Drones and Autonomous Systems Laboratory

Ball balancing on the beam class 3

-OPENCV Practice-

DONGBIN KIM –
Ph.D. Candidate
Mechanical Engineering,
University of Nevada, Las Vegas
1. Introduction – OPENCV

OPENCV – Introduction

1. Open source package/library for real-time computer vision
2. Library support C++, Python, Java, etc.
3. Developed by Intel
4. Used for various applications includes
   1. 2D and 3D feature toolkits
   2. Facial recognition
   3. Gesture recognition
   4. Human-Computer Interaction
   5. Motion understanding
   6. Object identification
1. Introduction – OPENCV for this course

1. OPENCV 4 Package installation on Ubuntu 16.04
2. C++ or Python library will be used
3. Used for various applications includes
   1. Camera Performance Check up
   2. Color Recognition
   3. Object identification
2. OPENCV4 Installation – Ubuntu 16.04

*OpenCV Installation confirm. And remove for new version installation

```bash
webnautes@webnautes-pc:~$ pkg-config --modversion opencv
Package opencv was not found in the pkg-config search path. Perhaps you should add the directory containing `opencv.pc' to the PKG_CONFIG_PATH environment variable
No package 'opencv' found

(if opencv is installed)
webnautes@webnautes-pc:~$ pkg-config --modversion opencv
2.4.9.1

(Removal initially installed OpenCV)
sudo apt-get purge libopencv* python-opencv
$sudo apt-get autoremove
```
2. OPENCV4 Installation – Ubuntu 16.04

*Package Upgrade and Update

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

*Required Package for compiling OpenCV

```
sudo apt-get install build-essential cmake
sudo apt-get install pkg-config
```
2. OPENCV4 Installation – Ubuntu 16.04

*Package Upgrade and Update

```bash
sudo apt-get update
sudo apt-get upgrade
```

*Required Package for compiling OpenCV

```bash
sudo apt-get install build-essential cmake
sudo apt-get install pkg-config
```
2. OPENCV4 Installation – Ubuntu 16.04

*The library to read /record the image file

```bash
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
```

*The library to read/ record the specific codec video file

```bash
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libxvidcore-dev libx264-dev libxine2-dev
```

* Video4Linux package - Real time video capture, device driver and API

```bash
sudo apt-get install libv4l-dev v4l-utils
```

*Gstreamer – Library for video streaming

```bash
sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
```
2. OPENCV4 Installation – Ubuntu 16.04

*Qt4 for GUI System

```
sudo apt-get install libqt4-dev
```

*OpenGL Supportive Library

```
sudo apt-get install mesa-utils libgl1-mesa-dri libqt4-opengl-dev
```

*Optimizing OpenCV support

```
sudo apt-get install libatlas-base-dev gfortran libeigen3-dev
```

*Python 2.7-dev and python3-dev package for OpenCV Python binding

```
sudo apt-get install python2.7-dev python3-dev python-numpy python3-numpy
```
2. OPENCV4 Installation – Ubuntu 16.04

*OPENCV Configuration

```bash
mkdir opencv
cd opencv
```

*OpenCV 4 source code download

```bash
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.0.1.zip
Unzip opencv.zip
```

*OpenCV contrib(Extra modules) download

```bash
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.0.1.zip
Unzip opencv_contrib.zip
```

*Confirmation for directories

```bash
ls -d */
-> opencv4.0.1/ opencv_contrib-4.0.1/
```
2. OPENCV4 Installation – Ubuntu 16.04

*Go to opencv-4.0.1 folder to create another folder “build”

```bash
mkdir build
cd build
```

*Type followings

```bash
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D WITH_TBB=OFF \n-D WITH_IPP=OFF \n-D WITH_1394=OFF \n-D BUILD_WITH_DEBUG_INFO=OFF \n-D BUILD_DOCS=OFF \n-D INSTALL_C_EXAMPLES=ON \n-D INSTALL_PYTHON_EXAMPLES=ON \n-D BUILD_EXAMPLES=OFF \n-D BUILD_TESTS=OFF \n-D BUILD_PERF_TESTS=OFF \n-D WITH_QT=ON \n-D WITH_GTK=OFF \n-D WITH_OPENGL=ON \n-D OPENCV_EXTRA_MODULES_PATH=../opencv_contrib-4.0.1/modules \n-D WITH_V4L=ON \n-D WITH_FFMPEG=ON \n-D WITH_XINE=ON \n-D BUILD_NEW_PYTHON_SUPPORT=ON \n-D OPENCV_GENERATE_PKGCONFIG=ON ../
```
2. OPENCV4 Installation – Ubuntu 16.04

*If you see the following messages, it is installed successfully

```
-- Configuring done
-- Generating done
-- Build files have been written to: /home/webnautes/opencv/opencv-4.0.1/build
```

*Make sure that you can see the below on your result

```
-- Python 2:
--   Interpreter: /usr/bin/python2.7 (ver 2.7.12)
--   Libraries: /usr/lib/x86_64-linux-gnu/libpython2.7.so (ver 2.7.12)
--   numpy: /usr/lib/python2.7/dist-packages/numpy/core/include (ver 1.11.0)
--   install path: lib/python2.7/dist-packages/cv2/python-2.7

-- Python 3:
--   Interpreter: /usr/bin/python3 (ver 3.5.2)
--   Libraries: /usr/lib/x86_64-linux-gnu/libpython3.5m.so (ver 3.5.2)
--   numpy: /usr/lib/python3/dist-packages/numpy/core/include (ver 1.11.0)
--   install path: lib/python3.5/dist-packages/cv2/python-3.5

-- Python (for build): /usr/bin/python2.7
```
2. OPENCV4 Installation – Ubuntu 16.04

*Check out the number of CPU core before you compiling

```
    cat /proc/cpuinfo | grep processor | wc -l
```

*Compiling it with “make” order in your build folder

```
    Time make -j4
```

*You can see the following results when you are done successfully

```
[100%] Built target opencv_stitching
Scanning dependencies of target opencv_python2
Scanning dependencies of target opencv_python3
[100%] Building CXX object modules/python2/CMakeFiles/opencv_python2.dir/__/src2/cv2.cpp.o
[100%] Building CXX object modules/python3/CMakeFiles/opencv_python3.dir/__/src2/cv2.cpp.o
[100%] Linking CXX shared module ../../../lib/cv2.so
[100%] Linking CXX shared module ../../../lib/python3/cv2.cpython-36m-x86_64-linux-gnu.so
[100%] Built target opencv_python2
[100%] Built target opencv_python3

real 13m36.700s
user 41m52.880s
sys 1m50.322s
```
2. OPENCV4 Installation – Ubuntu 16.04

*Install the result from the compile in your build folder

```bash
sudo make install
```

*find it /etc/ld.so.conf.d/ directory has the /usr/local/lib

```bash
cat /etc/ld.so.conf.d/*
```

```
/usr/lib/x86_64-linux-gnu/libfakeroot
# libc default configuration
/usr/local/lib
# Multiarch support
/usr/local/lib/x86_64-linux-gnu
/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu
# Legacy biarch compatibility support
/lib32
/usr/lib32
```
2. OPENCV4 Installation – Ubuntu 16.04

*Install the result from the compile in your build folder

```bash
sudo make install
```

*find it /etc/ld.so.conf.d/ directory has the /usr/local/lib

```bash
cat /etc/ld.so.conf.d/*
```

```bash
/usr/lib/x86_64-linux-gnu/libfakeroof
# libc default configuration
/usr/local/lib
# Multiarch support
/usr/local/lib/x86_64-linux-gnu
/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu
# Legacy biarch compatibility support
/lib32
/usr/lib32
```

*Lastly

```bash
sudo ldconfig
```
2. OPENCV4 Installation – Installation confirmation test

*C/C++ test - compile

g++ -o facedetect /usr/local/share/opencv4/samples/cpp/facedetect.cpp $(pkg-config opencv4 --libs --cflags) -std=c++11

*C/C++ test – run the tutorial, face detection

./facedetect --cascade="/usr/local/share/opencv4/haarcascades/haarcascade_frontalface_alt.xml" --nested-cascade="/usr/local/share/opencv4/haarcascades/haarcascade_eye_tree_eyeglasses.xml" --scale=1.3

*Python test – version confirmation

webnautes@webnautes-pc:/openvc/openvc-4.0.1/builds$ python
Python 2.7.15rc1 (default, Apr 15 2018, 21:51:34)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.0.1'

webnautes@webnautes-pc:/openvc/openvc-4.0.1/builds$ python3
Python 3.6.5 (default, Apr 1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.0.1'
2. OPENCV4 Installation – Installation confirmation test

*Python2 test, face detection

```bash
python /usr/local/share/opencv4/samples/python/facedetect.py --cascade
"/usr/local/share/opencv4/haarcascades/haarcascade_frontalface_alt.xml" --nested-cascade
"/usr/local/share/opencv4/haarcascades/haarcascade_eye_tree_eyeglasses.xml" /dev/video0
```

*Python3 test, face detection

```bash
python3 /usr/local/share/opencv4/samples/python/facedetect.py --cascade
"/usr/local/share/opencv4/haarcascades/haarcascade_frontalface_alt.xml" --nested-cascade
"/usr/local/share/opencv4/haarcascades/haarcascade_eye_tree_eyeglasses.xml" /dev/video0
```
2. OPENCV4 Installation – ROS Integration

*You may go through these error after ROS installation

```
pi@pi-15U560-MFLGL:~$ python3
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: /opt/ros/kinetic/lib/python2.7/dist-packages/cv2.so: undefined symbol: PyObject_Type
pi@pi-15U560-MFLGL:~$
```
2. OPENCV4 Installation – ROS Integration

*The reason : python3’s cv2.so has different name. We need to change this

```
$ cd /usr/local/lib/python3.5/dist-packages
$ sudo mv /opt/ros/kinetic/lib/python2.7/dist-packages/cv2.so /opt/ros/kinetic/lib/python2.7/dist-packages/cv2.so.old
$ ls -al /opt/ros/kinetic/lib/python2.7/dist-packages

total 5172
drwxrwsr-x 2 root staff 4096 8 월 10 13:03 ./
drwxrwsr-x 3 root staff 4096 7 월 31 09:31 ../
-rw-r--r-- 1 root staff 5286904 8 월 10 13:00 cv2.cpython-35m-x86_64-linux-gnu.so

$ sudo ln -s `pwd`/cv2.cpython-35m-x86_64-linux-gnu.so `pwd`/cv2.so
$ ls -al

total 5176
drwxrwsr-x 2 root staff 4096 8 월 10 13:34 .
drwxrwsr-x 3 root staff 4096 7 월 31 09:31 ..
-rw-r--r-- 1 root staff 5286904 8 월 10 13:00 cv2.cpython-35m-x86_64-linux-gnu.so
lrwxrwxrwx 1 root staff 74 8 월 10 13:34 cv2.so -> /usr/local/lib/python3.5/dist-packages/cv2.cpython-35m-x86_64-linux-gnu.so
```
3. OPENCV Tutorials

1. Camera Performance Checkup
   1. Important to know the sampling time
   2. Gives various performance choice for specific application
   3. Frame per seconds (FPS) measurements

2. Color Recognition
   1. Using HSV value to recognize the color
   2. Adjusting the HSV value will recognize different colors
4. Additional Information – HSV vs RGB

1. **RGB**: defined by listing how much red, green, and blue is contained in a single value. The more of each color added, the brighter it becomes.

2. **HSV**: a color system that describes a **Hue** shift, **Saturation**, and **Value**. It can provide the better selection of colors.

Ref: [https://www.kirupa.com/design/little_about_color_hsv_rgb.htm](https://www.kirupa.com/design/little_about_color_hsv_rgb.htm)
4. Additional Information – Field of View

1. Beam length : 50cm
2. Horizontally fit in the camera that has resolution of (360 x 240)

Important Question.

Is 1 pixel means 1.38mm???
(By 500 mm/360 pixels)
4. Additional Information – Field of View

Important Question.

Is 1 pixel means 1.56mm???

(By 500 mm/320 pixels)

1. Beam length : 50cm
2. Horizontally fit in the camera that has resolution of (320 x 240)

1.56mm/320 pixels
4. Additional Information – Field of View

- Field of View
  1. Every camera has different field of view
  2. Field of View is determined by focal angle and focal distance
  3. Example: SONY PS3 Camera
  4. Focal length: 75 degree
  5. Focal distance (Camera to the beam): 360mm

- Field of View: 552.456mm
- -> 1 pixels = 1.534 mm
- Position of the object should be recalculated after achieved

<table>
<thead>
<tr>
<th>angle</th>
<th>tan(α)</th>
<th>focal distance</th>
<th>opposite</th>
<th>field of view</th>
</tr>
</thead>
<tbody>
<tr>
<td>56°</td>
<td>1.4823</td>
<td>1 inch</td>
<td>.53 inches</td>
<td>1.06 inches</td>
</tr>
<tr>
<td>75°</td>
<td>1.4823</td>
<td>2 inches</td>
<td>.77 inches</td>
<td>1.54 inches</td>
</tr>
<tr>
<td>56°</td>
<td>1.4823</td>
<td>2 inches</td>
<td>1.06 inches</td>
<td>2.12 inches</td>
</tr>
<tr>
<td>75°</td>
<td>1.4823</td>
<td>2 inches</td>
<td>1.53 inches</td>
<td>3.06 inches</td>
</tr>
<tr>
<td>56°</td>
<td>1.4823</td>
<td>4 inches</td>
<td>2.13 inches</td>
<td>4.26 inches</td>
</tr>
<tr>
<td>75°</td>
<td>1.4823</td>
<td>4 inches</td>
<td>3.07 inches</td>
<td>6.14 inches</td>
</tr>
</tbody>
</table>
5. OPENCV Practice

Let’s Practice with Visual Studio Code or PyCharm
5. Homework

*Homework for next course*

1. Color Recognition tutorial revision *(Easy)*
2. Color Position tracking – Achieve the position of the object from your code *(Hard)*
3. ROS installation tutorials from 1-9 *(Intermediate)*