Yu-Ting Chung

Final Report

I. Description

This project is to develop a SLAM algorithm running on Raspberry Pi with camera. The SLAM algorithm does both mapping and localization (location and orientation) at the same time using only a camera. In addition, colored landmarks can be recognized during navigation.

II. Goal

- Visual SLAM algorithm for tracking location and orientation on Raspberry Pi
- Recognition of colored landmarks
- Motion tracking with ceiling cameras
- Documentation
- Experiment Videos

III. ORB-SLAM System

The overview of system is as below:

1. Tracking
    - Extracting ORB features
    - Pose estimation
    - Tracking map
    - Deciding key frame
2. Local mapping
    - Inserting map points
    - Covisibility graph
    - Reconstruction the surroundings
3. Loop closing
    - Searching for loops of new keyframe
    - Graph optimization
    - Essential Graph

ORB-SLAM uses Essential Graph for displaying the map. Essential Graph is built from covisibility graph, which is a graph that every keyframe shows as a node, and there is an edge between two nodes if they share the same map points. In Essential graph, only the edges with high covisibility from covisibility graph are contained. In the graph, the red and black points are both map points (corresponding to green boxes in images), where red points are current map points; blue frames are keyframes, which are selected frames for estimating the map; the green line is the trajectory of the camera.

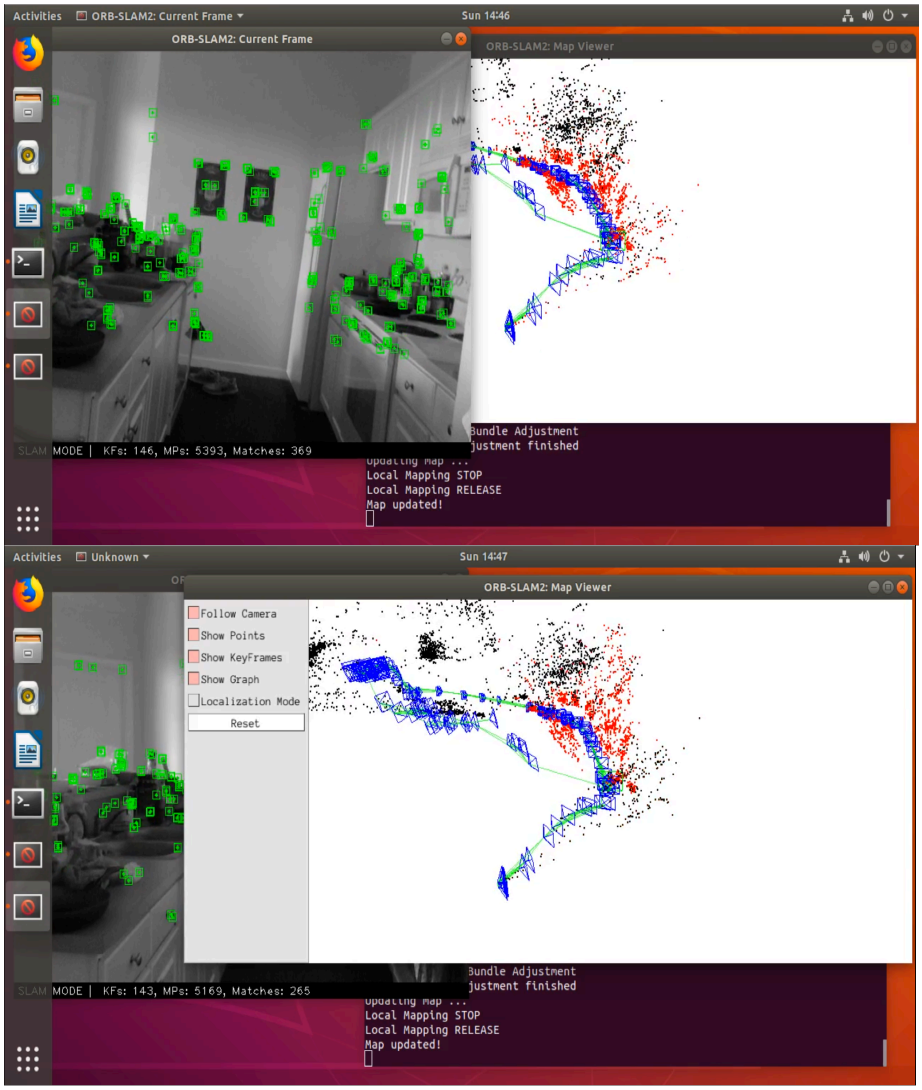
The function of each dependency:

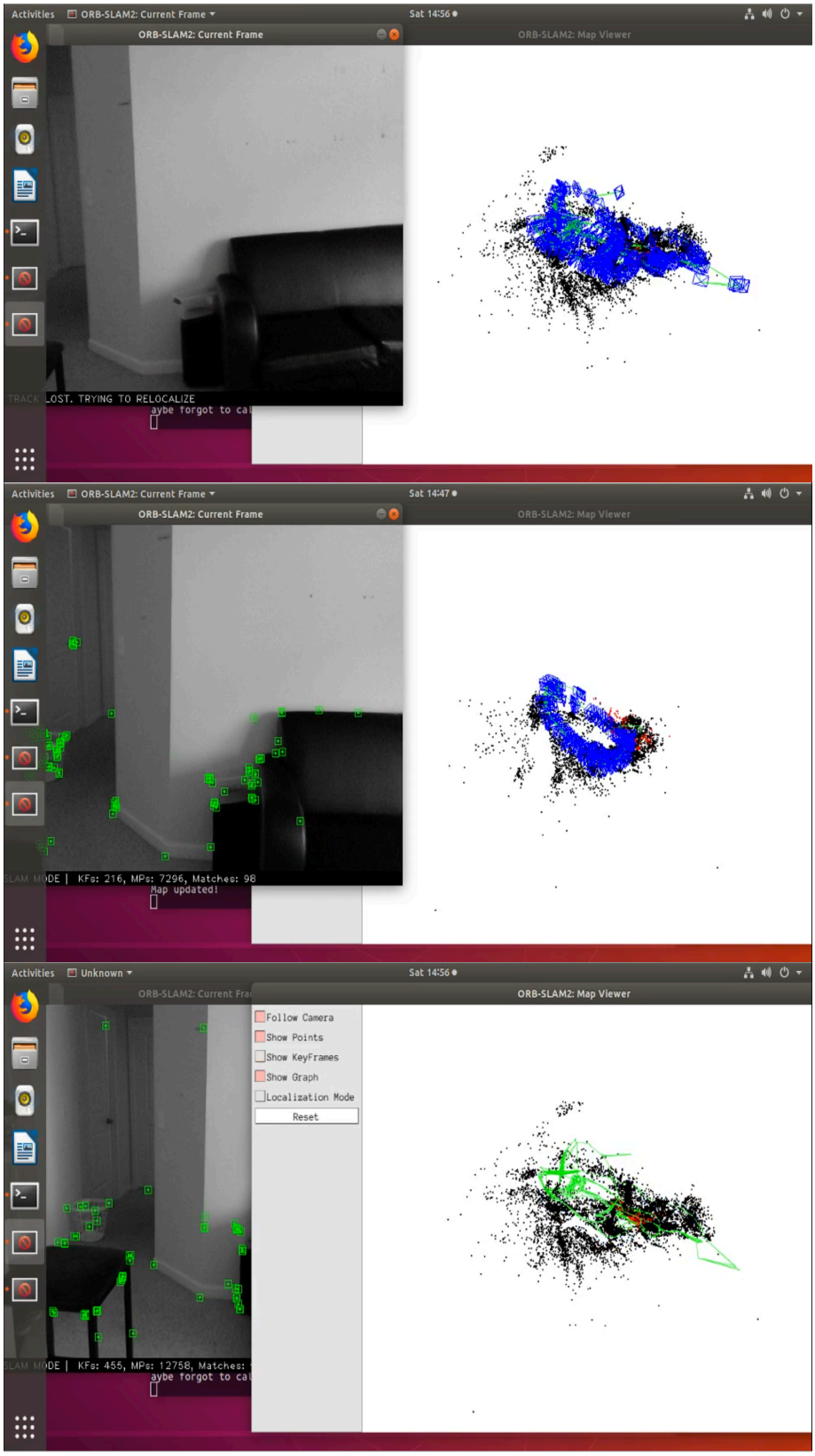
1. OpenCV – Image processing
2. Pangolin – Visualization and interface of map
3. DBoW2 – Feature recognition
4. g2o – Map optimization

Reference1
Reference2
Reference3
Reference4

IV. Result

Mapping and camera trajectory computing could be running in real time. Features in images were recognized and map generated correctly.





Note: Green boxes in images are ORB features, and detailed description of map is in part III.

V. Conclusion

After researching, I decided to code in C++ instead of Java because there are more powerful image processing libraries written in C++, such as OpenCV. There are several reasons of making decision to use ORB-SLAM2 library, including:

1. There are many resources online
2. It is able to use on a monocular camera
3. It was already used with Raspberry Pi
4. ROS is optional

SLAM algorithm works properly, but some goals are not met because image processing are complicated than expecting. I spent weeks to study how OpenCV processes live stream from camera, what parameters ORB-SLAM2 needed to track features, and so on. However, I think computer vision is an important topic for robotics since it makes robots more powerful and useful.

VI. Discussion

During the experiments, I found there are a few of issues as below:

- Features could not be detected if the light is too dim or bright.
- High computing and power consumption for running visual SLAM.

With RGB-D camera or adding a distance sensor to monocular camera, localization and mapping can be more complete and accurate. Computation of images is not easy to improve by current technique.