ELEC 5660: Introduction to Aerial Robotics
Project 2: Phase 2

Assigned: Mar. 26th, 2024 Due: 11:59 PM Apr. 12th, 2024

1 Project Work

In Project 2 Phase 2, you are required to implement the stereo visual odometry(VO). It involves several components you have learned in the lecture, including feature detection, feature tracking, 3D point generation and PnP-based pose estimation. You will be provided with a ROS package named `stereo_vo` which contains the skeleton code. Your task is to finish each major component within this skeleton to form a complete stereo VO.

Note:

(1) This project is rather complex, requiring several key functions to be implemented. DO NOT think you can finish it in one day unless you are very familiar with the details of VO, **PLEASE START AS EARLY AS POSSIBLE**.

(2) Please read the comments of the code carefully. **You are strongly recommended to follow the predefined structure / variables / functions** to implement the VO. Modifications of them may lead to direct deduction of points without checking the code if bugs occur.

2 Tutorial

The main functions you need to implement are contained in `stereo_vo_estimator/src/estimator.cpp`. Please follow the TODO comments and the following tutorial. You can also modify some of the parameters in `stereo_vo_estimator/config/realsense_1/realsense_n3_unsync.yaml` to tune the performance of the VO. After the compilation, use `stereo_vo_estimator/config/launch/stereo_vo_bag.launch` and `stereo_vo_estimator/bag/realsense_1.bag` to run the package. A screen-shot of the visualization when running the stereo VO with the provided bag is shown in Figure 1.

2.1 Main Procedure

The main processing function is the *inputImage* function. You are required to complete the whole processing procedure including feature detection, feature tracking, 3D point generation and PnP-based relative pose estimation. At the end of the *inputImage* function, the states of the current frame is required to be updated in the *updateLatestStates* function, which is required to be implemented. The position and the orientation are required to be transformed into the world frame (the IMU frame at initialization).

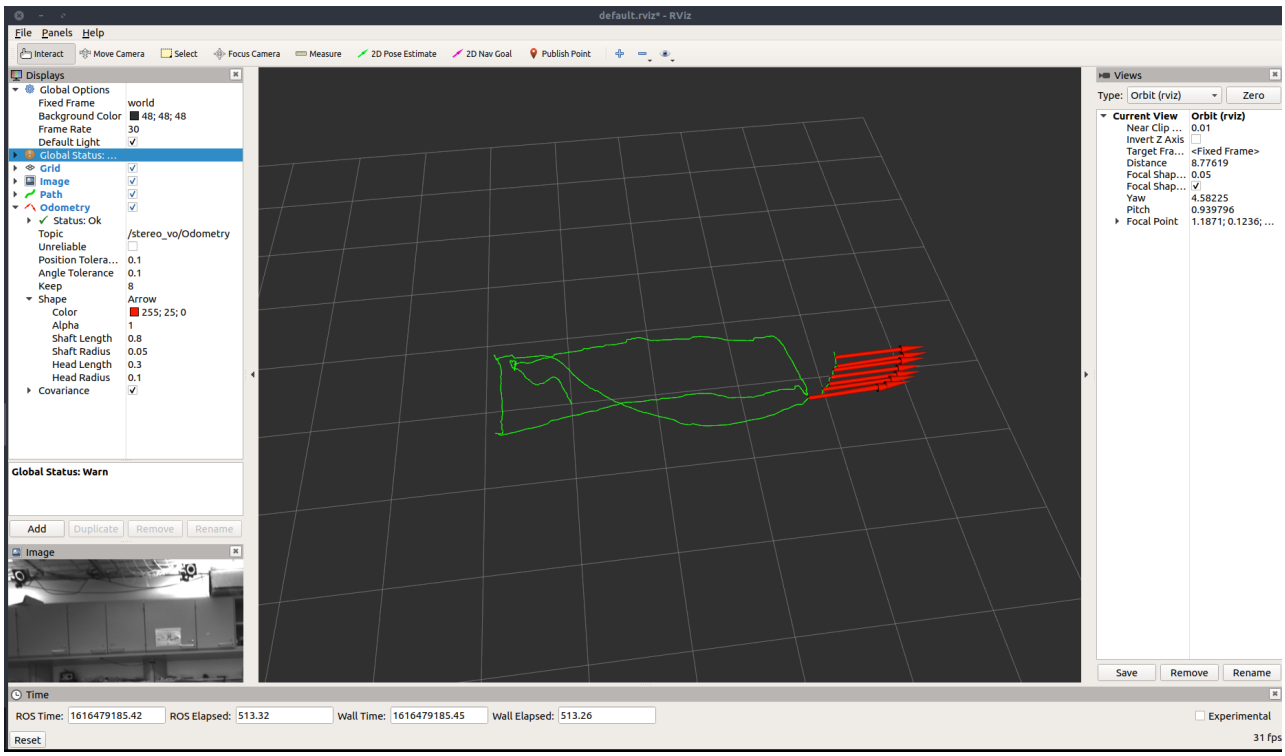

Figure 1: The screen-shot of the visualization when running the stereo VO with the `realsense_1.bag`. The axes indicate the origin of the world frame, while the arrows indicate the estimated odometry, and the green curve indicates the estimated trajectory.

2.2 Feature Detection

You are required to complete the function *extractNewFeature* to detect features that will be used for relative motion estimation. You may simply use the *goodFeatureToTrack* function in OpenCV.

2.3 Feature Tracking

In functions *trackFeaturesLeftRight* and *trackFeaturesBetweenFrames*, you need to implement the feature tracking. In *trackFeaturesLeftRight*, it finds features in the right camera image that associates with those in the left camera image. In *trackFeaturesBetweenFrames*, it finds features in the current left image corresponding to those in the keyframe. You can track the features using the LK optical flow. These two steps are essential for performing 3D point generation(Sect.2.4) and PnP-based pose estimation(Sect.2.5).

2.4 3D Point Generation

Given the matched feature pairs in the left and right image from the function *trackFeaturesLeftRight*, the 3D feature points are able to be computed. The function *generate3dPoints* is already provided, which computes the 3D position represented in the left camera frame for each undistorted pair of feature points. The *undistortedPts* function is also provided.

2.5 PnP-based Relative Pose Estimation

Given the 3D feature points of the keyframe and the associated 2D features in the current frame, the function estimateTBetweenFrames is required to be implemented to calculate the relative transformation between the keyframe and current frame. You can implement it using the *solvePnP* or *solvePnPRansac* in OpenCV. Note that the relative transform is in the camera's frame.

3 Submission

When you complete the tasks you could submit your code and documents to canvas before **Apr. 12th, 2024 23:59:00**. The title of your submission should be "`proj2phase2_YOUR-NAMES`".

Please cite the paper, GitHub repo, or code url if you use or reference the code online. Please keep academic integrity. **Plagiarism** is not tolerated in this course.

Your submission should contain:

1. A **maximum 2-page** document including:
   (a) Figures of the estimated path of the camera plotted by **rviz**.
   (b) Descriptions about your implementation of each key component.
   (c) Any other things we should be aware of.
2. The folder `stereo_vo_estimator` and all files you need to run your code **except for the bag. Uploading the bag will cause the deduction of points**.

You will be graded on successful completion of the code and the performance of your stereo VO.