INDIVIDUAL LAB REPORT 3

Progress Review - 2

Shubham Garg
TEAM H (PHOENIX)

Teammates:
Akshit, Parv, Steve

March 8, 2019

Individual Contribution:

1. Understanding of ORB-SLAM2 pipeline and it's ROS library
2. Setting up ORB-SLAM2 on Nvidia Jetson TX2

Understanding of ORB-SLAM2 pipeline and it's ROS library:

ORB-SLAM2 is a feature based SLAM method which has three major components: tracking, local mapping and loop closing. For initializing the map, the relative pose between two scenes are computed, one for the planar scene & other for the non-planar scene. Based on the relative score of both one of them is selected. This is estimated for multiple motion hypotheses to check if it is significantly better then the other and if it's better global bundle adjustment is done.

In tracking, there are two things done: one is the camera localization and other is the keyframe insertion decision. Features are matched with the previous frame and pose is optimized using motion only bundle adjustment. If tracking is lost, the place recognition module kicks in and tried to re-localize itself. Matches of the local map are searched on the frame and camera pose is optimized using these matches. Finally, a new keyframe is inserted when at least 20 frames have passed from the last frame.

New map points are created by triangulating the ORB features from the connected keyframes in the covisibility graph. The matches must fulfill epipolar geometry criteria. New map points are only considered if it's found in more than 25 percent of the frames in which it is predicted to be visible. Then local BA is applied on the current keyframe and all the other keyframes connected through the covisibility graph.

For loop closure, the bag of words of the current keyframe and its neighbor is checked in the covisibility graph. If the similarity between the current keyframe is greater than the minimum similarity of these bag of words, then all the keyframes associated with the current frame are removed. If there are three consecutive loop detection, then it is considered for the loop closure. Further, RANSAC is applied to these loops to optimize the pose of the camera. Then the current keyframe pose is adjusted and it is propagated to its neighbor and the corresponding map points are fused.

Setting up ORB-SLAM2 on Nvidia Jetson TX2:

ORB-SLAM2 has multiple dependencies on other ROS libraries which includes Pangolin, OpenCV, Eigen3, DBoW2, and g2o. Pangolin library is used for the visualization and user interface. OpenCV is used for image manipulation and feature extraction. Eigen3 library for performing mathematical operations on the Matrices. Finally, DBoW2 is an

improved version of the DBow library, for indexing and converting images into a bag-of-word representation. It implements a hierarchical tree for approximating nearest neighbors in the image feature space and creating a visual vocabulary. It also implements an image database with inverted and direct files to index images and enabling quick queries and feature comparisons. g2o is C++ library for optimizing graph-based nonlinear error functions. This helps in solving the global BA problem in ORB-SLAM2.

1. These dependencies are being added in the makefile of the ORB-SLAM ROS library.
2. I created new yaml file with the ZED sensor calibration parameters for different possible resolutions.
3. Remapping of the left and right camera frame output from ZED sensor to the required topics for ORB-SLAM2. A new ROS node is created for these ROS topics remapping.
4. Finally, a new launch file is created for running the ZED camera sensor node, ORB-SLAM2 node, and loading ORB vocabulary file.

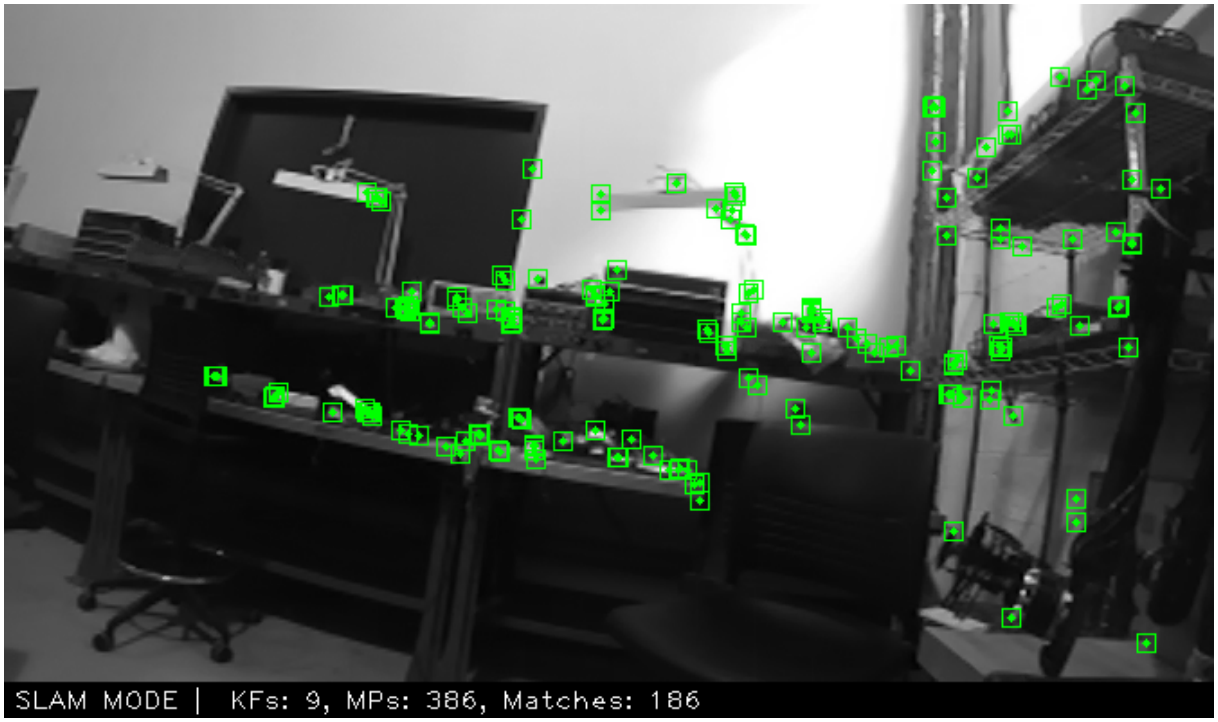

Figure 1: ORB Key points from the feature extraction layer

For creating yaml file, ZED camera is recalibrated and the intrinsic matrix is added in the camera calibration and distortion parameters section. Since the ZED camera sensor provides the rectified images, changes in the stereo rectification parameters were not required. Also, I kept the ORB and the viewer parameters the same as the default.

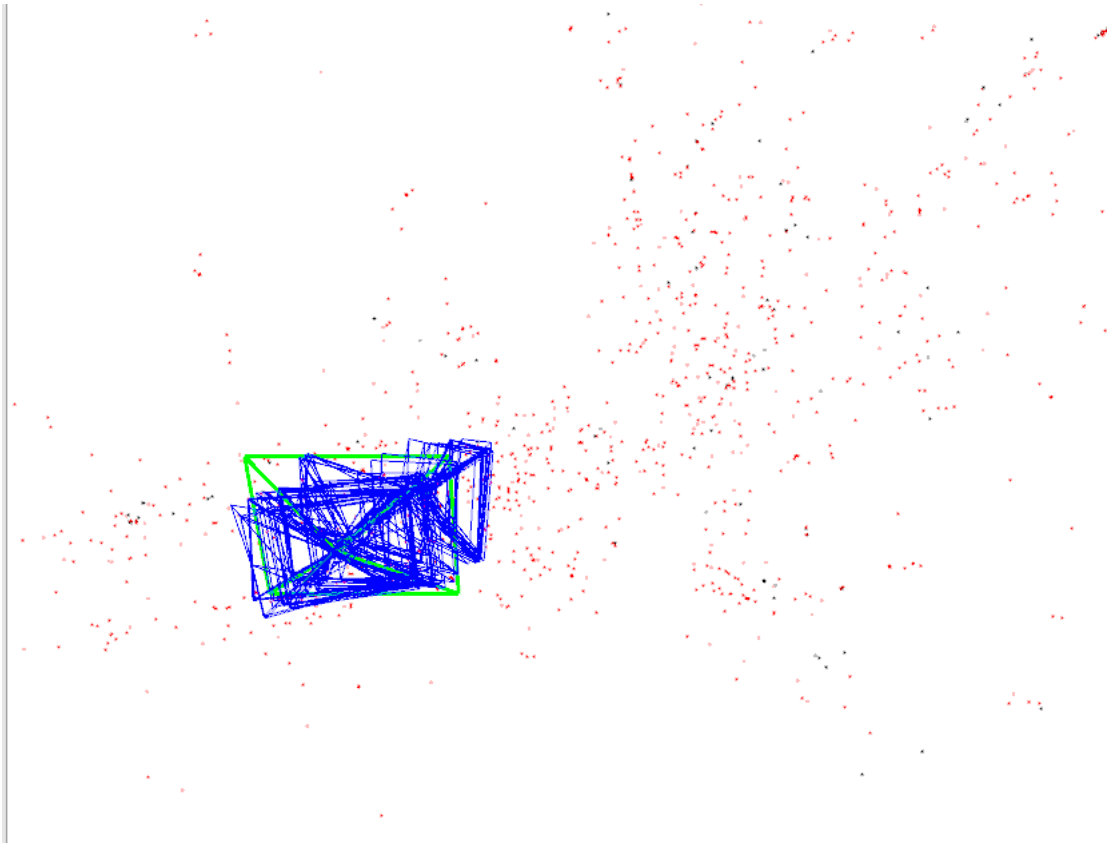

Figure 2: Sparse map generated by the ORB SLAM2

Challenges:

Challenges faced in the last two weeks are discussed below:

1. There are a number of dependencies for the ORB-SLAM2 which needs to be installed separately.
2. ZED sensor output published data on different ROS topic. So, I need to write a ROS node for remapping the same.
3. When I tried running the SLAM with default yaml parameters, the keyframes were not getting initialized and then I further changed the camera calibration and other ZED camera-specific parameters.
4. Similarly, when I tried running the SLAM with higher resolution images, keyframes were not getting initialized. Initially, suspicion was that the ORB SLAM is not able to process the data at high fps. So, I tried changing the fps of the camera from 30 to 15 fps and further to 1 fps but still the keyframe initialization was not working as expected. To further investigate the ORB SLAM-ZED sensor pipeline, we created a ROSbag to record all the images from the ZED camera and then it's played back to see if ORB is facing difficulty in fetching the images from the ROS bag. But still, similar issues were observed.

Further, the ROS publish rate from ZED camera was also verified. Above tests rules out the ZED camera frame rate latency issue or ORB SLAM processing latency. As of now, it looks that feature matching is not working because of the default vocabulary tree (which is built for VGA resolution). In case if the localization accuracy is poor with the VGA resolution, we are planning to create our own vocabulary from the custom dataset.

Teamwork

Akshit & **Steve** worked on thermal camera calibration and installing the thermal camera drivers. Steve fetched the thermal image frames from the camera and Akshit tried various segmentation algorithm on those images to capture the high-intensity region.
Parv: was responsible for setting up the ORB-SLAM2 on the husky and creating real-time map.
Shubham: Installing and setting up the ORB-SLAM2 on the UAV. **Akshit** helped me in remapping the ZED sensor output (left and right ZED camera frame) for the ORB-SLAM2. **Parv** helped me in solving the reinitialization problem in ORB-SLAM2 with VGA resolution. Our team also tested the flight outdoor using onboard GPS and the UAV was stable in the position hold mode with onboard GPS.

Future plans

Parv will responsible for designing the robotics arm for AGV (Husky) and mounting the same.
Steve will primarily work on interfacing (setting up the link between) microcontroller with TX2 and Zotac PC for controlling the manipulator arm and water deploying mechanism.
I will work with Parv in fusing the odometry data from IMU (both Pixhawk and UD7 IMY) with the SLAM output and will integrate it in the state machine. We will try to keep the estimation code same for both the UAV and AGV platform.
We also have a plan of building extra UAV platform, if our ordered parts arrived on time.