

PDE4420/21 Session: Processing ROS Video with OpenCV

1. Aim

- Review basic ROS concepts, e.g., packages, nodes, topics
- Know how to transform ROS message into OpenCV data, and vice versa
- Use the ROS cv_bridge utility for processing ROS image streams with OpenCV
- Understand basic video/image processing using OpenCV

2. Content

You can refer to wiki.ros.org/vision_opencv for introduction to the use of OpenCV with ROS. There are two packages in vision_opencv stack: cv_bridge and image_geometry. We will focus on using cv_bridge.

First of all, check if OpenCV is installed in your VM Ubuntu environment. In a terminal, type the following command (% represents the linux command line prompt, not your input).

```
% pkg-config --modversion opencv
```

If you do not see OpenCV version, then you need to install OpenCV in your computer. Suggested version is OpenCV 2 or 3 (Do nothing for now, you will be able to install it in the next page).

(1) Preparing your computing environment

- Launch your Ubuntu virtual machine (VM) in your VMWare Player.
- Create a new catkin workspace, or go into the src subdir of your existing catkin workspace
- In the src subdir, download two ROS stacks using the following git command.

```
% git clone https://github.com/pirobot/rbx1.git
```

This ROS stack contains all example programs for the book, **ROS by Examples**, by Patrick Goebel

```
% git clone https://github.com/ros-drivers/usb\_cam.git
```

The usb_cam_node interfaces with standard USB cameras (e.g. the Logitech Quickcam) using libusb_cam and publishes images as sensor_msgs::Image

- Now you need to go to the catkin workspace where the above packages downloaded, and compile your workspace using **catkin_make**. After compilation, do not forget to source your new environment.
- Now you can test your laptop webcam. Run the following commands.

```
% roslaunch usb_cam usb_cam-test.launch
```

Does not work? Spend **15 minutes** to study the errors information, and try to resolve the problem.

- Very good, it works now. Now let's run the following command.

```
% roslaunch rbx1_vision usb_cam.launch video_device:=/dev/video0
```

It should work! But it does not. Analyse it (**Hint**: compare `usb_cam-test.launch` in `usb_cam` package with this launch file in `rbx1_vision` package).

So far, you've watched ROS image streams using your webcam, but you have no much freedom to manipulate the video. Now let's try to convert the ROS image streams to OpenCV format.

Google for Patrick Goebel's **ROS by Example**, you will find a PDF version. Let's start from Chapter 10.4. Run the following command to install OpenCV in your Ubuntu.

```
% sudo apt-get install ros-melodic-vision-opencv libopencv-dev python-opencv
```

Optionally, you can test your OpenCV installation according to Page 145 of **ROS by Example**. So far so good! But unfortunately, most of the Python program snippets in **ROS by Example** are for ROS Indigo, they may not work directly for ROS Melodic. You need to do some research while you use them.

Check UniHub module site. We have a working program for ROS Melodic to convert the webcam ROS image streams to OpenCV and display it. Download and save it in a new scripts subdir in your `usb_cam` ROS package. Then provide a ROS image stream by running the following (if you have closed the previous operation).

```
% roslaunch usb_cam usb_cam-test.launch
```

You can check by running in another terminal `rostopic list` to see if a topic named `/usb_cam/image_raw` is there.

Open a new terminal, and go to the scripts subdir in `usb_cam` package, run the following command.

```
% python opencv_example_1.py
```

Now you should see a video window showing yourself! This is different from the last one. This is controlled by OpenCV, which means you could manipulate the video using OpenCV APIs.

Open a new terminal and run the following command to further analyse the scenario.

```
% rqt_graph
```

You can see the nodes and topics. You should see there are two video subscribers, one is within ROS framework, one is your OpenCV window.

Okay, now it is time for us to do two more exercise. Hopefully you would know ROS and OpenCV better by doing it yourself. You do not need to know very well Python programming for this exercise, you just need to have a basic skillset of analysing problems.

Exercise 1:

Go to http://wiki.ros.org/cv_bridge/Tutorials, choose whichever language you like. Here I choose Python. Read through the tutorial. Given now you have a ROS image published, you can directly use the python code on the page to subscribe it, and then re-publish. Make sure you analyse the python code to understand it, and see if you need to modify it.

Exercise 2:

Go to **ROS by Example** Page 152. Following the last line to obtain `ros2opencv2.py`. I am sure it will not work in your environment. Please analyse the program, identify where is the piece of code to display the text in the video stream, incorporate that piece of code in your program so that your webcam will shoot your video with a text on it. This is a way to manipulate the OpenCV data.

You should use the rest time to do the above exercise. I will upload the program before the next session. Have fun!