Visual Simultaneous Localization and Mapping

Introduction:

SLAM is an abbreviation for simultaneous localization and mapping, which is a technique for estimating sensor motion and reconstructing structure in an unknown environment. SLAM using cameras is referred to as visual SLAM (VSLAM) because it is based on visual information only. VSLAM refers to the problem of using images, as the only source of external information, in order to establish the position of a robot, a vehicle, or a moving camera in an environment, and at the same time, construct a representation of the explored zone. The computer vision techniques employed in visual SLAM, such as detection, description and matching of salient features, image recognition and retrieval, among others. VSLAM can be used as a fundamental technology for various types of applications and has been discussed in the field of computer vision, augmented reality, and robotics.

This report aims to categorize and summarize a comparison of various datasets and the issues we faced during the installation of ORBSLAM3. We focus on ORBSLAM3 algorithms to compare these datasets.

Datasets:

We used the following data set to compare the results:

- **EuRoC dataset (Stereo/Monocular, With/Without IMU).**
  EuRoC MAV is a visual-inertial datasets collected on-board a Micro Aerial Vehicle (MAV). The datasets contain synchronized stereo images, IMU measurements and accurate ground truth. The datasets facilitate the design and evaluation of visual-inertial localization algorithms on real flight data. It was collected in an industrial environment and contains millimeter accurate position ground truth from a laser tracking system
- **TUM VI dataset (Fisheye Stereo/Monocular, With/Without IMU).**
  The TUM VI benchmark, a novel dataset with a diverse set of sequences in different scenes for evaluating VI odometry. It provides camera images with 1024x1024 resolution at 20Hz, high dynamic range and photometric calibration. An IMU measures accelerations and angular velocities on 3 axes at 200Hz, while the cameras and IMU sensors are time-synchronized in hardware. For trajectory evaluation, we also provide accurate pose ground truth from a motion capture system at high frequency (120Hz) at the start and end of the sequences which we accurately aligned with the camera and IMU measurements.
- **KITTI dataset**
  KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) is one of the most popular datasets for use in mobile robotics and autonomous driving. It consists of hours of traffic scenarios recorded with a variety of sensor modalities, including high-resolution RGB, grayscale stereo cameras, and a 3D laser scanner.

- **NEU Dataset (car_IR_RGB_lidar)**
  In this dataset, Northeastern's autonomous car (NUANCE) is driven manually along the streets of Newbury Street in Boston. The dataset has stereo RGB cameras looking forward, IR camera looking forward, 2 Velodyne VLP-16 lidar mounted on top of the car, IMU, GPS. The main focus in this dataset was to collect camera data with at least one loop closures. The sensors like lidars, GPS, IMU in combination can serve as ground truth for visual slam algorithms**.**

Issues Faced During ORB SLAM Install and Corrective Measures:

We tried getting ORB SLAM 3 installed on the virtual machines on which we already have ROS installed, installing new virtual machines on which to install ORB SLAM 3, using Docker, using forked repositories of ORB SLAM 3, and changing versions of C++ and other installed dependencies. We realized that one major issue with our attempts to install the software was the limitations of the virtual machines we were using. Ultimately, we had to get it working on a dual boot machine to ensure we had enough memory to install and run the software.

Once we'd eliminated the memory allocation issues, we faced difficulties with different packages in ORB_SLAM using different versions of c++ compiler. Eventually, we successfully built ORB_SLAM in c++ 14, after changing all references to c++ 11 in example files to c++14 to eliminate syntax issues. This was implemented on two of our machines.

We were able to get ORB_SLAM and ROS working together on only one computer. As the NUance dataset was provided in rosbag form, we focused the ROS computer on running the NUance data, and the other computer on running the example datasets provided.

For building the ORB_SLAM3 library with ROS, we faced a lot of version and path issues at first. After a lot of permutations and combinations and going through the documentation so as to what versions will work perfectly (or atleast get built without compilation errors), we finally got it working with Eigen version 3.3.0 and OpenCV version 4.4.0.

Some other issues that we faced during this process was we needed to add the ROS_PACKAGE_PATH to source file which we weren't doing at first (should have been obvious). Also, there were some segmentation fault issues which we corrected by assigning null pointer to certain variables.

The system on which we got it working with ROS was very old and frequently crashed while building. So, the command make –j at all places was replaced by make –j2 which used less amount of processes and so even if the process took a little longer, we were successfully able to build the library with ROS.

Issues when working with NEU dataset:

While working on NEU dataset (car_IR_RGB_lidar (IR camera, Stereo RGB, 2 Lidars, IMU, GPS) there were issues regarding camera_info messages and feature detection. One of the issues with car_ir_rgb_lidar data set was, loss of features on sharp turns. These feature loss leads to failure in tracking local map resulting in the resetting of map. Also, the features which concentrated on distant objects leads to distorted odometry data and the features located on moving objects leads to distortions in map.

Corrective Measures:

We utilize the IR camera data which highly compensated for motion blur and provided accurate feature tracking even on sharp turns. Also, we examined various datasets like TUM_VI, KITTI, EuRoC, working on these datasets to understand issues and how to compensate them further, we did some configuration changes too.

The issue which we are not able to solve completely or the one that we would want to work further upon is getting **the inertial data integrated with the monocular version that we have working right now.** The transformation from IMU to the camera is tricky and we haven't been able to get those parameters just right (finely tuned along with the IMU noise parameters.)

Results:

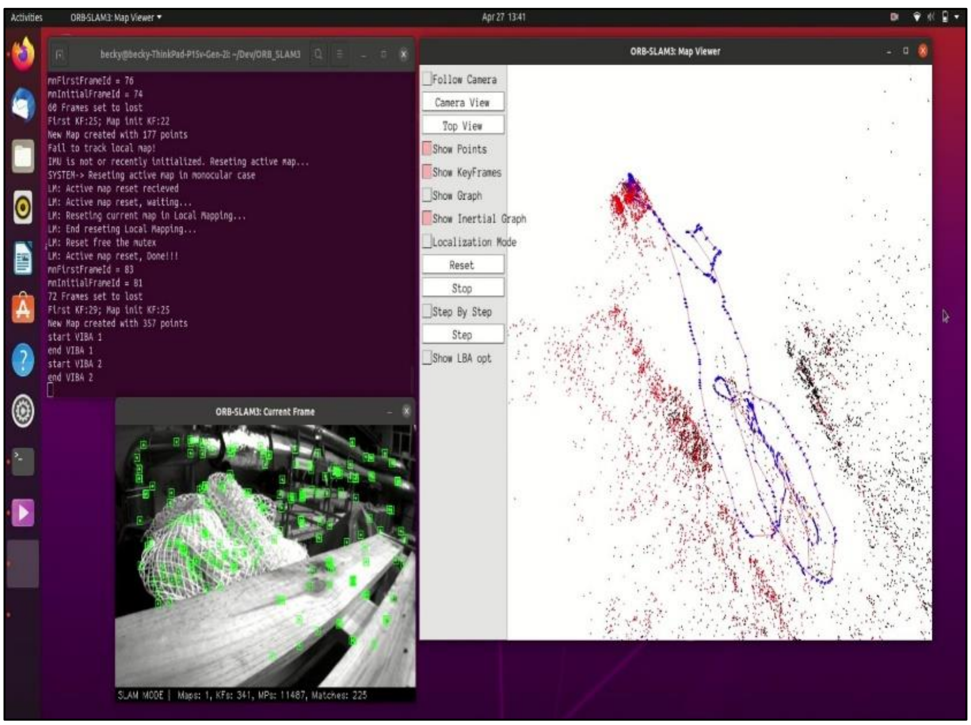

Figure 1: SLAM Results with EuRoC Dataset

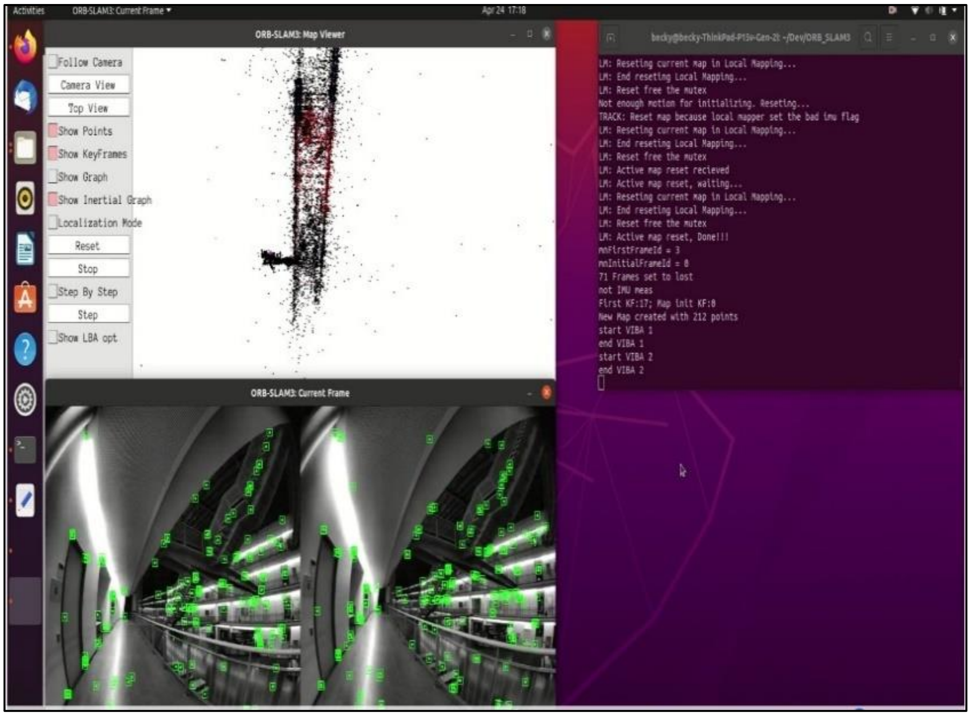

Figure 2: SLAM Results with TUM VI Dataset

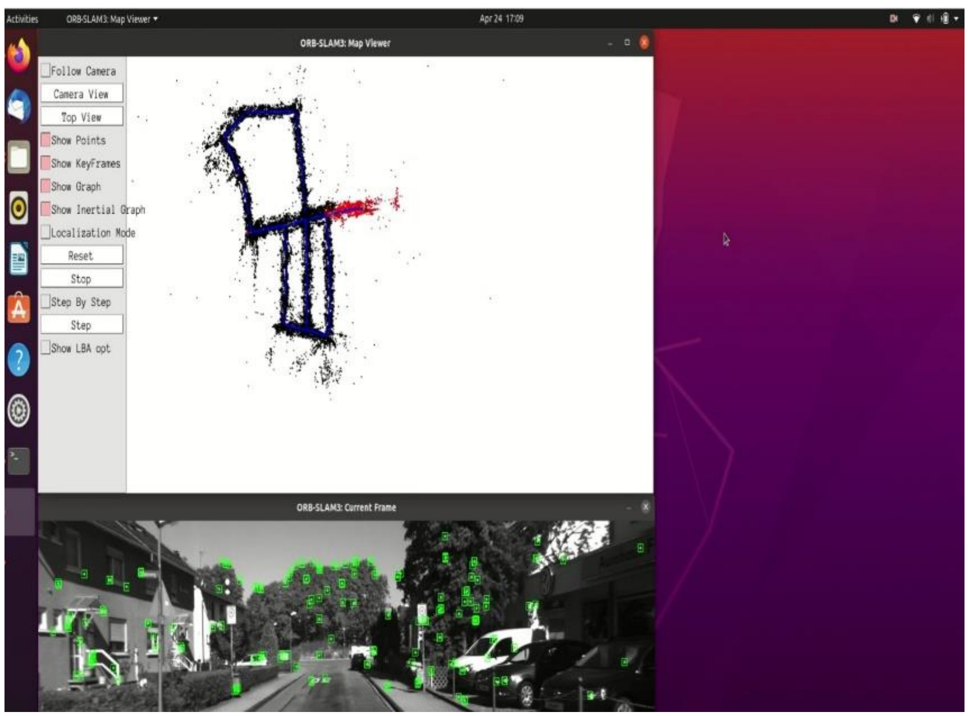

Figure 3: SLAM Results with KITTI Dataset

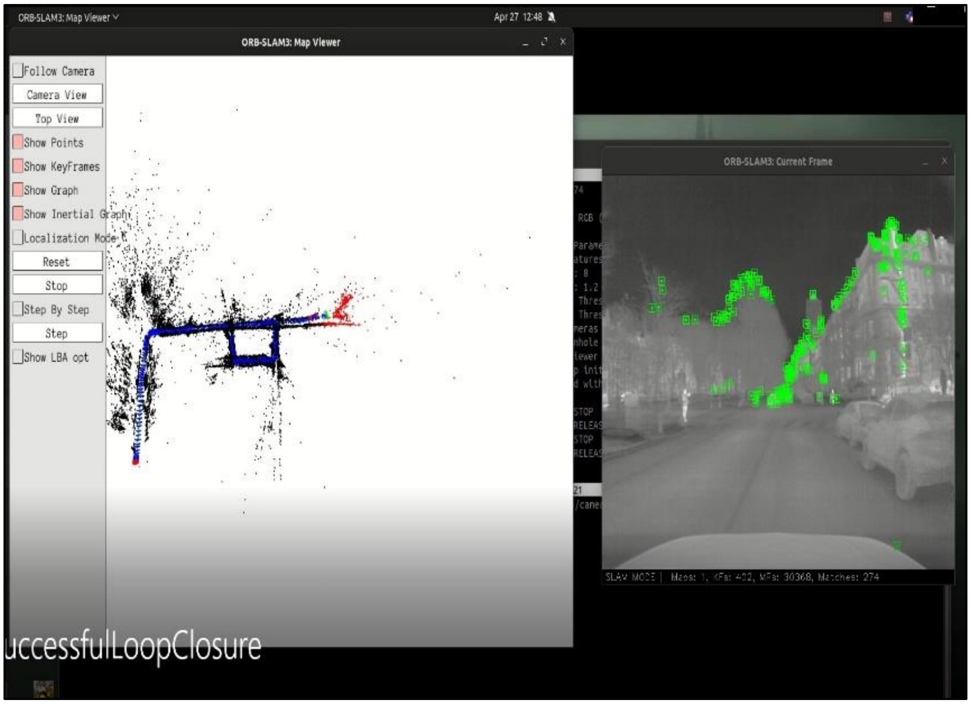

Figure 4: SLAM Results with NEU Dataset

Conclusion: ORBSLAM3 was successfully implemented for performing VSLAM on EuRoC, TUM_VI and KITTI datasets.

1.EuRoC dataset: For the Monocular and Stereo data without IMU we observed that stereo is necessary for good localization. In case of Monocular and Stereo data with IMU, we observe no significant difference, it just that Stereo just slightly better.

2.TUM-VI dataset: For this data set for room1 and room2 we observe that there are many loop closures which means we got the good mapping with both Monocular and Stereo datasets. Also, for outdoor mapping Stereo has a slight advantage over Monocular in low loop closure cases. Furthermore, in case of magistrale data set Stereo worked far better to pinpoint exact start and end location.

3.KITTI dataset: So far ORB SLAM 3 has worked the best on KITTI dataset. It could be because of the environment the data was collected in with good features to track and not so many dynamic objects nearby. While it does make mistakes while mapping initially, it compensates for it by doing an excellent job with the loop closures and bundle adjustment to generate an accurate map of the environment.

4.Nuance Dataset: The NUance Dataset was by far the trickiest as it was our custom dataset. Various parameters needed adjusting and we solved issues one step at a time as seen in the previous slides. We were able to successfully run the algorithm on monocular version and create a map of the environment. Integrating the inertial measurement data to further improve the accuracy and precision is the scope of improvement.

References for Report, Presentation, Datasets, and Analysis:

FAST
https://en.wikipedia.org/wiki/Features_from_accelerated_segment_test
https://medium.com/@deepanshut041/introduction-to-fast-features-from-accelerated-segment-test-4ed33dde6d65

BRIEF
https://medium.com/data-breach/introduction-to-brief-binary-robust-independent-elementary-features-436f4a31a0e6

ORB
https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6126544&tag=1

ORB SLAM
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7219438
https://ieeexplore.ieee.org/document/7946260
https://arxiv.org/pdf/2007.11898.pdf
https://github.com/UZ-SLAMLab/ORB_SLAM3

EuRoC
https://github.com/kevin-robb/eece5554-vslam
^used exclusively for ORB SLAM installation help and EuRoC download
KITTI
http://www.cvlibs.net/datasets/kitti/eval_odometry.php

TUM VI
https://vision.in.tum.de/data/datasets/visual-inertial-dataset

CAR_IR_RGB_LIDAR
https://drive.google.com/file/d/1ONXIQbz9TxC3U1J9puNLR61g3wyrVkWF/view?usp=sharing