

TOWARDS AUTONOMOUS HIGH-DEFINITION MAP CONSTRUCTION

Junli Wu¹, Xiaoqing Wang^{1*}, Peng Zhang¹, Weiwei Song², Ming Chen³

¹National Geomatics Center of China, Beijing, China, (jlwu, xqwang, zhangpeng)@ngcc.cn.

²National Engineering Research Center of Satellite Positioning System, Hubei, China, sww@whu.edu.cn.

³Ministry of Natural Resource, Beijing, China, 18028483@qq.com.

KEY WORDS: autonomous, high-Definition, Map, Construction

ABSTRACT:

This paper presents a GPS constrained SLAM solution, which adds reliable GPS observations as key frames to the state-of-the-art SLAM algorithm Lightweight and Ground-Optimized Lidar Odometry and Mapping (LeGO-LOAM). As the GPS has a much higher frequency than that of lidar, we first assign each lidar frame with GPS time, then every 5 seconds, a reliable GPS observation is inserted as a key frame to the pose graph, then optimizes the pose estimation and updates the old key frames. We test our method with two platforms in real driving scene, and compare its performance with LeGO-LOAM, where LeGO-LOAM cannot acquire real-time efficiency. The difference of lidar odometry of our solution with compare to RTK-GPS is less than 0.4m, with the globally referenced map can be used for relocalization.

1. INTRODUCTION

While the concept of fully automated vehicles has been widely reported worldwide, there is still no companies or institutes who have made it possible. Much challenges remaining for this task, one of them is to localize vehicles globally in almost all the scenarios, on the city roads, country lanes or in the tunnels. Accurate position can be obtained by using GPS in open areas, however, in a complex environment, a variety of sensors need to be utilized to complement GPS blackouts or multipath problems. Light Detection and Ranging (LiDAR)s and video cameras are among the popular sensors mounted on cars. The typical application of these sensors is generating digital maps from a set of point clouds with the help of Simultaneous Localization and Mapping (SLAM), and great efforts have been devoted to achieving real-time SLAM on small car-like platforms or robots [1-3]. However, lots of unexpected problems will be encountered if we employ the mature algorithm on our vehicle within real-road environment. Firstly, the sensors on the small platform are usually cheaper and the output data rate is smaller, as some of the algorithms are designed for these low-performance sensors, they cannot deal with tremendous amounts of data in real case. Secondly, some of the solutions are only tested on robots with a low and almost constant speed, for example, results are only tested with an average speed of 1.3m/s in [1] and around 5 km/h in [3], while for the actual situation, vehicles are running much more faster, this is why some solutions cannot achieve the same accurate results on vehicles. Thirdly, as SLAM mainly targeted for indoor case without GPS aiding, many solutions are only tested indoor or within campus, the map are orders of magnitude smaller than the vehicles', besides, without the GPS constraint, the robots' maps are meaningless for vehicles. Therefore, our goal is to bring SLAM into actual autonomous driving scenes, which is to say, to satisfy the sensors with huge amounts of data, to achieve real-time solution in relatively high-speed case and to generate the 3D map with GPS constraint.

To tackle these limitations, the fusion of the SLAM with geo-referenced information has been widely studied. Cameras are widely accepted for cheap price and advantages for loop closure detection, as Bundle Adjustment (BA) is a refinement of the coordinates of a set of images from different viewpoints, several solutions [4-6] introduce additional information in the BA process. In [4] and [5], the BA error accumulation is reduced by fusion of GPS thus highly increase the location accuracy as well

as maintaining a globally referenced map. Owing to the bad output of the altitude from some low-cost GPS, [6] merge both GPS measurement and Digital Elevation Model (DEM) data into BA, and produces better result, the drawback is significant, DEM data is only available at certain areas. On the other hand, some solutions focus on eliminating long-term drifts in Visual Odometry (VO) part [7,8], the first one is a hierarchical system with local map aiding, also this method is only restricted to some certain areas like [6], the latter one modifies the VO pose using GPS measurement by means of a Kalman filter.

However, considering their extreme sensitivity to illumination and unreliable capabilities for lack of discriminant texture, this dissertation focuses on using lidar for perception and mapping. When without GPS assistance, some algorithms concentrate on localization and mapping locally like [9], with relocation of the vehicle on the map generated by the previous scans. As this process is done by matching of the current scan with the map, it is strictly limited by the resolution of the map, besides, the resulted dynamic map is only a surrounding over 100m around the vehicle, this solution is unable to produce a highly accurate positioning and globally mapping. Therefore, additional geo-information is introduced similar with above. A theoretical background of GPS-SLAM fusion is presented in [10], by means of Smoothing and Mapping (SAM), the GPS errors are used to augment SLAM, finally a probabilistic urban map based on landmarks is created with improved accuracy and robustness. As this solution highly relied on good and stable landmarks, much time will be taken before this process and it is not suitable for large-scale map generation. Therefore, [11] matches the specific wall information from point cloud with additional digital map information for localization and fuses DEM information to make a full 3D mapping, despite the sub-meter accuracy evaluated with RTK-GPS, need of prior knowledge making this method not sufficient for common application.

A Lightweight and Ground-Optimized Lidar Odometry and Mapping (LeGO-LOAM) is proposed in [1], as a modification of the state-of-the-art method LOAM [2], this method uses a two-step optimization for pose estimation. Planar features from the ground are first extracted, which is a good news for the flat cement or asphalt road, then the edge features are extracted from the segmented point cloud.

In this paper, an efficient strategy based on introducing GPS constraints into the LeGO-LOAM algorithm is purposed. The solution was examined with two different kinds of testing vehicles in real driving scenes, the positioning accuracy was

evaluated with RTK-GPS, while the map consistency was examined by means of Normal Distribution Transformation(NDT) [12]. The results indicate that our method can not only achieve the real-time efficiency but also a lane-level accuracy for autonomous vehicles.

2. SYSTEM HARDWARE

Two kinds of testing vehicles from a company called IN-Driving Tech were used with similar sensors, one is a normal passenger car as in Figure 1(a), the other is a container truck usually seen in ports, the container is always with the truck during the whole examination process.



Figure 1. Two kinds of testing vehicles

Identical 40 channel lidar Pandar40 from Hesai is fixed on the roof of the two vehicles, some important specifications are listed in Table 1.

The state and position measurement are from a intergeted IMU-RTK GNSS M39 from MPSTNAV, with the horizontal positioning accuracy less than 0.05m, vertical accuracy less than 0.1m, and the heading angle accuracy less than 0.5°. The GNSS measurement output rate is up to 200Hz, the working status of the GPS can be directly found from the output.

All the raw data and the algorithm itself are being processed in real time by a controller TITAN III made by IN-Driving Tech, with one AURIX MCU from Infineon and two Nvidia Jetson TX2.

Rotation Rate	10Hz
FOV(Vertical)	23°(-16° to 7°)
FOV(Horizontal)	360°
Measurement Range	Up to 200 m
Measurement Accuracy	±2 cm
Angular Resolution(Vertical)	0.33°

Table 1. Important specifications of Pandar40

3. GPS AIDED SLAM

3.1 System Overview

The diagram of the system framework is described in Figure 2. The system receives input from 10Hz lidar and 50Hz IMU-RTK GNSS and outputs 10Hz pose estimation. The input GNSS signal is first transformed into a pseudo global coordinate, then a new thread is added to the original LeGO-LOAM algorithm, with this thread constantly match every point cloud frame with GPS-IMU frame. A new key frame with GPS information is acceded to the factor graph every 5 seconds, optimizing the pose-graph and updating the locations of old key frames.

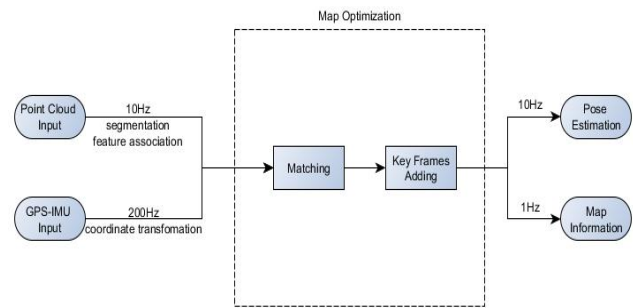


Figure 2. System Overview

3.2 Coordinate Transformation

As the transformation between coordinates is always the first issue to think over regarding autonomous driving, we choose same sensors with similar installation methods for different platforms for sake of saving efforts. Lidar provides the 3D representation of the surrounding environment with a lidar center based local coordinate, while the GNSS gives an absolute description in the global coordinate. The global position of lidar is the bridge for transforming the locally feature location into global coordinates. On account of the data rate of GPS-IMU is much higher than that of lidar, we can assign the nearest GPS position to lidar by means of time matching. We assume a lidar scan has the same position with a GPS epoch if the time difference between these two is smaller than 0.05s, then we can transform every point cloud into global coordinates with this information.

However, this will significantly increase the computational complexity with respect to the earth-centered coordinate. Therefore, we decided to use a pseudo global coordinate centered at the starting point of the vehicle, which also makes more convenience for vehicle based controlling and monitoring in a 2D coordinate. But this method is still not robust, as the following points compute the relative position to the origin, if this initial position is not correct, the errors will propagate to the entire estimation. Thus we choose a benchmark point to eliminate the possible deviations, we set the same benchmark point for the two platforms, and we transform the raw 3D GPS data to a 2D benchmark point centered coordinate at the very beginning of the algorithm, with x axis pointing the east, and y axis pointing the north. The z axis stays the same with RTK-GPS, which indicates the absolute height information.

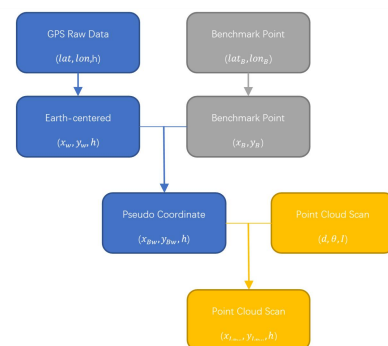


Figure 3. Coordinate transformation diagram

As shown in Figure 3, the latitude and longitude information of GPS raw data of the benchmark point is

first transformed into the Earth Centered Earth Fixed (ECEF) frame.

$$\begin{aligned} x &= (N + h) \cos(lat) \cos(lon) \\ y &= (N + h) \cos(lat) \sin(lon) \end{aligned} \quad (1)$$

$$N = \frac{a^2}{\sqrt{(a \cdot \cos(lat))^2 + (b \cdot \sin(lat))^2}}$$

where a is the equatorial radius, b is the polar radius.

Unlike the local ENU coordinate, the pseudo global coordinate is the difference between the position output (x_W, y_W) the benchmark point (x_B, y_B) ,

$$\begin{bmatrix} x_{BW} \\ y_{BW} \end{bmatrix} = \begin{bmatrix} x_W - x_B \\ y_W - y_B \end{bmatrix} \quad (2)$$

3.3 Key Frames Adding and Optimization

Iterative closet point (ICP) is the common algorithm for aligning two lidar scans at a point-wise level, also called registration. This method is used for adding new constraints after the two step Levenberg-Marquardt (L-M) optimization thus further eliminating estimation drift through pose-graph SLAM in LeGO-LOAM [1]. Besides, this is also very important for loop closure detection, with a successful loop closure test adding a loop closing constraint to the factor graph, updating the variables and reducing the estimation drift.

Our aim is to make the SLAM trajectory in a global sense, this is to say, every trajectory point should have a globally referenced coordinate, instead of the odometry frame. In light of the loop closure test process, which adds a constraint to the factor graph and carries out a constraint-based optimizing method, we add GPS constraints to the factor graph hence the pose estimation is moving towards the absolute position.

We can achieve our goal by adding keyframes, which works as the global constraint in the map optimization procedure in LeGO-LOAM. Along the vehicle trajectory, we add some reference frames with GPS location as keyframes, with some of these selected composing local maps (50m * 50m) for ICP algorithm, the others are stored and may be used for loop closure test.

(1) The first keyframe F_0 is assigned with the starting GPS position, and the transformation between this position and the benchmark point is added to the factor graph.

(2) Every five seconds, a new GPS position (good) will be added to the graph, containing the factor representing the new position with respect to the former keyframes. After that, the pose estimation is updated, and all the stored keyframes are relocated using the optimization result.

A factor graph describes the factorization of a function using variables and factors, pose graph is a special form of factor graph, where the variables are vehicle poses and the factors are corresponding pose transformation between these poses. Adding a new keyframe also adds a variable to the pose graph, and the new factor is the last transformation in company with estimated covariance.

The original LeGO-LOAM only uses a sliding window approach for holding the accumulated point cloud map around the vehicle, every time only a fixed-size sub map is visualable in rviz, which is not ideal for monitoring purpose. This is because LeGO-LOAM only updates a small region around its current position, we can modify this by concatenating this updated region into one big point cloud and publishing the complete map.

4. EXPERIMENTS

Two testing vehicle platforms are presented in this paper, with a normal passenger car and a container truck, the container is attached to the truck throughout every stage of the experiments. The algorithm is tested on the controller TITAN III using the robot operation system (ROS) [13] in Ubuntu-Linux.

In order to conveniently evaluate the accuracy of the algorithm and project the 3D point cloud into planar graph, we introduce a pseudo global coordinate system. However, as the accuracy of the origin of the coordinate will highly influence the whole process, we adopt a same benchmark point for the two experiments.

4.1 Passenger Car Test

We manually drive our vehicle on one of the main roads here, about 7 km, with traffic lights, buildings, lakes, vegetations and several viaducts around, and we choose to test at noon in weekdays to avoid congestions. The elapsed time is 503 s, with an average speed of 51 km/h. Figure 4(b) is the overall point cloud map generated by the algorithm.

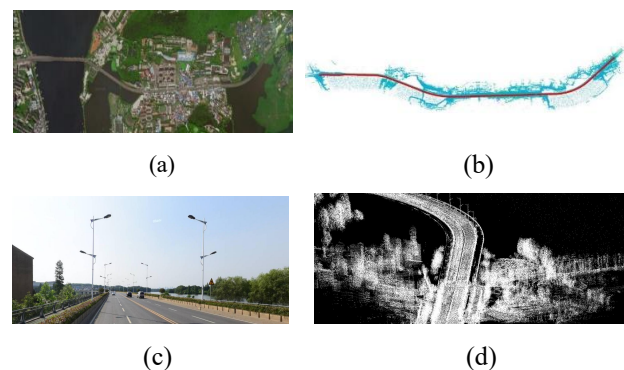
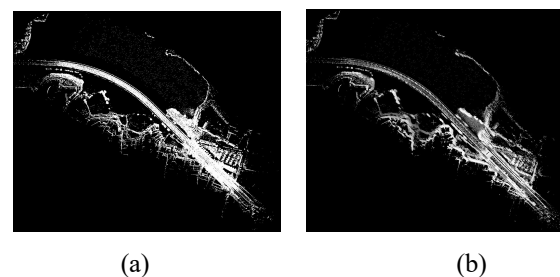


Figure 4. The testing scenario of the algorithm, (a) is the satellite map from Baidu map, (b) is the point cloud map generated by the algorithm, with the red line represented the trajectory from RTK-GPS. (c) is a planar image in (a), and (d) is the point cloud image of (c), with the road lamps clearly visualizable.

We also analyzed the LeGO-LOAM with the same datasets, with a comparison of the two results in Figure 5. Unlike the original LOAM which runs 10% of the real-time speed, both method is analyzed with full speed. We can notice the map is kind of blurred in Figure 5(b) compared with Figure 5(a), which is to say, some features are missing. Because when there is no GPS aiding, it is hard to find feature correspondences between two consecutive scans with limited overlap.



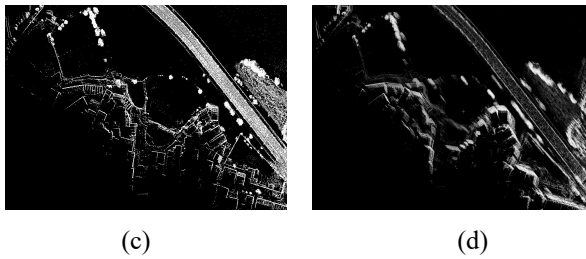


Figure 5. Results compared with LeGO-LOAM using the same dataset, (a) is the modified algorithm, and (b) is the output from Lego-LOAM. (c) is a little zoom in of (a), and (d) is the same part as (c) in (b).

Considering the extremely low possibilities for carrying out a loop closure test on the real roads and comparing the translational and rotational difference between the initial pose and final pose like what was done in [1]. We directly calculate the pose difference from SLAM result and RTK-GPS, and 10 tests were performed for examining the consistency, shown in Figure 6.

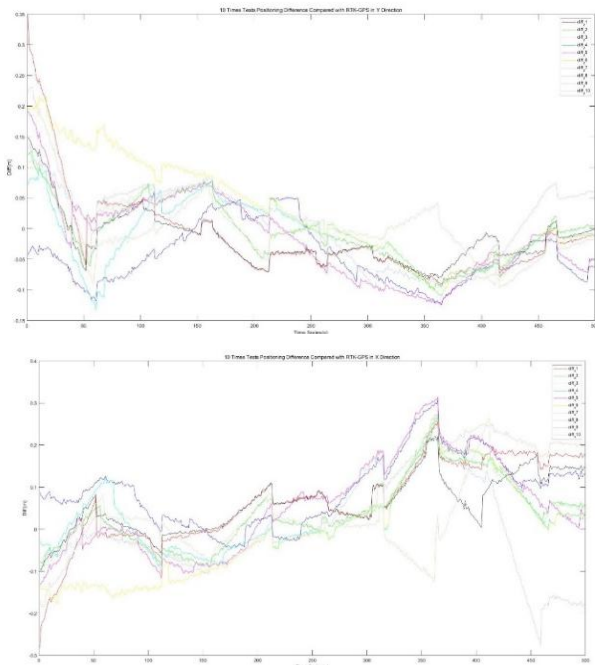


Figure 6. The 10 tests' error with compared to RTK-GPS, as the height information from GPS is straightly allocated to point cloud, we only take the planar errors into consider, with (above) the difference in X direction and (below) the difference in Y direction.

We can find from the Figure 6 that the errors in every direction is below 35cm, both the figure has some vibrations, which is because the periodic GPS intervention. Table 2 is a description of the mean error and the root mean square error (RMSE) of different tests. The errors between tests are caused from different feature extraction processes.

	$mean_x$	$RMSE_x$	$mean_y$	$RMSE_y$
TEST 1	0.0647	0.0952	-0.0125	0.0696
TEST 2	0.0400	0.0785	-0.0078	0.0490
TEST 3	0.0930	0.0822	-0.0445	0.0466
TEST 4	0.0376	0.0854	-0.0114	0.0500
TEST 5	0.0556	0.1110	-0.0159	0.0669
TEST 6	-0.0039	0.1163	0.0303	0.0817

TEST 7	0.0604	0.0639	-0.0192	0.0418
TEST 8	0.0607	0.1172	-0.0046	0.0666
TEST 9	-0.0037	0.1115	0.0126	0.0589
TEST 10	0.0194	0.0648	-0.0023	0.0353

Table 2. Mean errors (m) and the root mean square errors (RMSE) of the passenger car 10 tests

We can find from Table 2 that mean error in both direction is below 10cm, and the RMSE in Y direction is slightly smaller than that in X direction. The results indicate this system is able to provide sub-meter pose estimation with high consistency.

4.2 Container Truck Test

Different from the passenger car test which simulates the urban environment with many vehicles, we now test its performance in the outskirts with higher velocity. We drive the container truck on a suburb freeway connected to several country roads, with mountains, lakes, buildings around, and the GPS signal is not always good. The total length is approximately 15km, the time in all is 804s.

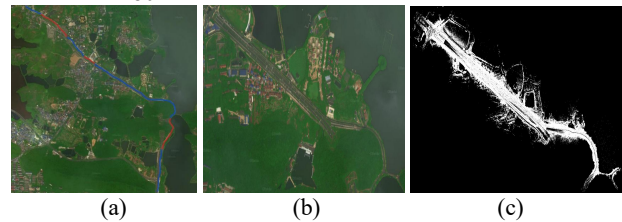
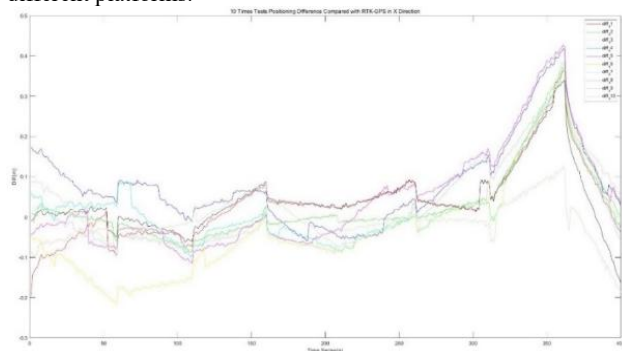


Figure 7. The testing scenario of the container truck, (a) is the overall length we have driven, with the blue lines indicated path with good GPS signals, red lines for bad signals. (b) is a partial sketch with good GPS signals, and (c) is the output map from the algorithm.

As this method omits the GPS constraints in poor signal areas, thus the system only relies Lego-LOAM for positioning and mapping in these environments, but with a view to the speed limit performance of the original algorithm, we only take a part of this scene shown in Figure 7(b) for analysis. This scene included a path about 7km, the time cost is 414s, the average speed is 61km/h, a little faster than the passenger car.

Similar with the passenger car experiment, we compute the difference of the SLAM trajectory with respect to the RTK-GPS, also 10 tests are performed to justify the consistency on different platforms.



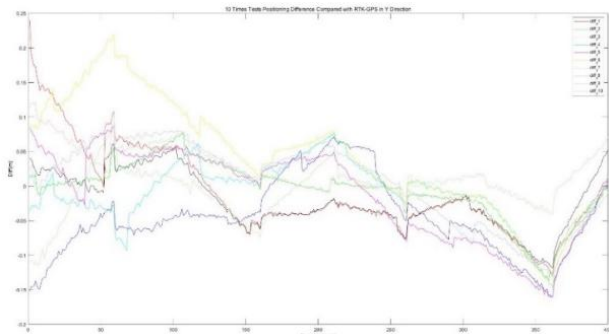


Figure 8. The container truck’s 10 tests error with compared to RTK-GPS, with (above) the difference in X direction and (below) the difference in Y direction.

	$mean_x$	$RMSE_x$	$mean_y$	$RMSE_y$
TEST 1	0.0360	0.0814	-0.0096	0.0566
TEST 2	0.0269	0.0792	-0.0068	0.0383
TEST 3	0.0800	0.0924	-0.0455	0.0489
TEST 4	0.0259	0.0861	-0.0097	0.0400
TEST 5	0.0439	0.1121	-0.0122	0.0583
TEST 6	-0.0258	0.1139	0.0423	0.0735
TEST 7	0.0419	0.0693	-0.0216	0.0357
TEST 8	0.0217	0.0890	0.0039	0.0484
TEST 9	0.0175	0.0970	0.0075	0.0525
TEST 10	-0.0013	0.0462	-0.0029	0.0314

Table 3. Mean errors (m) and the root mean square errors (RMSE) of the container truck 10 tests

The result of the container truck experiment shows a similar pattern with car experiment, all the mean errors are below 10cm, and the RMSE in Y direction is smaller than that in X direction. This is probably due to the not tremendous speed difference, only 10km/h. Thus we can say that our method has similar performance on different platforms if equipped with the same sensors.

4.3 Map Verification

The accuracy of lidar odometry has been presented above, and then we use two methods to validate the correctness and usability of the SLAM map.

Except from lidar and IMU-RTK GNSS, both platforms are also equipped with several video cameras for object recognition, so we can test the map accuracy by projecting the objects onto it and seeing if they stay the same with real world.

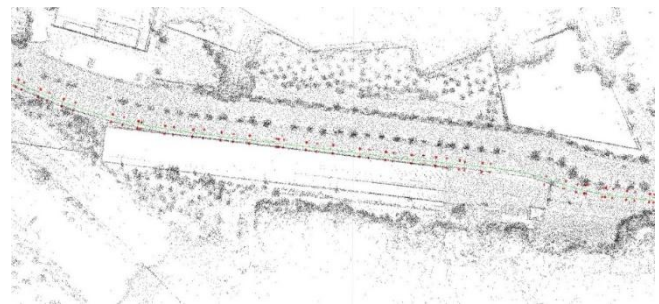


Figure 9. Projecting the lane markings recognized by the camera onto the generated map, (above) is the map from experiment 1 while (below) is the map from experiment 2, with the green line the GPS trajectory and red dots the lane markings.

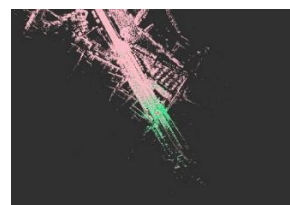
This Pandar40 lidar is not capable of recognizing lane markings on the test road, so we project the lane marking output from cameras onto the map as shown in Figure 9, obviously, one line of red dots matches well with the road boundary consists of point clouds. Thus, we can perform a further test with qualitative and quantitative analysis.

One big advantage of adding GPS constraints to map is making global positioning possible when using map-relative localization methods, which is done by registering the features from online scans with an offline map [14], the result is a rigid-body six DOF transformation parameters of the vehicle pose relative to the known map. As the map is globally referenced, we can acquire the vehicle position in the geodetic coordinate immediately.

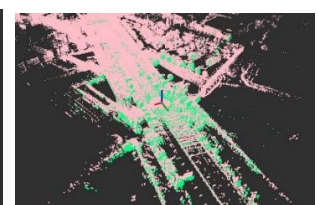
Here we use a ROS package `hdl_localization`, this package first executes the sensor pose with Unscented Kalman Filter (UKF), then a multi-threaded NDT scan matching is implemented between input point clouds and a global map. We have tested the two maps with the datasets accordingly. The fitness score starting from 0.1450, and always stay below 1 along the testing process.



(a)



(b)



(c)

Figure 10. The result of multi-threaded NDT matching of the passenger car test. The black circle in (a) indicates the area of (b) and (c), while (b) is the vertical view and (c) is a horizontal view

We also compare the trajectory from `hdl_localization` with RTK-GPS, the mean error in both directions is below 10 cm, which is good enough for lane-level localization.

5. CONCLUSIONS AND FUTURE WORK

We have proposed a GPS constrained SLAM algorithm in this dissertation. Real time odometry in the real-world scene can be achieved, with a lane-level accuracy which can be applied on intelligent vehicles. Besides that, this method can also provide a map with global constraints, thus can be used for relocalization. The results show this approach can achieve better map and odometry when compared with the state-of-the-art algorithm LeGO-LOAM. Future works including its application with different sensors, also with huge maps.

(1) We find that the 40-channel lidar is not sufficient to provide safe-enough data due to limited resolution in high speed case (>100km/h), an adult 30 meter away from the vehicle can be detected by at least 6 channels, while only less than 2 channels for a child because of its special design. So we are employing a new 300-channel lidar from Innovusion on our container truck. As its structure is quite different from any electromechanical lidars we are using nowadays, we need to modify our algorithm first.

(2) The two experiments only cover some certain roads shorter than 10km, the raw data is about 3.5GB per 100s, and the generated map is about 40MB per kilometer. This seems to have not much influence when driving downtown, but the real driving scenario requires at least a coverage of big cities, which might involve a thousands of kilometers journey. Concatenating such a huge map is a great challenge for our system, and further tests should be designed. We want to build the big map using small maps along the path, and with the help of some pre-set points, we use only the nearest part of map for map-based relocalization. Besides, we want to store only the corner feature for saving storage and efficient NDT matching.

(3) As this method connects the local SLAM mapping with the world geodetic coordinate, we can link the GPS-denied areas with the open areas, thus give the whole scenario a specific world coordinate. We decide to manually select certain points near the GPS-denied areas at first, such as tunnels, viaducts and underground garages. When in open areas, the vehicle continuously computes the distance to the certain points, once below a threshold, the vehicle then turns to the map-based localization. Because the NDT algorithm is limited to the amounts of features, and wrong matches are unavoidable, we intend to use deep learning-based feature matching and add visual features for secondary confirmation.

REFERENCES

Shan, Tixiao, and Brendan Englot, 2018. LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

Zhang, Ji, and Sanjiv Singh, 2014. LOAM: Lidar Odometry and Mapping in Real-time. *Robotics: Science and Systems*. Vol. 2.

Shamsudin, Abu Ubaidah, et al, 2018. Consistent map building in petrochemical complexes for firefighter robots using SLAM based on GPS and LIDAR. *Robomech Journal*, 5(1), 7.

Lhuillier, Maxime, 2012. Incremental fusion of structure-from-motion and GPS using constrained bundle adjustments. *IEEE transactions on pattern analysis and machine intelligence*, 34(12): 2489-2495.

Shepard, Daniel P., and Todd E. Humphreys, 2014. High-precision globally-referenced position and attitude via a fusion of visual SLAM, carrier-phase-based GPS, and inertial

measurements. *IEEE/ION Position, Location and Navigation Symposium-PLANS*. IEEE, 2014.

Larnaout, Dorra, et al, 2013. Vehicle 6-dof localization based on slam constrained by gps and digital elevation model information. *IEEE International Conference on Image Processing*. IEEE, 2013.

Agrawal, Motilal, and Kurt Konolige, 2006. Real-time localization in outdoor environments using stereo vision and inexpensive gps. *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 3. IEEE.

Schleicher, David, et al, 2009. Real-time hierarchical outdoor SLAM based on stereovision and GPS fusion. *IEEE Transactions on Intelligent Transportation Systems*, 10(3), 440-452.

Garcia-Favrot, Olivier, and Michel Parent, 2009. Laser scanner based slam in real road and traffic environment. *IEEE International Conference Robotics and Automation (ICRA09)*. Workshop on Safe navigation in open and dynamic environments Application to autonomous vehicles.

Carlson, Justin, 2010. Mapping large, urban environments with gps-aided slam. *Diss. figshare*, 2010.

Roh, Hyunchul, et al, 2016. Accurate mobile urban mapping via digital map-based SLAM. *Sensors*, 16(8), 1315.

Kato, Shinpei, et al, 2015. An Open Approach to Autonomous Vehicles. *IEEE Micro*, 35(6), 60-68.

Quigley, Morgan, et al, 2009. ROS: an open-source Robot Operating System. *ICRA workshop on open source software*. 3(2), 204-209.

Liu, R., Wang, J., & Zhang, B. (n.d.), 2013. High Definition Map for Automated Driving: Overview and Analysis. *Journal of Navigation*, 42(1), 1-18.