EE478 HW3

nijatabbasov06

Apr 2025

Table of Contents

1 Problem 1

1.1 Part 1

Checking the realsense_stereo_imu_config.yaml file in the KAIST VIO dataset github, we can see that there is written image0_topic: "/camera/infra1/image_rect_raw" image1_topic: "/camera/infra2/image_rect_raw". Also, in the orb_slam3_ros/config/Stereo/ folder in orb_slam3_ros package, we can see different yaml files for different camera models. For our assignment, we will use RealSense_D345i.yaml file.

Figure 1: Changed topics and camera model

1.2 Part 2

Looking at the config files at KAIST VIO dataset github repo, following lines were added to the orb_slam3_ros/config/Stereo/RealSense_D435i.yaml file:

Figure 2: Additions to the RealSense_D435i.yaml file

1.3 Part 3

Following lines were added to the common.cc to change the rotation matrix accordingly:

```
void publish_camera_pose(Sophus::SE3f Tcw_SE3f, ros::Time msg_time)
{
    geometry_msgs::PoseStamped pose_msg;
    pose_msg.header.frame_id = world_frame_id;
    pose_msg.header.stamp = msg_time;


    Eigen::Matrix3f R_cam_to_enu;
    R_cam_to_enu << 0, 0, 1,
                    -1, 0, 0,
                    0, -1, 0;

    Eigen::Matrix3f Rwc = Tcw_SE3f.rotationMatrix();
    Eigen::Vector3f twc = Tcw_SE3f.translation();

    Eigen::Matrix3f Rwb = Rwc * R_cam_to_enu;
    Eigen::Vector3f twb = twc;


    Eigen::Quaternionf qwb(Rwb);

    pose_msg.pose.position.x = twb.x();
    pose_msg.pose.position.y = twb.y();
    pose_msg.pose.position.z = twb.z();

    pose_msg.pose.orientation.w = qwb.w();
    pose_msg.pose.orientation.x = qwb.x();
    pose_msg.pose.orientation.y = qwb.y();
    pose_msg.pose.orientation.z = qwb.z();

    pose_pub.publish(pose_msg);
}
```

Figure 3: Publishing camera pose

1.4 Part 4

To run the bag file, we first need to run roscore. But as the euroc_stereo.launch file already starts the launch file, we can first run the launch file, and then run the bag file. The launch file opens rviz, map viewer, current frame windows and waits for the necessary topics to publish camera images. Then, we can run the bag file so that those camera images would be published to the topics that the orb_slam3_ros is listening. However, when we launch the both files we can see that there is no orb_slam3_ros/camera_pose topic in the rviz. So we add the topic in rviz because it is asked in the assignment:

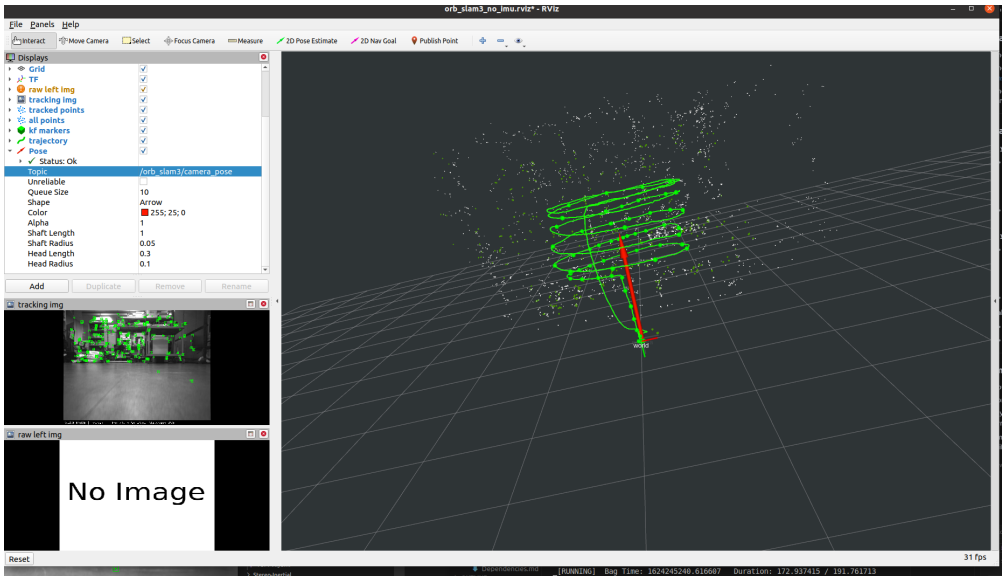

Figure 4: Circle

Here, I am using bag file corresponding to circular motion. The big red arrow is the position odometry for the drone.

2 Problem 2

2.1 Part 1

In this problem, we can realize that we need to compare /pose_transformed (groundtruth data) and /orb_slam3_ros/camera_pose topics. However, running the rqt_bag 'circle (1).bag' file we get the following:

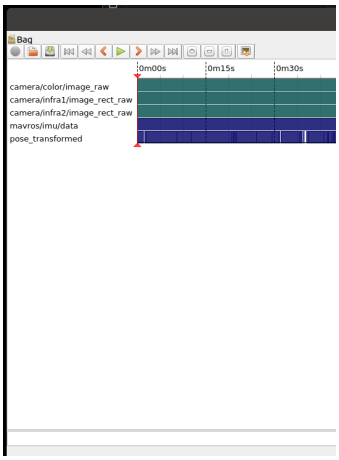

Figure 5: recorded topics for the bag file

So, we realize that we need to record the camera_pose topics ourselves. For this reason, I run the following command:

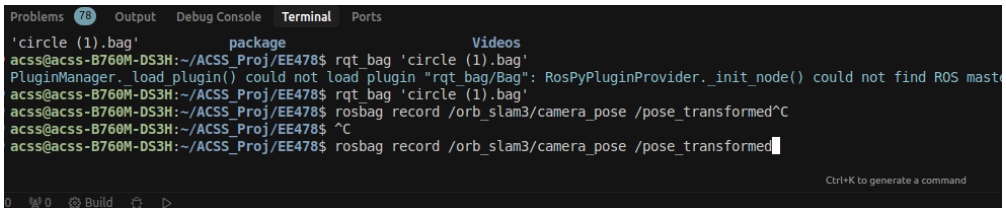

Figure 6: rosbag record

Then, using that file for comparing in evo_traj, I use the command with the following arguments:

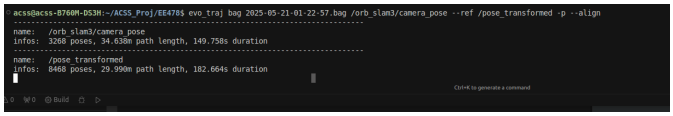

Figure 7: evo_traj arguments

Here, -p is used for plotting, –align is for telling evo_traj that I want the trajectories to be aligned. –ref is used for telling the evo_traj that the groundtruth is the topic that comes right after –ref so that the evo_traj can use that ground truth topic for aligning the trajectories. Following is the resulting plot:

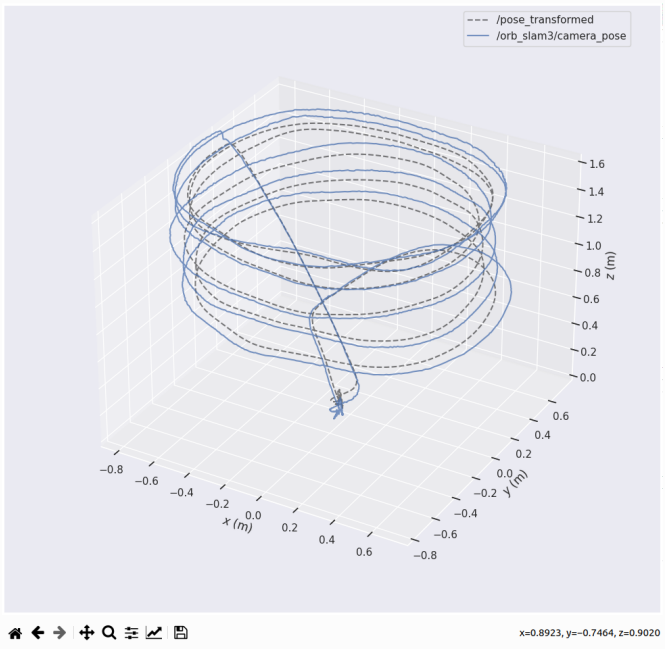

Figure 8: trajectories

2.2 Part 2

```
# ORB Parameters
#--------------------------------------------------------------------------------------------
# ORB Extractor: Number of features per image
ORBextractor.nFeatures: 1250

# ORB Extractor: Scale factor between levels in the scale pyramid
ORBextractor.scaleFactor: 1.2

# ORB Extractor: Number of levels in the scale pyramid
ORBextractor.nLevels: 8

# ORB Extractor: Fast threshold
# Image is divided in a grid. At each cell FAST are extracted imposing a minimum response.
# Firstly we impose iniThFAST. If no corners are detected we impose a lower value minThFAST
# You can lower these values if your images have low contrast
ORBextractor.iniThFAST: 20
ORBextractor.minThFAST: 7

#--------------------------------------------------------------------------------------------
# Viewer Parameters
#--------------------------------------------------------------------------------------------
Viewer.KeyFrameSize: 0.05
Viewer.KeyFrameLineWidth: 1.0
Viewer.GraphLineWidth: 0.9
Viewer.PointSize: 2.0
Viewer.CameraSize: 0.08
Viewer.CameraLineWidth: 3.0
Viewer.ViewpointX: 0.0
Viewer.ViewpointY: -0.7
Viewer.ViewpointZ: -3.5
Viewer.ViewpointF: 500.0
```

Figure 9: Parameters

Here, we can see that there are various parameters about the orb slam. The nFeatures parameter means how many features to extract from the image, we can see that, when we increase the feature amounts, the green squares in the image increases. While increasing feature numbers can help the slam do localization much faster and more accurate, it is also computationally expensive.

The followings are the examples of running the bag file with 2000 nFeatures parameter. We can see that, the green suqares in the image is much more than the one with 1250 features:

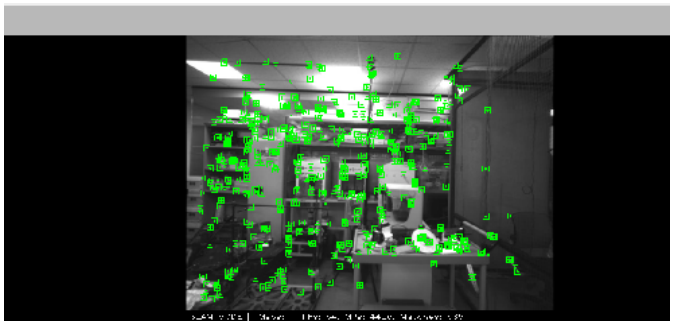

Figure 10: 2000 nFeatures

However, we cannot see much difference between the reference pose and the actual pose.

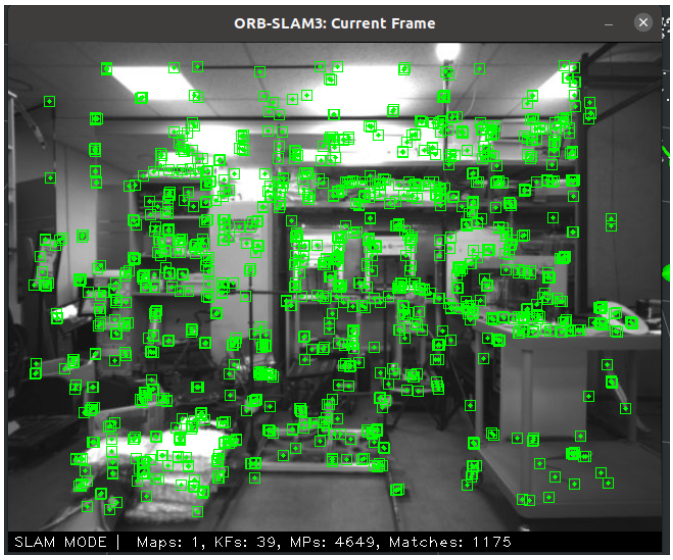

Figure 11: 2500 features

We can see that increasing the features to 2500, the features increase more.

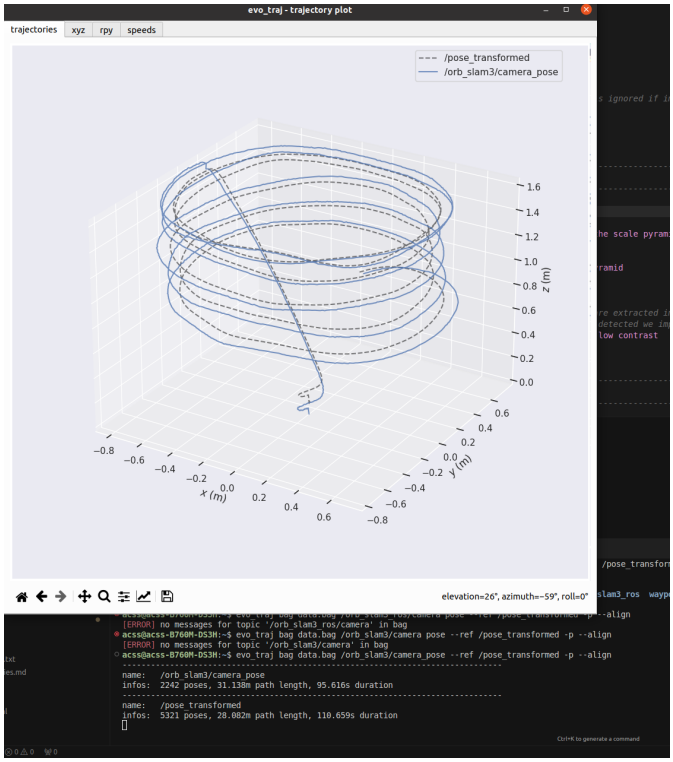

Figure 12: Trajectory

and the following is the trajectory comparison:

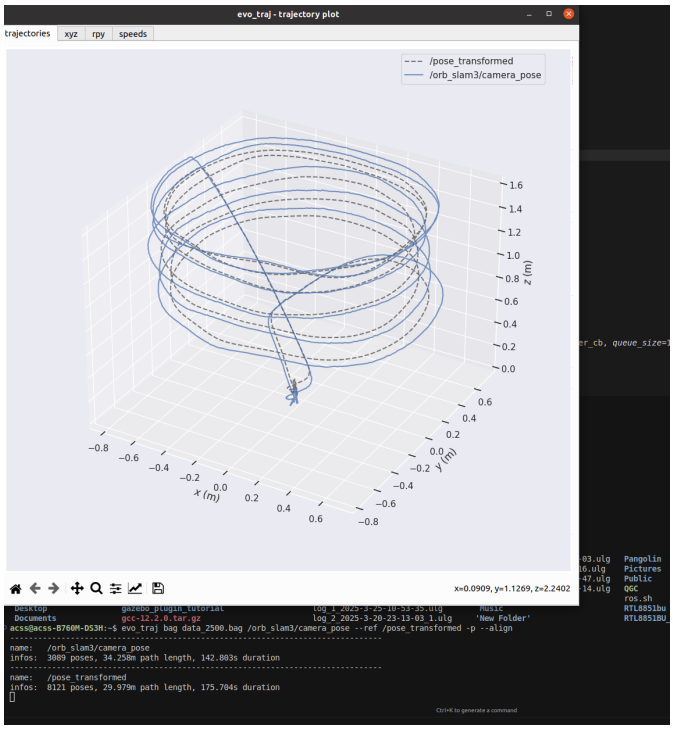

Figure 13: Trajectory comparison

3 Problem 3

3.1 Part 1

Using the christmas_tree.launch file in the launch directory of the px4 repository, I am using the following arguments for world and sdf arguments:

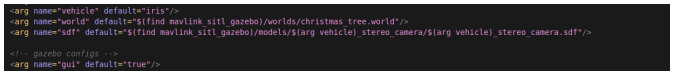

Figure 14: world and sdf arguments

To find the proper image topic needed for slam, we run the launch file, and in the new terminal we write rostopic list. There we can see that there are /stereo/left/image_raw and /stereo/right/image_raw topics. So, we use those topics in the euroc_stereo.launch:

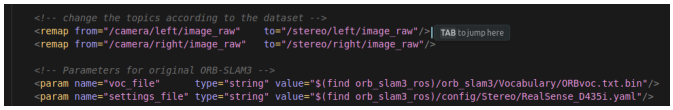

Figure 15: image topics

Then, we change the subscribed topic in the waypoint_mission/waypoint_mission.py :

Figure 16: waypoint mission.py

3.2 Part 2

For this part I gave the following positions and the following yaw list for the drone to follow:

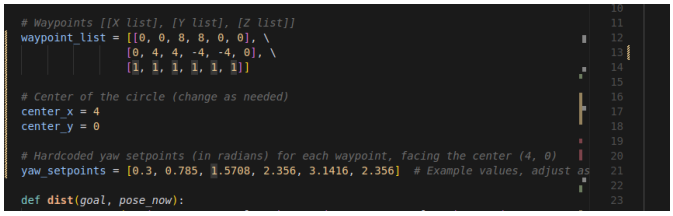

Figure 17: Drone setpoints

4 Reference

[1] EE478 KAIST lecture notes [2]