**References:**

* <https://www.e-consystems.com/blog/camera/how-to-access-cameras-using-opencv-with-python/>

**OpenCV Python to stream the color camera**

Open Computer Vision (OpenCV) is an open source BSD licensed image processing bundle that contains functions for all type of image processing functionality from basic image decoding, enhancement, color space conversion, object detection, object tracking and so on.

This blog is intended to show how to access and use the cameras for OpenCV by using a simple Python script.

* See3CAM Cameras

See3CAM is the USB 3.0 camera series from e-con Systems™. These cameras are UVC-compliant that has Plug & Play support on Windows/Linux which does not require to install additional device drivers manually.

For this sample application, we are going to use e-con Systems™ 13MP Auto focus USB 3.0 camera – See3CAM\_130. We’ll see step by step procedure on how to access the See3CAM\_130 camera from a simple OpenCV-Python application, which will grab the frame from camera and display in the preview window. Let us start from building the OpenCV and its dependencies in Ubuntu PC.

* Test System:

Host PC: Ubuntu-14.04

OpenCV Version: 3.4.1

Python Version: 2.7

* Download OpenCV source

Download the required OpenCV version for Linux from the following command

$ wget https://github.com/opencv/opencv/archive/3.4.1.zip

* Install Dependencies:

OpenCV requires the following dependencies to work with the USB cameras. Install the dependencies from the following commands:

# libav video input/output development libraries

$ sudo apt-get install libavformat-dev libavutil-dev libswscale-dev

# Video4Linux camera development libraries

$ sudo apt-get install libv4l-dev

# OpenGL development libraries (to allow creating graphical windows)

$ sudo apt-get install libglew-dev

# GTK development libraries (to allow creating graphical windows)

$ sudo apt-get install libgtk2.0-dev

# Install the OpenCV python

$ sudo apt-get install python-opencv

* Configure OpenCV:

After installing the dependencies, now we need to build and install OpenCV using the following commands:

$ unzip opencv-3.4.1.zip

$ cd opencv-3.4.1

$ mkdir build && cd build

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D WITH\_TBB=OFF -D BUILD\_TBB=OFF -D WITH\_V4L=ON -D WITH\_LIBV4L=OFF -D BUILD\_TESTS=OFF -D BUILD\_PERF\_TESTS=OFF ..

* Compile and install:

The following command will build and install OpenCV libraries in the location – “/usr/local/lib/”

$ sudo make j4 install. We now have built & installed OpenCV in the Ubuntu PC. Let us look how to prepare a sample Python application to stream the camera using OpenCV.

The following sample OpenCV python code explain how to open the device video node, set the resolution, grab the frame and then display the frame in preview window.

1. Import the OpenCV python package

import cv2

1. Open the camera video node to access the See3CAM\_130

# Open the device at the ID 0

cap = cv2.VideoCapture(0)

#Check whether user selected camera is opened successfully.

if not (cap.isOpened()):

print(“Could not open video device”)

1. Set the resolution of the camera by using OpenCV camera control properties

#To set the resolution

cap.set(cv2.cv.CV\_CAP\_PROP\_FRAME\_WIDTH, 640)

cap.set(cv2.cv.CV\_CAP\_PROP\_FRAME\_HEIGHT, 480)

1. Grab the frame continuously from the camera and show it in the preview window using the while loop. Enter ‘q’ key, to break the loop and exit the application

while(True):

# Capture frame-by-frame

ret, frame = cap.read()

# Display the resulting frame

cv2.imshow(‘preview’,frame)

#Waits for a user input to quit the application

if cv2.waitKey(1) & 0xFF == ord(‘q’):

break

1. Release the camera, then close all of the imshow() windows.

# When everything done, release the capture

cap.release()

cv2.destroyAllWindows()

* Save the sample code as “camera\_stream.py”.

import cv2

# Open the device at the ID 0

cap = cv2.VideoCapture(0)

#Check whether user selected camera is opened successfully.

if not (cap.isOpened()):

print(“Could not open video device”)

#To set the resolution

cap.set(cv2.cv.CV\_CAP\_PROP\_FRAME\_WIDTH, 640)

cap.set(cv2.cv.CV\_CAP\_PROP\_FRAME\_HEIGHT, 480)

while(True):

# Capture frame-by-frame

ret, frame = cap.read()

# Display the resulting frame

cv2.imshow(‘preview’,frame)

#Waits for a user input to quit the application

if cv2.waitKey(1) & 0xFF == ord(‘q’):

break

# When everything done, release the capture

cap.release()

cv2.destroyAllWindows()

* Now compile and run the program:

python2 camera\_stream.py

By using this simple python script, we shall be able to access and use the cameras for OpenCV.