Benchmarking different SLAM Approaches

Narathip Rodwarna, Sridharan Subramanian, Amey Gavale, and Yifei Li

I. Brief Summary

The project with GE Aerospace focuses on comparing and benchmarking different SLAM approaches and eventually fusing them together in a combination to obtain an output that outperforms the individual outputs. In pursuit of this goal, the team aims to explore modules like ORB-SLAM3, OpenVSLAM, etc to benchmark them on standard datasets which ensure the robustness of the modules and compare against each other on quantitative terms such as Absolute Pose Error and Relative Pose Error. This leads to the path of developing a fusion pipeline for these approaches to obtain more accurate and optimal trajectory output. We have completed testing both algorithms on the EuRoC dataset and will continue with the remaining datasets, including UZH-FPV Drone Racing, Kagaru Airborne, and TartanAir. This leads to the path of developing a fusion pipeline for these approaches to obtain more accurate and optimal trajectory output. Additionally, we have started training an LSTM model and will work with it to achieve good results.

II. Work Completed

i.) Environment Setup

First, we set up the environment on multiple laptops, acknowledging that different systems presented various challenges. Ultimately, we successfully configured the setup for OpenVSLAM and ORB-SLAM3 on several laptops. For ORB-SLAM3, there was no official Docker image available in the repository. Instead, we utilized a Docker image from an alternative source [2]. Additionally, we encountered an issue when attempting to build ORB-SLAM3 with ROS, which necessitated using a ROS wrapper from another repository [3]. We integrated the ROS code from this source with the original ORB-SLAM3 repository to resolve the build issues. Regarding OpenVSLAM, we employed two approaches: installing it directly on Ubuntu 22.04 and using Docker images [4] from the official repository. Both approaches proved successful, allowing the algorithm to run normally.

ii.) Algorithm Evaluation

Our goal is to evaluate four datasets, including EuRoC MAV, UZH-FPV Drone Racing, Kagaru Airborne, and TartanAir. We have completed the evaluation of the EuRoC dataset on both algorithms, and the results are shown in Tables 1 and 2. We evaluated both Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) using the evo package, making the results from both algorithms comparable as evo uses the Umeyama alignment method for alignment and scale adjustments.

Table.1 Average APE Comparison(in m)

APE	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
Open VSLAM	0.0292	0.0670	0.0379	0.0997	0.0498	0.0815	0.0668	0.0955	0.0504	0.1169	0.0977
ORB-SLAM3	0.0331	0.0267	0.0327	0.0496	0.1473	0.0821	0.0613	0.1332	0.1523	0.0498	0.0748

Table.2 RPE Comparison(in m)

RPE	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
Open VSLAM	0.0230	0.0242	0.0561	0.0550	0.0497	0.0226	0.0456	0.0430	0.0163	0.0371	0.0555
ORB-SLAM3	0.0232	0.0240	0.0562	0.0563	0.0522	0.0235	0.0472	0.0508	0.0212	0.0371	0.0503

iii.) Trajectory Fusion

There are two ideas of fusion planned:
1. Weighted Average Fusion of the trajectory based on APE
2. RNN-LSTM implementation - Data preprocessing is done with the help of trajectory files from both SLAM modules and ground truth files. The RNN pipeline has been developed to intake synchronized trajectories with one Dense and an LSTM layers. The optimizers to be tested are SGD and Adam with regularizations.

III. Issues and Challenges

i.) Issue 1:

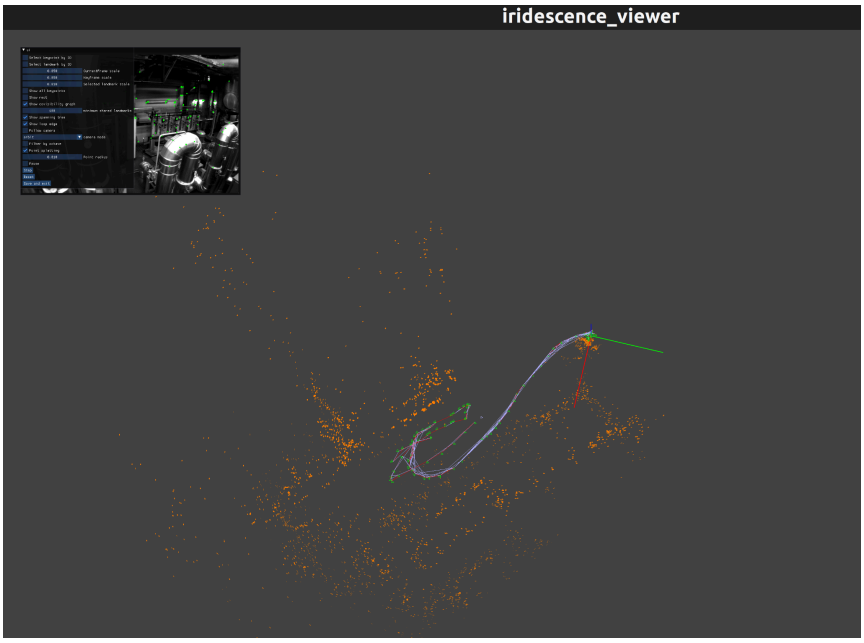
- **Finding the appropriate 3rd party visualization:**

  Here, as we can see, the OpenVSLAM's output should be visualized using a 3rd party viewer such as Pangolin Viewer or Iridescence viewer that is easier for debugging and configuration of settings.

- **Impact:**

  The initial viewer which was set up: Iridescence viewer. This was quite troublesome in the sense that the viewer wasn't properly supported by the Graphics Driver NVIDIA_535 in Ubuntu 22.04 - resulting in a very small window size making it difficult for visualization and configuring loop closure detection.

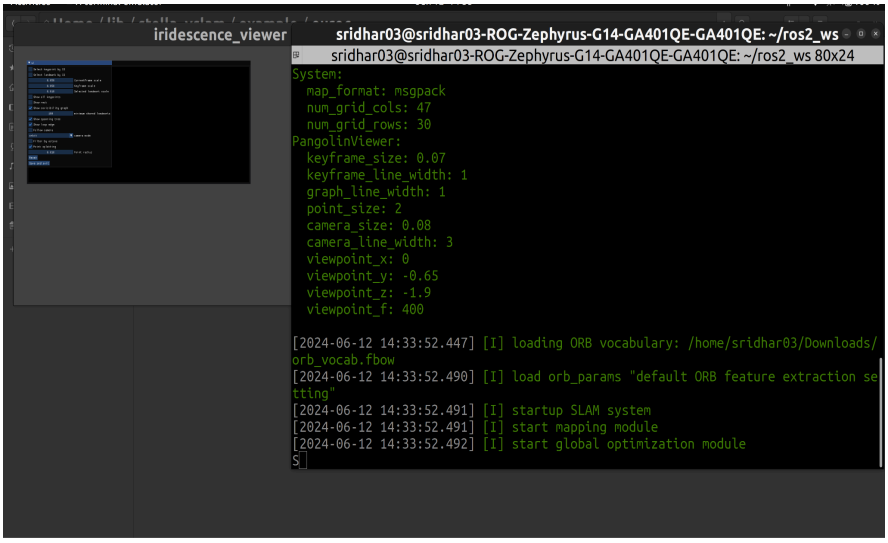
- **Solution:** The next step was to try Pangolin Viewer which works with the said driver.



ii.) Issue 2:

- **Empty Mapping in Iridescence viewer:**

  Iridescence Viewer when combined with ROS Wrapper for the OpenVSLAM, puts out an empty map during the mapping session.



- So, instead of ROS Wrapper - which also has the dependency issue with Backward_CPP and Backward_ROS, the OpenVSLAM system was set up successfully on Ubuntu 22.04 without ROS Wrapper, which solved the issue of mapping as well as that of Backward_CPP.

IV. Next Steps

- Create compatible camera configuration files for TartanAir dataset and Kagaru Airborne dataset which use stereo vision.
- Run each of the SLAM implementations on the TartanAir and Kagaru Airborne dataset.
- Implement weighted averaging fusion of two SLAM results with the corresponding trajectory files and ATEs.
- Complete the implementation of RNN-Long Short Term Memory which learns the temporal relationship in the positions & orientations from ORB3, OpenVSLAM with corresponding ground truth files as target during training, and in return outputs the optimal trajectory that outperforms individual counterparts in terms of ATE, RPE.

V. Conclusion

So far, we have successfully set up the corresponding environment to run OpenVSLAM and ORBSLAM_3 after trying out other SLAM approaches like LSD as well. Both the SLAM systems were run on multiple datasets and environments and majority of which are agile indoor environments of drone manipulation. Furthering this will be to extend on outdoor flight datasets such as Tartan Air and Kagaru Airborne datasets. The quantitative evaluation metrics such as APE and RTE will be used furthermore to fuse the results and compare against individuals, thus creating a trajectory fusion pipeline for Visual SLAM.

VI. Appendices

We have taken two quantitative aspects such as ATE and RPE.

- APE - Absolute Pose Error is the most common benchmarking aspect.
- RPE - Relative Pose Error

Here are some results on ATE from OpenVSLAM and ORB_SLAM3:

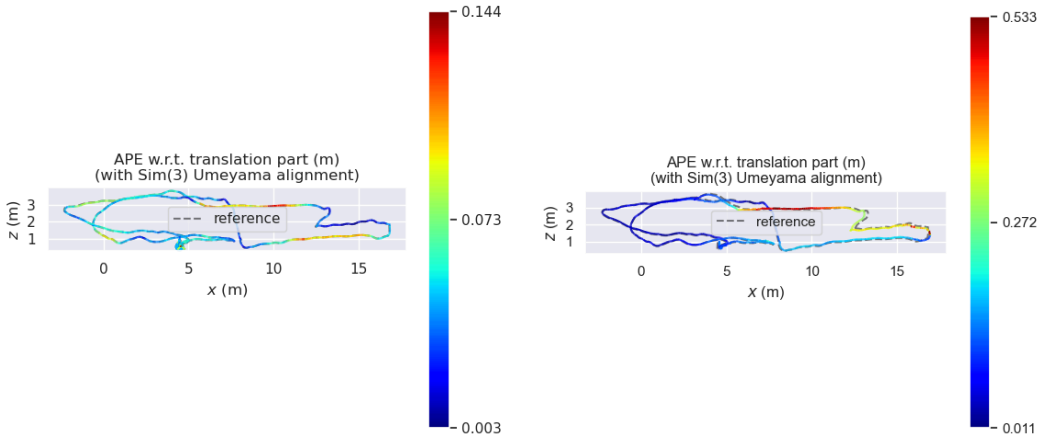


Fig.1 Heatmap of APE across the map of EuRoC MACHINE_HALL_05_DIFFICULT
left: OpenVSLAM, right: ORB-SLAM3

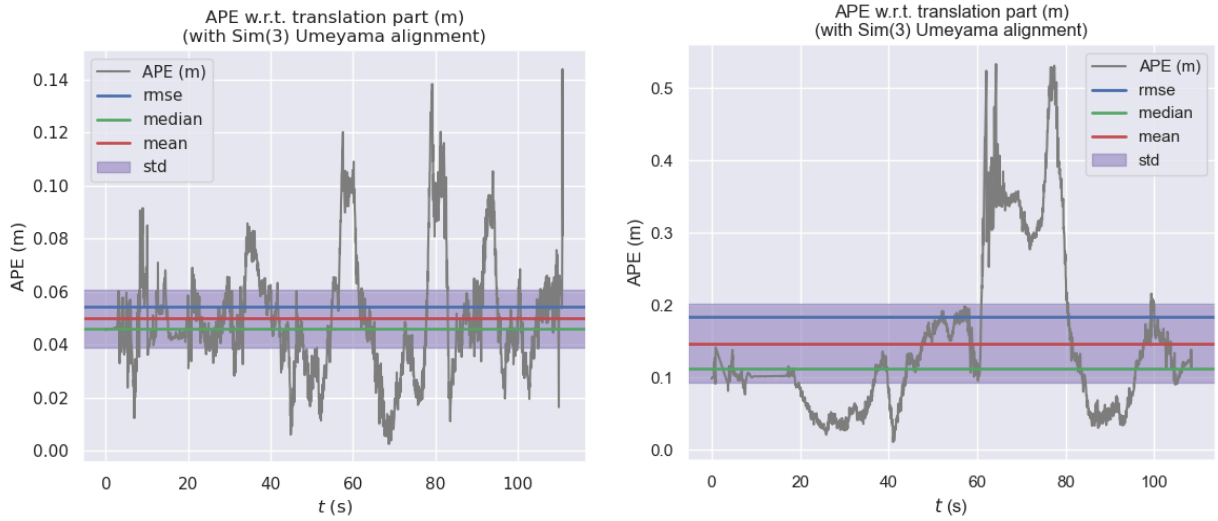


Fig.2 Raw graph of APE variation for EuRoC MACHINE_HALL_05_DIFFICULT
left: OpenVSLAM, right: ORB-SLAM3

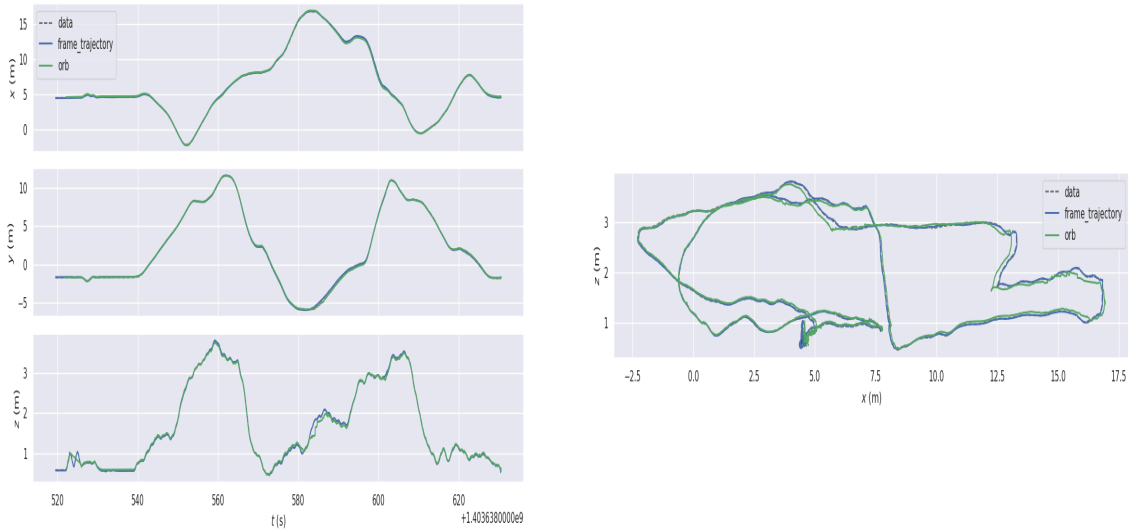


Fig.3 Trajectory Comparison between OpenVSLAM and ORB-SLAM3
of EuRoC MACHINE_HALL_05_DIFFICULT

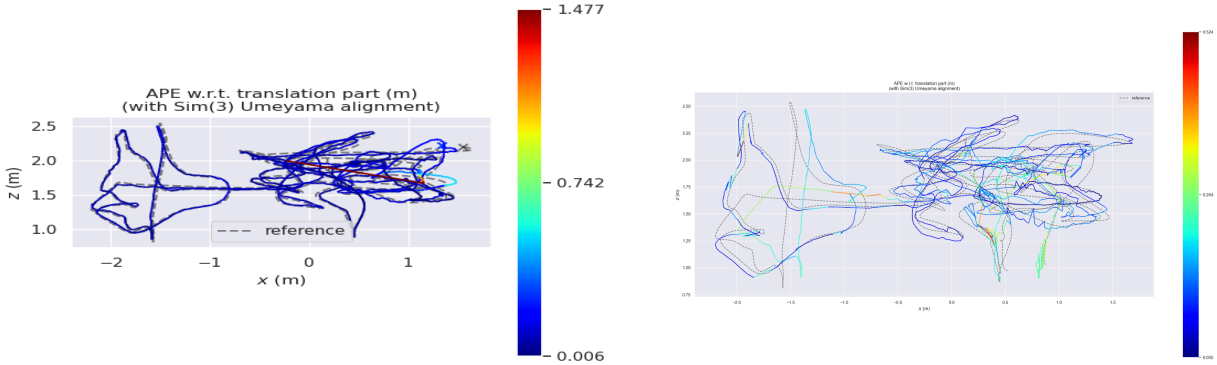


Fig.4 Heatmap of APE across the map of EuRoC VICON_ROOM_103_DIFFICULT
left: OpenVSLAM, right: ORB-SLAM3

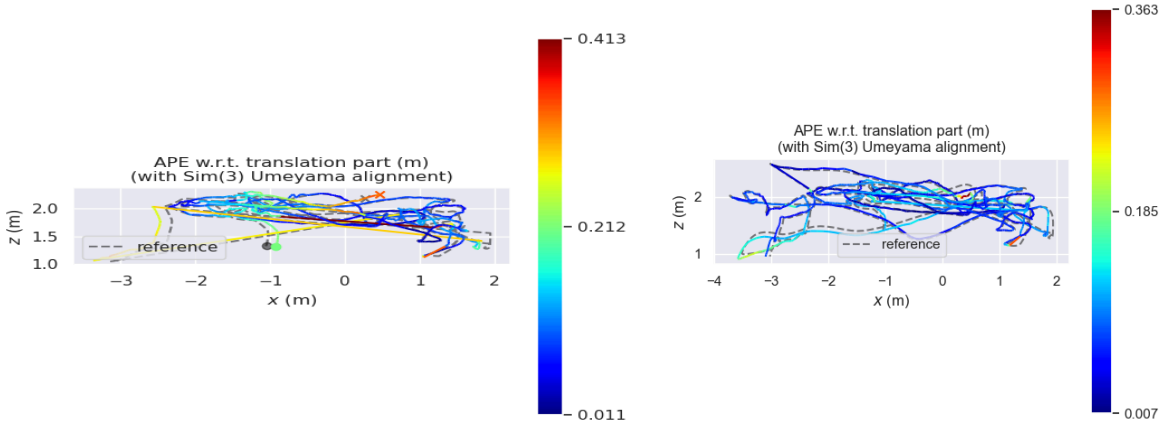


Fig.5 Heatmap of APE across the map of EuRoC VICON_ROOM_203_DIFFICULT
left: OpenVSLAM, right: ORB-SLAM3

11. References

1. https://github.com/UZ-SLAMLab/ORB_SLAM3
2. https://github.com/jahaniam/orbslam3_docker
3. https://github.com/thien94/orb_slam3_ros_wrapper
4. https://stella-cv.readthedocs.io/en/latest/overview.html
5. https://github.com/MichaelGrupp/evo
6. https://arxiv.org/abs/2007.11898
7. arXiv:1910.01122
8. https://github.com/nicolov/vslam_evaluation
9. https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets