RO10005 - Advanced Robotics
Project 2 Phase 1

Lecturer: Prof. Yi Zhou
eeyzhou@hnu.edu.cn
TA1: Jinghang Li
li_jinghang@163.com
TA2: Delei Kong
dl_kong@qq.com

1 Project Work

In Project 2 Phase 1, you need to implement the 3D-2D pose estimation algorithm to estimate the camera's poses using images as input.

1.1 Project Assignments

You are asked to accomplish the following.

1. Calculating the camera's pose at every image.
2. Publishing camera pose information in the form of `nav_msgs/Path`.
3. Plotting these poses with **rqt_rviz**.
4. Comparing your result with the reference signal, which is calculated using OpenCV's PnP function.

1.2 Project Details

You are provided with a ROS package named `tag_detector`, where a number of points and their positions will be calculated within images. You need to implement this project based on the point and position array. You can follow the following procedures for your implementation.

1. Move `aruco-1.2.4` and `tag_detector` into your workspace (`catkin_ws/src/`)
2. Install aruco (following `aruco-1.2.4/README`)
3. Setup your ROS environment and compile the `tag_detector` package.
4. Find `bag_tag.launch` in `tag_detector/launch`, and `images.bag` in `tag_detector/bag`.

5. Use `bag_tag.launch` and `images.bag` to run this package (roslaunch `bag_tag.launch`).
6. Read the comments in `tag_detector/src/tag_detector_node.cpp` **carefully**.
7. Add your code into `tag_detector/src/tag_detector_node.cpp`.
8. Note that the pose you calculated is $(\mathbf{t_{cw}}, \mathbf{R_{cw}})$, which represents the pose of world frame w.r.t to the camera frame.

2 Submission

Please submit your code and documents to hnu_ro10005@yeah.net. The project name for this assignment is titled "`proj2phase1-YOUR NAME-STUDENT ID`".

Your submission should contain:

1. A **maximum 2-page** document including:
   (a) Figures plotted by **rviz**.
   (b) Statistics about your result. (For example, RMS error between the poses you calculate with the reference ones)
   (c) Descriptions about your implementation.
   (d) Any other things we should be aware of.
2. `tag_detector_node.cpp` and any other c++ files that you need to run your code.

3 Basic use of RVIZ

1. Open one terminal and execute

```
roscore
```

2. open another termianl and input

```
rosrun rviz rviz
```

3. Click add button, add your topic.
4. Change the frame to "world".
5. Change the color of your path and the reference path.

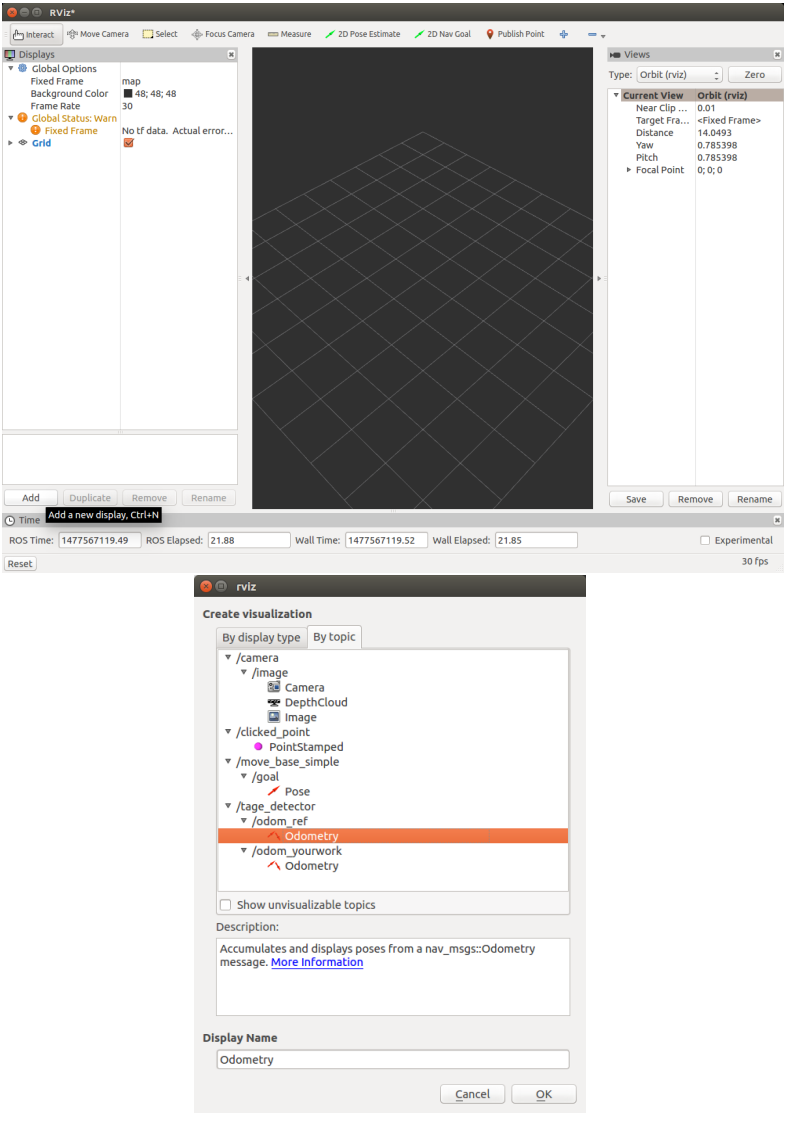

Figure 1: Step 3.

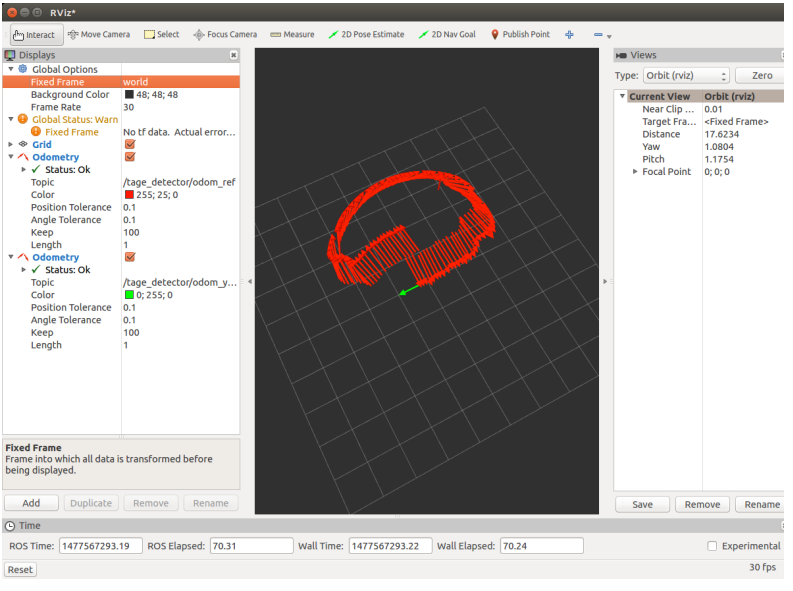

Figure 2: Step 4 and 5.