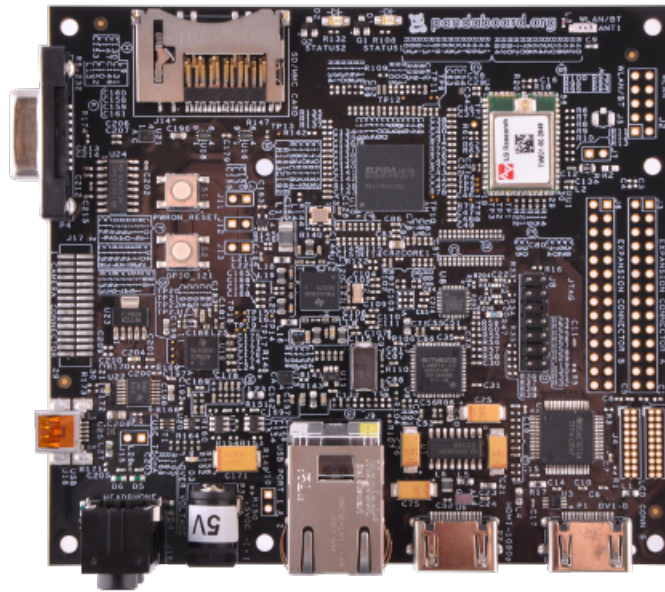




pandaboard.org



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>.

Pandaboard Guide: Computer Vision with Kinect, OpenCV on Pandaboard

Jayneil Dalal(jayneil.dalal@gmail.com)

November 12, 2012

Abstract

In this guide, I will describe how to install OpenCV with Kinect support on Ubuntu 12.04 running on the Pandaboard. This guide targets enthusiasts who would like to do computer vision on the pandaboard. These instructions will work for Pandaboard as well as Pandaboard ES.

Part-1: Installing OpenCV 2.4.2

These instructions will also work if you want to install OpenCV on your desktop machine. If you are not interested in installing the latest version of OpenCV, then you can install it from the Ubuntu software repositories by running the command below in the terminal:

```
$ sudo apt-get install libopencv-dev
```

Steps:

In this approach OpenCV will be built from source and you will have to install the dependencies prior to that.

1. Update the system

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

2. Install the dependencies

- Essentials: These are libraries and tools required by OpenCV

```
sudo apt-get install build-essential checkinstall cmake pkg-config yasm
```

- Image I/O: Libraries for reading and writing various image types. If you do not install then the versions supplied by OpenCV will be used

```
sudo apt-get install libtiff4-dev libjpeg-dev libjasper-dev
```

- Video I/O: You need some or all of these packages to add video capturing/encoding/decoding capabilities to the highgui module

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libxine-dev libv4l-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
```

- Other dependencies

```
sudo apt-get install checkinstall gir1.2-gst-plugins-base-0.10 gir1.2-gstreamer-0.10 libgstreamer-plugins-base0.10-dev libgstreamer0.10-dev libslang2-dev libxine-dev libxine1-bin libxml2-dev
```

- Python: Packages needed to build the Python wrappers

```
sudo apt-get install python-dev python-numpy
```

- Other third-party libraries Install Intel TBB to enable parallel code in OpenCV

```
sudo apt-get install libtbb-dev
```

- GUI: The default back-end for highgui in Linux is GTK. You can optionally install QT instead of GTK and later enable it in the configuration

```
sudo apt-get install libqt4-dev libgtk2.0-dev
```

3. Configure OpenCV 2.4.2

- Download Opencv from the link below: http://downloads.sourceforge.net/project/opencvlibrary/opencv_unix/2.4.2/OpenCV-2.4.2.tar.bz2
- Extract the downloaded file to your home folder -Navigate to the extracted folder using the terminal

- Make a sub folder 'build' and navigate to it using the terminal.
- Run the following command now:

```
sudo apt-get install cmake-gui cmake-gui
```

- Now provide the source folder and in in binary folder option, provide the 'build' folder path.
- Press configure.
- Now select the boxes to include those functionalities and press configure again to update.
- Once you are sure, press generate.

4. Compile OpenCV

Run the following code in the terminal:

```
make
```

5. Install OpenCV

```
sudo make install
```

6. Configure Linux

Edit the /etc/ld.so.conf.d/opencv.conf file and add /usr/local/lib to it

```
sudo gedit /etc/ld.so.conf.d/opencv.conf sudo ldconfig
```

Edit the bash.rc file and add the following to it: PKG_CONFIG_PATH=\$PKG_CONFIG_PATH:/usr/local/lib
export PKG_CONFIG_PATH

```
sudo gedit /etc/bash.bashrc
```

Now,logout of the system or restart

7. Configure OpenCV with codeblocks

Go to Project>Build Options -Go to Compiler>Other Settings

Write the following:

```
'pkg-config --cflags opencv'
```

Now go the Linker settings and write the following in other settings:

```
'pkg-config --libs opencv'
```

Note:- To run your code from the command line, write the following in the terminal:

```
g++ 'pkg-config --cflags --libs opencv' -o main main.cpp
```

Part -2: Remove OpenCV

Refer this section only if you have messed up your installation or you would like to install a newer version of OpenCV or you want to remove OpenCV.

- Go the build folder(Inside OpenCV folder)
- Now run the following command in the terminal:

```
sudo make uninstall
```

- Delete the entire OpenCV folder
- Run the following command in the terminal

```
sudo find / -name "*opencv*" -exec rm -i {} \;
```

Note:- The above command will delete every file with the 'opencv' in it!

- Edit the /etc/ld.so.conf.d/opencv.conf file and remove /usr/local/lib from it

```
sudo gedit /etc/ld.so.conf.d/opencv.conf  
sudo ldconfig
```

Edit the bash.rc file and remove the following from it:

```
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig  
export PKG_CONFIG_PATH
```

```
sudo gedit /etc/bash.bashrc
```

- To make sure check the following paths: /usr/local/bin /usr/local/lib

Part -3: Installing OpenNI+Kinect drivers on Pandaboard running Ubuntu 12.04

OpenNI Installation

1. Install the dependencies

```
sudo apt-get update  
sudo apt-get install gcc-multilib libusb-1.0.0-dev git-core  
build-essential  
sudo apt-get install doxygen graphviz default-jdk freeglut3-dev  
libopencv-dev
```

2. Create a folder for holding the download and the installation

```
mkdir kinect
cd kinect
```

3. Download the latest unstable version of the OpenNI software from GitHub:

```
cd kinect
git clone git://github.com/OpenNI/OpenNI.git
cd OpenNI
git checkout unstable
```

Note:- The version mentioned in the README file should at least be 1.5.4.0.

4. Modify the compiler flags

The software is set up for software floating point but the Ubuntu has been compiled for hardware floating point. So, the compiler flags need to be modified. To do this, enter:

```
cd Platform/Linux/Build
sudo gedit Common/Platform.Arm
```

Remove the “-mfloat-abi” option in the file so that it finally looks like this:

```
ifeq "$(CFG)" "Release"
# Hardware specifying flags
CFLAGS += -march=armv7-a -mtune=cortex-a8 -mfpu=neon #-
mcpu=cortex-a8
# Optimization level, minus currently buggy optimizing methods (which break
bit-exact)
CFLAGS += -O3 -fno-tree-pre -fno-strict-aliasing
# More optimization flags
CFLAGS += -ftree-vectorize -ffast-math -funsafe-math-optimizations -fsingle-
precision-constant
endif
```

5. Build OpenNI To build, enter the following:

```
cd ~/kinect/OpenNI/Platform/Linux/CreateRedist ./RedistMaker.Arm
```

Note:-It's possible that this will result in an error because MAKE_ARGS includes “-j0” which is illegal. If this occurs, edit Redist_OpenNI.py and find the line that looks like: MAKE_ARGS += ' -j' + calc_jobs_number() and change it to MAKE_ARGS += ' -j1' This will build the OpenNI binaries and create a folder called Redist.

6. Install OpenNI

```
cd ~/kinect/OpenNI/Platform/Linux/Redist/OpenNI-Bin-Dev-Linux-Arm-
v1.5.4.0
sudo ./install.sh
```

Note:-The version number in the path may need to be changed to reflect whatever is the current version.

Kinect Driver Installation

1. Download the source code like this:

```
cd ~/kinect
git clone git://github.com/avin2/SensorKinect.git
```

2. Configure the compiler flags The software floating point option has to be turned off. Enter the code below in the terminal

```
cd ~/kinect/SensorKinect/Platform/Linux/Build/Common
```

Edit the Platform.Arm file to remove the “-mfloat-abi” option as we had done earlier for OpenNI.

3. Compile

Execute the following code

```
cd ~/kinect/SensorKinect/Platform/Linux/CreateRedist ./RedistMaker
```

This will build the driver and create a Redist folder.

4. Configuring the USB port

```
cd ~/kinect/SensorKinect/Platform/Linux/Redist/Sensor-Bin-Linux-Arm-v5.1.2.1
sudo gedit Config/GlobalDefaultsKinect.ini
```

There’s an edit required to the config file or else it will not select the correct USB port. Edit Config/GlobalDefaultsKinect.ini. There’ll be a line that looks like:

```
;UsbInterface=2
```

Change this line to : UsbInterface=1

Note:- The line has been uncommented and the interface has been changed from 2 to 1.

5. Install the driver

```
cd ~/kinect/SensorKinect/Platform/Linux/Redist/Sensor-Bin-Linux-Arm-v5.1.2.1
sudo ./install.sh
```

6. Testing It’s worth trying to run a sample to see if this has worked. Enter:

```
cd ~/kinect/OpenNI/Platform/Linux/Redist/OpenNI-Bin-Dev-Linux-Arm-
v1.5.4.0
cd Samples/Bin/Arm-Release
./Sample-NiSimpleRead
```

This should result in a series of values being displayed which vary if something is waved in front of the Kinect.

Part -4: Sample code to access depth data from Kinect in OpenCV

```
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/opencv.hpp"
#include <iostream>

using namespace cv;
using namespace std;

int main( int argc, char** argv )
{
    VideoCapture capture( CV_CAP_OPENNI );
    for(;;)
    {
        //variable declaration
        Mat depthMap;
        Mat show;
        const float scaleFactor = 0.05f;
        capture >> depthMap;
        depthMap.convertTo( show, CV_8UC1, scaleFactor );
        imshow( "depth map", show );
        if( waitKey( 30 ) >= 0 )
            break;
    }
}
```

References

- [1] http://docs.opencv.org/2.4.3rc/doc/user_guide/ug_highgui.html