Lidar-only Localization with 3D Pose-Feature Map *

Chaokun Zhang[1,2], Taotao Shan[1], Bolin Zhou[1], and Yu Guo[1]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
[2] China Xiong'an Group Digital City Technology Co., Ltd
{zhangchaokun, starlight, zhoubolin, guoyu001}@tju.edu.cn

Abstract. Real-time and accuracy are the two most important indicators for unmanned vehicle localization. In this paper, we propose a novel map representation and its corresponding Lidar-only localization framework. In essence, we first extract geometric features from Lidar key-frames, and bundle these features with their observation poses (i.e., ground truth) to form a prior map named *Pose-Feature Map*. Then, the position of vehicle will be achieved by integrating Lidar-Odometry (LO) and Map-Matching (MM) with the Pose-Feature Map. In our framework, these two solutions are complementary. LO provides smooth and real-time pose estimation for MM, while MM can correct the accumulated drift of LO. During MM, we adaptively generate local maps to replace the global map for matching. Therefore, our proposed framework can further reduce the mismatch between the current frame and the map while maintaining low computational complexity. We demonstrated the framework in the KITTI dataset. The results confirm that our approach is superior to independent localization solutions in terms of real-time and accuracy.

Keywords: localization · Lidar Odometry · Pose-Feature Map matching · integration

1 Introduction

As one of the key technologies of unmanned vehicles, localization has a variety of mature technical solutions. However, due to the challenges posed by the environment and sensors, no localization solution can be applied to all scenarios. For example, the Global Navigation Satellite System (GNSS) is widely used in most open environments, but due to signal occlusion and canyon effects, it cannot be used in the closed or semi-enclosed environment (e.g., the garage, and the

* This research is supported in part by National Key R&D Program of China under Grant 2019YFB2102400, in part by China Postdoctoral Science Foundation 2020M680906, in part by Hebei Province High-level Talent Funding project B202003027, in part by National Natural Science Foundation of China under Grant 61832013.

urban building block). Therefore, in order to solve the localization problem of unmanned vehicle, we need multiple solutions to achieve localization redundancy.

Our research motivation comes from the specific requirements of Lidar for vehicle self-location. In the past decade, various Lidar-only localization solutions have emerged. It can be simply divided into Simultaneous Localization And Mapping (SLAM) and Map-Matching (MM) solutions. Lidar-SLAM solutions, such as [3], [7], [12], [14], estimate 6 Degrees of Freedom (6-DOF) of motion by comparing the continuously scanned point clouds frame by frame and registering them finely. Therefore, their localization frequency is equal to the Lidar, and their real-time performance is better. Nevertheless, all SLAM solutions have their inherent flaws. In long-distance driving, the lack of prior constraints will cause errors to accumulate, and the trajectory drift will become larger and larger, until the localization is lost. Lidar-MM solutions, such as [13], [9], [8], register the Lidar scans with the prior map to obtain the localization results, so that the above-mentioned error accumulation does not occur. However, a large prior map will lead to heavy computation and poor real-time localization, and its repetitive structure will lead to mismatches. This makes it unsatisfactory in practical applications.

To address these challenges above, we propose a novel map representation called **Pose-Feature Map** to reduce the computation and mismatch in MM, and a new framework that integrates MM and LO to increase the frequency of localization. During operation, the Pose-Feature Map change MM from frame-to-global-map matching into adaptive frame-to-local-map registration, thereby reducing the computation and avoiding mismatches caused by repeated environmental structures. Moreover, interpolating the SLAM localization result into MM can make it smoother and more real-time. Also MM localization results can correct the cumulative drift of SLAM. Their complementary allows us to obtain real-time and accurate localization results.

The remaining part of this paper is organized as follow. Related research work of Lidar localization solutions is presented in Section 2; Section 3 introduces the framework architecture and the background of Lidar-only odometry; Section 4 presents the Pose-Feature Map and the special Map-Matching base it in detail; the experiment and conclusion are located in the Section 5 and 6, respectively.

2 Related Work

Lidar is considered to be an indispensable sensor for future unmanned vehicles. How to apply Lidar as the only sensor to the localization system, in recent years, three methods have been mainly proposed: Lidar-SLAM, Lidar-MM, and integration scheme.

Lidar-SLAM is a branch of the SLAM system. Among the Lidar-SLAM solutions in recent years, the most popular one is LOAM [14]. It performs down-sampling through curvature features, and converts the real-time registration from point-to-point to point-to-line/surface in iterative optimization. LOAM has maintained a long-term leading position in the KITTI test. In LeGO-LOAM [12],

the outlier and ground points in scans were segmented out before registration, besides loop detection was added. This make LeGO-LOAM perform better than LOAM in certain environment. However, their biggest problem is drift, which can also be clearly observed in the experiment part.

Lidar-MM will continuously corrects drift errors during localization. The registration method is a decisive factor. With the development of deep learning on 3D point clouds registration, L3-Net [9] and DeLS-3D [13] have adopted artificial neural network methods for mapping between scans and the map. However, these AI methods are not interpretable and require expensive hardware to satisfy the computation, so they are not suitable for vehicle platforms. In contrast, the traditional method, such as ICP [11] and NDT [1], are lighter and more interpretable. For example, HDL [8] exploits multi-thread NDT. However, the general method of extracting local maps based on a certain radius can easily lead to loss of localization in many cases. In PoseMap [6], it innovatively divides global-map into surfel sub-maps based on the ground truth to solve the above problems, which inspired us. However, in vehicle platform, the frequency of MM still cannot meet the driving requirements.

Integration is actually a fusion of the above two solutions. The greatest advantage is to make them complement each other and eliminate their own shortcomings. In LOL [10], a global positioning method that merged Segmap [5] and LOAM is proposed to solve the long-distance drift of LOAM. [4] also integrates information from complementary nodes such as Lidar global matching and Lidar inertial odometry to achieve accurate and smooth position estimation. However, they are all complex and non-lightweight.

Our solution is also the integration of SLAM and MM. In SLAM, we exploit the LO algorithm part in LOAM and LeGO-LOAM; and in the MM scheme, we adopt the idea of sub-map similar to PoseMap and improved it. The mapping scheme relies on the ground truth to provide a coherent initial map that is sufficiently enough for waypoint localization. We believe that by combining LO and a prior map constraints, we can complete high-frequency and accurate localization in a long period of time.

3 Framework Architecture And Background

3.1 Framework Architecture

As shown in Fig. 1, the framework requires two inputs for operation. One is real-time scans generated by Lidar. These input point cloud frames will be processed and filtered by the pre-processing nodes to generate features. Another one is a special prior map called Pose-Feature Map. It will dynamically generate local maps based on some parameters during the running of the framework.

Next, we exploit two strategies to perform localization. LO maintains high frequency motion estimation. The matching module intermittently matches the scanned features and generated local maps to obtain the discrete anchor position of the vehicle.

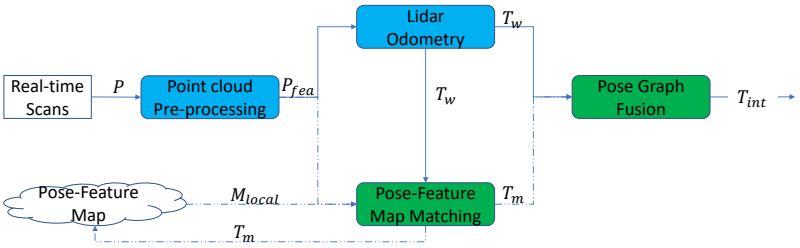


Fig. 1. Structure of Lidar-only localization framework with Pose-Feature Map. The white part is input data, the blue modules represent the LO algorithm from LOAM and LeGO-LOAM, while the green modules are our novel map matching and integration method. The localization results of LO and MM are T_w and T_m respectively. M_{local} is the local maps generated by T_m and global map. In addition, the solid line represents the high frequency, and the dashed line represents the low frequency, which are 10Hz and 1Hz, respectively.

Then, we apply pose graph optimization to further correct the discrete mapping position. Finally we combine the lidar odometry and the optimized mapping result through interpolation, and publish it as the trajectory of the vehicle.

3.2 Lidar-only Odometry Background

The main task of LO is to estimate high frequency self-motion based on the data flow transmitted by Lidar. Here we apply a more mature algorithm from LOAM [14] and LeGO-LOAM [12].

Point Cloud Pre-processing. Before performing the localization algorithm, the framework should extract features from real-time frames to reduce the computation in subsequent nodes. LOAM extract features P_{fea}^t in Lidar scans P^t base on the smoothness of each point. Furthermore, LeGO-LOAM noticed the interference of points generated by the ground, distant objects or plants, and designed a clustering algorithm to detect and segment them.

Lidar Odometry. The way that LO calculates the local movement T_l^t of Lidar between two consecutive scans is to register the features P_{fea}^{t-1} and P_{fea}^t. Both LOAM and LeGO-LOAM use KD-tree to find the corresponding point pairs, and then use iterative algorithms (e.g., Levenberg-Marquardt) to minimize their distance to calculate the relative motion between two frames. The whole process can be expressed as

$$\min_{T_t^l} d\left(T_t^l, X, C\right) \tag{1}$$

where X is a point in P_{fea}^t, C represent the correspond target in P_{fea}^{t-1}, and function d represent their distance.

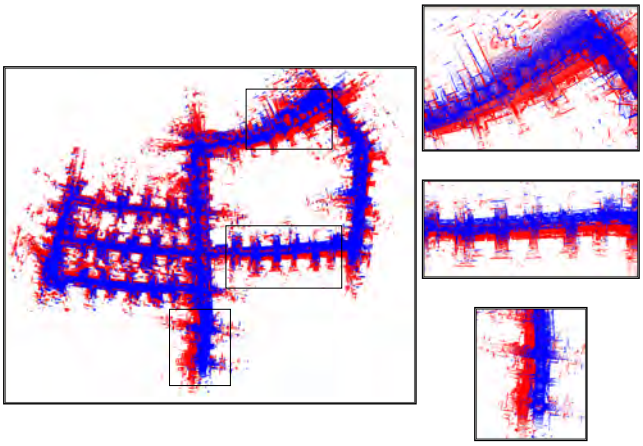


Fig. 2. Comparison of the maps generated by LeGO-LOAM and ground truth. The ground truth map (The driving time is $4'36''$ and the track length is 2200m.) is visualized with red points, while blue is LeGO-LOAM. The larger figure on the left is a global view, and the other three on the right are enlarged partial views.

By continuously multiplying all the transformation relations T_l^t, the vehicle's odometry world track estimation result T_w^t can be obtained in (2).

$$T_w^t = \prod_{i=1}^{t} T_l^i \tag{2}$$

4 Pose-Feature Map Localization

In the case of lidar-only, SLAM is unreliable. As shown in Fig. 2, in long sequence, the map and its trajectory generated by LeGO-LOAM have obvious drifts. For this reason, many researchers choose to treat ground-truth maps as prior benchmarks and use MM algorithms (e.g., NDT [1] and ICP [11]) to complete vehicle localization.

However, we found that in the real-time registration process of sans and global maps, not only the computation is heavy, but also the area in the map that does not need to be registered will seriously interfere with the result, especially when the surrounding environment changes drastically. In addition, the registration result between the local maps generated by general methods and the vehicle Lidar perception is also very poor. Based on long-term research and inspiration from PoseMap [6] we created the Pose-Feature Map. The way it generates a local map is to extract several point cloud frames near the position of the vehicle on the map. Compared with common methods, it can not only ensure the validity of local maps, but also reduce the computation of calculation.

Fig. 3. Pose-Feature Map composed of key-frames (light) and their observation poses (dark).

4.1 Pose-Feature Map

The point cloud map is a layer of a high definition (HD) map used for autonomous driving. It consists of a large number of points. Their coordinate value represent the geometric information of the three-dimensional world, and provide powerful and indispensable prior 3D scene knowledge. Obviously, the construction of a normal point cloud map M generated by splicing frames f_i collected from the same or different Lidars, vehicles and even at various times. Then we set a global coordinate system, and transform these key-frames to $f_i^{'}$ according to their pose T_i from GNSS or Surveying and Mapping Engineering,

$$M : \left\{ f_1^{'}, f_2^{'}, \cdots, f_n^{'} \right\}, i \in N, f_i^{'} = f_i \cdot T_i \tag{3}$$

Our Pose-Feature Map has the normal duty of a point cloud map: represents the surrounding environment. Specially, each feature point has it's observation pose after the map is formed. This is visible in Fig. 3. Points with the same observation pose constitute a node. The general data structure is as follow,

$$M : \left\{ \left(f_1^{'}, T_1 \right), \left(f_2^{'}, T_2 \right), \cdots \left(f_n^{'}, T_n \right) \right\} \tag{4}$$

For the purpose of reducing MM computation, we adopt two methods to reduce the map size, namely key-frame extraction and down-sampling. The extraction method is mainly based on the time interval. Since the speed of the vehicle is slow when collecting road curve features, details of the corner will be strengthened, although this will cause uneven map density. In order to better match the features extracted in pre-processing step, down-sampling strategy for generating Pose-Feature Map is the same.

4.2 Adaptive local-map matching

The main advantage of the Pose-Feature Map is the convenience and efficiency of fragmentation. When constructing the prior map, we also store all the key-frame poses in the KD-tree. With the vehicle localization at the last moment, we can quickly select the nearest key-frames to generate an adaptive local map M_{local}. In addition, our Pose-Feature Map matching can trade with low frequency for high accuracy. For example, multiple iterations and finer granularity can be set.

Next, we also need to consider the issue of selecting the number of frames. In our research, we found that there are two factors that affect the local map, namely driving environment and operating parameters.

Driving environment. We simply divide them into straight or curved roads, open or closed environments. Generally, the changes between consecutive frames of a straight road or an open environment is much smaller than that in curved or closed environment. Therefore, when the vehicle is turning or driving in a closed environment, more key-frames should be selected to generate a local map, and when driving in a straight or open environment, fewer should be selected.

Operating parameters. These parameters include vehicle speed, map matching frequency and key-frame distance, which should be adapted to the framework. Reflected in adaptive matching process, the proportion of overlap area between the vehicle perception and local map should be large enough. That is, the adaptive local map must cover the area that the vehicle may reach in the next cycle, while avoid mismatches caused by oversize.

Fig. 4 nicely shows the advantages of our novel Map. Compared with the method that roughly selecting a point cloud within a certain range (e.g., 50m), the adaptive local map we extracted overlaps well with the vehicle's perception, simultaneously the point cloud size is reduced by 70%.

Our next task is to register $P_{fea}^{k}(k \in t)$ with the M_{local}^{k} fragmented from Pose-Feature Map to obtain the anchor MM position T_{m}^{k} of the vehicle by NDT [1]. This is also the purpose of the entire matching node. Where k represents a certain moment in the time stream t. In this paper, the interval between k and $k-1$ is 1 second, and the interval between t and $t-1$ is 0.1 second. Their initial values are 0s.

4.3 Integrating Localization

During the Pose-Feature Map matching process, our global localization result T_{m}^{k} is obtained through NDT registration. Here, we will further optimize the NDT results with pose graph to increase confidence. Then, we integrated the odometry to increase frequency.

Optimization. We know that the errors come from the short-time (1 second) drift of LO and the NDT algorithm. Now we define the results of MM and odometry as vertices and edges to construct a pose graph. Their errors can be expressed as,

$$r_k = \left(T_w^k\right)^{-1} \cdot T_w^{k-1} \cdot \left(T_m^{k-1}\right)^{-1} \cdot T_m^k \tag{5}$$

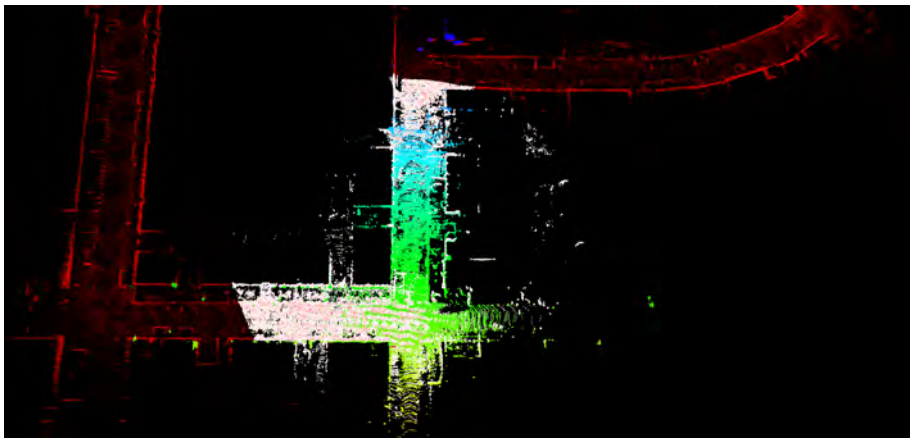


Fig. 4. Comparison of the points handed by radius search (white) and our Pose-Feature Map (green). The red points are the global map.

where the residual cost in k-th cycle is r_k. The non-linear least squares optimization is solved by using the GTSAM [2] solver, and get the optimized result defined as T_o^k.

Fusion. In our proposed framework, the frequency of odometry is the same as Lidar operating frequency. The integration process is actually interpolation. We treat the T_o^k as a anchor localization, the odometry result T_w^t continuously interpolated until a new MM optimization result is obtained. The method is expressed as follow,

$$T_{int}^t = T_o^k \cdot \left(T_w^k\right)^{-1} \cdot T_w^t \tag{6}$$

where T_{int}^t is the final output of our entire framework and consistent with the map coordinate system.

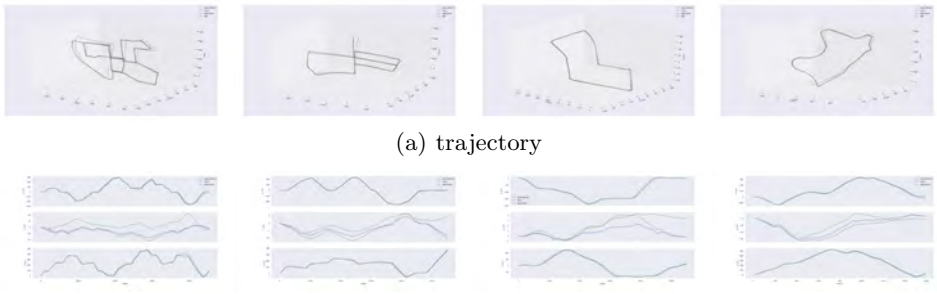
5 Experiments

5.1 Hardware, Datasets and Parameter Setting

The computing platform is NVIDIA JETSON AGX XAVIER, equipped with ARMv8 Processor rev 0(v8l), 32GB of RAM. Our framework has been tested many times on the KITTI dataset. The driving distance in data sets range from 800m to 3700m. Their environment is variable, and full of curves. The trajectories are also diverse, such as straight-line, closed-loop, and regional repeated driving.

5.2 Localization Performance

Fig. 5 is the quantitative result of our method compared with the original LeGO-LOAM algorithm, MM with Pose-Feature Map and the ground truth in certain



(a) trajectory

(b) error

Fig. 5. The result of KITTI Drive 00, 05, 07, 09.

datas. The Fig.5a shows the trajectory comparison that can be clearly seen that the vehicle trajectory calculated by our method and MM coincide with the ground truth, which corrects the drift in LeGO-LOAM. The Fig.5b shows the error comparison of the positioning results for each frame. In these datas, the drive distance are 3724m, 2205m, 695m and 1705m respectively. The LeGO-LOAM deviated by 23.4m, 23.93m, 4.19m and 22.57m, and our result (same as MM) are 9.57m, 2.94m, 0.45m and 6.13m. The above results can be concluded that: compared with SLAM, MM and our framework can effectively reduce drift.

Tab. 1 shows the results of several localization frequencies. It can be seen that the frequency of localization of our framework is consistent with LeGO-LOAM, and is almost consistent with the operating frequency of Lidar. However, the frequency of MM is only about one-tenth. It is worth mentioning that under the same NDT parameters, this computing platform can no longer increase the MM frequency. This also shows that if you completely rely on MM, the localization frequency will be greatly reduced.

6 Conclusion

This paper proposes a novel map representation called Pose-Feature Map for the map matching solution, which bundles key-frames with their correspond observation poses. In the registration step, the module extracts a more effective local map from it, which improves the localization accuracy and reduces the

Table 1. Localization frequency of each Drive sequence with different solutions.

solution\sequence	Drive 00 (470.4s)	Drive 05 (287.3s)	Drive 07 (113.1s)	Drive 09 (164.8s)
Ground truth	9.65Hz	9.61Hz	9.73Hz	9.65Hz
LeGO-LOAM	9.65Hz	9.61Hz	9.72Hz	9.64Hz
ILM(ours)	9.65Hz	9.61Hz	9.72Hz	9.64Hz
Pose-Feature Map matching	1.00Hz	1.00Hz	1.00Hz	0.99Hz

computation of calculation. At the same time, we integrated the above Pose-Feature Map-Matching and Lidar-Odometry into a new framework, enabling the two localization methods to make up for their respective shortcomings. The KITTI's data set confirms that our framework performs well in terms of accuracy and real-time.

References

1. Biber, P., Strasser, W.: The normal distributions transform: a new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). vol. 3, pp. 2743–2748 vol.3 (2003). https://doi.org/10.1109/IROS.2003.1249285
2. Dellaert, F.: Factor graphs and GTSAM: A hands-on introduction. In: (2012)
3. Deschaud, J.E.: Imls-slam: Scan-to-model matching based on 3D data. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 2480–2485 (2018)
4. Ding, W., Hou, S., Gao, H., Wan, G., Song, S.: Lidar inertial odometry aided robust lidar localization system in changing city scenes. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 4322–4328 (2020)
5. Dubé, R., Cramariuc, A., Dugas, D., Nieto, J., Siegwart, R., Cadena, C.: SegMap: 3D segment mapping using data-driven descriptors. In: Robotics: Science and Systems (RSS) (2018)
6. Egger, P., Borges, P.V.K., Catt, G., Pfrunder, A., Siegwart, R., Dubé, R.: Posemap: Lifelong, multi-environment 3D lidar localization. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3430–3437 (2018)
7. Gentil, C.L., Vidal-Calleja, T., Huang, S.: IN2LAMA: Inertial lidar localisation and mapping. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 6388–6394 (2019)
8. Koide, K., Miura, J., Menegatti, E.: A portable 3D LIDAR-based system for long-term and wide-area people behavior measurement. International Journal of Advanced Robotic Systems **16**(2), 1–13 (2019)
9. Lu, W., Zhou, Y., Wan, G., Hou, S., Song, S.: L3-net: Towards learning based lidar localization for autonomous driving. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6392–6391 (2019)
10. Rozenberszki, D., Majdik, A.L.: LOL: Lidar-only odometry and localization in 3D point cloud maps. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 4379–4385 (2020)
11. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Proceedings Third International Conference on 3-D Digital Imaging and Modeling. pp. 145–152 (2001). https://doi.org/10.1109/IM.2001.924423
12. Shan, T., Englot, B.: Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4758–4765 (2018)
13. Wang, P., Yang, R., Cao, B., Xu, W., Lin, Y.: DeLS-3D: Deep localization and segmentation with a 3D semantic map. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5860–5869 (2018)
14. Zhang, J., Singh, S.: LOAM: Lidar odometry and mapping in real-time. In: Robotics: Science and Systems (RSS) (2014)