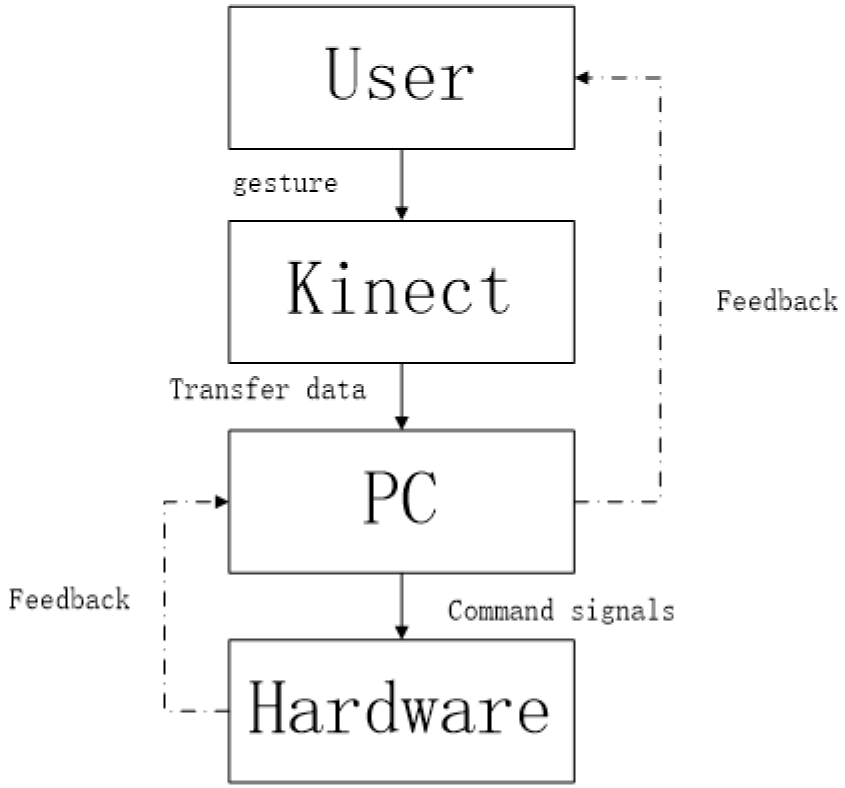
A Kinect-based HMI

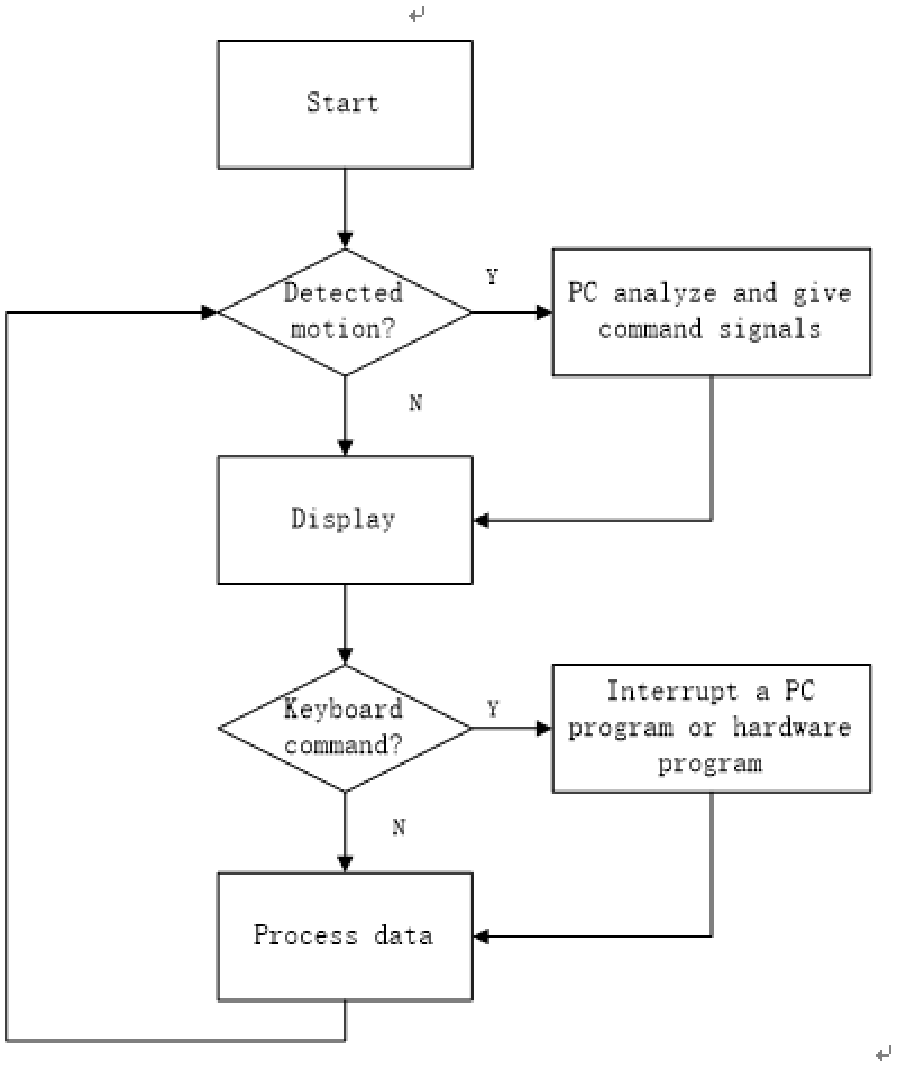
HMI: Human-Machine interface

A Kinect-based HMI

- How it works?
- What are the applications?
- How is it applied?
- What's the future?

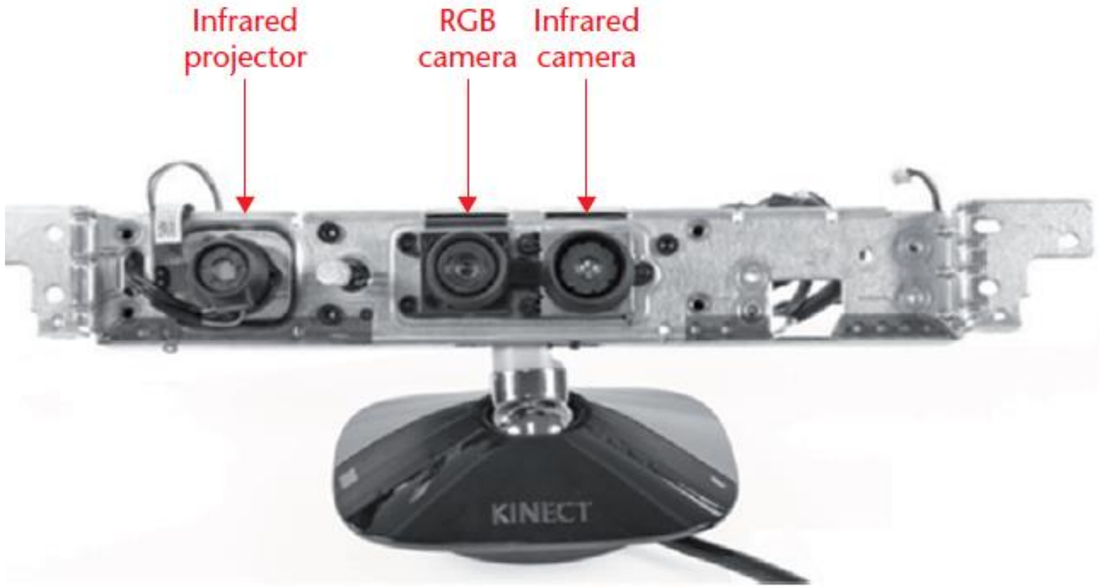
System overview




Start
Detected motion?
Y
PC analyze and give command signals
N
Display
Keyboard command?
Y
Interrupt a PC program or hardware program
N
Process data

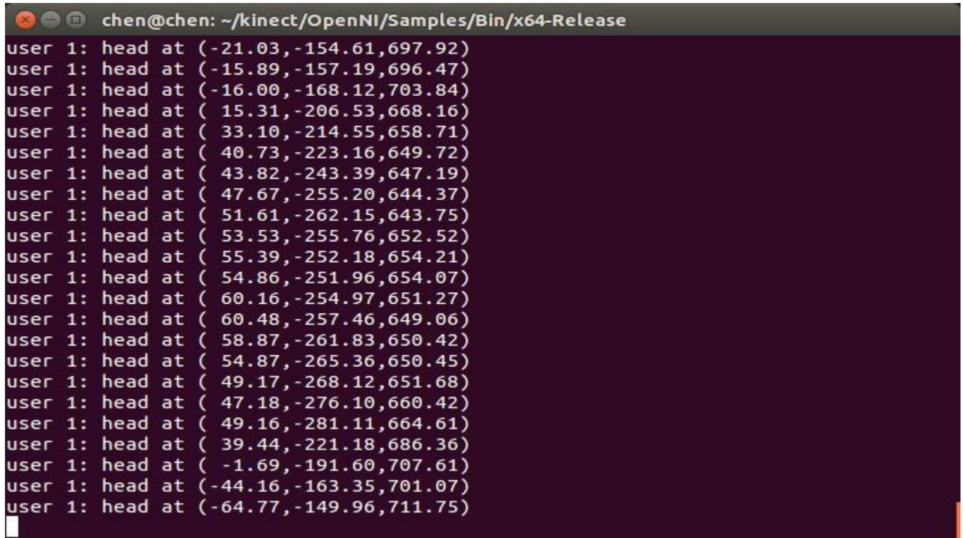
Kinect sensor

- A RGB-D Sensor
- Depth data of an object can be acquired
- Like a single human eye

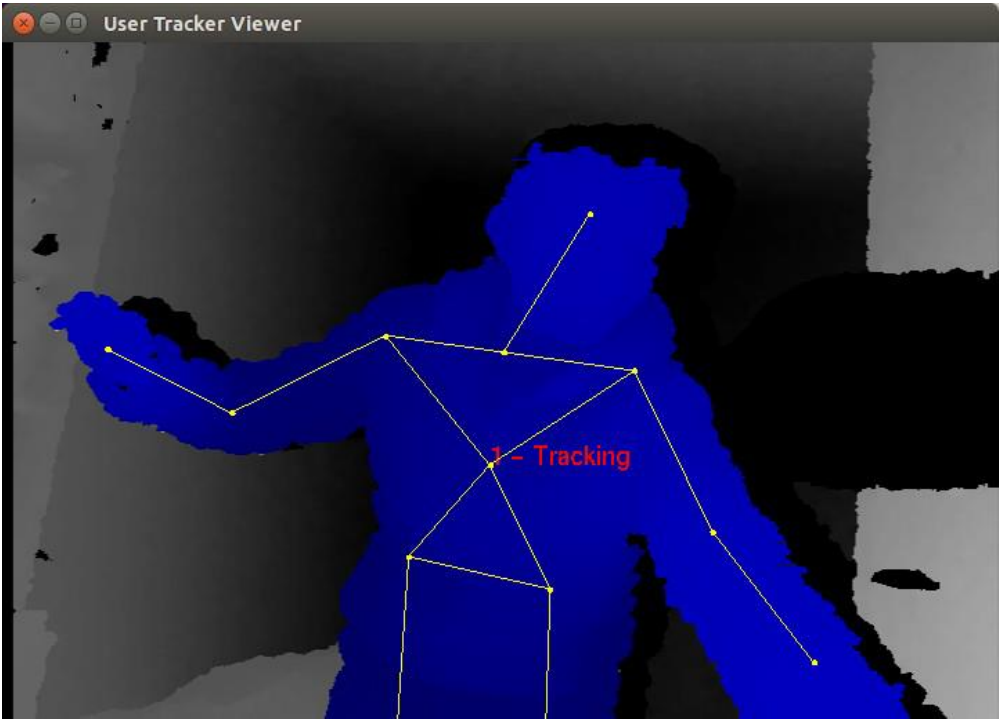


Ubuntu Linux 14.04 OS

- Portable, Free and APIs are defined by IEEE standards
- Long Term Support
- A rich selection of applications are available

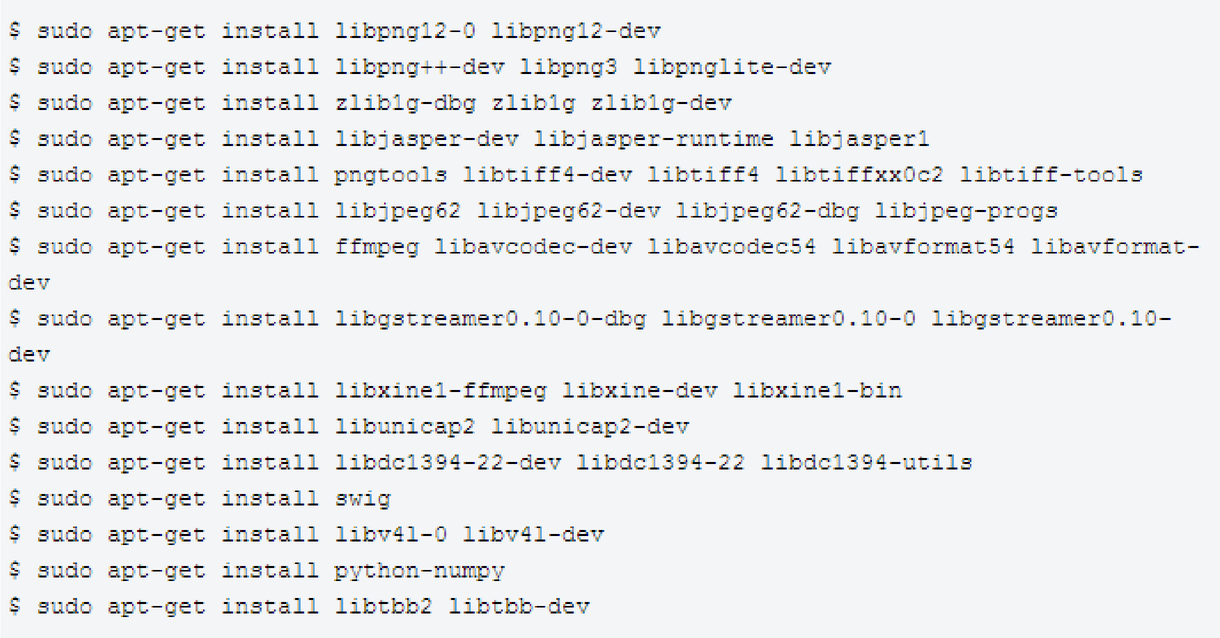


```
user 1: head at (-21.03,-154.61,697.92)
user 1: head at (-15.89,-157.19,696.47)
user 1: head at (-16.00,-168.12,703.84)
user 1: head at ( 15.31,-206.53,668.16)
user 1: head at ( 33.10,-214.55,658.71)
user 1: head at ( 40.73,-223.16,649.72)
user 1: head at ( 43.82,-243.39,647.19)
user 1: head at ( 47.67,-255.20,644.37)
user 1: head at ( 51.61,-262.15,643.75)
user 1: head at ( 53.53,-255.76,652.52)
user 1: head at ( 55.39,-252.18,654.21)
user 1: head at ( 54.86,-251.96,654.07)
user 1: head at ( 60.16,-254.97,651.27)
user 1: head at ( 60.48,-257.46,649.06)
user 1: head at ( 58.87,-261.83,650.42)
user 1: head at ( 54.87,-265.36,650.45)
user 1: head at ( 49.17,-268.12,651.68)
user 1: head at ( 47.18,-276.10,660.42)
user 1: head at ( 49.16,-281.11,664.61)
user 1: head at ( 39.44,-221.18,686.36)
user 1: head at ( -1.69,-191.60,707.61)
user 1: head at (-44.16,-163.35,701.07)
user 1: head at (-64.77,-149.96,711.75)
```

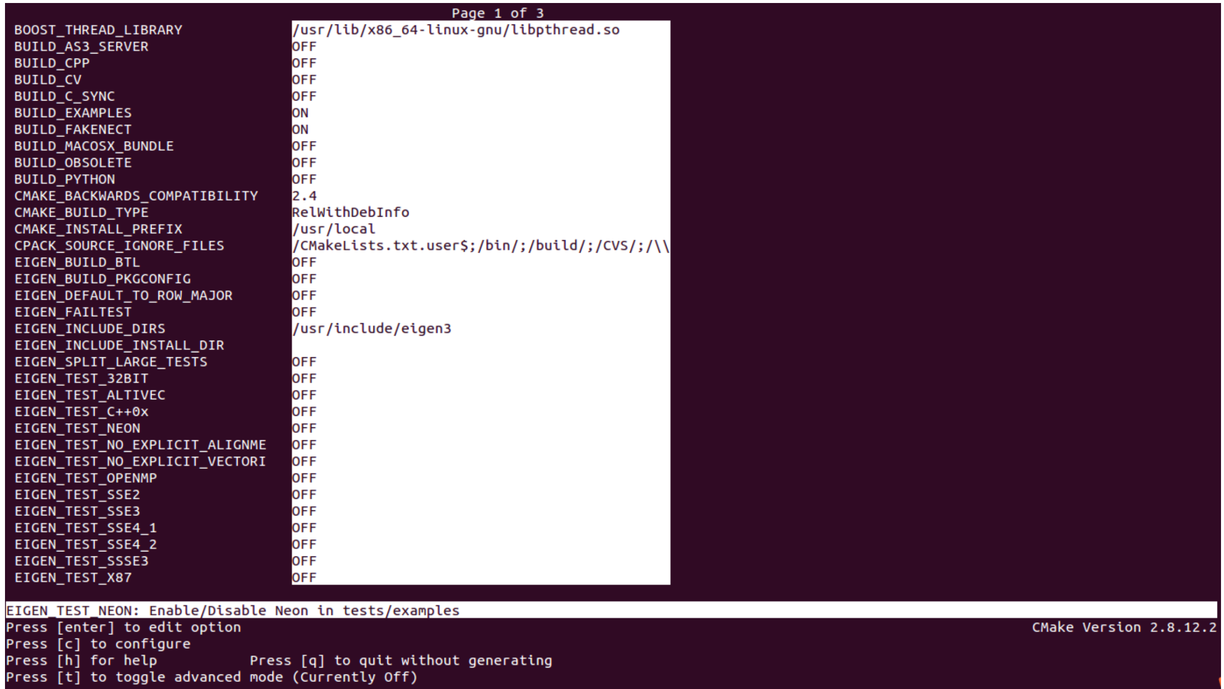


Implementation

- Free-Kinect Library
- Open NI 1.5.8 Library
- source code
- IDE & Compilers



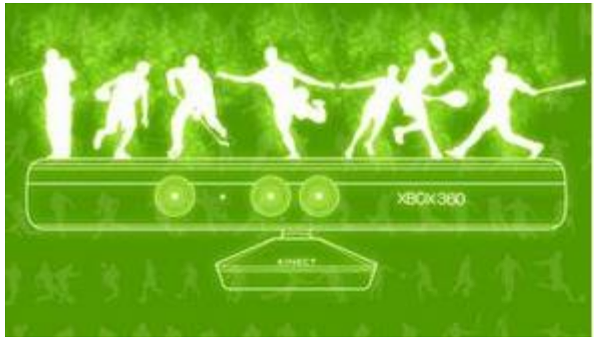
```
$ sudo apt-get install libpng12-0 libpng12-dev
$ sudo apt-get install libpng++-dev libpng3 libpnglite-dev
$ sudo apt-get install zlib1g-dbg zlib1g zlib1g-dev
$ sudo apt-get install libjasper-dev libjasper-runtime libjasper1
$ sudo apt-get install pngtools libtiff4-dev libtiff4 libtiffxx0c2 libtiff-tools
$ sudo apt-get install libjpeg62 libjpeg62-dev libjpeg62-dbg libjpeg-progs
$ sudo apt-get install ffmpeg libavcodec-dev libavcodec54 libavformat54 libavformat-
dev
$ sudo apt-get install libgstreamer0.10-0-dbg libgstreamer0.10-0 libgstreamer0.10-
dev
$ sudo apt-get install libxine1-ffmpeg libxine-dev libxine1-bin
$ sudo apt-get install libunicap2 libunicap2-dev
$ sudo apt-get install libdc1394-22-dev libdc1394-22 libdc1394-utils
$ sudo apt-get install swig
$ sudo apt-get install libv4l-0 libv4l-dev
$ sudo apt-get install python-numpy
$ sudo apt-get install libtbb2 libtbb-dev
```



```
                                  Page 1 of 3
 BOOST_THREAD_LIBRARY             /usr/lib/x86_64-linux-gnu/libpthread.so
 BUILD_AS3_SERVER                 OFF
 BUILD_CPP                        OFF
 BUILD_CV                         OFF
 BUILD_C_SYNC                     OFF
 BUILD_EXAMPLES                   ON
 BUILD_FAKENECT                   ON
 BUILD_MACOSX_BUNDLE              OFF
 BUILD_OBSOLETE                   OFF
 BUILD_PYTHON                     OFF
 CMAKE_BACKWARDS_COMPATIBILITY    2.4
 CMAKE_BUILD_TYPE                 RelWithDebInfo
 CMAKE_INSTALL_PREFIX             /usr/local
 CPACK_SOURCE_IGNORE_FILES        /CMakeLists.txt.user$;/bin/;/build/;/CVS/;/\\
 EIGEN_BUILD_BTL                  OFF
 EIGEN_BUILD_PKGCONFIG            OFF
 EIGEN_DEFAULT_TO_ROW_MAJOR       OFF
 EIGEN_FAILTEST                   OFF
 EIGEN_INCLUDE_DIRS               /usr/include/eigen3
 EIGEN_INCLUDE_INSTALL_DIR
 EIGEN_SPLIT_LARGE_TESTS          OFF
 EIGEN_TEST_32BIT                 OFF
 EIGEN_TEST_ALTIVEC               OFF
 EIGEN_TEST_C++0x                 OFF
 EIGEN_TEST_NEON                  OFF
 EIGEN_TEST_NO_EXPLICIT_ALIGNME   OFF
 EIGEN_TEST_NO_EXPLICIT_VECTORI   OFF
 EIGEN_TEST_OPENMP                OFF
 EIGEN_TEST_SSE2                  OFF
 EIGEN_TEST_SSE3                  OFF
 EIGEN_TEST_SSE4_1                OFF
 EIGEN_TEST_SSE4_2                OFF
 EIGEN_TEST_SSSE3                 OFF
 EIGEN_TEST_X87                   OFF

EIGEN_TEST_NEON: Enable/Disable Neon in tests/examples
Press [enter] to edit option                                        CMake Version 2.8.12.2
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

What are the applications?



1.Gaming

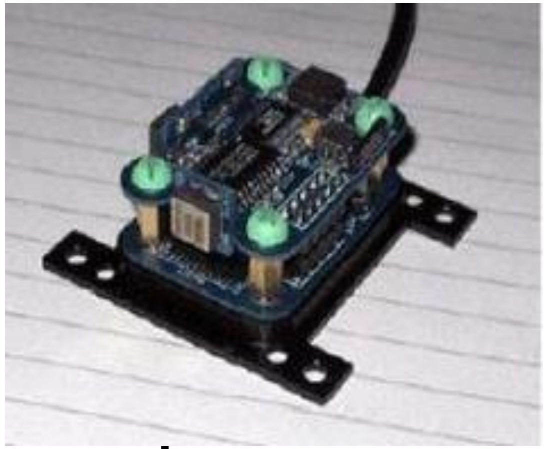
2.Gesture Recognition

3.Human Body Natural Interface

4.3D scanning

5.Motion data acquiring

How is it applied?

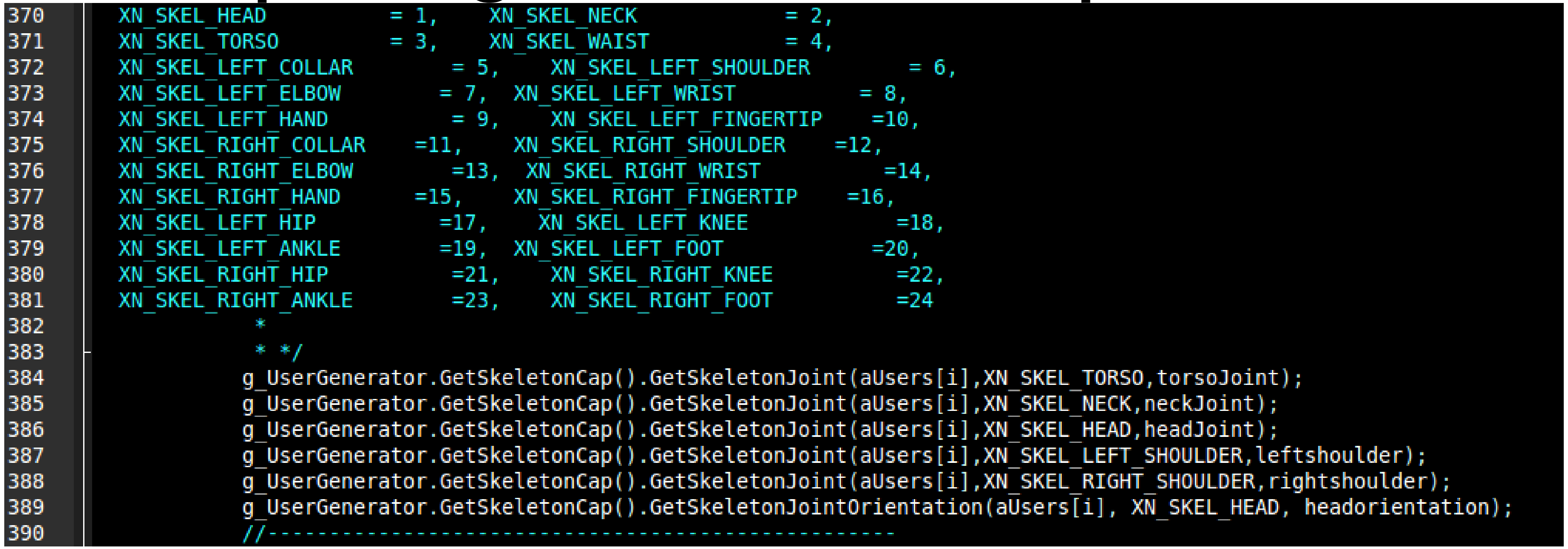


1. Keyboard
2. Sound Control/ IMU
3. Implementing Algorithms to Define a Gesture and To Send Signals

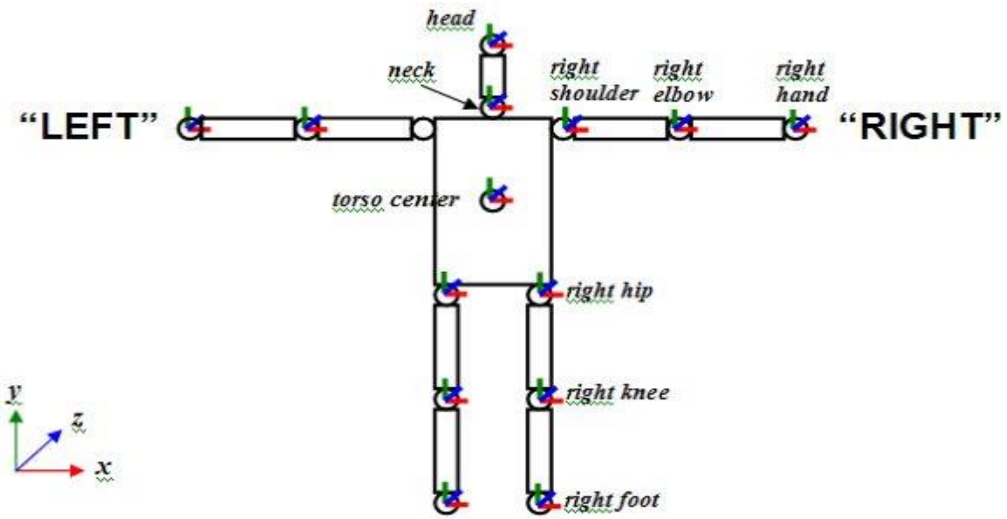
User Definition

- 1.Using Roll Pitch Raw of the head to control a three degree of freedom endoscope.
- 2.For surgery & medical scenario.
- 3.A doctor is controlling the whole system, not engineers.

Acquiring data from OpenNI



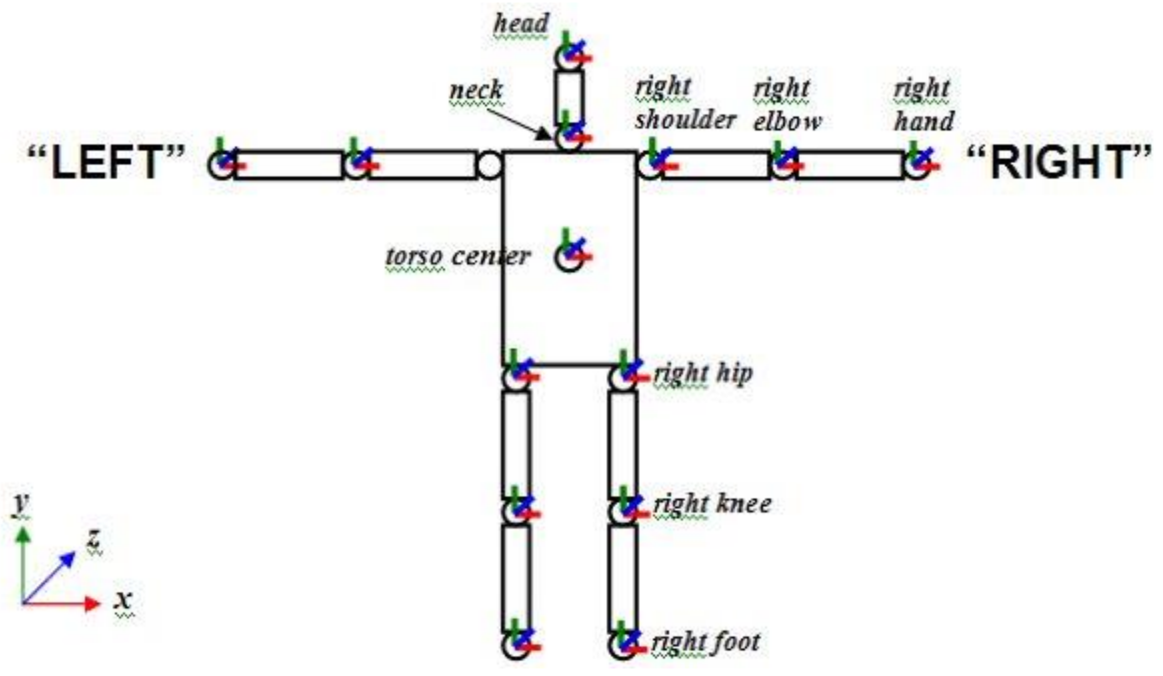
```
XN_SKEL_HEAD          = 1,    XN_SKEL_NECK            = 2,
XN_SKEL_TORSO         = 3,    XN_SKEL_WAIST           = 4,
XN_SKEL_LEFT_COLLAR        = 5,    XN_SKEL_LEFT_SHOULDER        = 6,
XN_SKEL_LEFT_ELBOW        = 7,  XN_SKEL_LEFT_WRIST          = 8,
XN_SKEL_LEFT_HAND          = 9,    XN_SKEL_LEFT_FINGERTIP    =10,
XN_SKEL_RIGHT_COLLAR    =11,    XN_SKEL_RIGHT_SHOULDER    =12,
XN_SKEL_RIGHT_ELBOW        =13,  XN_SKEL_RIGHT_WRIST          =14,
XN_SKEL_RIGHT_HAND      =15,    XN_SKEL_RIGHT_FINGERTIP    =16,
XN_SKEL_LEFT_HIP          =17,    XN_SKEL_LEFT_KNEE            =18,
XN_SKEL_LEFT_ANKLE        =19,  XN_SKEL_LEFT_FOOT            =20,
XN_SKEL_RIGHT_HIP          =21,    XN_SKEL_RIGHT_KNEE          =22,
XN_SKEL_RIGHT_ANKLE        =23,    XN_SKEL_RIGHT_FOOT          =24
           *
           * */
          g_UserGenerator.GetSkeletonCap().GetSkeletonJoint(aUsers[i],XN_SKEL_TORSO,torsoJoint);
          g_UserGenerator.GetSkeletonCap().GetSkeletonJoint(aUsers[i],XN_SKEL_NECK,neckJoint);
          g_UserGenerator.GetSkeletonCap().GetSkeletonJoint(aUsers[i],XN_SKEL_HEAD,headJoint);
          g_UserGenerator.GetSkeletonCap().GetSkeletonJoint(aUsers[i],XN_SKEL_LEFT_SHOULDER,leftshoulder);
          g_UserGenerator.GetSkeletonCap().GetSkeletonJoint(aUsers[i],XN_SKEL_RIGHT_SHOULDER,rightshoulder);
          g_UserGenerator.GetSkeletonCap().GetSkeletonJointOrientation(aUsers[i], XN_SKEL_HEAD, headorientation);
          //----------------------------------------------------
```



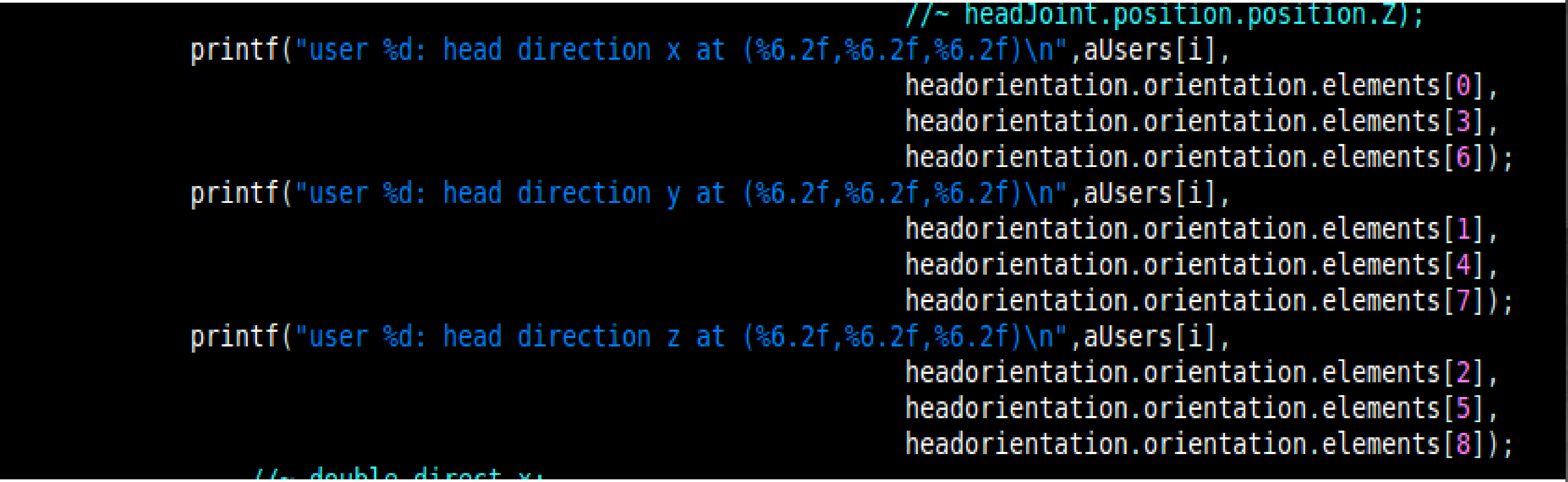
C/C++ & object orientated programing

- •Classes and Hierarchical Structure
- •With the documentation of a previous program, APIs can be used directly for development.

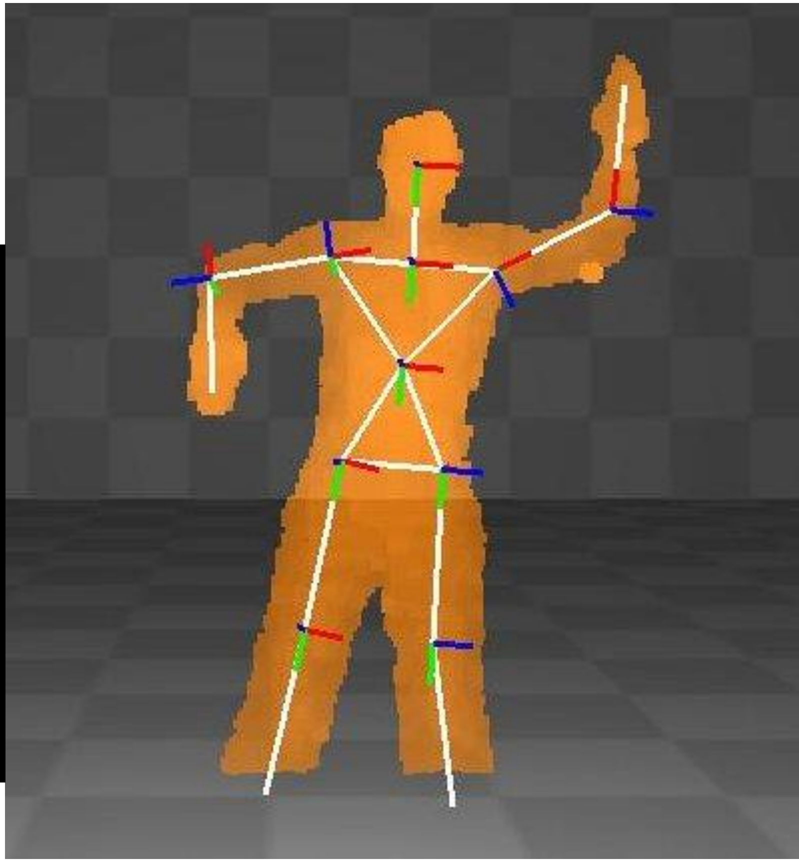
Machine - independent



OpenNI:



```
                                                             //~ headJoint.position.position.Z);
             printf("user %d: head direction x at (%6.2f,%6.2f,%6.2f)\n",aUsers[i],
                                                             headorientation.orientation.elements[0],
                                                             headorientation.orientation.elements[3],
                                                             headorientation.orientation.elements[6]);
             printf("user %d: head direction y at (%6.2f,%6.2f,%6.2f)\n",aUsers[i],
                                                             headorientation.orientation.elements[1],
                                                             headorientation.orientation.elements[4],
                                                             headorientation.orientation.elements[7]);
             printf("user %d: head direction z at (%6.2f,%6.2f,%6.2f)\n",aUsers[i],
                                                             headorientation.orientation.elements[2],
                                                             headorientation.orientation.elements[5],
                                                             headorientation.orientation.elements[8]);
                 //~ double direct_x;
```



Algorithm:(To increase the robustness)

- 0. Filters to get rid of the noises
- 1.Define a motion by its orientation
- 2.By its position
- 3.Creating threshold for comparison
- 4.Integration of a variable joint subtracting its threshold
- 5.Detecting the derivative of a joint's position
- 6.Creating vectors with other joints
- 7.Creating a logic tree of the above

Selected Elements From Transformation Matrix

$$Rot_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & \enclose{circle}{s_\phi} & c_\phi \end{pmatrix}$$

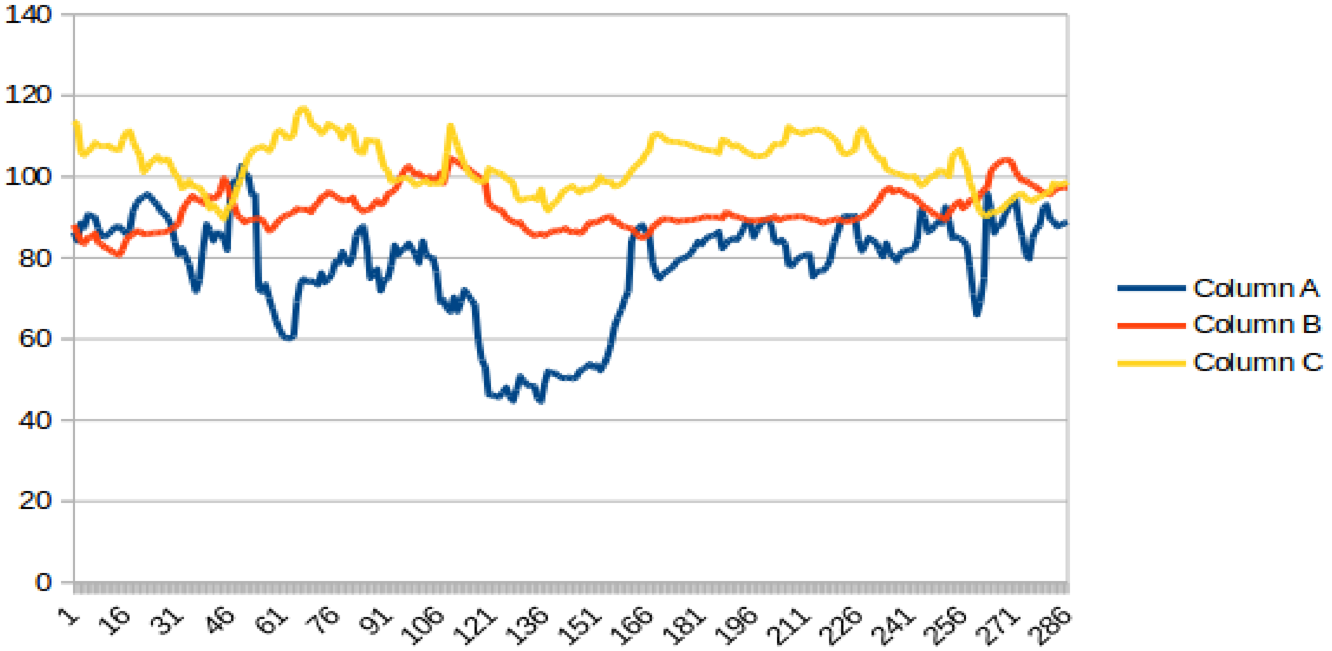
$$Rot_y(\theta) = \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix}$$

$$Rot_z(\varphi) = \begin{pmatrix} c_\varphi & -\enclose{circle}{s_\varphi} & 0 \\ \enclose{circle}{s_\varphi} & c_\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

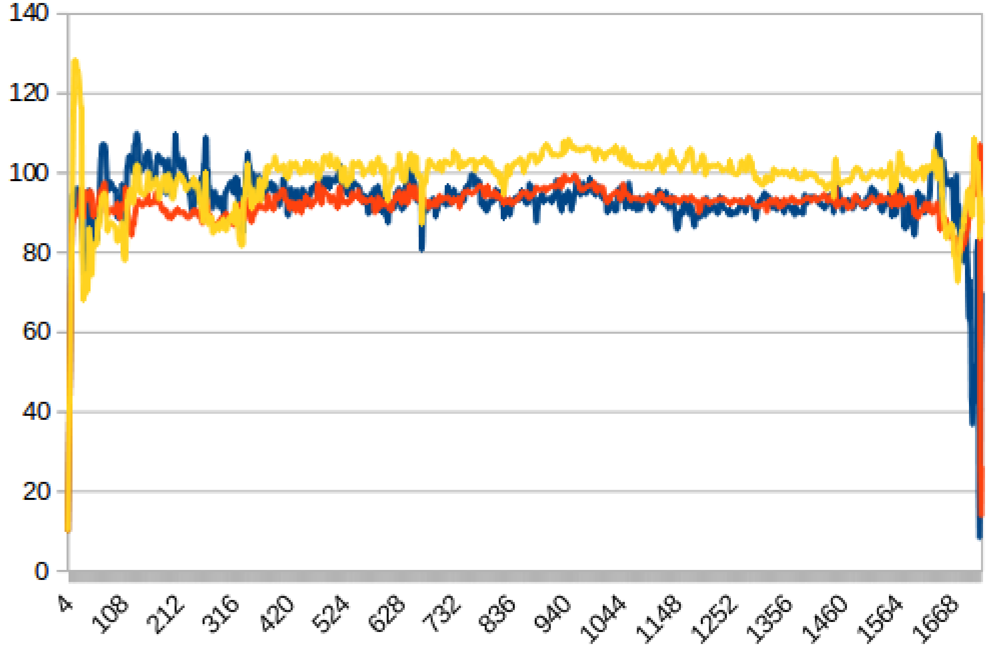
$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{p} \\ 0_{1\times 3} & 1 \end{pmatrix}$$

Filters

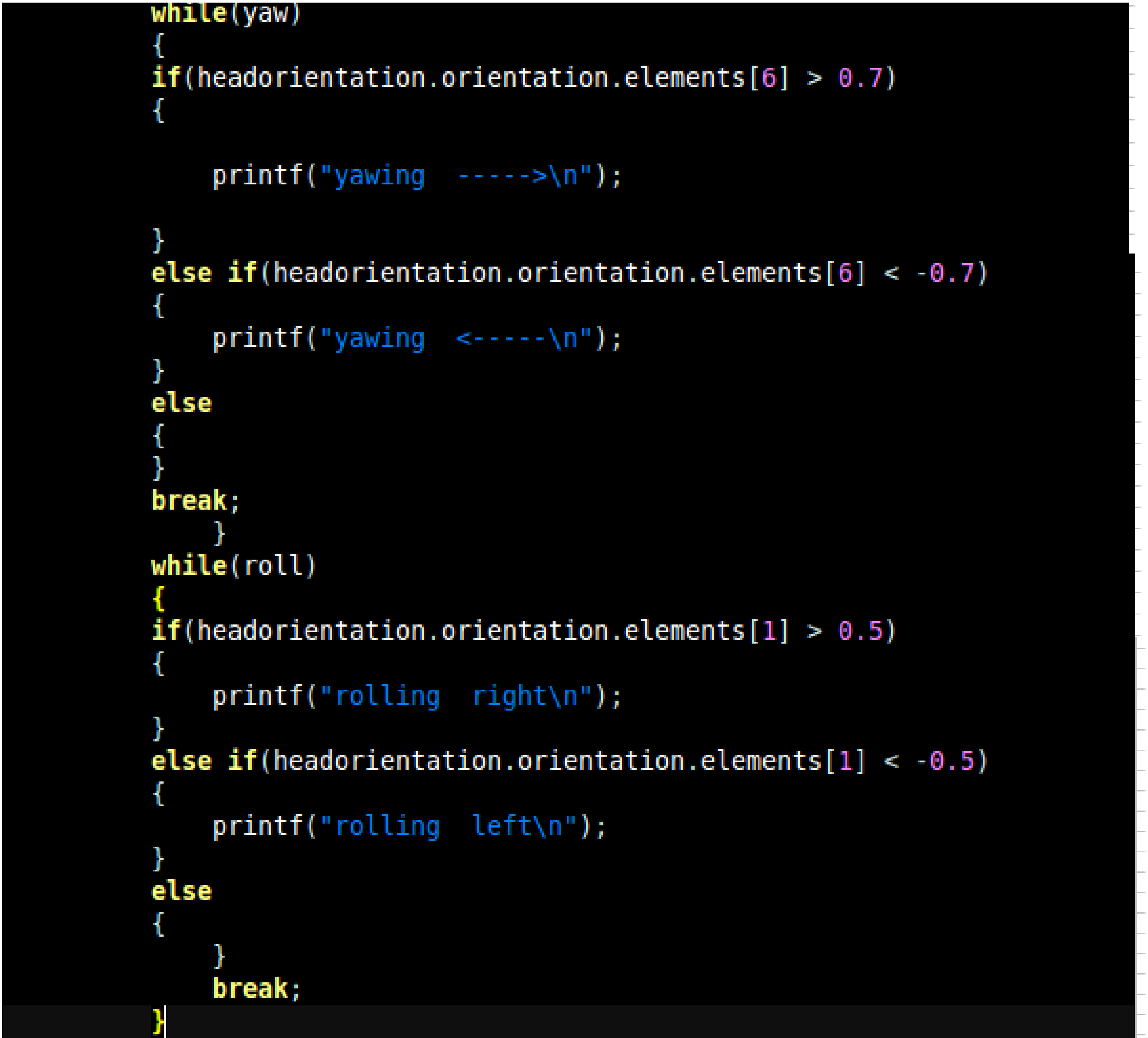
- Without Filters



- Filters Added
- (Drawback:
- Impulse)



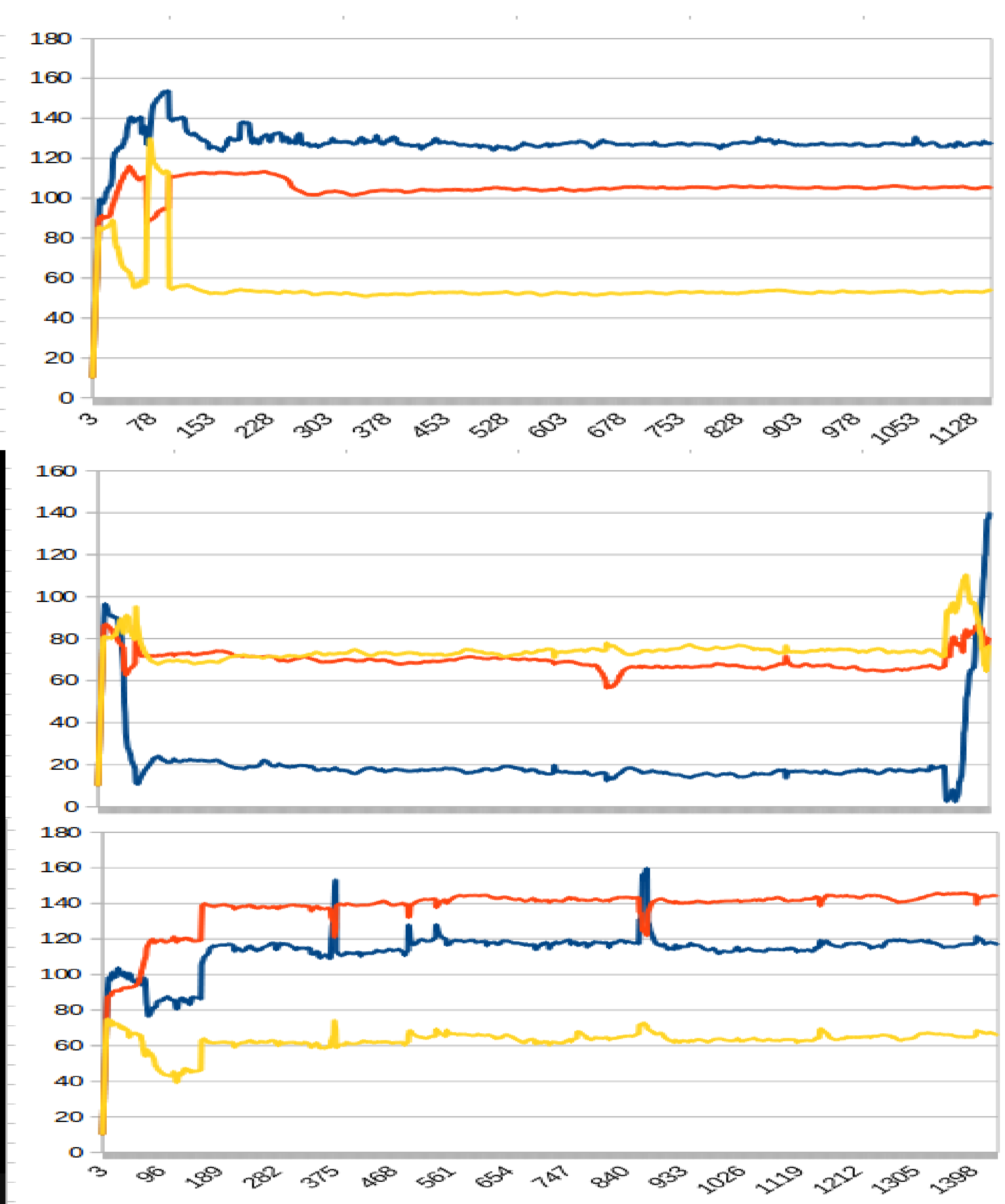
Algorithm: Orientation



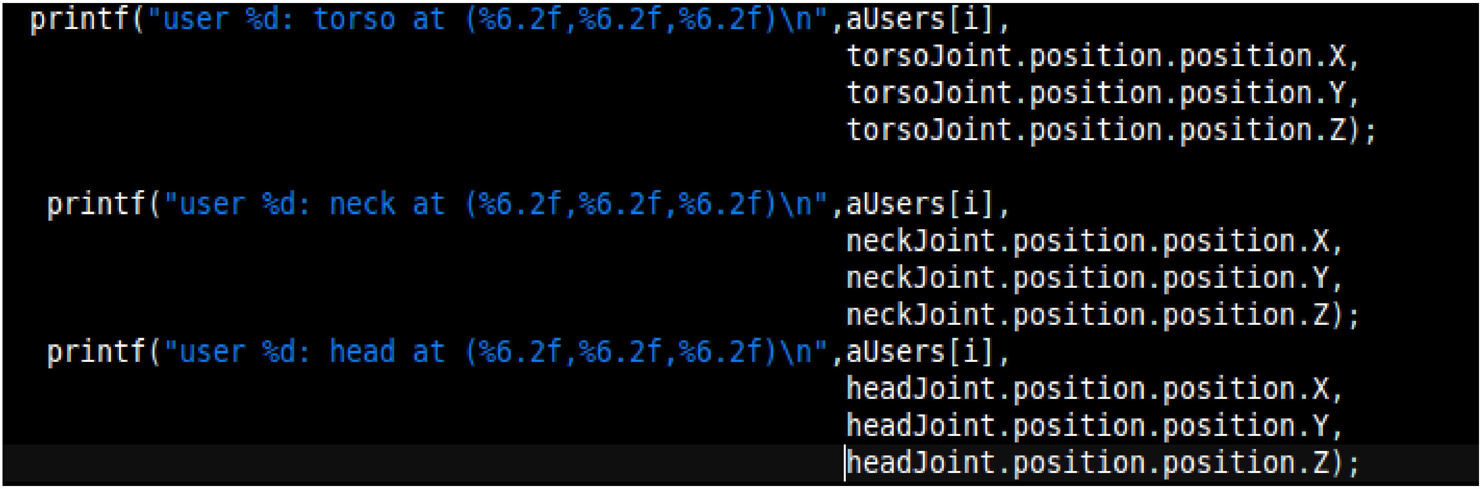
```
while(yaw)
{
if(headorientation.orientation.elements[6] > 0.7)
{

    printf("yawing  ----->\n");

}
else if(headorientation.orientation.elements[6] < -0.7)
{
    printf("yawing  <-----\n");
}
else
{
}
break;
    }
while(roll)
{
if(headorientation.orientation.elements[1] > 0.5)
{
    printf("rolling  right\n");
}
else if(headorientation.orientation.elements[1] < -0.5)
{
    printf("rolling  left\n");
}
else
{
    }
    break;
}
```

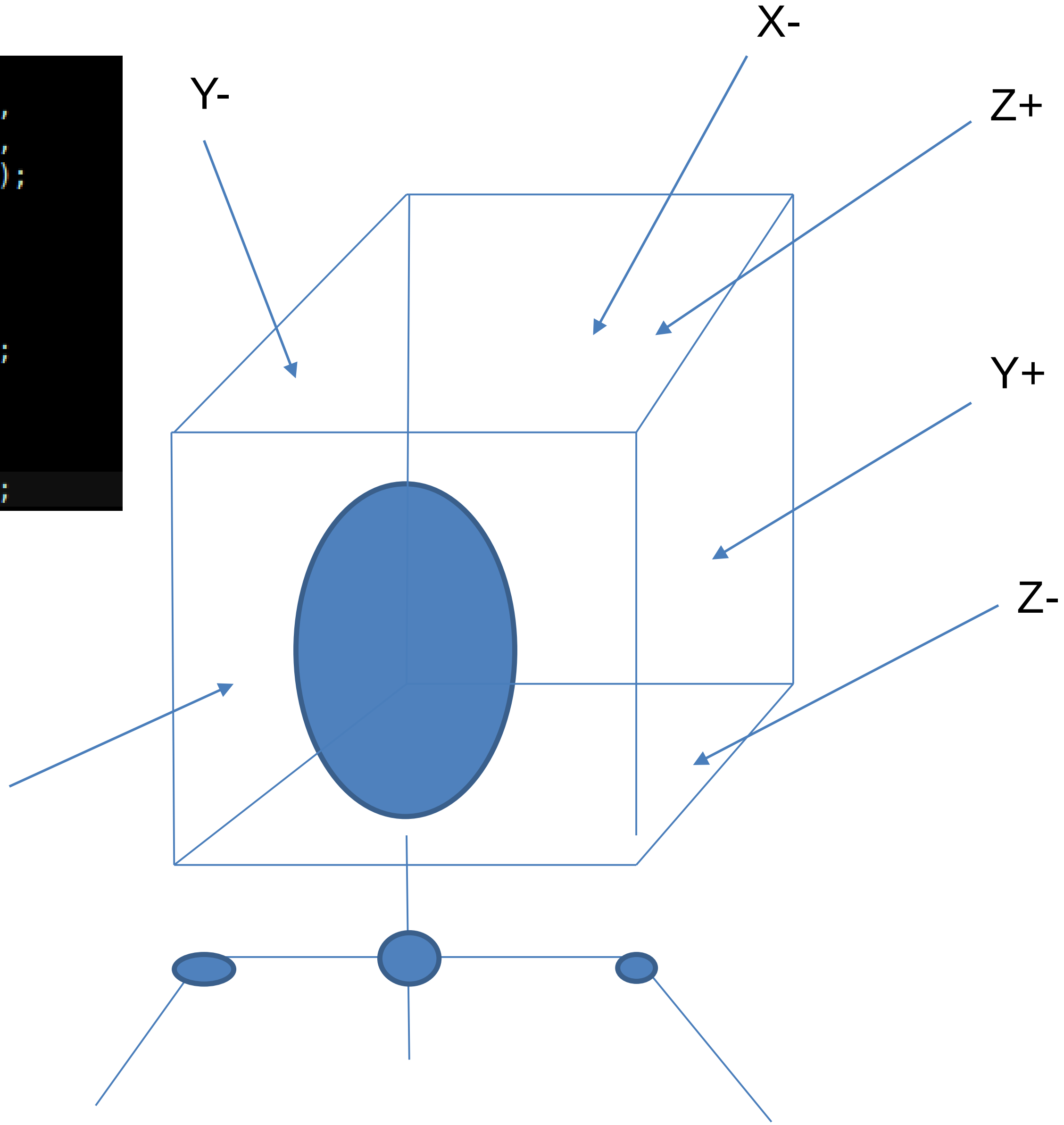


Position and Threshold

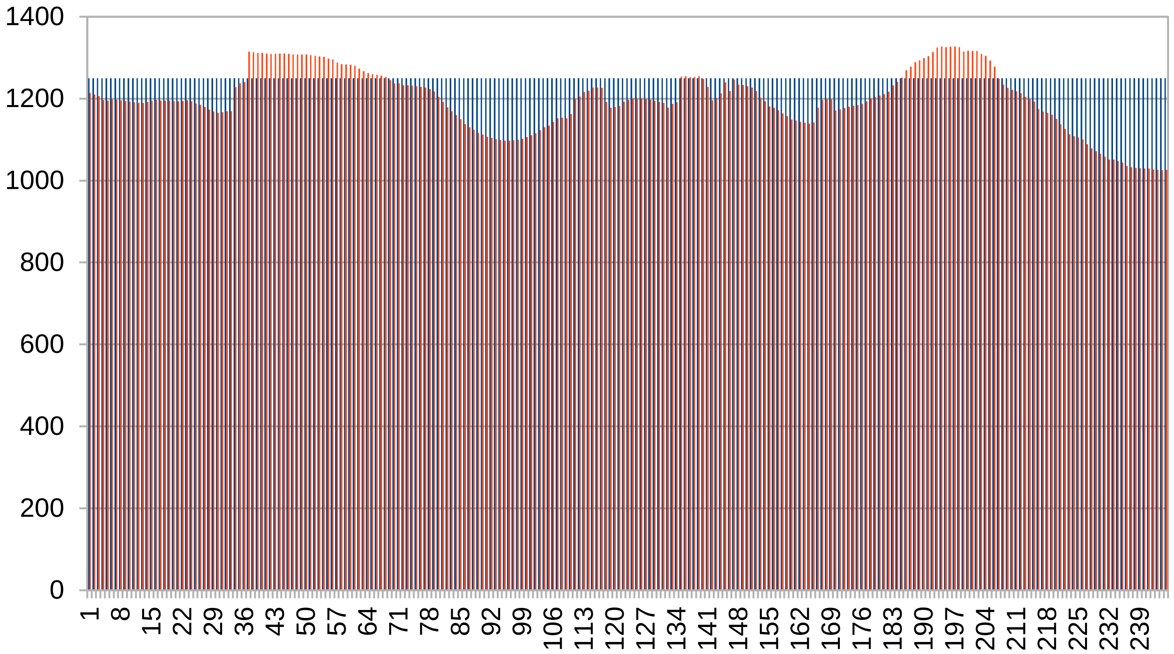


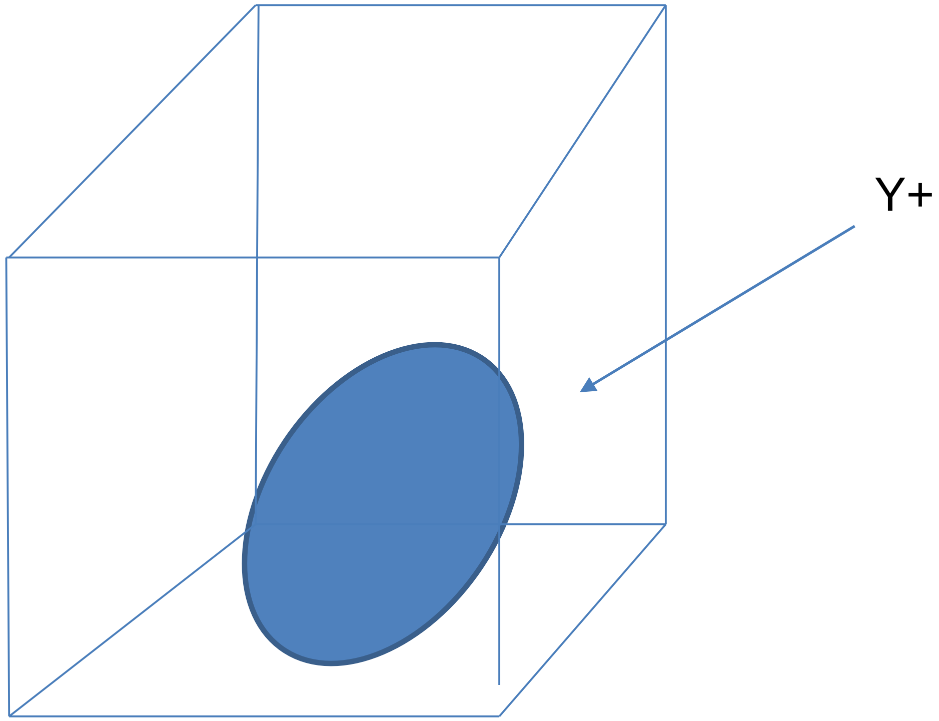
```
printf("user %d: torso at (%6.2f,%6.2f,%6.2f)\n",aUsers[i],
                                                  torsoJoint.position.position.X,
                                                  torsoJoint.position.position.Y,
                                                  torsoJoint.position.position.Z);

 printf("user %d: neck at (%6.2f,%6.2f,%6.2f)\n",aUsers[i],
                                                  neckJoint.position.position.X,
                                                  neckJoint.position.position.Y,
                                                  neckJoint.position.position.Z);
 printf("user %d: head at (%6.2f,%6.2f,%6.2f)\n",aUsers[i],
                                                  headJoint.position.position.X,
                                                  headJoint.position.position.Y,
                                                  headJoint.position.position.Z);
```

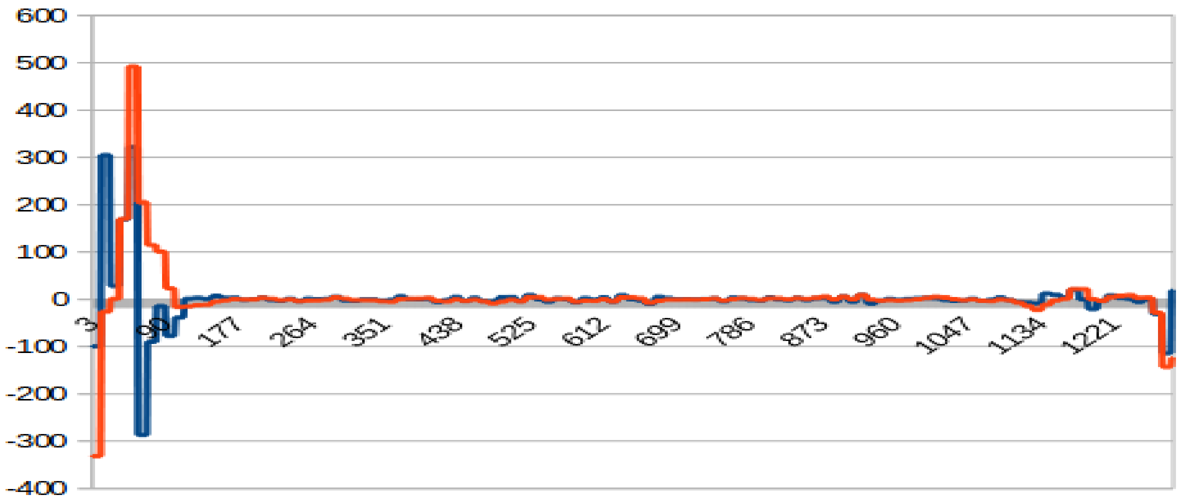


Integration of a variable

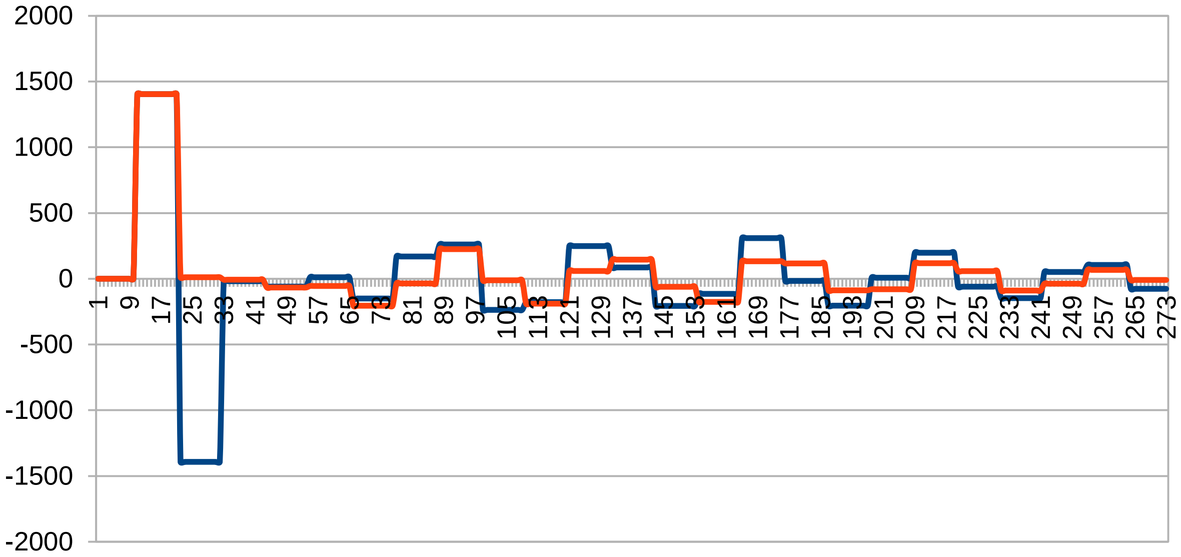




derivative of a joint's position

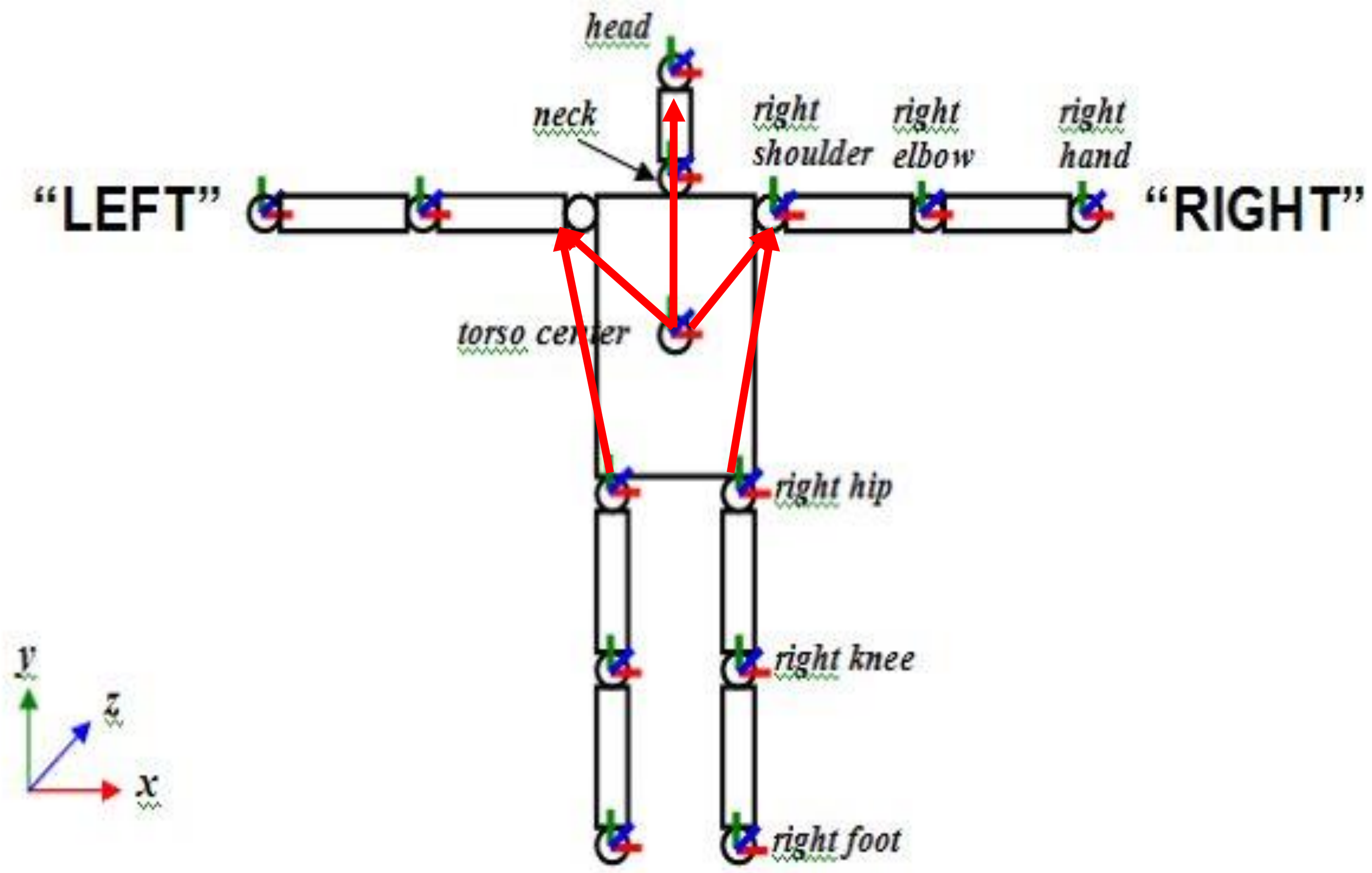


$$\frac{f(t_2) - f(t_1)}{t_2 - t_1}$$

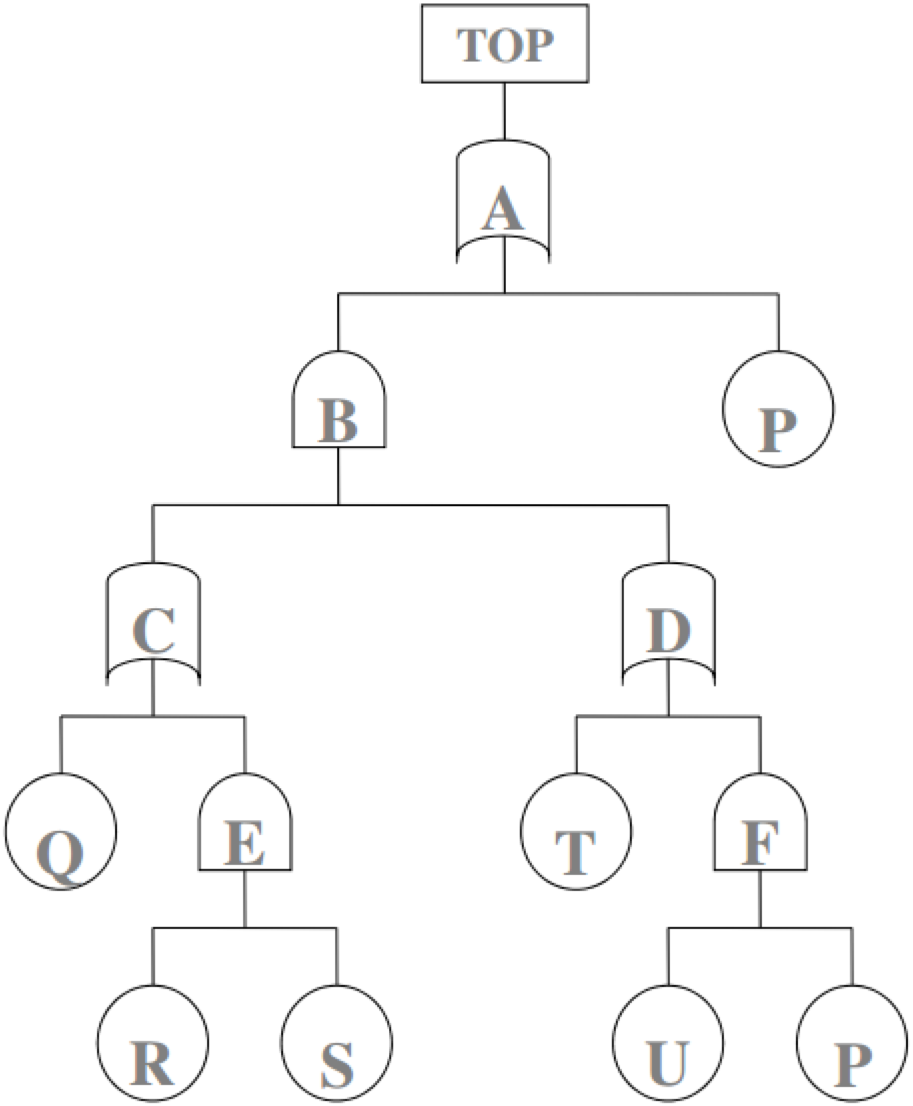


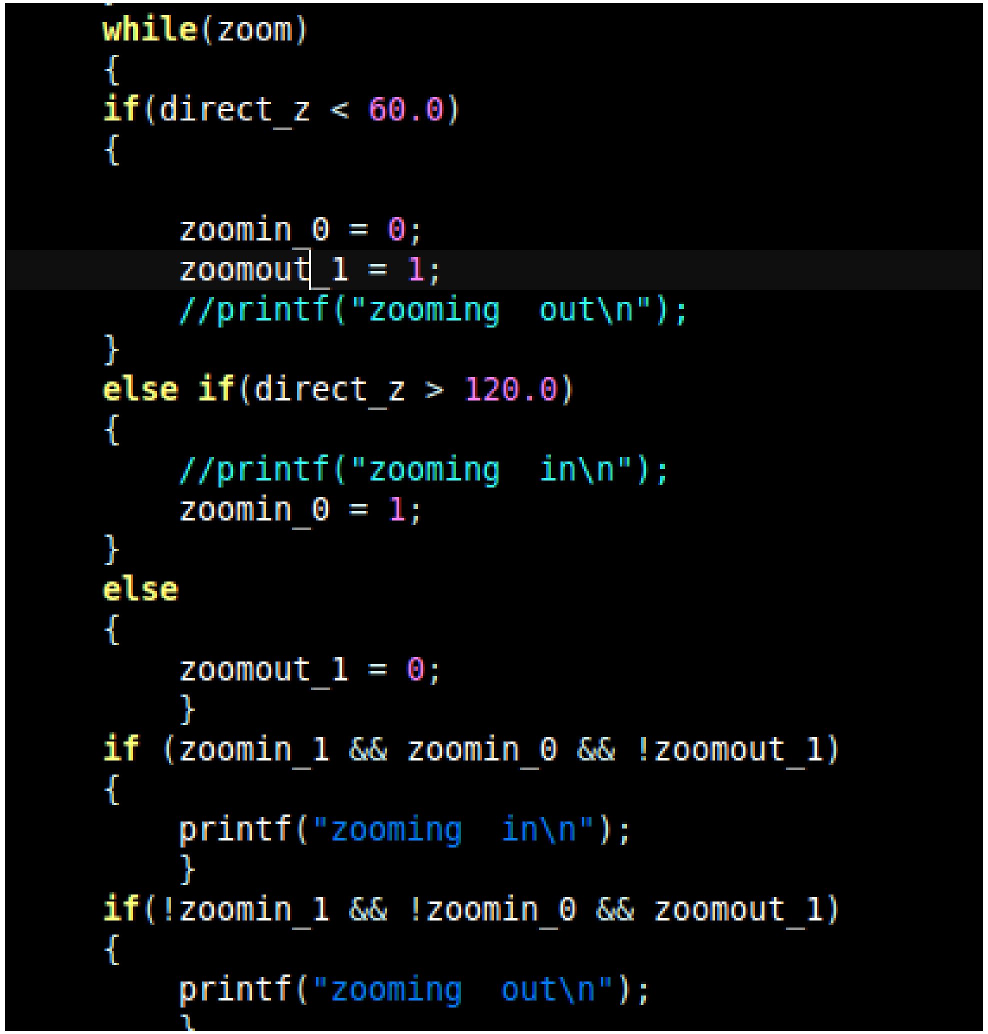
$$\frac{f'(t_2) - f'(t_1)}{t_2 - t_1}$$

Creating vectors with other joints



Creating a logic tree of the above





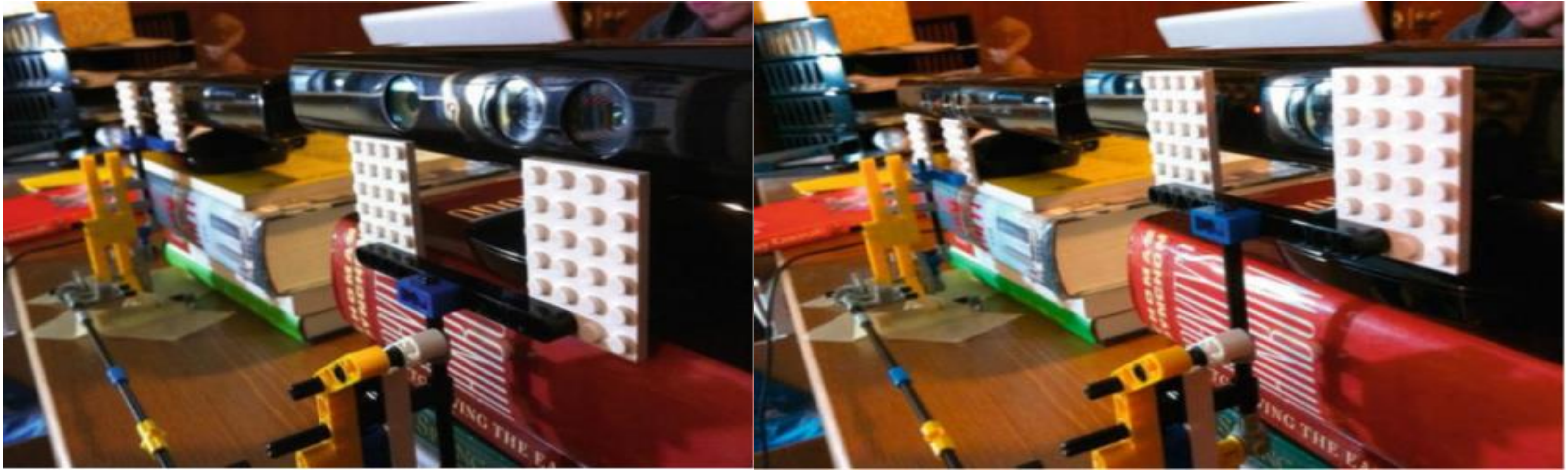
```
while(zoom)
{
if(direct_z < 60.0)
{

    zoomin_0 = 0;
    zoomout_1 = 1;
    //printf("zooming  out\n");
}
else if(direct_z > 120.0)
{
    //printf("zooming  in\n");
    zoomin_0 = 1;
}
else
{
    zoomout_1 = 0;
    }
if (zoomin_1 && zoomin_0 && !zoomout_1)
{
    printf("zooming  in\n");
    }
if(!zoomin_1 && !zoomin_0 && zoomout_1)
{
    printf("zooming  out\n");
```

Demonstration

Hardware Upgrade: Kinect-2.0

Multiple Kinect



The future of the technology

1.Humanoid

--more detailed motion recording

--related to the control of multi-motors

2.Security

--recognize abnormal motion

--keep record of human information and gives warning

3.Moore's Law

--Hardware would be much smaller

--More compilable to other devices(phones, glasses,etc)

4.Distant Human Robot Interface

--Military

--Global operation

References

- *C++ Primer 4th edition*
- *C++ How to program 8th edition*
- *A Practical Guide to Ubuntu Linux(3rd Edition)*
- *Hacking the Kinect*
- *RGB-D demo Google group:*
  https://groups.google.com/forum/#!topic/rgbdemo/WCFZzqT7JFI

Q&A

Thank you very much!