

# INDEPENDENT COURSEWORK

Independent Production

Fingererkennung  
mit Microsoft Kinect für Windows



**Hochschule für Technik  
und Wirtschaft Berlin**

*University of Applied Sciences*

Verfasser: Samuel Stein, s0534332  
Internationale Medieninformatik Master  
SS 2013

Betreuer: Prof. Dr. Carsten Busch

# Inhalt

Abstract .....	3
Einleitung .....	4
Gebärdensprache .....	5
Geschichte .....	5
(Deutsche) Gebärdensprache .....	5
Charakteristik und Grammatik .....	6
Fingeralphabet.....	6
Natural User Interface: Kinect.....	8
Funktionsweise .....	8
Frameworks und Bibliotheken .....	8
Experimente und Restriktionen .....	9
Implementierung .....	11
Softwarearchitektur.....	11
3D Hand- und Fingererkennung.....	12
Herausforderungen .....	13
Installation.....	14
Fazit .....	15
Anhang.....	16

## Abstract

---

This project tries to find out how the Microsoft Kinect for Windows could help to recognize and translate Sign language from the depth image of the sensor to ordinary speech or text. The independent coursework is separated in two parts: evaluation and prototypal implementation of a software who transcribes sign language symbols.

Keywords: Kinect, 3D hand- and finger detection, gesture recognition, sign language, nui, real-time, c#

Quellcode und Binärdateien sind unter <https://github.com/samuelstein/FingerSpellingKinect> zu finden.

## Einleitung

---

*„After the Command Line Interface (CLI) and the Graphical User Interface (GUI) the Natural User Interface (NUI) is the next big thing in human-machine interaction.“*

Stefan Stegmueller, Creator of Candescent NUI Library

Die technische Evolution schreitet voran und bietet neue Möglichkeiten der Interaktion und Kommunikation. Oftmals gehen damit auch Arbeitserleichterungen einher. Das Zitat von Stefan Stegmüller spiegelt dabei die Entwicklungen der letzten 40 Jahre in der IT-Branche wieder. Dabei ersetzt die Einführung einer neuen Technik bzw. Schnittstelle nicht zwangsläufig die vorangegangenen. Als prominentes Beispiel sei die Maus genannt oder das unverzichtbare Terminal für UNIX-Systemadministratoren.

Die Motivation für diese Arbeit resultierte auch aus einem interessanten Gespräch mit einer Freundin, die Dolmetscherin für Gebärdensprache ist. Des Weiteren fasziniert mich der Ansatz Technik für die Überwindung von Barrieren mit der Mensch-Maschinen Schnittstelle einzusetzen. Beispielsweise können körperlich beeinträchtigte Menschen mit den richtigen technischen Hilfsmitteln den Computer benutzen und somit mit anderen Menschen kommunizieren.

Die vorliegende Arbeit konzentriert sich wegen den technischen Möglichkeiten und Limitation der Kinect nicht auf die Erkennung vollständiger Gebärden sondern des deutschen Einhand-Fingeralphabets (siehe Anhang).

# Gebärdensprache

---

## Geschichte

Die Gebärdensprache blickt auf eine lange Geschichte zurück.

→ Indian Sign language

....

Der großen Durchbruch und die breite gesellschaftliche Akzeptanz wurde durch das Inkrafttreten des Behindertengleichstellungsgesetzes im Jahre 2002 ermöglicht. Es besagt, dass gehörlose Menschen nicht an offiziellen Stellen benachteiligt werden dürfen. Sie haben Anspruch auf einen Gebärdensprachdolmetscher.

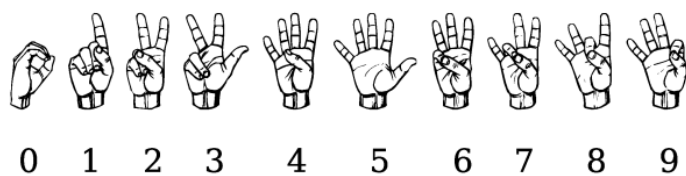
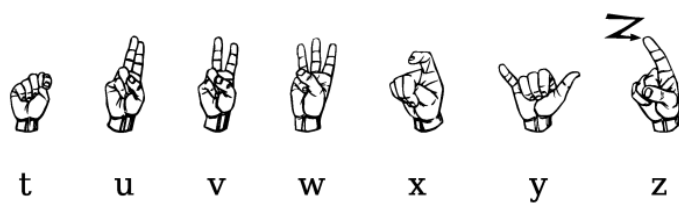
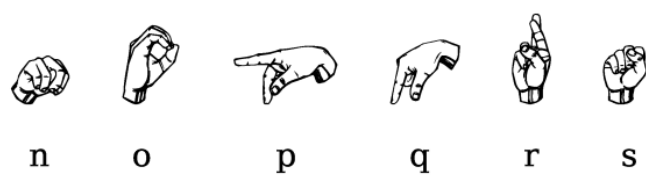
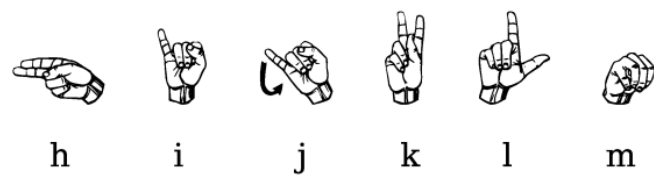
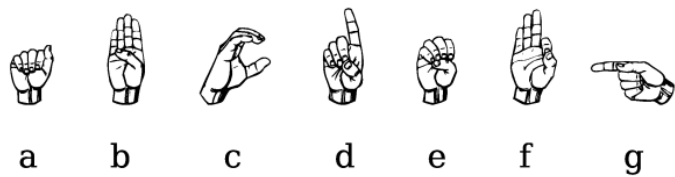
Sie lässt sich nicht nur auf den Ausdruck der Hände und die Gesten reduzieren sondern bezieht die ganze Körperhaltung mit ein. Mimik und Mundbild spielen eine wichtige Rolle in der Kommunikation.

Ein Blick in „die stille Welt“ offenbart

## (Deutsche) Gebärdensprache

Gebärdensprache besteht aus kombinierten Zeichen (Gebärden), die vor allem mit den Händen, in Verbindung mit Mimik und Mundbild (lautlos gesprochene Wörter oder Silben) und zudem im Kontext mit der Körperhaltung gebildet werden.

Das Fingeralphabet (auch als Fingersprache oder Daktylologie bezeichnet) dient dazu, die Schreibweise eines Wortes mit Hilfe der Finger zu buchstabieren.



## Charakteristik und Grammatik

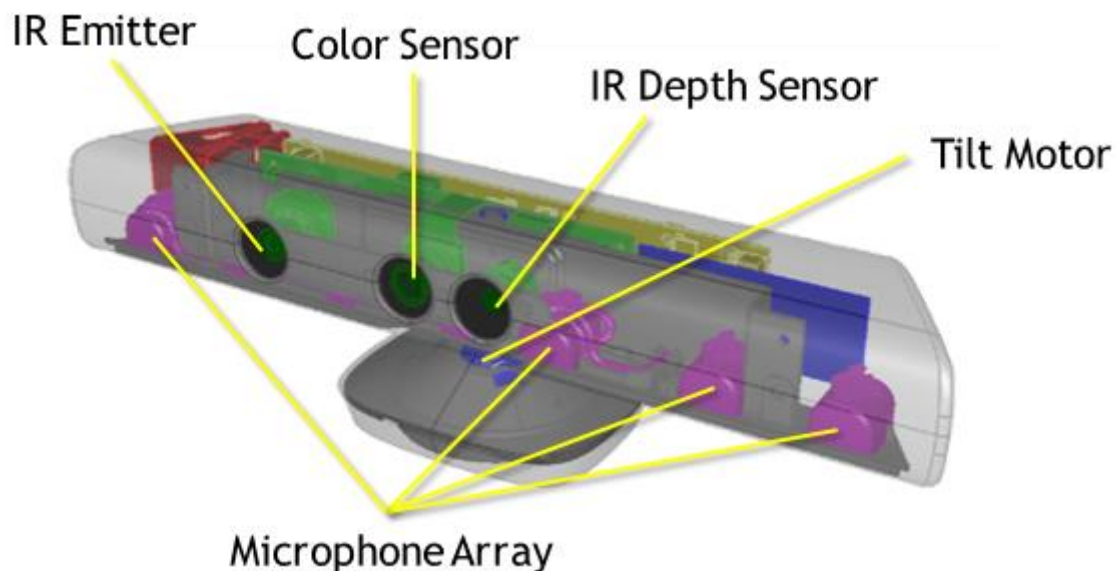
## Fingeralphabet



## Natural User Interface: Kinect

---

### Funktionsweise



Quelle: <http://msdn.microsoft.com/en-us/library/jj131033.aspx>

---

<b>Framerate</b>	30 FPS (Tiefen- und Farbsensor)
<b>Audioformat</b>	16-kHz, 24-bit mono (PCM)
<b>Auflösung</b>	640 x 480 Pixel

---

PrimeSense, the company who invented the Kinect hardware, has just released Version 2 of its OpenNI SDK and it's a major revision. What does it have to offer the Kinect or depth camera programmer?

OpenNI is a framework for 3D Natural Interaction sensors including PrimeSense's own reference device and the Asus Xtion sensor. It also supports Microsoft's Kinect but in this case you also have to install the Microsoft SDK to provide the special drivers that are needed.

### Frameworks und Bibliotheken



OpenNI + NITE

<http://www.openni.org/files/nite/>

Gute Alternative zu Candescent NUI.

OpenNI + NITE has a hand tracking feature where only a rectangle around the tracked hand is used for finger detection (in all other cases a fixed depth threshold is used, so all objects in that volume are processed).

Candescent NUI

<https://candescentnui.codeplex.com/>

Gute, ressourcenschonende Open-Source Bibliothek.

3 Gear Systems

<http://www.threegear.com>

Sie starteten ursprünglich nur mit einer kommerziellen Lizenz und einem Versuchsaufbau mit zwei Kinect Sensoren.

FORTH 3D Hand Tracking Library

<http://cvrlcode.ics.forth.gr/handtracking/>

Zu langsam und rechenintensiv

Microsoft Kinect for Windows SDK 1.5

<https://www.microsoft.com/en-us/kinectforwindows/>

Unterstützt bis dato keine Finger- und Handgestenerkennung.

## **Experimente und Restriktionen**

Dies mit der Kinect zu erkennen ist nahezu unmöglich, da die Auflösung und Verarbeitungsgeschwindigkeit (Real-time Systeme antworten in max 10 ms) zu gering ist.

Deshalb wird sich auf die Erkennung von Finger und Fingergesten fokussiert. Das unterstützt aber das Kinect SDK von Microsoft (noch) nicht (<http://social.msdn.microsoft.com/Forums/en-US/kinectsdk/thread/dd4d3e9a-fa8f-4ff7-9262-d9e8caa5952a> ). Es wird auf einen experimentellen Open-Source Treiber der ursprünglichen Entwicklungsfirma der Kinect PrimeSense zurückgegriffen. In Verbindung mit OpenNI und NITE lässt sich das Projekt umsetzen. Demos: [http://www.youtube.com/watch?v=H\\_UqGuAVp1M](http://www.youtube.com/watch?v=H_UqGuAVp1M)

Abhängigkeit von der Helligkeit und des Kontrastes.

Geringe Tiefenauflösung

Max 30 Frames möglich.

# Implementierung

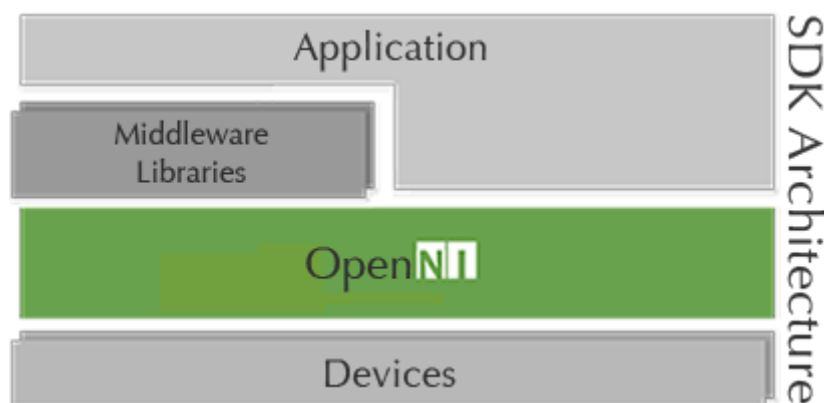
---

## Softwarearchitektur

Vor dem Beginn der Entwicklung von Software mit der Kinect bieten sich grundsätzlich zwei Möglichkeiten um diese programmatisch anzusprechen.

1. Offizielles Microsoft Kinect SDK + Treiber  
Nachteil: kein Finger- und Handtracking möglich
2. Experimenteller PrimeSense Treiber mit OpenNI + NITE  
Vorteil: Unterstützung von Handtracking sowie generische Softwareschnittstelle

Der entstandenen Applikation liegt nicht das Microsoft Kinect SDK zugrunde sondern die Open-Source Programmbibliothek OpenNI (<http://www.openni.org/>). Sie zeichnet sich besonders durch die Abstraktion der Schnittstelle zur Kinect aus, was dazu führt das auch Sensoren von anderen Herstellern wie Asus und Primesense verwendet werden können. Das ist vorteilhaft, da die Applikation nicht der Politik des Herstellers ausgeliefert ist. Weiterhin bietet OpenNI noch weitere Middleware Bibliotheken zur Erkennung von Händen, Objekten, Volumina an ([http://www.openni.org/software/?cat\\_slug=file-cat1](http://www.openni.org/software/?cat_slug=file-cat1)) Siehe auch „Frameworks und Bibliotheken“.



Treiber PrimeSense

Microsoft Visual Studio 2012

C# Windows Forms Applikation.

MVC

GestureDetector Singleton

NoSQL Datenbank für Gesten

Eventbasiert

WebView:

<http://localhost:8080/raven/studio.html#/documents?database=%3Csystem%3E&collection=Gestures>

Reaktives System, Echtzeitsystem

### **3D Hand- und Fingererkennung**

Hierbei muss erkannt werden:

- Umrisse der Hand
- Anzahl der Finger
- (Handvorderseite oder –rückseite)
- (Bewegung der Hand)

Anschließend kann das Volumen der Hand und der Mittelpunkt der Hand berechnet werden. Optimal funktioniert die Fingererkennung, wenn die Entfernung zum Gerät zwischen 500 – 800 mm liegt.

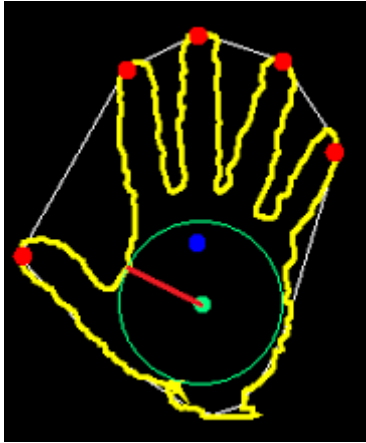
Arbeitsweise der Candescent NUI Bibliothek:

Handerkennung:

1. Pixel (der Hände) vom Tiefenbild erkennen (zRange)
2. Projektion in 2D Raum
3. Pixel mittels K-Means Algorithmus in zwei Gruppen unterteilen
4. Konvexe Hülle finden: Graham Scan
5. Handkonturen mittels modifiziertem Moore-Neighbor Tracing Algorithmus finden

Fingererkennung:

6. Fingerspitzen erkennen: Triangulation
7. Mittelpunkt der Handfläche finden
8. Hände zwischen beiden Frames synchronisieren
9. Finger erkennen



Die meisten Beispielprojekte und Bibliotheken erlauben es nur Standardgesten wie Wischen nach rechts, links, oben und unten, sowie Winken durchzuführen. Siehe: ([http://www.openni.org/software/?cat\\_slug=file-cat1](http://www.openni.org/software/?cat_slug=file-cat1), <https://kinectdtw.codeplex.com/>)

## **Herausforderungen**

Den richtigen Algorithmus zur Berechnung der Ähnlichkeit des Bildes finden.

Guter Kompromiss zwischen Performance und Berechnungsgenauigkeit

Richtige Treiber und Bibliotheksversionen finden

## Installation

Die folgenden Schritte sollten in der vorgegebenen Reihenfolge abgearbeitet werden, damit die Software ordnungsgemäß funktioniert.

Wichtig dabei ist vor allem die Einhaltung der Versionsnummern der Bibliotheken.

Außerdem ist die Software auf die Kinect für Windows optimiert.

Der gesamten Anwendung liegt eine 64-Bit Architektur zugrunde, deswegen werden die Treiber und Bibliotheken auch nur in 64-Bit Version installiert.

### 1. Installation von OpenNI SDK v1.5.4.0 x64

Download der Bibliothek von: <http://www.openni.org/openni-sdk/openni-sdk-history-2/>

Anschließend installieren.

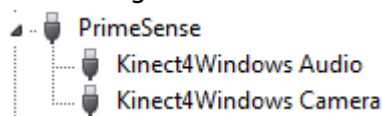
### 2. Installation von alternativen Kinect Treiber (SensorKinect 5.1.2.1 x64)

Der Treiber ist unter <https://github.com/avin2/SensorKinect/tree/unstable/Bin> zu finden. Er sollte SensorKinect093-Bin-Win64-v5.1.2.1.msi heißen.

Anschließend installieren.

### 3. Installation prüfen

Nach erfolgreicher Installation muss im Gerätemanager folgendes zu sehen sein:



Danach am besten den NiViewer64 im Samples-Ordner von OpenNI starten (Im Programme Menü zu finden). Falls die Installation erfolgreich war sollte der NiViewer64 ein Video- und Tiefenbild anzeigen.

Weitere gute Tipps und Vorgehensweisen sind auch unter den folgenden Adressen zu finden:

- <http://kinect-i.blogspot.com.br/2012/05/how-to-install-and-use-openni-microsoft.html>
- <http://www.kinecthacks.com/guides/install-kinect-on-your-pc-and-start-developing-your-programs-disclaimer/>
- <http://ramsrigoutham.com/2012/06/07/installation-of-kinect-on-windows-7-openni-sensor-kinect-and-nite/>

### 4. Installation des Programms FingerSpellingApp

Download von <https://github.com/samuelstein/FingerSpellingKinect>.

Den Installer im Ordner FingerSpelling ausführen. Anschließend starten.

## Fazit

---

Bedauerlich, dass Microsoft in ihrem SDK keine Möglichkeit mitbringen, Hand- bzw. Fingererkennung durchzuführen.

Lessons learned.

## Anhang

---

Literaturverzeichnis

Deutsches Fingeralphabet

### Links

Kinect-Hack erkennt Gebärdensprache: <http://www.computerwoche.de/a/kinect-hack-erkennt-gebaerdensprache,2486073>

<http://kommunikation-21.blogspot.de/2011/03/gebardensprache-mit-kinect.html>

<http://www.geekosystem.com/kinect-sign-language/>

Microsoft Tutorials: <http://www.pil-network.com/Resources/Tutorials/>

Gebärdensprache für Gamer: <http://www.golem.de/1008/77077.html>

Gesture Keyboarding: <http://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fmetahtml%2FPTO%2Fsearch-adv.html&r=1&f=G&l=50&d=PG01&p=1&S1=20100199228.PGNR.&OS=dn/20100199228&RS=DN/20100199228>

<http://www.engadget.com/2012/07/25/sigma-kinect-lightsaber-sign-language/>

CopyCat and Kinect: <http://www.cats.gatech.edu/content/copycat-and-kinect>

<http://kinectsignlanguage.codeplex.com/documentation>

<http://www.hansottotheater.de/preise-besucherservice/angebote-fuer-gruppen/vorstellungen-fuer-taube-hoerbehinderte-tamidos.htm>

Konkurrenz: [http://www.asus.com/Multimedia/Motion\\_Sensor/Xtion\\_PRO/#overview](http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO/#overview),  
<https://leapmotion.com/>

Microsoft's Next Kinect May Feature Finger Tracking:

<http://www.webpronews.com/microsofts-next-kinect-may-feature-finger-tracking-2012-10>, <http://arstechnica.com/gaming/2012/10/microsoft-research-gives-kinect-style-controls-the-fingers/>

Hack the Kinect to Control DJ Lasers & Lighting: <http://hacknmod.com/hack/hack-the-kinect-to-control-dj-lasers-lighting/>

KinectSDK OpenNI Bridge: <http://code.google.com/p/kinect-mssdk-openni-bridge/>

OpenKinect: [http://openkinect.org/wiki/CSharp\\_Wrapper](http://openkinect.org/wiki/CSharp_Wrapper)

Finger Tracking mit zwei Kinects: <http://www.i-programmer.info/news/194-kinect/4891-kinect-tracks-a-single-finger-with-a-new-sdk.html>, <http://www.threegear.com>



Its two shortcomings are that it requires calibration and it only recognizes a fixed set of gestures. The company is working to remove both constraints.

Gesture Recognition: <http://www.openni.org/files/igesture3d/#.UUSJvRxhWcl>,  
<http://www.covii.pt/viim/>

Kinect for the Blind: <http://www.zoneos.com/kinectfortheblind.htm>

<http://www.fingeralphabet.org/alphabets/germany-v02>