ELEC 5660: Introduction to Aerial Robotics
Project 2: Phase 1

Assigned: Mar. 26th, 2019 Due: Apr. 9th, 2019 (two weeks)

1 Project Work

In Project 2 Phase 1, you need to implement the algorithm on Lecture 6 to give an velocity estimator based on LK optical flow algorithm. This is an individual project, which means you must complete it by yourself.

1.1 Project Assignments

Two assignments are required for this project including:

1. Implement velocity estimator based on LK optical flow algorithm.
2. Compare your result with the ground truth.

1.2 Project Contents

You will be provided with a ROS package named `optical_flow`, and are required to complete both 2 TODO parts. It implements the velocity estimator based on LK optical flow algorithm with the help of the below subscriptions. The sensors are just like the UAV you assembled in the lab course.

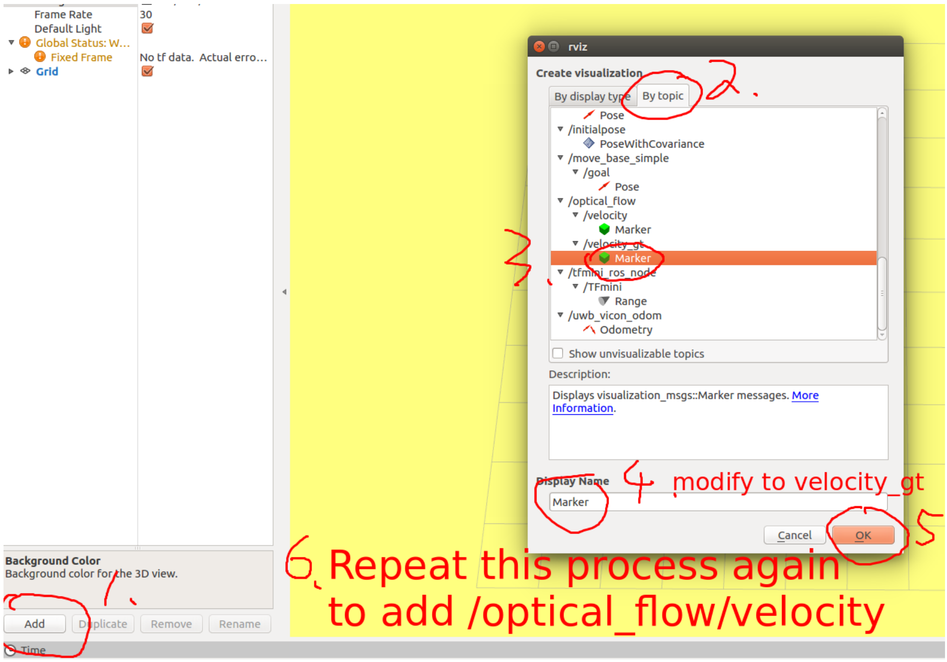
1. **/tfmini_ros_node/TFmini**, as the height information provided by the ToF sensor.
2. **/camera/image_raw**, as the images used for the LK algorithm, which are pictures of the birdview.
3. **/uwb_vicon_odom**, as the ground truth.

A rosbag named `optical_flow.bag` is also provided to you, contains the above subscriptions to test your code. The bag file can be obtained in the link below.
`https://www.dropbox.com/s/dm2ywn5z56bzmth/optical_flow.bag?dl=0`

1.3 Project Details

You should learn **ROS**(Robot Operating System) before starting this project. At least, basic usage of **RViz** & **rosbag** is required. And basic usage of **Eigen** & **OpenCV** is also required.

You are supposed by finish this project step by step as follows, and **the code** means `optical_flow_node.cpp` in the package.



Frame Rate 30

Default Light ☑

Fixed Frame No tf data. Actual erro...

Grid ☑

Background Color
Background color for the 3D view.

□ Show unvisualizable topics

Description:
Displays visualization_msgs::Marker messages. More Information.

Display Name: Marker

4 modify to velocity_gt

6. Repeat this process again to add /optical_flow/velocity

Figure 1: Usage of rviz

1. Run the framework before coding

   (a) Learn ROS(at least usage of RViz & rosbag), and some functions related to this project from OpenCV & Eigen

   (b) Set up your ROS environment, compile the `optical_flow` package, and rosrun it.

   ```
   mkdir -p ~/catkin_ws/src
   move the optical_flow package into ~/catkin_ws/src
   cd ~/catkin_ws && catkin_make
   source devel/setup.sh
   rosrun optical_flow optical_flow
   ```

   (c) Open your RViz by running `rviz` instruction and add visulization(see Figure 1) of two topics `/optical_flow/velocity` & `/optical_flow/velocity_gt`

   (d) Play the given rosbag file. You should see the ground truth result in the RViz.

   ```
   rosbag run optical_flow.bag
   ```

2. Implement the velocity estimator in the **TODO 1** part in the code. There are two different implementations shown below. **KEEP IN MIND THAT THE (a) IMPLEMENTATION IS ALWAYS REQUIRED**, because it will be the version on-board. You should give options to use (a) or (b), or put them into 2 seperate files.

   (a) (3pts) Use LK optical flow algorithm provided by OpenCV to implement the velocity estimator. You are supposed to use uniform feature (i.e. without using feature extraction methods of OpenCV), in order to use less computing resourse on-board.

For this part, you can assume the UAV is flying slowly, which means height changes slowly and orientation seldom changes.

(b) (2pts) Implement PyramidLK by yourself, follow the same assumption as (a).

Good filters are neccessary to remove outlier. Possible choices are consistency check or RANSAC, which will covers in Lecture 7.

3. (1pt) Caluclate the RMS Error between the velocities you calculate with the grount truth in the **TODO 2** part in the code, and show the result in the report.

2 Submission

When you complete the tasks you could submit your code and documents to canvas before **Apr. 9th, 2019 23:59:59**. The title of your submission should be "`proj2phase1_YOUR-NAMES`".

Your submission should contain:

1. A **maximum 2-page** document including (1pt):

    (a) Figures plotted by **rviz**.

    (b) Statistics about your result, such as RMS error between the velocities you calculate with the grount truth.

    (c) Descriptions about your implementation and filters to remove outlier.

    (d) Any other things we should be aware of. You should describe the way to choose from using (a) and (b).

2. Files `optical_flow_node.cpp`, as well as any other c++ files you need to run your code.