

# MRT ASSIGNMENT 3

Madhav Agrawal

January 2024

## 1 Introduction

ROS (Robot Operating System) is an open-source middleware framework for developing and controlling robots. It has very helpful tools for development, debugging, and simulation, fostering a collaborative and extensive developer community. OpenCV is a python library, very helpful in image processing. ROS and OpenCV together is a great combination and can be used to do various difficult tasks involving robotics and image processing applications.

## 2 Summary

I already had `mrt_ws` in my home directory. So I created a new package named `imgprocess` and in that package I started building nodes for publishing and subscribing to the image message.

### 2.1 Publisher

In publisher node (python file) I imported `Image` (msg type) for pushing the message to topic (topic msg type is `Image`). Then I imported `cv2` module to capture video from webcam and get the captured frame in every loop. Then I imported `cv_bridge` to convert the opencv image into `Image` msg type so that it can be published over a topic.

In main code, I made a class of `ImagePublisher` where I initialize the node, webcam and `cv` bridge, then defined looping function and video capturing function both which converted captured frame into `Image` data and published it on the `imgmsg` topic and finally I made a main function to enter the script.

### 2.2 Subscriber

In subscriber node (python file) I imported `Image` (msg type) for getting the message from topic (topic msg type is `Image`). Then I imported `cv_bridge` to convert `Image` data into opencv image, imported `cv2`, `numpy` to analyse the canny image.

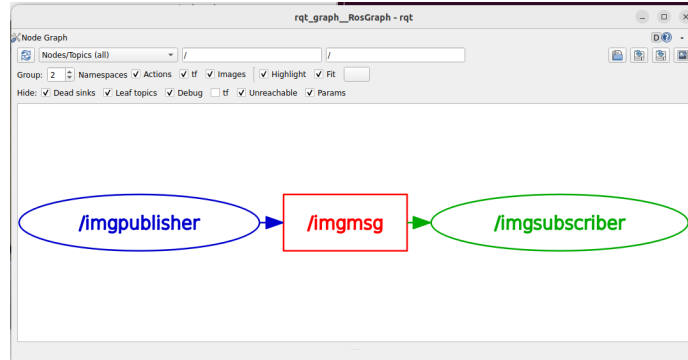


Figure 1: This is the rqt\_graph of the nodes currently running in ROS.

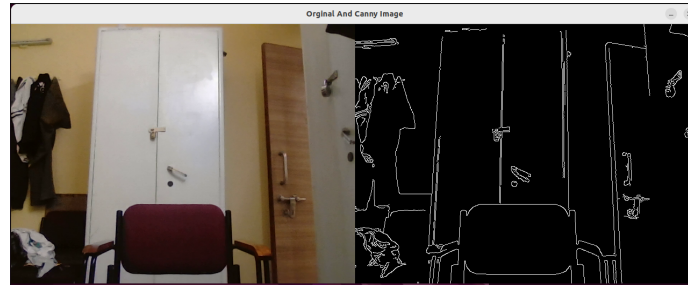


Figure 2: Sample of original image and canny image stacked together.

In main code, I made a class of ImageSubscriber where I initialize the node, cv\_bridge and threshold values for canny images. Then I made listener callback, canny\_image and stack\_image functions which get the image from the imgmsg topic and made canny image out of it, then converted the canny image of 1 channel into 3 channel (RGB) image and finally stacked the images horizontally. Then I made main function to enter the script. It initialized the object of ImageSubscriber class and finally run the node.

### 2.3 Launch File And Rqt\_graph

After those 2 nodes, I made a launch file to launch both the nodes simultaneously and I checked the communication between the nodes using rqt\_graph.

## 3 Conclusion

From this assignment of capturing image from webcam using one ROS node and publishing it to the ROS topic, then subscribing to the ROS topic and getting the image using another node, then applying image processing algorithms on

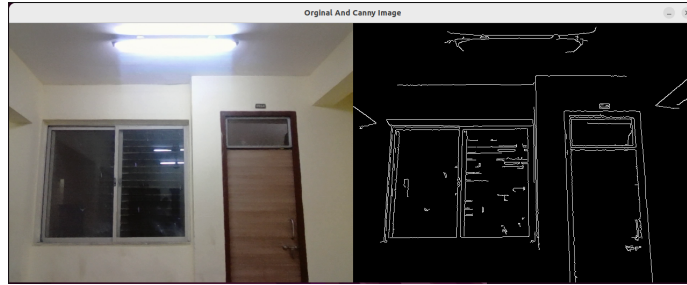


Figure 3: Sample of original image and canny image stacked together.

it, it is clear that ROS and opencv together can be used for various image processing related tasks in robotics and hence is a very great combo.

### 3.1 Challenges

- Finding the cv\_bridge commands to transform opencv image into Image datatype and vice-versa.
- Horizontal stacking of original and canny image as the canny image was of 1 channel and original was of 3 channel. So I had to change the shape of canny image to stack it with original image.