

Lab/Prac 2: Sensing & Perception: The Kinect “Can Do” Challenge

Matlab+Kinect Setup Guide

DRIVER CLEANUP AND COMPILER SETUP

1. Uninstall the Microsoft Kinect SDK if you have it installed.
2. Download Microsoft Visual Studio C++ Compiler (Make sure it matches your 32bit or 64bit system). This is available for free from Dreamspark for students of UQ.
3. In Matlab, you should be able to run: ‘mex -setup’ and choose the option for the Microsoft Visual Studio C++ compiler.

SOFTWARE DOWNLOAD

All Kinect software and toolboxes needed for this prac can be found in the following link:

<http://robotics.itee.uq.edu.au/~metr4202/software/>

Look under the ‘Kinect OpenNI Drivers and Middleware (NITE)’ heading. Download and install all the software listed.

All downloads in in this document can **also** be found in the following two links. Download the contained files from the links and install all components.

- **Kinect OpenNI installers and Matlab Codes:**

<http://robotics.itee.uq.edu.au/~metr4202/kinect/>

- **(and for a larger cache of OpenNI/Kinect SDK files:**

<http://dropbox.eait.uq.edu.au/uqssing7/kinect/>

KINECT DRIVERS

The following uses the direct download links for the drivers - use the cached links given in ‘SOFTWARE DOWNLOAD’ for easy listing.

1. Download the following three driver sets from this site or you can use the links given in the

'Software Download' section **(USE THE CORRECT BIT-AGE)**

(<http://www.openni.org/Downloads/OpenNIModules.aspx>)

- a. OpenNI Binaries / Unstable / Windows (32 or 64 bit)
- b. OpenNI Compliant Middleware Binaries / Unstable / Windows (32 or 64 bit)
- c. OpenNI Compliant Hardware Binaries / Unstable / Windows (32 or 64 bit)
2. Install all 3 files (in any order).
3. Download the Kinect Drivers (<https://github.com/avin2/SensorKinect/downloads>)
 - a. SensorKinect093-Bin-WinXX-v5.1.2.1.msi / Windows (32 or 64 bit)
4. Install this driver.
5. Plug the Kinect into both the wall power and the USB port on your computer.
6. Go into Control Panel > Device Manager and under 'Primesense' there should be devices like 'Kinect Motor', 'Kinect Device', 'Kinect Audio', 'Kinect Video' or similar.
7. If this shows up then you are doing well!
8. If the drivers show up under 'Microsoft Kinect' instead of 'Primesense' then you need to redo step 3 or uninstall the Official Kinect SDK if you have it installed. This is because the Official drivers can sometimes interfere with the ones we will be using.
9. It is possible to have both driver sets installed though. To change the drivers, right click on the device in 'Device Manager', click Update Drivers, click 'Browse Drivers', then 'Let me pick' and then choose the Primesense Drivers.
10. Navigate to the OpenNI -> Sample -> Bin -> Debug Folder and run and of the sample '.exe' files. (Usually in the directory C:\Program Files\OpenNI\Samples\Bin\Debug).
11. If they work without error you are done this section!

QUICK TEST

- Grab an object and hold it in front of the camera.
- Run the 'NiViewer64' file, usually in (C:\Program Files\OpenNI\Samples\Bin64\Release\NiViewer64.exe)
- Take the object at about 50cm away from the kinect and spin it around. Save the file as a .ONI file from NiViewer64 (Right Click -> Capture -> Start)

Now that all the drivers have been installed – the Kinect should be all connected to your system.

Now we need to make the device (through the drivers we installed) interface with MATLAB.

MATLAB INTEGRATION (PRIMARY)

1. This version requires that you have Microsoft Visual Studio compiler (which you can download from Dreamspark for free). All the university computers have this installed already.
2. Down the following Matlab Toolbox (<http://www.mathworks.com/matlabcentral/fileexchange/30242>)
3. Unzip the files and copy paste them into your Matlab toolbox directory
4. Load up Matlab and add ('Add Folders and Subfolders') the directory you copied Kinect Mex too.
5. See if you can run the compile_cpp_files.m Matlab file by using the command:
`>> compile_cpp_files('C:\Program Files\OpenNI\')`
6. If you are using 64 bit, then you will have to change the .m file a tiny bit. Change the lines:
 - a. `OpenNiPathLib=[OpenNiPath 'Lib'];` **TO** `OpenNiPathLib=[OpenNiPath 'Lib64'];`
 - b. `mex('-v', ['-L' OpenNiPathLib], '-lopenNI');` **TO** `mex('-v', ['-L' OpenNiPathLib], '-lopenNI64');`
7. After this run any of the Example.m files. Examples of what you should get is on the last page.

MATLAB INTEGRATION (ALTERNATE) (WILL NOT WORK WITH MATLAB NEWER THAN R2011A)

1. Download the following Matlab Toolbox (<http://sourceforge.net/projects/kinect-mex/>)
2. Documentation found here: (<http://kinect-mex.sourceforge.net/>)
3. Unzip the files and copy paste into your Matlab toolbox directory.
4. Load up Matlab and add ('Add Folders and Subfolders') the directory you copied Kinect Mex too.
5. Run any of the .m files with 'sample_' in them.
6. This should automatically open up one or several windows from the Kinect. They will iterate for ~100 cycles and print the FPS in the Matlab terminal.

TROUBLESHOOTING

Often times in the installation of the OpenNI drivers, the Path Environment variables will not be set, or only some of them will be set. To access the Environment Variables, go to Control Panel -> System Settings -> Advanced System Settings -> Environment Variables.

In here, check that you have the following entries:

-OPEN_NI_BIN	= C:\Program Files (x86)\OpenNI\Bin
-OPEN_NI_INCLUDE	= C:\Program Files (x86)\OpenNI\Include
-OPEN_NI_INSTALL_PATH	= C:\Program Files (x86)\OpenNI\

-OPEN_NI_LIB = C:\Program Files (x86)\OpenNI\Lib

or for 64 bit:

-OPEN_NI_BIN64 = C:\Program Files\OpenNI\Bin

-OPEN_NI_INCLUDE64 = C:\Program Files\OpenNI\Include

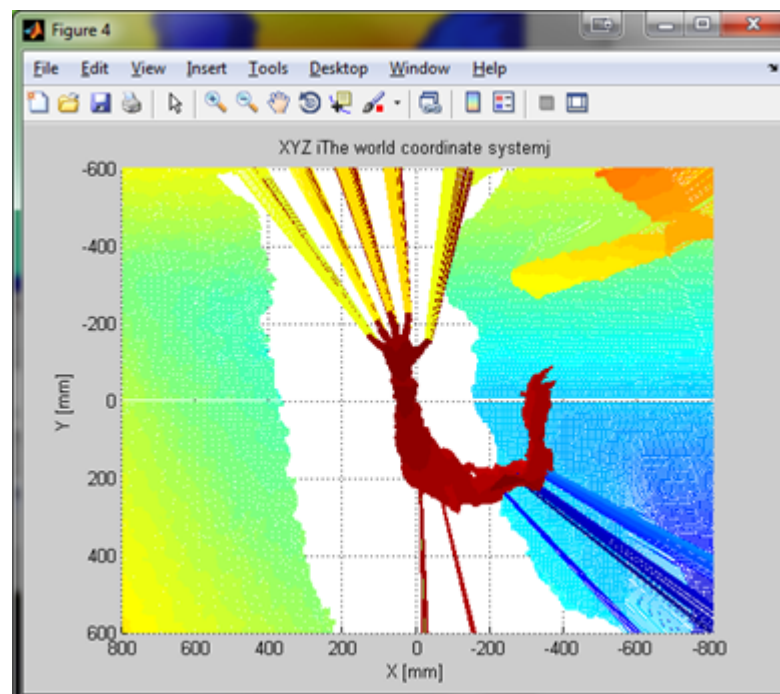
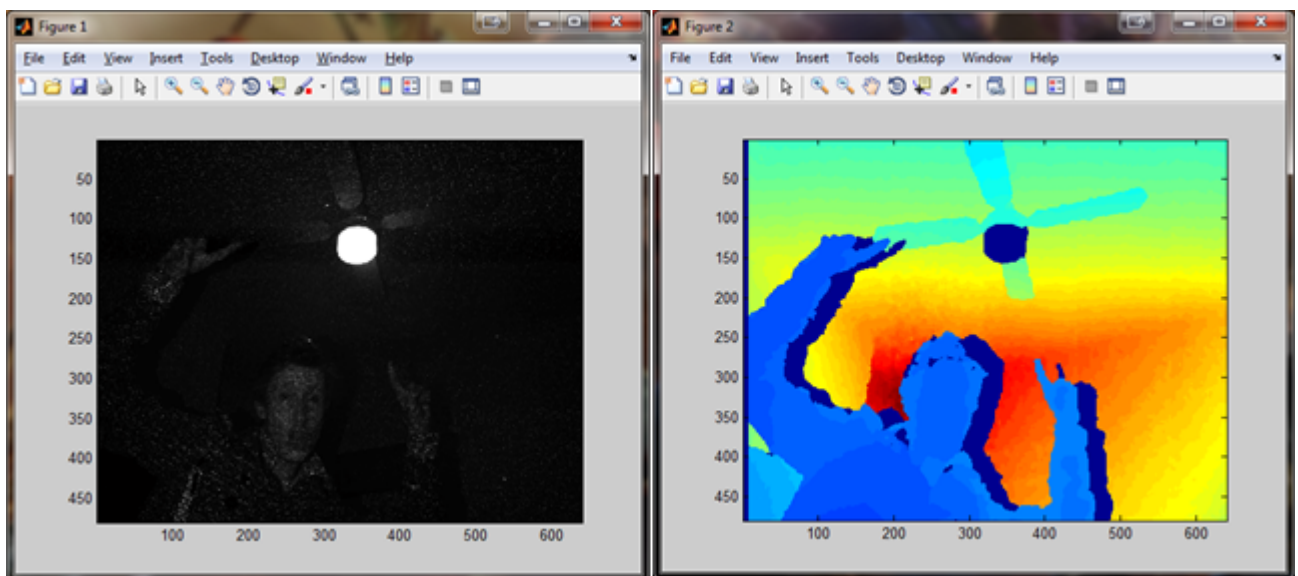
-OPEN_NI_INSTALL_PATH64 = C:\Program Files\OpenNI\

-OPEN_NI_LIB64 = C:\Program Files\OpenNI\Lib

EXAMPLES

1. An example .m file with basic blob detection and edge perception done by the tutors will be shown in the tutorial (but you can't have it! J)
2. Now go forth and investigate how to manipulate these images you receive from the Kinect in Matlab with the given documentation and Matlab functions!
3.but help will be given in the Tutorials and Practical sessions.

This is what your Matlab interface should look like:



An Example of what your screen should look like:

