Fakultät Technik

Masterstudiengang Mechatronik

2. Semester

Dokumentation

Bildverarbeitung Praktikum
Prof. Matthias Rätsch

Gesture Recognition with MS Kinect Sensor

Shasindran Poonudurai	700461
Alexander Mielchen	692163

Sommersemester 2014

Inhaltsverzeichnis

1 Aufgabenstellung

Ziel der Praktikumsaufgabe ist eine beispielhafte Applikation zur Erkennung von Gesten mit einem RGBD-Sensor (MS Kinect) auf dessen Grundlage z.B. Fahrbefehle für den SCITOS Assistenzroboter generiert werden sollen.

Ausgehend von dieser Aufgabenstellung ergeben sich folgende Teilaufgaben:

- Kinect Sensor unter Linux erkennen können
  - Treiber auswählen und installieren
  - Schnittstelle für Daten Akquisition auswählen und installieren
- Beispiel Programm über Terminal ausführen (in OpenNi Installation enthalten)
- Eclipse als Entwicklungsumgebung einrichten
- Beispiel Programm in Eclipse importieren und lauffähig machen
- Wenn Beispiel Programm in Eclipse lauffähig, dieses auf SCITOS exportieren und Fahrbefehle erzeugen
- Alternativen recherchieren
- Beispiel Applikation erstellen

2 Grundlagen

Hier werden kurz die verwendeten Softwarekomponenten beschrieben. Prinzipiell wurden **zwei verschieden Lösungswege** verfolgt. Zum einen die Auswahl und Installation von open source Softwarekomponenten und Treibern, die eine freie Programmierung mit Hilfe von geeigneten Bibliotheken und einer Entwicklungsumgebung, hier Eclipse, ermöglichen sollen. Zum anderen wurde die Verfügbarkeit von fertigen Programmen zur Gestenerkennung überprüft. An dieser Stelle sind wir auf das Tool KineticSpace aufmerksam geworden, welches im Anschluss ebenfalls beschrieben wird.

Zu Beginn wurde versucht die Kinect aus einer VMWare heraus zu starten. Obwohl die Kinect zwar vom System erkannt wurde konnte keines der Beispielprogramme nach der Installation entsprechender Treiber ausgeführt werden. Daher wurde auf dem Rechner **tecraetsch-2** ein **natives Ubuntu 14.04 LTS**

parallel zu Windows installiert (**Passwort: bv.stud3nt**). Im nativen Ubuntu ist das Ausführen von Beispielprogrammen mit nachfolgend beschriebenen Varianten möglich.

2.1 Variante 1 - OpenNI, NITE und Kinect Driver

OpenNI (Open Natural Interaction) ist eine „non-profit“ Organisation, die sich auf die Zertifizierung und Verbesserung der Kompatibilität von natürlichen User-Interfaces und organischen User-Interfaces (OUI – organic user interface, ist definiert als Schnittstelle mit einem Display welches nicht wie gewohnt flach ist, sondern in seiner Form veränderbar), Anwendungen die solche Geräte verwenden und „Middleware“ die die Verwendung von solchen Geräten erleichtert, spezialisiert hat. Natürliche User-Interfaces oder auch Natural Interaction Devices sind Devices die Körperbewegungen und Geräusche aufnehmen bzw. verfolgen (tracken) können um dadurch eine natürlichere Interaktion mit Computern zu ermöglichen.

Die Organisation wurde im November 2010 gegründet. Eines der Hauptmitglieder ist **PrimeSense**, das Unternehmen hinter der Technologie, die in der Kinect zur Bewegungserkennung zum Einsatz kommt. Im Dezember 2010 hat PrimeSense seinen eigenen open source Treiber veröffentlicht. Zusammen mit einer „motion tracking middleware“ – **NITE**.

PrimeSense wurde Ende 2013 von Apple gekauft und die Website *OpenNI.org* am 23. April 2014 geschlossen. Unmittelbar im Anschluss wurden Dokumentationen und Binaries von Unternehmen und Organisationen, welche OpenNI verwendeten wie z.B. **Structure.io**, erhalten.

Das OpenNI Framework stellt ein Set von **open source APIs** zur Verfügung. Diese APIs sind darauf ausgelegt einen Standard für Anwendungen im Bereich *natural interaction devices* zu etablieren.

Die APIs bieten Support für:

- Spracherkennung
- Hand Gesten
- Body Motion Tracking

Quelle: http://en.wikipedia.org/wiki/OpenNI

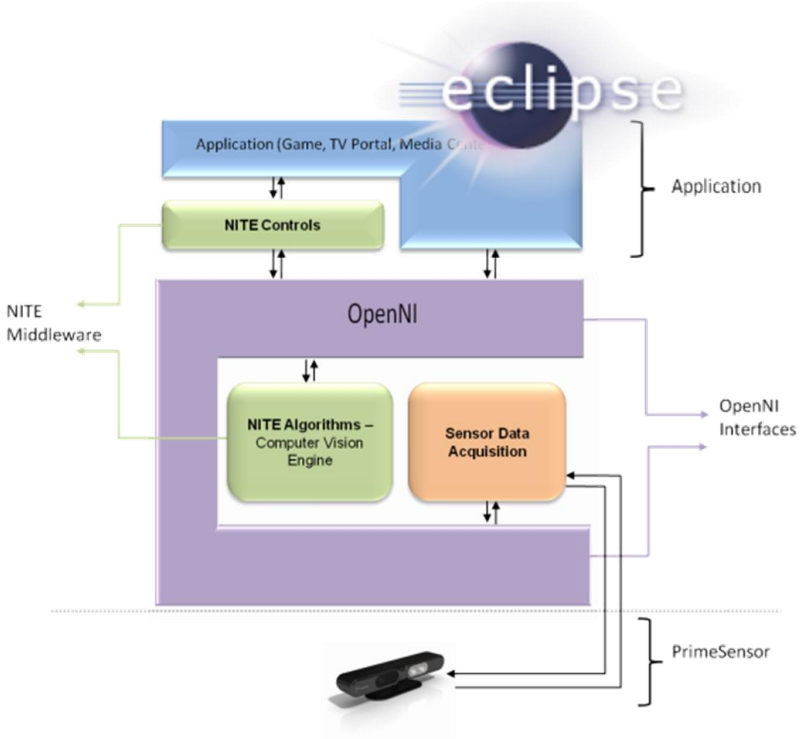

Figure 3: Layered View of Depth Acquiring and Processing

Abbildung 1 Auf OpenNI basiertes Framework

2.2 Variante 2 - KineticSpace

KineticSpace ist ein Softwaretool das jedermann ermöglicht eigene Ganzkörper-Gesten aufzunehmen und zu erkennen welche auf einem Tiefenbild, so wie z.B. von der Kinect geliefert, basieren. Fünf Highlights des Tools sind, dass die Gesten:

- **leicht zu trainieren sind**
  der Anwender kann das System direkt durch dich Aufnahme einer Bewegung trainieren
- **personenunabhängig sind**
  das System kann von einer Person trainiert und von einer anderen verwendet werden
- **orientierungsunabhängig sind**
  Gesten werden erkannt, auch wenn die trainierte und die getestete Geste nicht die gleiche Orientierung haben
- **geschwindigkeitsunabhängig sind**
  das System kann Gesten erkennen, auch wenn diese schneller oder langsamer ausgeführt werden wie die trainierte Geste
- **konfiguriert werden können**
  System und Gesten können über ein XML File konfiguriert werden

KineticSpace ermöglicht die Kontrolle von third party Software über das **OSC Protokoll**. Die Software wurde in Processing geschrieben und basiert auf SimpleOpenNI, OpenNI und NITE.
Quelle: KineticSpace User Manual

Entwickelt wurde das Tool von Herrn **Matthias Wölfel von der HS Pforzheim (***matthias.woelfel@hs-pforzheim.de)*. Durch mehrfachen eMail Kontakt zu Herrn Wölfel hat dieser uns die Pro Version von KS zur Verfügung gestellt. Die HS Reutlingen kann das

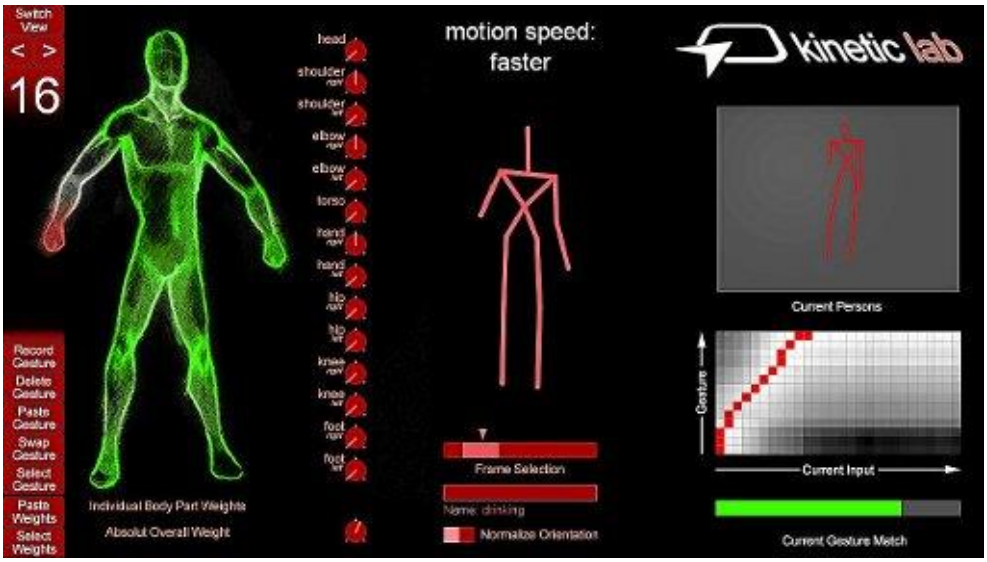

Abbildung 2 KineticSpace

Tool somit für nicht kommerzielle Projekte im Rahmen der Hochschule verwenden mit der ausdrücklichen Bitte seitens Herrn Wölfel ihn in entsprechenden Arbeiten zu **zitieren.** Außerdem bittet er um eine kurze **Beschreibung** des realisierten Projektes in **englischer Sprache** sowie ein kurzes **Video**.

Abbildung 2 KineticSpace, zeigt eines der Menüs des Tools. Hier können neue Gesten eingelernt werden. Sind die Gesten einmal abgespeichert kann darüber hinaus eine Gewichtung aller Körperteile vorgenommen werden. Besteht eine Geste beispielsweise nur aus Bewegungen des Oberkörpers können die Beine und Füße für die Wiedererkennung ausgeblendet werden. Die Bedienung der Software wird hier nicht explizit aufgeführt da diese nach dem Programmstart selbsterklärend ist. Es wird außerdem im Kapitel Ergebnisse nochmals auf den Umgang mit KineticSpace eingegangen.

3 Ergebnisse

In diesem Abschnitt werden die Ergebnisse und Schwierigkeiten bei der Umsetzung der zuvor beschriebenen Varianten aufgezeigt.

3.1 Programmierung mit Eclipse

An dieser Stelle sei darauf hingewiesen, dass für die Verwendung der in Kapitel 2.1 vorgestellten Softwarekomponenten die Einhaltung der in Kapitel 6.1**Fehler! Verweisquelle konnte nicht gefunden werden.** angegebenen Versionen zwingend erforderlich ist. Es handelt sich hierbei nicht um die aktuellsten Versionen! Da aber die Bereitstellung von Dokumenten und Source Code nach dem Verkauf von OpenNI an Apple nicht ideal ist, war es nicht leicht zueinander kompatible Versionen zu finden. Kombinationen neuerer Treiber sind denkbar aber uns nicht bekannt. Nach erfolgreicher Installation (Kapitel 6.1) können sämtliche Beispiele von OpenNI und NITE über die Konsole gestartet werden. Dazu in den entsprechenden Ordner navigieren und über die Konsole starten (vgl. Abbildung 3 Starten eines Beispiels aus der Konsole). Die Kinect sollte angeschlossen und vom System erkannt worden sein.

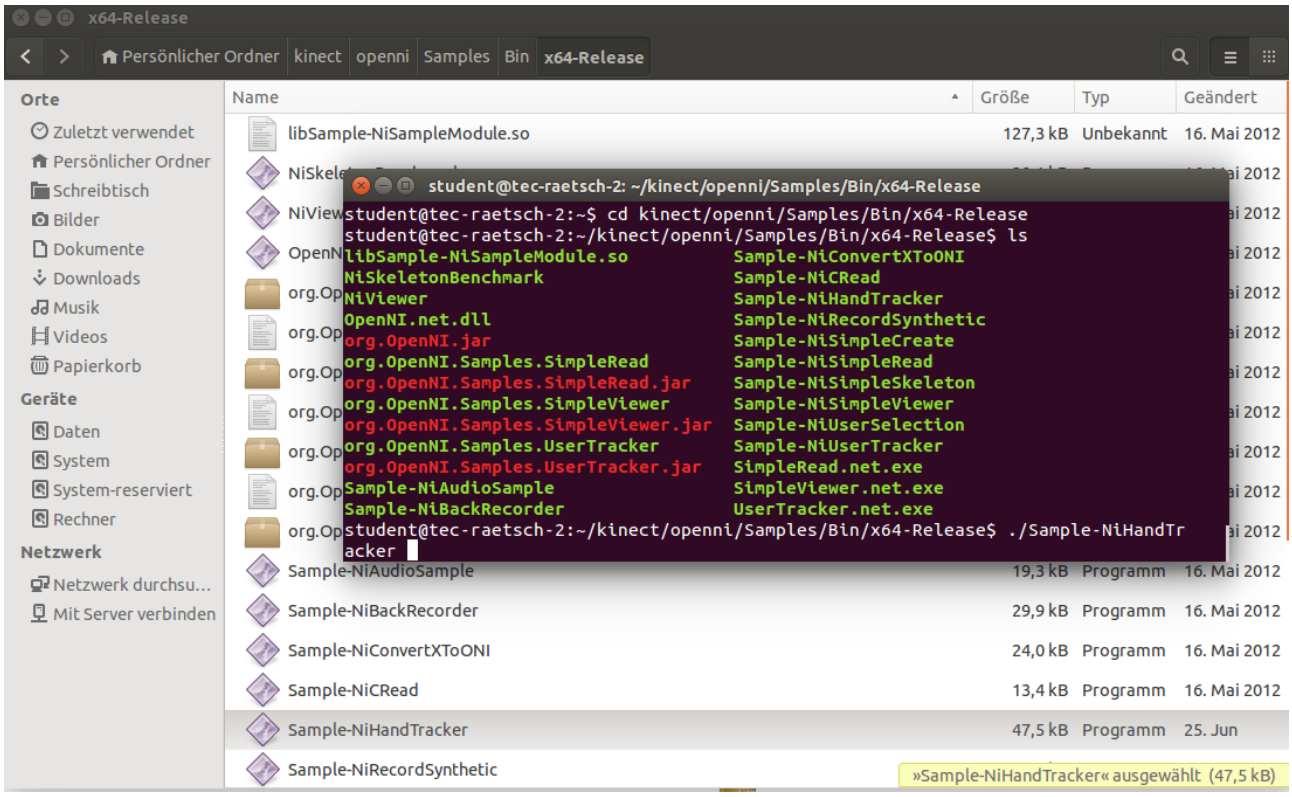

Abbildung 3 Starten eines Beispiels aus der Konsole

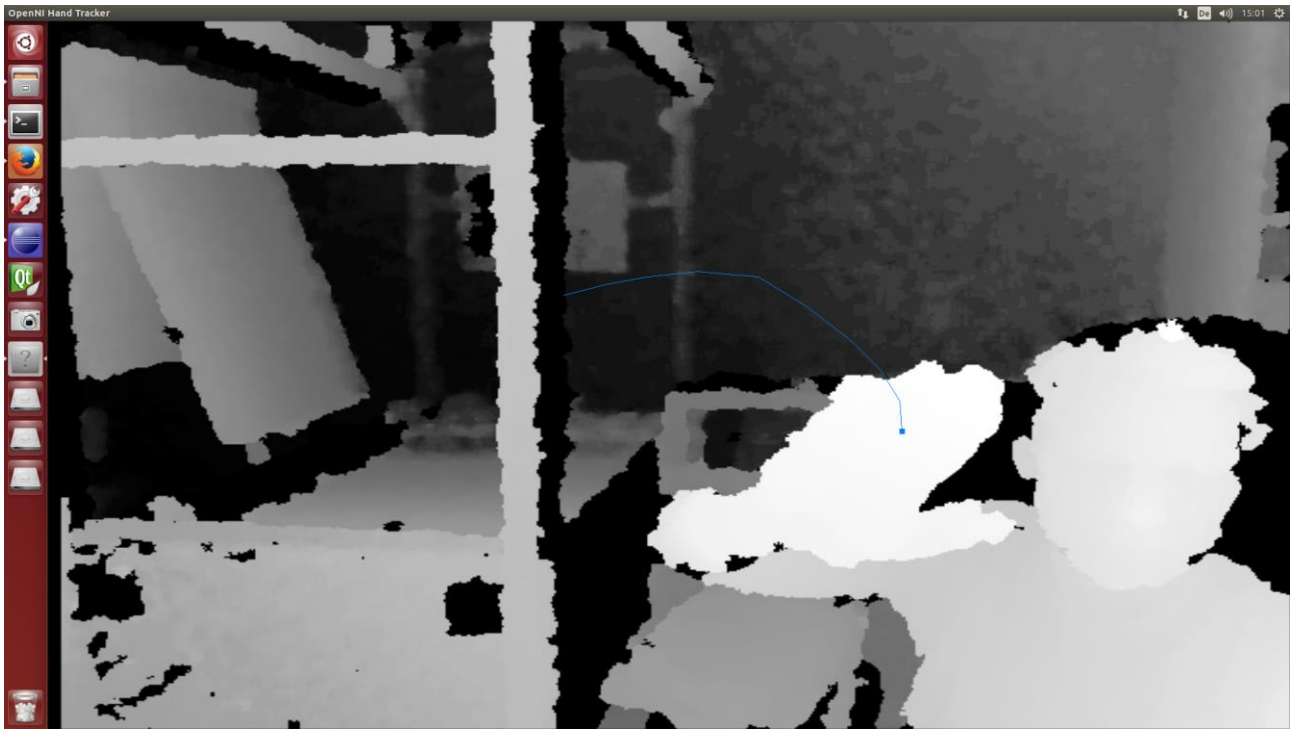

Abbildung 4 Ausführung des Beispiels Sample-NiHandTracker

Unsere Wunschlösung war die beispielhafte Einbindung eines dieser Beispiele in Eclipse, so dass dort der Source Code entsprechend erweitert werden kann um später auf dieser Grundlage eine Unit im Mira Center zu erzeugen.

Für das recht einfache Beispiel *NiSimpleViewer* hat das auch funktioniert (gilt für die Einbindung in Eclipse, Unit im Mira Center wurde nicht versucht). Es wurde dafür ein workspace im Verzeichnis *Persönlicher Ordner/workspace* angelegt. Der dortige *include* Ordner enthält Kopien der drei Ordner welche während der Installation erstellt wurden (da wir nicht mit Sicherheit sagen konnten welche Ordner/Files tatsächlich notwendig sind wurden kurzerhand die kompletten Ordner kopiert). In Eclipse wurde ein neues C++ Projekt aus vorhandenem Quellcode erzeugt (aus dem Ordner *Persönlicher Ordner/workspace/include/openni/Samples/NiSimpleViewer*). Das Projekt kann in Eclipse gebaut werden und eine entsprechende neue ausführbare Datei wird erzeugt (vgl. Abbildung 6). Die so erzeugte Datei kann im Anschluss wieder über das Terminal ausgeführt werden. Die wichtigsten Einstellungen in der Eclipse Umgebung sind in nachfolgenden Screenshots zu sehen.

Abbildung 5 Ordnerstruktur der verwendeten Komponenten

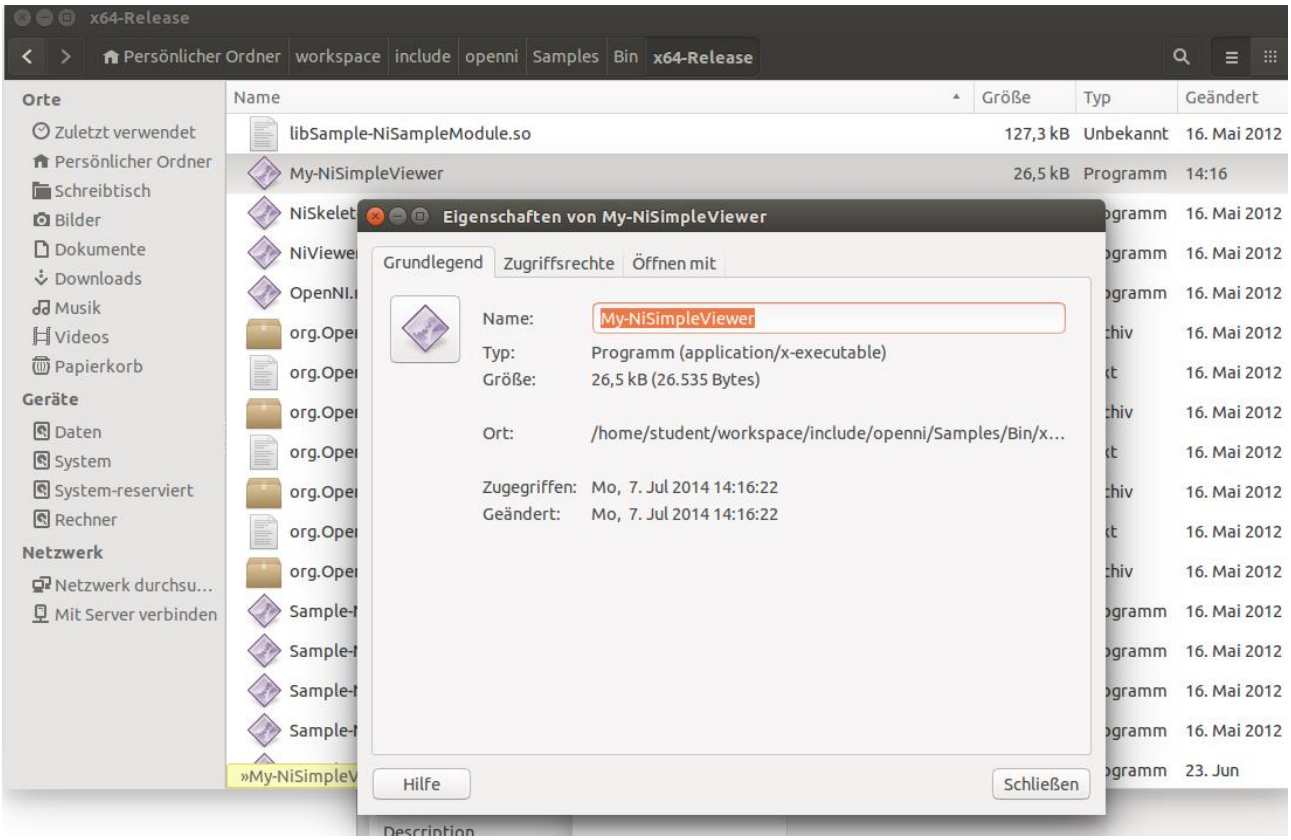

Abbildung 6 Build Ordner indem die ausführbaren Dateien erzeugt werden

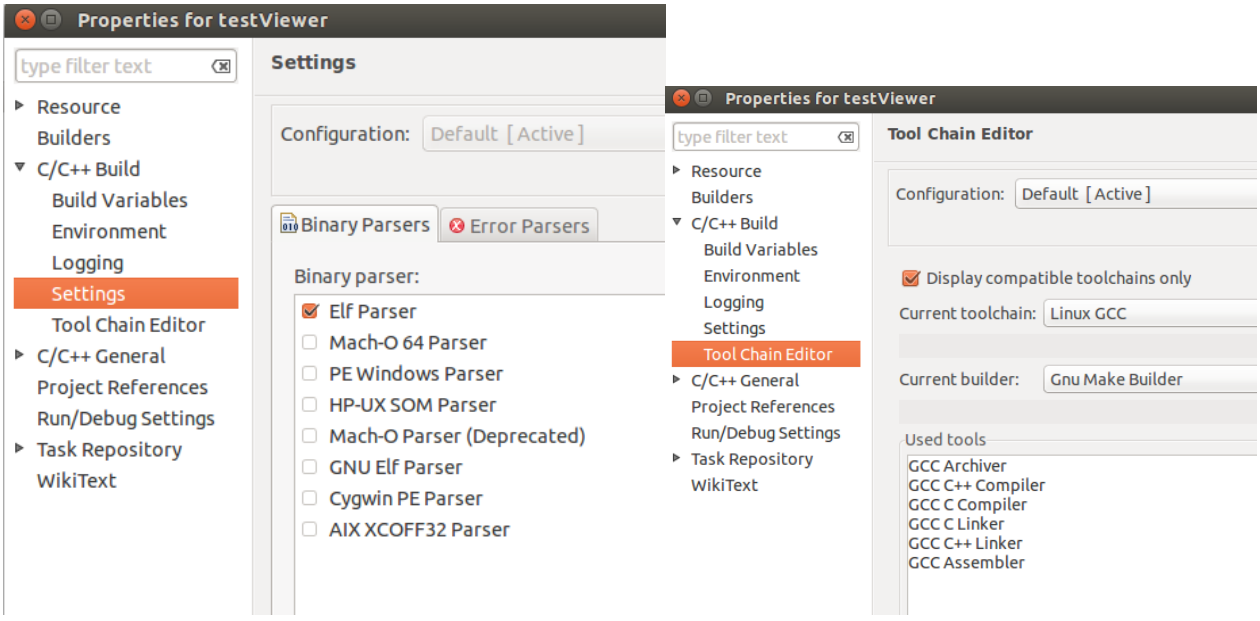

Abbildung 7 Properties des NiSimpleViewer

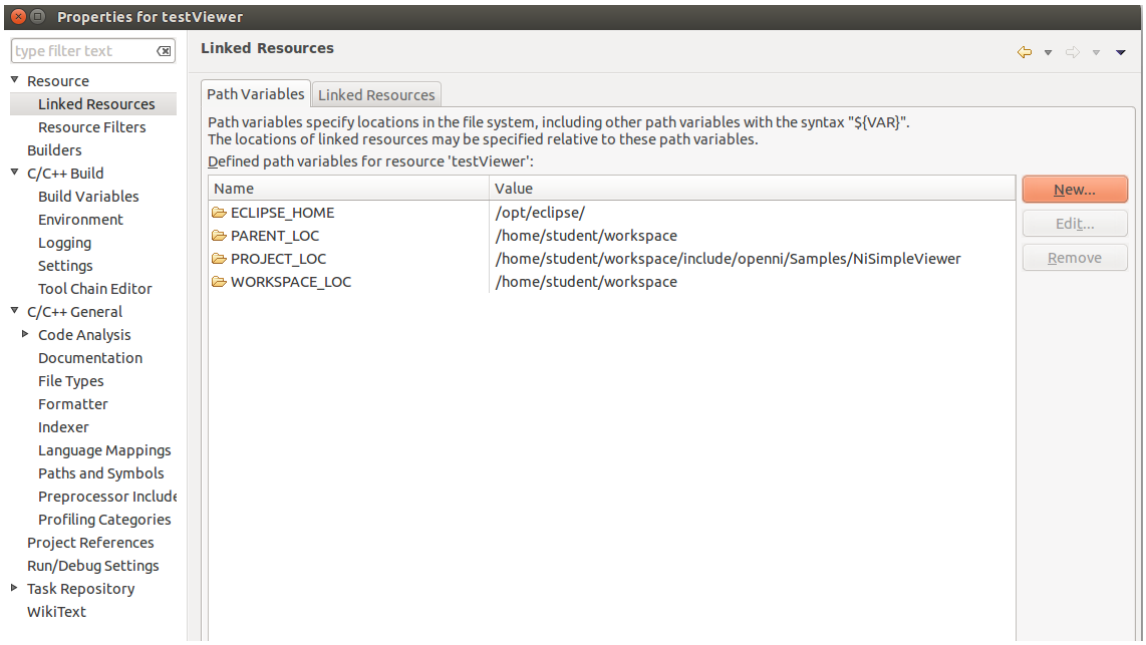

Abbildung 9 Linked Ressources des NiSimpleViewer

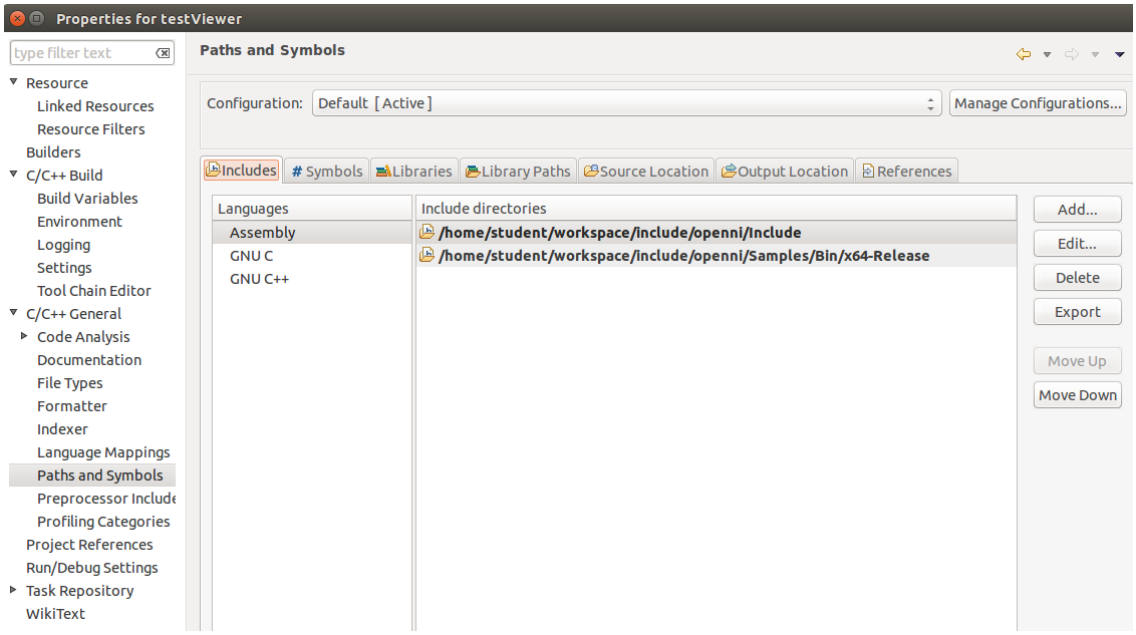

Abbildung 8 Paths and Symbols Einstellungen des NiSimpleViewer

Das gleiche Vorgehen haben wir dann für das Beispiel *My-HandTracker* versucht. Hier gab es jedoch Komplikationen die wir letztendlich nicht lösen konnten. Es wurden verschiedenste Möglichkeiten ein Projekt in Eclipse zu erstellen und die entsprechenden Bibliotheken und Header einzubinden versucht. Am Ende blieben etliche Fehlermeldungen des Compilers bzw. Linkers. Trotz vieler Fehlermeldungen war es dennoch möglich das Projekt zu „bauen“. Es erschien wie schon im ersten Beispiel eine neue (vgl. Erstellungs-

datum) ausführbare Datei die dann sogar im Terminal ausgeführt werden konnte. Letzten Endes wurde diese Variante aufgrund der vielen nicht lösbaren Fehler nicht weiter verfolgt. Die Einstellungen in Eclipse werden dennoch in den folgenden Abbildungen dargestellt.

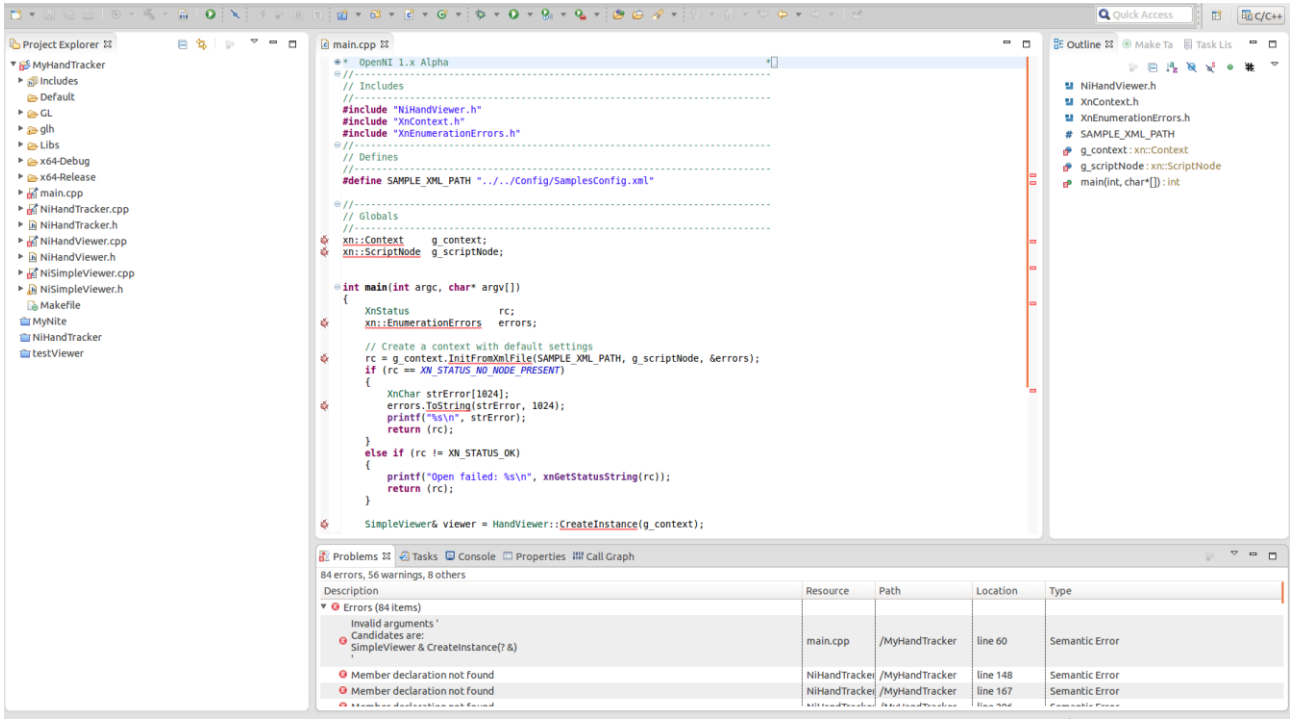

Abbildung 11 Linker Errors beim NiHandTracker

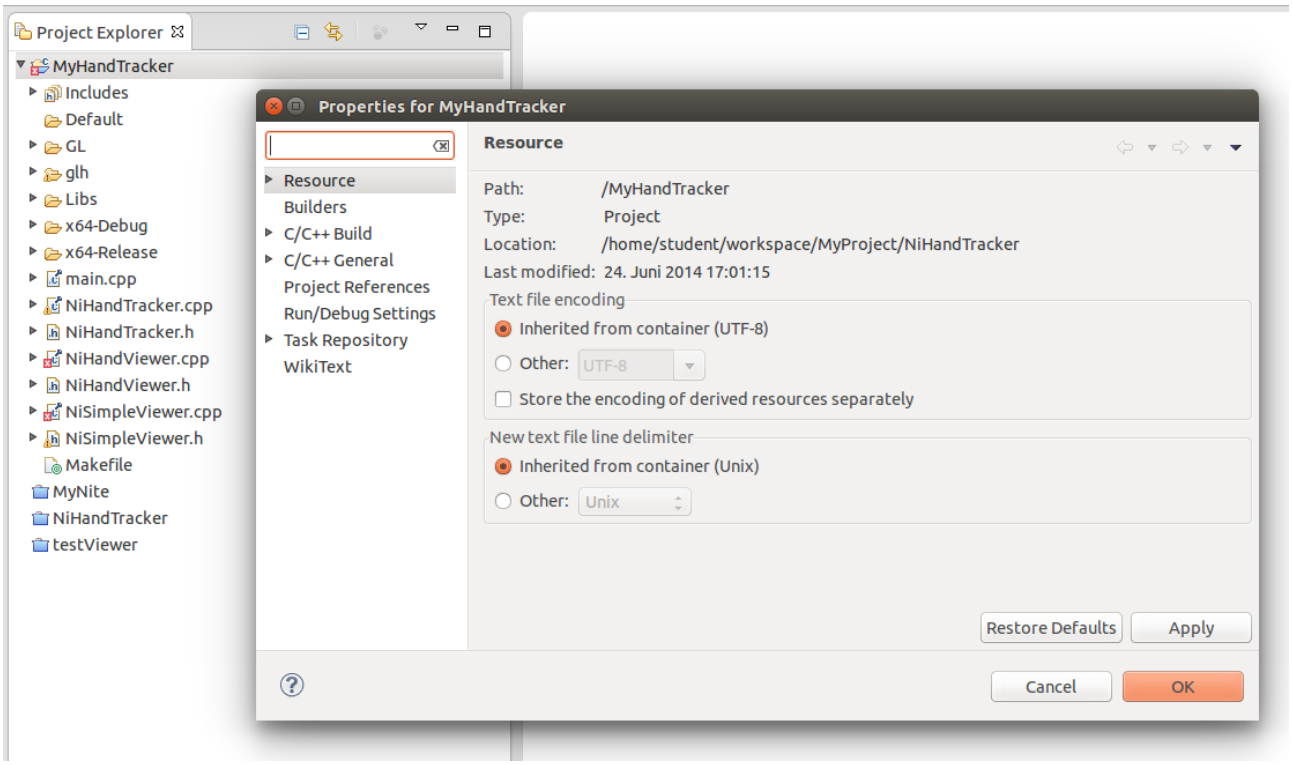

Abbildung 10 Einstellungen für MyHandTracker - Resource

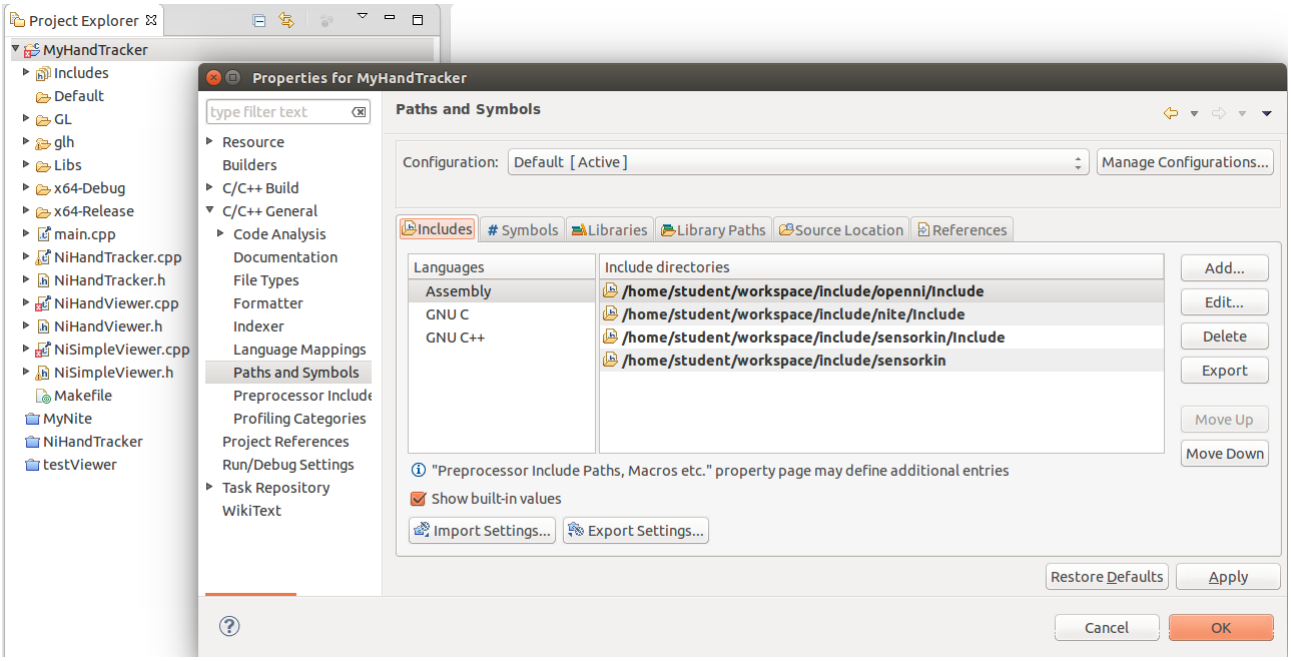

Abbildung 12 Einstellungen für MyHandTracker - Paths and Symbols, Includes

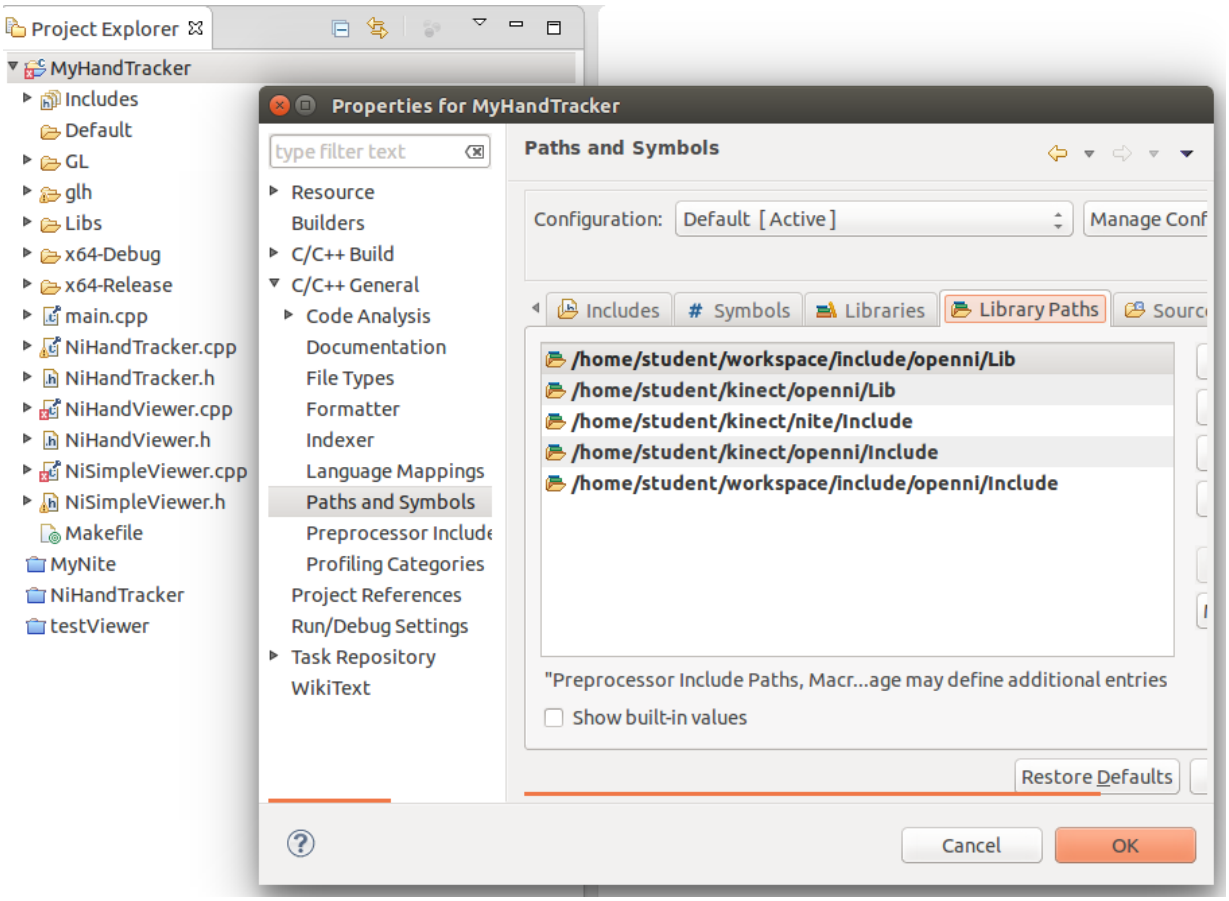

Abbildung 13 Einstellungen für MyHandTracker - Paths and Symbols, Library Paths

3.2 Anwendung von KineticSpace

Da KineticSpace in erster Linie unter Windows entwickelt wird kann es durchaus zu Bugs unter den Linux Versionen kommen. Das ist auch der Grund warum verschiedene Versionen vorhanden sind. Die **Version 25** wurde von uns getestet und funktioniert soweit.

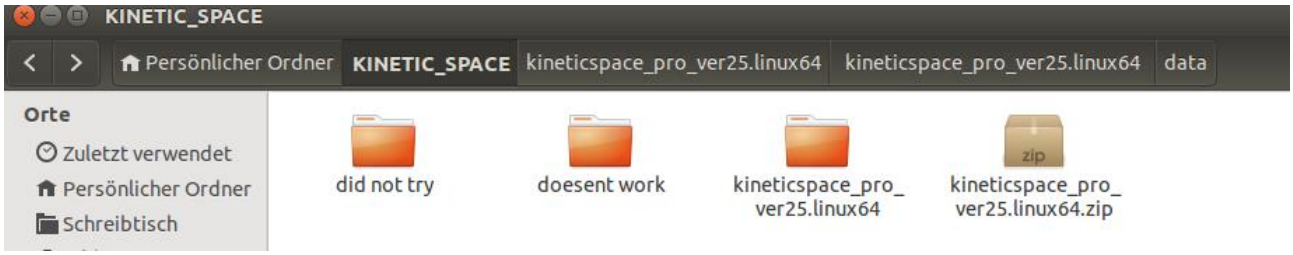

Abbildung 14 Ordnerstruktur KineticSpace

Der große Vorteil dieses Tools liegt ganz klar darin, dass die komplette Gestenerkennung inklusive der Teach-In Funktionalität bereits auf einem sehr hohen Niveau realisiert ist. Anstatt sich um die äußerst komplexe Aufgabe einer Gestenerkennung von Grund auf neu zu entwickeln kümmern zu müssen, kann man hier sofort damit beginnen eine Applikation für den SCITOS zu realisieren. D.h. es wird lediglich folgende Informationen von der Software KineticSpace benötigt:

- Wurde ein User erkannt?
- Wie viele User sind erkannt?
- Welche Geste wurde erkannt?
- Wie sicher wurde die Geste erkannt?
- Welcher User hat die Geste ausgeführt?

Um diese Informationen zu erhalten kann das **OSC Protokoll** verwendet werden.

> *Open Sound Control (OSC) ist ein nachrichtenbasiertes Kommunikationsprotokoll, welches hauptsächlich für die Echtzeitverarbeitung von Sound über Netze und Multimedia-Installationen verwendet wird.*
>
> *Steuersignale können von Hardware (z. B. MIDI-Keyboard) oder Software (z. B. Processing, Vvvv, Csound, Max/MSP, Pure Data, SuperCollider, ChucK, EyesWeb) erzeugt und dann via OSC in Form von sog. Nachrichten (OSC-Messages), welche wiederum in Bündel (OSC-Bundles) verpackt werden, an eine Schnittstelle weitergegeben werden und so eine Ausgabe steuern. Dieses können z. B. weitere Soundausgaben sein, etwa eine Soundanwendung auf einem anderen Computer.*

> *OSC ist unabhängig vom Transportprotokoll, wobei in der Regel jedoch UDP verwendet wird. Je nach Anforderung ist es aber auch möglich, OSC etwa über TCP oder eine serielle Schnittstelle zu transportieren*
>
> *[http://de.wikipedia.org/wiki/Open_Sound_Control].*

Da eine endgültige funktionsfähige Version von KS erst am Tag der Abschlusspräsentation zur Verfügung stand konnte das Auslesen von OSC-Messages nicht realisiert werden. Allerdings ist **Processing** mit der erforderlichen **Bibliothek oscP5** bereits installiert.

KineticSpace kann über die Konsole **gestartet** werden. Im Home Verzeichnis (hier: Persönlicher Ordner) befindet sich ein Bash File, welches im Terminal mit **./kineticSpace_Bash.sh** ausgeführt werden kann. Wenn die Kinect angeschlossen ist und erkannt wird erscheint folgender Bildschirm.

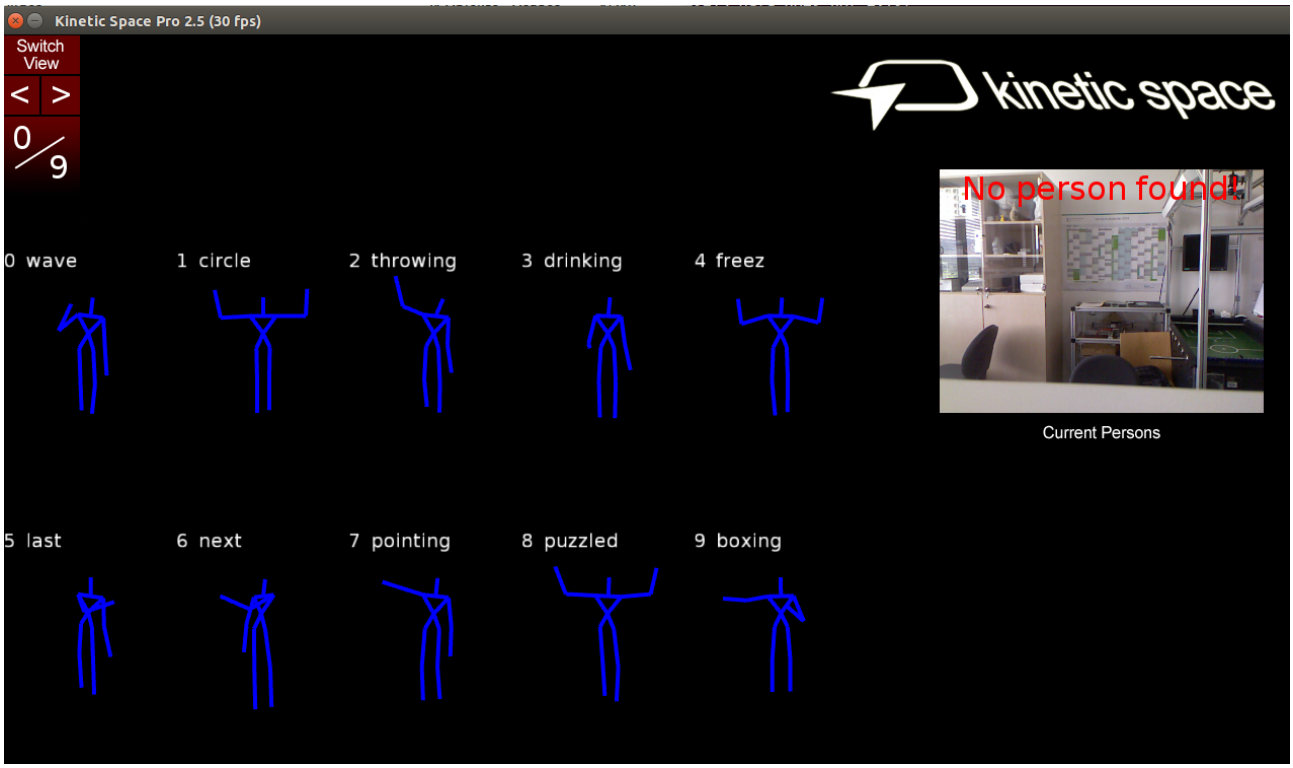

Abbildung 15 KineticSpace Startbildschirm

Für eine Erkennung reicht es sich vor der Kinect beliebig zu bewegen. Es ist keine „Kalibrier-Pose“ notwendig. Wurde eine Person gefunden beginnt das Tracking und das entsprechende Skeleton wird im Fenster oben rechts dargestellt (vgl. Abbildung 16). Die in Abbildung 15 sichtbaren Posen, sind die bereits eingelernten. Versucht man nun eine dieser Posen nachzumachen, z.B. die „Freez“ Pose, erscheint ein roter (User 1) Balken der angibt wie sicher die aktuelle Geste mit der trainierten Version übereinstimmt. Je länger der Balken, desto sicherer. Bei 100% wird er Balken dann grün. Es können maximal zwei Personen gleichzeitig getrackt werden. Es werden dann zwei verschieden farbige Balken unter den Gesten angezeigt. Über den „Switch View“ Button oben links im Fenster gelangt

man zu einer weiteren Ansicht, die in Abbildung 17 dargestellt ist – der Kinetic Lab Ansicht.

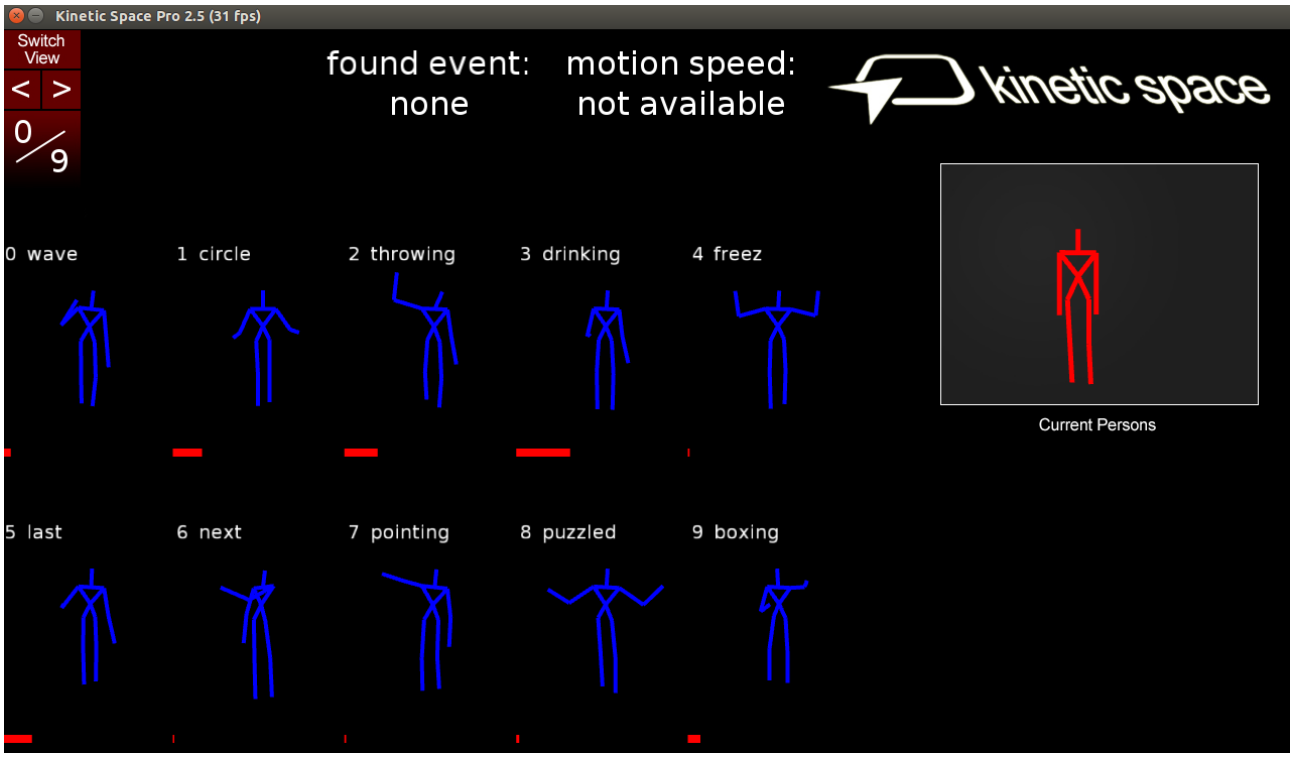

Abbildung 16 Person erkannt, Tracking ist jetzt aktiv

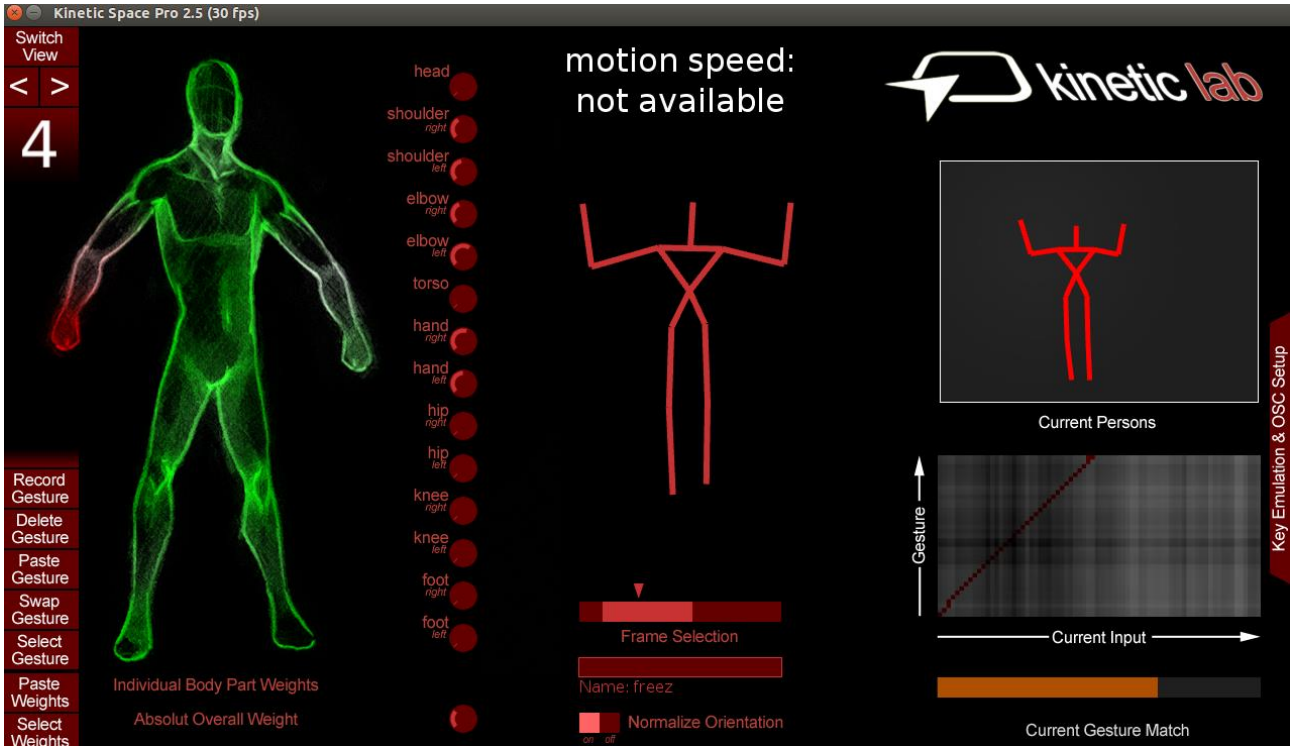

Abbildung 17 Kinetic Lab Ansicht

Diese Ansicht ermöglicht die Gewichtung sämtlicher Körperteile (vgl. Abbildung 18 Gewichtung einzelner Körperteile. Werden, wie z.B. bei der „Freez“ Pose der Fall die Beine nicht für eine Erkennung benötigt können diese entsprechend geringer gewichtet werden. Durch die Umfangreichen Einstellungen kann die Erkennung sehr gut optimiert werden. Befindet sich die getrackte Person am Bildrand bzw. hat den Bildausschnitt gerade verlassen zeigt ein Symbol im live View an in welche Richtung man sich bewegen muss um wieder innerhalb des Bildausschnittes zu sein, vgl.Abbildung 19.

Abbildung 19 Person befindet sich zu nah am Bildrand

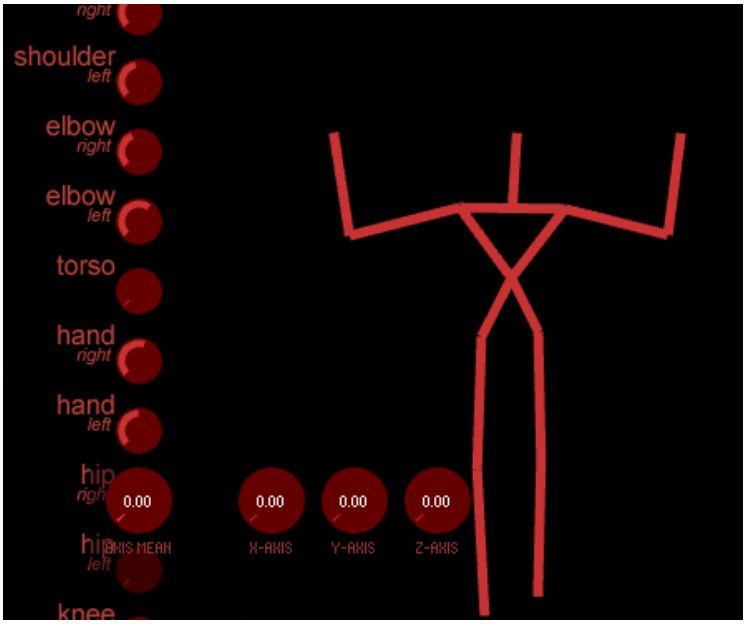

Abbildung 18 Gewichtung einzelner Körperteile

Alle Funktionen und Möglichkeiten können und sollen im user manual kinetic_space.pdf nachgelesen werden!

4 Further Work

- **Toolchain OpenNi**
  - Linker Fehler in Eclipse Projekt lösen
  - OpenNi Beispiel via Eclipse auf SCITOS implementieren
  - Kommunikation SCITOS und Kinect Daten
- **Toolchain KineticSpace**
  - OSC Protokoll implementieren um Events auszulesen (mit Processing Lib oscP5, bereits installiert!)
  - Installation auf SCITOS
  - Ansteuerung SCITOS durch Events von KineticSpace
- **Weitere Ideen**
  - Alternative zu Eclipse finden, z.B. QtCreator
  - Integration Wii Controller
  - Integration FPV

5 Fazit

Leider ließ sich unsere Wunschlösung über das OpenNI Framework in Kombination mit Eclipse nicht realisieren. Trotzdem bietet es unserer Meinung nach die beste Option wenn Flexibilität und eigene Entwicklungen mit der Kinect im Vordergrund stehen. Die Behebung der Linkerfehler selbstverständlich vorausgesetzt!

Soll hingegen lediglich eine Interaktion mit dem SCITOS, basierend auf Gesten die durch die Kinect erkannt werden realisiert werden, stellt das Tool KineticSpace eine sehr einfach einzubindende und anschauliche Lösung dar. Leider war es aus zeitlichen Gründen nicht mehr möglich die Übermittlung von Informationen über das OSC Protokoll zu realisieren. Bisher existiert auch noch keine Dokumentation wie das genau angestellt werden kann. Unserer Einschätzung nach sollte dies aber trotzdem machbar sein. Sollten dennoch weiter Fragen auftauchen kann auf den direkten Kontakt mit dem Entwickler Matthias Wölfel von der HS Pforzheim zurückgegriffen werden. Er war sehr hilfreich und hat unsere Mails immer zügig beantwortet.

6 Installationsanleitung

Es wird nachfolgend beschrieben, wie die beiden zuvor vorgestellten Varianten installiert werden. Die Verwendung von KineticSpace setzt ebenfalls die Installation von OpenNI voraus. Im Abschnitt 6.3 sind zusätzlich nochmals sämtliche Links anhand derer die Installation durchgeführt wurde enthalten.

6.1 OpenNI, NITE, Kinect Driver

Die Installation von OpenNI, NITE und einem Treiber für die Kinect ermöglicht es diese unter Linux verwenden zu können.

Ob die Kinect vom System erkannt wird kann am einfachsten über die Eingabe des Befehls **$lsusb** im Terminal überprüft werden. Eine positive Ausgabe sieht dann in etwa so aus:

$ Bus 003 Geräte 007: ID 045e: 02ad Microsoft Corp Xbox NUI Audio

$ Bus 003 Geräte 004: ID 045e: 02B0 Microsoft Corp Xbox NUI Motor

$ Bus 003 Geräte 008: ID 045e: 02ae Microsoft Corp Xbox NUI Kamera

Vor der Installation -> Kinect ausstecken!

Herunterladen von erforderlichen Dateien

Folgendes Package muss vor der Installation heruntergeladen werden:

OpenNI_NITE_Installer

[https://code.google.com/p/simple-openni/downloads/detail?name=OpenNI_NITE_Installer-Linux64-0.27.zip&can=3&q=]

Das Package enthält die folgenden Files:

1. NITE-Bin-Dev-Linux-x64-**v1.5**.2.21 die NITE Middleware für Skeleton-Tracking
2. OpenNI-Bin-Dev-Linux-x64-**v1.5**.4.0 die OpenNI SDK für Kinect
3. Sensor-Bin-Linux-x64-v5.1.2.1 Kinect Treiber um HW nutzbar zu machen
4. Kinect wird **nicht benötigt**!

Davon werden drei Files benötigt. File Nr.1 und Nr.2 von oben. Anstelle von Nr.3 wird das Sensor Package von hier heruntergeladen [https://github.com/avin2/SensorKinect]. Jetzt muss ein Ordner mit dem Namen **kinect** im Home Verzeichnis (Persönlicher Schreibtisch) erstellt werden. Anschließend werden die drei Ordner von oben umbenannt (vgl. auch Abbildung Abbildung 20).

1. nite
2. openni
3. sensorkin

Abbildung 20 Ordnerstruktur für OpenNI Installation

Installation

Als nächstes müssen alle benötigten Bibliotheken und openJDK mit folgenden Kommandozeilen installiert werden:

$ sudo apt-get install libusb-1.0-0-dev freeglut3-dev g++

$ sudo apt-get install openjdk-7-jdk openjdk-7-jre

Jetzt können die drei Files die zuvor in den Ordner **kinect** kopiert wurden installiert werden [3],[4],[5].

OpenNi Installation

$ cd ~ / kinect / OpenNI /
$ chmod 777 install.sh
$ sudo. / Install.sh

Sensorkin Installation

$ cd ~ / kinect / sensorkin / Plattform / Linux / CreateRedist /
$ chmod 777 RedistMaker
$ sudo. / RedistMaker
$ cd .. / Redist/Sensor-Bin-Linux-x64-v5.1.2.1
$ sudo chmod 777 install.sh
$ sudo. / Install.sh

NITE Installation

$ cd ~/kinect/nite/
$ chmod 7777 install.sh
$ sudo ./install.sh

Anpassen der USB-Rules

Erstellen einer Datei mit den Regeln für den Linux-Gerätemanager. Auch für die spätere KineticSpace Installation notwendig! Diese Datei befindet sich auch in der Dokumentation im Ordner **openni**.

$ sudo nano / etc/udev/rules.d/51-kinect.rules

Kopieren und folgenden Text einfügen:

```
# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02b0", MODE="0666"

# ATTR{product}=="Xbox NUI Audio"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ad", MODE="0666"

# ATTR{product}=="Xbox NUI Camera"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ae", MODE="0666"

# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02c2", MODE="0666"

# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02be", MODE="0666"

# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02bf", MODE="0666"
```

Speichern der Datei mit Strg+X.

Wird hier auf die bereits bestehende Datei **51-kinect.rules** aus der Dokumentation zurückgegriffen muss sichergestellt werden, dass diese in den richtigen Ordner kopiert wird.

etc/udev/rules.d

Test der OpenNI Installation

Kinect mit PC verbinden und folgende Datei ausführen:
/Persönlicher Ordner/kinect/openni/Samples/Bin/x64-Release/NiViewer

Es sollte ein RGB und ein Tiefen Bild zu sehen sein, vgl. Abbildung 21.

Abbildung 21 OpenNI Test - NiViewer

Wahlweise können auch andere Beispiele ausgeführt werden (auch aus dem Samples Ordner von NITE)

Fehler

Sollte ein Fehler auftreten der da heißt:

> *Open failed: failed to set USB interface.*

Kann entweder ein Neustart durchgeführt werden, oder wenn das nicht hilft kann noch folgender Befehl im Terminal ausgeführt werden:

$ rmmod gspca_kinect

(Eventuell muss dieser Befehl vor jeder Verwendung der Kinect ausgeführt werden. War aber zuletzt nicht der Fall.)

6.2 Installation KineticSpace

Um KineticSpace unter Linux zu installieren bzw. auszuführen müssen folgende Komponenten installiert werden. KineticSpace selbst kann in einem geeigneten Ordner gespeichert werden. Wir empfehlen die Version **kineticspace_pro_ver25.linux64**. Es sind wei-

tere Versionen vorhanden, welche teilweise nicht funktionieren oder noch nicht getestet wurden, vgl. Abbildung 22.

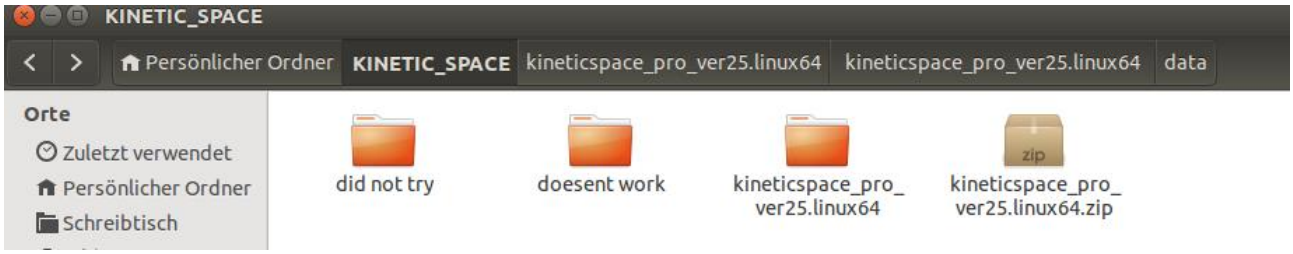

Abbildung 22 Ordnerstruktur KineticSpace

OpenNI 1.X und **NITE 1.X** -> siehe Kapitel 6.1

Zusätzlich muss **Processing** [6] heruntergeladen werden [http://processing.org]. Es kann nach dem Download an einem beliebigen Ort gespeichert und von dort aus direkt ausgeführt werden. Es ist keine Installation erforderlich.

Außerdem sind die drei folgenden **Bibliotheken** notwendig:

1. SimpleOpenNI - ein einfache OpenNI und NITE Wrapper für Processing [7]. http://code.google.com/p/simple-openni
2. oscP5 - Implementierung von OSC Protocol für Processing [8]. http://www.sojamo.de/libraries/oscP5/index.html
3. fullscreen - Vollbild Unterstützung für Processing [9]. http://www.superduper.org/processing/

Processing legt im Verzeichnis *Persönlicher Ordner* einen Ordner Namens *sketchbook/libraries* an. In diesen müssen die zuvor heruntergeladenen Bibliotheken hinein kopiert werden. Die Einbindung kann auch von Processing über *Add Libraries* erfolgen.

Auch hier ist es notwendig die Datei **51-kinect.rules** zu erzeugen bzw. zu verwenden (befindet sich im Ordner *openni*).

Starten von KineticSpace erfolgt über die Konsole. Dazu in den entsprechenden Ordner navigieren und die entsprechende Datei ausführen, hier:

```
cd ~/KINETIC_SPACE/kineticspace_pro_ver25.linux64/kineticspace_pro_ver25.linux64
./kineticspace_pro_ver25
```

Bei erfolgreicher Installation sollte jetzt das Tool starten und der Startbildschirm wie bereits in Abbildung 15 dargestellt erscheinen.

6.3 Installations Quellen

[1] OpenNI NITE Installer

URL:https://code.google.com/p/simple-openni/downloads/detail?name=OpenNI_NITE_Installer-Linux64-0.27.zip&can=3&q=

[2] Kinect Sensor Paket

URL: https://github.com/avin2/SensorKinect

[3] How To Install Kinect in Linux

URL: http://igorbarbosa.com/articles/how-to-install-kin-in-linux-mint-12-ubuntu/

[4] Kinect on Ubuntu with OpenNI

URL: http://www.20papercups.net/programming/kinect-on-ubuntu-with-openni/

[5] Installation OpenNi on Ubuntu

URL:http://venturingtodance.blogspot.pt/2012/09/installating-openni-on-ubuntu-12.html

[6] Processing

URL: http://processing.org

[7] SimpleOpenNI

URL: http://code.google.com/p/simple-openni

[8] oscP5

URL: http://www.sojamo.de/libraries/oscP5/index.html

[9] fullscreen

URL: http://www.superduper.org/processing/

[10] Kinetic Space

URL: https://code.google.com/p/kineticspace/