Innerhalb des Gehäuses, eine Kinect enthält folgende Sensoren:

- 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 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 automatischen Justierung der Sensoren

Tabelle 3.1: Additional InfoObjects for Report

<b>Kinect</b>	<b>Spezifikation</b>
Blickwinkel	43° vertikales, 57° horizontales Blickfeld
Vertikaler Neigebereich	$\pm 27^\circ$
Bildwiederholrate (Farb und Tiefensignal)	30 Bilder pro Sekunde (FPS)
Audioformat	16-kHz, 24-bit mono pulse code modulation (PCM)
Audioeingang	
Accelerometer	

### 3.1.2 Roboter - Ausblick

## 3.2 Software

### 3.2.1 Microsoft Kinect SDK

### 3.2.2 Java

### 3.2.3 GUI

## 3.3 Mustererkennung

## 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

3.1 Additional InfoObjects for Report . . . . . 12

# Quellcodeverzeichnis

# Literaturverzeichnis



# Glossar

## Datenhandschuh

Der Datenhandschuh ist ein Eingabegerät in Form eines Handschuhs. Durch Bewegungen der Hand und Finger erfolgt eine Orientierung im virtuellen Raum. 6

## Mensch-Computer-Interaktion

Die Mensch-Computer-Interaktion (englisch *Human-Computer Interaction, HCI*) als Teilgebiet der Informatik beschäftigt sich mit der benutzergerechten Gestaltung von interaktiven Systemen und ihren Mensch-Maschine-Schnittstellen. 5, 6, 11

## Mensch-Maschine-Schnittstelle

Die Benutzerschnittstelle (nach Gesellschaft für Informatik, Fachbereich Mensch-Computer-Interaktion auch Benutzungsschnittstelle) ist die Stelle oder Handlung, mit der ein Mensch mit einer Maschine in Kontakt tritt. 6