Efficient Scan-Matching Odometry for Robot Navigation with a High-Resolution Lidar

Haotian Gu
2020 Spring ME800
Advisor: Prof. Englot

Introduction

Lidar odometry is an effective navigation technique for robot localization when a robot does not have access to GPS. A robot's motion is estimated by solving for the translation and rotation across consecutive lidar scans. For the process to be real-time viable on an embedded system, salient features must be carefully selected from the lidar data, and used as the basis for scan matching. This project will aim to augment a successful lidar odometry framework, LeGO-LOAM, for use with a high-resolution, 64-beam lidar, over which existing algorithms currently do not work in real-time.

1.Literature survey about the robot localization and lidar odometry techniques

The traditional localization technique that has been widely applied in autonomous platforms is the Global Positioning System (GPS). It is a global satellite system that uses radio signals to determine the position and speed of mobile platforms with global coverage. Although advanced GPS systems can at best provide accurate positioning within a few centimeters, it is still not reliable enough for a core navigation system of autonomous platforms, especially for localization of aquatic and aerial vehicles [1]. In forests the signal strength is weaker than urban areas, which means the strength of a satellite signal varies depending on the place and environment conditions.. Moreover, GPS is not suitable for indoor navigation, since radio signals are affected by walls and other objects. All these factors disturb the process of acquisition and tracking of GPS signals at receivers, making self-localization less reliable [2], [3].

Inertial odometry or an inertial navigation system (INS), is a localization method that uses the measurements from the IMU sensor to determine the position, orientation, altitude, and linear velocity of a vehicle/robot, relative to a given starting point. However, these systems suffer from a drifting issue due to errors originated from different sources e.g., constant errors in gyroscope measurements and accelerometers. These errors, later, lead to an increasing error in the estimated velocity and position [4]. Therefore, inertial odometry systems are inaccurate and unsuitable for applications that require localization for long periods of time. To tackle this problem, different solutions have been proposed. In [5], for instance, a probabilistic approach based on double-integration rotated acceleration using the extended Kalman filter framework (EKF) is presented. Even with such improvements, inertial odometry is not capable enough to be used as the primary navigation method for autonomous vehicles in GPS-denied environments. Figure 1 illustrates the structure of the inertial navigation system in which the measurements obtained from the IMU sensor are integrated using the dead reckoning method to estimate the current pose.

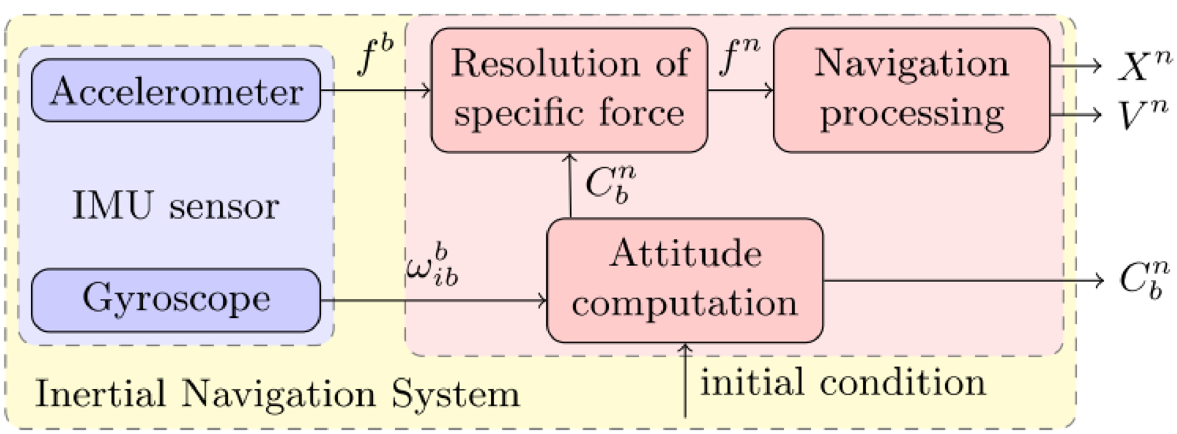

Fig 1 The general block diagram of inertial navigation system

2.Analysis about the reason why LeGO-LOAM can not totally support the calculation of ouster data set in 64 beams.

Figure 2 illustrates working framework of LeGO-LOAM. In the middle of rectangle with round angle is four independent nodes. With point cloud input in standard pcl::PointXYZI and imu messages input optionally, the output messages are global pointcloud map and robot gesture.

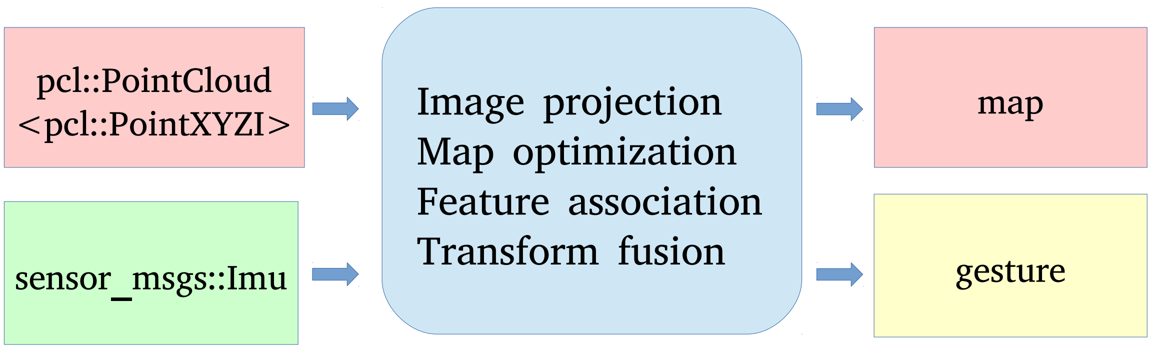

Fig 2 LeGO-LOAM framework

2.1 Comparison about rosbag info between VLP bag and ouster bag.

```
path:        2019-05-07-12-58-37_0.bag
version:     2.0
duration:    1:04s (64s)
start:       May 07 2019 14:39:57.03 (1557254397.03)
end:         May 07 2019 14:41:01.58 (1557254461.58)
size:        500.1 MB
messages:    47768
compression: none [656/656 chunks]
types:       ouster_ros/PacketMsg [4f7b5949e76f86d01e96b0e33ba9b5e3]
topics:      /os1_node/imu_packets      6456 msgs    : ouster_ros/PacketMsg
             /os1_node/lidar_packets   41312 msgs    : ouster_ros/PacketMsg
```

Fig 3 rosbag info about ouster-dataset in 64 beams

```
path:        2018-05-18-14-49-12_0.bag
version:     2.0
duration:    60.0s
start:       Jun 17 2017 09:24:11.25 (1497705851.25)
end:         Jun 17 2017 09:25:11.24 (1497705911.24)
size:        444.5 MB
messages:    24395
compression: none [530/530 chunks]
types:       sensor_msgs/Imu         [6a62c6daae103f4ff57a132d6f95cec2]
             sensor_msgs/PointCloud2 [1158d486dd51d683ce2f1be655c3c181]
             tf2_msgs/TFMessage      [94810edda583a504dfda3829e70d7eec]
topics:      /imu/data           4299 msgs    : sensor_msgs/Imu
             /tf                19523 msgs    : tf2_msgs/TFMessage      (6 connections)
             /velodyne_points     573 msgs    : sensor_msgs/PointCloud2
```

Fig 4 rosbag info about VLP in 16 beams

-Information type

In ouster data set, the information type pointclouds and IMU are both ouster_ros/PacketMsg which is not standard ROS message. Whereas information type of pointclouds and IMU of VLP is sensor_msgs/Imu and sensor_msgs/PointCloud2 respectively. Those standard information in bag is what the LeGO-LOAM can subscribe. Therefore in order to make sure the IMU and point cloud information can be send to LeGO_LOAM, those information should be converted to be standard.

-Topics

What the topics that LeGO_LOAM can subscribe is sensor_msgs/Imu and sensor_msgs/PointCloud2. But in ouster set, the topics about imu and point cloud are both ouster_ros/PacketMsg, which can not be subscribed by LeGO_LOAM directly.

-IMU

The term IMU stands for "Inertial Measurement Unit," and we use it to describe a collection of measurement tools. When installed in a device, these tools can capture data about the device's movement. IMUs contain sensors such as accelemeters, gyroscopes, and magnetometers. The original 3-axis angle velocity and accelerometer can be resolved to orientation angle by Velodyne IMU and then recorded in dataset. LeGO-LOAM is able to subscribe yaw roll pitch in VLP dataset directly, which is used for function updateInitialGuess in featureAssociation node so that provide a initial estimation for least square method to optimize transformation between two continuous frames. However,the Linear acceleration and angle speed contained in ouster data set can not be used for update initial guess. So the third package which is responsible for calculating orientation angle is necessary to make use of IMU data. The framework shown in fig3 is about How LeGO-LOAM subscribe point cloud data and IMU respectively based on the analysis above.

2.2 Steps for solving problem

Step 1

Convert pointcloud type and IMU information from PacketMsg to pcl::PointCloud and ROS messages. Sample data is recommended to visualize by ouster studio or ROS. Ouster studio is software to visualize the pointcloud output. Ouster provides sample code for connecting to the sensor, visualizing the output data, and interfacing with the popular ROS robotics suite. The source code repository can be found at: www.github.com/ouster-lidar/ouster_example This package include four nodes.

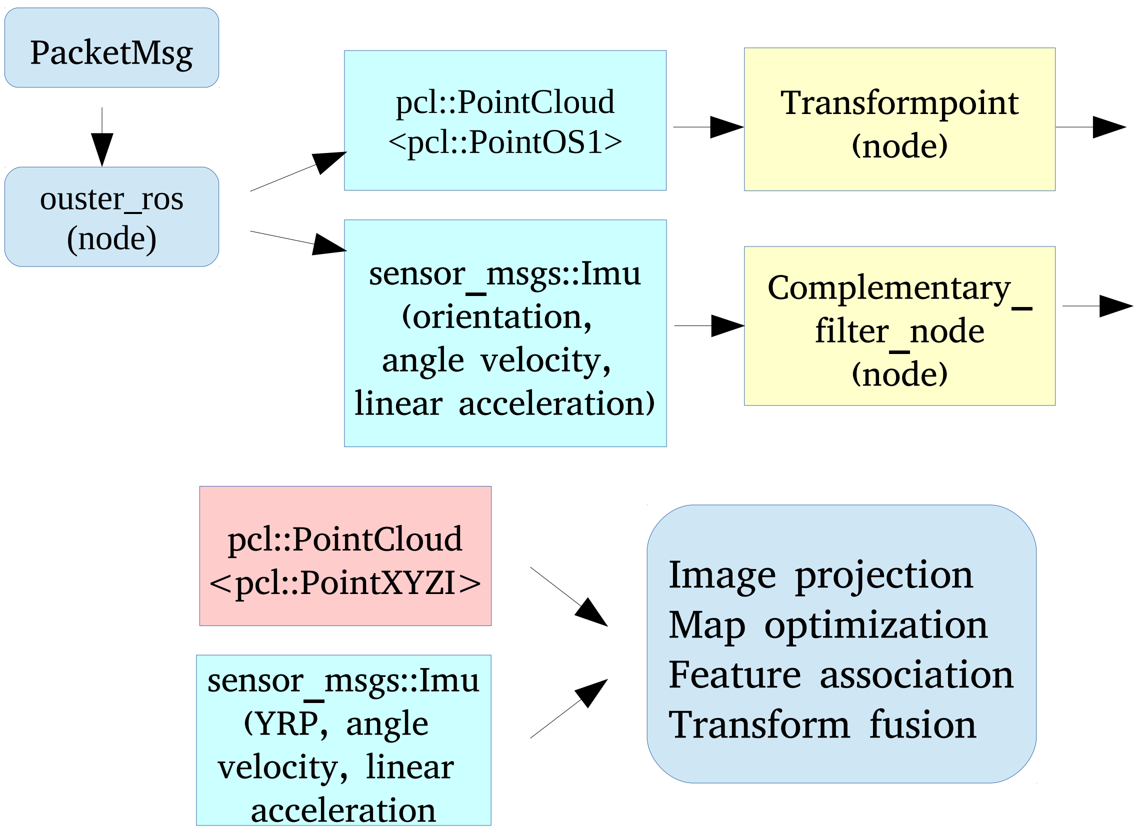

Fig 3 direction of information delivery

-os1_node: The primary sensor client that handles the sensor initialization and configuration as well as publishing the raw IMU and lidar packets

-os1_cloud_node: Converts the raw IMU and lidar packets to ROS IMU and PointCloud2 messages

-viz_node: Runs the provided visualizer displaying the pointcloud and the intensity/noise/range images

-img_node: Publishes the intensity/noise/range images as ROS image messages os1_cloud_node is node with function of converting pointcloud type to standard pcl version. The picture below is point cloud of ouster dataset displayed in openGl. Through diplaying point cloud in Rviz to make sure LeGO-LOAM can subscribe it directly.

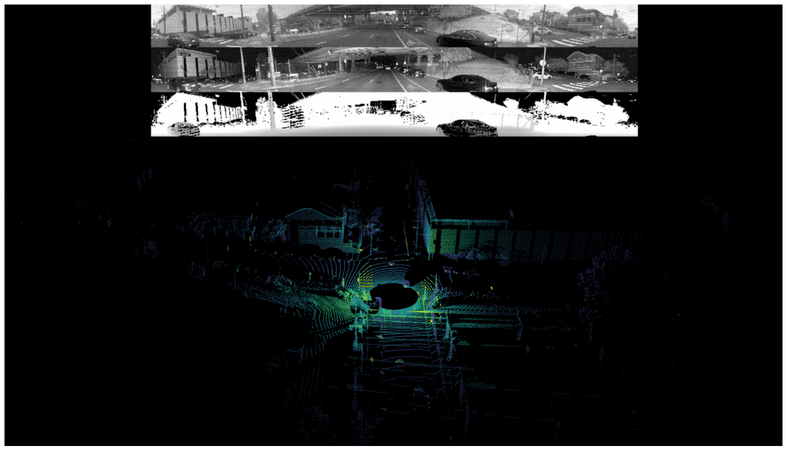

Fig 4 ouster dataset shown in openGL

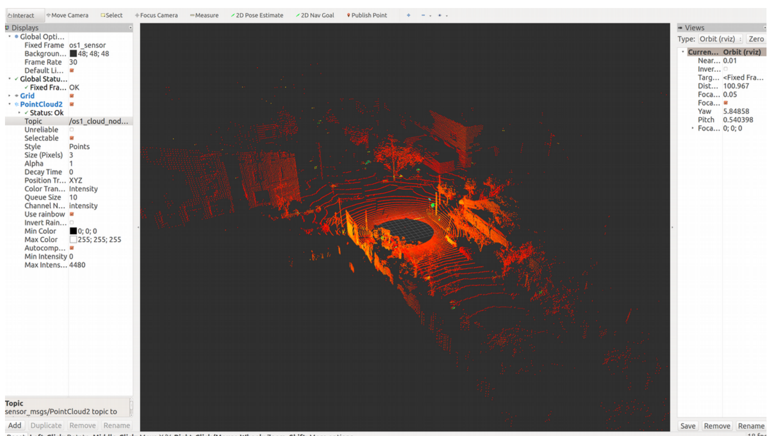

Fig 5 ouster dataset shown in Rviz

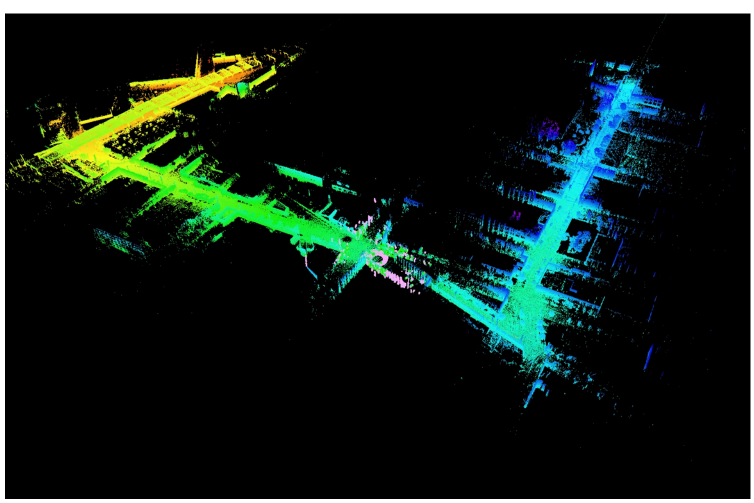

Fig 6 map building after only subscribing pointcloud

Step 2

Download gtSAM4.0.0_alfa2 before compiling LeGO-LOAM. Then use roslaunch lego_loam run.launch to run launch file so that open navigation package LeGO-LOAM. Enter the file which include ouster dataset and play existing bag files by rosbag play *.bag.

Fig 6 is final map building after running a whole ouster dataset,which is in the case of no IMU information input in LeGO-LOAM.

Step 3

Calculate orientation angle. The third package imu_tool can fuses angular velocities, accelerations, and (optionally) magnetic readings from a generic IMU device into a quaternion to represent the orientation of the device with respect to the global frame in real time, which will be applied into use to subscribe original information in ouster dataset and then publish a quaternion to represent the orientation of the device with respect to the global frame. In this package, there are three nodes.

-imu_filter_madgwick: a filter which fuses angular velocities, accelerations, and (optionally) magnetic readings from a generic IMU device into an orientation. Based on the work of [7].

-imu_complementary_filter: a filter which fuses angular velocities, accelerations, and (optionally) magnetic readings from a generic IMU device into an orientation quaternion using a novel approach based on a complementary fusion. Based on the work of [8].

-rviz_imu_plugin a plugin for rviz which displays sensor_msgs ::Imu messages

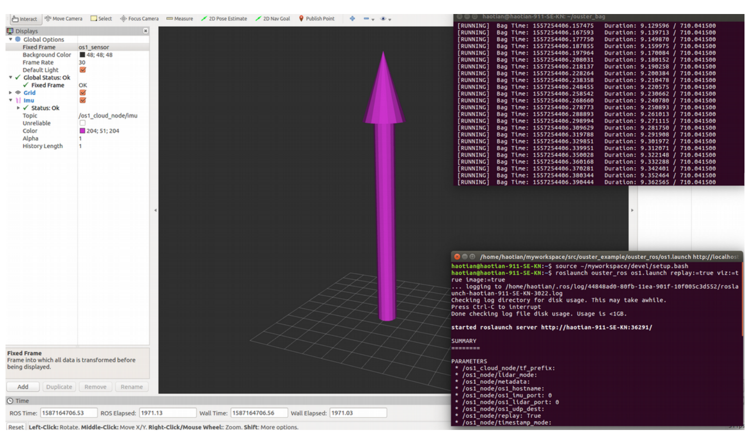

Fig 7 resolve orientation angle of ouster IMU

In Fig 7, Purple Arrow can simulate the actual orientation angle. The direction of the arrow shows yaw roll pith in real time after subscribing the orientation angle published by package imu_tools.

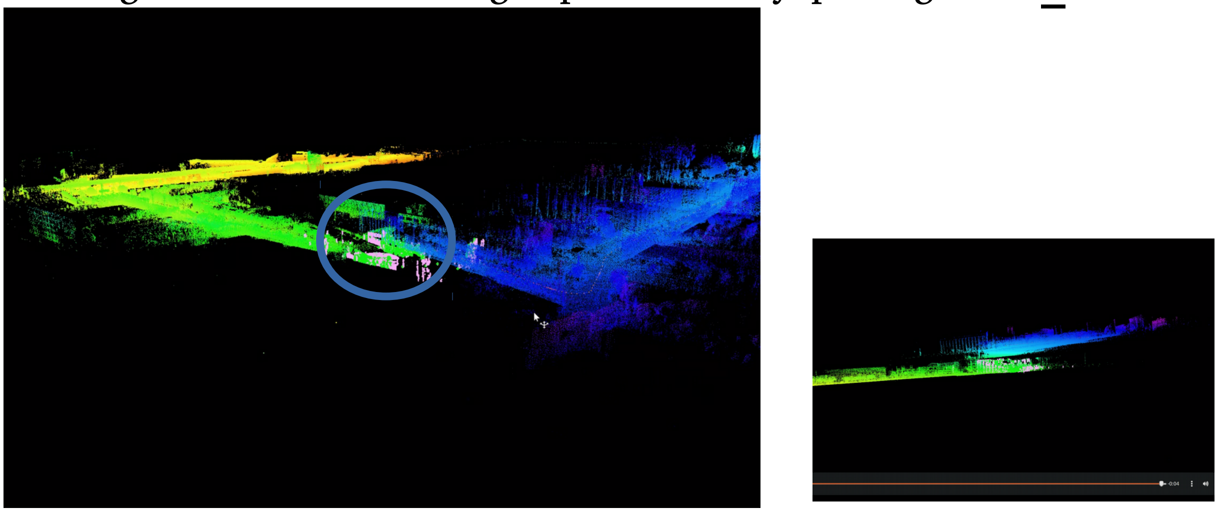

Fig 8 map building with orientation angle in use

the result in Fig 8 shows is what LeGO-LOAM subscribe point cloud and orientation of the lidar together. There should not have height gap between the start and end point. The reason lies in the principal disadvantage of an IMU, which is that they are prone to error that accumulates over time, also known as “drift.” Because the device is always measuring changes relative to itself (not triangulating against an absolute or known outside device), the IMU constantly rounds off small fractions in its calculations, which accumulate over time. Left uncorrected, these tiny imprecisions can add up to significant errors.

3.Conclusion

This research aim to validate whether the proposed LeGO-LOAM framework can achieve calculation in real time with high resolution Lidar in use. Through making use of ouster sample drive and imu_tool together, LeGO-LOAM can subscribe the original point cloud , angle velocity and linear acceleration together. However, Due to the drift error of IMU, there is a big height gap between the start and end position with subscribing point cloud and orientation angle which is resolved by the third package imu_tool.

References

[1] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," IEEE Trans. Robot., vol. 24, no. 5, pp. 1015–1026, Oct. 2008.
[2] D. Borio, L. Camoriano, and L. L. Presti, "Impact of GPS acquisition strategy on decision probabilities," IEEE Trans. Aerosp. Electron. Syst., vol. 44, no. 3, pp. 996–1011, Jul. 2008.
[3] M. Mozaffari, A. Broumandan, K. O'Keefe, and G. Lachapelle, "Weak GPS signal acquisition using antenna diversity," in Proc. Ubiquitous Positioning Indoor Navigat. Location Based Service (UPINLBS), 2014, pp. 11–18.
[4] S. Du, W. Sun, Y. Gao, and Z. Li, "An investigation on MEMS IMU error mitigation using rotation modulation technique," in Proc. 27th Int. Tech. Meeting Satell. Division Inst. Navigat. ION GNSS, vol. 3, Jan. 2014, pp. 1822–1838.
[5] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "Inertial odometry on handheld smartphones," in Proc. 21st Int. Conf. Inf. Fusion (FUSION), Jul. 2017, pp. 1–5.
[6] http://www.x-io.co.uk/open-source-imu-and-ahrs-algorithms/

[7] http://www.mdpi.com/1424-8220/15/8/19302