MRT Assignment I

Annirudh K P

210070009
Electrical Engineering, IIT Bombay

April 2022

Summary

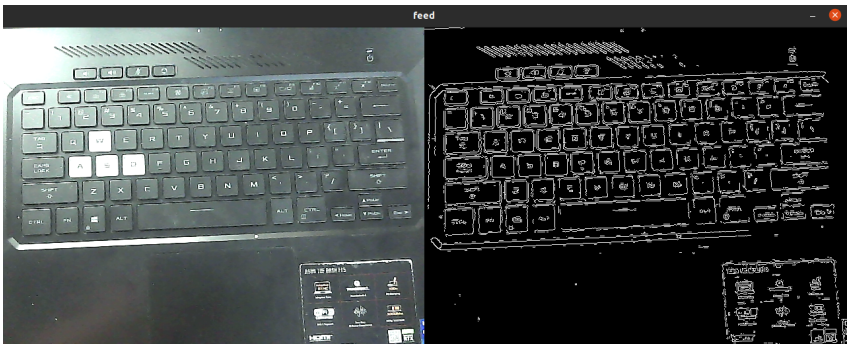
First I created a workspace, followed by creation of a package called `edgecam`. I built the package using `catkin_make`. Following this, I started writing the python codes for the nodes.

The publisher node python script, `img_publisher.py`, initiates the publisher node `webcam`. The rostopic, `imagefeed` is also put in place. The script runs a `while` loop around the condition of roscore not being shutdown. I then used the `OpenCV` tools to create a `VideoCapture` object called `cap` and captured the frame from this object and store it in the variable mat `frame`. Then using `cv_bridge`, I convert the image feed to a standard ros message. Following this, I publish it into the rostopic. And also importantly release the capture once the node is killed.

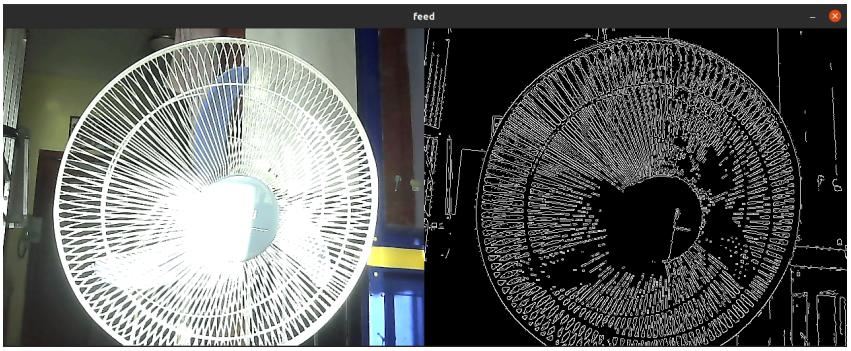
The subscriber node python script, `img_subcriber.py` initiates the subscriber node `vision`. Also it is subscribed to the rostopic `imagefeed`. Every time a message is received by this node, we call a `callback` function, which

- Converts the standard ros message into `OpenCV` image for handling
- Applies Canny Edge Detection Algorithm (understood how it works from here) using the `cv2.Canny()` function.
- converts the *edge image* into a 3-channel image using the `cv2.cvtColor()` function, following which I stack the original image and the converted *edge image* horizontally together.

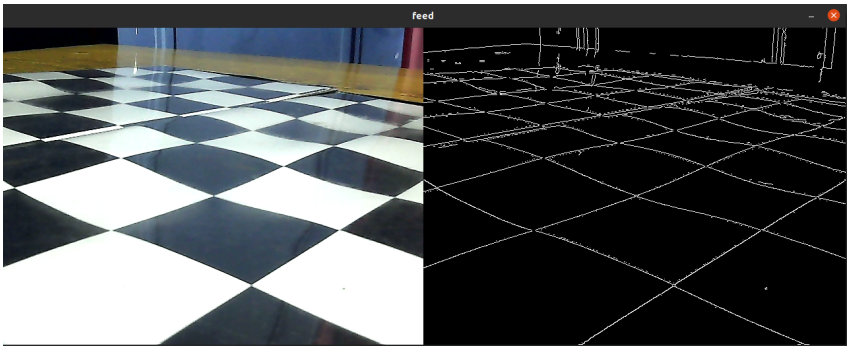
Using `cv2.imshow`, I display this continuous feed. I used a `.launch` file to run both the nodes. Below are some example feeds and the `rqt_graph` which shows the connection between the nodes.



Example 1



Example 2



Example 3

rqt_graph showing connections