Individual Lab Report 11

Name : Ardya Dipta Nandaviri Giri
Team B : Space Jockey
Team Members : Brian Boyle, Nathaniel Chapman, Songjie Zhong
ILR09
Submission Date : April 16, 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 April 3, 2014, I continued working on visualization and localization using IP camera and AR-Tags.

2. Individual Progress & Challenges

My priority for this progress review is making the robot able to detect its location based on the AR Tag detection. From the AR-Tag, the robot shall know the transformation matrix of the camera. I spent most of my time working on building the camera driver for our IP camera to be integrated with “AR-Track-Alvar” module, a built in module in ROS.

My last progress review, I was able to send the images from the IP camera to ROS nodes. As it turns out, images only is not enough to be considered as a complete camera driver. Camera driver in ROS has to have image message and camera_info message, which contains the information about camera calibration and its intrinsic matrix. To get the calibration, I used a tool from ROS to calibrate the camera and a checkerboard to calculate the distance and orientation of the camera. The calibration process can be seen in figure 1. After the calibration, I stored the calibration into a yaml file and then I can generate the camera_info message from IP camera.

One issue I faced was the conversion I used to convert image from OpenCV to ROS Message was using first version of CV library, while Songjie mostly used CV2 library. Hence I have to clean up the code and fix my code so that can be used by Songjie.

Another issue is the camera driver was not read properly by Rviz, which meant it was also unable to read the AR-Tag. After reading a lot of sample cameras driver code that are available in ROS website, I found that each camera_info and image message must have

the same FRAME-ID to be detected by rviz. After setting the image message and camera_info FRAME-ID name to “camera”, the IP camera was finally detected as a camera by Rviz, and then using AR-Track-Alvar built in module, I was able to detect AR-Tag as seen in figure 2. This module also publish a message about the distance and orientation of each AR-Tag.

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

Nate and Brian worked on inverse kinematics code and finishing the new design of magnetic foot. Brian also worked on the calibration of each joint’s servo to ensure that the robot moves smoothly.

I worked with Songjie to integrate my code with his code to build a transformation matrix code. As I mentioned, since Songjie used CV2 library, I must clean up my code and move from CV library to CV2 library. There are some differences between those two library in the terms of methods used.

Nate also helped me a lot with the camera driver for the IP camera. He realized that the output frame of the camera should relate to the world frame, and he also helped me for cleaning up the code with submodules, and then integrate it with most of the existing code.

4. Goals for following weeks

My goals for the SVE demo next week are helping Nate with the localization code and working with Songjie to develop the comparison image. We will debug and clean up all the code as needed so that the robot can walk properly on the wall and detecting flaws.
I also will debug the localization code with Nate based on my AR-Tag detection code from this progress review.

5. Figure(s)

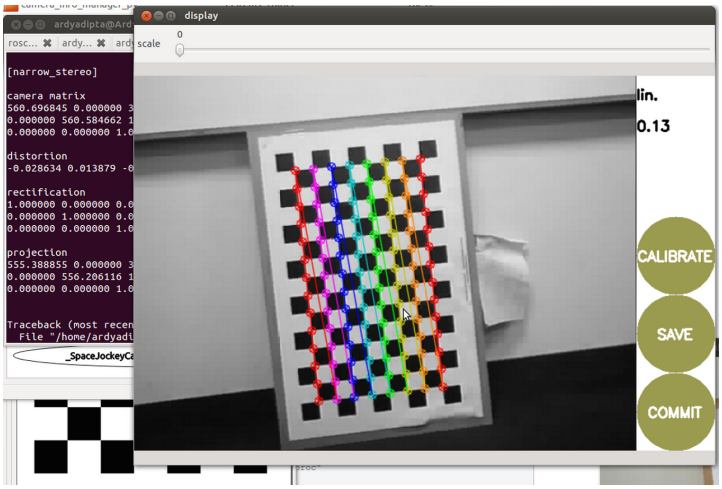

Figure 1. Camera calibration using checkerboard.

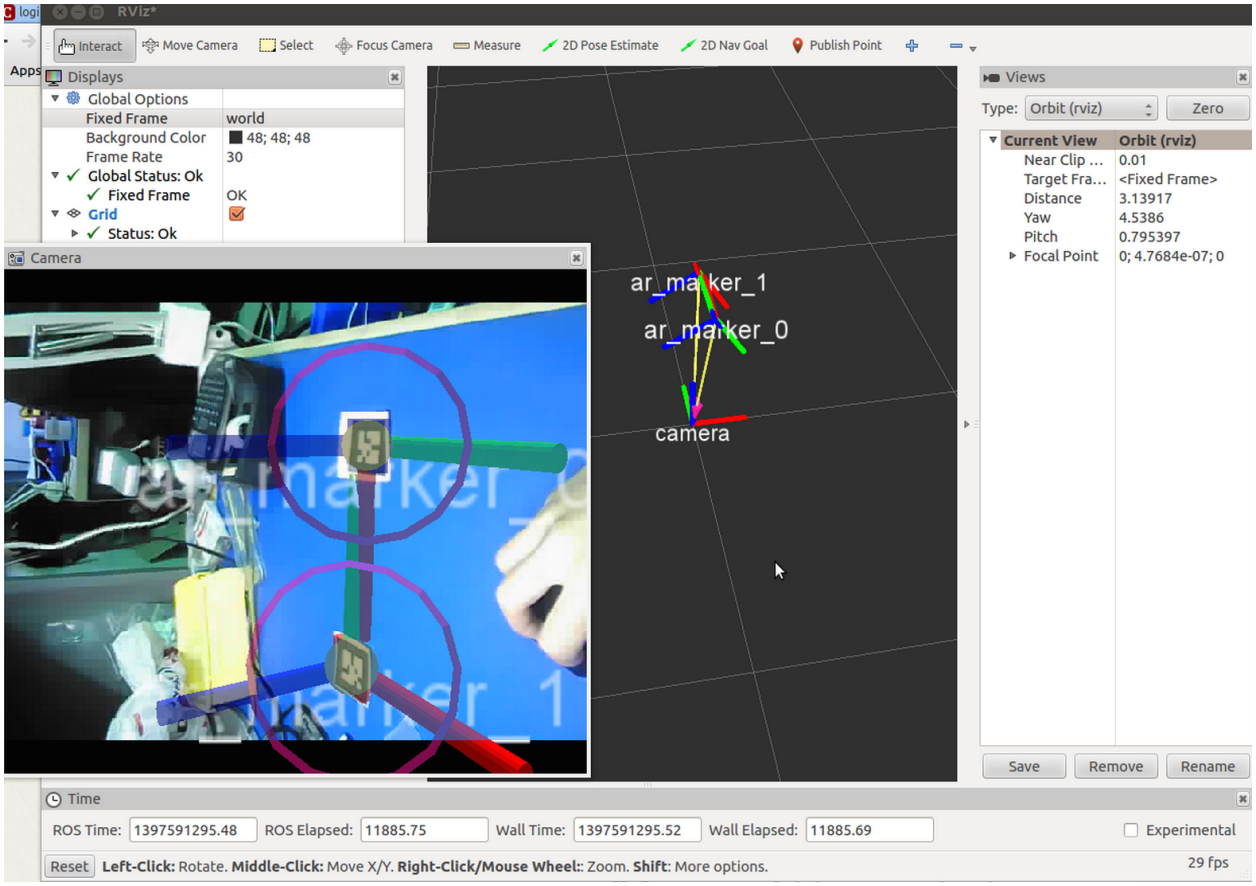

Figure 2. Camera is detecting AR-Tag