EECE-5554 RSN FINAL PROJECT REPORT

***Abstract*—This document provides a final project report on Visual SLAM using the GPS, IMU, and camera data from the NUANCE car dataset. The GPS and IMU data were analyzed in MATLAB for ground truth, motion, and orientation over time. The camera data is extracted and implemented in the ORB SLAM for the localization and mapping of the car path.**

I. INTRODUCTION

Visual Simultaneous Localization and Mapping (vSLAM) refers to the problem of using images as external information to establish the position of the robot and build a map of the environment. When the robot is in an unknown environment, a positioning component is required to solve the self-localizing problem [1]. Visual Slam is to derive the 6-DOF camera pose using algorithms and the information collected by optical sensors. It can be used for robot or unmanned vehicle navigation and mapping. The techniques are divided into monocular, stereo, and RGB-D approaches depending on what kind of sensor is adopted. The main algorithm of the framework usually includes feature point extraction, feature point matching, and feature point tracking. Two common types of motion models are used to estimate the camera pose. In the first type, only a few feature points in the image are extracted, matched, and tracked. Compared to the first type, it takes much more time due to the extraction of all feature points from the images. Nevertheless, it provides a more reliable and accurate camera pose. It can be realized by low-cost hardware because of the low computation time requirement and has been widely adopted in robots, self-driving vehicles, and UAVs. The algorithms consist of the following steps: First, the effective feature points are extracted from the images. Second, the common feature points between the images are matched. Finally, the feature points are tracked by searching the image sequence to estimate the camera pose. The feature-based approaches only utilize part of the image, and the traditional methods for feature extraction include SIFT, SURF, and ORB. The visual odometer extracts the feature points from the images and then matches and tracks among frames to obtain the relative distance of the movement. The GPS provides the coordinates of absolute positions and the heading angles of the robot in the real world. The IMU sensor gives the acceleration, angular velocity, and orientation of the robot [2].

II. APPROACH

We chose the ORB-SLAM approach because it is simple to comprehend and use, and highly accurate in both translation and mapping. The major advantage of ORB-SLAM is that it can run on a single CPU. It is an open-source technology that has a huge development community and is constantly being updated and improved. The usage of the ORB feature descriptor, which is invariant to rotation, scale, and illumination changes, makes it resistant to various lighting situations, camera motions, and occlusions.

III. SENSORS AND THEIR PRINCIPLES

A. Global Positioning System (GPS)

GPS operates on the principle of measuring the time delay between the transmission of a signal from a GPS satellite and its reception by a GPS receiver. This time delay is used to calculate the distance between the satellite and the receiver. By receiving signals from multiple GPS satellites, the receiver can determine its precise location, track movement, and calculate speed and direction.

B. Inertial Measurement Unit (IMU)

The IMU sensing principle is based on inertial sensors, including accelerometers, gyroscopes, and magnetometers, which work together to provide a continuous stream of motion data. This data can be used to track changes in position and orientation over time in applications such as navigation and robotics. However, IMUs are subject to errors that can accumulate over time, and their accuracy can be improved by combining them with other sensors and algorithms.

C. Stereo Camera (two cameras)

Cameras use the principle of optical imaging to capture visual information by focusing light through a lens onto a sensor or film. In digital cameras, the sensor converts the light into electrical signals, while in film cameras, the light-sensitive emulsion on the film creates a latent image. Cameras can also use other types of sensors and lenses to capture images in various parts of the electromagnetic spectrum and with different perspectives.

IV. EXPERIMENTATION

A. COLLECTION OF DATASETS

The data is collected using Nuance car that has sensors such as 5-point gray BlackflyS 5MP cameras, 2 Velodyne VLP 16 Lidars, VectorNav IMU, and GPS.

Regarding the Visual Slam aspect of the project, we specifically focused on the data collected from GPS, IMU, and the two cameras: the left camera, accessed through the topic “/camera_array/cam0/image_raw”, and the right camera, accessed through the topic “/camera_array/cam1/image_raw”.

B. Algorithm Principles

ORB-SLAM is a feature-based SLAM system that can be used in real-time in both indoor and outdoor environments. The main three principles that are used in ORB Slam are Tracking, Local Mapping and Loop closing.

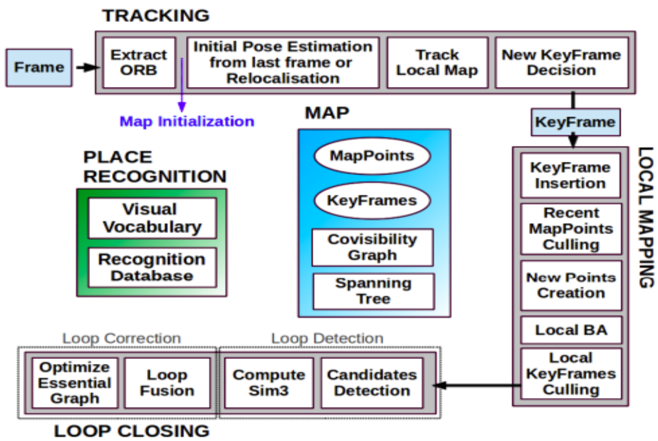


Figure 1: Main system components of ORB-SLAM.

I. *Tracking:* ORB features are extracted from the current frame using OpenCV and by matching these features with those already stored in the map will get the initial camera pose. The estimation of the camera's motion is computed using RANSAC concerning the previous frame by matching the ORB features and doing the necessary tracking of the local map, and the last step is to find the new keyframe, this is decided when the camera is moved certain distance or angle for the previous keyframe.

II. *Local Mapping:* The loop mapping process involves optimizing the camera pose and the 3D map points in the current frame and its neighbors by inserting a new keyframe into the map at the current camera pose. To reduce the computational burden, the map points in the previous key frame are removed and the creation of new points is done by triangulation method. This also uses a motion-only bundle adjustment algorithm that focuses on reducing the re-projection error between the observed feature points and their corresponding 3D map points. The local map is updated by removing redundant map points and adding new ones.

III. *Loop Closing:* The first is to identify the set of candidate keyframes and compare the bag of word descriptors of the current frame with those of the previous keyframes by using a hierarchical clustering algorithm that groups the descriptors into clusters based on their similarity. After this, the similarity transforms(sim3) are computed between the two frames and estimate the loop closure constraints. Now the local map and candidate keyframes are merged into a single map by reprojecting the map points of the candidate frames into the current frame. Finally, the essential graph is optimized and the accuracy of the estimated camera trajectory.

C. Methodology

After building the ORBSLAM package for ROS Noetic, SLAM could be run once a configuration file was created using the camera calibration information stored in the “/camera_array/cam0/camera_info” and “/camera_array/cam1/camera_info” topics [9]. Using this configuration file, the provided ORB vocabulary file, and the rosbag, visual SLAM could be performed.

V. Analysis

To analyze the results from the acquired dataset, the IMU and GPS data were used as ground truth. Below is a series of graphs that show the path that the NUANCE car took. They show a regular path, with nine total turns taken at an altitude of 6 meters. The altitude of 6 meters is consistent with the city of Boston’s average altitude of 5.8 meters. The turns taken by the NUANCE car are reflected both in the vehicle path using the GPS data as well as the orientation and angular velocity plots from the IMU.

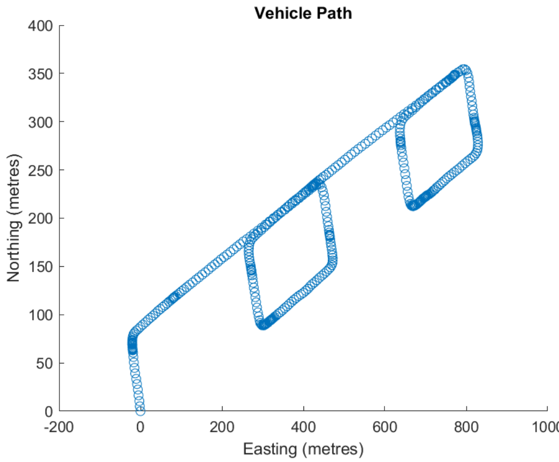
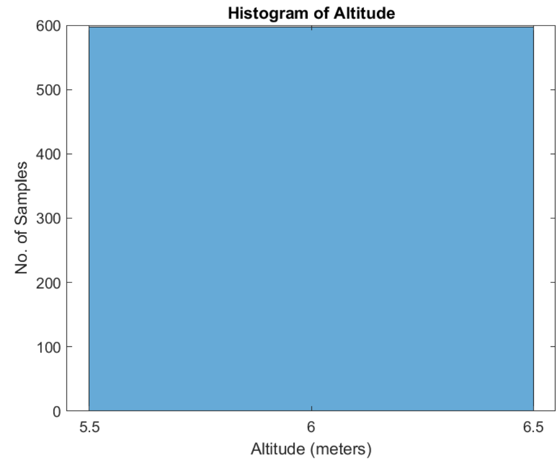


Figure 2: (a) Vehicle path acquired from the GPS data



(b) Histogram of the vehicle's altitude throughout the path

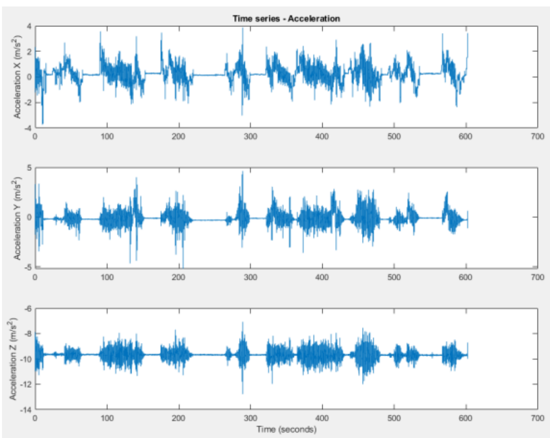
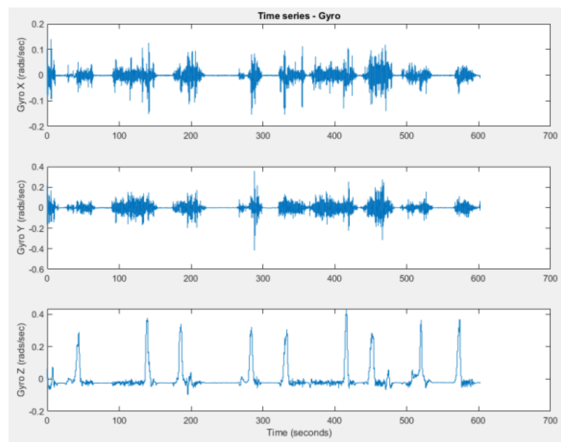


Figure 3: (a) Acceleration VS Time plot from the IMU



(b) Angular velocity VS time plot from the IMU

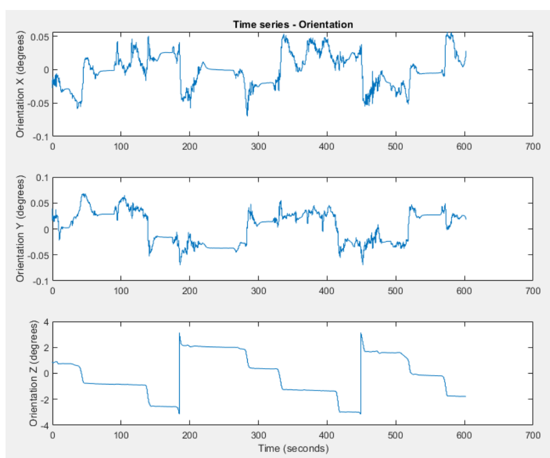


Figure 4: Orientation VS time plot from the IMU

Using this IMU and GPS data, we can then compare the vehicle path to that which is shown as the output of the ORBSLAM algorithm. Additionally, we can compare the performance to a sample datasets SLAM dataset, provided by the KITTI Vision Benchmark Suite.

VI. Results

The results from the ORBSLAM algorithm with the NUANCE car dataset were not quite what we expected. The performance was suboptimal. On almost every turn, the algorithm was unable to keep the car localized and had to start over. As a result, the vehicle path output from the algorithm showed a straight line, with many incomplete right turns. To attempt to remedy this, the number of maximum ORB features detected was increased to 5000, but there was no significant improvement. If the consecutive paths were placed end to end, they may have created an accurate representation of what the NUANCE car did, similar to the GPS data. In contrast, the KITTI dataset provides an uninterrupted map, showing an example of what an accurate path should show.

Figure 5: Resulting vehicle path from Monocular ORBSLAM3

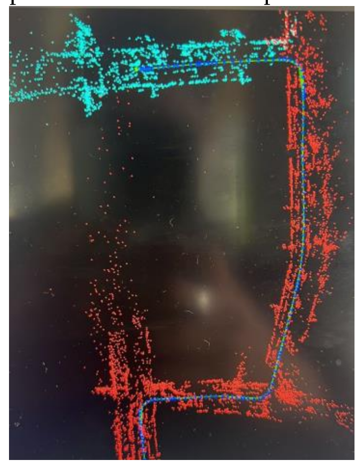


Figure 6: Output from the KITTI dataset

This mismatch is most likely related to the framerate of the car and the angular rate of the turns the car was taking. The roads in Boston are relatively tight, especially where this dataset was taken, so paired with the ~8.5Hz rate of images being published in the rosbag, there was not enough similarity between frames during turns to continuously localize the NUANCE car.

REFERENCES

1. T. Ruan, V. A. Aryasomyajula and N. Houshangi, "Performance of monocular and stereo camera in indoor environment for Visual SLAM using ORB method," 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 2022, pp. 273-278, doi: 10.1109/eIT53891.2022.9813775.
2. G. -S. Cai, H. -Y. Lin and S. -F. Kao, "Mobile Robot Localization using GPS, IMU and Visual Odometry," 2019 International Automatic Control Conference (CACS), Keelung, Taiwan, 2019, pp. 1-6, doi: 10.1109/CACS47674.2019.9024731.
3. S. Shao, "A Monocular SLAM System Based on the ORB Features," 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 2023, pp. 1221-1231, doi: 10.1109/ICPECA56706.2023.10075945.
4. N. H. K. Tran and V. -H. Nguyen, "An EKF-Based Fusion of Visual-Inertial Odometry and GPS for Global Robot Pose Estimation," 2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Kedah, Malaysia, 2021, pp. 1-5, doi: 10.1109/ICRAIE52900.2021.9703965.
5. H. Lezki and İ. Ş. Yetik, "Localization Using Single Camera and Lidar in GPS-Denied Environments," 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, Turkey, 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302512.
6. K. Ismail, R. Liu, J. Zheng, C. Yuen, Y. L. Guan and U. -X. Tan, "Mobile Robot Localization Based On Low-Cost LTE And Odometry In GPS-Denied Outdoor Environment," 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 2019, pp. 2338-2343, doi: 10.1109/ROBIO49542.2019.8961750.
7. K. J. de Jesus, H. J. Kobs, A. R. Cukla, M. A. de Souza Leite Cuadros and D. F. T. Gamarra, "Comparison of Visual SLAM Algorithms ORB-SLAM2, RTAB-Map and SPTAM in Internal and External Environments with ROS," 2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE), Natal, Brazil, 2021, pp. 216-221, doi: 10.1109/LARS/SBR/WRE54079.2021.9605432.
8. A. Mahmoud and M. M. Atia, "Hybrid IMU-Aided Approach for Optimized Visual Odometry," 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, 2019, pp. 1-5, doi: 10.1109/GlobalSIP45357.2019.8969460.
9. RAFALAMAO. (n.d.). ORB_SLAM3_NOETIC Version (1.0.0). GitHub. Retrieved April 29, 2023, from https://github.com/RAFALAMAO/ORB_SLAM3_NOETIC.