

Entwicklung einer Gestiksteuerung mittels Kinect für den humanoiden Roboter Nao

STUDIENARBEIT

für die Prüfung zum
Bachelor of Engineering
des Studienganges Informationstechnik
an der
Dualen Hochschule Baden-Württemberg Karlsruhe

von
Lukas Essig
und
Michael Stahlberger

Abgabedatum 12. Mai 2014

Bearbeitungszeitraum	24 Wochen
Matrikelnummer	8898018, 1367912
Kurs	TINF11B3
Ausbildungsfirma	Fiducia IT AG Karlsruhe
Betreuer	Herr Prof. Hans-Jörg Haubner

Erklärung

Gemäß §5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.

Ich habe die vorliegende Studienarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ort Datum

Unterschrift

Sperrvermerk

Die nachfolgende Arbeit enthält vertrauliche Daten und Informationen der Fiducia IT AG.

Veröffentlichungen oder Vervielfältigungen -auch nur auszugsweise- sind ohne ausdrückliche schriftliche Genehmigung des Unternehmens nicht gestattet.

Die Arbeit ist nur den Korrektoren sowie den Mitgliedern des Prüfungsausschusses zugänglich zu machen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Umfeld	1
1.2	Thema	1
1.3	Inhalt	2
2	Kinect	3
2.1	Kinect Hardware	3
2.2	Kinect Software	3
2.3	Kinect SDK	4
2.3.1	Richtiger Umgang mit dem SDK	4
3	Nao	5
3.1	Nao Hardware	5
3.2	Nao Software	7
3.3	Nao SDK	7
4	Realisierung	8
4.1	Architektur	8
4.2	Prototyp	8
	Literaturverzeichnis	9

Abbildungsverzeichnis

3.1	Übersicht Nao V3.2	6
-----	------------------------------	---

Tabellenverzeichnis

Abkürzungsverzeichnis

CFS	Completely Fair Scheduler
EDD	Earliest Due Date
FCFS	First come - First served
IS25TO	Transaktionen und Operation-Tools
ITS	IT Services
LIFO	Last In - First Out
SJF	Shortest Job First

1 Einleitung

In dieser Einleitung wird das zentrale Thema dieser Arbeit beleuchtet, beschrieben in welchem Umfeld die Arbeit entworfen wurde und wie sich dieses Dokument inhaltlich aufbaut.

1.1 Umfeld

Diese Studienarbeit mit dem Thema **To do** ⁽¹⁾ wurde während zwei Theoriephasen an der Dualen Hochschule Baden-Württemberg am Standort Karlsruhe von Michael Stahlberger und Lukas Essig durchgeführt. Betreut wurde die Arbeit durch Herrn Prof. Hans-Jörg Haubner, sowie Herrn Michael Schneider.

1.2 Thema

Zentrales Thema der Arbeit ist die Interaktion zwischen Mensch und Roboter. Dabei soll ein Modell des humanoiden Roboters Nao durch eine Gestik seines Benutzers gesteuert werden. Dabei ahmt der Roboter die Bewegungen nach, die der Benutzer zeitgleich bzw. etwas zuvor ausgeführt hat. Zudem sollen bestimmte Steuergesten implementiert werden, die den Roboter in verschiedene Modi versetzen, in denen er gleiche Gesten auf verschiedene Wege interpretiert. Die Gestik-Erkennung soll dabei mit Hilfe eines Xbox - Kinect Moduls erfolgen. Von dieser Stereokamera werden bestimmte Gesten erkannt und über eine Software verarbeitet und an den Nao-Roboter weitergeleitet, der diese dann nachahmt. Dabei ist natürlich darauf zu achten, dass keine Bewegungen von Nao ausgeführt werden, die ihm mechanisch oder elektronisch Schaden zufügen können.

Das Vorgehen ist folgendermaßen strukturiert: Nach der Einarbeitung in die Materie Nao - Roboter und Kinect - Kamera soll die Wahl einer geeigneten Programmiersprache getroffen werden. Daraufhin müssen zwei verschiedene Anwendungen konzeptioniert und implementiert werden. Erstens die Erkennung der Gesten mittels Kinect und zweitens das Empfangen der Gesten und die kinema-

tische Umsetzung durch den Roboter. Zwischen diesen beiden Anwendungen muss eine geeignete Kommunikationsschicht entworfen und implementiert werden.

1.3 Inhalt

To do (2)

2 Kinect

Im folgenden Kapitel wird der Aufbau und die Funktion des Kinect Sensors erläutert.

2.1 Kinect Hardware

Microsoft entwickelte 19nn ein Produkt namens XBox Kinect. Dieses Produkt war ursprünglich als Zusatzhardware für die XBox 360 gedacht.

[3]

To do (4) To do (5) To do (6)

2.2 Kinect Software

Das Produkt Xbox Kinect kam bereits vor einigen Jahren auf den Markt. Somit hatten die Entwickler Zeit, um ein SDK zu entwickeln, was alle wichtigen Programmfunktionen bereits enthält. Dies macht es einem Entwickler relativ leicht eine Anwendung zu erstellen, die bestimmte Kinect-Funktionen bereitstellt. Die Hardware liefert folgende Werte: -Audiosignal von n Microphonen -RGB Bildsignal der Kamera -Tiefensignal (wobei von Treiber errechnet) -¿ IR Tiefenscan... näher erläutern

Für dieses Projekt sind die Audiodaten jedoch irrelevant. Wichtig sind primär die Bilddaten inklusive Tiefenwerte. Das Kinect SDK bietet bereits standardmäßig Zugriff auf die grundlegenden Funktionen: Dabei können bestimmte Proxyobjekte registriert und abgerufen werden. Die wichtigste Implementierung eines dieser Proxyobjekte ist für dieses Projekt der Skeletonstream.

Dieser Stream berechnet anhand der Tiefendaten und der RGB-Daten werte für ein Menschliches Skelett:

-Armwinkel -Positionen -Kinectraum...

DELME [2]

To do (7) To do (8) To do (9) To do (10)

2.3 Kinect SDK

Nach dem Erscheinen der Hardware war diese auch schnell in Entwicklerkreisen gefragt. Doch Microsoft selbst gefiel dies zunächst nicht, denn der Konzern befürchtete, dass Cheater sich an ihren Spielen zu schaffen machen würden. So gab es zunächst kein SDK von Microsoft selbst. Die Open Source Gemeinde jedoch erkannte das Potential des Produktes schneller und entwickelte (eine Schnittstelle? zu) OpenNI, einem Framework, das die Auswertung von 3D-Sensordaten verschiedenster Hersteller unterstützt. Dieses Framework bietet somit durch seine Plattformunabhängigkeit die Möglichkeit unterschiedliche Betriebssysteme mit unterschiedlichen 3D-Sensoren zu kombinieren. [3] Microsoft zog nach und gab am 17. Juni 2011 die freie Beta Version des Microsoft SDKs frei. Somit hatte nun jeder Entwickler freien Zugang zu allen Kinectfunktionen, die auch von Microsoft selbst bisher genutzt wurden. Einer der Vorteile des Kinect SDKs besteht darin, dass die Skelett-Erkennung (siehe Kapitel 2.2 Kinect Software) ohne initiale Pose möglich ist, was im Gegensatz zum OpenNI-Framework steht. [3] Da für dieses Projekt die Plattformunabhängigkeit nicht relevant ist, sowohl aber die Skeletterkennung, wurde sich für das Microsoft SDK entschieden.

2.3.1 Richtiger Umgang mit dem SDK

To do (11)

3 Nao

Nao ist ein humanoider Roboter des französischen Herstellers *Aldebaran Robotic*. Die erste Version des Roboters wurde 2006 vorgestellt und unter anderem mit einem Investment durch Intel weiterentwickelt. Es gibt den Roboter in verschiedenen Ausführungen mit verschiedenen verbauten Sensoren und kostet ungefähr 10.000 Euro. [1]

Im folgenden Kapitel wird der Aufbau und die Funktionen der Hardware, sowie die zugehörige Software zur Bedienung und Programmierung des Nao vorgestellt.

3.1 Nao Hardware

Der in dieser Arbeit verwendete Nao ist der typ *NAO V3+*, *V3.2*. Dieser Typ ist 573.2mm hoch, 290mm tief und 273.3mm breit. In Abbildung 3.1 *Übersicht Nao V3.2* sind alle Sensoren und Aktoren aufgeführt.

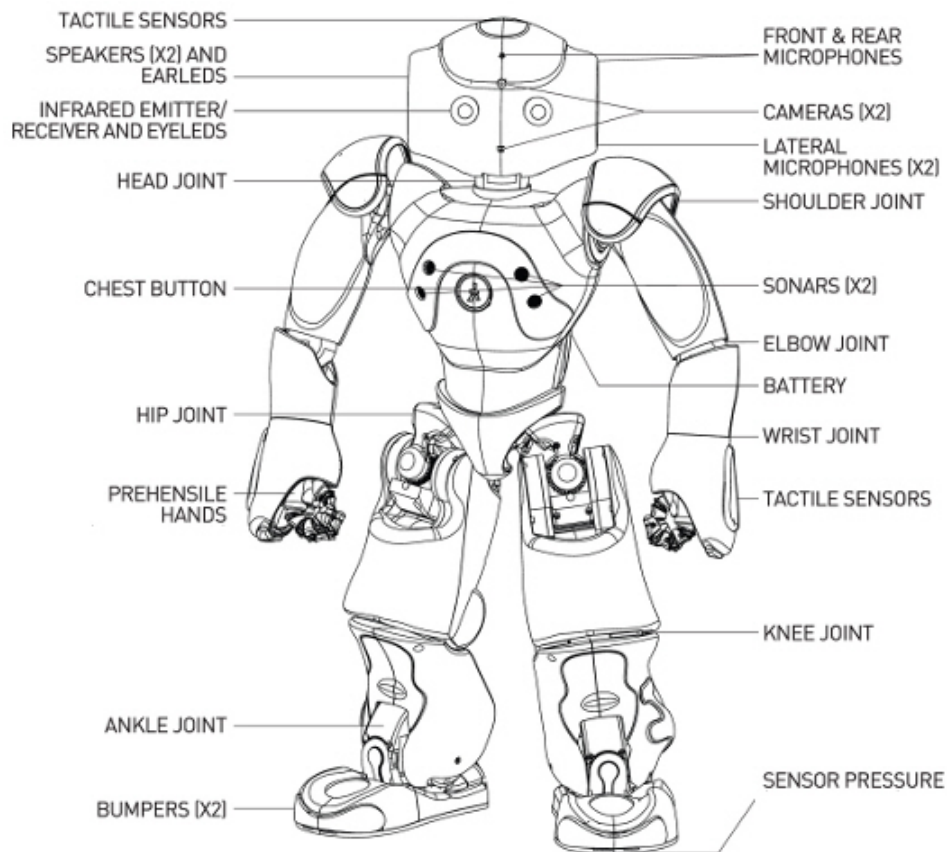


Abbildung 3.1: Übersicht Nao V3.2

Gelenke

Nao besitzt im Kopf, den beiden Armen, dem Becken und den beiden Beinen jeweils mehrere Gelenke. Damit ist eine umfangreiche Bewegung in alle Richtungen der drei Achsen möglich. Der Kopf lässt sich in Z-Richtung drehen und in Y-Richtung neigen, damit Nao auch räumlich sehen bzw. Objekte verfolgen kann. Die Arme besitzen die gleichen Gelenke wie in einem menschlichen Körper. Dazu gehören Schulter-, Ellenbogen- und Handgelenk. Das Schultergelenk dient dazu, den Arm zu heben/senken und ihn zu öffnen bzw. schließen. Das Drehen und Öffnen/Schließen des Unterarms geschieht durch das Ellenbogengelenk in Kombination mit dem Handgelenk. Die Finger der Hand können nur als Ganzes geöffnet bzw. geschlossen werden. Das Beckengelenk wird dazu benutzt, den Torso von Nao nach vorne oder hinten zu neigen. Die Beine bestehen aus drei Gelenken: Einem Hüft-, einem Knie- und einem Fußgelenk. Die Bewegungsfreiheit der Beine ähnelt dem des menschlichen Beins, wobei keines der Gelenke in Z-Richtung gedreht werden können.

Aktoren

Im Nao sind vier verschiedene Typen von Motoren verbaut. Diese unterscheiden sich im wesentlichen in ihrer maximalen Anzahl an Drehungen pro Minute, dem Drehmoment und der Drehzahlrückstellung. Dies ist wichtig, da nicht jedes Gelenk und der zugehörige Aktor mit der gleichen Masse belastet wird.

Elektronik & Sensoren

Das Herz von Nao ist dessen Motherboard mit einer x86 AMD CPU mit 500MHz. Der Arbeitsspeicher mit 256MB RAM und die 2GB Flash-Speicher befinden sich zusammen mit dem Prozessor im Kopf. Die Batterie mit rund 30Wh hält für die aktive Nutzung ca. 60min und die normale Nutzung ca. 90min.

Links und Rechts am Kopf befinden sich jeweils ein Lautsprecher und ein Mikrofon. Zusätzliche Mikrofone sind am Kopf auch noch vorne und hinten angebracht. Damit ist es Nao möglich, ein Geräusch zu lokalisieren und gegebenenfalls dahin zu folgen. Um gleichzeitig die Ferne und die Nähe visuell zu verarbeiten wurde über und unter den Augen jeweils eine VGA - Kamera mit einer Auflösung von 640x480 Pixeln installiert. Die Augen selbst dienen selbst zur Erkennung von Infrarotlicht, wobei auch hier in jedem Auge jeweils ein Sensor verbaut wurde.

Auf der Brust von Nao befinden sich Ultraschallsensoren zur Distanzermittlung (je 2 Emitter und Empfänger). Diese haben eine Auflösung von 1cm und eine Erkennungsweite von 0.25m bis 2.55m. Unter 0.25m erkennt Nao nur noch, dass ein Objekt im Weg ist, aber nicht wie weit es entfernt ist.

Sensoren zur Kontakterkennung befinden sich auf dem Kopf, dem Brustbutton, auf und neben den Händen, sowie vorne an den Füßen. Unter den Füßen befinden sich zudem noch piezoresistive Drucksensoren mit einem Arbeitsbereich von 0 bis 25 Newton. Damit lässt sich unter anderem erkennen, ob Nao nur auf einem Bein oder auf unebenen Untergrund steht.

To do (12)

3.2 Nao Software

3.3 Nao SDK

4 Realisierung

In diesem Kapitel wird näher auf das Zusammenspiel von Sensorik (Kinect) und Aktorik (Nao) eingegangen. Zunächst wird die Idee der Softwarearchitektur vorgestellt. Im Anschluss daran, wird auf die entworfenen Algorithmen näher eingegangen.

To do (13) To do (14) Wahl Bibliothek, warum?... To do (15) To do (16) To do (17)

4.1 Architektur

Die Architektur der Anwendung kann grob in zwei Teilbereiche gegliedert werden, den Sensorteil und den Aktorteil. Der Sensorteil ist dafür zuständig, die entsprechenden Gesten des Benutzers zu ermitteln und zu verarbeiten. Der Aktorteil der Anwendung ist dafür zuständig die Werte entsprechend auf die Wertebereiche des Nao-Koordinatensystems zu transformieren und diese dem Roboter zu übermitteln.

To do (18) To do (19) To do (20)

4.2 Prototyp

Anhand der obigen Überlegungen wurde der erdachte Algorithmus zunächst anhand eines Prototyps implementiert. Dieser sollte zunächst dazu dienen, das Microsoft Kinect SDK näher kennen zu lernen^{To do (21)}. Der erste Prototyp erfüllte folgende Funktionen:

-Kinect SDK Library -(Connect-Disconnect von Kinect) -Winkelerkennung des Benutzers mit Anzeige in Grad -Anzeige des Kamerabildes mit Gelenkkennzeichnung von Kopf und Armen

To do (22)

Literaturverzeichnis

- [1] Wikipedia contributors. *Nao (Roboter)*. [http://de.wikipedia.org/wiki/Nao_\(Roboter\)](http://de.wikipedia.org/wiki/Nao_(Roboter)), November 2013.
- [2] Joachim Hertzberg. *Mobile Roboter : Eine Einführung aus Sicht der Informatik*. Springer, Berlin, 2009.
- [3] Jarrett Webb. *Beginning Kinect programming with the Microsoft Kinect SDK*. Apress, New York, 2012.

To do...

- ☐ 1 (p. 1): titel!?
- ☐ 2 (p. 2): Inhalt der Studienarbeit
- ☒ 3 (p. 3): Kapitel schreiben
- ☐ 4 (p. 3): -Erstmals preiswerter 3D Sensor -Zunächst für Spielkonsole -Später für PC erhältlich -Aufbau: -Microphonarray zweck=richtung d. Geräusch -IR Sensor = Deep Stream -RGB Kamera = Bild -Rauminformationen für Software -Vorteil: Günstig, leicht zu entwickeln, da SDK vorhanden
- ☐ 5 (p. 3): Hardware Bilder
- ☐ 6 (p. 3): erklärungen zu Bilder
- ☐ 7 (p. 3): SDK Funktionen erklären: skelton, mehrere user, deep stream, color stream
- ☐ 8 (p. 3): Screenshot von Deep Stream (Kinect Studio)
- ☐ 9 (p. 3): Unterschied Microsoft SDK und Freie Implementierung OpenNI
- ☐ 10 (p. 3): Bild von Kinect-Koordinatensystem -¿ Bezug auf Nao
- ☐ 11 (p. 4): evtl. vergleich SDK OpenNI?
- ☐ 12 (p. 7): Bild Overview, vllt. einzelne Bilder rein?
- ☐ 13 (p. 8): Wahl Programmiersprache (warum? -¿Kinect und Nao gleich, Visual Studio)
- ☐ 14 (p. 8): (
- ☐ 15 (p. 8): Erklärung Webots, Choreograph, Nao Verbindung, Libraries...

- ☐ 16 (p. 8): Danach -¿ echter Nao im Laber -¿ Unterschied zur Simulation
- ☐ 17 (p. 8): Laborbedingungen Steuerungen, Vor- und Nachteile der Bedienung
- ☐ 18 (p. 8): Architekturidee: Input, Verarbeitung(evtl. Filter-¿Probleme Ruckeln?), Output,
- ☐ 19 (p. 8): Klassendiagramm
- ☐ 20 (p. 8): Sequenzdiagramm eines bestimmten Anwendungsfalls
- ☐ 21 (p. 8): getrennt zusammen?
- ☐ 22 (p. 8): Screenshot von Prototypprogramm