Individual Lab Report 10

Name : Ardya Dipta Nandaviri Giri
Team B : Space Jockey
Team Members : Brian Boyle, Nathaniel Chapman, Songjie Zhong
ILR09
Submission Date : April 4, 2014

1. Introduction

This report gives a brief description of our team progress along with my individual contribution so far towards the MRSD project. After our last progress review on March 21, 2014, I have successfully created a ROS package for reading images from IP Camera and publish messages in ROS-image-message type.

2. Individual Progress & Challenges

Since 2 weeks ago, I mainly focused on creating a ROS package for reading images from IP Camera and publishing ROS image messages to be subscribed by Songjie's transformation matrix node. Additionally, I also integrated the image reading with the IMU sensor reading node from the last progress review.

The challenges of this work were so many ROS packages available for reading images from cameras. However, most of them were meant for PR2 robot or specific brand IP Cameras such as AXIS. Those packages were complicated with a lot of configurations and format, which were specified for that camera. Doing reverse-engineering on those packages was a hassle since a lot of configurations need to be understood. So, I decided to keep learning from OpenCV module and tried to build my own package from scratch.

At the end, I learned how to get images from IP camera. I used a stream module from OpenCV library to read every byte of the image, then stored it to OpenCV matrix CvMat and converted it to ROS Message using **cv_bridge,** a ROS Module which is able to convert OpenCV images to ROS Messages and vice versa. The chart of this work can be seen in figure 1. To demonstrate it, I opened 3 windows as seen in figure 2: Publisher Node, Subscriber Node and rqt_graph window. Publisher node captured the images from the IP camera, converted the images using **cv_bridge**, and then sent the ROS Image message named "/spacecam_image". Subscriber node received the messages from

publisher node, reconverted it back to OpenCV Image and then showed the OpenCV Image. The last window is rqt_graph, which was showed the relation between those nodes and the name of the message that was sent by the publisher to the subscriber node.

3. Cross-referencing with the work of fellow team members

Nate and Brian finalised the design of new magnetic foots and assembled the whole prototype of the robot. Brian also helped Nate in the wiring, including the wiring of IMU sensor.

I also worked with Nate to fix the IMU reading data from last progress review. Nate found that the last IMU reading code could be catastrophic to the other serial communication since it sent raw data through serial communication. If this program combined with another sensor reading in the Arduino, such as battery monitoring, it would not be able to send those data at the same time. So, instead of sending IMU raw data through serial communication, we created a ROS Node inside the Arduino program and published it through Serial comms. At the end, there are two ROS messages published by the Arduino: IMU data and Battery monitoring data. This messages were read in the PC using ROS Serial node.

Songjie worked with me most of the time to integrate the ROS Image message from my program to read images from the IP Camera. It turned out that the format of the ROS Message that my program published need to be confirmed by Songjie's program. We will keep continue working on that for the next progress review.

4. Goals for following weeks

My goals for the following weeks are: integrating AR Tags packages with the IP Camera reading from this progress review, and working closely with Songjie to calibrate the images and make it useful for Songjie's program to calculate the transformation matrix of the robot.

In order to simplify the work, our team are likely to change method from April Tags to AR Tags, since there are so many AR Tags packages already available from Willow Garage such as: measuring the position and orientation of the AR Tags, and also visualizing it into a virtual 3D workspace.

5. Figure(s)

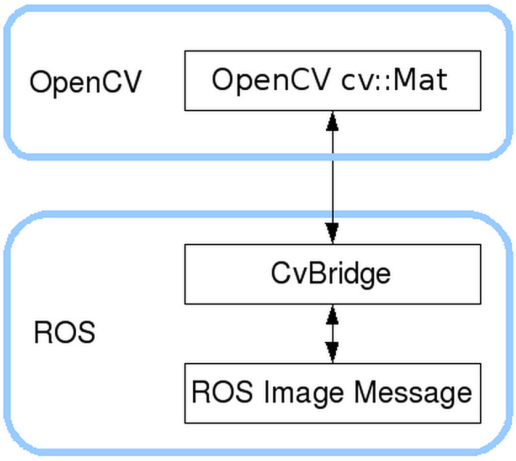

Figure 1 . Chart of CV_Bridge module, which converts OpenCV images (in Matrix named cv::Mat) to ROS Image Message

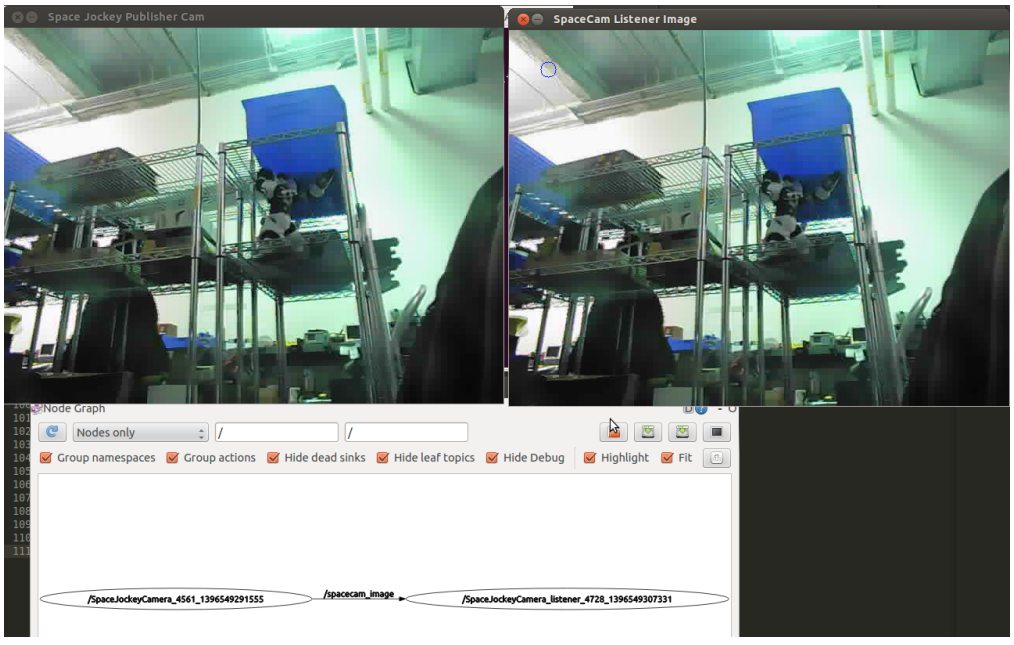

Figure 2. Images from IP Camera are published via “SpaceJockeyCamera” node as ROS image message