Development and testing of different SLAM systems for a wheeled robot

Yue Pan[1,2] , Ruifan Wu[1] , Martin Barzcyk[1*]

1.Department of Mechanical Engineering, University of Alberta, Alberta, Canada.

2.School of Geodesy and Geomatics, Wuhan University, Hubei, P.R.China.

*mbarczyk@ualberta.ca

Introduction and related work

Autonoumous robots must be able to navigate in unstructured, dynamic and unknown spaces in order to operate in real world scenarios. Simultaneous localization and mapping (SLAM) is the ability for a robot to build a map of Its surrounding environment and to localize themselves in it. SLAM has been a popular research topic over the last two decades. Many approaches for solving the SLAM problem have been proposed, which are listed in Tabel 1. They are basically separated into two parts (Visual SLAM and Laser SLAM) based on the sensor they choosed.

Tabel 1.Popular open source SLAM solutions

SLAM Type	SLAM Solution	Published Year	Primary Sensor	Front-end Type	Back-end Type
Laser SLAM (L-SLAM)	Cartographer	2016	Laser Scan	Scan Matching	Ceres Graph Optimization
	Gmapping	2005	Laser Scan	Scan Matching	Particle Filter
	Hector SLAM	2011	Laser Scan	Scan Matching	BA
	LOAM	2014	Laser Scan	Scan Matching	/
Visual SLAM (V-SLAM)	RTAB-MAP	2014	RGBD camera	Feature-Based Dense	g2o Graph Optimization
	ORB SLAM	2012	Mono camera	Feature-Based Sparse	g2o Graph Optimization
	RGBD SLAM	2014	RGBD camera	Feature-Based Dense	g2o Graph Optimization
	LSD SLAM	2014	Mono camera	Direct Semi-Dense	Graph Optimization

Although these SLAM solutions use variant sensors, matching and solving strategy to acquire the map and pose, they generally follow the following workflow (Front-end plus Back-end), as shown in Fig.1. Different state estimation methods such as Extended Kalman Filter (EKF), Particle Filter (PF) and Graph Optimization are applied in this situation. Besides, equipped with all kinds of sensors like IMU, RGBD camera (Kinect v2) and Lidar, sensor integration turns out to be a direct way to get more accurate result.

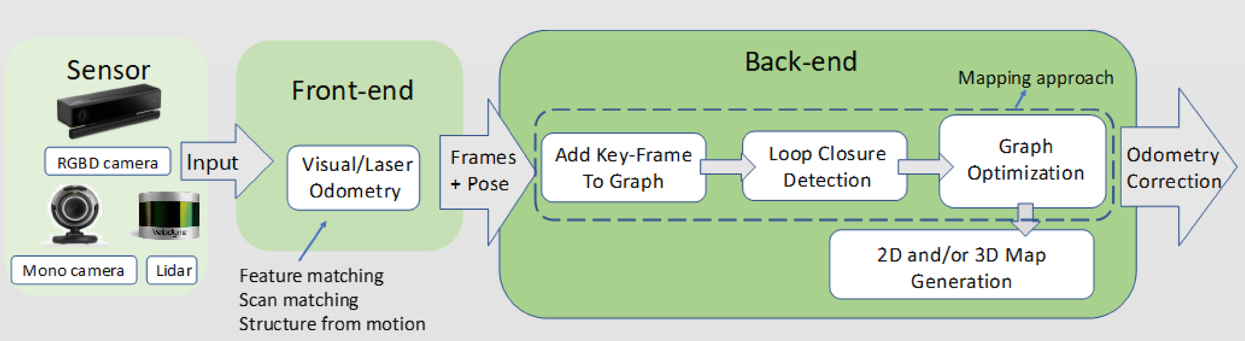

Fig.1. General workflow of SLAM system

In this work, we tested several popular SLAM systems on a wheeled robot (Jackal UGV) for the following purposes:

1. Evaluate the feasibility, applicability and accuracy of the SLAM solution.
2. Search for improvement of these SLAM solutions by parameters tuning and variant sensor integration.
3. Compare these SLAM solutions with each other for practical usage.

For this poster, RTAB-MAP[1] and Google Cartographer[2] are selected for discussion in detail.

Experimental Set-up

The system is implemented onboard a Jackal Unmanned Ground Vehicle (UGV). Several sensors are mounted on the it. A motion capture system from Vicon is used to obtain a ground truth of the robot's motions, as shown in Fig.2.

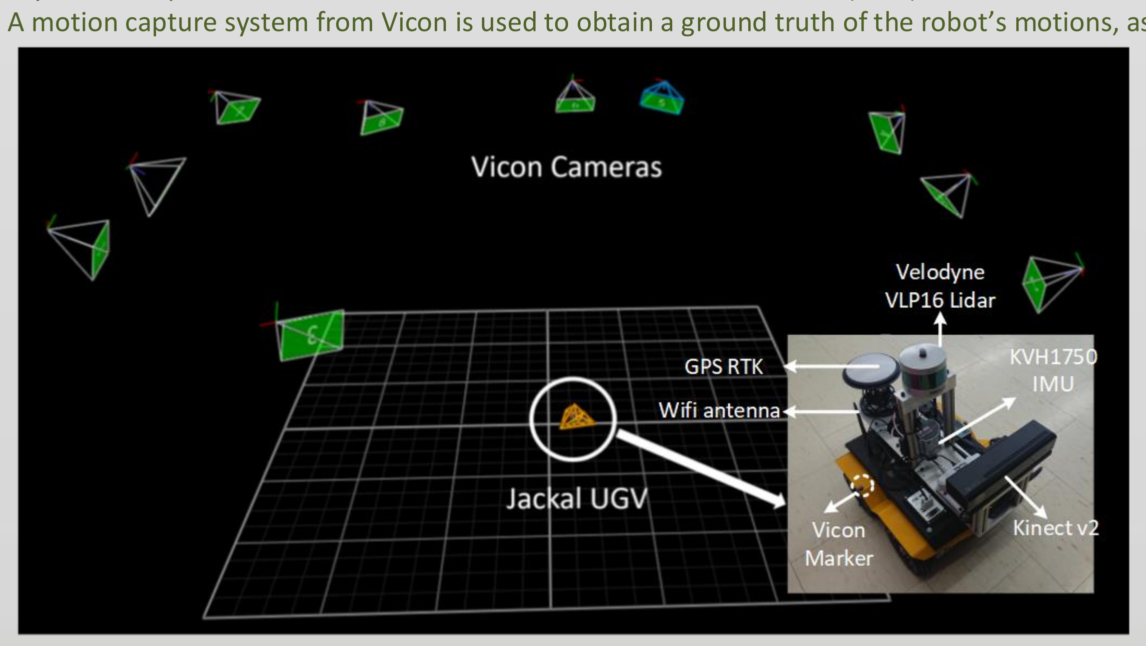

Fig.2. The experiment set-up. 10 VICON cameras are mounted around the test field. Jackal is attached with six motion markers to track its motion. A Kinect v2 RGBD-camera, a Velodyne VLP16 Lidar, a KVH1750 IMU and a GPS RTK rolling station is equipped on Jackal.

RTAB-Map

Real Time Appearance Based Mapping (RTAB-MAP) is a computationally tractable and relatively accurate SLAM solution based on RGBD camera. However, its resulting pose data is available only at 1 Hz. To overcome the disadvantage of the low data frequency from RTAP-MAP, we utilize an EKF filter to fuse in odometry with variant of sensors and poses with covariance to obtain higher-rate (50 Hz) pose estimates, as shown in Fig.3.

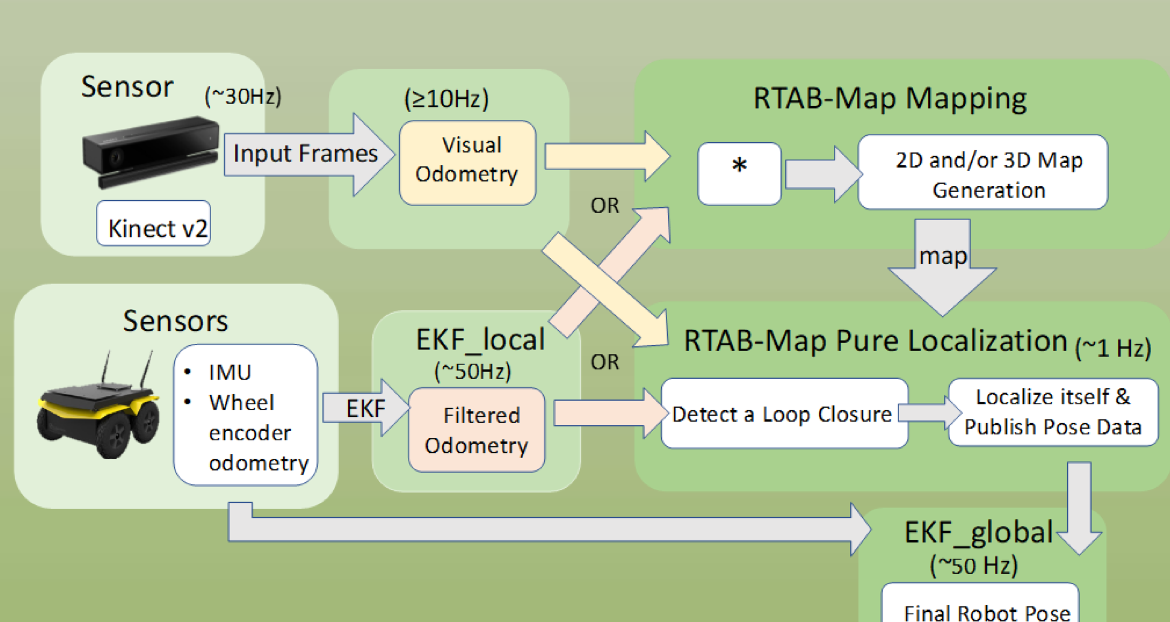

Fig.3. The flow chart of RTAB-Map mapping and pure localization process. RTAB-Map uses the visualization information from Kinect v2 to update the map, detect loop closure, correct odometry and realize localization with built map. To further refine the localization pose output by RTAB-Map, the filtered odometry from EKF_global is the final estimate pose data of Jackal UGV. (* refers to the mapping approach step Back-end in Fig.1)

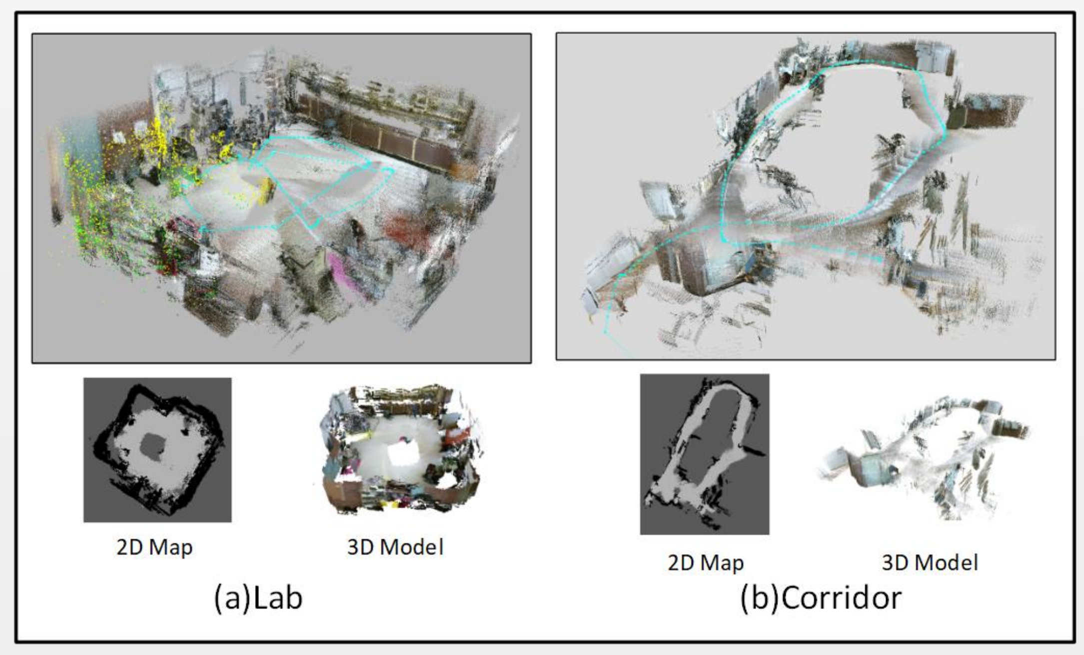

Fig.4. Mapping result of RTAB-MAP (2D grid map and 3D map for Lab and Corridor Dataset)

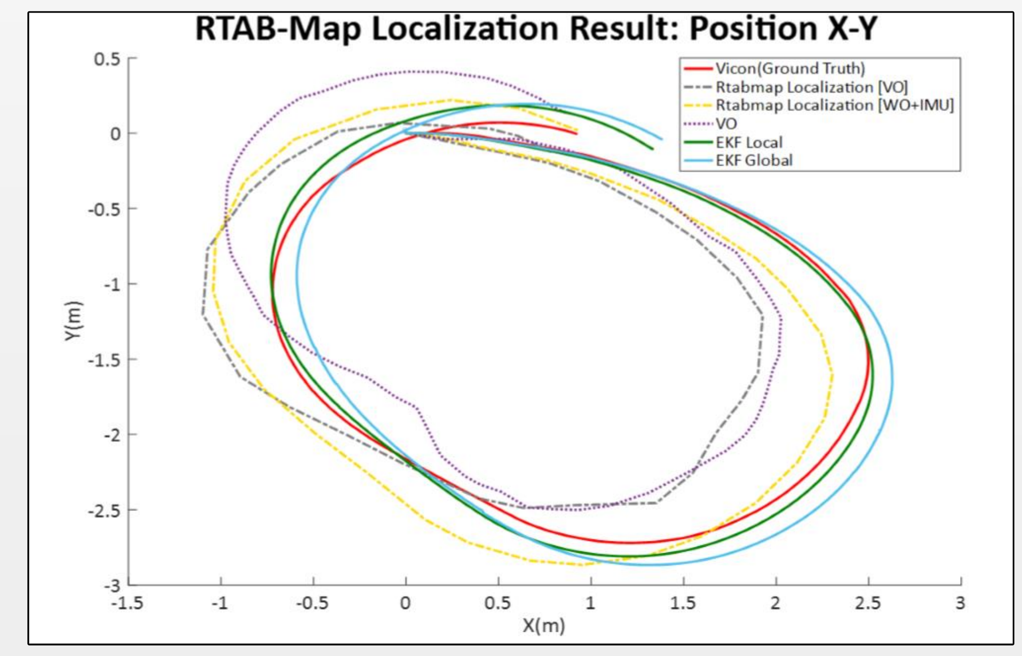

Fig.5. Localization result for different sensor integration strategy compare with Vicon ground truth trajectory, a 2D case

We apply RTAB-MAP to build the map of our lab and the corridor out of the lab using visual odometry (VO) and EKF filtered odometry separately. SURF feature points are extract from each frame and then matched with those in its neighboring frames to achieve VO. By using EKF filtered odometry instead of VO, images are only used to detect the loop closure, thus decreasing the accumulated error caused by mobile mapping to a great extent. As shown in Fig.4, the map of lab is generated using VO while corridor map is generated using EKF filtered odometry. Both the 2D map and 3D point cloud (or mesh) are generated via RTAB-MAP.

After we got a map (>30 locations has collected), we can switch to localization mode. As can be seen in Fig. 3, we can use a EKF_local node as the odometry input of RTAB-Map instead of using visual odometry. The EKF_local is fused with data from wheel odometry and Inertial Measurement Unit (IMU) mounted on Jackal UGV. The output of RTAB-Map localization mode and Jackal wheel odometry are feeding into the second EKF node EKF_global. The raw pose data of RTAB-Map have high accuracy but low frequency at around 1 Hz. The second EKF node successfully generates robust pose data at frequency of 50 Hz.

As shown in Fig.5, RTAB-MAP Localization Pose generated using wheeled odometry (WO) and IMU perform better than its counterpart generated using VO and achieve a precise loop closure as its terminal point is most close to ground truth. EKF local and EKF Global has similar accuracy compared to Vicon ground truth and they outperform all the other output poses. This may due to the limited spanning time of testing localization process thanks to IMU's high accuracy within a short time.

Cartographer

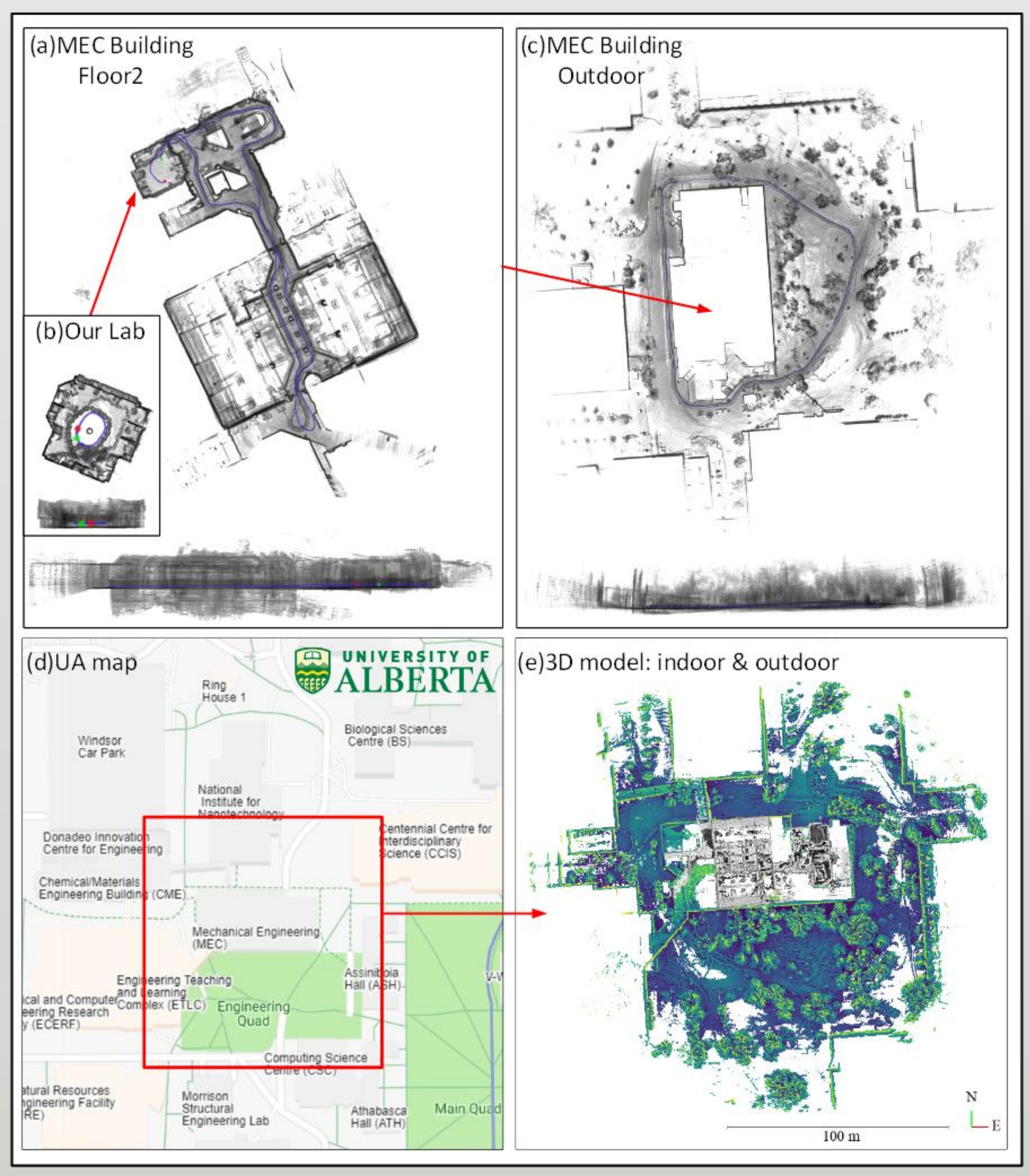

Fig.6. Google Cartographer test on UA campus (SLAM Result)

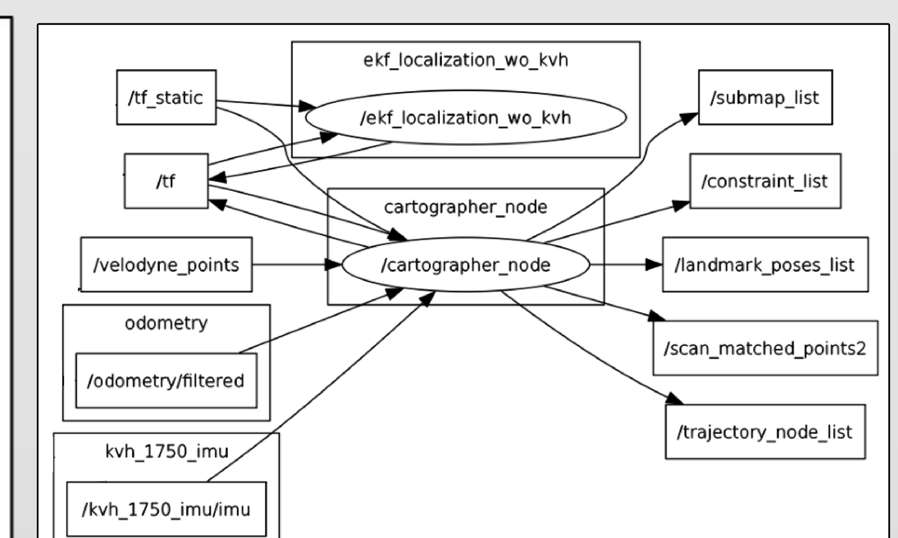

Fig.7. Subscribing and publishing topic and message of Cartographer (ROS Graph)

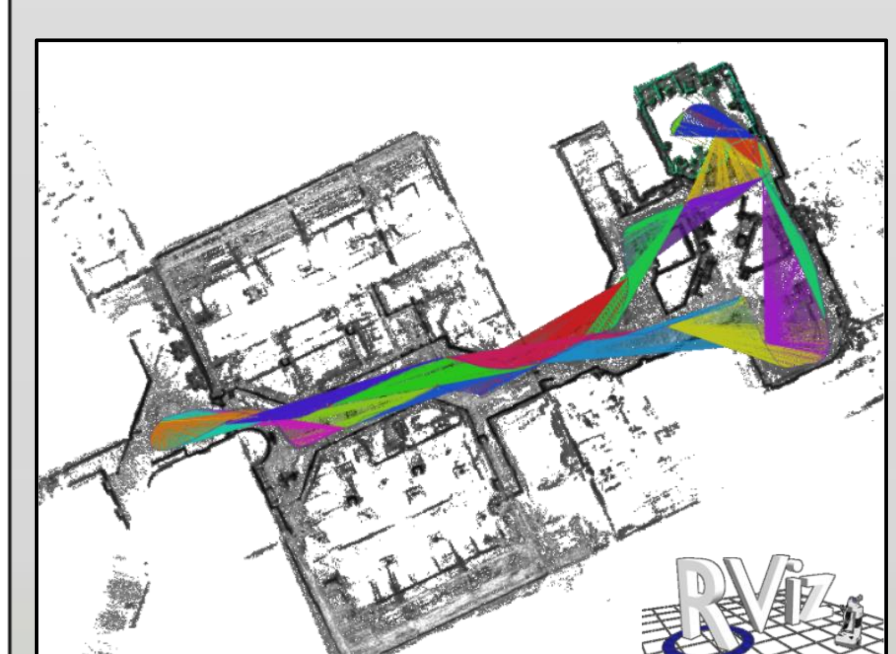

Fig.8. Visualize Cartographer SLAM process via Rviz. Constraint among submaps are shown as lines in different colors.

Google Cartographer is a state-of-art L-SLAM solution which provides both 2D and 3D mode. It builds local maps by scan matching and detects among local maps for loop closure so as to construct global map. Ceres Solver is used to do all these non-linear optimization.

Cartographer has a lot of parameters to tune. After tuning both local and global graph parameters, we tested it on both indoor and outdoor datasets. Though Cartographer is developed for indoor application, our test proves that it can also perform well outdoor with Velodyne VLP16 and a decent IMU.

Fig.6 show both the indoor and outdoor SLAM result tested on UA campus. An overall 3D model for MEC building and its surroundings is also generated by Cartographer. Fig.7 demonstrates the rostopic graph of Cartographer and Fig.8 shows the visualization of Cartographer with Rviz.

For the comparison between Cartographer and other L-SLAM solutions such as Gmapping and LOAM, please refer to our videos on this topic. It is shown that with its ability to detect loop closure, Cartographer outperform other solutions on both the rate of success and the average accuracy.

Conclusion and Future work

For RTAB-MAP, our test results show good agreement between the pose estimates from our system and the ground truth. Our work demonstrates the ability of an inexpensive RGB-depth sensor such as the Kinect v2, combined with an open-source SLAM package and fused with high-rate odometry estimates through an EKF, to achieve good performance and accuracy results.

For Cartographer, our test results demonstrate Cartographer capability on both indoor and outdoor dataset with a set of proper tuned parameters.

Our future work will focus on the following topics:

1. Test other popular open source V-SLAM and L-SLAM solutions and compare them with each other.
2. Since GPS RTK can reach millimeter horizontal accuracy, it can be used as the ground truth for outdoor SLAM.
3. Build a detailed 3D model for UA campus (both indoor and outdoor).
3. Dig into the possibility of 3D semantic L-SLAM, which uses deep learning to do semantic segmentation on the input point cloud and use these semantic information for better scan matching at the same time.

Primary Reference and Open Source Community

Reference

[1] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," Intelligent Transportation Systems Magazine, IEEE, vol. 2, no. 4, pp. 31–43, 2010.

[2]Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D LIDAR SLAM[C]// IEEE International Conference on Robotics and Automation. IEEE, 2016:1271-1278.

OpenSource Community

Comparison of different V-SLAM and L-SLAM solutions

Codes now available on Github: https://github.com/YuePanEdward/VSLAM-LSLAM-Comparison

Videos now available on Youtube: https://www.youtube.com/watch?v=wJPFnWXptLo https://www.youtube.com/watch?v=zGrvtwrzm64