LiDARs meet ShanghaiTech: A mobile mapping platform and its dataset

A project of the 2019 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University

https://robotics.shanghaitech.edu.cn/teaching/robotics2019

Yizhou Wang (wangyzh1@shanghaitech.edu.cn)
Jingyang Hou (houjy@shanghaitech.edu.cn)

Jan 2020

The car project is to build a mobile mapping platform mountable to a car and, with that platform, map the campus of ShanghaiTech University. Multiple LiDARs and cameras will be used to measure the surroundings and pass data to on-platform computation center to generate a colored model of precision.

1 Introduction

An accurate model of the campus has applications in various fields including automobile, navigation, offline localization, etc. and, therefore, is highly demanded. Though our school already have virtual version online, it is just collection of photos and meets few needs. Our project is aimed at 1) building a mobile mapping platform consisting of multiple laser scanners and cameras, and 2) mapping the campus and parking garage of ShanghaiTech University with the platform.

2 Related Works

2.1 Papers

Liu, Arief and Zhao [1] use Artificial Bee Colony algorithm and cuboid-base segmentation to analyze the efficiency of LiDAR placement on autonomous cars and to determine how to optimize according to the number and type of LiDAR.

Kim and Chung [2] propose a new method to overcome some major difficulties in LiDAR sensor-based localization is presented. It solves the problems of optical characteristics of the sensor, dynamic obstacles and changes in environment at the same time. This sensor model can be used in robots and autonomous vehicles.

Most of today's LiDARs are based on spinning mechanism, which cannot handle the movement of the sensor itself well enough. Gentil, Vidal-Calleja and Huang [3] propose a framework called IN2LAMA (Inertial LiDAR Localization And Mapping), which exploits preintegrated measurements over upsampled inertial data to handle motion distortion without any explicit motion-model. Furthermore, this paper validates the effectiveness of such framework through both simulated and real data.

Song, Guan, Tay, Law and Wen [4] propose a collaboration with UWB (ultra-wideband) and LiDAR. This idea is inspired by 2 facts: LiDAR can improve UWB-only localization accuracy; UWB ranging measurements may be able to remove the error accumulation in the LiDAR-based algorithm. This UWB/LiDAR fusion enables drift-free SLAM in real time use only ranging measurements.

In a typical LiDAR-based sensor, short laser pulses are emitted into the scene and the distance between sensor and object is derived from the time measured until an "echo" is received. In case multiple laser pulses of the same wavelength are emitted at the same time, the detector may not be able to distinguish between correct and false matches of laser pulses and echoes, resulting in erroneous range measurements and 3D points. In the study of Diehm, Hammer, Hebel and Arens [5], they examine this crosstalk effects and propose a data-based spatiotemporal filtering, which only needs three consecutive overlapping point clouds at a time to perform, allowing this approach to be performed synchronously with the data acquisition. Though the authors concern that this approach might remove small highspeed object, their experiment show their algorithm distinguish moving objects from noise points excellently [5].

In multi-LiDAR systems, calibration is essential for making collaborative use of different LIDAR data, while existing methods usually require modifications to the environments, such as putting calibration targets, or rely on special facilities, which is labor intensive and put many restrictions to potential applications. He, Zhao, Davoine, Cui and Zha [6] developed a calibration method for multiple 2D LIDAR sensing systems that could be conducted in a general outdoor environment using the features of a nature scene. The algorithm extracts the features such as points, lines, planes and quadrics from the 3D points of each LIDAR sensing. By matching such features, transformation parameters from each sensor to the frame is estimated and applied to laser scanner calibration. With average angular difference between planes as the assessment index, their experiment results have prove the algorithm impressively effective [6].

Ivan, Santiago and Josep [7] applied Kinitinuous and RTAB-Map method (both available on Github) to LiDAR datasets to do 3D reconstruction. The application proves successful on publically available datasets. The experiments happened both outdoor and indoor. Their results reveal that for indoor condition, Kintinuous is slightly better, but for outdoor condition, RTAB-Map is apparently better. The results also shows that movements in environment would create duplications in the map since the methods assume that the world is static. Besides, they test the methods without a visible color camera. This directly relates with our scenario, since we are only dealing with LiDARs on the car, thus these two tested methods would be taken into consideration. Furthermore, it warns us to take care of moving objects while reconstructing.

2.2 ROS Packages

2.2.1 gmapping

The gmapping package contains a ROS wrapper for OpenSlam's GMapping. The package provides laser-based SLAM (Simultaneous Localization and Mapping), as a ROS node called slam_gmapping. Using this node, people can create a 2-D occupancy grid map (like a building floorplan) from laser and pose data collected by a mobile robot.

GMapping is a highly efficient Rao-Blackwellized particle filer to learn grid maps from laser range data, authored by Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. The developers of this ROS package present adaptive techniques to reduce the number of particles in a Rao-Blackwellized particle filter for learning grid maps. They also propose an approach to compute an accurate proposal distribution taking into account not only the movement of the robot but also the most recent observation. This drastically decrease the uncertainty about the robot's pose in the prediction step of the filter. Furthermore, they apply an approach to selectively carry out re-sampling operations which seriously reduces the problem of particle depletion.

2.2.2 velodyne

Velodyne is a ROS package for Velodyne high definition 3D LIDARs, which offers basic ROS support for the Velodyne 3D LIDARs, maintained by Josh Whitley, authored by Jack O'Quin.

This package supports all current Velodyne HDL-64E, HDL-32E, VLP-32C and VLP-16 models. It contains four sub-packages: Velodyne_driver, Velodyne_laserscan, Velodyne_msgs and Velodyne_pointcloud. Package Velodyne_driver is ROS device driver for Velodyne 3D LIDARs and provides basic device handling for Velodyne 3D LIDARs. The driver publishes device-dependent velodyne_msgs/VelodyneScan data. Package Velodyne_laserscan extract a single ring of a Velodyne PointCloud2 and publish it as a LaserScan message. Package Velodyne_msgs collects and defines all ROS messages that are specific to Velodyne 3D LIDARs, simplifying the dependencies between velodyne stack components and their users. It also provides bag migration scripts for translating old captured data to the latest format. Package Velodyne_pointcloud provides point cloud conversions for Velodyne 3D LIDARs.

3 System Description

3.1 Overview

The hardware platform is a frame made of aluminum alloy, designed to be easily mounted on the roof of an SUV car. It consists of four Velodyne

16-beam laser scanners and one Velodyne 32-beam scanner to obtain both surroundings and ground features, multiple Ladybug5 to capture images, and four computers to process and store all data.

3.2 LiDARs

Five Velodyne laser scanners are attached to the platform, the 32-beam one horizontally installed on the roof and other four 16-beam scanners distributed on all sides of the vehicle at certain angles to the horizontal plane. To obtain environment features as rich as possible, the scanner towards front, left and right are installed at the largest degree where they won't feel the car (on our target vehicle the side scanners are at 45°, the front scanner is at 22.5°). The backward scanner is mounted on a piece of aluminum and extends about one meter behind, whose scanning plane is perpendicular to the ground. After removing a bolt, the rear alloy arm can be rotated 90 degrees, which is convenient for users to open the trunk.

After the degrees were decided, we designed corresponding supports that hold our sensors at fixed angles. We also designed special transition pieces and a holder that fits Velodyne 16-beam LiDARs. With mentioned transition pieces, we can replace our sensors in the future without efforts. All components will be made of aluminum to ensure rigidity of the whole system. Figure 1 and 2 give an overview of the whole frame with laser scanners. Figure 3 are close-ups of their holder.

3.3 Synchronizing

Referring to VLP16 manual provided by Velodyne, we find Velodyne sensors are able to synchronize their data with GPS-supplied time pulses. Synchronizing to the GPS pulse-per-second (PPS) signal provides us the ability to compute the exact firing time of each data point.

To utilize these features, we must provide Velodyne sensors a once-per-second synchronization pulse in conjunction with a once-per-second NMEA $GPRMC sentence. Figure 4 give the timing characteristics of the synchronization based on GPS pulses.

Since we will not use GPS locator, a Raspberry Pi is programmed to forge the pulses and GPS data sentences. Because we do not use GPS data for

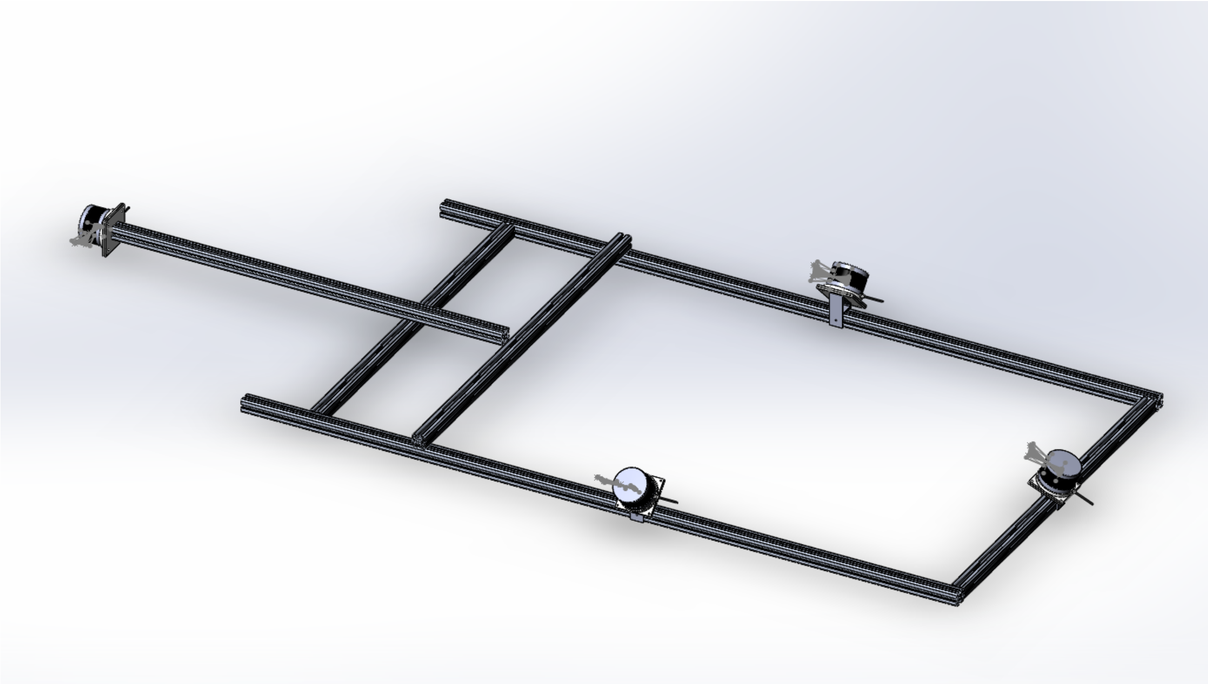

Figure 1: Frame overview

mapping, forged GPS signals will have no effect on our later steps.

3.4 Processing

The platform have four 4-core computers running Ubuntu 18.06 LTS to process and store all data into eight SSD drives, connected to each other through a Gigabit switch. All scanners are connected to computers' dedicated Ethernet ports, except the Velodyne Puck in the front. ROS (Melodic Morenia version) with velodyne package is used to retrieve and process LiDAR data.

Currently, the clocks of four computers are not strictly synchronized. Besides, the power supply and computation capacity on-vehicle are hugely restrained. Therefore, instead of online mapping, we tend to collect data for offline mapping. High quality online mapping may be a future interest.

Since we have synchronized Velodyne sensors' clock using forged GPS PPS signal, the timestamp of velodyne_msgs/VelodynePacket.msg message will give the relative time and can be used to align measures even if the four ROS are running with different clocks. We will use a preprocessing program to go through all data frames recorded by ROS, which aligns data

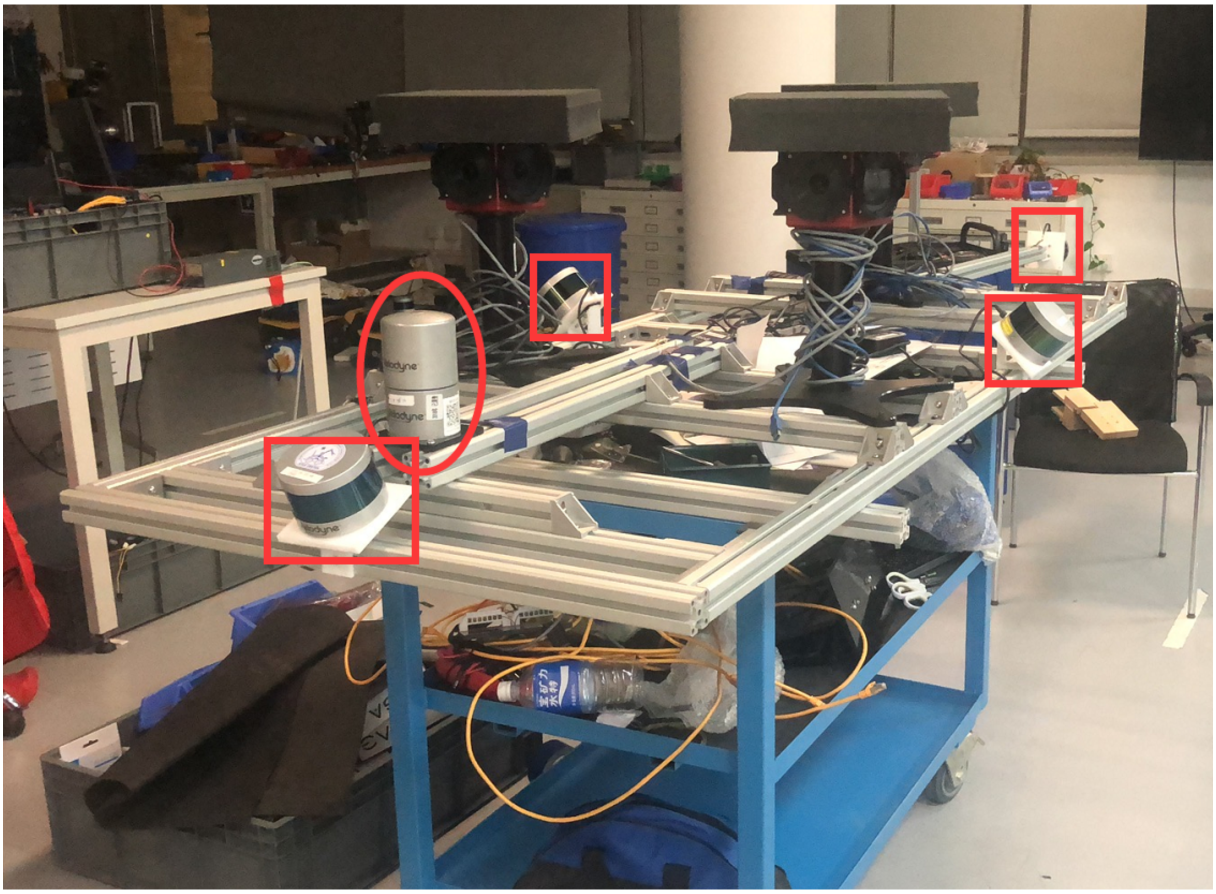

Figure 2: Picture of the Final Frame

frames with velodyne timestamps and republish messages to SLAM program. Then we may use open-source SLAM algorithms to do mapping and modeling.

4 System Evaluation

After the platform was all set, it was mounted to a real car and driven around the campus (including the parking garage) for three times, and over 100 GB laser scanner data were collected and uploaded onto the data station. Figure 5 is a screenshot of laser data playback. We will try to generate a model of the campus with this platform. Later, the map generated by SLAM will be compared with ground truth. For underground data, the ground truth is the official drawings of the plot; for ground data, we choose Google Earth pictures as ground truths. In addition, the path calculated by SLAM will also be compared to IMU record.

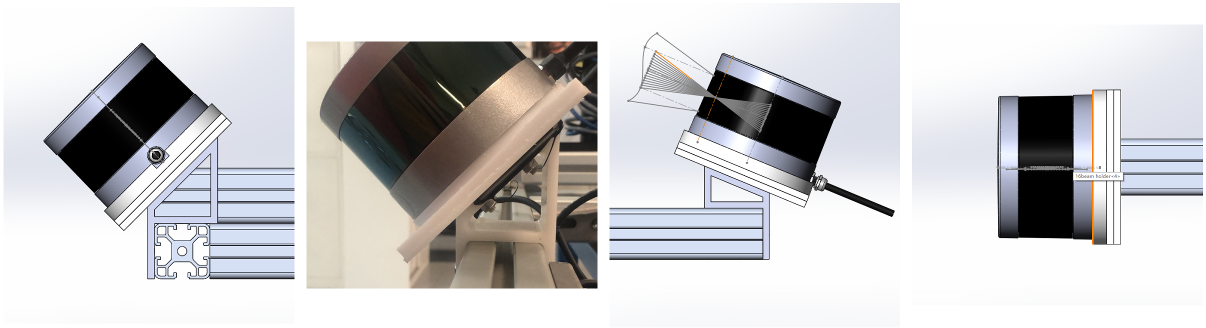

Figure 3: Holder close-ups

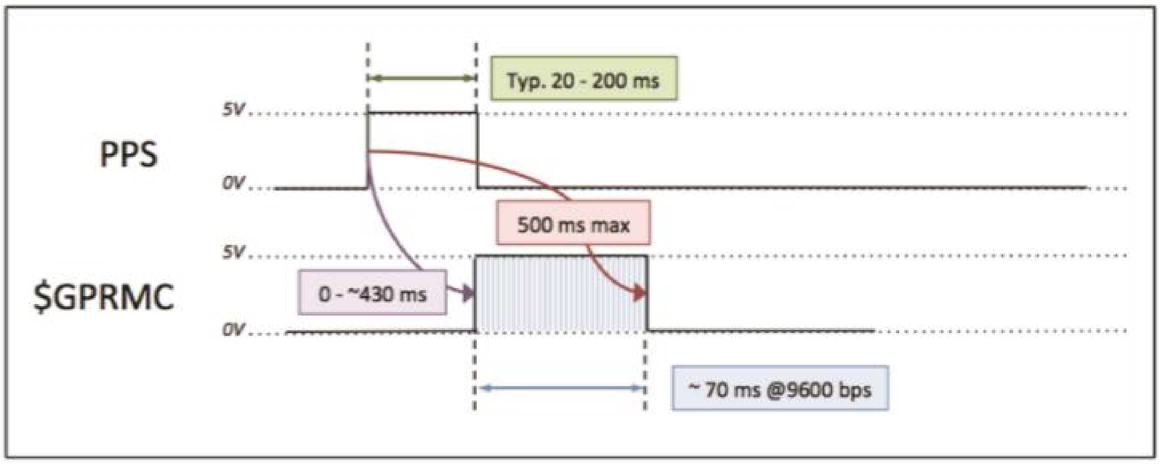

Figure 4: Timing characteristics of GPS synchronization

5 Work Plan

The whole project was divided into the following stages:

1. machine mounts for sensors and prepare electronics
2. install all sensors and devices properly
3. configurate ROS to correctly process and store data
4. collect sufficient data in field
5. generate the model based on collected data
6. final demo
7. final report
8. website

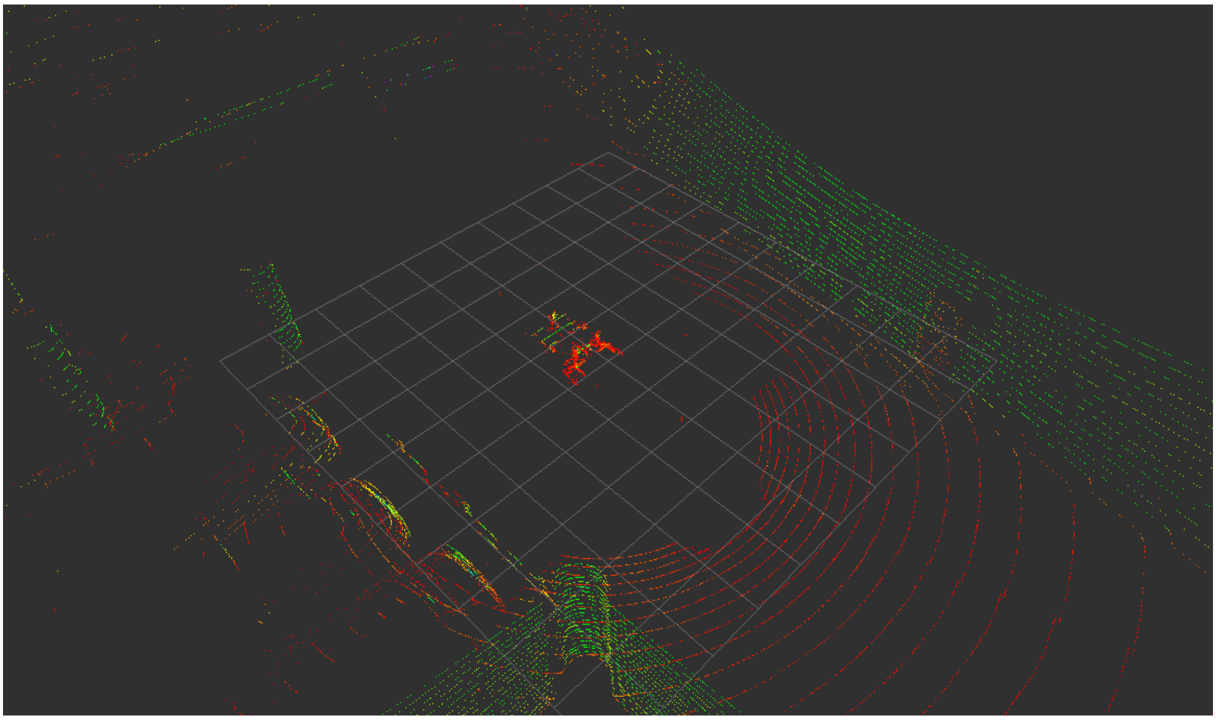

Figure 5: Screenshot of Laser Data Playback

6 Conclusion

The project proposed a mobile mapping platform, which uses multiple LiDARs and RGB cameras to obtain rich surrounding data and to generate colored models of urban area. Limited by on-vehicle power supply and computation capacity, the platform will not perform real-time mapping but only collect data for offline usage. Future work could be focused on real-time mapping and/or composing more sensors into current system.

References

[1] Z. Liu, M. Arief, and D. Zhao, "Where should we place lidars on the autonomous vehicle? - an optimal design approach," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2793–2799, May 2019.

[2] J. Kim and W. Chung, "Robust localization of mobile robots considering reliability of lidar measurements," in *2018 IEEE International*

Conference on Robotics and Automation (ICRA), pp. 6491–6496, May 2018.

[3] C. L. Gentil, T. Vidal-Calleja, and S. Huang, “In2lama: Inertial lidar localisation and mapping,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6388–6394, May 2019.

[4] Y. Song, M. Guan, W. P. Tay, C. L. Law, and C. Wen, “Uwb/lidar fusion for cooperative range-only SLAM,” *CoRR*, vol. abs/1811.02854, 2018.

[5] A. L. Diehm, M. Hammer, M. Hebel, and M. Arens, “Mitigation of crosstalk effects in multi-LiDAR configurations,” in *Electro-Optical Remote Sensing XII* (G. Kamerman and O. Steinvall, eds.), vol. 10796, pp. 13 – 24, International Society for Optics and Photonics, SPIE, 2018.

[6] M. He, H. Zhao, F. Davoine, J. Cui, and H. Zha, “Pairwise LIDAR calibration using multi-type 3d geometric features in natural scene,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, nov 2013.

[7] I. Caminal, J. R. Casas, and S. Royo, “Slam-based 3d outdoor reconstructions from lidar data,” in *2018 International Conference on 3D Immersion (IC3D)*, pp. 1–8, Dec 2018.